UNIVERSITÀ CA' FOSCARI VENEZIA

DOTTORATO DI RICERCA IN INFORMATICA, 32° CICLO

(A.A. 2016/2017 – 2019/2020)

Structured-Light 3D Reconstruction and Applications

SETTORE SCIENTIFICO DISCIPLINARE DI AFFERENZA: INF/01

TESI DI DOTTORATO DI MARA PISTELLATO
MATR. 839976

TUTORE DEL DOTTORANDO
Dr. Filippo Bergamasco

COORDINATORE DEL DOTTORATO
Prof. Agostino Cortesi

December, 2019

Author's e-mail: mara.pistellato@unive.it

Author's address:

Dipartimento di Scienze Ambientali, Informatica e Statistica
Università Ca' Foscari Venezia
Via Torino, 155
30172 Venezia Mestre – Italia
tel. +39 041 2348465
fax. +39 041 2348419
web: http://www.dais.unive.it

You know who you are.
Thank you.

- Mara

Abstract

An increasing number of modern Computer Vision applications rely on the usage of depth data, allowing the accomplishment of several tasks otherwise being complex or impractical. As a consequence, a large amount of 3D acquisition methods have been proposed in literature, each one designed for a specific objective. In particular, structured-light approaches employ a projector to cast additional information over the scene to compute correspondences between two views and obtain a dense reconstruction.

This thesis presents two main contributions in the field of 3D reconstruction: first, some improvements over a state-of-the-art structured light technique called fringe projection are presented. We start from a theoretical characterization of the acquired signal and formulate a novel phase unwrapping technique offering high resilience to noise and outliers. Additionally, we propose a code recovering technique and a phase correcting method to further improve the final reconstruction accuracy. The second main contribution involves some real-world applications exploiting 3D reconstruction techniques in heterogeneous fields. In particular, we present a cylinder extraction technique designed to work in industrial settings where non-oriented, noisy point clouds are acquired from the scene and a structured-light approach for micrometric surface reconstruction. Another application involves the employment of the described techniques in the field of cultural heritage, where the final outcome yields to a reverse engineering process in addiction to the main preservation purpose.

Contents

List of Figures

Published Papers

[1] MARA PISTELLATO, FILIPPO BERGAMASCO, ANDREA ALBARELLI, ANDREA TORSELLO Cylinders Extraction in Non-oriented Point Clouds as a Clustering Problem, *Pattern Recognition* (under review)

[2] MARA PISTELLATO, FILIPPO BERGAMASCO, LUCA COSMO, ANDREA GASPARETTO, ANDREA ALBARELLI, ANDREA TORSELLO Robust Phase Unwrapping by Probabilistic Consensus, *Optics and Lasers in Engineering 121*, 2019

[3] MARA PISTELLATO, ANDREA ALBARELLI, FILIPPO BERGAMASCO, ANDREA TORSELLO Robust Cylinder Estimation in Point Clouds from Pairwise Axes Similarities, *International Conference on Pattern Recognition Applications and Methods (ICPRAM)*, 2019

[4] MARA PISTELLATO, ANDREA ALBARELLI, FILIPPO BERGAMASCO, LUCA COSMO, ANDREA GASPARETTO, ANDREA TORSELLO Stochastic Phase Estimation and Unwrapping, *International Conference on Pattern Recognition Applications and Methods (ICPRAM)*, 2019

[5] MARA PISTELLATO, LUCA COSMO, FILIPPO BERGAMASCO, ANDREA GASPARETTO, ANDREA ALBARELLI Adaptive Albedo Compensation for Accurate Phase-Shift Coding, *International Conference on Pattern Recognition (ICPR)*, 2018

[6] MARA PISTELLATO, FILIPPO BERGAMASCO, LUCA COSMO, ANDREA GASPARETTO, DALILA RESSI, ANDREA ALBARELLI Neighborhood-Based Recovery of Phase Unwrapping Faults, *International Conference on Pattern Recognition (ICPR)*, 2018

1

Introduction

During the last years, the convenience resulting from the availability and processing of 3D data has become a major driving force in recent Computer Vision research. Indeed, depth data and their integration with classical imaging techniques allows the fulfilment of several tasks which were challenging or infeasible in the past.

Many Computer Vision applications involve massive usage of 3D data in a number of situations, for instance in object classification [206] and segmentation [60]. In particular, recent improvements on Deep Convolutional Neural Networks allow the fusion of data coming from imaging devices and 3D sensors to perform the same tasks [168,205] achieving high accuracy levels. Popular applications of such techniques include autonomous driving [47], human body detection [73] and tracking [149], or consumer entertainment applications [76, 221].

Moreover, the increased accessibility given by the relatively low cost hardware and software allowed the spread of such techniques in multiple disciplines. Indeed, three-dimensional data are nowadays widely employed in a variety of fields, thanks to the wide range of practical and analytical tasks which can benefit from the additional information otherwise discarded in 2D imaging. In industrial settings, 3D data can be thoroughly exploited to improve the performances of inspection systems [7, 133] or to enable accurate robot control [155]. Other examples include medical applications, such as tailored prostheses development [42], human body scan [207] or accurate vascular reconstruction [154]. Another field where image depth information brings several enhancements is Cultural Heritage: 3D knowledge is often used to assist archaeologists for accurate excavation reconstructions [17, 55] or for digitalization and preservation purposes [71]. The second part of this thesis is dedicated to specific applications exploiting 3D data, with the aim of highlighting the benefits that such techniques bring to the final outcome. In detail, we will put particular emphasis on industrial settings and cultural heritage applications.

Over the years, many techniques have been proposed for 3D data acquisition; usually categorized on the basis of the exploited physical principle and on the design of the adopted sensor. Some approaches are not even based on imaging. For instance, coordinate measuring machines [81] are instruments which utilize a probe attached to a moving support to measure material point positions by means of direct contact. Although they offer a good precision, these machines are confined to highly specialized contexts due to

the necessity of physical contact with the object. This constraint, indeed, results in severe limitations in terms of speed and range of objects that can be acquired.

To overcome these shortcomings, the majority of 3D sensors implements contactless measurement, combining various vision-based technologies. We can distinguish between two major groups of approaches, namely *passive* and *active* reconstruction techniques.

In passive 3D reconstruction methods the scene is captured "as it is" by two or more cameras, with no additional hardware (Figure 1.1, left). In this case, the position of captured items is computed by means of geometric triangulation [111]. The basic principle of triangulation is quite straightforward: the depth of each observed point is recovered computing the intersection of projective rays coming from corresponding pixels in different views [79]. For this process to work, two problems must be addressed.

first, the imaging geometry of cameras must be accurately known. Parameters of the imaging model can be computed through a calibration process which is usually performed once for each configuration. Such calibration can be carried out using many different methods that have been extensively covered by recent research and have reached a high degree of accuracy and repeatability [179].

The second problem to be addressed is seeking for correspondences in different views. Unfortunately, the search for matches between two points of view is a much challenging task. The performance of this step greatly depends on a deluge of factors, including scene geometry, object surface features, illumination and materials. As a result of this variability, literature counts a large number of approaches, each one of them dealing with a specific scenario. If the scene is characterized by recognizable visual features, a possible approach consists in direct exploitation of images to assess correspondences from low-level data. In this case, some relevant areas of an image are extracted and identified as image feature descriptors. Two common algorithms are SIFT [131] and SURF [15]. These techniques extract for each image a number of feature vectors, each of which encloses information about location and properties like scale and rotation of the extracted area. The advantage of such descriptors is that they ensure scale and rotation invariance, thus they can be easily compared in order to assess correspondences among different points of view. Another image-based approach consists in dense stereo matching [186], which searches for areas with high correlation along epipolar lines.

All these systems provide an acceptable set of correspondences to be triangulated and get a 3D point cloud. However, some scenarios could involve a scene which does not show significant features so that image-based matching would not work. Moreover, other applications could require a more dense 3D reconstruction.

Active methods employ additional hardware, which is jointly used with at least one capturing camera. The principal aim of such techniques is to assess the image depth or points correspondences casting some supplementary information directly on the acquired scene. Time-of-flight cameras, for instance, consist in a pulsating light source combined with a sensor measuring the round-trip time of the signal (Figure 1.1, centre) [77, 110]. A popular approach is LIDAR (LIght Detection And Ranging) method [67]. Such values are exploited to compute a depth map, which in turn can be used for reconstruction if imaging geometry (i.e. camera calibration) is known [53]. Despite this technology have been

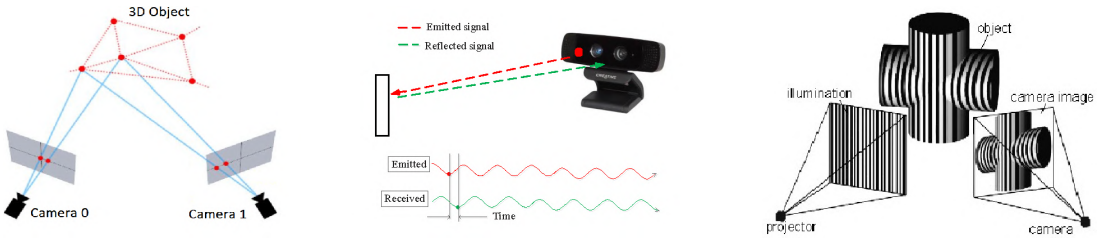


Figure 1.1: Different 3D reconstruction techniques. From left to right: passive stereo, time-of-flight, structured light.

proven reliable in some specific applications, it suffers from two major drawbacks. First, the sensor does not usually provide a high resolution response. Moreover, the light round-trip does not offer a high precision level so the error on depth computation is significant. For this reason such technology is often used for large-scale environment reconstructions. While performances of time-of-flight technology are fair in some applications, many scenarios demand higher resolution.

Structured Light

Generally speaking, structured light techniques consist in any approach which exploits an external light source of known behaviour in order to fix point correspondences and proceed to triangulation (Figure 1.1, right). Such methods are considered to be the most reliable choice for an accurate surface reconstruction, thanks to their flexibility and the ease of employment in a large variety of contexts. In fact, the additional light pattern projected on the captured scene allows for direct signal decoding to assess projector-camera correspondences or introduces visual features that can be matched through a dense stereo algorithm. The literature counts a huge number of structured-light 3D reconstruction techniques, as well as specific approaches meant to be applied in highly specialised applications. Chapter 2 includes an overview of the more representative structured-light techniques and coding patterns.

A basic example of a structured light technique is laser scanning [66], where a laser line is swept through the scene and correspondences are found by its intersection with the epipolar line in each captured image [79]. The same principle is applied by frame coding approaches, where the laser is replaced by a digital projector and a specific sequence of patterns is projected all over the scene. Such patterns are designed to easily decode the column of the projector frustum hitting a material point, and thus to simulate the laser sweep. Many pattern strategies have been proposed, ranging from simple binary sequences [185], to colour schema [38] or single shots exploiting specially crafted grids [98]. Moreover, in the literature several coding/decoding strategies have been proposed, each one with a specific goal in mind. For instance, some approaches are specifically designed to perform real-time reconstruction [58,229] using a reduced number of patterns in order to increase the speed or exploiting the interaction between projected signal and the natural texture of the scene [216]. In [63], the authors propose to infer the depth from

the signal itself using a learning technique, dropping the actual triangulation step.

Despite the abundance of highly specialized patterns, one of the most popular is the phase shift method [197]. This is particularly true in an industrial setting, since this approach allows high accuracy, resilience to noise and surface texture and great flexibility. The basic idea underlying phase-shift is indeed quite simple: the projected patterns are sine wave intensity frames which are periodic along one direction of the digital projector frustum. Several shifted patterns are projected and retrieved over time, then the initial phase (i.e. the phase at frame zero) is computed for each image pixel by convolving the acquired signal with the ideal projected sinusoid. Unfortunately, due to the periodic nature of the pattern in space, many pixels will be characterized by the same initial phase, thus a disambiguation step must be employed: such step takes the name of phase unwrapping.

The initial part of this thesis is focused on phase-shift technique and on some improvements over the current state-of-the-art. In particular, we propose a novel phase unwrapping approach based on a probabilistic framework, and a subsequent outlier detection and code recovery which employs the likelihood values from the unwrapping process or a voting schema in a spatial neighbourhood. Moreover, we also introduce some phase correction techniques related to signal acquisition in the presence of a scene exhibiting high-contrast regions.

1.1 Thesis Contents

This thesis is organized in the following way: Chapter 2 introduces a detailed background for the imaging process and structured-light scanning methods, highlighting the connections of the presented topics with the novel techniques comprised in the different chapters. The core of this thesis is then divided into three main parts: Part I contains some improvements over the state-of-the-art 3D reconstruction employing phase-shift technique, while Part II presents a collection of practical applications where the proposed unwrapping and correction techniques are employed in real-world settings. Finally, Part III explores the topic of surface detail transfer, introducing a preliminary study applying learning techniques to perform such task.

In details, Chapter 3 tackles the phase unwrapping task in a multi-phase structured light approach, presenting a novel unwrapping technique based on a stochastic formulation of the problem. We start by modelling the signal acquisition process, then each observed phase is considered as a value sampled from a Wrapped Normal distribution of unknown mean. The unwrapped phase is then computed through maximum likelihood estimation, and an additional recovery framework based on the same likelihood values is presented. The work has been published in [161], where the relation with the signal noise is studied in details, and in [160] where the unwrapping technique and the neighbourhood recovery algorithm are thoroughly analysed.

Chapters 4 and 5 propose two techniques finalized at improving the unwrapped phase quality, in the case of multi-period phase-shift 3D reconstruction. Such chapters offer a

good theoretical basis for two practical correction techniques with the aim of improving the final 3D reconstruction accuracy. In particular, Chapter 4 describes a novel phase correction method operating on image zones where non-uniform albedo causes the recovered values to be biased, introducing artefacts in the final reconstructed surface. Such correction technique has been proposed in [164]. On the other hand, Chapter 5 proposes a neighbourhood-based recovery technique allowing for restoration of the final coded image in the case of faulty unwrappings. Several recovering strategies are simultaneously analysed in terms of final coding precision. Such work has been presented in [163].

The second part of this thesis is focused on some real-world applications employing 3D data and structured-light techniques. Chapter 6 describes a calibration technique for a camera-projector pair setup employed in the subsequent works. The proposed calibration procedure directly exploits the coded images computed via the phase unwrapping algorithm presented in Chapter 3 and allows for an accurate extrinsic parameter estimation without using any kind of marker. Chapters 7 and 8 present two industrial-related applications working with 3D data. The first proposes a cylindrical primitive extraction from non-oriented point clouds, presented in [162]. It exploits the geometrical properties of the scene to extract a set of candidates and a game-theoretical inlier selection to devise the best cylinder estimation. In Chapter 8 a technique for microscopic surface reconstruction and inspection is presented, along with a specific application for gear teeth inspection. The method includes a novel calibration procedure for a structured-light system employing a telecentric camera and a projector with a fixed line pattern. Chapter 9 presents a structured-light application for 3D reconstruction in cultural heritage. The complete scanning of an ancient Roman sundial is first described, then the 3D model is exploited to recover the original working latitude through an optimization process which takes into account the shadows casted on the object's surface. Finally, in Chapter 10 we introduce a preliminary study on the topic of surface detail transfer exploiting learning techniques.

2

Related Work

The recent evolution of increasingly affordable and powerful 3D sensors and the consolidation of fast reconstruction algorithms enabled the widespread adoption of 3D data in both consumer [76] and industrial [134] off-the-shelf devices.

While consumer applications place more emphasis on speed and performance, in an industrial setup accuracy is of greater importance. To this end, a lot of effort has been put in reconstruction techniques trading design simplicity and speed for higher precision in 3D recovery and robustness to surface characteristics of captured objects.

Active 3D sensing technology exploits the projection of some light patterns of known spatio-temporal structure onto the object. To decode those patterns, rays emitted by the projector are matched with the ones entering the camera to recover the depth of each scene point. Digital projector-based structured light is an example of such approaches. The main idea behind this technique is to decode from acquired image the corresponding projector coordinates (in pixels) to establish a map between the observed scene and the projector itself (camera-projector setup) or between different acquisitions of the scene (multi camera setup).

This Chapter first introduces the fundamental concepts of the imaging process and projective geometry, then revises the main contributions in the literature in the field of structured-light 3D reconstruction approaches, with a particular attention to phase unwrapping techniques, that is the core topic of Part I of the thesis. Moreover, the chapter offers an overview on structured-light system calibration, for which we propose a technique in Chapter 6, and introduces the main motivations behind industrial and cultural heritage applications explored in the thesis. The last section is dedicated to model fitting, that is a topic often faced in Computer Vision practical applications.

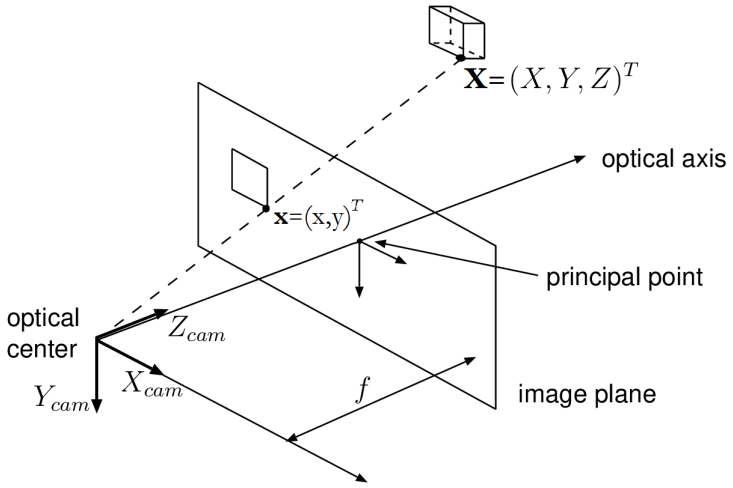


Figure 2.1: Pinhole camera model. The optical centre corresponds to the origin of the camera reference frame, while the image plane is orthogonal to the z-axis and located at a fixed distance f (focal length). The projection (u, v) of a point is obtained by the intersection of a ray coming from the origin to the point with the image plane.

2.1 Imaging Process and Epipolar Geometry

The geometry of imaging process in the basis of each vision-based reconstruction technique. Usually the capturing camera is modelled as an entity which observes physical points in the real world and maps them in an *image plane*, forming the image that we commonly know.

There are several camera models proposed in the literature [79], but the most used is the so called *pinhole* camera model. This camera model is formed by a finite centre of projection (or *optical centre*) and a plane. The former element is a point in three-dimensional space and corresponds to the origin of an Euclidean coordinate system, also known as camera reference frame. The plane located at $z = f$ in such coordinate system is identified as the image plane. The scalar value f is called *focal length* of the camera (in pixels), and it is one of the parameters involved in the imaging process. In the pinhole model a point $\mathbf{X}$ in three-dimensional space is mapped into a point $\mathbf{x}$ on the image plane by intersecting the ray coming from the camera centre and passing through $\mathbf{X}$ with the image plane.

The ray exiting the camera centre and perpendicular to the image plane is known as *principal axis*, and it meets the image plane in a specific point called *principal point*. Usually the acquired image is represented with a different coordinate system (in pixels), and its origin does not correspond to the origin projected on the image plane. For this reason an offset (c_x, c_y) has to be defined, denoting the real coordinates of the principal point. Figure 2.1 summarises the anatomy of the pinhole camera model, characterizing the elements previously described.

If world and image points are represented with homogeneous coordinates, we can

denote a world point as $\mathbf{X} = (X, Y, Z, 1)^T$ and its projection on the image plane $\mathbf{x} = (x, y, 1)^T$. Then, the mapping from three-dimensional space to the image plane can be expressed as follows

$$\begin{pmatrix} fX + Zc_x \\ fY + Zc_y \\ Z \end{pmatrix} = \begin{pmatrix} f & 0 & c_x & 0 \\ 0 & f & c_y & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} \tag{2.1}$$

where

$$K = \begin{pmatrix} f & 0 & c_x \\ 0 & f & c_y \\ 0 & 0 & 1 \end{pmatrix} \tag{2.2}$$

is called *intrinsics camera matrix*, and contains the camera inner parameters that characterise the way in which the world is captured on the image plane.

This formulation implicitly assumes that the world points are expressed in the camera coordinate frame, with the origin corresponding to the centre of projection. In practice it is common to describe the points observed by the camera in another reference frame, denoted as *world reference frame*. The relation transforming a point in world reference frame to the camera reference frame can be expressed with a rotation matrix R and a translation vector $\mathbf{t}$. Finally, the camera projection matrix P can be composed in the following way

$$P = K[R|\mathbf{t}] \tag{2.3}$$

where K is the intrinsics matrix and $[R|\mathbf{t}]$ is a 3×4 roto-translation matrix moving the points from world to camera reference frame. The roto-translation parameters are also called *external* or *extrinsic* parameters of the camera, since they express the relation of the internal camera coordinate system with the external world.

Lens Distortion

In practice the acquisition process does not work as a perfect pinhole camera, this because usual camera lenses deviate the entering rays, causing a distortion effect which is particularly visible in peripheral areas of the acquired image [79]. For this reason the resulting images need to be corrected before applying any reconstruction algorithm, so that the acquisition process is again a perfect pinhole system. Several distortion models have been proposed [51, 220], some of them specifically designed for highly distorted cameras [189].

Among them, the most diffused model includes radial and tangential factors. Radial distortion causes straight lines to be bended the more they are far from the principal point, while tangential distortion is caused by inaccuracies in the lenses alignment with respect to the image plane.

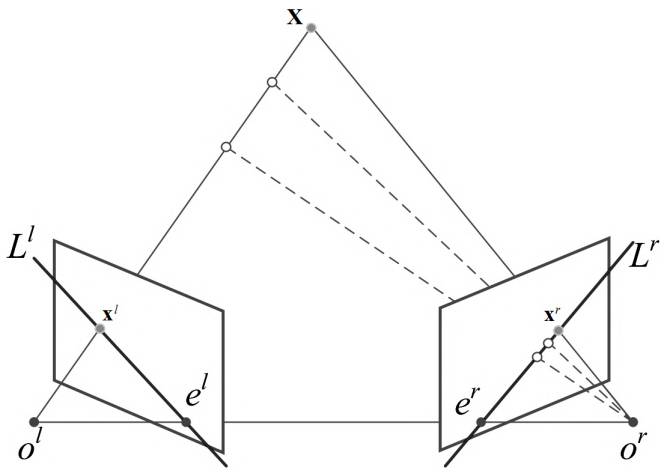


Figure 2.2: Epipolar constraint. The 3D point $\mathbf{X}$ generates the epipolar plane that intersects the two image planes in correspondence of the two epipolar lines L^l and L^r. The intersections e^l and e^r of the line connecting the camera centres and the image planes are called *epipoles*.

Such model is described through five parameters

$$\mathbf{k} = (k_1, k_2, k_3, p_1, p_2) \tag{2.4}$$

where k_1, k_2, k_3 express the radial coefficients, and p_1, p_2 are tangential coefficients. Radial distortion is corrected using the following formula

$$x' = x(1 + k_1 r^2 + k_2 r^4 + k_3 r^6); \qquad y' = y(1 + k_1 r^2 + k_2 r^4 + k_3 r^6) \tag{2.5}$$

where r is the distance of (x, y) from the principal point and (x', y') is the corrected pixel coordinate. For the tangential distortion, the mathematical model is the following

$$x' = x + (2p_1 xy + p_2(r^2 + 2x^2)); \qquad y' = y + (p_1(r^2 + 2y^2) + 2p_2 xy) \tag{2.6}$$

Since the distortion function is applied in retinal coordinates, $\mathbf{k}$ have to be calibrated along with the camera intrinsic parameters to allow an accurate reconstruction.

Epipolar Geometry

In classical stereo vision method the scene is observed from at least two points of view. When we have two cameras observing the same scene, if their reciprocal position is known, a set of constraints deriving from their intrinsics geometrical properties can be defined and used to perform reconstruction. Assume that both cameras observe the same material point $\mathbf{X}$, then it is mapped in $\mathbf{x}^l$ and $\mathbf{x}^r$, respectively in the left and right cameras image planes. If the imaging parameters are known and the point correspondences

are given, the 3D coordinates of point $\mathbf{X}$ can be easily computed via triangulation by intersecting the two rays originated from the two camera centres and passing through the points' projections.

In the case the correspondences between two views are not known a priori, epipolar geometry constraints are used to significantly reduce the search space on the image plane. Moreover, many structured-light systems exploit such properties to assess the correspondences between camera and projector image planes.

The two camera centres and the observed point $\mathbf{X}$ define a plane called *epipolar plane* which intersects the two image planes along two lines L^l and L^r, denoted as *epipolar lines* [79]. This relation is shown in Figure 2.2, along with the epipole points e^l and e^r, that are the intersections of the line $o^l o^r$ with the image planes. As a consequence of such geometrical relation between the two image planes, the projected images of $\mathbf{X}$ lie on the correspondent epipolar lines. This constraint is widely employed in stereo reconstruction approaches, since the search for the correspondence of $\mathbf{x}^l$ in the right image plane is limited to the line L^r and practically becomes a 1D search.

Some coding techniques allow only for the identification of vertical or horizontal projector coordinate and then exploit epipolar constraints to triangulate the points. Assuming an horizontal camera-projector setup, so that the projected pattern only disambiguates among its columns. Each point on the camera image plane generates a line on the projector image plane, so that when the camera observes the projector code h at point (u, v), the corresponding projector coordinate can be obtained intersecting the computed epipolar line and the vertical line $x = h$ on the projector image plane.

2.2 Structured-Light Scanning

All the triangulation-based 3D reconstruction techniques work by seeking for a correspondence between points observed by two or more devices [159, 174].

In structured light approaches such correspondences are assessed by projecting some known information directly on the scene, so that in practice one viewpoint is substituted by a projector.

When one or more cameras capture the scene, the projected pattern results distorted by the geometry of the target objects. Specific light patterns are designed to encode a unique label in each observed point, so that a mapping between camera points and pixel coordinates in the projector's image plane can be uniquely defined. Once the correspondences are established 3D structure is computed from the geometric relationship of the corresponding points.

2.2.1 Coding Techniques

The literature counts a huge number of different patterns involved in structured-light reconstruction, together with several coding approaches. Each pattern is specifically

designed to satisfy some requirements, such as acquisition speed [229], coding precision [214] or outlier resilience. Some approaches have a precise target application like the measurement of microscopic physical phenomena [203] or the acquisition of specific materials like glass, black objects [30] or metallic surfaces [103].

Among the variety of light patterns, one major distinction is in the nature of the retrieved codes, which can be *discrete* or *continuous* [180]. Discrete patterns encode the same codeword in several pixels, so that the final resolution depends on the density of projected codes. A classical example of discrete approach is binary coding. On the other hand, continuous encoding exhibit a smooth profile, and each decoded pixel present a continuous value in a range. These methods offer a higher reconstruction density and accuracy: phase shifting technique belongs to this category.

Another common classification for structured-light approaches exploits the way in which codes are embedded and decoded in the patterns [181]. In details, we can distinguish among *neighbourhood*, *direct* and *time-multiplexing* coding.

Neighbourhood-based

In neighbourhood based techniques all labels are encoded in a single pattern, and each codeword is computed as a function of the values observed around it. Numerous approaches have been developed for this kind of task: some techniques propose spatial coding schemas that are repeated and recognized through template matching [62]. Statistically random codes are projected over the scene such that any camera point can be associated to a projector point via correlation. Thanks to the simplicity in the implementation and the speed of acquisition, such approach is actually employed on several consumer products like Microsoft Kinect V1 (see Figure 2.3, right) and Intel RealSense. Other single-pattern approaches, like the one proposed in [98] exploit geometrical properties of a single grid pattern to compute 3D data. In this approach, horizontal and vertical lines are detected and uniquely identified by looking at their intersections and the distance among neighbour intersections.

Other methods employ specific structures like De Bruijn sequences [88], which are pseudo-random values exhibiting specific inner properties. The algorithm used to compute such patterns is based on de Bruijn graphs [68] and generates univocal sequences in the one-dimensional pattern, so that a stripe's codeword can be identified by analysing its neighbourhood (see Figure 2.3, left). M-arrays [156] are random arrays in which an $n \times m$ sub-matrix appears once in the whole pattern. They are a two-dimensional representation of De Bruijn sequences and can be represented with a binary encoding, coloured dots or geometric features like circles or stripes (see Figure 2.3, centre for an example). The main disadvantage of this approach is that the decoding robustness is influenced by the chosen window size.

Neighbour-based approaches have several advantages in terms of speed and simplicity, but they involve some drawbacks that should be taken into account while developing a structured-light reconstruction system. First, most of the times they work with coloured patterns, which requires a fundamental colour calibration step before the proper acquisi-

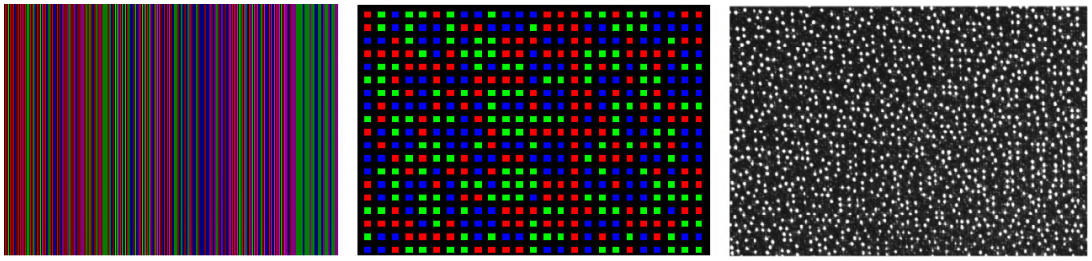


Figure 2.3: Neighbourhood-based coding techniques, from left to right: De Bruijn stripe sequence, coloured M-array and statistical pattern (used in Micorsoft Kinect V1)

tion. During such calibration extra patterns are usually employed to acquire the necessary information to correct the acquired scene. Moreover such approaches offer discrete coding and, if compared with other methods, they offer a low resolution (lower then camera) and high sensitivity to noise. The code recovery process could also fail in some points if their neighbourhood is partially occluded.

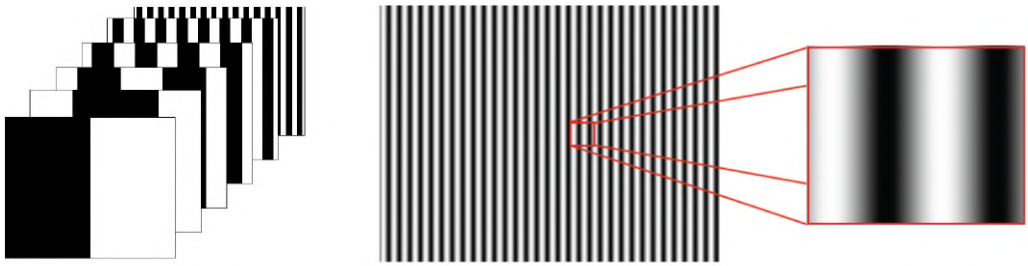
Direct Coding

Another class of encoding strategies consist in direct coding: in this technique each pixel independently encodes its own label, with no need of contextual information. Such result can be obtained either using a wide range of different colours or introducing periodicity.

These approaches can be based on several grey levels or coloured patterns. In the first case, an early work proposed an approach named *intensity ratio depth sensor* [37]. Such technique projects along the vertical direction a linear wedge including different grey levels, then the ratio between every pixel of the same wedge and the pixel value with constant illumination is computed. This ratio can be directly related to the index of the column pattern. In practice this approach needs the projection of two pattern and offered a low accuracy. Subsequent works proposed some improvements over the original work [41, 143] introducing more patterns to be projected and making this technique similar to time-multiplexing approaches.

Other direct coding methods are based on coloured patterns. The work described in [202] propose the rainbow pattern, containing vertical stripes encoded with different wavelengths. The scene was acquired using two colour filters and then the ratio between the two images was taken. The works [70, 184] improved the initial proposal adding periodicity in projected pattern.

Theoretically direct coding techniques offer a high resolution of 3D information, but in practice these approaches are extremely sensitive to noise since the distance between close codewords is narrow. Moreover the acquired colours are highly influenced by the scene features, like illumination or albedo, and thus a proper tuning step or multiple acquisitions are often required. For these reasons, they are not suited for applications involving moving scenes or accurate measurements.



(a) (b)

Figure 2.4: Time-multiplexing patterns. (a) The projected sequence of binary patterns for Gray coding approach. (b) Phase shift sinusoidal pattern: the pattern is shifted N times to complete a period.

Time-multiplexing

In time-multiplexing approaches, coded data is given through a sequence of patterns projected over time and acquired in a synchronized way. After the acquisition, the sequence of intensity values observed at each point during the projection are used to compute the final code. Usually time-multiplexing techniques offer a high measurement accuracy, thus many approaches have been proposed during the years. The coding strategies can be categorized into the following groups: binary codes, n-ary codes and phase shifting. There are also several hybrid approaches that employ a combination of time-multiplexing and neighbourhood-based encoding.

Binary coding approaches use two levels of illuminations, that can be labelled as ”zeros” and ”ones” during decoding. A classical example of this technique is Graycoding [90], where a series of binary striped patterns is projected. After acquisition each pixel is labelled with a binary code computed from the sequence of observed intensity values. The advantages of Gray codes is that adjacent codes are designed to have a Hamming distance equal to one, preventing coding errors. This approach has two major drawbacks: (i) the difficulty of identifying code discontinuities with sub-pixel accuracy and (ii) the errors derived from a scene with non uniform albedo. These issues have been addressed in several ways, but in practice they make the procedure more complex introducing more patterns [194] or an error model for edge localization [213].

Another drawback of binary codes is that they require a high number of patterns to obtain a good density reconstruction, since they only employ two levels of intensity. An approach leading to the reduction of projected patterns is the use of n different intensity levels, so that we compute n-ary codes. The work presented in [38] proposes a multilevel Gray coding approach based on coloured patterns, while in [83] multiple grey levels are used.

A well-known temporal coding technique is phase shift, in which light patterns are composed by fringes of a sinusoidal signal. Such pattern is shifted over time and projected consecutively on the scene during acquisition. Then, the phase component of the signal is computed pixel-wise from the intensity values [238]. An in-depth theoretical explanation

of phase-shift technique is given in Section 3.2. This method is widely adopted because of its intrinsic robustness to noise and its ability to recover continuous codes with sub-pixel accuracy. Given the periodic nature of the signal the phase values are repeated across the scene, thus a disambiguation algorithm is required. Such process is called *phase unwrapping* or *absolute phase retrieval*, and consists in identifying the correct fringe order for each observed phase value. Section 3.3 illustrates a novel probabilistic phase unwrapping algorithm exploiting multiple sinusoidal signals of different period lengths observed over time.

A large number of phase unwrapping algorithms have been proposed in the literature, among them we can distinguish two main classes of approaches: spatial and temporal phase unwrapping [228]. There is also a third category of hybrid approaches that involves heterogeneous solutions like the usage of pre-known scene information, additional hardware or combinations with other methods.

In spatial phase unwrapping the relative fringe shifts are computed by looking at neighbour values in a single phase map [167]. The most popular approach is the so called reliability-driven: first a quality map is defined over the acquired image, then the points are iteratively unwrapped following a path traced by decreasing values of the quality map. In this way the points associated with a higher quality are unwrapped before the others so that errors coming from noisy observations are not propagated. In [200] and [233] a number of quality-guided algorithms and strategies are presented. Even if this approach requires only a single phase map acquisition, it exhibits several drawbacks. First of all, the scene must not generate a phase discontinuity greater than π: this narrows the kinds of surface that can be acquired, and also depends on the system's geometry. Moreover, this technique only allows the unwrapping of isolated, connected patches of the scene, thus a whole reconstruction may not be possible. These limitations, in practice, make the method unflexible and not suitable in many scenarios.

In temporal phase unwrapping some additional information is added after the sinusoidal pattern to provide a discrimination among fringes. Several solutions have been proposed, for instance the addition of Gray codes to assign a different label to each fringe [182]. Such combination of course suffers from the same issues of the pure Gray coding approach, and involves a significant number of additional projections. Other works consist in different kinds of additional signal, for instance a binary random pattern to perform stereo matching [13], or a coloured striped pattern [46]. Some approaches exploit distinct phase maps of different period lengths, that are projected after the first one. Multiple periods leads to better accuracy because additional signals are involved in code computation instead of being used only for disambiguation purposes.

Also in this case, several algorithms have been proposed to perform phase unwrapping using different phase maps [239]. In the literature, one of the first proposals consists in multi-wavelength phase unwrapping [48, 49]. This technique is also called *phase difference* and exploits two close frequencies to generate another phase map from their difference with no ambiguities. Another technique is the multi-frequency or hierarchical phase unwrapping [89, 150]. In this approach the last pattern contains only one fringe (thus it exhibits no discontinuities) and it is used to unwrap the subsequent phase maps.

The technique described in [236] and [125] introduces a number-theoretical unwrapping algorithm which exploits look up tables to allow for fast computation. This algorithm offers good results in optimal conditions, but it is very sensitive to small phase inaccuracies and outliers, leading to a table miss.

Other phase unwrapping methods involve the usage of additional hardware, like cameras or projectors [91,123] to exploit geometrical constraints of the acquisition system. In particular, the minimum phase method proposed in [12] exploits geometrical constraints to generate a virtual phase map at a known depth that is used to compute the unwrapped phase values. Some other techniques involve the prior knowledge of the scene such as CAD models [116].

The phase unwrapping technique proposed in Chapter 3 belongs to temporal phase unwrapping methods, since it exploits more than one phase maps projected over time to retrieve the absolute phase value in each image pixel. The techniques described in Chapters 4 and 5 consist in a phase correction and code recovery methods that can be applied to any acquired phase map, independently on the unwrapping approach.

2.2.2 Calibration

Any 3D reconstruction method based on stereo vision requires a fundamental calibration step involving the estimation of the imaging parameters (i.e. intrinsics and distortion) along with the geometry of the system. Such task directly affect the reconstruction quality since the more accurate the procedure is, the more reliable are the following measurements.

In the case of stereo vision applications, we distinguish between two calibration processes, namely *intrinsic* and *extrinsic* calibration. Intrinsic calibration involves the computation of the imaging parameters (described in Sec. 2.1) defining how to map observed physical points on the scene to points on the 2D image plane. In particular, for a pinhole camera, the required parameters are the focal length, the principal point offset and the distortion coefficients. In the case of active stereo approaches such procedure has to be carried out for both camera and projector devices, since the projector imaging process can be practically modelled as an ”inverse camera”, casting a known image onto the scene.

As for intrinsic calibration, the literature includes numerous techniques to retrieve camera parameters. Popular approaches employ some specifically designed target objects to be captured by the camera in several positions. A calibration target can consist in a simple planar object or in a more sophisticated object of known geometry, and usually their surface exhibit salient features to be extracted and easily identified (like squares, circles or ellipse patterns). In this way the calibration algorithm is able to robustly associate each detected image point with a physical point lying on the target's surface. After the target object has been acquired and its salient features have been extracted, the estimation of the camera parameters is usually carried out through an optimization process [80, 212, 232]. Other calibration approaches does not employ specific targets but directly exploit the observed scene features and are addressed as auto-calibration techniques [166, 212].

In camera-projector setups, the extrinsic calibration step consists in assessing the relationship between the camera reference frame and the projector image plane. This kind of process exhibits some differences with respect of classical extrinsic calibration techniques since the projector can not acquire images like a camera, but rather it behaves like a camera which observes a known image, that is the projected pattern. Numerous camera-projector calibration methods have been proposed in the literature [84, 85, 226]. One of the most used is [230], which proposes a projector calibration technique enabling the projector to capture a planar checkerboard. This is performed establishing correspondences between camera and projector points, so that the calibration can be carried out as if the projector was a normal camera. Finally, other approaches discard the assumption that both projector and camera are focused and developed calibration solution for out-of-focus calibration. In [117] the calibration process allows the projector to capture only the centre point of a pixel so that the blurring effect is avoided, while in [16] the out-of-focus camera is calibrated using an active calibration target.

Chapter 6 proposes a camera-projector calibration technique which computes extrinsic parameters as well as intrinsic projector parameters at the same time. The method does not require a specifically designed calibration target, since it only requires any planar surface to be acquired from various orientations. The problem of seeking for correspondences between image and world points is resolved through the proposed phase unwrapping algorithm, that offers a dense set of point correspondences over the image plane of both camera and projector. Then, the system parameters are simultaneously estimated through an optimization process.

2.2.3 Applications

In the second part of this thesis we explore two completely different scenarios, namely: industrial quality inspection and cultural heritage. They are both good examples on how structured-light reconstruction techniques are effective in practical applications and can be exploited to obtain significant improvements over classical imaging techniques. In particular, the novel phase unwrapping and correction methods presented in the first part of the thesis have been applied in the described applications.

Structured-light 3D scanning and phase shift are widely employed in a variety of industrial applications, thanks to the method's flexibility and its potential high degree of accuracy. In such settings, the adoption of a precise measurement system is a key element to assess the quality of the manufactured objects during construction, specially because an accurate and extensive control also reduces potential issues that may emerge during the subsequent production steps. Commonly employed tools are CMM (Coordinate Measuring Machines), but they require physical contact with the object and thus are not always the best choice. On the contrary, structured-light approaches are portable solutions which can be easily integrated in industrial settings, allowing for high precision measurements without touching the artefact [222, 223]. Moreover, 3D data are often exploited for automation, in order to perform automatic scene understanding [78], and in robotics

applications to support the manipulation of objects [112, 115]. Finally, industrial applications have been developed in order to acquire materials with specific surface reflectance properties (i.e. metallic or shiny), which result particularly challenging [43, 102, 103].

In this thesis two specific industrial applications will be discussed. First, Chapter 7 proposes an algorithm for cylindrical primitive fitting from acquired point clouds (for which the related work is described in the next Section). Second, a structured light system for micrometric surface measurement is presented in Chapter 8.

The latter topic is interesting for its relevance in several applications. These range from quality control [217], fabric defect inspection [106], weld pool surface depth measurement [178] or laminated plastic wrinkles evaluation [225] to small coins or electrical component verification [126, 141]. These approaches usually employ telecentric lenses, thanks to their extended depth of field [120]. For example, the method described in [119] proposes to combine fringe projection with binary defocusing technique [114] for 3D microscopic profilometry. Two different setups are analysed: the first including both camera and projector telecentric lenses and the second one employing only a telecentric camera. These specialised applications often need specific calibration procedures for systems with telecentric optics. Calibration targets are often used, such as one or more planar surfaces with squared or circular features [118], that can be easily detected and identified by the imaging system. Once the initial parameters are estimated, they are refined through a bundle-adjustment procedure [127].

The approach proposed in Chapter 8 is based on a camera-projector setup, both equipped with telecentric lenses: a fixed pattern including several parallel lines is projected onto the scene to recover the depth information corresponding to the intersections with the analysed object. The novelty resides in a new calibration technique using a sphere acquired by the system to devise the intrinsic and extrinsic parameters together with a specific application for gears surface inspection.

A completely different field explored in this thesis, which benefits from the usage of 3D reconstruction methods, is cultural heritage. There are a number of applications in which 3D data can substitute other kinds of recording techniques [153]. For instance, classical close-range photogrammetry approaches are designed to capture the object geometry and can be integrated or substituted with a 3D scanning process for some kinds of artefacts [28]. One of the main purposes of 3D scanning in cultural heritage is the digitalisation with the subsequent documentation and archiving of historical objects, in addition to the analysis of some characteristics that would have been infeasible with other instruments [57]. Moreover, the available 3D scanning techniques offer a high flexibility level so that the digitalisation process can involve small or medium items, like a statue, or larger scenes as entire temples [192] or cities [104]. Other applications in this field include the monitoring of archaeological excavation fields [56, 59], serving both as documentation and monitoring support; and the reassembly of fractured objects [86].

The application presented in Chapter 9 explores two different aspects of 3D reconstruction for cultural heritage: the digitalisation of an ancient artefact for preservation purposes, and the exploitation of the acquired model to devise some relevant features

about its usage. The first part of the Chapter is indeed dedicated to the description of a specific pipeline that we adopt to recover the complete 3D model of an ancient Roman sundial using structured-light technique. In the second part we describe a mathematical approach to devise the optimal working latitude and the original gnomon's shape based on the acquired 3D model.

2.3 Model Fitting

A fundamental task in many Computer Vision application employing 3D data is the extraction and identification of one or more well-known objects included in the scene: two examples are human body detection [73] or object segmentation [205]. In some situations 3D data are far from being perfect: this can be due to limitations of the working environment (light conditions, occlusions, etc.) or to the nature of the scanned objects themselves. For instance, the scanning process of objects presenting shiny metallic areas could easily lead to an incomplete or noisy 3D surface [11]. Furthermore, many times the acquired scene is complex and contains several elements which are to be filtered out as they are not relevant for the subsequent analysis. The extraction of elementary shapes such as geometrical primitives (planes, spheres, cylinders, pyramids) is an essential task in several scenarios [93, 121, 208], such as industrial automation and inspection [74, 172], reverse engineering [135] or robot manipulation [196]. These applications often require a robust and reliable parameter estimation in order to perform additional operations.

Chapter 7 focuses on the specific task of cylinder extraction from generic 3D point clouds, proposing a novel extraction algorithm. Indeed, cylindrical shapes are often found in many applications including natural landscape analysis [157, 165], robot manipulation [142], automated pipe-run reconstruction [169] or industrial quality inspection [138]. In this section we briefly cover a number of state-of-the-art cylinder fitting and extraction techniques pertinent with the proposed method. Among these solutions we can distinguish two different approaches: the ones requiring oriented point clouds (i.e. each point is associated with a normal vector) and the ones which only uses 3D points locations. While surface normals significantly reduce the parameter search space and the problem complexity, their availability in acquired data is not always ensured and the reliability of computed normals strongly depends on the quality of the point cloud itself. We can distinguish between two conceptually different tasks, that are *fitting* and *extraction*.

The aim of fitting methods is to compute the best set of parameters for a cylindrical model (axis, location and radius) to suit the given point cloud. Usually this kind of task is performed after the acquired data has been segmented, so that we can associate a subset of 3D points with a geometrical primitive to be fitted. The parameter estimation is usually carried out through a linear or non-linear least-squares minimization of the points distance from the fitted surface [136]. Since these techniques assume a Gaussian distribution of noise, they exhibit high instability with respect to outliers. Consequently, the approach works if the segmentation is almost exact (so there is no clutter) or when the scene in-

cludes only a single cylindrical shape acquired with a good degree of accuracy. In [151] the authors propose a Maximum Likelihood Estimation assuming a Gaussian mixture distribution for errors, and they apply the method to a terrestrial laser scanning application. The algorithm proposed in [148] extracts a single cylinder from noisy and possibly incomplete data samples. It adopts robust Principal Component Analysis and regression to reliably compute the parameters of a single cylinder. The approach described in [33] proposes algebraic methods to efficiently compute cylinder parameters from a minimal set of points, in both oriented and non-oriented cases. In [171] a fitting method for laser scanner is proposed: an ellipse is fitted for each line acquisition, then cylindrical parameters are devised from the set of 3D ellipses. The method does not require point normals but it is designed to work on a scene containing a single cylinder in a specific industrial scenario, therefore this approach can not be applied to other contexts.

Extraction algorithms are applied when capturing more complex scenes, including a number of primitives to be detected and other elements that should be automatically discarded. In the literature, the two major classes for extraction algorithms consist in RANSAC and Hough-based.

Thanks to their robustness, RANSAC-based methods [65, 209] are widely employed for primitive extraction and fitting [29]. They directly exploit acquired data to exclude outliers and devise a good set of parameters. The work described in [39] splits the extraction task in two phases, both exploiting random sampling techniques. First, the set of points is filtered extracting a possible cylinder direction from the Gaussian image of the whole point cloud, then its size and location are extracted in the same way. Note that the first step of this method highly relies on point normals to extract the cylinder orientation. In [187] the authors propose a shape detection algorithm operating in unorganized point clouds where the point normals have been precomputed. They formulate a method which works on large point clouds, exploiting points spatial proximity to devise their belonging to a shape, and then apply a hierarchical sampling strategy. Despite the advantages offered by RANSAC-based approaches, their non-deterministic nature makes them impractical when dealing with large-scale point clouds. Moreover, data exhibiting severe noise levels could be problematic for shape detection also because the performances are highly influenced by point normal accuracy.

Another class of approaches for primitive extraction is Hough-based methods. As in the well-known Hough transform [61], the generalised method uses specifically designed voting spaces to extract the primitives in the scene. In the case of cylinders the five parameters needed translate in a 5-dimensional voting space, resulting infeasible. To overcome this problem, the approach in [170] proposes a sequential Hough transform divided into two steps in order to reduce the dimensionality. This method first exploits normals in a 2-dimensional accumulator to estimate the axis rotation, then it estimates radius and location in a 3D accumulator.

Some works present variations of Hough-based technique for some specific applications. For example, the authors in [165] propose a fallen tree detection for terrestrial laser scanning. In particular, they make some assumptions on the position of the cylinders and apply some filtering to have an initial prior over data, then they propose a cylinder extrac-

tion based on [170] and a final refinement to obtain single tree stems. Another popular application is pipe-run reconstruction: in [152] the authors propose a prior segmentation and an area-based adaptive Hough transform to reduce time and space complexity. Hough-based methods are usually time-consuming for large amount of data and the voting space discretization could lead to poor performances. Moreover, in absence of point normals the parameter space dimensionality makes the approach infeasible so surface normals are always required and their precision highly influences the final outcome.

In addition to these two methods, several works combine RANSAC, Hough and other approaches within a pipeline, or exploit some prior knowledge of the scene to get better results. In [64] the author propose a cylinder detection algorithm for robot grasping applications. They assume the objects lying on a planar surface and perform an initial image classification using a CNN, then they propose an extension of the two-step Hough transform employing a randomized sampling scheme and finally use surface curvature to filter the results. Another cylinder fitting for robot grasping is proposed in [72], where the authors propose a significant data preprocessing phase, followed by a RANSAC model fitting employing a custom validation and a Hough-based voting scheme. The method proposed in [211] is composed by several steps: first, they select points belonging to potential cylindrical surface by analysing the curvature values, then for each of them a fitting procedure is run starting from its neighbourhood. The fitting algorithm is applied several times so that at each iteration additional inlier points are appended to the current cylindrical model; moreover a validation procedure determines the reliability of the extracted cylinder. Finally the set of cylinders is obtained through mean shift clustering.

Many extraction algorithms are designed to solve specific tasks, so they are based on strong structural assumptions, for example a prior over the cylinder orientation and location. In [128] an approach for large-scale point clouds is presented. In such application normals can not be exploited, but the authors assume to observe only horizontal or vertical pipes, so the cylinder orientations are significantly constrained. The approach proposed in [138] exploits an initial registration of the CAD model to the point cloud in order to have some prior information on the acquired data. Another common assumption consists in observing only objects lying on a planar surface, like in robot grasping [64, 72].

The majority of described methods require normal or curvature pre-computation, which we already stated to be sensitive to noise and outliers. Moreover, they need a specific tuning step for several parameters involved in the process. An accurate parameter calibration is a time-consuming task, and could be an advantage if working with stable conditions, but could be a serious limitation when working with heterogeneous scenes.

The extraction algorithm presented in Chapter 7 is partially inspired by [171], and improves the concept to extract multiple cylinders from an unknown configuration with several sources of noise. The method does not require point normals: it basically slices the scene with randomly generated virtual planes to compute a set of ellipses, each of which generates two cylinder candidates. Then, a specifically designed cylinder similarity function is employed in a clustering process based on game theory, which ensures the selection of the subset with the best mutual consensus.

I

Structured-Light Scanning

3

Robust Phase Unwrapping by Probabilistic Consensus

In phase shift profilometry sinusoidal patterns are projected and each point is labelled according to the observed phase. Then, due to the periodic nature of the signal, a disambiguation method (known as phase unwrapping) is needed. Several unwrapping techniques have been proposed in the literature, since noisy signals lead to inaccuracies in phase estimation.

This Chapter presents a novel phase unwrapping approach based on a probabilistic framework. The method involves the projection of multiple sinusoidal patterns with distinct period lengths, encoding different phase values at each point location. Phase values are then modelled as samples from a Wrapped Gaussian distribution with an unknown mean, determined by the projector code that generated the values. This formulation allows us to robustly perform phase unwrapping via Maximum Likelihood Estimation, recovering code values from the observed phases. Furthermore, the same likelihood function can be exploited to identify and correct faulty unwrappings by gauging mutual support in a spatial neighbourhood. An extensive experimental assessment validates the Gaussian distribution hypothesis and verifies the improvements in coding accuracy when compared to other classical unwrapping techniques.

3.1 Introduction

As already discussed in Chapter 2, the literature counts a vast number of solutions when dealing with phase unwrapping applications. Unfortunately, in case of noisy acquisitions many approaches discard the measurements that do not attain some prescribed precision levels. For instance, in number-theoretical methods the discrete look up table approach results in errors or missing codes, while in Graycoding approach all phases are used but only one (possibly noisy) phase map is exploited to devise the final unwrapped values.

The technique introduced in this chapter is designed to work in the opposite direction: we propose a robust multiple-period method that does not discard any acquired phase value. The very basic idea in fact is to keep all phase observations and use them to compute the code which most likely generated such values. We propose a novel phase unwrapping algorithm based on a probabilistic approach: phase values $(\varphi_1, ..., \varphi_n)$ observed at each point are considered as independent samples from a continuous distribution of unknown mean with variance $\bar{\sigma}^2$. They are used to generate a likelihood function defined over the space of unwrapped codes, then the final code is computed through Maximum Likelihood Estimation. In this way we are able to retrieve the most likely code value for each pixel, resulting in a simple and flexible approach. Cramér-Rao bounds are also exploited to relate the phase variance $\bar{\sigma}^2$ to the variance of the observed signal, that can be easily estimated online during fringe acquisition. Moreover, we propose a simple yet effective recovery approach that exploits Likelihood values in combination with spatial neighbours to perform outlier detection and correction.

Novel Contribution

The rest of this Chapter is organized as follows: Section 3.2 covers the basic principles of phase shift technique and introduces the error model we adopted with the mathematical notation. In Section 3.3 the novel probabilistic unwrapping method is presented, while in Section 3.4 the connection between the signal standard deviation and the standard deviation of the likelihood function is studied using Cramér-Rao lower bounds. In Section 3.5 the neighbourhood-based error recovery is explained and finally in Section 7.3 both synthetic and real-world experiments are displayed in order to assess the algorithm performances. The techniques presented in this Chapter have been published in [160] and [161].

3.2 Phase Shift

Phase-shift technique consists in generating a sinusoidal pattern spanning either the horizontal or vertical extent of the projector image plane (Fig. 3.2, left). During the acquisition, the pattern is spatially shifted so that each projector pixel encompasses an entire period after m subsequent shifts. Together with m, the period length λ (measured in pixels) of the sine pattern is chosen a-priori. It affects both the phase difference of two neighbouring pixels along with the spatial ambiguity of all pixels in the image. Indeed,

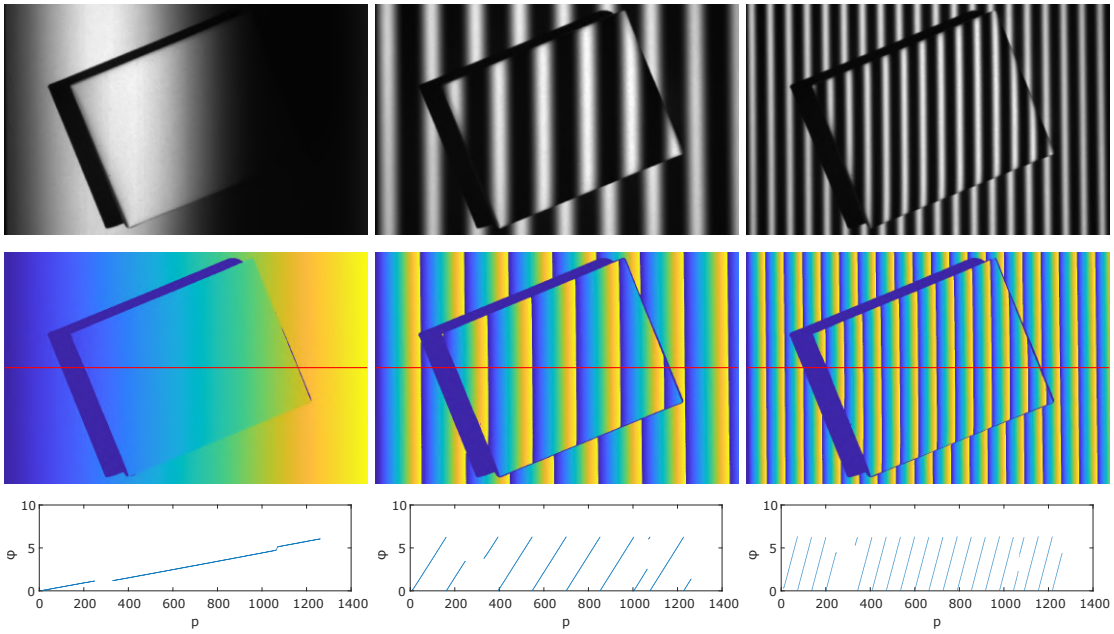


Figure 3.1: Acquisition process and different period lengths. First row shows one of the shifted light pattern projected on the scene, each one with a different period length (1300, 160 and 60 pixels). In the second row the correspondent phase maps are shown. Third row shows the computed phase values along the red line.

two adjacent pixels will exhibit a phase-difference of $\frac{2\pi}{\lambda}$. Consequently, points which are λ pixels away will be characterized by the same phase value. The first row in Figure 3.1 shows three examples of a simple scene while projecting one of the shifted light patterns. The different patterns differ in period length (from left to right: 1300, 160 and 60 pixels). After acquisition, the signal phase is computed for each pixel from observed intensity values in that point over time. From now on, we will address this output as phase map (or phase image), *i.e.*, an image where values consist of the computed phases at each image coordinate. Examples of phase maps are shown in Figure 3.1, second row.

The choice of period length λ is driven by two opposing needs. On one hand, a longer period will reduce the phase ambiguity along the image, with the extreme case of λ larger than the projector size causing no ambiguity at all (as in Figure 3.1, right). On the other hand, it is well known that the phase localization of the signal is proportional to the frequency, so we need to keep it small to increase the accuracy. Usually, λ is kept in the order of dozens of pixels (Fig. 3.2, Center) and an unwrapping step is used to distinguish different fringes. The disambiguation technique is based on the idea of projecting multiple different sinusoidal signals (at different periods $\lambda_1 \dots \lambda_n$) that can be combined to get a unique code for each projector's pixel.

The problem of the aforementioned multi-phase shift approach is that the signal acquired by the camera is perturbed by different noise sources, causing an imperfect phase estimation. Possible sources are, for example: (i) The thermic white noise of the cam-

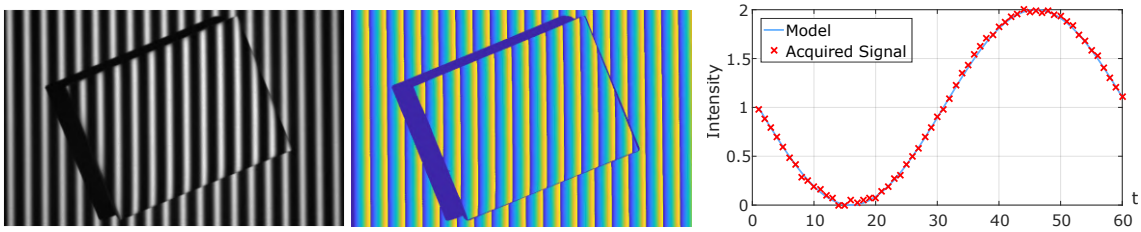


Figure 3.2: Left: Vertical sinusoidal pattern with a period length of 17 px. The pattern is shifted from right to left so that each pixel observe one complete period after m samples. Center: The recovered phase for each pixel. Since the period is shorter than the with of the projected image, we observe a phase ambiguity among the fringes. Right: Real-world example of the acquired signal compared to the projected ideal signal for a pixel with phase 0. Note how the samples are slightly noisy especially in the lower portion of the sine period.

era, especially at high gains or very short exposure times; (ii) The non-linear interactions between neighbouring pixels due to intrinsic properties of each material (micro-facets and reflections causing a wrong signal response); (iii) An imperfect mechanical and/or electronic functioning of the projecting device (particularly true for laser projectors), (iv) External noise sources like indoor ambient lighting subject to the oscillatory nature of the power outlet. Whatever the reason, a noisy phase estimation affects not only the accuracy of the reconstructed surface, but may cause a completely wrong phase unwrapping. Indeed, methods not particularly tolerant to phase errors produce a lot of erroneous codes for challenging materials, like brushed metals or glossy plastic.

In the following sections we briefly cover the basic Phase Shift method and introduce definitions and symbols that will be used throughout the Chapter.

3.2.1 Signal Model

Before describing our method, we start by formalizing the mathematical details of the projected signal and the corresponding noise model.

For any projector pixel P at coordinates (ξ, v), we project over time the following sinusoidal signal, discretized as a sequence of m subsequent samples:

$$Z(t) = \sin\left(2\pi\frac{t}{m} + \phi_P\right), \quad t = [0 \dots m). \tag{3.1}$$

Assuming a vertical pattern, the ideal pixel phase ϕ_P depends both on the period λ and its horizontal position on the image plane, according to the following simple relation:

$$\phi_P = 2\pi\left(\frac{\xi}{\lambda} - \left\lfloor\frac{\xi}{\lambda}\right\rfloor\right). \tag{3.2}$$

The value of ϕ_P cannot be measured directly, but it can be estimated from observed samples. Such values are subject to different noise sources affecting the acquisition process. As commonly performed in signal theory applications [139, 175], we derive our observation model assuming to acquire a set of noisy samples:

$$z(t) = \sin\left(2\pi\frac{t}{m} + \phi_P\right) + w[t] \tag{3.3}$$

where $w[t]$ is a zero-mean white Gaussian noise of variance σ^2. Albeit not completely motivated by physical considerations, the assumption we made on the statistical nature of the noise represents a good trade-off between the simplicity of the model and the observed behaviour. Such convenience is supported by our experimental evidence (see Sec. 7.3 for details).

Given the acquired signal $\mathbf{z}$, the Maximum Likelihood estimator of the phase is defined as:

$$\hat{\phi}_P = \underset{\phi_P}{\operatorname{argmax}}\, p(\phi_P; \mathbf{z}) \tag{3.4}$$

where, considering our assumption on the statistical nature of the noise, we have:

$$p(\phi_P; \mathbf{z}) = \frac{1}{\sigma(2\pi)^{m/2}} e^{-\frac{1}{2\sigma^2}\sum_{t=0}^{m-1}\left(z(t)-\sin(2\pi\frac{t}{m}+\phi_P)\right)^2}. \tag{3.5}$$

Switching to the natural logarithm of the likelihood and arranging the terms, we obtain the common non-linear least-squares minimization:

$$\hat{\phi}_P(\mathbf{z}) = \underset{\phi_P}{\operatorname{argmin}} \sum_{t=0}^{m-1}\left(z(t) - \sin\left(2\pi\frac{t}{m} + \phi_P\right)\right)^2. \tag{3.6}$$

Since the least squares best fit based of a lower order Fourier series is exactly equivalent to the truncated DFT, the phase of the acquired single-tone signal can be easily recovered as:

$$\hat{\phi_P}(\mathbf{z}) = atan_2(x, y) + \pi \tag{3.7}$$

with

$$x = \sum_{i=0}^{m-1} cos\left(\frac{2\pi}{m}i\right)z(i); \qquad y = \sum_{i=0}^{m-1} sin\left(\frac{2\pi}{m}i\right)z(i) \tag{3.8}$$

and where $atan_2 : \mathbb{R}^2 \to [-\pi, \pi]$ evaluates the correct arctangent angle of x/y selecting the appropriate quadrant based on the arguments signs.

3.2.2 Phase Recovery

Suppose that the sinusoidal pattern is composed of m shifted patterns to be projected on the scene, then the amount of shifting in each pattern is set to $\frac{2\pi}{m}$ radians. This way, after m shifted images, each pixel has observed one entire period of the sine pattern.

Given the set of captured grayscale images $I_1, I_2, ..., I_m$, we denote as $I_i(u, v)$ the intensity value observed in I_i at pixel coordinates (u, v). As we stated in the previous section, the resulting phase value $\varphi(u, v)$ is then computed as follows:

$$\varphi(u, v) = \frac{1}{2\pi} atan_2(x, y) + 0.5 \tag{3.9}$$

plugging the observed values $I_i(u, v)$ as the acquired signal $z(i)$ in Equation (3.8):

$$x = \sum_{i=0}^{m} cos\left(\frac{2\pi}{m} i\right) I_i(u, v); \qquad y = \sum_{i=0}^{m} sin\left(\frac{2\pi}{m} i\right) I_i(u, v)$$

Where $atan_2 : \mathbb{R}^2 \to [-\pi, \pi]$ computes the correct arctangent angle of x/y selecting the appropriate quadrant based on the signs of the arguments. Note that the angle is normalized in order to get phase values between 0 and 1.

The result of this process is an image of phase values computed at each pixel location, as shown in the second row of Figure 3.1. In these examples we fixed the sample number $m = 20$. In the leftmost phase image the period length is equal to the image width, so we obtain increasing phase values along horizontal sections of the image. Other images show the typical repeated fringes of phase values, caused by the periodic nature of the signal. The first choice seems convenient because it implies the total absence of ambiguities. In fact, one could think of using such phase values as a direct coding technique, but this solution is highly inaccurate because the accuracy of phase recovery is proportional to signal frequency. On the other hand, acquiring periodic values introduces ambiguity but leads to a better precision.

This is clear when observing third row of Figure 3.1. Here, phases along the horizontal red line are plotted against their spatial position. Each fringe in phase image corresponds to a segment in the plot, where the phase function is strictly monotonic. If we restrict the observation to a single known fringe, there is no ambiguity and we could directly compute the code from which the phase values were originated; thus, a robust technique which is able to disambiguate among the fringes is fundamental. In other words, if we are able to identify the correct fringe in which a phase value have been observed, we can easily compute the code.

3.2.3 Projector Codes and Phases

Computing projector codes from an observed phase is our primary goal, since it establishes a unique correspondence between camera and projector points. Phase values depend on projector coordinates by design, thus in this section we present the fundamental equations which model the relationship between these two values.

We identify projector's horizontal coordinates with variable ξ: these values range in $[0, \xi_{max} - 1]$, with ξ_{max} equal to projector width. Then, the projected signal's phase can be modelled as a periodic function in ξ, that we will identify as $\varphi(\xi) \in [0, 1)$. Note that phase values have been normalized in $[0, 1)$ using Eq. (3.9) from acquired images while projecting shifted patterns.

We denote with λ the period length expressed in pixels. This value controls the signal periodicity and implies $\varphi(\xi) = \varphi(\xi + k\lambda)$ with $k \in \mathbb{N}$. Phase function can be interpreted as the fractional part inside a fringe and is defined in the following way:

$$\varphi(\xi) = \frac{\xi}{\lambda} - \left\lfloor \frac{\xi}{\lambda} \right\rfloor. \tag{3.10}$$

If the observation is limited to a single fringe, phase values are increasing, not periodic and range between zero and one.

We label these fringes from left to right with sequential natural numbers in order to distinguish them. The fringe identifier is called fringe number and it is also a function of ξ. We denote the function identifying the fringe number as $\eta(\xi)$:

$$\eta(\xi) = \left\lfloor \frac{\xi}{\lambda} \right\rfloor \in \mathbb{N} \tag{3.11}$$

which can be interpreted as the integer part of the signal.

Given these definitions, we can easily express the relationship between projector coordinates ξ, phase value and fringe number in the following way:

$$\xi = (\eta(\xi) + \varphi(\xi))\lambda. \tag{3.12}$$

The simple intuition behind this equation comes from the definitions of $\varphi(\xi)$ and $\eta(\xi)$. Phase value is interpreted as the fractional part inside a fringe, while fringe number gives the integer part of a code by definition. As a consequence, if we put together integer and fractional part and then multiply by period length λ, we obtain exactly the projector coordinate ξ. Values of $\varphi(\xi)$ are available directly in the acquired phase image, while λ is a constant value. Thus, $\eta(\xi)$ is the only element which allows to disambiguate among points with the same phase.

3.2.4 Error Sources

Errors in phase estimation have a direct impact in the final reconstruction accuracy. We define as phase error the distance (in radians) between observed phase and the true phase value in the same projector coordinate:

$$\varepsilon_\varphi = d_c(\varphi^*(\xi), \bar{\varphi}(\xi)) \tag{3.13}$$

where $\bar{\varphi}(\xi)$ is the correct phase in ξ, defined in Eq. (3.10), while $\varphi^*(\xi)$ is the observed phase at the same point. The distance between two phases has to consider their periodicity. For this reason, we define d_c as a signed circular distance (in the range $[-0.5, 0.5]$) of two normalized angles:

$$d_c(\varphi_1, \varphi_2) = sign(\varphi_1 - \varphi_2)\, min\{|\varphi_1 - \varphi_2|, 1 - |\varphi_1 - \varphi_2|\}. \tag{3.14}$$

The formulation follows the fringe design: phase images are split in several stripes where values range in $[0, 1)$. Values near 1 are indeed spatially close to values around zero because they belong to different adjacent fringes. This is evident in second row of Figure 3.1. For example, we have that $d_c(0.8, 0.1) = 0.3$.

We already mentioned that several external factors could interfere in phase recovery process. Some error sources are directly caused by physical characteristics of the acquiring procedure, and most of the time such elements are not predictable, *e.g.*, reflections or unexpected changes in light conditions. Given the non deterministic nature of the process, we define phase error ε_φ as a random variable:

$$\varepsilon_\varphi \sim N(0, \sigma_\varphi). \tag{3.15}$$

We modelled ε_φ as a Normal Distribution with zero mean and standard deviation equal to σ_φ[1]. Even if the error is limited in the range $[-0.5, 0.5]$, the Normal distribution is still a good approximation for this model because in practice we never observed σ_φ greater than 0.1 radians. Therefore, less than 10^{-6} of the values drawn from such distribution fall outside the 5 standard deviations coverage. To study the impact of this error in the final code computation, we rewrite Equation 3.12 as:

$$\begin{aligned} \xi &= (\eta(\xi) + (\varphi^*(\xi) + \varepsilon_\varphi))\lambda \\ &= \lambda\eta(\xi) + \lambda\varphi^*(\xi) + \lambda\varepsilon_\varphi. \end{aligned} \tag{3.16}$$

It is evident that phase error ε_φ has a significant impact on final code value ξ, since the code is affected by an additive error proportional to phase length λ. We define the final code error in pixels as the multiplication of phase error ε_φ and phase length λ:

$$\varepsilon_\xi = \lambda\varepsilon_\varphi. \tag{3.17}$$

[1]Further details on this characterization will be given in Experimental section.

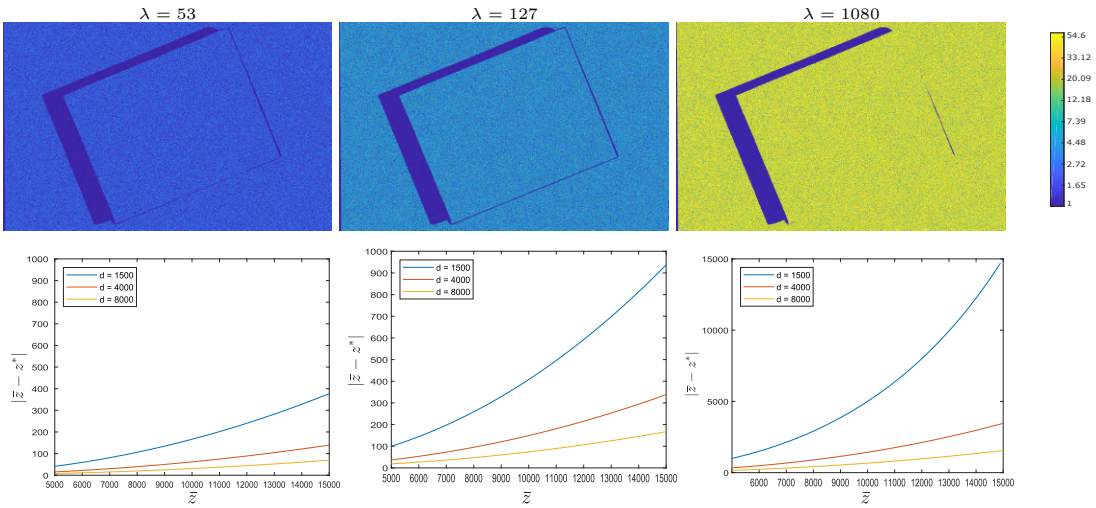


Figure 3.3: Coding errors with different period lengths (left to right: λ = 53, 127, 1080 px) and constant error standard deviation $\sigma_\varphi = 0.05$ rad. In second row the same values are used to measure depth errors for different disparities (d in three distinct curves). Plots show in x-axes the actual object depth and in y-axes the depth error after triangulation.

This implies that also code error ε_ξ is a random variable distributed as a Normal Distribution with the standard deviation amplified by λ:

$$\varepsilon_\xi \sim N(0, \lambda\sigma_\varphi) \tag{3.18}$$

indeed, period length λ plays a key role in absolute phase recovery since it magnifies code error ε_φ and thus has an important impact on the final triangulation.

In Figure 3.3 we show some examples on how phase length directly affects reconstruction accuracy: in each column we tested different period lengths, respectively $\lambda = 53, 127, 1080$. Since the projector's width is 1080 pixels, the last example contains a single fringe. Each figure has been generated simulating a Gaussian phase error with $\sigma_\varphi = 0.05$. The first row shows, for different λ, the correspondent code error $|\varepsilon_\xi|$ in the acquired phase images. While in the first picture we have an average error of 2.11 pixels, increasing the period length raises the value to 5 in the case $\lambda = 127$ and reaches 43 pixels when $\lambda = 1080$. As expected, this result shows that longer periods will lead to a low coding precision and thus in an inaccurate 3D reconstruction. Second row in Figure 3.3 shows the impact of code error ε_ξ when triangulating an object located at increasing distance $\bar{z}$. We simulated a camera-projector configuration, both with focal length equal to 1080. The camera observes a material point at $(0, 0, \bar{z})$ labelled with a code perturbed with a zero-mean Gaussian with standard deviation of σ_ξ pixels, and the depth of triangulated point is denoted with z^*. The error $|\bar{z} - z^*|$ is plotted in y-axes against the actual point depth $\bar{z}$ (in x-axes). Such error is also influenced by camera-projector disparity, so we simulated three different disparities which are comparable with the focal length. Each curve in the plots represents a distinct configuration. These plots confirm the severe

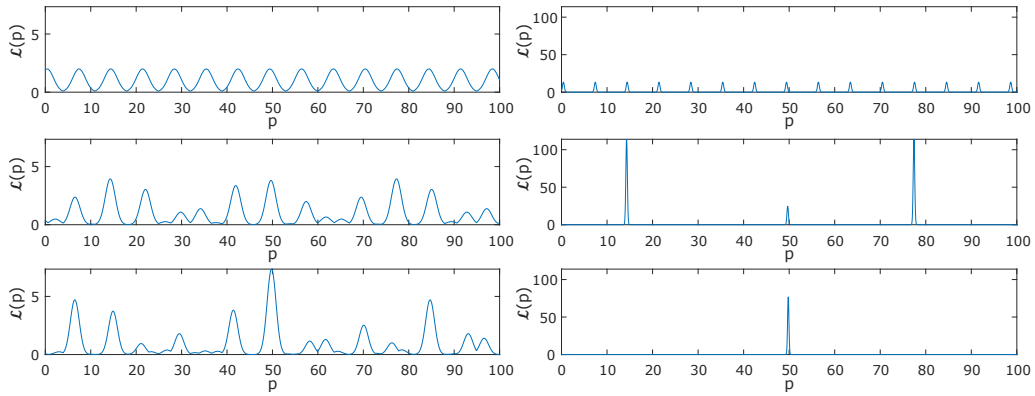


Figure 3.4: Examples of likelihood functions for different pattern configurations and phase standard deviations. First row shows $n = 1$ and $\lambda_1 = 7$, second row $n = 2$ and $\lambda_1 = 7, \lambda_2 = 9$, third row $n = 3$ and $\lambda_1 = 7, \lambda_2 = 9, \lambda_3 = 11$. We set $\sigma_\varphi = 0.05$ rad for both, and used two different estimations: $\sigma^* = 0.1$ (left) and 0.03 (right).

degeneration of 3D reconstruction precision when choosing a long period length.

3.3 Probabilistic Phase Unwrapping

Our approach employs n sinusoidal patterns of different coprime period lengths $\lambda_1, \lambda_2, ..., \lambda_n$. At each point, n phase values are observed, corresponding to the same unknown code ξ. The true (hidden) phases associated to each period length λ_i are denoted as $\bar{\varphi}_i$. We group such values in a phase vector:

$$\bar{\boldsymbol{\varphi}} = (\bar{\varphi}_1, ..., \bar{\varphi}_n) = (\varphi_1(\xi), ..., \varphi_n(\xi)) \tag{3.19}$$

Where each $\varphi_i(\xi)$ is defined as in Equation 3.10. The observed phases computed from intensity images are estimations of true phase values under different noise sources. We denote these values as

$$\boldsymbol{\varphi}^* = (\varphi_1^*, ..., \varphi_n^*). \tag{3.20}$$

Considering the periodicity of the signal, we characterize them as a vector of independent random variables distributed as Wrapped Normal distributions of support $[0, 1]$ and mean $\bar{\varphi}$:

$$\varphi_i^* \sim N_w(\bar{\varphi}_i, \sigma_i) \tag{3.21}$$

where σ_i is the standard deviation of phase values for pattern i. Given the small value of σ_i, such distribution can be efficiently approximated as a Gaussian [107], and the Probability Density Function relative to observed phases in the i-th pattern is defined as

$$f_i(\varphi_i^*;\xi,\sigma_i) = \frac{1}{\sqrt{2\pi}\sigma_i} e^{-\frac{d_c(\varphi_i^*,\bar{\varphi}_i)^2}{2\sigma_i^2}} \tag{3.22}$$

In this notation we place the variable φ_i^* before the semicolon and we consider all other parameters fixed. Note that parameter ξ is the distribution mean and it is implicitly included in the definition of $\bar{\varphi}_i$. Given a code ξ and a standard deviation σ_i, the integral of such function expresses the probability of observing phase value φ_i^* inside an interval.

Assume that we fixed the values of $\boldsymbol{\sigma} = (\sigma_1, ..., \sigma_n)$, the values of $\boldsymbol{\varphi}^*$ can be considered as independent samples from the distribution (3.21) of unknown mean. Therefore, we can switch the roles of phase value and projector code in Equation (3.22) considering φ^* as constant value and ξ as the variable. The Likelihood function $\mathcal{L} : [0, \xi_{max}) \rightarrow \mathbb{R}^+$ is defined as the product of all individual PDFs and it is defined over the space of projector codes:

$$\mathcal{L}(\xi;\boldsymbol{\varphi}^*,\boldsymbol{\sigma}) = \prod_{i=1}^{n} f_i(\xi;\varphi_i^*,\sigma_i) = \prod_{i=1}^{n} \frac{1}{\sqrt{2\pi}\sigma_i} e^{-\frac{d_c\left(\varphi_i^*,\left(\frac{\xi}{\lambda_i} - \left\lfloor \frac{\xi}{\lambda_i} \right\rfloor\right)\right)^2}{2\sigma_i}} \tag{3.23}$$

This function expresses the likelihood of projector codes ξ given that we observed $\boldsymbol{\varphi}^*$ as phase values. In this way we move code recovery from discrete to continuous domain, computing a measure of likelihood over the space of codes ξ, and the code which most likely generated the phases is the one which maximizes the likelihood function:

$$\xi^* = argmax_{\xi \in [0,\xi_{max})} \quad \mathcal{L}(\xi;\boldsymbol{\varphi}^*,\boldsymbol{\sigma}). \tag{3.24}$$

This approach does not exclude *a priori* any value of $\boldsymbol{\varphi}^*$, indeed, we preserve as much information as we can and compute the best possible code which generated the observed data. This is a fundamental aspect that distinguish this method with other unwrapping techniques, in fact number theoretical approaches fail when facing noisy observations. We prefer to accept such values and compute the best possible code rather than having no code at all: our goal is to exploit all measured phases as best as possible.

Following the theoretical discussion made so far, we now summarize our proposed stochastic code recovery approach. Before starting the acquisition, one must choose how many periods to project and their period length in pixels. To avoid ambiguity along the projector's codes space, all period lengths must be co-prime and their *LCM* smaller than the projector width (or height, if we project horizontal patterns). Moreover, one must specify the expected standard deviations of estimated phases, *i.e.* the vector $\bar{\boldsymbol{\sigma}}$.

It is worth spending a couple of words here, because $\bar{\boldsymbol{\sigma}}$ is in fact the only required parameter of our algorithm. First, we formulated the solution assuming a possibly different $\bar{\sigma}_i$ for each projected period. This accounts the fact that, for the microscopic characteristics of the surface, different sine frequencies may exhibit different signal noises. Moreover, longer periods are less localizable in space, so it is a good practice to use more samples m for long periods than for the short ones. Second, this approach allows us to not exclude a-priori any value of the estimated $\hat{\boldsymbol{\varphi}}$. Indeed, the vector $\bar{\boldsymbol{\sigma}}$ is essentially a

weight acting on all the acquired patterns. One may choose to keep extremely short or long periods, or to use very few samples, and modify the values of the relative $\bar{\sigma}_i$ accordingly. The choice of $\boldsymbol{\sigma}$ is a fundamental part in this process, since it controls the degree of selectivity of likelihood function. In fact, an overestimation of σ_i brings to a more tolerant likelihood, which could give more relevance to very unlikely values. On the other hand, an underestimation leads to a very selective function which drops close to zero in many intervals.

Figure 3.4 shows some examples of likelihood with a variable number of sinusoidal patterns (denoted as n) and sigma estimations. We simulated a 100-pixels projector and a vector of perturbed phase observations for $\xi = 50$. We assumed the same value for all σ_i and we used $\sigma_\varphi = 0.05$ to perturb phases. In the first row we set $n = 1$ with $\lambda_1 = 7$: in this scenario we have complete ambiguity since each maximum value has the same probability of being observed in every fringe. In the second row we have $n = 2$ and we added a pattern of length $\lambda_2 = 9$, while in the last row a third signal is added with $\lambda_3 = 11$. The introduction of further signals causes the exclusion of some codes and a higher likelihood value for others. In leftmost plots we overestimated the standard deviation setting $\sigma_i^* = 0.1\ \forall i$, while in rightmost plot we underestimated the same value setting $\sigma_i^* = 0.03\ \forall i$. In the first case the function includes a very large interval of values around the correct one and does not penalize the rest. In the case $\sigma = 0.03$ the function is more selective and does not have more than one or two peaks.

Once $\bar{\boldsymbol{\sigma}}$ is chosen and the n sinusoidal patterns (one for each period) are acquired, for each pixel of the camera we estimate the phases $(\varphi_1^*, ..., \varphi_n^*)$ using equation (3.9). At this point, for each camera pixel, we search for an initial integer estimate of the projector code by taking the value of $\dot{\xi} \in \{0, 1, \dots \xi_{max}\}$ for which $\mathcal{L}(\dot{\xi}; \hat{\boldsymbol{\varphi}}, \bar{\boldsymbol{\sigma}})$ is maximum. Due to the different spatial resolution between camera and projector, it is unlikely that each camera pixel observe exactly one projector pixel. Indeed, one of the strengths of phase-shifting is that one can precisely recover the projector coordinate with sub-pixel precision. In other words, the code ξ^* that maximizes the likelihood function can assume any real value between 0 and ξ_{max}. Unfortunately, due to the signed circular distance, we cannot give a direct analytical solution for the global maximum ξ^*. However, if we restrict the search in a small neighbourhood of $\dot{\xi}$ so that the circular distance never wraps, and take the logarithm of the Likelihood, $\mathcal{L}(\xi; \hat{\boldsymbol{\varphi}}, \bar{\boldsymbol{\sigma}})$ becomes a parabola for which we can compute the maximum by sampling three distinct points on it. Therefore, once we identified the integer part of the projector code $\dot{\xi}$, we recover the sub-pixel maximum ξ^* with the following formula:

$$\xi^* = \frac{\mathcal{L}(\dot{\xi} + \delta; \boldsymbol{\varphi}^*, \boldsymbol{\sigma}^*) - \mathcal{L}(\dot{\xi} - \delta; \boldsymbol{\varphi}^*, \boldsymbol{\sigma}^*)}{4\mathcal{L}(\dot{\xi}; \boldsymbol{\varphi}^*, \boldsymbol{\sigma}^*) - 2\big(\mathcal{L}(\dot{\xi} + \delta; \boldsymbol{\varphi}^*, \boldsymbol{\sigma}^*) + \mathcal{L}(\dot{\xi} - \delta; \boldsymbol{\varphi}^*, \boldsymbol{\sigma}^*)\big)} \tag{3.25}$$

We are still working with a process subject to several sources of noise and wrong measurements could lead to an incorrect code. In the case of coding errors, we could exploit spatial information to validate codes which are incoherent with respect to their neighbourhood. We explore this recovery option in the next section.

3.4 $\bar{\sigma}$ Lower-Bounds

Now that we described the general algorithm, we still need to clarify how to provide reasonable values for the vector $\bar{\boldsymbol{\sigma}}$.

One simple way is to empirically measure the phase error on a set of repeated experiments and compute its standard deviation.

There are two drawbacks in this approach. First, this sort of "calibration" must be performed every time a new object is acquired or any other condition changes the expected signal-to-noise ratio of acquired sinusoids. Second, the operation cannot be performed along with the acquisition because the exact phase of each acquired signal is not known, as it depends on the scene geometry.

To overcome that, the only way is to project the same known phase for all the projector pixels (essentially setting $\lambda = 1$) to collect statistics on its distribution. Unfortunately, this special pattern is useless for 3D reconstruction so it will just consume projector time to calibrate the parameter needed for our method.

Instead of directly calibrate $\bar{\boldsymbol{\sigma}}$, we can empirically measure the variance of the signal noise $w[t]$ and relate it to the variance of the phase estimator.

Since ϕ_P is an unknown deterministic parameter of some probability function via the unbiased estimator $\hat{\phi_P}$, the variance of the estimator is subject to the Cramér-Rao bound:

$$\text{var}(\hat{\phi_P}) \geq \frac{1}{J(\phi_P)} \tag{3.26}$$

where $J(\phi_P)$ is the Fisher information defined as:

$$\begin{aligned} J(\phi_P) &= E\left[\left(\frac{\partial \ln p(\mathbf{z};\phi_P)}{\partial \phi_P}\right)^2\right] \\ &= -E\left[\frac{\partial^2 \ln p(\mathbf{z};\phi_P)}{\partial^2 \phi_P}\right] \end{aligned} \tag{3.27}$$

By substituting (3.5) in (3.27) we obtain:

$$\begin{aligned} \ln p(\mathbf{z};\phi_P) &= \ln\left(\frac{1}{\sigma(2\pi)^{m/2}}\right) \\ &\quad -\frac{1}{2\sigma^2}\sum_{t=0}^{m-1}\left(z(t) - \sin\left(2\pi\frac{t}{m} + \phi_P\right)\right)^2 \end{aligned} \tag{3.28}$$

$$\begin{aligned}
\frac{\partial^2 \ln p(\mathbf{z};\phi_P)}{\partial^2 \phi_P} &= -2K\Big[\sum_{t=0}^{m-1}\cos^2\left(2\pi\frac{t}{m}+\phi_P\right) \\
&\quad -\sum_{t=0}^{m-1}\sin^2\left(2\pi\frac{t}{m}+\phi_P\right) \\
&\quad +\sum_{t=0}^{m-1}\sin\left(2\pi\frac{t}{m}+\phi_P\right)z(t)\Big] \\
E\left[\frac{\partial^2 \ln p(\mathbf{z};\phi_P)}{\partial^2 \phi_P}\right] &= -2K_1\Big[\sum_{t=0}^{m-1}\cos^2\left(2\pi\frac{t}{m}+\phi_P\right) \\
&\quad -\sum_{t=0}^{m-1}\sin^2\left(2\pi\frac{t}{m}+\phi_P\right) \\
&\quad +\sum_{t=0}^{m-1}\sin\left(2\pi\frac{t}{m}+\phi_P\right)E[z(t)]\Big]
\end{aligned} \tag{3.29}$$

where $K = -\frac{1}{2\sigma^2}$. Since

$$E[z(t)] = \sin\left(2\pi\frac{t}{m}+\phi_P\right) + E[w[t]] \tag{3.30}$$

and $E[w[t]] = 0$ by definition, we obtain:

$$\begin{aligned}
E\left[\frac{\partial^2 \ln p(\mathbf{z};\phi_P)}{\partial^2 \phi_P}\right] &= -2K\sum_{t=0}^{m-1}\cos^2\left(2\pi\frac{t}{m}+\phi_P\right) \\
\text{var}(\hat{\phi_P}) &\geq \frac{\sigma^2}{\sum_{t=0}^{m-1}\cos^2\left(2\pi\frac{t}{m}+\phi_P\right)}
\end{aligned} \tag{3.31}$$

Equation (3.31) states that the variance of our phase estimator must be greater than a value proportional to the variance of the Gaussian white noise σ^2 and inversely proportional to the number of samples m. Indeed, since $\cos^2(x) \leq 1$, we have:

$$\sum_{t=0}^{m-1}\cos^2(2\pi\frac{t}{m}+\phi_P) \leq m. \tag{3.32}$$

This makes sense, as we expect to reduce the estimation error either by acquiring more samples or by increasing the signal-to-noise ratio. With this new formulation, the value of σ can be empirically estimated online with the acquisition of the sinusoids, just by computing the observation errors with respect to our ideal projected pattern.

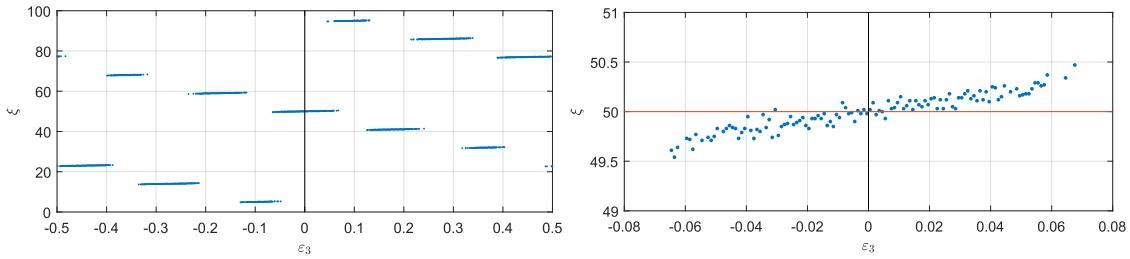


Figure 3.5: Code estimation in the case $n = 3$ with one erroneous phase value. The first two phases φ_1, φ_2 are slightly perturbed with a random zero-mean Normal noise with $\sigma_\varphi = 0.01$ rad. Error ε_3 on the last phase value ranges from -0.5 to 0.5 (shown in x-axis). The $argmax$ of each phase triplet is computed and the code ξ^* is plotted (in y-axis). Note that the correct code is $\xi = 50$. Rightmost plot shows the magnified central area of the other one. The red line identifies the correct code.

Even if we did not solved the problem completely, we experimentally observed that the empirically estimated phase standard deviation is actually very close to the theoretical lower-bound (3.31). Hence, by just using a value of $\bar{\boldsymbol{\sigma}}$ equal to the lower bound, we almost always obtain satisfactory results.

3.5 Neighbourhood-Based Recovery

Maximum Likelihood Estimation allows a robust unwrapping, even with noisy acquisitions. Nevertheless, if we have large errors, the maximum likelihood value could still not correspond to the correct code. For this reason, in this section we propose a neighbourhood-based technique that performs error detection and recovery based on local spatial information. Error detection is a relevant issue in any application, since the identification of wrong codes has a direct impact in the final reconstruction. Luckily, due to the multi-phase design and the co-primality property of period lengths, wrong codes tend to fall very far from the correct one. In other words, when observed phase values are incorrect, they result in a code significantly far from the exact solution.

3.5.1 A Faulty Unwrapping Scenario

Suppose to observe three phases $\boldsymbol{\varphi}^* = (\varphi_1^*, \varphi_2^*, \varphi_3^*)$. By chance, the first two have a small error, but φ_3^* is far from the correct $\bar{\varphi}_3$ (this typically happens when light conditions changed during acquisition). Since periods are coprime, the likelihood function computed only over φ_1^*, φ_2^* will exhibit several well separated local maxima (similar to Fig.3.4, second row), one of them corresponds to the correct code $\bar{\xi}$, while the others are generated from the signals ambiguity. When we add the observation φ_3^*, the likelihood function cannot exhibit any new local maxima (for its multiplicative nature) so the average separation between those peaks is not reduced. Therefore, if the resulting global maximum is

erroneous, this will be separated as well from the correct code $\bar{\xi}$.

Figure 3.5 clarifies this concept. We simulated a small random error on φ_1 and φ_2 and set an increasing phase error on φ_3. The leftmost plot shows values of code ξ^* (computed as the maximum likelihood) as the error ε_3 on the third phase changes. The rightmost figure shows the behaviour around the correct value $\xi = 50$. An error in φ_3 between -0.06 and 0.06 results in a correct projector coordinate, with a precision of ± 0.5 pixels, but when the error exceeds a certain level we have a significant change of projector codes in terms of spatial position. In this case, we jump from $\xi^* = 49.5$ to $\xi^* = 5$ when ε_3 is below -0.08 and the value closest to the correct code is $\xi = 60$, which is 10 pixels far from the solution.

This feature allows a simple error detection because erroneous values are clear outliers when compared to a limited spatial neighbourhood. In the system we are going to describe, if a code has not enough support from its neighbours, it is labelled as invalid. When this happens, likelihood function can be exploited a second time to retrieve other code candidates. In fact, if we consider the function peaks (thus not only the absolute maximum), we collect a set of values which show a high likelihood of being the correct code.

Figure 3.6 shows an example in a range of 50 pixels. Likelihood values are plotted along each row using an intensity colour scale. Maximum values correspond to correct codes most of the time, producing the peaks in the diagonal. Moreover, we can notice that other peaks are aligned at about 10 pixels distance. When the maximum is not the correct code, the second or third peak will lie on the diagonal; thus in case of an error, it is still a valid candidate to recover the correct code. In the next section we propose a validation system that exploits these properties to fix possible code errors.

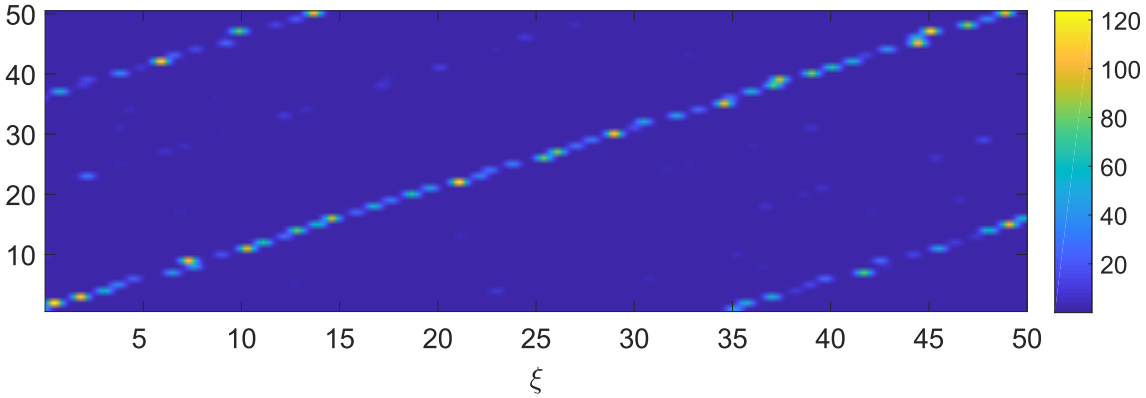


Figure 3.6: Likelihood values (plotted as intensity values in a colour scale) over the code space (x-axis) for each observed pixel (y-axis). Ideally, values lying on the diagonal should correspond to maximum values in each row.

3.5.2 Code Recovery by Exploiting Neighbouring Likelihoods

After acquisition, at each camera point (u, v) we collect a vector of phases $\boldsymbol{\varphi}^*(u, v)$ and a likelihood function $\mathcal{L}_{(u,v)}(\xi)$ is defined as in Equation 3.23. We then compute the first K local maxima to obtain a set of code candidates with a high likelihood. Such set is denoted as

$$C(u, v) = \{\xi_1^*, \xi_2^*, ..., \xi_K^*\} \tag{3.33}$$

For each camera pixel (u', v') we compute a score for each candidate by looking at its neighbourhood values, then we select the candidate which obtained the highest score. The score for each value $\xi_i^* \in C(u', v')$ with respect to its neighbourhood is computed in the following way

$$V(\xi_i^*, (u', v')) = \sum_{(u,v)} \omega(u' - u, v' - v) S(\xi_i^*, C(u, v)) \tag{3.34}$$

where $\omega_{(u,v)} : \mathbb{Z}^2 \rightarrow \mathbb{R}^+$ is a Gaussian kernel function with a standard deviation σ_ω weighting the points in the neighbourhood. The function S expresses the consensus of code ξ_i^* in the set $C(u, v)$. In our implementation, it returns the normalized Likelihood value of the code $\xi' \in C$ closest to candidate ξ_i^* and below a certain threshold ε.

$$S(\xi_i^*, C) = \begin{cases} 0 & \text{if } \nexists\, \xi' \in C \text{ st } |\xi_i^* - \xi'| < \varepsilon \\ \frac{\mathcal{L}(\xi')}{\mathcal{L}_{max}} : \xi' = min(\xi_i^*, C) & \text{otherwise} \end{cases} \tag{3.35}$$

Where $min(\xi_i^*, C)$ returns the point $\xi' \in C$ which minimizes the distance from ξ_i^* to any code in set C:

$$min(a, C) = \underset{c \in C}{argmin} \, |a - c| \tag{3.36}$$

We return likelihood value of the closest code among its neighbours because it is a good indicator of the relative importance of a code with respect to other values in C. In other words, the higher its likelihood is, the more we trust a neighbour. Since we can not compare likelihood values coming from different points, we measure its relevance dividing it by the maximum value. The value of ϵ is usually set to the minimum period length. The final code at (u', v') is computed as the value ξ_{max}^*, which maximizes function V:

$$\xi_{max}^* = \underset{\xi_i^* \in C(u', v')}{argmax} \, V(\xi_i^*, (u', v')) \tag{3.37}$$

that is the code which receives the higher support from its neighbourhood.

3.6 Experimental Evaluation

To provide an exhaustive evaluation of the proposed unwrapping method, we performed both synthetic and real-world experiments. In particular, synthetic experiments allowed us to have a complete control over the parameters involved in the process, while real experiments show the ability of the proposed method to obtain good results in a variety of real-world applications. The experimental section is divided in three parts. First, we focus on the parameter $\boldsymbol{\sigma}$ and on its impact on the final unwrapping correctness Second, we analyse the potential of the proposed neighbourhood recovery and finally we compare quantitatively and qualitatively our technique with respect to other state-of-the-art approaches.

The real-world experiments were performed with a structured-light scanner developed in our lab. The device is composed by a single MatrixVision BlueFox3 5Mpix camera and an LG projector with a resolution of 1920×1080 px. Camera and projector relative pose was calibrated using the fiducial markers descried in [21]. We manually corrected the projector gamma to obtain a camera signal response as linear as possible. Once calibrated, the defined gamma settings remained unchanged throughout the experiments.

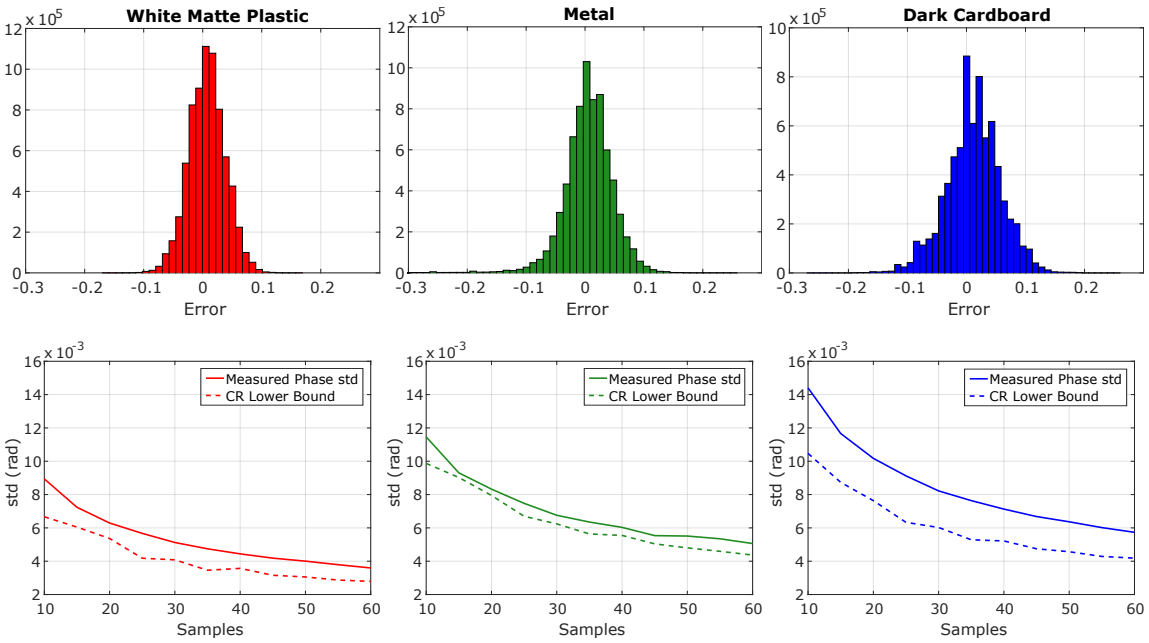


Figure 3.7: Relation between the acquired signal distribution and standard deviation σ (top row) with the theoretical lower-bound and the empirical phase standard deviation $\bar{\boldsymbol{\sigma}}$ (bottom row) for the three different chosen materials.

3.6.1 Phase and Signal Standard Deviation

We performed a first set of experiments to assess the validity of our statistical noise model on a real scene. We considered three different kinds of planar surfaces, namely: matte white plastic, a brushed aluminium metal sheet and a dark coloured cardboard. The goal is to compare the acquired signal standard deviation σ with the standard deviation of the phase estimator $\bar{\sigma}_i$. To do that, we generated a pattern composed by a variable number of samples m with a period equal to one. This way, each pixel P in a scene observed exactly the same true phase $\phi_P = 0$, allowing us to empirically estimate the standard deviation of the phase error among pixels. Similarly, by knowing the phase, we computed the error of acquired signal with respect to the projected ideal sinusoid. In this way we can test if our zero-mean white Gaussian noise assumption is supported by experimental evidence.

In Figure 3.7 we show the results of this experiment for the three different materials. On the top row we plotted the histogram of the error of raw acquired signal for a sinusoidal pattern composed by $m = 60$ samples.

In all cases, the signal was analyzed by collecting the errors in a ROI of size 400×300 pixels selected to cover the central region of the projected area. We can see that the empirical distributions follow quite well our supposed Gaussian model with a mean close to zero. Especially for the cardboard material, we report a little positive bias that we guess is caused by blooming effect. In fact, blooming clearly bias the acquired intensity by always overshooting the actual value. Nevertheless, we think that the proposed assumptions are fair enough to approximate the real behaviour of the phenomenon. For each material, we computed the empirical standard deviation σ (from the $m = 60$ samples case) to compute the lower-bound according to (3.31).

In the second row of Figure 3.7 we plotted both the lower-bound and the empirically measured phase standard deviation varying the number of samples m for the three different materials. As expected, the empirical standard deviation of the estimated phase is proportional with the standard deviation of the signal. In fact, the white matte plastic exhibits a lower error due to its nearly Lambertian nature. Metal sheet is more noisy due to small reflections and dark cardboard is the worst since the low albedo is heavily decreasing the signal-to-noise ratio. In all the cases the estimated phase standard deviation is slightly above the theoretical lower bound computed from the signal. This validates our claim that the given bound can be effectively used to give a good estimation of the unknown $\bar{\boldsymbol{\sigma}}$.

3.6.2 Sigma Estimation

In our approach we model phase error as a zero-mean Gaussian distribution, and we characterize such random noise with a standard deviation vector $\boldsymbol{\sigma}$: the analysis of such parameter defines the starting point of the subsequent evaluation. Since error sources come from real-world phenomena, the first group of tests have been designed to observe the empirical noise distribution affecting actual phase measurements. In all the experiments the acquired objects were located at about 1m away from the projector.

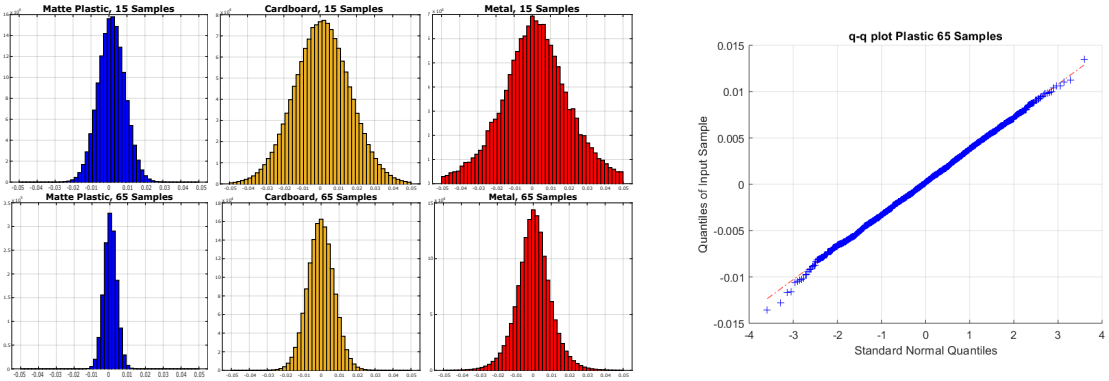


Figure 3.8: Empirical validation of Gaussian noise assumption: in these acquisitions all expected phases were equal to zero. Left: distributions of measured phase values for different materials (from left to right: matte plastic, cardboard and metal) and samples (1st row 15 samples, 2nd row 65 samples). All histograms are centred at zero and clearly exhibit a bell shape, that validates the Gaussian assumption on noise affecting phase measurements. Right: quantile-quantile plot of observed phase values with respect to a Normal distribution. The good fitting of our data with respect to the line means that the Normal distribution is suitable for representing the phenomenon.

As in the previous tests, we generated a mono-phase pattern (setting $\lambda = 1$), i.e. a sequence such that all acquired pixel will exhibit the same expected phase value, equal to zero. This pattern of course has no utility for unwrapping, but since the phase is known for all pixels, we used it in this preliminary study to empirically characterize the noise affecting phase estimation. Since the acquired scene has a direct impact on the signal to be processed, we analysed the response of different materials as we did before (white matte plastic, dark cardboard and silver metal).

In the first experiment the signal was acquired for each material, then phase values were computed over all camera pixels. Histograms on Figure 3.8 display the resulting distributions of estimated phase values for each material in the case of 15 (first row) and 65 samples (second row). The number of samples directly affects phase accuracy: in fact for a higher sampling rate each surface shows a tighter histogram, and consequently a smaller standard deviation. As we expected, each material exhibits a different response in terms of noise, but all histograms show a coherent behaviour: they are centred on zero and clearly exhibit a bell shape, that suggest a Gaussian data distribution. Such hypothesis is validated by the q-q plot shown in right side of Figure 3.8. In this graphical test the quantile values from a Standard Normal Distribution are plotted against quantiles from a sub-sample of our data. The resulting figure shows a good correspondence between the two distributions, which supports our assumption that the error can be modelled as a zero-mean Gaussian random variable.

The leftmost plot of Figure 3.9 shows, for each material, the standard deviation values obtained increasing the number of samples (from 5 to 70). As we already discussed, the standard deviation decreases as the number of samples raises. In particular, different

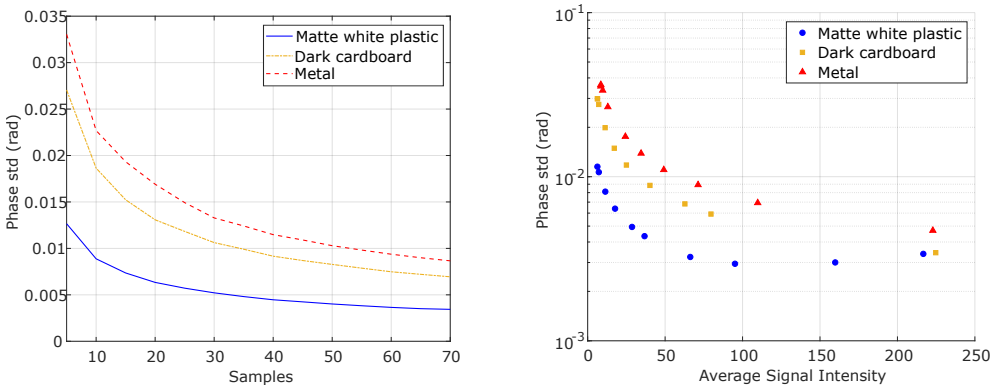


Figure 3.9: Left: Standard deviation observed on phase values varying the number n of pattern samples. Right: phase standard deviation measured with respect to average signal intensity of acquired images. In both cases, three different materials had been tested.

materials exhibit different variability but they have the same behaviour when varying the sample number. In detail, the matte white surface provides the best response, while a darker or metallic object causes a more noisy signal.

Next, we exploited the same experimental setup to examine the relation of phase standard deviation with respect to the average signal intensity. In the second experiment we projected the previous pattern with $\lambda = 1$ and $m = 30$. Then, for each acquisition, we gradually changed the global illumination of the scene and recorded the average image intensity (from 0 to 255). The rightmost plot of Figure 3.9 shows the standard deviation values for each material plotted against the average signal intensity (note the logarithmic scale in the y-axis). In correspondence of darker scenes (from 0 to 50) we notice an initial significant decreasing in the standard deviation for all the tested materials, then when optimal conditions are passed (around 100 for matte plastic), a slight rising in the values can be noticed. Of course the optimal light condition is determined by the scene features and it should be adjusted depending on the material properties.

After assessing the plausibility of the Gaussian model, we now study the effect of parameter $\boldsymbol{\sigma}$ on the phase map accuracy. To do so, we simulated a virtual projector and generated three phase maps, with period lengths respectively of 11, 13 and 17 pixels. The phase values were independently perturbed with a zero-mean additive Gaussian noise (our theoretical noise model) characterized by a known standard deviation $\boldsymbol{\sigma}$. Note that we made no assumptions on the relation among the σ_i: each period can be characterized by a different sigma. Tests were performed in the following way: first, we randomly selected $\boldsymbol{\sigma}$ as a 3D vector inside a cube of size 0.1, and used it to perturb the ground truth phase maps; then we generated another random vector $\boldsymbol{\sigma}^*$ to be used as sigma estimation in the Likelihood function. Finally, for each estimation test we computed the RMS (Root Mean Square) of coding errors, in pixels.

The leftmost plot in Figure 3.10 shows the Euclidean distance between real and estimated sigma vectors, that is $|\boldsymbol{\sigma} - \boldsymbol{\sigma}^*|$, against the final error RMS values in pixels (on y-axis). In case of small estimation distances (from 0 to 0.03) the error ranges from 0.2

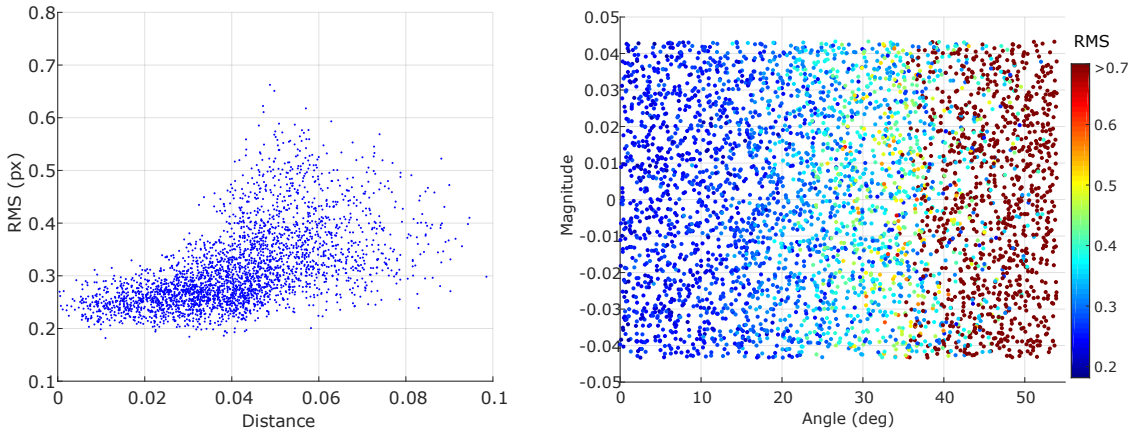


Figure 3.10: Impact of different sigma estimations on the final RMS (in pixels). Left: effect of distance $|\boldsymbol{\sigma} - \boldsymbol{\sigma}^*|$ on the final coding error. Right: the same experiment, separating vector relative orientation (on x axis) and magnitude of estimated sigma vector (better viewed in colour).

to 0.3 pixels, while as the distance increases the RMS of the coding error exhibits more variability and reaches values up to 0.7 pixels. Many samples with high distance (not shown in the figure) resulted in an error of several pixels and were considered outliers. In the rightmost plot of Figure 3.10 we used the same data of the previous experiment and separated the angle formed by $\boldsymbol{\sigma}$ and $\boldsymbol{\sigma}^*$ (x-axis) from their magnitude difference (y-axis); each point is coloured with the corresponding RMS value. The RMS values that we obtained is essentially uniform along the vertical direction (that is the magnitude difference); and they grow proportionally with the relative angle between the vectors. This result implies that, in terms of decoding accuracy, the relative angle between $\boldsymbol{\sigma}$ and its estimation $\boldsymbol{\sigma}^*$ is more relevant than their distance. In other words, it is in general sufficient to guess the correct proportion among the σ_i instead of their optimal values.

3.6.3 Comparisons on Synthetic Setup

In this section we present the set of experiments performed to compare the accuracy and reconstruction capabilities of our technique with respect to other unwrapping methods. We performed both synthetic and real-world tests: the former let us test the system excluding any uncontrolled factor to obtain a comparison that is as precise as possible, while the latter allowed us to reveal the potential of the proposed method in challenging real-world scenarios.

We decided to compare our method with a number-theoretical unwrapping approach described in [125] and a classical Graycoding disambiguation. Spatial phase unwrapping has not been considered for two main reasons: first, it allows the separate recovery of connected surface areas and makes some assumptions about the geometry of the system (large discontinuities are not supported); this limitations restrict a lot our target, that is more general-purpose. Second, it employs only one phase map: this means that the mea-

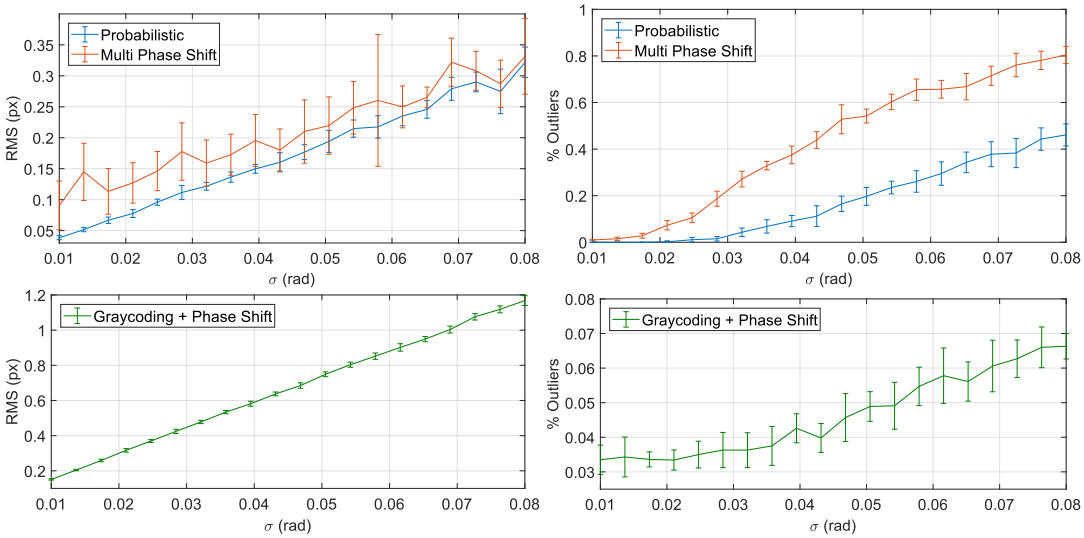


Figure 3.11: Comparison of Probabilistic approach with respect to number theoretical method [125] (1st row) and Graycoding disambiguation technique combined with just the first phase pattern (2nd row). Left: coding RMS is plotted (on y-axis) against increasing phase error $\boldsymbol{\sigma}$. Right: percentage of outlier codes over all decoded points (on y-axis) varying phase error $\boldsymbol{\sigma}$. A code was classified as outlier if its distance from the correct code is greater than half the shortest period length.

surements errors on such values have a direct impact on the final reconstruction, despite a (hopefully) optimal quality map. Indeed, the precision offered by a single phase map can not be further improved, even with a perfect disambiguation technique. This drawback also holds for all approaches based on a single phase map, like the minimum phase method [12] or Graycoding. For this reason, we can group all these methods under a general "Graycoding approach" that offers a good fringe disambiguation, using just one phase measurement.

The synthetic setup emulates the real conditions that we observed in the previous section: we generated three sinusoidal patterns of length 17, 23, 27 and simulated a 1080p projector. Note that in these experiments we did not apply the neighbour-based recovery, but simply choose the code which exhibit the higher likelihood value.

The first row of Figure 3.11 shows the code error RMS (left) and codes outlier percentage (right) for both number-theoretical approach and our probabilistic method (displayed as two separate curves). The experiment was performed by generating a phase map for each period length, perturbed according to a Gaussian random noise with increasing sigma values (from 0.01 to 0.08, on x-axis). Then, both unwrapping methods have been applied on the same data and the coding RMS and outliers percentage were computed. The outlier percentage was computed considering as outliers all the codes such that their distance from the correct code was greater than half of the minimum period length; in the plots the ratio between the number of outliers and the total number of codes is displayed. Such process has been repeated 10 times, and the errorbars denote clearly that the number-

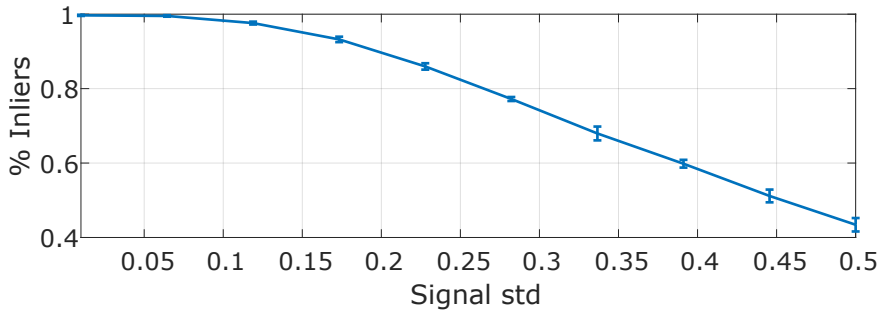


Figure 3.12: Projector delay recovery: percentage of correctly estimated shifts (inliers) against the acquired noise standard deviation.

theoretical approach was in general more unstable with respect to our proposed method. Moreover, probabilistic technique exhibits a linear behaviour as σ (i.e. the amount of noise) increases, offering more stable results. In terms of outlier percentage, probabilistic approach offers almost 100% of correct codes when σ is smaller than 0.03 rad, while number-theoretical method exhibits a small outlier percentage (up to 20%) at the same low noise levels. In case of high noise level, probabilistic approach recovered almost 60% of codes, while number-theoretical provides only 20% of them. In the second row of Figure 3.11 we used only the first phase map from the previous tests and performed fringe disambiguation through Graycoding to compute projector codes, then we computed code error RMS and outlier percentage. Graycoding method offers a good disambiguation, but the recovered codes are less accurate because they are subject to the measurements of a single phase map, while our technique and number-theoretical exploit multiple phases to disambiguate and recover the correct code. In the case of Graycoding, outlier percentage never exceed 10% even at high noise levels: this was expected because the error can not be greater than a period length. Despite this, such inlier ratio can not be accepted because the coding precision is extremely inaccurate if compared with probabilistic or number-theoretical approaches. Moreover, we considered an infallible fringe disambiguation: in real-world scenarios a disambiguation signal could lead to more outliers, especially along the code discontinuities.

3.6.4 Projector Delay Recovery

One of the key strengths of our method is the resilience against noisy phase estimations. To demonstrate that, we tested the approach with the challenging task of recovery the correct code even if the projector and camera are not synchronized.

This can be useful in practice because ensuring a proper synchronization of the camera frames with the projected patterns usually requires a custom electronics that increases the cost and complexity of the scanner.

For simplicity, we assumed that the exposure time of the camera is far lower than the display time of each projected pattern. This way, all the acquired samples $z(t)$ are affected

by a fixed unknown integer shift but the camera never acquires in-between two projector frames.

We generated a synthetic 3-periods signal composed by $m = 60$ samples (as in previous experiments) perturbed with a variable zero-mean noise with variance σ^2. Then, we shifted all the patterns by a random shift $0 \leq k \leq m$. Afterwards, we run our method for each possible circular shift k of the signal collecting, for each pixel, the value of the maximum likelihood obtained. Pixel wise, we compared all the maximum likelihood values to select the shift corresponding to the higher one. The result is that each pixel votes for the shift producing the higher maximum likelihood of the codes.

Since the data was synthetically generated, we marked as inliers all the pixels that actually voted for the correct shift and plotted the inlier percentage against the noise in Fig.3.12.

The experiment shows an inlier percentage greater than 50% (sufficient to correctly recover the unknown shift) for input signal noise std. up to 0.45, which is more than 5 times the noise levels we measured for all the three different materials in Fig.3.7. Therefore, we are confident that in the vast majority of real world cases the synchronization may be avoided with no severe consequence on the unwrapping result.

3.6.5 Neighbourhood-Based Recovery

In this section we evaluate the performances of the proposed neighbourhood-based fault recovery. To this end, we run some synthetic tests to assess the feasibility of this approach and its sensitivity to the parameters. We employed the setup from the previous section and simulated the acquisition of a planar surface. In Figure 3.13, the first 50 pixel values are plotted against their corresponding codes: probabilistic (left) and number-theoretical (right) unwrapping are displayed. Since we are observing a plane, we expect all the codes to increase linearly with the camera pixel (this gives us the ”ground truth” to compare with). In the case of probabilistic approach we found that outliers are actually (i) far from

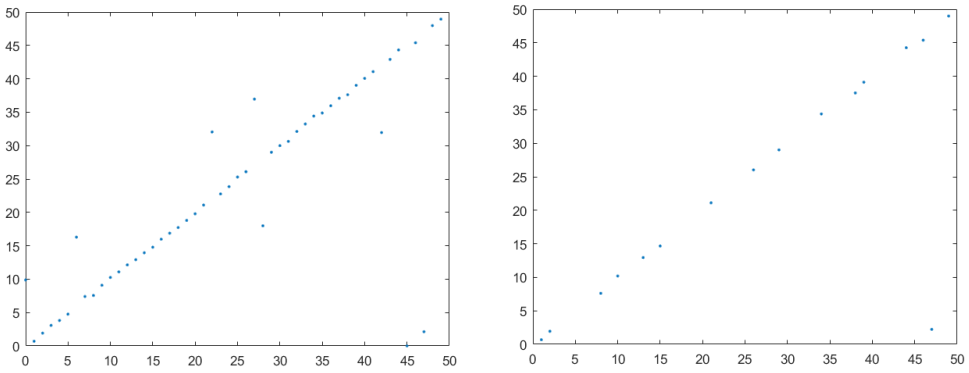


Figure 3.13: Code recovery comparison. Left: codes computed with our probabilistic method choosing maximum likelihood values. Right: codes computed from the same phase images using a number-theoretical approach for disambiguation.

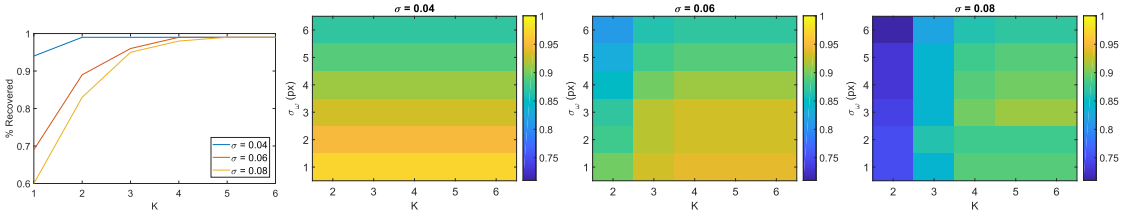


Figure 3.14: Left: ratio of correctly recovered points wrt the number of peaks extracted from the likelihood function. Other figures show the same ratio combining number of peaks K and different window size.

the correct value (ie. far from the diagonal) and (ii) sparse, showing the characteristics of salt and pepper noise. Those conditions fits perfectly our approach since the sparsity of the faulty unwrappings suggests the usage of the neighbourhoods and the severity of the code error allows a simple identification of the outliers. Indeed, the outlier neighbours have been often correctly unwrapped, so the neighbourhood values could be useful in the recovery of the correct code.

We analysed the recovery behaviour varying K and σ_ω, which respectively control the number of likelihood peaks for each code and the relevance of the involved neighbours. To this end, we designed a first experiment to verify the presence of the correct code among the first K likelihood peaks in each observed point. We simulated $n = 3$ phase patterns of lengths $\lambda = 17, 23, 27$ and we computed the percentage of correct codes within the first K peaks for different noise levels (phase errors $\sigma = 0.04, 0.06, 0.08$). Results are shown in leftmost plot of Figure 3.14: the case $K = 1$ corresponds to the approach of previous section (without neighbourhood recovery) that is taking the code associated to the maximum likelihood value. In the case of a low noise level, $K = 2$ is sufficient to collect 90% of correct codes, while at higher noise levels 3 or 4 peaks are enough to include the exact code in the set of candidates. This test shows that, in case of errors, we can still exploit the likelihood values to recover the correct code exploiting a small number of the function's peaks as candidates.

In the second experiment we evaluated the recovery ability of the voting technique described in Section 3.5. In particular, we varied the parameters K and σ_ω to compute the percentage of correctly recovered points for each combination. The intensity images in Figure 3.14 display the results: each plot is characterized by a different noise level (from left to right 0.04, 0.06 and 0.08) and covers several values for K (x-axis) and σ_ω (y-axis); the colour scale denotes the percentage of recovered codes. We already observed that, in a low noise configuration, two peaks are enough to find the correct code among the candidates. Indeed, the leftmost image ($\sigma = 0.04$) shows that K is irrelevant and the results are better involving less pixels in the voting window. In all tests, the percentage of recovered values results always above 90%. In the other two plots we have higher noise conditions, but in both cases the percentage decreases for $\sigma_\omega > 3$. In general, we have good results with small values of σ_ω and 4 or 5 peaks, depending on the magnitude of noise.

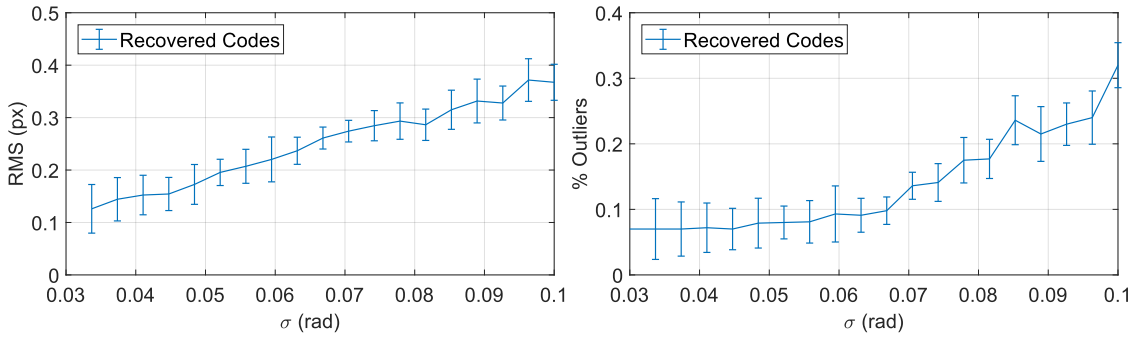


Figure 3.15: Code RMS (left) and outlier percentage (right) for points recovered through the proposed neighbourhood-based technique.

Finally, we measured error RMS and outlier percentage on recovered codes. This experiment can be compared with Figure 3.11, in which we showed the performances of our approach in the same setting but without neighbourhood recovery.

Following the results of the previous parameter analysis, we chose $K = 4$ and $\sigma_\omega = 3$ and run our code recovery method 10 times for different values of σ. Results are shown in Figure 3.15. The RMS error of the recovered codes (left plot) is in general lower than the curve obtained by applying our method without neighbourhood recovery: this proves that the recovered codes are indeed correct. Similarly, the right plot shows a strong reduction in the number of outliers, in particular for noise level below 0.07 radians. Overall, the test suggests that the neighbourhood recovery greatly improves the number of correctly decoded points, resulting in a denser triangulated surface.

3.6.6 Qualitative Evaluation

In the following experiments we tested the reconstruction ability of the proposed method on some real-world acquisitions. In particular, we decided to compare the quality of coded images and reconstructions quality with other techniques. The employed setup was the same used in the previous section: we projected 20 samples of three sinusoidal patterns with lengths 17, 21 and 31 pixels.

In the first experiment we focused on an ideal Lambertian object to compare acquired coded images in a scenario subject to low noise conditions. We scanned a white chalk low relief (Fig. 3.16, top-left) and computed projector codes using our method, number-theoretical approach and Graycoding disambiguation. The unwrapped values computed by our method are shown in Figure 3.16 on bottom-left corner. The bottom-right plot of Figure 3.16 displays the code values for all three methods which have been sampled along the segment displayed as a thick black line in the corresponding coded image (bottom-left). Graycoding results are in general more noisy with respect to the others (especially along code discontinuities) because it exploits only the first phase image to locate the fractional part of codes. Values resulting from number theoretical and probabilistic approaches are similar but the proposed method appears more stable and exhibits no dis-

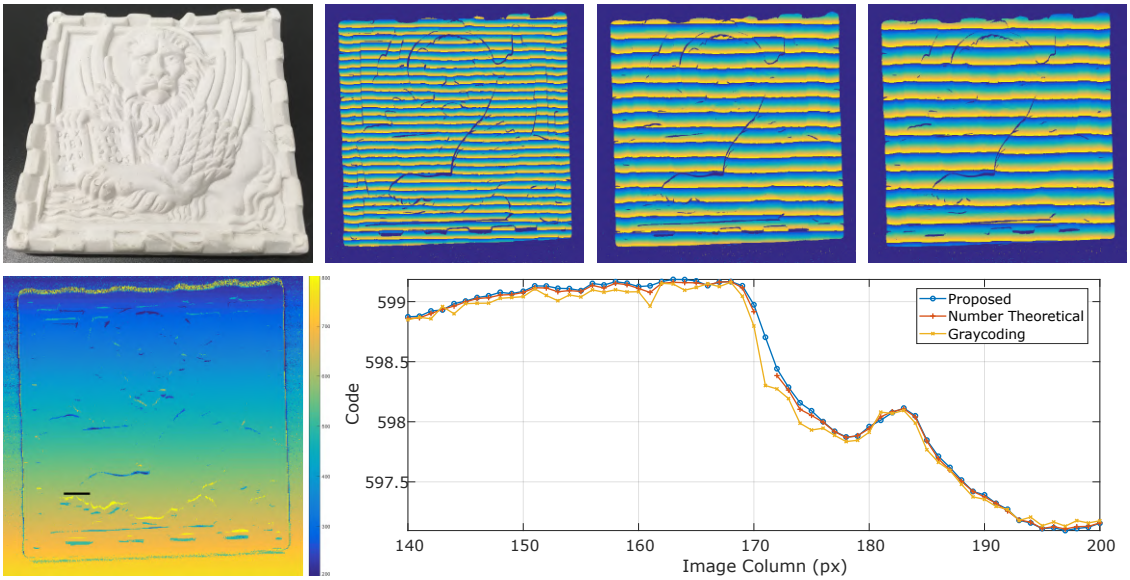


Figure 3.16: First row: the scanned object and the three acquired phase maps (period length 17, 21 and 33). Second row: computed projector codes using our method, the black segment denotes the selected codes plotted on the right as example. The rightmost plot shows the code values from the three techniques (probabilistic, number theoretical and Graycoding using just the first phase image). Graycoding results are in general more noisy and number theoretical exhibits some discontinuities due to algorithm failures.

continuities (ie. no coding errors). The missing values from the number theoretical unwrapping are mostly due to a noisy estimation of phases that leads the method to fail, given its discrete nature based on look up tables. This frequently happens in some areas where the signal acquisition is made difficult by particular surface responses or scarce illumination. The effects on the final 3D reconstruction are visible in Figure 3.18 (first row), showing the final triangulated meshes acquired during this experiment using the number-theoretical (left) and our technique (right).

In the second test we scanned a silvery metal sink (Fig.3.17, top-left), a particularly challenging object for the reflective nature of the surface and the concavity of the shape. We have chosen this kind of item to stress the methods in case of high noise levels, that involves a higher phase standard deviation and thus the presence of several outliers. Figure 3.17 (first-row) shows the obtained 3D reconstructions for probabilistic, number-theoretical and Graycoding methods respectively. The reconstructions clearly show that the proposed technique allows for the computation of a larger number of correctly unwrapped phases with respect to other techniques. This results in a denser triangulated mesh with lower error. The profile plots shown in the second row (corresponding to the red line superimposed in the 3D rendering) confirms the higher quality the acquired surface. Finally, Figure 3.18 qualitatively shows that the proposed unwrapping method offers a better reconstruction even in problematic regions where the signal is affected by shadows and inter-reflections caused by surface geometry.

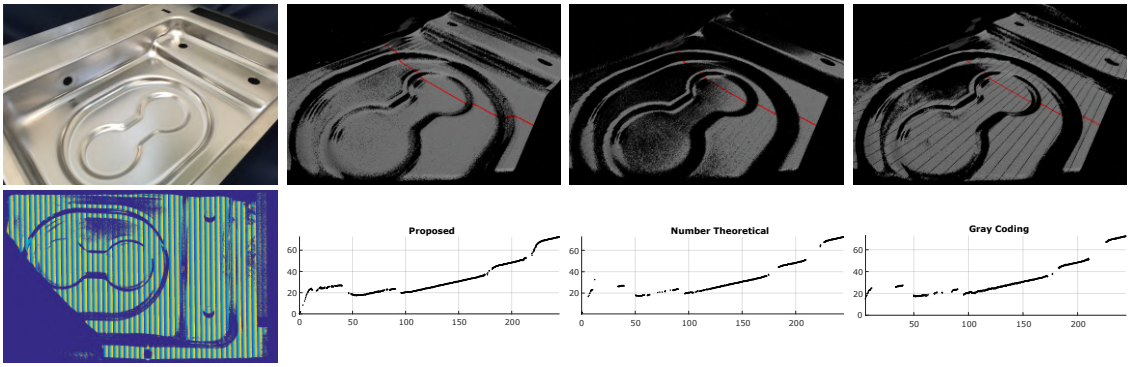


Figure 3.17: First row: picture of the scanned object and the 3D range-maps obtained applying respectively (from left to right): proposed, number-theoretical and Graycoding. Second row: one of the three phase maps used for Graycoding and the cross sections of 3D data obtained by slicing the scene in correspondence of the red line.

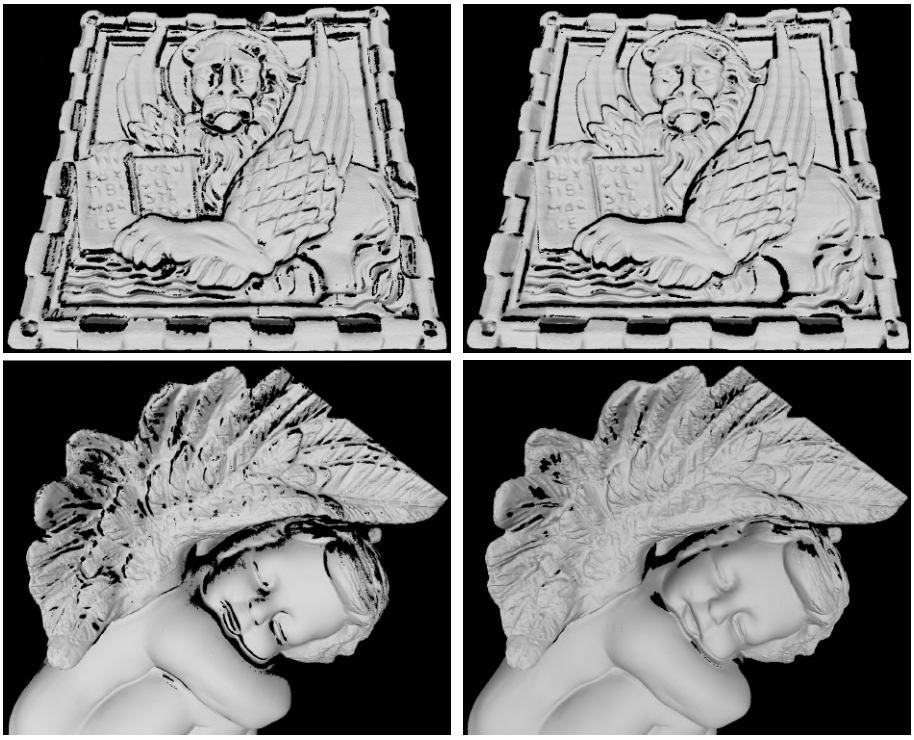


Figure 3.18: Qualitative comparisons of reconstructed scenes. Left: number theoretical approach, right: proposed probabilistic technique. We display two simple objects acquired using three fringe patterns of different period lengths. Our technique offers a better reconstruction even in simple scenarios, especially in areas where the signal results noisy due to the surface geometrical characteristics.

3.7 Conclusions

In this Chapter we presented a robust technique for accurate code recovery and phase unwrapping for multi-phase structured-light scanning. Specifically, we proposed a probabilistic approach in which we model the phases as samples from a Wrapped Normal distribution with a given variance and unknown mean. This modelling allows us to recover the projector codes in terms of a Maximum Likelihood Estimation of each distribution mean. We also exploit the Cramér-Rao bound to derive a lower-bound for the phase estimator variance according to the variance of acquired signal. This way, the estimation of the vector $\bar{\sigma}$ (the only parameter required by our method) can be efficiently computed on-line with the acquisition of the sinusoidal patterns. Additionally, our method exploits likelihood functions and spatial neighbourhood consensus to further detect coding errors and eventually recompute the correct value.

We performed multiple experiments to assess accuracy and robustness of the method. Both in synthetic and real-world scenarios, our approach outperformed similar unwrapping methods based on combinations of multiple sinusoidal signals, without requiring specific patterns to disambiguate among fringes (like the common Graycoding technique). From the the experiments, we observed how the empirically estimated phase variance is close to the theoretical lower-bound, suggesting that we can effectively use that value for the subsequent code recovery. In particular, when compared with a number theoretical algorithm, our technique offers a more accurate code recovery and a smaller outlier percentage, at a price of a slower unwrapping compared with similar methods. One possible limitation of our method with respect to a number theoretical approach is the algorithm execution time: while other approaches rely on pre-computed values, our technique needs to compute (for each acquired pixel independently) the likelihood values in all integer projector codes, that is the product of three functions evaluated in that point. This could be a limitation in scenarios where a faster execution is preferred over an accurate phase unwrapping. The estimation of a suitable value for σ (that is the only parameter required by the unwrapping algorithm) is a relevant task, even if we show that our approach is not particularly sensitive to the given prior on the phase variances. Experimental results show that the phase standard deviation directly depends on the sampling value and on the acquired scene; in fact, a good variance estimator can be easily recovered empirically or through a calibration step performed at the beginning of an acquisition session.

4

Adaptive Albedo Compensation for Accurate Phase-Shift Coding

Phase shift coding technique are considered to be the most adaptive with respect to the features of the objects to be captured. The theoretical invariance to signal strength and the absence of discontinuities in intensity make this approach an ideal candidate to deal with complex surfaces of unknown geometry, color and texture. However, in practical scenarios, unexpected artifacts could still result due to the characteristics of real cameras. This is the case, for instance, with high contrast areas resulting from abrupt changes in the albedo of the captured objects. In fact, the not negligible size of pixels and the presence of blur can produce a mix of signal integration from adjacent areas with different albedo. This, in turn, would result in a bias in the phase recovery and, consequentially, in an inaccurate 3D reconstruction of the surface. While this problem affects most structure light methods, little effort has been put in addressing it. With this Chapter we propose a model for the phase corruption and a theoretically sound correction step to be adopted to compensate the bias. The practical effectiveness of our approach is well demonstrated by a complete set of experimental evaluations.

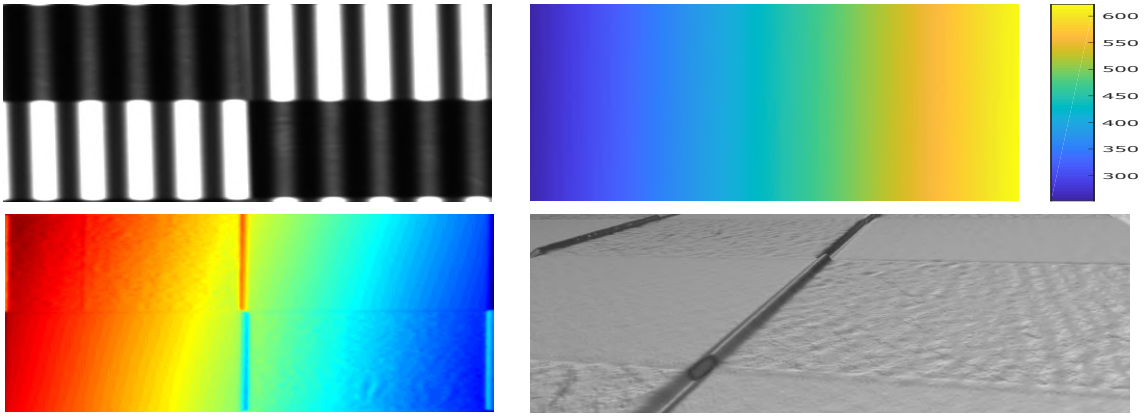


Figure 4.1: Effect of the phase offset induced by abrupt changes on the albedo of object texture. See the text for details.

4.1 Introduction

Phase shift coding works by projecting a sinusoidal pattern whose initial phase is progressively shifted between sequential frames. By convolving the observations over time [201], each image pixel $\boldsymbol{x}$ is assigned to its initial phase $\varphi(\boldsymbol{x})$. Given the periodic nature of the signal, $\varphi(\boldsymbol{x})$ is expected to repeat throughout the scene. This ambiguity problem is resolved by means of an additional phase unwrapping step [125] which reliably translate $\varphi(\boldsymbol{x})$ to a unique projector coordinate $\phi(\boldsymbol{x})$, corresponding to the (possibly fractional) column of the projector frustum (i.e. the plane of a virtual laser stripe).

One of the best features of phase shift is that, as long as the projected signal is a proper sine wave [227], the recovery of $\phi(\boldsymbol{x})$ is not affected by the albedo of the surface, since the convolution is inherently invariant with respect to scale. Unfortunately, this invariance holds only if the albedo is uniform within the integration surface of each pixel, which in turn depends both on pixel size and focus. It is well-known that if the camera pixel falls between two surface portion with very different reflectance (i.e. high contrast areas) the recovered codes will suffer from a displacement error due to the uneven signal integration.

This undesired phenomenon is well exemplified in Figure 4.1, where the case of the reconstruction of a planar checkerboard is shown. The fringe pattern is projected along the horizontal direction and thus the recovered codes, shown in the top-right subfigure, grow from left to right. The effect of the abrupt albedo change corresponding to the edge of the checks is not easy to spot in the figure. Indeed its just a fraction of projector pixel. Nevertheless, when such biased estimate is used to reconstruct the 3D surface, the triangulation step amplifies its impact. This can be observed in the bottom-left and bottom-right parts of the figure, where we show respectively the depth map and a shaded perspective view of the reconstructed surface. From this representations the effect of the bias is much more apparent and it can be observed that it results in an overestimation or underestimation of the code depending on the change being from dark to light colour or vice versa.

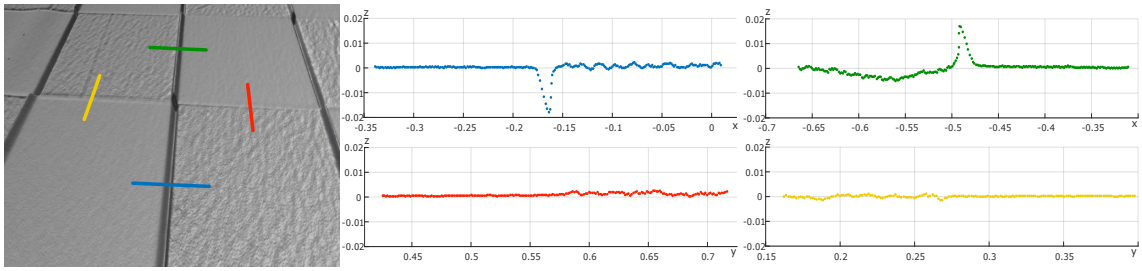


Figure 4.2: Details about the code recovery bias due to contrast changes along the signal direction (see the text for details).

In Figure 4.2 we analyse this behaviour with more detail by highlighting four separate slicing of the reconstructed mesh. From such slicing we can see that the passage from a white to a black check (blue stretch) results in a slight underestimation of the code, and thus to an excessive depth in the reconstructed surface. The opposite happens when the transition is from dark to light albedo (green stretch). This effect, however, can be noticed only when the albedo gradient is aligned with the fringe direction. In fact, in this cases, the only visible artifact is the slightly larger noise exhibited by the surfaces reconstructed over black checks. This is indeed expected and it is due just to the smaller range of the signal in dark areas.

While a moderate amount of effort have been devoted to the modulation of an accurate sine signal [190], its compensation with respect to distortion due to non-uniform albedo is a much less covered research topic.

To our knowledge, this topic has been recently addressed by a couple of research groups. Zhong and Burke [32] propose a very simple model for the phase error generation and correction. They consider high-to-low and low-to-high transitions in 2x2 pixels windows. Once such transitions are detected, the magnitude difference in albedo is estimated and corrected during the phase computation step. Differently, Xiaochang et. al [224] propose to address the problem by working directly on the recovered phases rather than on the image intensities. Their method first detects the high contrast areas by means of a Sobel operator and then suppresses the artifacts through linear interpolation of the surrounding phases, which are supposed to be recovered over areas of uniform albedo.

Novel Contribution

In this chapter we present a novel correction approach that is quite different with respect to the current literature. We propose to model the code bias produced by contrast changes as a convolution occurring between the surface albedo and the code values that should have been observed without errors. Then, a correction factor can be applied directly to the computed values by simply inverting the convolution formula. The resulting technique is very simple to implement, yet it exhibits a solid performance, as assessed through an in depth experimental session involving both synthetically generated data and real scans. The technique presented in this Chapter has been published in [164].

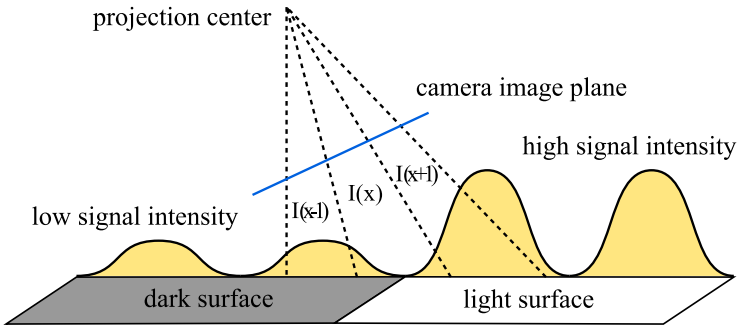


Figure 4.3: Principle underlying the integration error through high-contrast edges (the ratio between pixel size and fringe length is exaggerated for illustration purposes. In practice pixel size are a very small fraction of fringe length).

4.2 Albedo-Induced Phase Recovery Errors

Pixels in real-word camera do not perform punctual sampling of the observed objects. Rather they operate an integration over both the time of exposure and the finite surface area from which they gather reflected photons. The extent of the captured area is determined by two factors. The first one is the relation between optics geometry and pixel size on the sensor. The second factor is the blur due to the non pinhole behaviour of optics. In Figure 4.3 we illustrate the consequences of a non punctual integration over signal fringes. If the integration happens within regions of uniform albedo, its finite extent in space has little influence. This is mainly because, during the phase shift steps, all the phase displacements are presented within the integration area an the symmetric nature of the signal compensates any bias, for reasonable sizes of the window. Differently, when the albedo changes within the integration window this symmetry breaks and a bias is introduced. Since no information is available at sub-pixel level, we propose to model this effect with a discrete approximation computed over neighbour pixels:

$$\hat{\phi}(\boldsymbol{x}) = \frac{\sum_{\boldsymbol{h}\in\Omega} I(\boldsymbol{x}+\boldsymbol{h})\phi(\boldsymbol{x}+\boldsymbol{h})W(\boldsymbol{h})}{\sum_{\boldsymbol{h}\in\Omega} I(\boldsymbol{x}+\boldsymbol{h})W(\boldsymbol{h})} \tag{4.1}$$

where $\hat{\phi}(\boldsymbol{x})$ is the observed code, $\boldsymbol{x}$ is a 2D vector of pixel coordinates, $\boldsymbol{h} \in \Omega$ is a displacement vector, W is a weight function, $I(\boldsymbol{x})$ and $\phi(\boldsymbol{x})$ are respectively the albedo and the correct code that should be assigned to pixel $\boldsymbol{x}$.

In detail, W is a symmetric radial weight function and we are assuming that $\sum_{\boldsymbol{h}\in\Omega} W(\boldsymbol{h}) = 1$. $\Omega \subset \mathbb{Z}^2$ represents a symmetric convolution window which depends on the camera characteristics. As an example, if if the integration happens over a 3×3 window, Ω is:

$$\Omega = \left\{ \begin{array}{ccc} (-1,-1), & (0,-1), & (1,-1), \\ (-1,0), & (0,0), & (1,0), \\ (-1,1), & (0,1), & (1,1) \end{array} \right\} \tag{4.2}$$

We should clearly point that our model makes an important assumption. Basically, we are deeming the convolution of the phase and of the signal to be equivalent. This is not true in general, still it could be a good approximation for integration intervals small enough to consider both linear.

While these assumption seems sound approximations at first glance, they must at least be checked experimentally to verify the practical effectiveness of the proposed approach. In order to perform such evaluation we must introduce a proper compensation strategy.

4.3 Adaptive Albedo Compensation

We formulate our compensation strategy by considering the following approximation of the real albedo I and phase ϕ in a neighbourhood of a point $\boldsymbol{x}$ given by the first order Taylor expansion:

$$I(\boldsymbol{x}+\boldsymbol{h}) = \hat{I}(\boldsymbol{x}) + \nabla\hat{I}(\boldsymbol{x})^T\boldsymbol{h} \tag{4.3}$$

$$\phi(\boldsymbol{x}+\boldsymbol{h}) = \phi(\boldsymbol{x}) + \nabla\hat{\phi}(\boldsymbol{x})^T\boldsymbol{h} \tag{4.4}$$

In practical terms, we approximate the real albedo as a linear combination of the observed albedo and its gradient. Similarly, we linearly approximate the phase with the gradient of the observed phase. Both are motivated by the fact that we expect the convolution modelled in eq.4.1 to have a major impact on the phase $\phi(\boldsymbol{x})$ more than the phase gradient or the albedo. The former because it is a differential quantity affected by a simmetric integration window. The latter because the albedo can be obtained without the projector and thus do not suffer from the integration of different fringes.

Using these approximations, eq. 4.1 can be rewritten as:

$$\hat{\phi}(\boldsymbol{x}) = \frac{\sum\limits_{\boldsymbol{h}\in\Omega}\big(\hat{I}(\boldsymbol{x}) + \nabla\hat{I}(\boldsymbol{x})^T\boldsymbol{h}\big)\big(\phi(\boldsymbol{x}) + \nabla\hat{\phi}(\boldsymbol{x})^T\boldsymbol{h}\big)W(\boldsymbol{h})}{\sum\limits_{\boldsymbol{h}\in\Omega}\hat{I}(\boldsymbol{x})W(\boldsymbol{h}) + \sum\limits_{\boldsymbol{h}\in\Omega}\nabla\hat{I}(\boldsymbol{x})^T\boldsymbol{h}W(\boldsymbol{h})} \tag{4.5}$$

From this formulation, with a few algebraic passages, we can make $\phi(\boldsymbol{x})$ explicit:

$$\begin{gathered}\hat{\phi}(\boldsymbol{x})\sum_{\boldsymbol{h}\in\Omega}\hat{I}(\boldsymbol{x})W(\boldsymbol{h}) + \hat{\phi}(\boldsymbol{x})\sum_{\boldsymbol{h}\in\Omega}\nabla\hat{I}(\boldsymbol{x})^T\boldsymbol{h}W(\boldsymbol{h}) = \\ \sum_{\boldsymbol{h}\in\Omega}\hat{I}(\boldsymbol{x})\phi(\boldsymbol{x})W(\boldsymbol{h}) + \sum_{\boldsymbol{h}\in\Omega}\hat{I}(\boldsymbol{x})\nabla\hat{\phi}(\boldsymbol{x})^T\boldsymbol{h}W(\boldsymbol{h}) + \\ \sum_{\boldsymbol{h}\in\Omega}\nabla\hat{I}(\boldsymbol{x})^T\boldsymbol{h}\phi(\boldsymbol{x})W(\boldsymbol{h}) + \sum_{\boldsymbol{h}\in\Omega}\boldsymbol{h}^T\nabla\hat{I}(\boldsymbol{x})\nabla\hat{\phi}(\boldsymbol{x})^T\boldsymbol{h}W(\boldsymbol{h})\end{gathered}$$

$$\phi(\boldsymbol{x})\left(\sum_{\boldsymbol{h}\in\Omega}\hat{I}(\boldsymbol{x})W(\boldsymbol{h})+\sum_{\boldsymbol{h}\in\Omega}\nabla\hat{I}(\boldsymbol{x})^T\boldsymbol{h}W(\boldsymbol{h})\right)=$$
$$\hat{\phi}(\boldsymbol{x})\left(\sum_{\boldsymbol{h}\in\Omega}\hat{I}(\boldsymbol{x})W(\boldsymbol{h})+\sum_{\boldsymbol{h}\in\Omega}\nabla\hat{I}(\boldsymbol{x})^T\boldsymbol{h}W(\boldsymbol{h})\right)-$$
$$\sum_{\boldsymbol{h}\in\Omega}\hat{I}(\boldsymbol{x})\nabla\hat{\phi}(\boldsymbol{x})^T\boldsymbol{h}W(\boldsymbol{h})-\sum_{\boldsymbol{h}\in\Omega}\boldsymbol{h}^T\nabla\hat{I}(\boldsymbol{x})\nabla\hat{\phi}(\boldsymbol{x})^T\boldsymbol{h}W(\boldsymbol{h})$$

$$\phi(\boldsymbol{x})=\hat{\phi}(\boldsymbol{x})-\frac{\sum_{\boldsymbol{h}\in\Omega}\hat{I}(\boldsymbol{x})\nabla\hat{\phi}(\boldsymbol{x})^T\boldsymbol{h}W(\boldsymbol{h})+\sum_{\boldsymbol{h}\in\Omega}\boldsymbol{h}^T\nabla\hat{I}(\boldsymbol{x})\nabla\hat{\phi}(\boldsymbol{x})^T\boldsymbol{h}W(\boldsymbol{h})}{\sum_{\boldsymbol{h}\in\Omega}\hat{I}(\boldsymbol{x})W(\boldsymbol{h})+\sum_{\boldsymbol{h}\in\Omega}\nabla\hat{I}(\boldsymbol{x})^T\boldsymbol{h}W(\boldsymbol{h})}$$

$$\phi(\boldsymbol{x})=\hat{\phi}(\boldsymbol{x})-\frac{\hat{I}(\boldsymbol{x})\nabla\hat{\phi}(\boldsymbol{x})^T\sum_{\boldsymbol{h}\in\Omega}\boldsymbol{h}W(\boldsymbol{h})+\nabla\hat{I}(\boldsymbol{x})^T\nabla\hat{\phi}(\boldsymbol{x})\sum_{\boldsymbol{h}\in\Omega}\boldsymbol{h}^T\boldsymbol{h}W(\boldsymbol{h})}{\hat{I}(\boldsymbol{x})\sum_{\boldsymbol{h}\in\Omega}W(\boldsymbol{h})+\nabla\hat{I}(\boldsymbol{x})^T\sum_{\boldsymbol{h}\in\Omega}\boldsymbol{h}W(\boldsymbol{h})}$$

Since both the window Ω and the weight function W are symmetric we have that $\sum_{\boldsymbol{h}\in\Omega}\boldsymbol{h}W(\boldsymbol{h})=\boldsymbol{0}$.

Moreover, we have that $\sum_{\boldsymbol{h}\in\Omega}W(\boldsymbol{h})=1$, thus:

$$\phi(\boldsymbol{x})=\hat{\phi}(\boldsymbol{x})-\frac{\nabla\hat{I}(\boldsymbol{x})^T\nabla\hat{\phi}(\boldsymbol{x})\sum_{\boldsymbol{h}\in\Omega}\boldsymbol{h}^T\boldsymbol{h}W(\boldsymbol{h})}{\hat{I}(\boldsymbol{x})} \tag{4.6}$$

$$\phi(\boldsymbol{x})=\hat{\phi}(\boldsymbol{x})-k\frac{\nabla\hat{I}(\boldsymbol{x})^T\nabla\hat{\phi}(\boldsymbol{x})}{\hat{I}(\boldsymbol{x})} \tag{4.7}$$

Where $k=\sum_{\boldsymbol{h}\in\Omega}\boldsymbol{h}^T\boldsymbol{h}W(\boldsymbol{h})$, is a parameter depending on Ω and W which, in turn, depend on the overall sum of physical characteristics of the imaging process, including the pixel size and the optics geometry.

Before proceeding with the experimental evaluation of the proposed compensation (4.7), we would like to discuss two of its aspects in detail.

The first observation is about the parameter k: since this value is bound to the physical aspects of the capturing process, it should be calibrated for any given structured light scanner. In the following section we will suggest a practical method to perform this task. It should also be noted that the assumption made about the energy conservation in the weight function (i.e. $\sum_{\boldsymbol{h}\in\Omega}W(\boldsymbol{h})=1$) is not really needed since this value could be embedded in the multiplicative constant k as a result of calibration.

Second, if we go back to qualitative example of the phenomenon, shown in Figure 4.2, we can verify the consistency of (4.7) with respect to the observed artifacts. In particular, we notice that the compensation magnitude is monotonically increasing with respect to

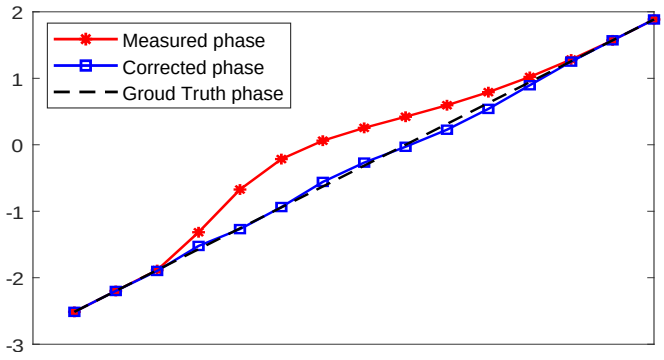


Figure 4.4: Example of code bias correction before and after calibrating k.

the dot product between the albedo gradient and the code gradient. This makes sense, as the artifacts in Figure 4.2 are strong when the abrupt change in texture intensity is aligned with the fringe orientation, while negligible bias can be observed when they are orthogonal. Moreover, also the sign of the bias is modeled correctly, since (4.7) is providing a negative compensation when the albedo gradient grows with the fringe direction and vice versa.

4.4 Calibration

Calibrating the right value of k for a given (real or synthetic) scanning system is in general quite straightforward. In practice, the only required condition is to knwon in advance the expected phases $\phi(x)$ for a stretch of image pixels and then to find the value of k minimizing the RMS error with respect to the observed phases $\hat{\phi}(x)$.

We solve the ground-truth problem by using as a calibration object a perfectly planar checkerboard with bright and dark checks. We select two pixels on a horizontal line (u_1, v) and (u_2, v), located respectively in two dark and bright adjacent checks. According to our model, we assume that for the pixels inside areas with uniform albedo there is no bias, thus $\phi(u_1, v) = \hat{\phi}(u_1, v)$ and $\phi(u_2, v) = \hat{\phi}(u_2, v)$. Since the calibration object is planar and the projector code is linear, the expected ground-truth can be obtained by linear interpolation:

$$\phi(u, v) = \hat{\phi}(u_1, v) + (u - u_1)\frac{\hat{\phi}(u_2, v) - \hat{\phi}(u_1, v)}{u_2 - u_1} \tag{4.8}$$

Using eq. (4.8) we can define the following set of equations:

$$\phi(i, v) - \hat{\phi}(i, v) = k\frac{\nabla\hat{I}(i, v)^T \nabla\hat{\phi}(i, v)}{\hat{I}(i, v)} \qquad u_1 \leq i \leq u_2 \tag{4.9}$$

By solving system (5.2) using least squares, a good estimate for k can be easily recovered. Note that, since the estimation of $\phi(u_1, v)$ and $\phi(u_2, v)$ is still subject to noise,

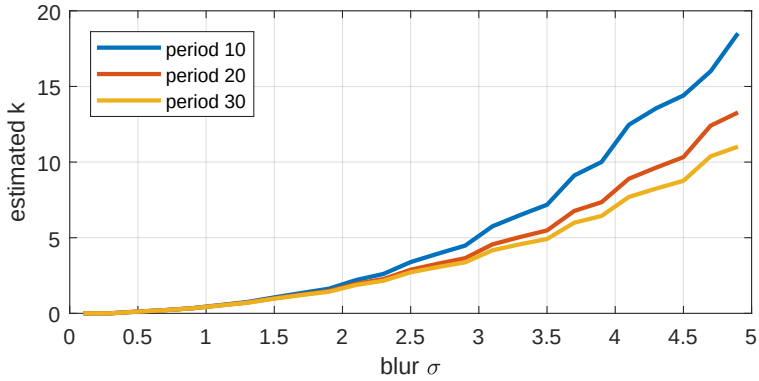


Figure 4.5: Behaviour of the parameter k for different levels of blur and different base phase periods.

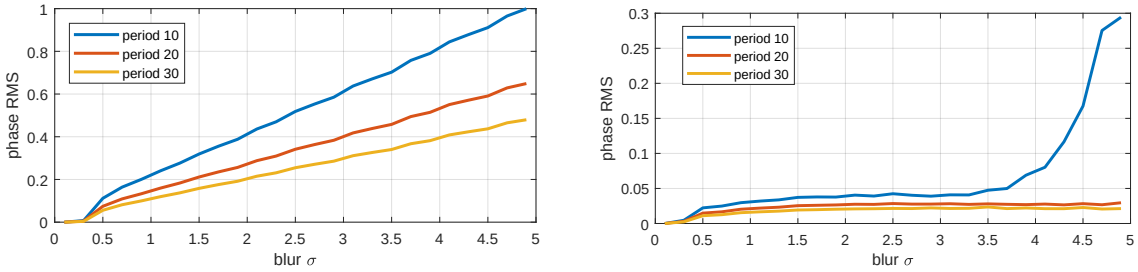


Figure 4.6: Actual impact of the bias without correction (left) and after the compensation has been applied (right) with increasing blur levels on a synthetic setup.

we suggest to build the system to be optimized by combining different pixel stretches in different parts of the image. Moreover, we assume that the observed image albedo is always greater than zero. This is not a limiting factor since zero albedo means no projector signal and hence no projetor code to correct.

In Figure 4.4 we show and example of the calibration process, including the estimated ground-truth (black line), the observed codes before correction (red line) and the resulting corrected codes.

4.5 Experimental Results

The validation of the proposed albedo compensation method has been performed on both synthetically generated data and real scans. For all the experiments we adopted the phase recovery and unwrapping method described in [125]. The calibration of k has always been performed using a checkerboard target and 20 stretches of 30 pixels each to create the system (5.2) with 600 equations. The compensation has been performed by applying equation (4.7) to all the pixels of the recovered image, in fact one of the advantages of our approach with respect to others is that it does not require any step to isolate high-contrast areas to be corrected.

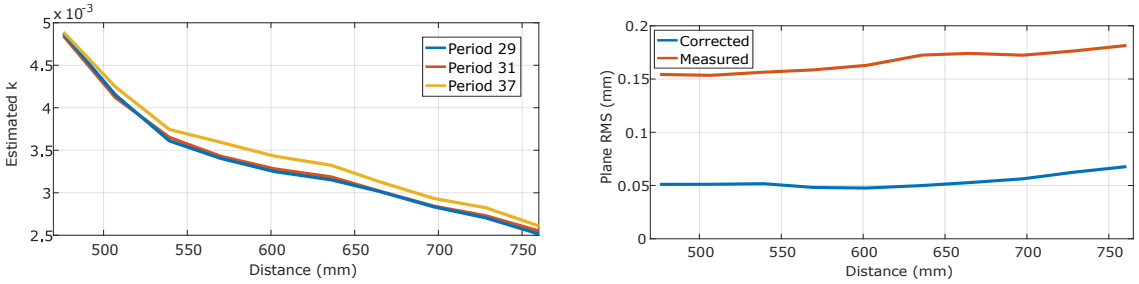


Figure 4.7: Experiments performed with a real scan system on a planar checkerboard. On the left the estimated value of k and on the right the RMS of the 3d reconstruction. Both experiments have been performed at different blur levels achieved by decreasing the distance of the acquired surface.

4.5.1 Synthetical Evaluation

The goal of this first batch of experiments is to demonstrate some properties of the method in a controlled simulated environment. Specifically, we created a test scene made up of a combination of 3D objects textured with high contrast checkerboard patterns. Note that there is no loss of generality introduced by choosing a checkerboard pattern, since the interaction between the texture and the object surfaces will still result in a wide range of different combinations of viewing conditions. We simulated the scanning process by projecting the sine wave patterns using ray tracing. The resolution of the virtual projector and of the virtual camera, and their relative position, have been chosen so that the size of projector pixels is about the same of camera pixels (which is the typical condition with real scanners). To simulate the effects of both non punctual integration and defocus we applied an artificial blur by means of a convolution performed with a Gaussian kernel with varying standard deviation σ.

The first experiment that we performed is designed to study the behaviour of parameter k with respect to changes in blur and in the signal. Specifically we varied the blur strength σ in a range from 0 to 5, and we adopted 3 different base period length, respectively of 10, 20 and 30 pixels of the virtual projector. The resulting plot is shown in Figure 4.5. As a general observation, k is growing with the amount of blur, which is of course expected. With medium levels of blur k seems to be independent from the length of the period of the scanning signal. This independence actually breaks for higher levels of defocus. It must be noticed that this happens when the standard deviation of the blur become comparable with the period length. In fact the short periods require stronger correction. This could be partially justified by the observation that, under those conditions, the linearity assumptions we made break. However, in practice, blurs above 2 or 3 pixels are symptoms of severe image defocusing and should not happen with a properly designed setup.

The second experiment is about assessing the compensation ability of the method and its breaking point. To this end, we computed the RMS error on phase recovery again for different blur levels, adopting for each one of these levels the optimal calibrated value of

k. Since most pixels are not edges, we computed the RMS only on the pixels with high response to a Sobel operator. The results are shown in Figure 4.6. In the left part of the figure we plot the error in phase recovery without any correction. As expected it grows with the blur level and the short periods are affected the most. In the right part we show how the phase error get compensated by our method. The compensation is very effective even with the shortest period, with close to negligible phase errors (notice that the two plots have a different scale). By using the most suitable value for k the correction works up to several pixels of blur radius, with the exception of the shortest period which breaks earlier than the other two.

4.5.2 Evaluation on Real Scans

Since we are proposing a method to be adopted in actual products, our evaluation would not be complete without studying how well the compensation works with real-world scanners.

This second batch of experiments has been performed using a camera-projector scanner featuring a 3Mpixels CCD monochromatic camera and a 1080p digital projector. The camera optics have been configurated to match the projector frustum, thus providing similar pixel sizes between them. The baseline between camera and projector is about 10cm. The camera intrinsics have been calibrated using [24] and refined with [19] and the extrinsics using markers [18, 22].

Since it is not so easy to obtain a ground-thruth from real-world objects we performed our experiments by scanning a planar checkerboard and, subsequently, by fitting a plane on the points triangulated by the scanner. The effectiveness of the compensation is measured by means of RMS error of triangulated points with respect to the fitted plane. This is of course an indirect measurement, but, at the end of the day, it is almost directly related to phase recovery errors and it is a critical parameter for evaluating the accuracy of scanners. Since most of the points are not on edges (and thus unaffected by contrast changes) we computed the RMS error on the 10% of points corresponding to the highest albedo gradient magnitude. Finally, given that the camera optics are fixed, in order to simulate the blur due to different setup we exaggerated the iris aperture (to obtain a shallow depth of field) and changed the distance of the target.

The results of this experiment are shown in Figure 4.7. Regarding the relation between k and blur, we can observe that it still changes monotonically with it. The curve trend is a bit different from the one obtained with synthetically experiments, but this can be justified by the complexity of the actual focusing process in real lens. The dependence on period length seems to be negligible (albeit here our choice of period length was constrained by the scanning software). The performance of compensation is plotted on the right part of Figure 4.7. It can be observed a noticeable increase in accuracy, regardless of the distance, and thus of the blur level.

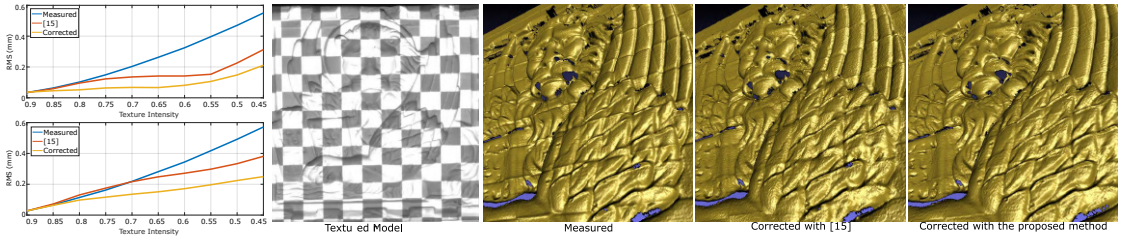


Figure 4.8: Comparisons of the results achieved by our method with those obtained using [224]. The two plots on the left show the RMS of two reconstructed surfaces with increasing texture intensities. The qualitative examples on the right show the applied mask and three reconstructions of the same surface, respectively (from left to right) with no correction, compensated using [224] and corrected by our method.

4.5.3 Comparisons and Qualitative Examples

Our last experiment is about rating the effectiveness of our method when dealing with complex physical objects and to compare it with an alternative approach. As for the other experiments one of the main concerns is how to obtain a reasonable ground-truth. To this end, we modified the projector software to multiply the original sine wave patterns with a white and gray checkerboard template. This way, the signal is superimposed on a high-contrast mask, resulting in an effective simulation of a varying albedo on the observed shape. Indeed, from the point of view of the camera, there is no way to distinguish changes in albedo due to a natural texture and to our multiplicative pre-processing. Finally, we ruled out any influence of natural textures by using only objects made of white plaster with various shapes and sizes.

In our tests, we first performed a full reconstruction without applying any mask, with the aim of obtaining the best possible estimate of the real surface. Then we measured the RMS error on points with albedo gradient magnitude greater than 0.5. The idea is to increase the contrast of the object texture under a controller environment.

As discussed in the introduction, albedo compensation is a surprisingly little covered topic in literature. Among the few methods addressing this problem we decided to compare with the one proposed by Xu, Wang and Liu [224]. This choice is mainly due to the fact that, like our approach, it works directly in the code space, making the comparison fair.

The results of this test are shown in Figure 4.8 by means of both an error plot conceptually similar to the one presented in Figure 4.7, and of a few qualitative examples. The texture intensity express the modulation applied to the signal inside the gray blocks, ranging from 0.9 to 0.45.

The upper plot shows the results obtained on a plaster cube, which is a simpler object made of flat surfaces. The lower one covers the more complex winged lion figurine for which we also show the renderings of the reconstructed surface obtained for each one of the tested approaches.

4.6 Conclusions

With this chapter we described a simple yet very effective method to compensate the bias suffered by phase shift techniques when dealing with surface of non-uniform albedo. The method has several advantages: it is based on a sound modelling of the code corruption phenomenon that has been verified through an in-depth experimental analysis. Differently from other methods, it does not require a preliminary time-consuming step to detect high contrast areas in the captured scene, so it can be applied blindly to the whole image during acquisition. Moreover, it depends on a single parameter k, which has been shown to be easy to calibrate for a given scanning system and can also be performed during the structured-light system calibration. Finally, the proposed compensation is very easy to implement and its computation complexity is linear with the number of pixels to be processed, making the correction immediate.

5

Neighborhood-Based Recovery of Phase Unwrapping Faults

Several unwrapping methods exploit additional signals of mutually prime periods, in order to observe a distinct combination of phases for each spatial point. Unfortunately, for such combination to be properly recognized, a very high accuracy in phase recovery must be attained for each signal. In fact, even modest errors could lead to unwrapping faults, making the overall approach much less resilient to noise than plain phase shift. With this Chapter we introduce a feasible and effective fault recovery method that can be directly applied to multi-period phase shift. The combined pipeline offers an optimal accuracy and coverage even with high noise conditions, overcoming the setbacks of the original method. The performance of such pipeline is established by means of an in depth set of experimental evaluations and comparison, both with real and synthetically generated data.

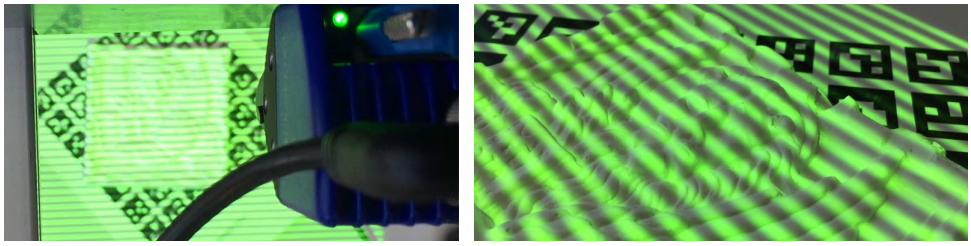


Figure 5.1: Capturing of a 3D surface by means of phase shift coding.

5.1 Introduction

In structured-light 3D reconstruction the signal is observed by one or more cameras, can then be used to assign a distinctive code to each material point in the scene. This code enables the labelling of corresponding points between different views and thus perform triangulation. In this Chapter we are not dealing with the triangulation step, which is itself a wide research topic [23, 79]. Instead, we are interested in the coding signal recovery.

Regardless of the wide choice in coding strategies, most commercial solutions still adopt the old-fashioned phase shift method [197]. The underlying idea is indeed quite simple: the projected frames are sine wave intensity patterns that are periodic (usually) along one direction (see Figure 5.1). A total of n patterns is projected over time, each one being shifted by an offset of $\frac{2\Pi}{n}$ periods. After all the patterns have been captured by a camera, each image pixel u, v is labelled with a base phase value $\varphi(u, v) \in [0, 1)$ recovered by means of correlation (for details see for instance [201]). Since the signal is periodic in space, the same value of φ appears several times and an additional unwrapping step is needed. Most phase unwrapping approaches resort to the projection of an additional pattern sequence (often Gray codes), exhibiting lower accuracy, but which is not affected by ambiguity. However, such methods have the drawback that not all the projected patterns effectively contribute to the accuracy of φ. To allow a better exploitation of captured signals, some authors proposed multi-period approaches which use phase shift also for disambiguation (for instance [125]). While promising, these latter techniques are seldom adopted in actual devices. In fact, as we will show in the experimental section, they are quite sensitive to noise as the unwrapping step requires an extremely high accuracy in phase recovery.

Novel Contribution

In this Chapter we introduce a practical method to address such noise sensitivity, proposing a set of code recovery strategies exploiting spatial proximity of computed codes. In particular, for each code we estimate an error measure and propose three voting schemas that use the values of the n independently observed phases in different ways: vector fringe consensus, independent fringe consensus and complete fringe-set check. The resulting pipeline enables the practical adoption of multi-period phase shift with a minimal effort and guarantees maximal accuracy at any noise level. This work has been presented in [163].

5.2 Multi-period Phase Shift

Multi-period phase shift, proposed by Lilienblum and Michaelis [125], combines phase recovery and unwrapping by mean of n independent phase shift sequences of different (and possibly coprime) period lengths $\lambda_1, \lambda_2, ..., \lambda_n$, resulting in a vector of recovered phases $\boldsymbol{\varphi} = (\varphi_1\ \varphi_2\ ...\ \varphi_n)^T$. The main idea, depicted in Figure 5.2, is that, given an unique projector coordinate ξ, the phases vector $\boldsymbol{\varphi}$ will be also unique. In fact, the same combination of phases will not appear again until $\xi'' = \xi' + k\ \xi_{max}$, where $\xi_{max} = LCM(\lambda_1, \lambda_2, ..., \lambda_n)$ (that is the Least Common Multiple of all period lengths).

More in detail, we can define functions $\varphi_i(\xi)$ and $\eta_i(\xi)$ respectively as phase value and fringe number expected to be recovered from projector coordinate ξ by the i^{th} pattern sequence:

$$\eta_i(\xi) = \left\lfloor \frac{\xi}{\lambda_i} \right\rfloor \qquad \varphi_i(\xi) = \frac{\xi}{\lambda_i} - \left\lfloor \frac{\xi}{\lambda_i} \right\rfloor \tag{5.1}$$

From these definitions we can infer the following system of equations

$$\xi = (\eta_i(\xi) + \varphi_i(\xi))\lambda_i \qquad \forall i = 1, ..., n \tag{5.2}$$

After the patterns have been projected and the phases $\varphi_i(\xi)$ recovered, the only unknowns left are the components of the fringe numbers $\boldsymbol{\eta} = (\eta_1(\xi)\ \eta_2(\xi)\ ...\ \eta_n(\xi))^T$ and the projector coordinate ξ itself. Imposing the condition $\xi < \xi_{max}$, the system has an unique solution, thus allowing to recover the sought projector coordinate ξ.

Such unique solution can be easily found by considering the differences between phases. Specifically, fixing two pattern sequences j and k, the right sides of the equations in system (2) can be equated, obtaining:

$$\begin{aligned} (\eta_j(\xi) + \varphi_j(\xi))\lambda_j &= (\eta_k(\xi) + \varphi_k(\xi))\lambda_k \\ \lambda_j\eta_j(\xi) - \lambda_k\eta_k(\xi) &= \lambda_k\varphi_k(\xi) - \lambda_j\varphi_j(\xi) \end{aligned} \tag{5.3}$$

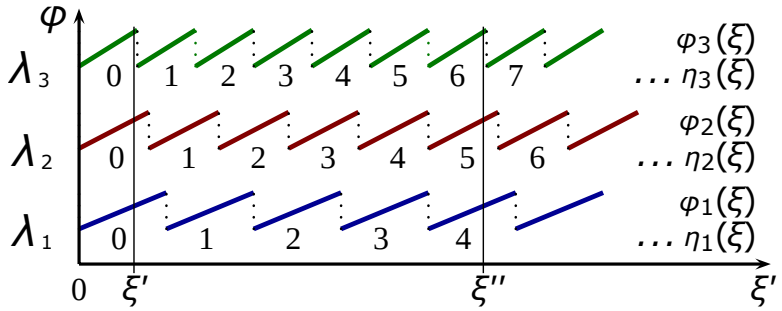


Figure 5.2: Disambiguation principle in multi-period phase shift.

In particular, from the right part of such relation, we can define a phase difference vector based on the offset from pattern sequence 1:

$$\Phi(\xi) = \boldsymbol{a} = (a_1, ..., a_n); \qquad a_i = \lambda_1\varphi_1(\xi) - \lambda_i\varphi_i(\xi) \tag{5.4}$$

Since we already limited the code values, elements a_i are not ambiguous and a map H can be defined in the following way:

$$H(\boldsymbol{a}) = \begin{cases} (h_1, h_2, ..., h_n) : & \forall i\{a_i = \lambda_i h_i - \lambda_1 h_1\} \\ \text{undefined} & \text{otherwise} \end{cases} \tag{5.5}$$

allowing for an easy conversion between phase difference vectors and fringe numbers:

$$H(\Phi(\xi)) = (\eta_1(\xi), \eta_2(\xi), ..., \eta_n(\xi)) \tag{5.6}$$

The range of values in H is a finite set, which depends only on period lengths $\lambda_1, ..., \lambda_n$. Therefore, it is possible to pre compute all the values for H and store them in a LUT (Look Up Table). This data structure allows to compute fringe numbers in a fast way given the sequence of phase observations $\boldsymbol{\varphi}$.

Of course, in the real-world $\boldsymbol{\varphi}$ will be affected by errors due to several error sources. So it is not obvious that $\Phi(\xi)$ will be accurate enough for H to map it to the correct fringe vector. As a matter of fact, as we show in Sec. 5.4, even modest levels of noise lead to difference vectors that are not in the domain of eq. 5.5. This, in turn, leads to LUT access faults and, in general, to the inability of recovering the correct value for ξ (see [125] for details). It should be noted that, once the LUT access fails, the difference vector offers no hint about where to search for a fringe vector that would justify the obtained $\boldsymbol{\varphi}$. To this end, a brute force solution would be to check for all the feasible combinations of fringe values and select the one that minimizes the radius of estimated ξ. Unfortunately this would lead to a huge number of tests. In the following section we propose three strategies to reduce the number of fringe vectors to be checked.

5.3 Neighborhood-based Fault Recovery

The following strategies are actually rather naive. However, the main contribution of this Chapter is about their evaluation in terms of effectiveness and feasibility with respect to computational efficiency. Our goal, indeed, is to show that multi-period phase shift can be easily adopted in real-world scenarios without drawbacks.

The shared mechanics of all the following strategies involves the selection of a reduced set of fringe vectors $\boldsymbol{\eta}$ to be tested, that we call S_η, and a rule to extract among its elements the optimal fringe vector $\boldsymbol{\eta}^*(S_\eta)$.

For a given vector of observed phases $\boldsymbol{\varphi}$ and a fringe vector $\boldsymbol{\eta} \in S_\eta$, we define an estimation for both the projector coordinate ξ and its error ϵ. The obvious estimator for ξ is the average of the independent values obtained from each period sequence.

According to eq. 5.2, it can be computed as:

$$\xi(\boldsymbol{\varphi}, \boldsymbol{\eta}) = \frac{\sum_{i=1}^{n}(\eta_i(\xi) + \varphi_i(\xi))\lambda_i}{n} \tag{5.7}$$

One of the main advantages of eq. 5.7 is that, assuming that no outliers are present, each single pattern equally contributes to a better assessment of ξ.

A reasonable estimator for the error of $\xi(\boldsymbol{\varphi}, \boldsymbol{\eta})$ could be of course the standard deviation exhibited by the averaged values. However, since the number of signals involved in multi-period phase shift is usually small, we opted for a more conservative choice. We assess the error commited as the radius of the independent estimates for ξ, that is:

$$\epsilon(\boldsymbol{\varphi}, \boldsymbol{\eta}) = \max_{1 \leq i,j \leq n} |(\eta_i(\xi) + \varphi_i(\xi))\lambda_i - (\eta_j(\xi) + \varphi_j(\xi))\lambda_j| \tag{5.8}$$

The optimal fringe vector can thus be found as:

$$\boldsymbol{\eta}^*(S_\eta) = \arg\min_{\boldsymbol{\eta} \in S_\eta} \epsilon(\boldsymbol{\varphi}, \boldsymbol{\eta}) \tag{5.9}$$

Consequently, the optimal estimate for ξ within the set of candidates S_η is $\xi(\boldsymbol{\varphi}, \boldsymbol{\eta}^*(S_\eta))$.

Additionally, we also define a general criterion to be adopted in order to retain or discard such estimates. This is needed because if no valid candidate for $\boldsymbol{\eta}$ exists in S_η, then the obtained value for ξ could be totally random. Since this is just an outlier detection measure, the criterion can be very coarse. We adopt a threshold t_ϵ over $\epsilon(\boldsymbol{\varphi}, \boldsymbol{\eta})$. To this end $\xi(\boldsymbol{\varphi}, \boldsymbol{\eta}^*(S_\eta))$ is considered recovered only if $\epsilon(\boldsymbol{\varphi}, \boldsymbol{\eta}^*(S_\eta)) < t_\epsilon$. Otherwise, the point is deemed to be non-recoverable. We propose to set $t_\epsilon = 0.5\,(\sum_{i=1}^{n} \lambda_i)/n$. This is a quite coarse estimate, since the typical error is much lower than half the average period length. Still in all experiments this threshold never resulted in an outlier accepted as a valid ξ.

5.3.1 Vector Fringe Consesus

All the strategies proposed to build S_η for a pixel (u, v) works by looking at the neighborhood of the pixel itself. We define $N_k(u, v)$ as the set of the k nearest pixels to (u,v) that are correctly mapped using map H. Moreover, we define $N_k(u, v, \boldsymbol{\eta})$ as the set of pixels in $N_k(u, v)$ that are mapped by H to $\boldsymbol{\eta}$ and we say that $\boldsymbol{\eta} \in N_k(u, v)$ if $|N_k(u, v, \boldsymbol{\eta})| > 0$.

The Vector Fringe Consensus strategy (VFC) defines the set of fringe vectors to be checked as:

$$S_{vfc}(u, v) = \{\boldsymbol{\eta} \,|\, |N_k(u, v, \boldsymbol{\eta})| \geq |N_k(u, v, \boldsymbol{\eta'})|\,, \forall \boldsymbol{\eta'}\} \tag{5.10}$$

This means that all the most frequent vectors in the neighborhood of (u, v) are checked.

5.3.2 Independent Fringe Consesus

We define $N_k(u, v, \eta, i)$ as the set of pixels in $N_k(u, v, \boldsymbol{\eta})$ such that $\eta_i = \eta$ for any $\boldsymbol{\eta}$, and the set of best candidates for a given fringe i as:

$$S^i_{ifc}(u,v) = \{\eta \,|\, |N_k(u,v,\eta,i)| \geq |N_k(u,v,\eta',i)|\,, \forall \eta'\} \tag{5.11}$$

This means that each set $S^i_\eta(u,v)$ contains the most frequent fringe numbers for period i within the set $N_k(u,v)$.

Using these sets, the Independent Fringe Consensus strategy (IFC) defines the set of fringe vectors to be checked as:

$$S_{ifc}(u,v) = \prod_{i=1}^{n} S^i_{ifc}(u,v) \tag{5.12}$$

That is the Cartesian product of all the sets $S^i_{ifc}(u,v)$.

It should be noted that the set $S_{ifc}(u,v)$ is not necessarily a super set of $S_{vfc}(u,v)$ as there is no guarantee that the most frequent combinations as whole fringe vectors $\boldsymbol{\eta}$ are composed of the most frequent independent components. Indeed, the rationale of IFC is to decouple single coordinate of $\boldsymbol{\eta}$ in order to deal with corner cases including fringe boundaries, where only one or two fringe numbers actually changes.

5.3.3 Complete Fringe-set Check

The third strategy is the most exhaustive, since it checks all the possible combinations of fringe numbers that appear in the neighborhood of the pixel. To this end, we first define the set of single fringe coordinates as:

$$S^i_{cfc}(u,v) = \{\eta \,|\, |N_k(u,v,\eta,i)| > 0\} \tag{5.13}$$

In a similar manner to IFC, also the complete Fringe-set Check strategy (CFC) defines a set of fringe vectors to be checked as a Cartesian product:

$$S_{cfc}(u,v) = \prod_{i=1}^{n} S^i_{cfc}(u,v) \tag{5.14}$$

This time, however, for a fringe number to be included it has just to be present in at least one neighbor. This is a very relaxed constraint and it is not clear if such allowance would result in an unfeasible number of fringe vectors to check. As we will show in the experimental section, this is not the case and the number of actual fringe combination to validate is in practice quite modest.

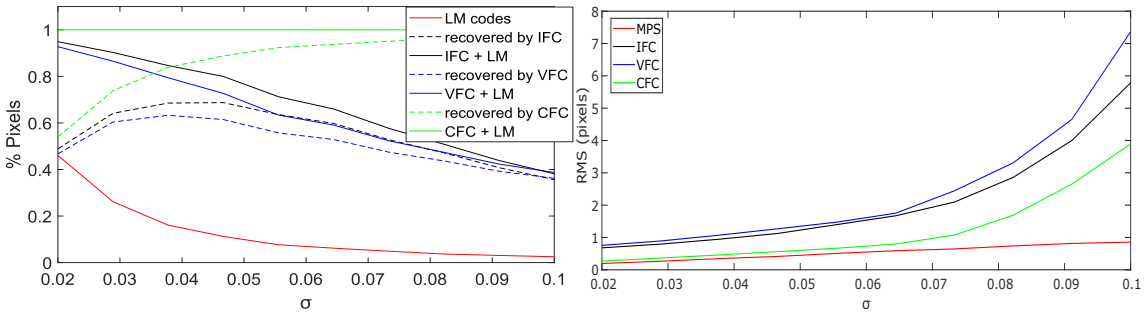


Figure 5.3: Comparison of the recovery rate and resulting RMS error achieved by the proposed methods.

5.4 Experimental Evaluation

The proposed approach is introduced in order to solve a practical problem with multi-period phase shift. To this end, it is paramount to assess its effectiveness with an in depth experimental evaluation.

In this section we perform such analysis with a set of different goals in mind. First of all, we want to show the ability of our method in recovering unwrapping faults both in terms of percentage of recovered points and of their accuracy. Then, we demonstrate that such ability is not critically affected by the number of candidates inspected, and thus the method can be applied without a significant performance loss. Finally a proper comparison with a standard non multi-period approach is reported. This last test highlights the improvement in accuracy granted by the redundant information offered by different (and independent) signals.

5.4.1 Fault Recovery with Noisy Signal

For an useful evaluation of the effectiveness of our method as a recovery tool, it is very important to know exactly the expected fringe numbers η_i for each given image pixel. This is hard to obtain accurately with real-world scans since it would require to perfectly know the geometry of the observed object, of the projector frustum and the relative pose of camera and projector.

On the other hand, real-world observation is not critical to the relevance of this evaluation. In fact, the error committed with the reconstruction of each phase φ_i is indeed the only factor affecting the unwrapping step of multi-period phase shift. Such error can originate by various sources, however, at the end of the day, the only significant element is its magnitude. If we model the error as zero-mean Gaussian additive noise, such magnitude can be expressed as a standard deviation σ and we can evaluate the resilience to noise by simulating perturbed phase observations:

$$\tilde{\varphi}_i = \frac{\xi}{\lambda_i} - \left\lfloor \frac{\xi}{\lambda_i} \right\rfloor + N(0, \sigma) \qquad i \in 1..n \tag{5.15}$$

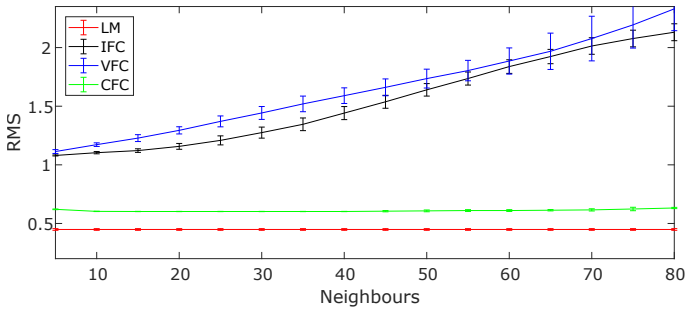


Figure 5.4: Analysis of the accuracy obtained with different values of parameter k (i.e. amount of tested neighbors).

where ξ is a randomly generated projector coordinate, n is the total number of periods and $N(0, \sigma)$ is a normally distributed unbiased random variable with standard deviation σ. We also assume the noise magnitude σ to be identical for all the phase observations, which is reasonable since it mainly depends on the number of samples, which is usually the same for all the periods.

In our first batch of experiments we generated 10^6 samples for different values of σ and an amount of neighbors checks fixed to 10. We applied respectively IFC, VFC and CFC to recover the faults from plain multi-period phase shift (MPS). We classified a point as recovered when a method assigns to it the correct set of fringe numbers.

The results are shown on the left plot of Figure 5.3. The red line represents the amount of points correctly unwrapped by MPS at the first round. The dotted lines show the amount of points respectively recovered by IFC, VFC and CFC and the continuous lines of the same color the total number of unwrapped points by combining the initial set with the ones recovered by each method. There are several observation that can be made by looking at this data. First of all, even with a small amount of noise (about 2% over the normalized phase value), MPS fails about 50% of the times. This is a well-known limitation of MPS, which indeed reduces greatly its feasibility in real-world products. Interesting enough, both IFC and VFC work reasonably with low noise levels, still their performance drops fast. The failure of VFC means that with high noise levels it is difficult for the correct fringe vector to consinstenly gain major consensus. In a similar manner, the failure of IVC means that even by seeking independently the consensus for each phase component, it is quite common to obtain a broken vector. This effect is likely due to the fact that each phase is independently observed and thus the probability of getting a full set of n correct consensus over fringes, gets smaller as n increases.

Differently, CFC works remarkably well also with a very high noise level. As a matter of fact it is able to recover all the correct fringes even when MPS offers less than 10% of unwrapped points. This is partially expected, since for CFC to work it is required that the correct fringe appers at least in one neighbor. This is a very loose requirement, given that the observations are independent and the probability of not getting the correct fringe in n neighbors decreases quickly with n.

While this is a very encouraging result, the recovery of the unwrapping does not auto-

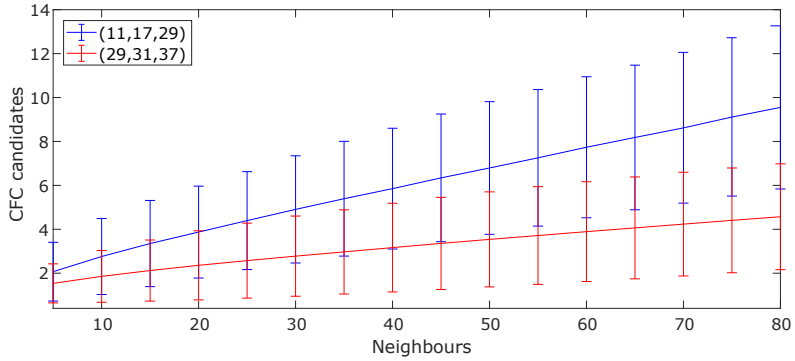


Figure 5.5: Number of candidate vectors actually tested by CFC for different values of parameter k.

matically implies a good accuracy. Undoubtedly the overall error obtained by combining errors in phase recovery when computing the average ξ could still lead to an unacceptable result. To analyze the overall accuracy we plotted, in the right part of Figure 5.3, the RMS error of the recovered points with respect to the ground-truth.

By looking at the plot it seems that CFC offers consistently better accuracy than IFC and VFC. Specifically, the accuracy of CFC is actually comparable with the degree of precision obtained by points directly unwrapped by MPS at the first round. The breaking point seems to be around a standard deviation on the synthetically-generated data of about 6%, which is indeed huge (about $\frac{1}{3}$ radians). Note also that, until that point, the recovery error is around a half code unit, which in practice corresponds to sub-pixel accuracy in projector coordinates. Finally, it should also be noted that the slow increase in the error exhibited by MPS after the 6% threshold is not really due to some particular merit, but its a simple consequence of the implied biased selection. In fact, only the observation characterized by low error are actually unwrapped by MPS at the first round.

5.4.2 Effect of the Number of Neighbor Checks

At this point CFC appears to be the best candidate for unwrapping faults recovery. However, since it works on the complete Cartesian product of fringe observation sets, it could end up checking much more candidates than IFC and VFC. For this reason, it is very important to verify that its accuracy can be achieved without needing an extensive search. To this end, we performed a batch of tests by setting the noise level at 6% and by exploring the effect of different choices for parameter k, that is the number of decoded surrounding pixels to be considered. The results are shown in Figure 5.4. It can be noted that the number of neighbors has actually a very limited effect on CFC, which is of course an important feature. In detail, while a choice of only 5 neighbors implies a slight drop in performance, 10 check points seem to be more than enough. Finally, we can observe an interesting phenomenon appearing when the number of checked neighbors grows too much. As a matter of fact, a large amount of neighbors results indeed in a slightly dimin-

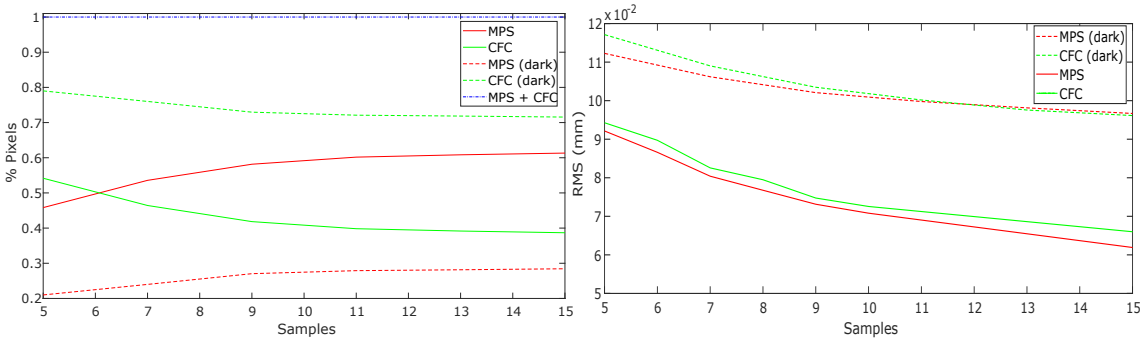


Figure 5.6: Comparison of the recovery rate and resulting RMS error achieved by CFC and MPS with a real plane.

ished accuracy and a larger standard deviation. This is probably due to the fact that, when the product set becomes very large, it could happen randomly that a wrong fringe number configuration is blessed by a higher coherency than the correct one. This is an important observation, in fact it offers an additional reason (beside computational feasibility) to avoid the naive approach of checking all the possible fringe configurations.

Regarding the computational feasibility of CFC, we are also interested in assessing how many candidates result from different number of neighbors checks. This is shown in Figure 5.5 for two different sets of period lengths. The number of actual candidates exhibits a large variance, since it depends a lot on the position of the observed point. Still, its magnitude is in general quite low and grows in a linear manner with the size of the neighbor set. In practice, since 10 neighbors have been shown to be a reasonable choice for a good performance of CFC, we can conclude that the recovery step would require to check a very small amount of candidates, with minimal impact in the overall execution time of the full pipeline over plain MPS.

5.4.3 Real-World Evaluation

In the previous sections we adopted synthetically generated data to enable an evaluation under controlled conditions of noise and with a well defined ground-truth.

Nevertheless, for a complete validation of the approach a demonstration of its effectiveness with an actual camera-projector setup is needed. To this end, we used a calibrated camera-projector pair composed of a CCD machine vision camera with 3Mpixels resolution, a full hd video projector and a disparity of about 20cm. The system was calibrated using [145] and verified using artificial markers [18, 22]. Since a proper ground-truth could not be available, we evaluated the performance by capturing planar surfaces of different materials and then by computing the average distance of each reconstructed point from an ideal plane obtained by fitting all the points reconstructed by using only MPS. This should be a reasonable substitute for a ground-truth since its a statistical measure based on a large number of reliable points. Since it's not possible to set the amount of desired noise in such kind of experiment, we evaluated the RMS error with respect to the

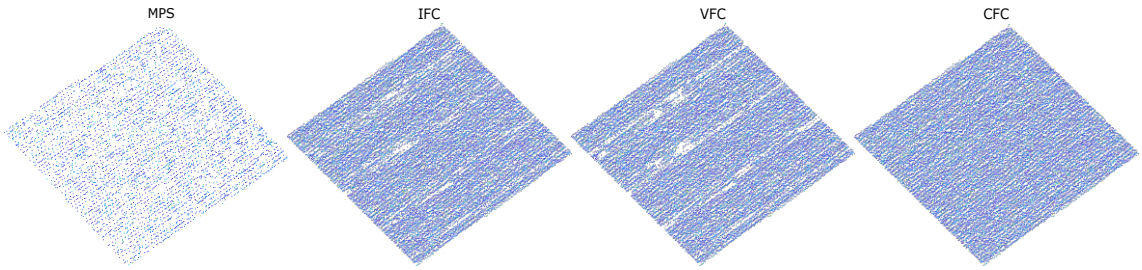


Figure 5.7: Examples of reconstruction of difficult surfaces using plain MPS (1st column), IFC (2nd column), VFC (3rd column) and CFC (4th column).

plane for different amounts of samples used for phase recovery. Under these conditions, the phase recovery error, and thus the observation noise, is expected to decrease with the square root of the number of samples. The results are shown in Figure 5.6, by mean of two plots similar to those shown in Figure 5.3. The observed trends confirm those obtained with synthetically generated data.

Finally, in Figure 5.7 we also show actual reconstruction examples to supply a basic intuition about the real effects of the different recovery rates and accuracy levels. The items reconstructed are a set of planar surfaces placed at a distance of about 1 meter from the projector-camera system.

5.4.4 Comparison with Gray Coding

As discussed, multi-period phase shift is very sensitive to noise. In fact, even a naive approach using a single phase shift, combined with gray coding disambiguation, can lead to more stable unwrapping results. Moreover, approaches designed to deal with the few unwrapping errors from gray coding have already been proposed in literature [234]. For this reason the practitioner could ask if there is any compelling reason to adopt our CFC extension of MPS. Actually there is a really important difference between the two approaches. When gray coding is used for disambiguation, the projected patterns do not contribute to the accuracy of the coding. In fact they are simply discarded once the fringe number for a point is recovered. Differently, with multi-period phase shift, and thus with CFC, all the single observed phases provide useful information. In Fig. 5.8 we report the result of an experiment performed with the same setup presented in Sec. 5.4.1, comparing the RMS error with respect to ground-truth obtained with CFC and the technique presented in [234]. The standard deviation of the noise is set to 6%. We applied this error to all the phases recovered with CFC (over three periods) and to the single phase observed by the state-of-the-art method [235]. In addition we assumed the unwrapping from gray coding to be always perfect. The advantage of CFC in terms of accuracy is quite prominent.

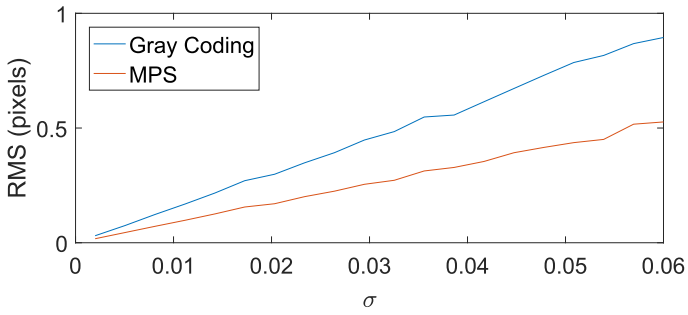


Figure 5.8: Enhanced accuracy of MPS/CFC with respect to gray coding

5.5 Conclusions

With this chapter we examined three strategies for phase unwrapping faults recovery. Among these, Complete Fringe-set Check (CFC) exhibited the best behaviour and we think it to be suitable to be adopted in practical scenarios. In comparison with other unwrapping techniques, the recovery method directly exploits the measured phase values in a neighbourhood of the missing code, computing the best values for the fringe number that allows absolute phase measurements. The advantages of such recovery methods are the flexibility with respect to the acquired scene and the ease of implementation, which in practice make it suitable to work in environments where a dense 3D reconstruction is needed. Despite being rather simple, the proposed technique definitely fixes a long standing problem with multi-period phase unwrapping methods, that is the inherent high sensitivity to noise. The effectiveness, efficiency and accuracy of CFC has been demonstrated by means of a thorough experimental evaluation performed over both real and synthetically generated data.

II

Applications

6

Calibrating a Camera-Projector Stereo Setup

Regardless the adopted coding strategy, extrinsic calibration is a necessary and fundamental process in every structured-light 3D reconstruction system. In fact, such procedure assesses the mutual relation between camera and projector and, after the pattern decoding step, allows for the subsequent triangulation of the observed scene. This chapter presents the calibration approach developed for the acquisition device employed in the practical applications discussed in this thesis. The system acquires a simple planar object with different orientations and, for each acquired image, the correspondences between camera and projector pixels are computed through phase unwrapping. Then, an optimization process computes the optimal transformation between camera and projector reference frame exploiting two-view geometrical properties of the system.

6.1 Introduction

The proposed calibration approach involves a camera-projector system and exploits the phase unwrapping technique as described in the previous chapters to assess the correspondences between camera image plane and projector image plane. In this method several shots of a planar surface are acquired. For each shot, the correspondences of camera and projector pixels induce a homography between the two image planes. Such relations are then decomposed and exploited to extract both the roto-translation between camera and projector plane and the projector's intrinsic parameters. Finally, an optimization step involving the projector lens distortion is carried out in order to assess the best system calibration.

6.2 Calibration Approach

In this approach we assume a camera-projector pair structured-light system, with the camera intrinsic parameters already calibrated through an independent procedure. The projector is also modelled as a camera, thus it is characterized by an intrinsic matrix and five radial distortion coefficients. In the initial step we consider the projector as a perfect pinhole system, with no lens distortion. This assumption is fair for the initial parameter estimation because the projector lenses are specifically designed not to be particularly distorted.

Suppose that the camera and the projector observe the same planar surface π in the scene. Then, each 3D point $\mathbf{X}$ lying on such plane has two projections $\mathbf{x}_c$ and $\mathbf{x}_p$, respectively on the camera and projector image plane. The images of such points captured by the camera are related to points on projector's image plane through a planar homography [79]. In other words, a plane π induces a homography H between the two views such that points on the camera image plane can be mapped to points on projector image plane and vice versa. Note that such homography is uniquely determined by the plane π. Figure 6.1 depicts the described scenario.

Since the two images of the same planar surface are related through an homography, we can express the relation between the points $\mathbf{x}_c$ and $\mathbf{x}_p$ (belonging to the plane) in the following way

$$\mathbf{x}_c = K_c \, H \, K_p^{-1} \, \mathbf{x}_p \tag{6.1}$$

where K_c and K_p are the calibration matrices respectively of the camera and the projector, while H is the homography matrix induced by the plane.

The matrix H can be decomposed as

$$H = R - \frac{\mathbf{t}\,\mathbf{n}^T}{d} \tag{6.2}$$

where R and $\mathbf{t}$ are the rotation matrix and the translation vector between the projector and the camera reference frames. The vector $\mathbf{n}$ is the normal of the observed plane and d is

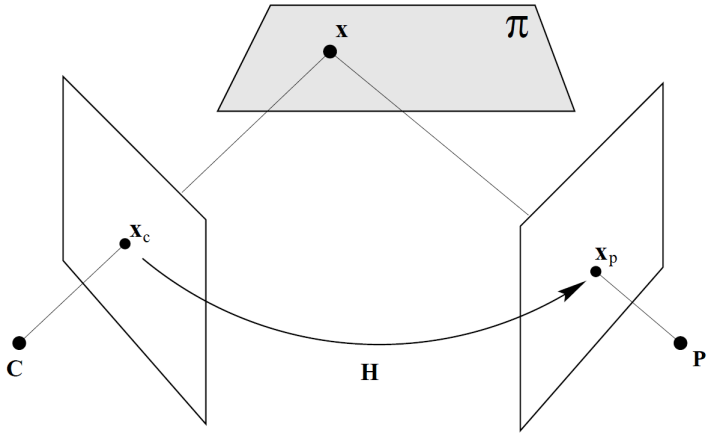


Figure 6.1: Homography H is induced by a plane π and two views that capture the same point $\mathbf{x}$ lying on the plane. Such homography maps points $\mathbf{x}_c$ on camera image plane to points $\mathbf{x}_p$ on projector image plane.

its distance from the camera centre. If we are not interested in the magnitude of $\mathbf{t}$, we can ignore the value of d including it in the computation of $\mathbf{t}$ and obtaining the calibration up to a scale factor.

6.2.1 Initial Parameter Estimation

During the calibration process, N phase maps are acquired. Each image captures a plane with a different pose. The phase unwrapping algorithm is applied to the acquired phase maps and thus the correspondences between camera pixels and projector coordinates are known for each acquired plane. Point correspondences are extracted by uniformly selecting m_i valid projector codes in each acquired image I_i, with $i = 1, ..., N$.

Each acquisition can then be associated with a homography matrix H_i with $i = 1, ..., N$, which in turn can be decomposed as described in Equation (6.2). In this way, each different pose is characterized by a triple $(R_i, \mathbf{t}_i, \mathbf{n}_i)$. Note that the value d in (6.2), that is the distance of the plane, can be ignored since it is only affected by a scale factor. While all the plane normal vectors $\mathbf{n}_i$ are different in each shot, the values R_i and $\mathbf{t}_i$ should be the same, since they represent the roto-translation between camera and projector image planes. The acquisition process and the phase localization are of course two tasks subject to several sources of noise, therefore the values R_i and $\mathbf{t}_i$ are not going to be identical, and they need to be averaged in a final common transformation. The operation is not trivial since we need to compute a combination of rotations and translations that is an average of the N values obtained from the different poses.

Each rigid motion can be associated with a dual quaternion, that is a mathematical tool allowing an effective representation of rigid motions [52]. Dual quaternions consider at the same time both the rotation and translation involved in a motion, offering good theoretical properties, so they are best suited to combining several roto-translations [95].

Thus, each pose $[R_i|\mathbf{t}_i]$ generates a dual quaternion denoted as $\mathbf{q}_i$. Such values are averaged following the linear blending procedure as described in [95], to obtain an averaged transformation $\bar{\mathbf{q}}$. Then, in order to make the original homographies represent the same initial relations, each matrix H_i needs to be adjusted according to the new roto-translation $\bar{\mathbf{q}}$, while keeping the original normal vectors $\mathbf{n}_i$.

After this process we need to compute an initial estimation of K_p, that is the intrinsic matrix for the projector. We can exploit the relation expressed in Equation (6.1), connecting the corresponding points in both image planes with the observed plane and the computed homography. Since we observed N planes, the associated homographies H_i can be used to compute an initial estimation of matrix K_p through a single linear least squares.

Assume we have m_i point correspondences in each $i = 1, ..., N$ acquisition, then we have $M = \sum_{i=1}^{N} m_i$ point correspondences in total for all the views. Equation (6.1) can then be rewritten in the following way

$$H^{-1}\, K_c^{-1}\, \mathbf{X}_c = K_p^{-1}\, \mathbf{X}_p \tag{6.3}$$

where $\mathbf{X}_c$ and $\mathbf{X}_p$ are $3 \times M$ matrices containing all the extracted point correspondences in homogeneous coordinates, for the camera and the projector respectively. Defining $C = \{c_{i,j}\} = H^{-1}\, K_c^{-1}\, \mathbf{X}_c$, we can define the coefficient matrix as

$$A = \begin{bmatrix} c_{1,1} & 1 & 0 & 0 \\ 0 & 0 & c_{2,1} & 1 \\ \vdots & \vdots & \vdots & \vdots \\ c_{1,M} & 1 & 0 & 0 \\ 0 & 0 & c_{2,M} & 1 \end{bmatrix} \tag{6.4}$$

that is a $2M \times 4$ matrix. Then we define the linear system

$$A\, \mathbf{x} = \mathbf{b} \tag{6.5}$$

where $\mathbf{b}$ is a $2M$-vector that is the linearisation of $\mathbf{X_p}$ (expressed with 2D inhomogeneous coordinates) and $\mathbf{x} = (f_x, c_x, f_y, c_y)^T$ contains the intrinsic parameters of the projector system, namely the focal length and the center offset. Such system can be easily solved to compute an initial estimation of the projector imaging parameters.

6.2.2 Distortion and Parameter Optimization

After the first estimation of both K_p and the roto-translation from projector to camera reference frame, these parameters need to be further optimized minimizing an energy function defined as follows

$$E(K_p, \mathbf{q}, \mathbf{t}, \mathbf{k}) = \sum_{i=1}^{N} \sum_{j=1}^{m_i} ||\mathbf{x}_c^j - (K_c\, H(\mathbf{q}, \mathbf{t}, \mathbf{n}_i)\, K_p^{-1}\, d(\mathbf{x}_p^j, \mathbf{k}))||^2 \tag{6.6}$$

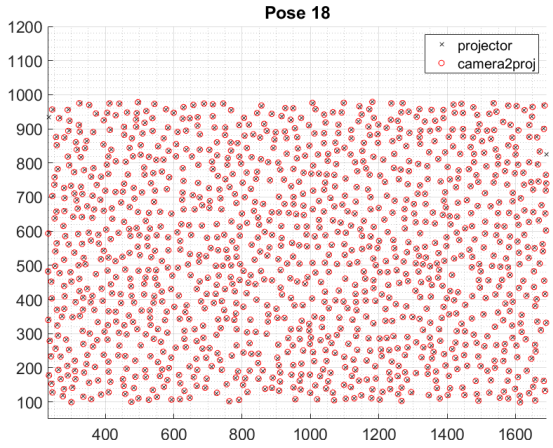


Figure 6.2: Example of extracted points in projector image plane (black cross) and corresponding points on camera image plane after applying the relative transformation (red circles). Such configuration is obtained after the first optimization step.

where $\mathbf{x}_c^j$ and $\mathbf{x}_p^j$ are the j-th observed points in camera and projector image planes expressed in homogeneous coordinates and $\mathbf{k}$ is a 5-dimensional vector containing the distortion parameters. The rotation between camera and projector is expressed as a unitary quaternion $\mathbf{q}$ rather than an usual rotation matrix R, that should be forced to be orthonormal during the optimization, making the process a constrained optimization. The function H takes the roto-translation and the i-th plane normal and gives the associated homography matrix as output, while the function d distort the observed points according to the distortion parameters in $\mathbf{k}$. Note that the distortion function here is used because the projector coordinates $\mathbf{x}_p$ observed from the camera are the "perfect" codes as produced in the projector image plan, thus they need to be distorted before applying the inverse projection and the homography.

The defined energy function computes the sum of the squared reprojection error (in pixels) for all the acquired points in all N plane acquisitions and it is first initialized with the initial estimated parameters as described in the previous section. At the end of the process, the optimized value for both projector intrinsics K_p and distortion and the relative pose between camera and projector image frames are simultaneously computed.

6.3 Experimental Setup

The described calibration process was carried out adopting a structured-light scanner developed in our lab. The device is composed by a single MatrixVision BlueFox3 5Mpix camera and an LG projector with a resolution of 1920×1080 px, both mounted in a structure with a vertical setup. The projector gamma has been manually corrected to obtain a camera signal response as linear as possible.

In each test we collected a set of 10 to 20 shots of a white planar surface with different

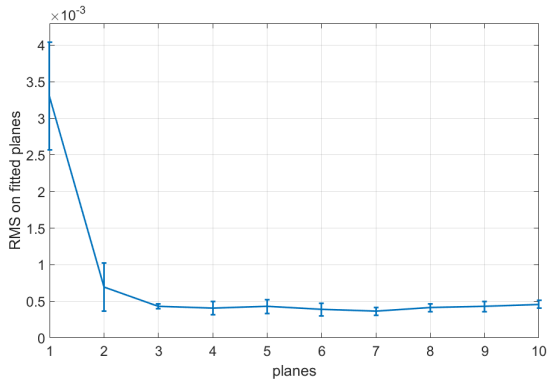


Figure 6.3: RMS of the triangulated point distance from the fitted planes increasing the number of planes used for calibration. Errorbars represent the standard deviation.

inclinations with respect to the scanner. The phase unwrapping technique described in Chapter 3 was applied to the acquired phase maps in order to obtain a coded image for each different plane.

From each coded image I_i a set of valid projector codes was selected through an extraction technique that ensures a uniform distribution of codes over the camera image plane. The extracted points were usually $\sim$ 1000 for each shot. Figure 6.2 shows an example of extracted points for one shot in the projector image plane.

An initial guess for the projector intrinsic parameters has been used for all the experiments, that is $f_x = f_y = 3000$ and $c_x = 1080$, $c_y = 960$ and the initial homographies were computed using all the given points with a basic least squares approach.

The optimization technique used for minimize the energy function (with projector distortion coefficients) uses the Nelder-Mead simplex method as described in [108]. We fixed a maximum number of iterations equal to 10000, with a tolerance of 10^{-6} pixels.

The algorithm stability was tested using an increasing number of planes to calibrate the system and validating the calibration parameters using another plane that was not included in the calibration process. In such experiment we had a pool of 18 different plane poses. In turn we removed one of these planes (used for validation) and calibrated using an increasing number of randomly extracted set of planes.

Figure 6.3 shows the validation results, computed as the RMS of the distance of triangulated points with respect to the fitted plane model (not used during calibration). We tested the calibration performances by randomly extracting 1 to 10 planes and repeating the process 18 times, one for each different validation plane. Results show that the algorithm offers a precision below 0.5 pixels even using three planes. Moreover, the standard deviation decreases as the number of plane raises, implying a good stability.

Finally, Figure 6.4 shows the calibrated system together with some of the 3D points lying on the planes used for calibration.

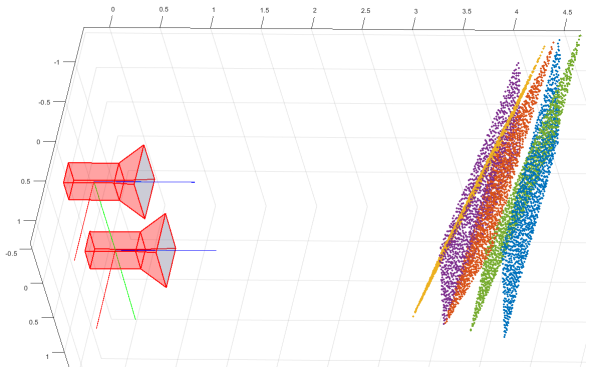


Figure 6.4: Final calibration results with a subset of the acquired planes and the associated triangulated 3D points extracted.

6.4 Conclusions

In this Chapter we introduced a novel calibration method for a camera-projector pair structured-light scanner. The method employs a number of acquisitions of a planar surface with different poses and uses our phase unwrapping technique to devise dense camera-projector point correspondences. The final optimization allows for an accurate estimation of both projector intrinsic and extrinsics parameters in a simultaneous way with no need of a specialised calibration target.

7

Cylinders Extraction in Non-oriented Point Clouds

Finding geometric primitives in 3D point clouds is a fundamental task in many engineering applications such as robotics, autonomous-vehicles and automated industrial inspection. Among all solid shapes, cylinders are frequently found in a variety of scenes, comprising natural or man-made objects. Despite their ubiquitous presence, automated extraction and fitting can become challenging if performed ”in-the-wild”, when the number of primitives is unknown or the point cloud is noisy and not oriented.

In this Chapter we pose the problem of extracting multiple cylinders in a scene by means of a Game-Theoretic inlier selection process exploiting the geometrical relations between pairs of axis candidates. First, we formulate the similarity between two possible cylinders considering the rigid motion aligning the two axes to the same line. This motion is represented with a unitary dual-quaternion so that the distance between two cylinders is induced by the length of the shortest geodesic path in SE(3). Then, a Game-Theoretical process exploits such similarity function to extract sets of primitives maximizing their inner mutual consensus. The outcome of the evolutionary process consists in a probability distribution over the sets of candidates (ie axes), which in turn is used to directly estimate the final cylinder parameters. An extensive experimental section shows that the proposed algorithm offers a high resilience to noise, since the process inherently discards inconsistent data. Compared to other methods, it does not need point normals and does not require a fine tuning of multiple parameters.

7.1 Introduction

The identification and extraction of geometrical primitives (planes, spheres, cylinders, pyramids) from a scene is an essential task in several scenarios [93, 121], such as industrial automation and inspection [74, 172], reverse engineering [135] or robot manipulation [196]. Despite the high quality offered by modern acquisition devices, in some situations 3D data are far from being perfect. For instance, the scanning process of objects presenting shiny metallic areas could easily lead to an incomplete or noisy 3D surface [11]. Furthermore, many times the acquired scene is complex and contains several elements which are to be excluded as they are not relevant for the subsequent analysis. Indeed, any practical shape extraction algorithm must provide a reliable method to filter out outliers, in addition to a good parameter estimation.

In this Chapter we focus on the specific task of cylinder extraction from generic 3D point clouds. Cylindrical shapes are often found in many applications such as natural landscape analysis [157, 165], robot manipulation [142], automated pipe-run reconstruction [169] or industrial quality inspection [138].

In general, when dealing with primitives identification, we can distinguish two conceptually different tasks, namely *fitting* and *extraction*, which differ in the type of problem and how it is addressed. In the case of fitting the point cloud has already been segmented, so a subset of 3D points is associated to a possible cylindrical shape. The objective of a fitting algorithm is to compute the best parameters according to the given data, typically minimizing the average error with respect to the theoretical surface. On the other hand, the goal of extraction methods is to detect the primitives included in a complex point cloud and then devise their parameters. These approaches are often employed when the whole scene is captured, making the task more challenging than a simple fitting since data could be incomplete, noisy or include clutter. Usually there are no assumptions on the number of primitives in the scene, neither on their orientations. Moreover, the presence of point normals is not always ensured and their quality may depend on the acquisition device or the used reconstruction algorithm. Extraction methods employ statistically robust algorithms that ensure a high resilience to outlier data, but they could be less accurate when compared with the previous techniques.

Novel Contribution

The method proposed in this Chapter improves the concept to extract multiple cylinders from an unknown configuration with several sources of noise. In particular, we formulate a cylinder similarity function expressing the compatibility between two possible candidates, and a simple yet effective candidate extraction technique based on geometrical properties of cylindrical primitives. These two ingredients are then combined in a clustering process based on evolutionary game theory, that ensures the extraction of the subset of candidates with best mutual support and exhibits excellent robustness to outliers.

The described approach offers three main advantages: first, it does not require an initial tuning of several parameters nor any prior knowledge of the scene. Second, the

inlier selection provides a probability distribution over the final subset of candidates, so that the final model model is computed by averaging the relative fitting scores. Third, we achieve good estimation accuracy even if most of the cylindrical shapes are occluded or covered by clutter. An extensive experimental evaluation presented, demonstrating all these features on both synthetic and real-world cases. This extraction method has been published in [162].

7.1.1 Dual Quaternions and Rigid Motions

In this Section we give a minimal introduction to dual quaternions to understand the proposed distance function, suggesting the reader to [54, 96] for more information on dual numbers and dual quaternion algebra for geometrical applications.

Dual quaternions are mathematical objects defined by a composition of quaternions and dual numbers. The former are common for researchers and practitioners working with rotations, since it is well known that rotations around the origin can be efficiently represented by unitary quaternions. The latter is less known, but simple in practice. A dual number can be written as $\hat{a} = a_0 + \epsilon a_\epsilon$ where ϵ is called *dual unit*. The algebra is similar to complex numbers, except that $\epsilon^2 = 0$. Indeed, the dual conjugate is analog to complex conjugate $\bar{\hat{a}} = a_0 - \epsilon a_\epsilon$ and the multiplication is given by the formula $(a_0 + \epsilon a_\epsilon)(b_0 + \epsilon b_\epsilon) = a_0 b_0 + \epsilon(a_0 b_\epsilon + a_\epsilon b_0)$. The square root of a dual number with non zero scalar part a_0 is given by the formula

$$\sqrt{a_0 + \epsilon a_\epsilon} = \sqrt{a_0} + \epsilon \frac{a_\epsilon}{2\sqrt{a_0}}. \tag{7.1}$$

Interestingly, the Taylor series of a function with dual argument is limited to the first order because higher powers of ϵ are zero. Therefore, sine and cosine of a dual number are easily defined as:

$$\begin{aligned} \sin(a_0 + \epsilon a_\epsilon) &= sin(a_0) + \epsilon a_\epsilon cos(a_0) \\ \cos(a_0 + \epsilon a_\epsilon) &= cos(a_0) + \epsilon a_\epsilon sin(a_0). \end{aligned}$$

Since the dual unit commutes with quaternion units (ie $\mathbf{i}\epsilon = \epsilon\mathbf{i}$), a dual quaternion $\hat{\mathbf{q}}$ can be written as an ordinary quaternion of dual numbers $\hat{\mathbf{q}} = \hat{w} + i\hat{x} + j\hat{y} + k\hat{z}$ (where $\mathbf{i} = (i, j, k)$) or as a dual number in which the scalar and dual part are ordinary quaternions $\hat{\mathbf{q}} = \mathbf{q_0} + \epsilon\mathbf{q}_\epsilon$. The norm of a dual quaternion is defined as $\|\hat{\mathbf{q}}\| = \sqrt{\hat{\mathbf{q}}^* \hat{\mathbf{q}}}$, where $\hat{\mathbf{q}}^*$ denotes the classical quaternion conjugation $\hat{\mathbf{q}}^* = \mathbf{q}^* + \epsilon\mathbf{q}_\epsilon^*$. The set of dual quaternions for which $\|\hat{\mathbf{q}}\| = 1$ are called *unitary*. It can be demonstrated that every rigid transformation can be represented by a unitary dual quaternion, and conversely, every unitary dual quaternion represents a rigid transformation. Finally, for the so-called *antipodal* property of dual quaternions, $\hat{\mathbf{q}}$ and $-\hat{\mathbf{q}}$ represent the same transformation but with two different trajectories.

Unitary Dual Quaternion Manifold

The screw motion between l_1 and l_2 can be related to the scalar and vector part of a unit dual quaternion $\hat{\mathbf{q}}$ according to the following formula:

$$\begin{aligned} \hat{\mathbf{q}} &= \cos\frac{\hat{\theta}}{2} + \hat{\mathbf{s}}\sin\frac{\hat{\theta}}{2} \\ \hat{\theta} &= 2\beta + \epsilon 2\|T\| \\ \hat{\mathbf{s}} &= \mathbf{i}\vec{s} + \epsilon\mathbf{i}(m_1 \times \vec{s}) \end{aligned} \tag{7.2}$$

where $\hat{\mathbf{s}}$ is a dual quaternion (with zero scalar part) describing the rotation axis and $\hat{\theta}$ is the dual angle expressing the angle of rotation and the amount of translation.

The set of unit dual quaternions can be seen as 6-dimensional manifold embedded an 8-dimensional Euclidean space [140]. The corresponding log map can be defined in closed form as:

$$log(\hat{\mathbf{q}}) = \hat{\mathbf{s}}\frac{\hat{\theta}}{2}. \tag{7.3}$$

Such log map allow us to define a distance function between two lines l_1 and l_2 induced by the length of the geodesic path in SE(3) from the origin (ie. the identity motion) to $\hat{\mathbf{q}}$. Let $\hat{\mathbf{q}}$ be the screw motion described in 7.2.1, we define such function as:

$$d(l_1, l_2) = \min(\|\log(\hat{\mathbf{q}})\|, \|\log(-\hat{\mathbf{q}})\|) \tag{7.4}$$

In fact, it represents the length of the shortest geodesic path of the two trajectories described by $\hat{\mathbf{q}}$ due to the antipodal property.
Operatively, the distance between two axes l_1 and l_2 in our application is computed as follows:

1. Determining the points m_1 and m_2, and the distance vector $T = m_2 - m_1$
2. Computing the axis vector $\vec{s}$
3. Computing the dual quaternion $\hat{\mathbf{q}}$ as in (7.2)
4. Computing the length of the geodesic path as in (7.4)

7.2 The Proposed Method

Our goal is to extract cylindrical shapes from non-oriented point clouds. We simplify the operation by considering only cylinders with an infinite extent along the axis, thus reducing their parametrization to a 3D line in space, called *axis*, and a radius. In practice, this is not a limiting assumption since (i) co-axial cylinders with different radii are rare

in common scenes, and (ii) it is simple to estimate the cylinder height once the 3D points associated with its shape are identified in the cloud.

We parametrize a cylinder ζ with the triplet $(p, \vec{v}, r)$, in which ($p \in \mathbb{R}^3$, $\vec{v} \in \mathbb{R}^3$) and $r \in \mathbb{R}$ describe its axis and radius respectively. We pose the additional constraints of $\|v\| = 1$ and $\langle p, \vec{v} \rangle = 0$, resulting in only 5 degrees of freedom out of the 7 parameters used for the parametrization.

Our method considers the cylinder extraction as a clustering problem. We suppose to have a set of *cylinder candidates* representing possible alternative cylindrical shapes observed in the scene. We designed a robust Game-theoretic process so that candidates coherent with a common cylindrical model are grouped together and weighted according to their mutual fitness. Therefore, by repeating this selection until all candidates are grouped, we robustly extract a sequence of cylinders ordered by their overall consistency with the relative candidate cluster. To support this principle, in section 7.2.1 we formulate a distance function among pairs of candidates that properly accounts for the 3D displacement of their relative axes. Then, in section 7.2.2 we propose a possible way to generate robust candidates and in 7.2.3 we discuss the details of the Game-theoretical process.

7.2.1 Cylinder Distance Function

Cylinder candidates are clustered together by maximizing a similarity among elements of the same cluster. The non-trivial aspect is that, a part from the relative radii, a well-defined function must deem the combined contribution of axes positions and orientations. Therefore, defining a similarity between two cylinders boils down in defining a distance between 3D lines (ie. axes) in space.

Let $l_1 = (c_1, \vec{v}_1)$ and $l_2 = (c_2, \vec{v}_2)$ be two 3D lines defined respectively by a point c and a unitary-norm direction $\vec{v}$. Following [20], we construct a rigid motion $(\bar{\mathbf{R}}, \bar{T})$ mapping line l_1 to l_2 or, more formally, the roto-translation such that $\forall p \in l_1, \exists\ k \in \mathbb{R}.\ \bar{\mathbf{R}}p + \bar{T} = c_2 + k\vec{v}_2$. Since every rotation around $\vec{v}$ and every translation along $\vec{v}$ transform a line in itself, there exist infinitely many $(\bar{\mathbf{R}}, \bar{T})$ mapping l_1 in l_2. Among those, we can express the one exhibiting the shortest translation length and smaller rotation angle taking advantage of the kinematic notion of a screw. Chasles' theorem [40] states that every rigid motion can be described by means of a translation along a unique line (called screw axis) followed (or preceded) by a rotation about the same axis. The motion described in the theorem is called *screw* and, since its discovery, is one of the most convenient ways to describe spatial movements. In our setting, we can directly describe $(\bar{\mathbf{R}}, \bar{T})$ in terms of a screw motion, as sketched in Fig. 7.1. Let $m_1 = c_1 + t_1\vec{v}_1$ (for some scalar t_1) be the point lying on l_1 closest to the line l_2. Similarly, let $m_2 = c_2 + t_2\vec{v}_2$ be the point of l_2 closest to line l_1. By construction, the vector $T = m_2 - m_1$ is orthogonal to both l_1 and l_2 and its length is the minimum distance between all the points of the two lines. This implies that T has the same orientation (but possibly different magnitude and sense) of the vector $\vec{v}_\perp = \vec{v}_1 \times \vec{v}_2$. So, the rotation with angle $\beta = \arcsin(\|\vec{v}_\perp\|)$ around the axis defined by point m_1 and vector $\vec{s} = \vec{v}_\perp / \|\vec{v}_\perp\|$ will let l_1 be parallel to l_2. Since m_1 lies on the rotation axis, its position will not change after the transformation, hence the distance

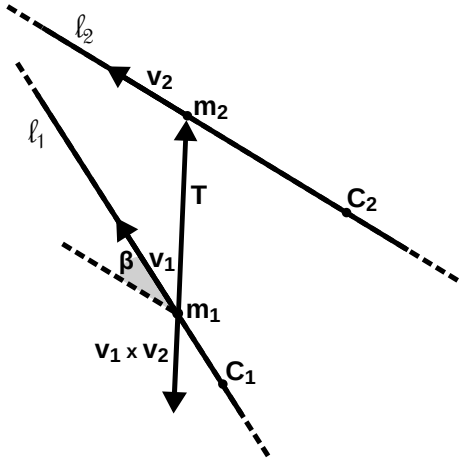


Figure 7.1: The defined screw motion between lines l_1 and l_2

between the two lines will remain equal to $\|T\|$. Trivially, a final translation along T will let l_1 coincide with l_2. To summarize, the minimum screw motion between the two lines is the one with axis $(m_1, \vec{s})$, translation length $\|T\|$ and rotation angle β.

This modelling is convenient for our purposes because we can relate the parameters of a screw-motion with the scalar and vector part of a unitary dual quaternion $\hat{\mathbf{q}}$ (refer to Section 7.1.1 for more details about dual quaternions).

The distance is then defined as following

$$d(l_1, l_2) = \min(\|\log(\hat{\mathbf{q}})\|, \|\log(-\hat{\mathbf{q}})\|). \tag{7.5}$$

Given such axes distance, we can easily derive a similarity function between a couple of cylinders $\zeta_1 = (p_1, \vec{v}_1, r_1)$ and $\zeta_2 = (p_2, \vec{v}_2, r_2)$.

$$\pi(\zeta_1, \zeta_2) = \begin{cases} e^{-\frac{d\left((c_1,\vec{v}_1),(c_2,\vec{v}_2)\right)}{\lambda}} & \text{if} \quad Ad(\zeta_1, \zeta_2) < r_1 + r_2 \\ & \quad \wedge \quad \frac{min(r_1,r_2)}{max(r_1,r_2)} > p \\ 0 & \text{otherwise} \end{cases} \tag{7.6}$$

where $Ad(\zeta_1, \zeta_2) = \|T\|$ is the minimum distance between two axes $(c_1, \vec{v}_1)$ and $(c_2, \vec{v}_2)$ as defined previously, and $\lambda > 0$ is a free parameter [1]. The value $p \in [0, 1]$ is the minimum ratio between the radii r_1 and r_2 required for two primitives to be compatible. The similarity function takes values between 0 and 1 and is inversely proportional to the axis distance defined in (7.5), thanks to the negative exponential with factor λ. Note that a cylinder is fully compatible with itself (the ray distance is equal to zero, so the similarity will be one) and not compatible at all if the axes relative position and the radii are not consistent with each other.

[1] In all our tests we fixed $\lambda = 5$.

We will employ such definition of similarity in a clustering process based on Game-Theoretical framework, which allows the extraction of the more coherent subset of primitives among all candidates.

7.2.2 Candidate Extraction

There are several ways to obtain possible cylinder candidates from a 3D point cloud. The choice is influenced by the acquisition device, that could introduce some limitations due to its functioning and level of accuracy. For example, data could not be uniformly distributed in space and point normals may be available or not. In this approach we do not assume points to have a specific structure, neither to have normals. In this way the problem is kept as general as possible, and the method can be applied in a variety of practical applications.

Our candidate extraction is based on random slicing planes over the scene: we randomly generate N planes and compute the intersection between them and the point cloud. In this way we obtain N sets of 2-dimensional scattered points used to fit all possible ellipses, which in turn are used to compute the cylinder candidates.

To simplify the discussion, let's assume to have acquired a scene containing one single cylindrical shape and many other points resulting from other objects or noise. The intersection between a cylinder ζ and a plane P (whose normal $\vec{n}$ is not orthogonal with $\vec{v}$) results in an ellipse E with the following properties (see Fig. 7.3, Left):

- The minor semi-axis b is equal to the cylinder's radius r;
- the ellipse center c lies on the cylinder's axis;
- the ratio between the major semi-axis a and minor semi-axis b is a function of the angle α between $\vec{v}$ and $\vec{n}$, according to:

$$\cos\alpha = \frac{b}{a} \tag{7.7}$$

If 3D data is dense enough, a subset S of points lying sufficiently close to the plane P can be extracted, generating a set of 2D points. Figure 7.2 shows a simple example of this extraction process, starting with a point cloud and a slicing plane. According to what said before, some of these points will lie on the ellipse corresponding to the intersection between plane and cylinder, while some others will be distributed without any particular pattern (ie. they are outliers with respect to the elliptical shape). Therefore, any sufficiently robust fitting operation that extracts the elliptical shape E out of S will give clues on the unknown cylinder enclosed in the scene.

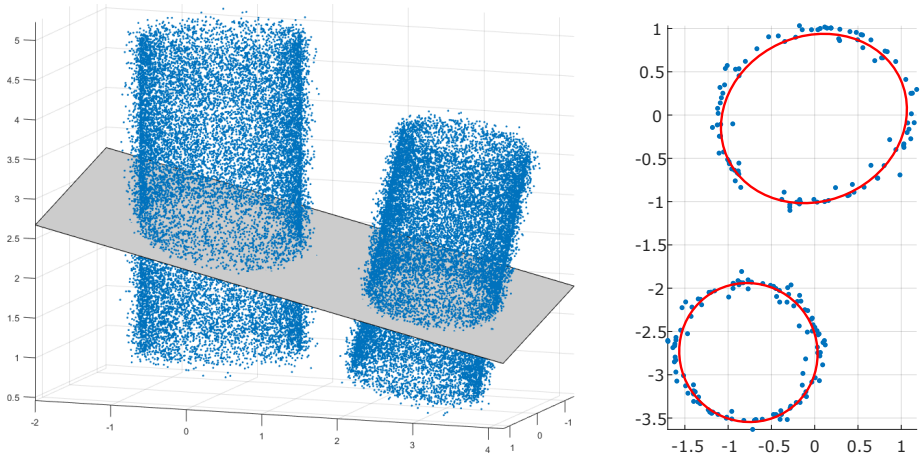


Figure 7.2: Left: an example of point cloud containing two cylinders and a random slicing plane. Right: scattered points lying on the planar slice and the corresponding fitted ellipses. Each ellipse will be associated with two possible axes so this plane will generate four cylinder candidates.

In particular, we already observed that the minor axis of E corresponds to the radius of the cylinder, while the center c defines the position of the axis. The ratio between semi-axes a and b will determine the axis direction $\vec{v}$ up to two equally-possible configurations (see Fig.7.3, Right). Due to this intrinsic ambiguity, at least two plane slices are needed to fully recover a cylinder.

In [171] the authors propose to use multiple slices to collect only the centers $(c_1 \dots c_n)$ and the minor axes $(b_1 \dots b_n)$ of each fitted ellipses. With this method, the cylinder radius is recovered by averaging each $b_1 \dots b_n$ and the axis $(p, \vec{v})$ by fitting a 3D line to $c_1 \dots c_n$ in a least-squares sense. The approach has three drawbacks. First, it totally discards information given by each ellipse regarding the possible orientation of cylinder axis (due to the already mentioned ambiguity). Second, it can only handle a single cylinder in the data. Third, it can badly suffer from unlucky plane slices (like areas with few scene content), since all values are averaged without weighting their reliability. This aspect is partially relevant though, since the subsequent non-linear optimization may still be able to recover the correct shape.

Our approach follows a similar strategy: we slice the scene with n different random planes. From each 2D scattered point set obtained from the i-th slice, we robustly extract all the $j = 1 \dots N_i$ possible ellipses using RANSAC and the ellipse model estimation method described in [75]. When considered in 3D space, each extracted ellipse $E_{i,j}$ is defined by its major and minor semi-axes vectors $\vec{a}_{i,j} \in \mathbb{R}^3$ and $\vec{b}_{i,j} \in \mathbb{R}^3$ and its center $c_{i,j} \in \mathbb{R}^3$. By construction, $\vec{a}_{i,j}$ and $\vec{b}_{i,j}$ are both defined up to a sign and orthogonal to the normal $\vec{n}_i$ of the i-th plane.

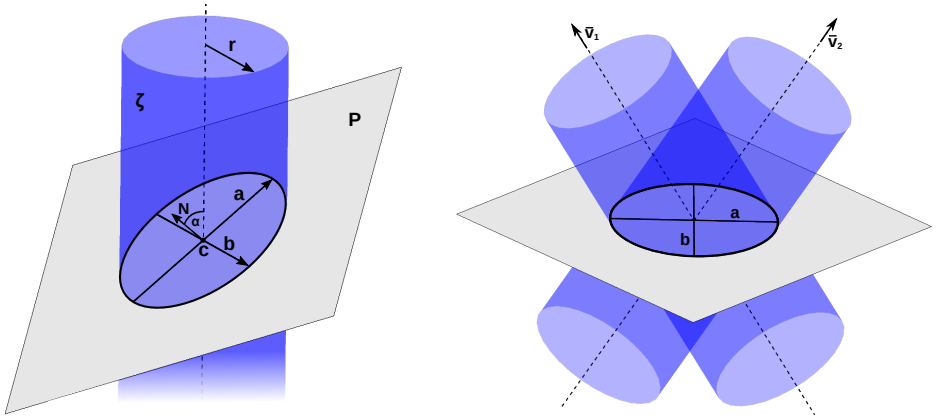


Figure 7.3: Left: Intersection between a cylinder ζ and a plane P. Right: The two possible cylinders deriving from an ellipse in 3D space. They have radius and center point in common but differ in the axis direction (denoted as $\vec{v}_1$ and $\vec{v}_2$).

From each $E_{i,j}$, we compute the two possible cylinder candidates as the tuples $(r, c, \vec{v} = \mathbf{R}\vec{n}_i)_{i,j}$ and $(r, c, \vec{v} = \mathbf{R}^T\vec{n}_i)_{i,j}$, where:

$$\begin{aligned} r &= \|\vec{b}_{i,j}\| \\ c &= c_{i,j} \\ \mathbf{R} &= \mathbf{I} + \sqrt{1 - \frac{\|\vec{b}_{i,j}\|^2}{\|\vec{a}_{i,j}\|^2}} \left[\frac{\vec{b}_{i,j}}{\|\vec{b}_{i,j}\|}\right]_\times + \\ &\quad + (1 - \frac{\|\vec{b}_{i,j}\|}{\|\vec{a}_{i,j}\|}) \left[\frac{\vec{b}_{i,j}}{\|\vec{b}_{i,j}\|}\right]_\times^2. \end{aligned} \tag{7.8}$$

In the tuples, r and c denote the cylinder radius and axis point shared by the two candidates and $\mathbf{R}\,\vec{n}_i$, $\mathbf{R}^T\vec{n}_i$ are two possible axis orientations obtained by rotating the plane normal $\vec{n}_i$ around the minor-axis of the ellipse $E_{i,j}$ with an angle of $\pm \arccos \frac{\|\vec{b}_{i,j}\|}{\|\vec{a}_{i,j}\|}$[2].

Note that we can extract several ellipses from a single plane slice (each one originating two possible axes), but all cylinder candidates $\xi = (r, c, \vec{v})_{i,j}$ generated from slice i are not compatible by construction and hence should be assigned to different clusters. Indeed, it is simple to observe that distinct coplanar ellipses can only be generated by distinct cylinders and the two axes coming from the same ellipse are mutually exclusive.

Thanks to this property we can reformulate the similarity function, taking into account the plane which generated each candidate.

[2]Matrix $\mathbf{R}$ is obtained by the Rodrigues' formula. The symbol $[.]_\times$ denotes the skew-symmetric matrix of the cross-product.

Therefore, the new similarity between two candidates $\xi_1 = (r_1, c_1, \vec{v}_1^k)_{i_1,j_1}$ and $\xi_2 = (r_2, c_2, \vec{v}_2^h)_{i_2,j_2}$ is the following

$$\pi'(\xi_1, \xi_2) = \begin{cases} 1 & \text{if } \xi_1 = \xi_2 \\ \pi(\xi_1, \xi_2) & \text{if } \quad i_1 \neq i_2 \\ 0 & \text{otherwise} \end{cases} \tag{7.9}$$

where $\pi(\xi_1, \xi_2)$ is the cylinder similarity function as defined in (7.6). The additional condition is added in order to explicitly exclude primitives generated from coplanar ellipses: in this way all pairs of candidates coming from the same slicing plane will have zero similarity. This formulation is particularly convenient in the clustering approach explained in the next section since the evolutionary process used to extract sets of coherent observations will inherently exclude incompatible candidates from the final result.

7.2.3 Inlier Selection Process

In the proposed approach the extraction of a subset of candidates containing elements with a high mutual support is carried out in the context of Evolutionary Game Theory [219]. We consider a scenario in which we have an initial population of candidates, where each distinct couple of individuals play a game one versus the other on the basis of preprogrammed strategies. Then, a selection process brings some elements which match well together to thrive, while driving the "unfitting" ones to extinction.

In our specific case, we consider all the $M = \sum_{i=1}^{n} N_i$ cylinder candidates $\xi_1 \dots \xi_M$ as the individuals, that we call *hypotheses*. Rather than the singular characteristics of each of them, we are interested in expressing how well two hypotheses play together. In other words, we need to define their behaviour when they are put together in the final population. We define such fitness through a *payoff-function*, which in our case is formulated as in (7.9). This function can be conveniently represented as an $M \times M$ symmetric matrix Π in which an element at row i and column j is defined as follows:

$$\Pi(i, j) = \pi'(\xi_i, \xi_j). \tag{7.10}$$

We then consider a discrete probability distribution $\mathbf{x} = (x_1, x_2, \dots, x_M)$ over the set of hypotheses (the entire population). The problem of finding a coherent set of candidates can be defined formally as finding a particular distribution in the standard simplex $\mathbf{x} \in \Delta^M$ maximizing the average payoff $\mathbf{x}^T \Pi \mathbf{x}$.

One possible solution to the problem employs an evolutionary process starting from a uniform distribution $\mathbf{x}^{(0)}$ and making it evolve through the well-known *discrete-time replicator dynamic* [204]

$$x_i^{(t+1)} = x_i^{(t)} \frac{(\Pi \mathbf{x}^{(t)})_i}{{\mathbf{x}^{(t)}}^T \Pi \mathbf{x}^{(t)}} \tag{7.11}$$

Equation (7.11) belongs to a class of evolutionary dynamics called *Payoff-Monotonic Dynamics* that are guaranteed to converge to a solution $\hat{\mathbf{x}}$ in which the set of hypotheses

with associated probability greater than 0 (namely the *support* of a population) cannot include any strategy with mutual payoff equal to 0. In other words, the candidates with associated non-zero probability in $\hat{\mathbf{x}}$ will only be the ones generated from different planes and with the distance between the two axes greater than the sum of the two radii. As a result, only the most consistent cylindrical model will emerge from the whole set of candidates after the inlier selection process.

7.2.4 Averaging the Winning Candidates

At convergence, a probability distribution over the candidates $\xi_1 \dots \xi_M$ is obtained, but we are still left with the problem of estimating the resulting cylinder parameters out of it. If we just consider the support of the population, we might use the same approach of [171] to just fit the cylinder axis line to the centers of the winning candidates. However, we now have the chance to produce a better estimation by weighting the contribution of each candidate considering the probability distribution $\hat{\mathbf{x}}$.

The estimation of the final cylinder $\zeta = (p, \vec{v}, r)$ is divided in two parts. The radius r is estimated as the weighted average of the candidate radii (7.12)

$$r = \sum_{i=1}^{M} \hat{x}_i r_i \tag{7.12}$$

where r_i is the radius of ξ_i. Then, the axis $(p, \vec{v})$ is averaged using the same approach of [20], posing the problem of 3D lines interpolation in terms of rigid motions blending. The general approach is as following:

1. An initial axis estimate $(p', \vec{v}')$ is computed by linearly interpolating the axis point p' and direction $\vec{v}'$ independently, according to the weights given in $\hat{\mathbf{x}}$

2. All the screw motions between $(p', \vec{v}')$ and the cylinder candidates $\xi_1 \dots \xi_M$ are computed according to (7.2)

3. The screw motions are averaged using the *Dual-quatertions Iterative Blending* [97] to obtain an average motion $\mathbf{q}*$

4. $\mathbf{q}*$ is applied to $(p', \vec{v}')$ to get the final cylinder axis.

7.2.5 Extracting Multiple Cylinders

The process described so far can extract only one cylinder at a time. To extract multiple cylinders, the two operations are simply repeated by ensuring that the candidates resulting from one run of the game theoretical process cannot be selected anymore. Specifically, after the extraction of each cylinder, we manually modify the payoff matrix Π according to the current population support.

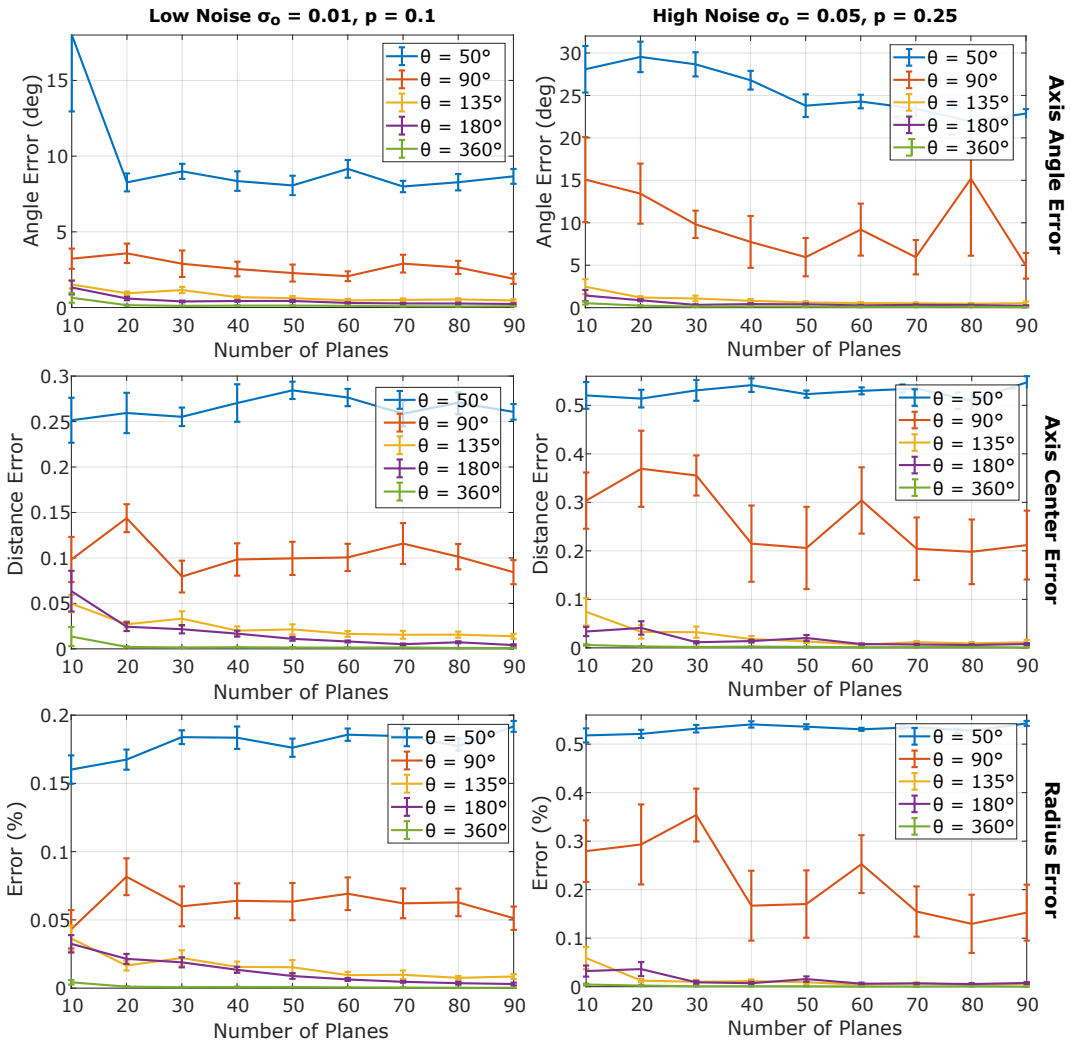


Figure 7.4: Angle, center and radius errors varying the number of randomly generated planes. Each curve corresponds to a different surface coverage (the angle θ). Results are shown for low (1st row) and high (2nd row) noise conditions.

Let $k_1, k_2, \dots k_h$ be the set of indices corresponding to the winning strategies. This means that they are non-zero elements in the final distribution $\hat{\mathbf{x}}$ (ie. $\hat{x}_K \neq 0 \quad \forall K \in k_1 \dots k_h$). The payoff matrix Π is then modified so that all rows and columns at indices $k_1 \dots k_h$ are set to zero (they are considered fully incompatible with any other candidate in the population). After that, the initial population $\mathbf{x}^0$ is reset to the uniform distribution and the inlier selection process (7.11) is run again until convergence. Since we are using a payoff-monotonic dynamic, all the previous candidates indexed by $k_1 \dots k_h$ cannot be present again in the next set of winning candidates. The whole process is repeated until all cylinders are extracted from the scene.

7.3 Experimental Evaluation

In this section we present a set of experiments that we performed to assess the correctness and robustness of the proposed technique in several situations. Since this approach is designed to work with data coming from real scans, we tested it against both synthetic and real-world data. The synthetic setup allows us to estimate the behaviour of our cylinder fitting algorithm while adjusting some external factors such as noise levels, outlier quantity and portion of acquired surface. Moreover, we used such controlled environment to compare the proposed technique with other fitting approaches. Finally, at the end of this section, we use real-world data to test the extraction ability in some real-world situations.

Synthetic generated data consist in a point cloud lying on a unitary-radius cylinder, centred at the origin and with its axis parallel to the z-axis. Such points are perturbed with two different types of noise: zero-mean Gaussian additive noise and salt and pepper noise. The former is controlled by its standard deviation σ, while the latter is regulated by p, that is the probability of a point to be an outlier, with σ_o (such that $\sigma_o > \sigma$) being the outliers' standard deviation. We also introduced an additional parameter θ, denoting the angle underlying the portion of observed surface (when $\theta = 360^\circ$ the cylinder surface is complete). This parameter is used to simulate a range-map acquisition, since real data usually include only a portion of the cylinder surface. We measured the algorithm precision in terms of three quantities, that are: radius relative error (in percentage), the angle and relative distance between the fitted axis and the ground truth.

7.3.1 Sensitivity Analysis

One of the strengths of our approach is that it only depends on the number of slicing planes used to generate axis candidates. In fact, other possible parameters (like RANSAC thresholds for ellipse fitting) depend only on the scale and on the density of data and are fixed at the beginning of the process. The number of random planes has a direct impact on the algorithm's performances, so in the first group of experiments we evaluated our fitting approach varying the nature of the slicing planes against different kinds of data. In particular, we are interested in analysing the effects of two factors: the number of planes involved and their inclination with respect to the actual cylinder's axis. Moreover, we evaluated the robustness and the stability of the algorithm in different noise conditions (low and high noise levels) and with several cylinder's surface coverages.

In the first experiment we generated N planes with random inclination and position, and run the inlier selection process on the same randomly-perturbed data increasing the value N from 10 to 90. We used synthetically generated data as described before, testing several coverage angles θ and simulating different noise conditions. In Figure 7.4 the error values are plotted against the increasing number of slicing planes (on x-axis, from 10 to 90), each curve denotes a different surface coverage angle. We measured the axis angle error (in degrees, first row), the axis translation error (second row) and the radius relative error (third row). The first column displays the results in a low noise setup ($\sigma = 0.005$,

$\sigma_o = 0.01$ and $p = 0.1$), while in the second column we show results obtained increasing the outlier parameters ($\sigma_o = 0.05$ and $p = 0.25$). Each configuration was tested 10 times and the error bars display the mean standard error.

As expected, a higher coverage angle implies a better ellipse fitting, producing a set of more reliable axis candidates. For this reason, in the lower noise scenario, the fitting precision increases for angles greater than $90°$ and become essentially the same for angles larger than $135°$, where the curves overlap. In the case of high noise levels we have basically the same behaviour: for greater values of θ we observe a very good cylinder fitting, that is comparable to the low-noise scenario. For smaller angles ($50°$ and $90°$) the fitting can be improved by adding more planes, but still the axis inclination error results the most critical parameter to resolve. Overall, we observe a slight improvement when we slice the scene with a higher number of planes, but the choice of this parameter does not result so critical in terms of fitting precision. The number of planes affects the robustness of the proposed algorithm, and thus should be adjusted depending on the features of the acquired scene. In the case of low or medium levels of noise we can safely set the parameter N from 30 to 60 to obtain acceptable outcomes. The results obtained on partially occluded surfaces are a good indication that, with a suitable number of planes, the algorithm offers good results with typical scanner range data. Of course, this parameter depends on the kind of application: if the point cloud contains several cylindrical primitives the number of planes should be kept relatively high in order to capture all the elements to extract.

In the previous experiments we generated several planes with a random center and normal in order to test the general case in which locations and orientations of the primitives to be extracted are unknown. Nevertheless, there are scenarios in which an approximate orientation of the objects to be detected is known. In such cases, it makes sense to exploit such prior information to extract better candidates.

For this reason we analysed the performances of our algorithm varying planes inclinations with respect to the cylinder axis. We fixed the number of slicing planes to 20, and $\sigma_o = 0.05$, with $p = 0.1$; then we performed two sets of experiments, for low ($\sigma = 0.001$) and very high ($\sigma = 0.1$) levels of additive noise. After generating a randomly perturbed cylinder, we run the fitting algorithm using planes with a random orientation which have been restricted to a small interval (from $0°$ to $10°$, from $10°$ to $20°$ and so on). The experiment was repeated 20 times for each configuration. In Figure 7.5 we show the three measured errors (one for each row) as boxplots for each different plane orientation interval. The first column displays results in the low noise setup, while the second column the results for very noisy conditions. Note that the angle values of intervals refer to the relative angle between the plane normal vector and the cylinder's axis, so a small angle implies the plane to be almost orthogonal to the cylinder's axis. In practical terms, a larger plane angle stretches out the ellipses to be fitted, increasing their eccentricity, while an angle close to zero implies more circular sections. Results show that in both cases (low and high noise) a larger angle improves the cylinder fitting precision, especially in terms of axis localization (rotation and translation). This happens because the ellipse fitting becomes less ambiguous in the case of higher eccentricity of sampled data, therefore the

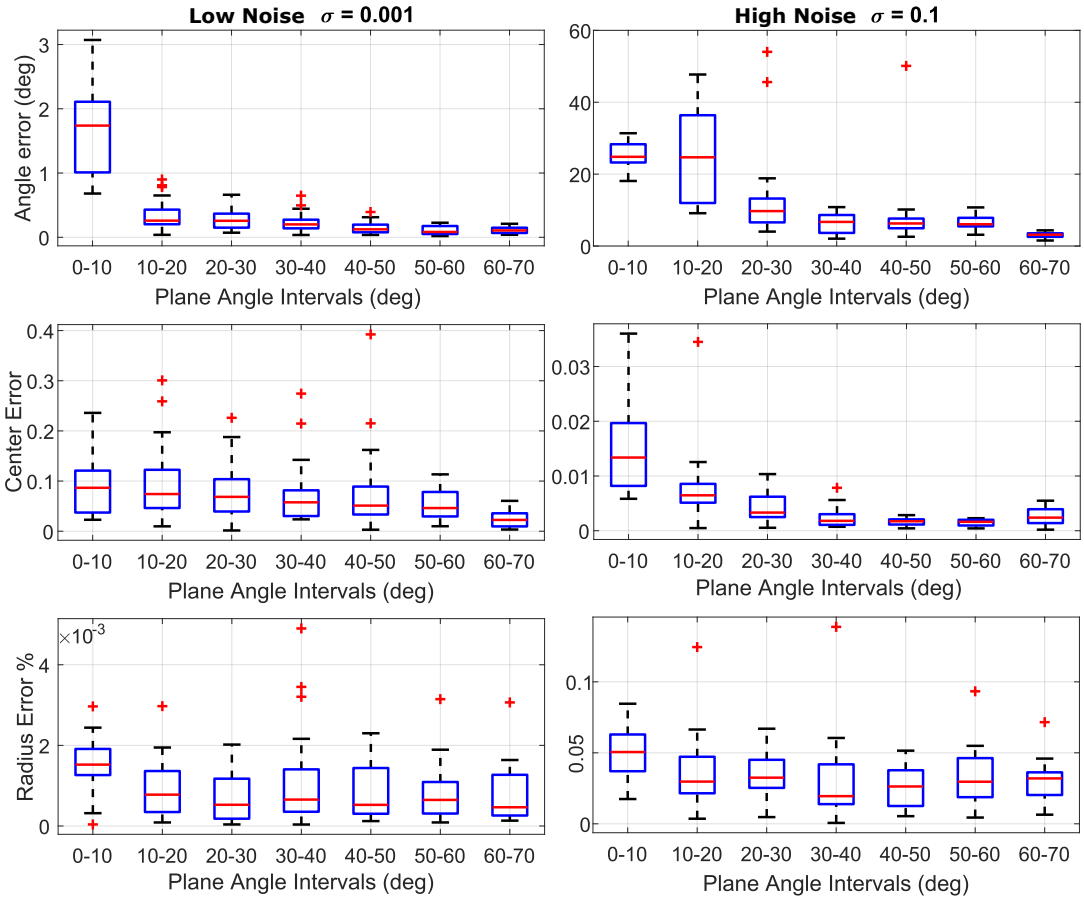


Figure 7.5: Fitting performances using planes at different inclinations, in the case of low or high noise levels (respectively 1st and 2nd column). The error is splitted into axis angle, center displacement and radius error. Each boxplot represents an orientation interval in terms of angle between the cylinder axis and plane normals.

computation of major and minor axes is more accurate. In fact, if we sample noisy points from a circle, a lot of ellipses with small eccentricity are equally acceptable results, but in practice they make the axis fitting more unstable. Conversely, the cylinder's radius is less affected by the planes inclination, but still we can observe small improvements for higher angles. This observation should be considered especially when extracting cylinders from a scene where some prior on their position is known and a high precision is required, even with noisy point clouds.

7.3.2 Comparisons

In the following set of experiments we focused on the performances of the proposed algorithm when compared with other cylinder fitting techniques, analysing different noise conditions. We generated the usual synthetic scene containing a unitary-radius cylinder to be fitted, and we applied our method against the techniques proposed in [209] and [171].

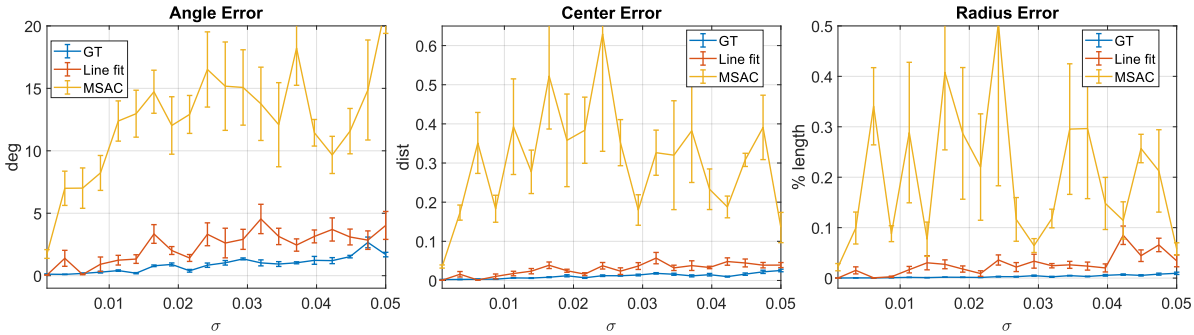


Figure 7.6: Angle, center and radius error means increasing the standard deviation of additive, zero-mean Gaussian noise. We tested three different cylinder fitting technique, each one denoted by a curve.

The first approach [209] consists in a sample consensus technique (MSAC) that also needs point normals to estimate the best cylinder fitting from the given point cloud. If vertex normals are not present, they are estimated by locally fitting a plane in the point's neighbourhood. We run such algorithm setting the maximum number of iterations equal to 5000 and the maximum inlier distance equal to 0.03. Conversely, the technique proposed in [171] exploits fitted ellipses as cylinder sections: the cylinder's axis is computed by fitting a line through the centres and the final radius is the average of all radii values. After that, the cylinder is refined by fitting through a non-linear optimization. To have fair comparisons, we used the same extracted ellipses as input to both our and the described techniques.

In the first test we gradually increased the standard deviation of additive Gaussian noise (from 0.001 to 0.05) and run all the fitting algorithms on perturbed data using 20 slicing planes. Salt and pepper noise was set with $\sigma_o = 0.04$ and $p = 0.3$. Figure 7.6 shows in different curves the error values for each fitting technique, increasing the level of noise (on x-axis). Each plot displays a different measured error, namely: (from left to right) axis angle, centre displacement and percentage of radius error. As usual, the bars denote standard error values and each experiment has been repeated 20 times. In the case of axis angle error, the proposed game-theoretical inlier selection offers more stable results and a better axis orientation precision with respect other approaches. When comparing axis dislocation and radius errors, our method and the line fitting approach are similar in terms of precision, while MSAC algorithm is always unstable. In general, the proposed method exhibits more accurate results in the case of Gaussian additive noise.

In the following set of experiments we compared the fitting performances in the case of different noise conditions. Figure 7.7 shows the behaviour of the three methods when the salt and pepper noise increases in terms of outliers frequency (left column) and magnitude (right column). We set $\sigma = 0.001$ and 20 random slicing planes, to measure the different fitting errors as in the previous test (from top to bottom: axis angle, axis distance and radius error). Data were perturbed 10 times for each configuration, and the error bars show the standard error. The leftmost column shows the results with a fixed $\sigma_o = 0.1$,

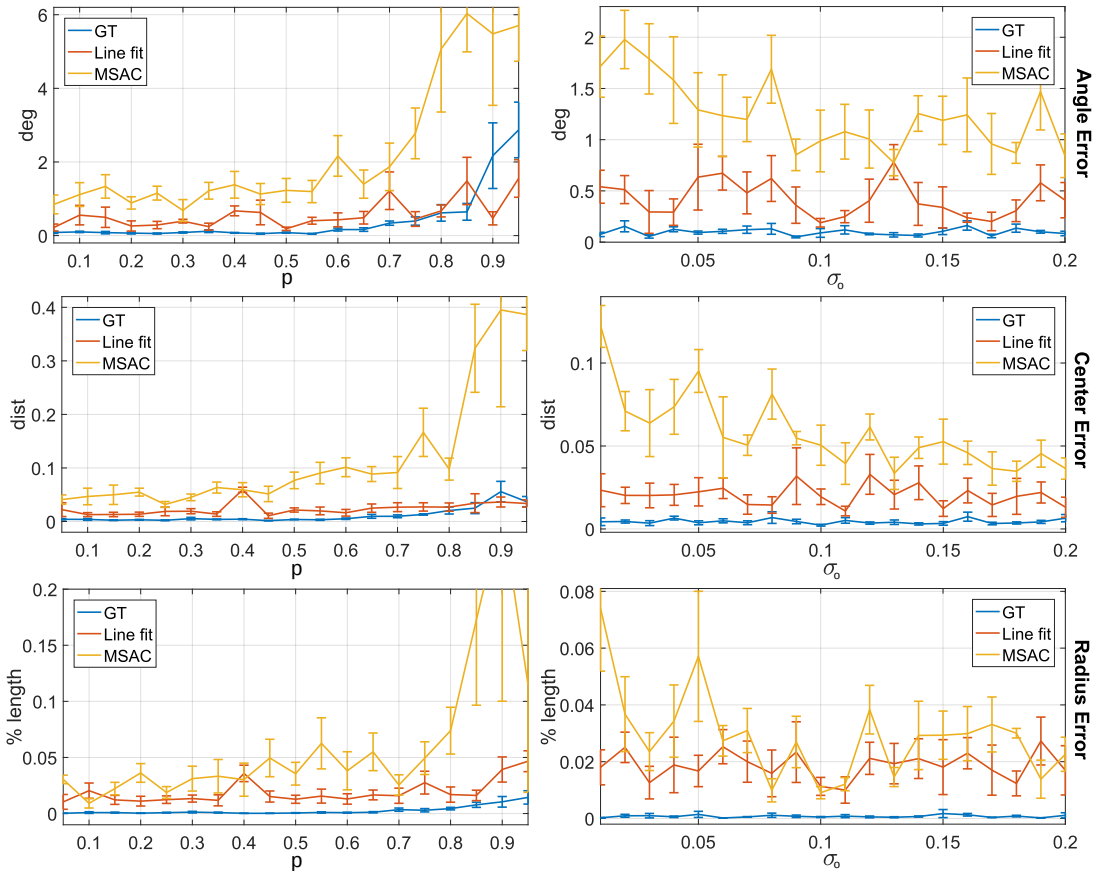


Figure 7.7: Angle, center and radius error means varying the salt and pepper noise parameters. In the first column the standard deviation σ_o of the outliers was gradually increased, fixing outlier probability as $p = 0.4$. In the second column we increased the outlier probability p and keep $\sigma_o = 0.1$;. We tested three different cylinder fitting techniques, each one denoted by a curve.

increasing the value of p (the probability of a point to be an outlier). In the rightmost plots we fixed the outlier probability $p = 0.4$ and gradually increased the outliers standard deviation σ_o from 0.001 to 0.2. In both cases the proposed fitting algorithm exhibits more accurate and stable results with respect to other techniques, even with a severe noise component.

Finally, we simulated a typical scenario with a partially occluded object; indeed, this situation is common in many applications, since most of the times the acquired data is a range-map or a laser scan line. We run the fitting algorithms for increasing cylinder's surface coverage angle θ. Results are shown in Figure 7.8, where each plot denotes a different error value, as in all the previous tests. Note that we displayed only our technique and the line fitting approach because MSAC errors were so high to hinder a good visualization of the first two. The additive Gaussian noise standard deviation was set to $\sigma = 0.005$, while the salt and pepper noise was characterized by a standard deviation $\sigma_o = 0.01$ and an

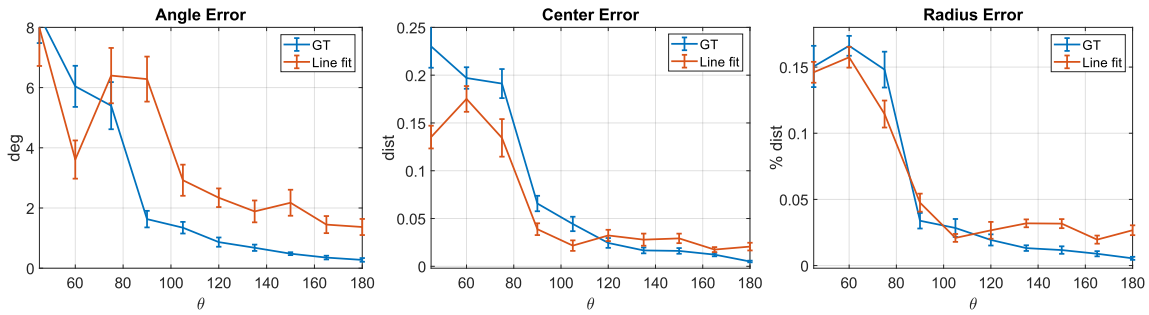


Figure 7.8: From left to right: angle, center and radius error means increasing the cylinder coverage angle θ.

outlier probability $p = 0.1$. We also increased the number of slicing planes (40, with any random inclination) to have more stable results. As we already noticed, the cylinder's axis angle is the most critical parameter to be computed: the leftmost plot of figure 7.8 shows that the proposed algorithm offers a better axis estimation with respect to the simple line fitting. In terms of axis dislocation and radius error, the proposed technique and the line fitting offer similar results; in some cases the line fitting algorithm exhibits slightly better results for angles smaller than 90°.

In conclusion, we observed that in general the proposed inlier selection gives more stable results and a better accuracy in terms of both axis and radius estimation. In some cases the results are slightly better with respect to the simple line fitting approach, but especially when we have situations with a large amount of outliers, the performances are significantly improved, as displayed in Figure 7.7. We also observed that the axis angle is the most critical element to be estimated, and our approach allows for the best set of candidates to be selected in order to filter out noisy data. The basic case of a single cylinder scene allowed us to analyse the precision of the fitting algorithm in an independent way from the primitive extraction ability, that will be addressed in the following part.

7.3.3 Real-world Experiments

In this experimental section we tested the multiple cylinder extraction capability of our algorithm employing real scan data to evaluate the performances in real-world scenarios. We adopted a well-known model dataset containing everyday items, that is the Yale-CMU-Berkeley Object and Model Set [35]; specifically we used 3d scan data presented in [34]. Such dataset includes scans of the objects from the Model Set, each one consisting in coloured 3d vertices with no orientation. We selected a subset of cylindrical items and composed them in several scenes from which some cylindrical primitives are to be extracted.

In the first experiment we analysed the repeatability and accuracy of our method when applied to real-world point clouds. We compared the proposed algorithm only with MSAC approach, that allows the extraction of multiple primitives from a generic point cloud The line fitting method [172] was excluded because it is designed to work with a single cylin-

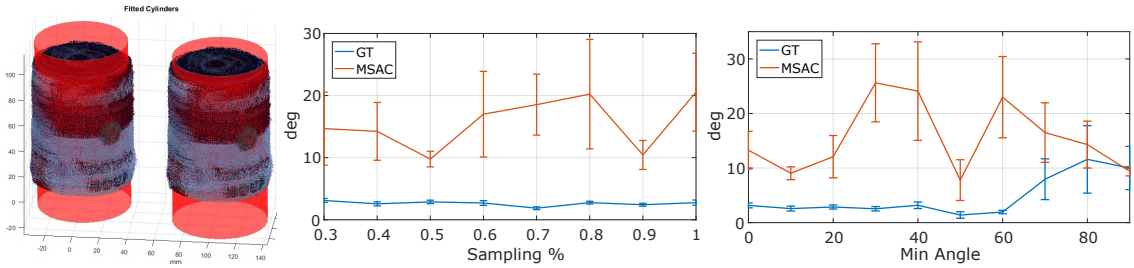


Figure 7.9: Left: an example of the generated scene with two scans of tomato soup object. Red cylinders have been fitted by our method. Center and right: relative angle means of the two fitted axes from the scene. The bars represent the standard error.

drical shape to be fitted, therefore its application in such scenario has no sense. Since MSAC approach requires vertex normals, we computed them for each point by locally fitting a plane on its 80 neighbours[3]. To extract multiple cylinders with MSAC approach, we iteratively removed all points falling inside the region delimited by $\pm 5\%$ of the estimated radius. Note that, while MSAC approach needs to run each time the input is modified, our algorithm performs the ellipses fitting just once at the beginning, then the extraction is performed by dynamically run the game-theoretical selection with a modified payoff matrix (as explained in 7.2.5).

We choose one single object (*"tomato soup"*), with radius equal to 33 mm and height equal to 101 mm. The point cloud contains 373,205 vertices; it has no outlier points but the surface is quite thick, probably due to scanner rangemap registration errors. We generated a simple scene containing two instances of the object, separated with a known random translation. In Figure 7.9 an example of such scene is displayed, together with the cylinders extracted by our algorithm (in red). Since the exact axis of the scanned object is not known, we evaluated the accuracy of the algorithms measuring the relative angle between the two estimated axes from the scene. Ideally, since the object was transformed with a pure translation, such angle should be almost zero. Plots in figure 7.9 display the relative angle values between the couple of extracted primitives using our technique and MSAC. In the leftmost plot we simulated various point cloud densities (from 30% to 100%) starting from the original scan; we plot the relative angle value against the percentage of randomly sampled points (in x-axis). In this case we used 40 slicing planes and we performed 4000 MSAC iterations, with minimum inlier distance equal to 2mm. The experiment was repeated 20 times and the same uniformly sampled data was used as input for both methods. In the rightmost plot of Figure 7.9 we simulated a partial scene acquisition (as in a range-map), varying the minimum angle between a fixed point of view and vertices normals. For this tests we removed all the points for which the relative angle between the normal and a fixed direction vector is below an increasing threshold (displayed on x-axis, note that min angle = 0 is the whole point cloud) and computed the relative angle between the two axes. In this setup we used 50 planes and increased

[3]We empirically determined that this value gives good results without being too slow in processing.

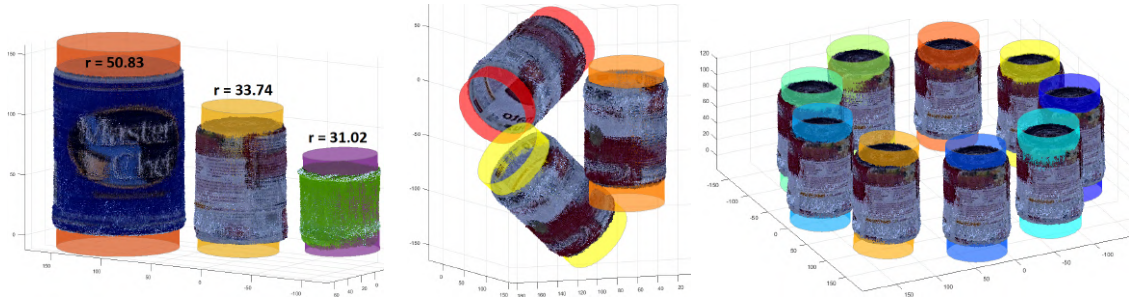


Figure 7.10: Qualitative examples obtained in different kinds of scenes, generated using real scans data.

MSAC iterations up to 5000. The view vector was $(0, 1, 0)^T$ and each test was performed 20 times with a random subsampling equal to 0.8. For both experiments we observed that our algorithm gave a more stable angle value ranging between 2 and 5 degrees, except for point clouds that have higher occlusion (with a minimum angle greater than 65 degrees). Moreover, the proposed approach is not influenced by variations in the data density. In general MSAC approach exhibited a more unstable behaviour and a greater standard error.

Finally, we performed some qualitative tests in various conditions to show the output of our algorithm in a general cluttered scene. Figure 7.10 displays a selection of generated point clouds with the extracted cylindrical shapes. We composed the leftmost scene with objects of different radius and size: *"tomato soup"* (radius 33 mm), *"Master Chef"* (radius 51 mm) and *"c cup"* (radius 32.5 mm). We used 50 random planes to extract the three cylinders and displayed the computed radii. Other configurations include objects with random rotations and a scene with several cylindrical objects to be extracted (Figure 7.10, centre and right respectively). In particular, for the latter we employed 60 planes and run 9 times the game-theoretical inlier selection.

Figure 7.11 shows two of the various scene that have been generated, with objects of various shape. We sliced such scene with 60 planes to extract the cylindrical primitives, that have been successfully localized with no further refinement. Experimental results show the effectiveness of our extraction method in different cluster conditions, effectively excluding other elements in the input point cloud.

7.4 Conclusions

In this Chapter we proposed a novel technique to extract multiple cylindrical primitives from generic, non-oriented point clouds. The method combines two crucial elements: the definition of an effective similarity function between couples of candidate cylinders based on dual quaternion algebra, and a candidate extraction based on geometrical properties of the scene. The game theoretical inlier selection exploits a specially designed payoff-function among cylinder candidates and allows the method to be general enough to be used without prior knowledge of the scene. Our approach exhibits a high robustness with

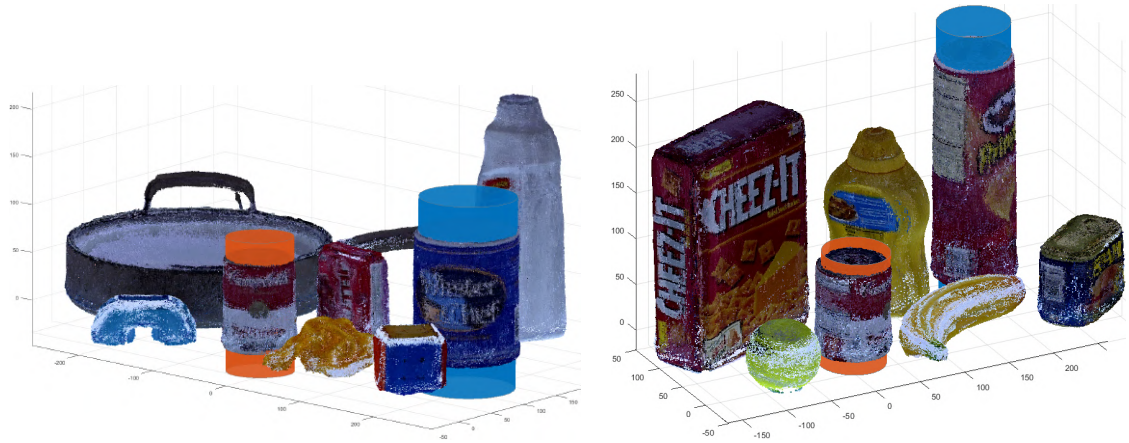


Figure 7.11: Cylindrical primitives extracted ”in the wild” from two difference scenes.

respect to outliers, that makes it perfect to be applied in practical scenarios where a noisy point cloud is acquired without normal vectors. Qualitative experiments show that the proposed extraction method offers high flexibility in very heterogeneous and complex scenes.

8

An Industrial Application: Microscopic Surface Reconstruction

Many practical applications of structured-light 3D reconstruction are targeted to industrial scenarios. In fact, the flexibility and the different accuracy levels offered by such techniques makes them easily adaptable in a number of applications with different purposes. This Chapter presents a specialized application in the field of microscopic surface measurement systems, with the focus on a specific case-of-study involving industrial gearwheel quality inspection. We employ an acquiring device setup composed by a fixed line pattern projector and a telecentric camera, both mounted on a rigid structure. We studied a new calibration approach employing a sphere with a known radius, observed in different locations. Moreover, we formulated an axis calibration approach that allows to locate the observed object orientation with respect to the imaging device, offering a convenient reference frame to carry out relevant measurements.

8.1 Introduction

Structured-light approaches allow the disambiguation of points on the scene to recover their depth via triangulation. The projected pattern can be static (so that it is acquired with a single shot) or mutable over time, involving multiple acquisitions for the same scene configuration. A classical example of the latter approach is of course the already described phase-shift method.

In the literature, fringe projection is widely employed in the context of microscopic applications, where the acquired object features' size are in the order of millimitres or microns. For instance, [25, 176] propose an object shape verification system using stereo correspondences between the acquired scene and a CAD model of the same artefact. Other works propose the use of phase-shift to perform microscopic profilometry [119] or for specific applications like reconstruction of electrical components [120] or small coins [126]. Even if phase shift offers high resolution and accuracy, it requires the acquisition of multiple frames, with the burden of synchronizing projector and camera, that is not always a feasible option. Other approaches avoid multiple acquisitions favouring fixed patterns like random distributions [45] or binary and coloured codes [237].

Despite the selected approach, all the mentioned applications require an initial calibration process. The aim is to assess the geometric relations between the imaging process and the components that are part of the system. Such calibration has a direct impact on the quality of the final reconstruction, particularly for industrial-related applications, where the accuracy level needs to be defined up to a certain confidence.

Novel Contribution

In this Chapter we describe a structured-light case-study based on telecentric fringe projection profilometry for microscopic measurements. Specifically, we combined a telecentric projector with a fixed stripes pattern (composed by equally spaced parallel lines) and a telecentric camera. Both camera and projector imaging systems are not affected by distortion, producing an orthographic projection model.

The contributions are twofold: first we propose a novel calibration solution involving the observation of a sphere with a known radius in different positions. The intersections of the spherical surface with the parallel planes generated by the projector pattern originate a series of ellipses on the camera image plane. We will show how such ellipses can be used to geometrically compute the system parameters and obtain the correct scale by fitting the 3D sphere. The second contribution of this Chapter consists in a specific case-of-study where the described sensor is employed. Such configuration is in fact used to acquire parts of the surface of industrial gearwheels, with the final aim of performing some measurements along the object's teeth. To perform such tasks the relative position of the acquired object with respect to the camera reference frame needs to be known. For this reason we propose a robust axis calibration that directly exploits some object features and a pipeline that allows for accurate surface reconstruction and analysis.

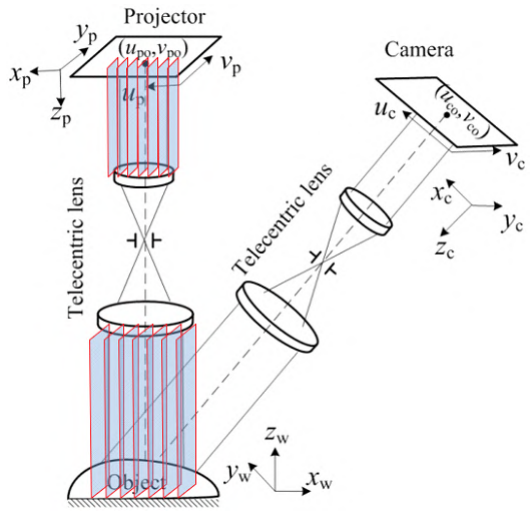


Figure 8.1: Schema of the acquisition device. Both projector and camera have telecentric lenses.

8.2 System Configuration

The choice of a telecentric camera for this kind of application is driven by two main reasons: first, they are characterized by a larger depth of field, second the projective imaging model is orthographic and hence simpler to calibrate. In contrast, the classical pinhole lenses offer a very narrow depth of field when used in microscopic applications, that can be problematic during the acquisition process, specially if used in conjunction with a projector.

An orthographic camera can be seen as a perspective one with its centre of projection located at infinity [79]. In this way, the rays coming from the captured objects in the scene intersect the image plane in an orthogonal way: this implies that the apparent object size on the camera image plane is not affected by its depth. As a consequence, the size of each pixel corresponds to a certain length in the real world, and therefore is straightforward to measure distances along x and y axes. On the other hand, the depth computation and reconstruction are more complicated.

A schematic configuration of the proposed acquisition system is shown in Figure 8.1. The projector casts on the scene a fixed pattern consisting in a series of equally spaced straight lines. As result of orthographic projection, such lines generate a group of equally spaced parallel planes in 3D space orthogonal to the projector's image plane. The planes orientation with respect to the camera coordinate system nor their relative distance in 3D space (called *stride*) is known a priori. Indeed, the stride value depends on the 2D distance of the lines on the projector's image plane, but it is influenced by a multiplicative unknown scale factor. Such scale depends on the combination of (i) the magnifying factor characterizing the telecentric lens and (ii) the inclination of the projector's focus plane (adjustable via a simple mechanism), that needs to be aligned with the camera image

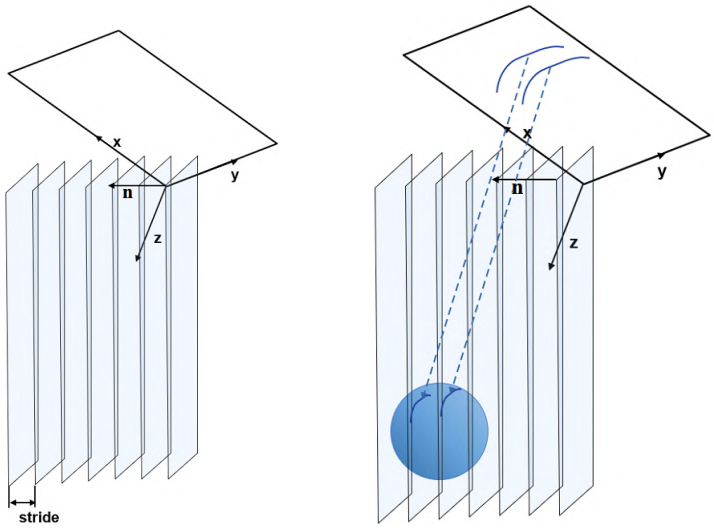


Figure 8.2: Left: the parallel planes coming from the projector (not in the picture) and the camera image plane. The normal vector $\mathbf{n}$ and the stride must be calibrated. Right: the target sphere and its intersection with the projected planes. The parallel projection of the intersection between sphere and plane is visualized as an ellipse on the camera image plane.

plane to obtain a better depth of field.

The projected planes are also identical, therefore is not possible to distinguish their absolute offset or numbering within the projector image plane. For this reason, we assume to only recognize their relative order (i.e. which one comes before another) but not their absolute index with respect to the first one.

8.2.1 Calibration Technique

In order to proceed with 3D reconstruction, some geometric parameters of the system must be computed. We identify three elements that need to be estimated by the calibration process (shown in Figure 8.2, left):

- the normal vector $\mathbf{n}$ of all the parallel planes with respect to the camera reference frame;
- the *stride*, which is the distance between each couple of adjacent planes (measured in camera pixels);
- the scale factor s between camera pixels and the real world (i.e. the pixel size in millimetres).

During the calibration process, a sphere with a known radius r is placed in the acquisition area of the device (with both camera and projector in focus) and an image is captured.

We assume that a subset of N parallel planes originated from the projector are cast on the sphere's surface. Their intersections with the sphere generates N circles in 3D space. This set of circles exhibits some particular properties: (i) they are are coaxial since all the slicing planes are parallel, and hence all their 3D centres lie on a line L in space. (ii) the line joining the centres is also orthogonal with the planes (i.e. parallel to their normal vector $\mathbf{n}$). Such circles are partially visible on the sphere's surface in correspondence of the projected light stripes. When the sphere is acquired on the camera image plane, the N circle arcs on the sphere surface are imaged as portions of 2D ellipses $E_1, ..., E_N$.

For each ellipse E_i with $i = 1, ..., N$, the planes normal vector $\mathbf{n}$ can be recovered up to four equally possible configurations. The recovery of the normal vector $\mathbf{n}$ can be converted into the computation of the orientation of axis L, that in turn corresponds to the axis of the cylinder passing through the i-th circle. The camera image plane (which normal vector is $\boldsymbol{\pi} = (0, 0, 1)^T$) intersects such cylinder, originating the observed ellipse E_i. Then, the ratio between the ellipse semi-major axis a_i and the semi-minor axis b_i is a function of the angle θ between the normal $\mathbf{n}$ and $\boldsymbol{\pi}$:

$$cos\,\theta = \frac{b_i}{a_i}. \tag{8.1}$$

Two possible axes orientations (up to a sign) can be computed, according to the following relation

$$\theta = \pm\, arccos \frac{||\mathbf{b}_i||}{||\mathbf{a}_i||} \tag{8.2}$$

where $\mathbf{a}_i$ and $\mathbf{b}_i$ are vector representations of the major and the minor semi-axes of ellipse E_i on the image plane. The angle θ is indeed the rotation amount to be applied to the image plane normal $\boldsymbol{\pi}$ around the minor axis $\mathbf{a}_i$ to obtain the orientation of $\mathbf{n}$ (corresponding to the line L). In this way, each ellipse E_i generates two equally possible normal vectors $\mathbf{n}'_i$ and $\mathbf{n}''_i$. Note that also $-\mathbf{n}'_i$ and $-\mathbf{n}''_i$ are valid solutions but can be ignored in the disambiguation process since they represent the same plane.

The 3D line L connecting all the circles' centres is projected on the image plane as the line l passing through the ellipses centres. The computation of the correct $\mathbf{n}$ exploits line l to filter out the wrong plane normals, recalling that the required plane normal is parallel to the axis L and that all normal vectors generated by the N ellipses must agree. Finally, the stride (in pixels) is obtained by computing the relative distances between the adjacent ellipses centres.

Once the stride and plane normal $\mathbf{n}$ are computed, the observed points belonging to the sphere surface cut by the planes can be triangulated by intersecting the 3D planes with the points observed on the image plane. Note that the geometry of the triangulated points is correct, but their exact depth is not known, because the planes are not distinguishable one from the other. This is indeed a fringe disambiguation problem, in fact the final reconstruction is up to an unknown integer shift (k times the stride value) along the plane normal direction.

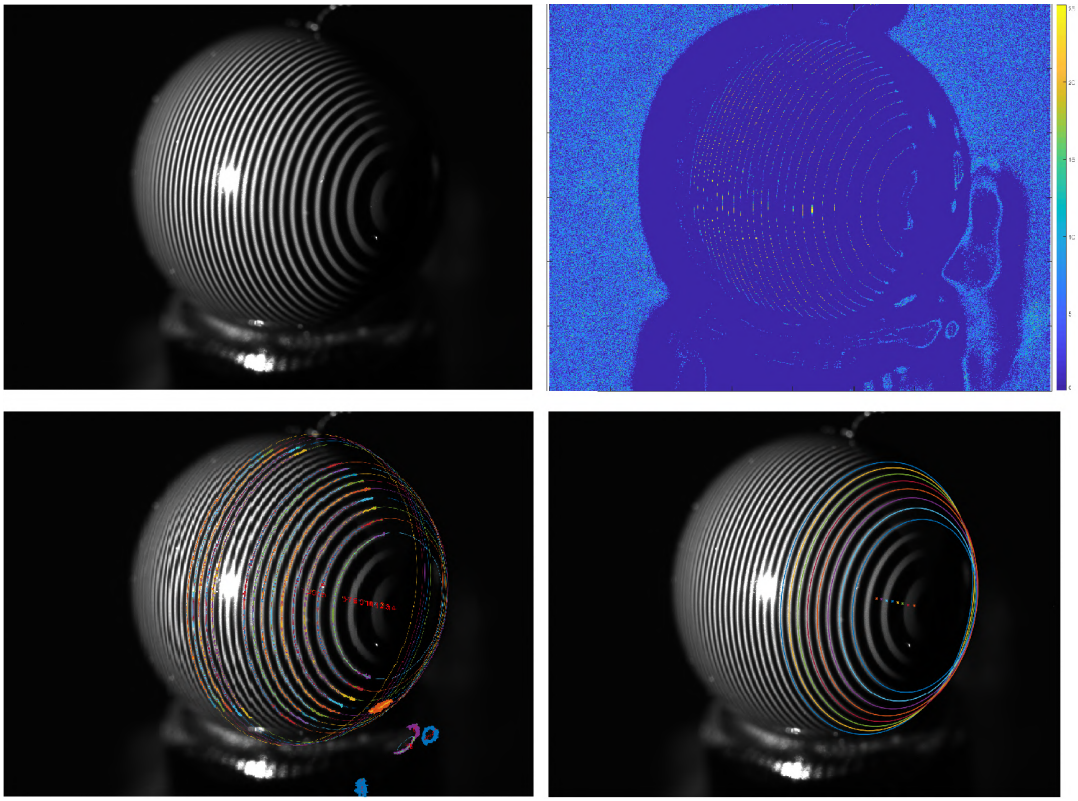


Figure 8.3: Image processing and inlier ellipses extraction. Top left: acquired image. Top right: skeleton accumulator from which the connected components are extracted. Bottom left: selected connected components and all fitted ellipses. Bottom right: inlier ellipses used for system calibration.

The 3D points are then fitted to a sphere model to obtain the radius in camera pixels. Since the sphere has a known radius, this value can be exploited to compute the scale between pixels and millimetres.

In order to obtain a robust parameter estimation, the same sphere is moved and acquired at different locations and the procedure is repeated for each acquisition. The final parameters are computed by averaging the resulting values, excluding the acquisitions for which the RMS of the distance between the fitted sphere and the associated 3D points is above a threshold.

8.3 Calibration Process

In our tests we employed a camera-projector pair, both equipped with orthographic lenses. The camera acquires greyscale images with a 12 Mpixels resolution, and we used a 0.05 mm thick line pattern on the projector. The devices were mounted on a solid structure with a relative angle of approximately $45°$, in a way that the plane of focus for both camera

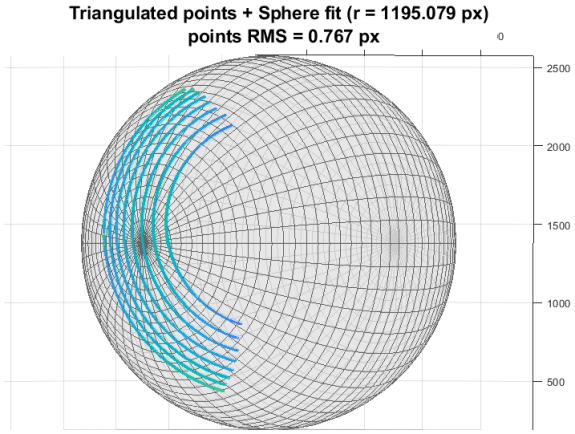


Figure 8.4: Triangulated ellipse points and fitted 3D sphere. The points are coloured with their distance from the sphere's surface and then final RMS is 0.76 pixels.

and projector are aligned to a common area of interest. The sphere used for calibration is a smooth ceramic object with a diameter of 8 mm and a declared surface tolerance of 0.7 microns.

After the acquisition, the resulting image is processed in order to extract (i) the sets of 2D points corresponding to the projected lines hitting the sphere's surface and (ii) the parameters of each captured ellipse containing the observed arcs.

Figure 8.3 (top-left) shows an example of acquired shot. Some ellipses are out of focus but most of them are clearly recognizable and exhibit a high contrast on the sphere's surface. The points extraction task was performed applying a series of image filters: first, five binary images were computed applying five different threshold levels to the original image. Then, for each binary image, its skeletonization is computed, following the approach described in [113]. The skeleton results are joined into an image accumulator, that is slightly dilated in order to join the isolated portions of each arc. Figure 8.3 (top-right) shows the final accumulator: note that higher values correspond to the ellipses that need to be extracted. Such accumulator is then filtered (values below a threshold are discarded) and the connected component extracted from it. Figure 8.3 (bottom-left) displays the extracted connected components: most of then actually belong to the sphere surface and are part of the ellipses that are to be extracted.

The second task to be addressed is ellipse fitting. An ellipse model is fitted against the set of 2D points belonging to each connected component. We adopted a RANSAC-based fitting algorithm, using the approach described in [75] and keeping only the primitives exhibiting at least 80% inliers among the point data, setting an inlier tolerance of 2 pixels. In this way all the extracted components that are not ellipses in the image are automatically discarded. Figure 8.3 (bottom-left) shows the computed ellipses superimposed to the extracted connected components. An additional inlier selection is then performed over the set of ellipses by fitting a line through their centres and checking their relative distance in pixels. The ellipses whose centers are not uniformly distributed along the line

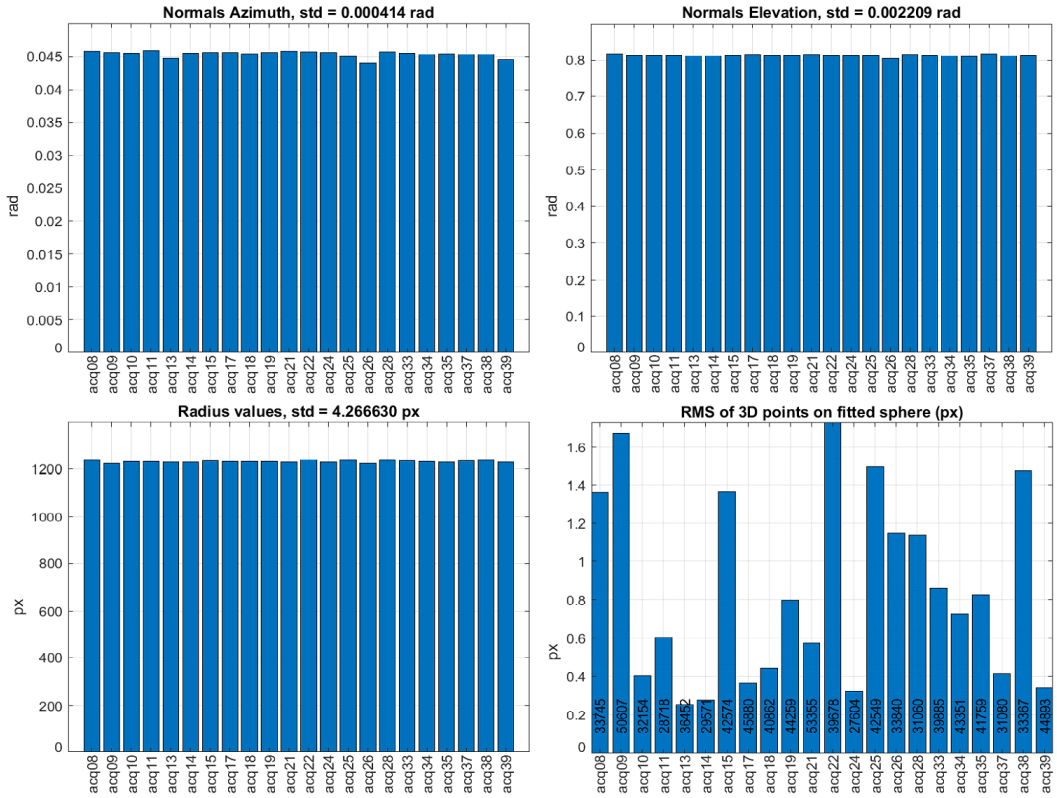


Figure 8.5: Final calibration values and their standard deviation. First row: plane normal (azimuth and elevation in polar coordinates). Second row: sphere fitting, radius values and fitting RMS.

are discarded: in Figure 8.3 (bottom-right) the inlier ellipses used to calibrate the system are displayed.

Finally, the selected ellipses are used to compute the plane normal and the stride value as described in the previous section. After that, the points belonging to the ellipses are triangulated and used to fit a 3D sphere in a least-squares sense and to fix the pixel scale. Figure 8.4 shows the triangulated points and the fitted sphere.

We performed 30 acquisitions of the same sphere in different positions with respect to the camera. The multiple acquisitions allowed the assessment of the repeatability of the proposed calibration method and the accuracy of acquired 3D points with respect to the fitted sphere. In order to obtain a robust parameter estimation we also selected 22 observations exhibiting a final RMS on the fitted sphere surface below a 2px threshold. Such acquisitions are finally averaged to obtain the final calibration parameters.

Barplots in Figure 8.5 (first row) show the estimated azimuth and elevation values of the vector $\mathbf{n}$ expressed in polar coordinates. The second row shows the measured radii for each acquisition and the RMS of distance of 3D points with respect to the fitted sphere model. All the measured values have a good stability: the plane normal is estimated with a

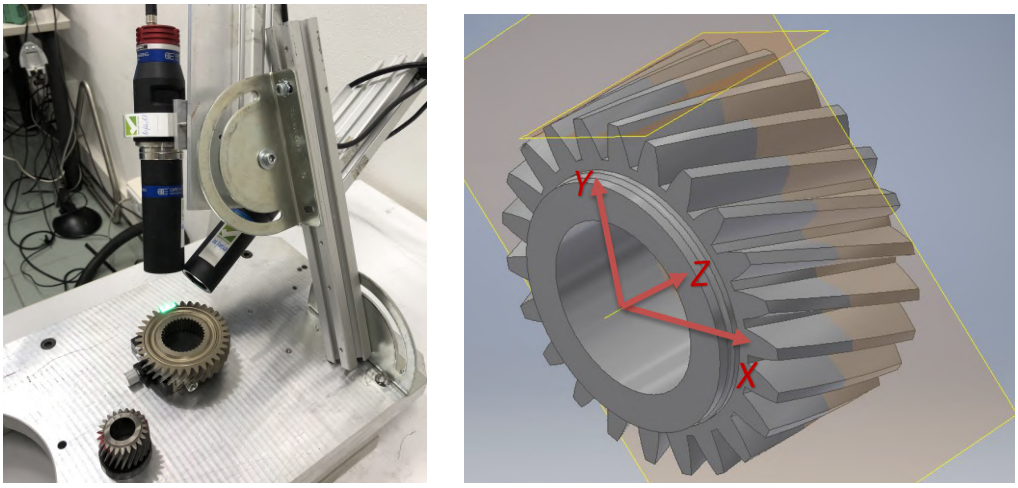


Figure 8.6: Left: acquisition setup. Right: gear coordinate system and the two cutting planes used to perform the tooth profile analysis.

standard deviation in the order of 10^{-4} radians while the sphere radius exhibits a standard deviation of 4.27 pixel camera. Considering that the computed scale s is $3.24 \cdot 10^{-3}$ mm/px, the standard deviation is around 13 microns. The RMS over the sphere surface ranges from 0.3 to 1.7 pixels, corresponding to 5 pixels on average.

8.4 Case-of-Study: Gearwheel Surface Analysis

This section presents a specific application of the described measuring system in an industrial setup. In particular, we are interested in acquiring the shape profile of the internal gearwheel surface during the final steps of their production. Figure 8.6 (right) shows an example of the specific kind of gears analysed in this application. These objects are employed in equipments of various fields (both industrial or not) and often the requirements on the manufacturing precision is high, since the gear's teeth need to perfectly fit the shape of other components to provide the best energy transmission efficiency.

The adopted acquisition setup for gear measurements is the same described in the previous calibration section (i.e. camera and projector with telecentric lenses) and it is shown in Figure 8.6, left.

8.4.1 Acquisition and Image Processing

As we did for the calibration sphere, we established a sequence of image filters to be applied to the acquired images so that the line pattern profiles are accurately extracted and triangulated. In general, the gear surface appears as a semi-shiny material, characterized by diffused micro-scratches that alter the light reflection and are captured as shown in Figure 8.7, left. Surface roughness causes the line edges to be indented (clearly visible

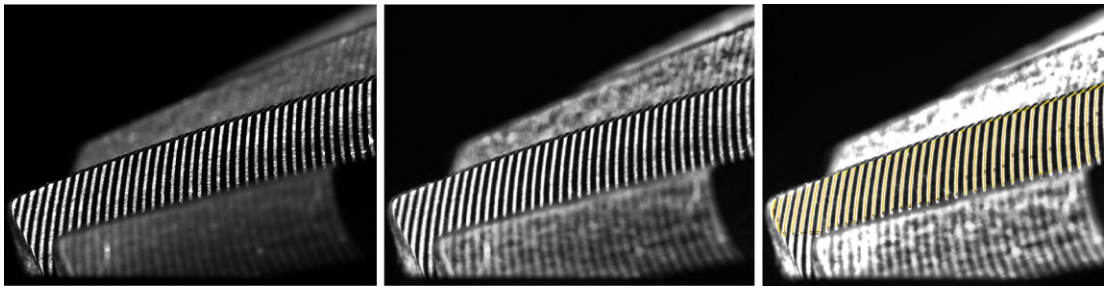


Figure 8.7: From left to right: original acquired image, after pre-processing filters, after curve detection.

in Figure 8.7, left), and thus could possibly lead to a wrong median line extraction. The pre-processing task has the purpose to uniform the projected stripes edges along their direction and have a better extraction of the curves lying on the tooth surface.

The same scene is first acquired with five different camera exposure time. Different exposures cause a blooming effect on the image, but assuming to acquire locally planar areas the effect is symmetrical on both edges of the projected stripes, so it does not affect the median curve extraction.

Each image is then processed applying the following filters:

- Contrast Limited Adaptive Histogram Equalization (CLAHE);
- morphological operators to reduce spikes;
- Bilateral filtering.

The result of image pre-processing task is shown in Figure 8.7, center. In the processed image the edges of the projected lines appear more sharp and visible, thus their median curve can be detected easily.

After the pre-processing task, the projected curves are extracted following a procedure similar to the ellipse arc extraction on the sphere surface. For each image an adaptive binary threshold is computed and the skeleton is extracted, then all skeletonizations are joined in an accumulator which is appropriately filtered to obtain the final connected components belonging to the desired curves. In figure 8.7 (right) the extracted connected components are highlighted. For each connected component, a spline curve is fitted and the adjacent curves are detected in order to number each subsequent plane. Finally the fitted splines are triangulated to obtain a 3D point cloud.

8.4.2 Axis Calibration

During the gear examination process, different teeth surfaces are sequentially acquired while the gear rotates around its axis. The object's axis corresponds to the axis of the cylinder in which the gear is inscribed. Such line in 3D space defines (up to an arbitrary rotation around it) a reference system for the object itself, thus it is used to define the

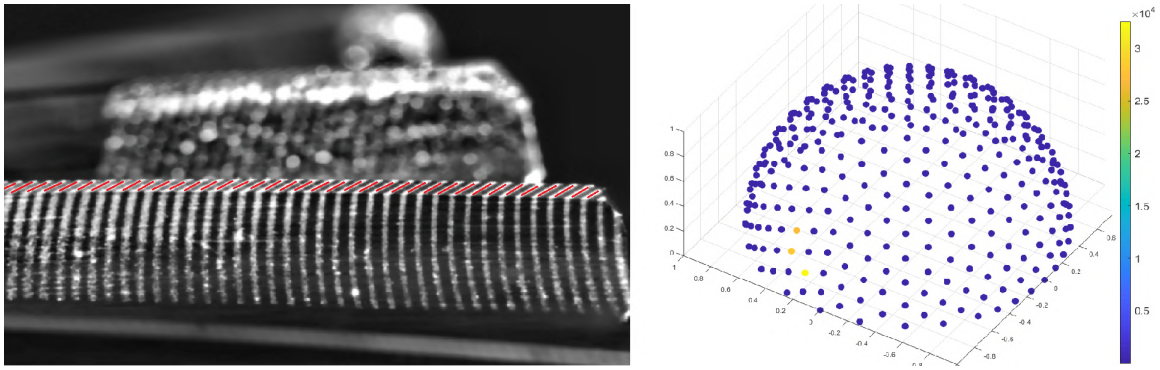


Figure 8.8: Gear axis calibration. Left: extracted points belonging to the cylinder surface. Right: spherical axis accumulator, all directions but three collected zero inliers.

profiles we need to compute over the teeth surfaces. For this reason, to perform such measurements, the first task consists in gear axis calibration. Figure 8.6 (right) describes the gear reference system and highlights two cutting planes defined in such coordinates. The two planes are respectively parallel to xy and xz planes and are used to produce the required teeth profiles as intersections with the acquired surface. Since such planes are defined in axis coordinate system, an accurate measurement is a direct consequence of a precise localization of the described rotation axis. Thus, its position with respect to the camera reference frame must be accurately calibrated.

The proposed calibration method assumes to work with circular gears, i.e. objects that can be inscribed in a cylinder of known radius such that all the points on the exterior part of the teeth belong to the cylinder's surface. To perform axis calibration we first take multiple acquisitions of the same tooth, applying each time a small rotation on the gear spanning the space volume in which the object remains in focus. The final part of the observed tooth belongs to the surface of the cylinder that we need to fit to recover its axis, so such points are isolated and triangulated. Figure 8.8 (left) highlights the described surface and the relative extracted curves in red. At the end of this process, we obtain a 3D point cloud containing a set of points lying on the cylinder surface.

To compute the cylinder axis, we propose an approach divided into two steps: first, the most plausible axis orientation is computed employing a spherical accumulator, then the outliers points are excluded and the final orientation is refined through a robust technique, fitting a cylinder model over acquired data.

The aim of the first step is to identify the best axis estimation over a discretized set of directions. We define a semi-spherical accumulator containing 337 bins, uniformly distributed over a unitary sphere surface. Each vector in this accumulator is interpreted as the normal vector of an associated plane in 3D space. It is easy to verify that, as the plane normal vector gets parallel to the cylinder axis, the projection of the 3D points on such plane is more likely to fit a circle. In fact, in absence of noise, each cylinder section orthogonal to its axis is a circle with the same cylinder radius.

For each vector the previously extracted points belonging to the cylinder surface are

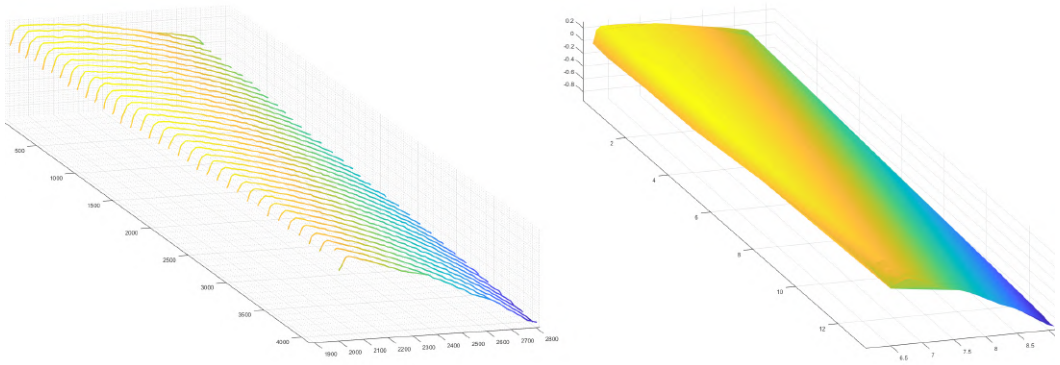


Figure 8.9: Final tooth reconstruction. Left: 3D curves from the triangulated splines. Right: final surface after registration. In both plots the colour denotes the z coordinate to highlight the curvature.

projected on the plane, obtaining a set of 2D points. Then a RANSAC-based approach is applied to fit a circle with a known radius and the inlier points (within a tolerance) are counted and associated to the corresponding accumulator element. The bin which obtained more votes is considered the best orthogonal cutting plane, and thus its normal vector is the best initial estimation for the axis orientation. Moreover, inlier points which are coherent with the cylindrical model are selected and used in the refinement step. Figure 8.8 (right) shows the final accumulator values over the unitary sphere surface: most bins exhibit almost zero inliers, while three elements are clearly close to the required cylinder's axis.

The second step in the axis calibration procedure involves the refinement of the cylinder axis. The first estimation obtained from the accumulator approach is used as initial value in a non-linear optimization task, minimizing the sum of the squared distances of each point from the axis. The result of such optimization is the final gear axis orientation.

8.4.3 Surface Reconstruction

The final teeth reconstruction is carried out by first triangulating the fitted splines (as in Figure 8.7, after image pre-processing) detected on the image. Each spline is uniformly resampled and intersected with the corresponding plane exiting the projector to generate a 3D curve (Figure 8.9, left).

To obtain the complete surface, adjacent curves are then registered in order to compute point-to-point correspondences between them. In details, each curve is first projected on its containing plane, then point-to-point correspondences are assessed through ICP (Iterative Closest Point) algorithm [27], allowing to compute the best rigid transformation between each couple of curves. Each correspondence is then linearly interpolated to fill the point cloud. After that, Delaunay 2D triangulation [44] is applied to join the 2D points projections on the camera image plane and obtain a dense triangulated surface. An example of the final triangulated surface is shown in Figure 8.9 (right).

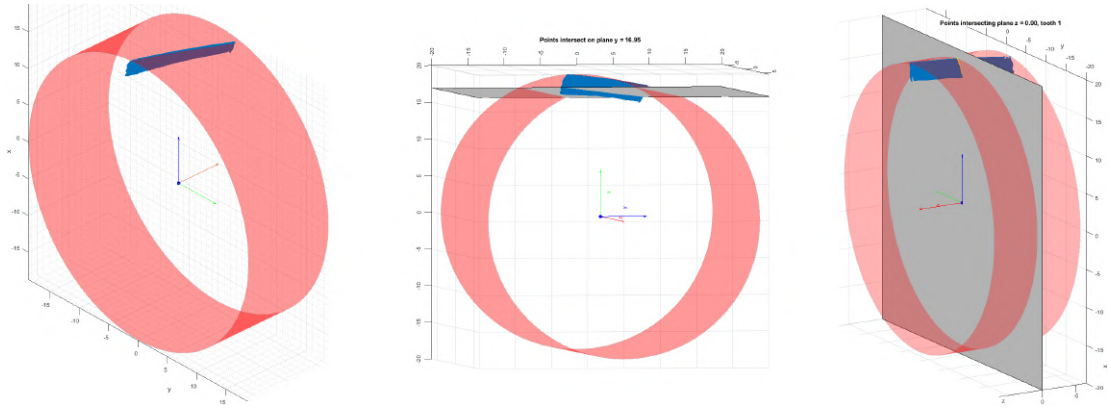


Figure 8.10: Right: tooth surface and fitted cylinder. Centre and right: surface together with two intersecting cutting planes used to obtain the two profiles.

Tooth Measurements

Once the gear axis is properly calibrated and the surface is acquired, the tooth profile measurements can be carried out by simply moving the 3D surface in the axis reference frame. Figure 8.10 (left) shows the acquired data of a single tooth together with the calibrated cylinder containing the gear. Then, two profile measures are performed by intersecting the computed surface with two different planes defined in the axis coordinate system. The intersecting planes are displayed in Figure 8.10: they are parallel to yz-plane (centre) and xy-plane (right).

The acquisition is performed for different teeth on the same gear and the computed profiles are compared and plotted in Figure 8.11.

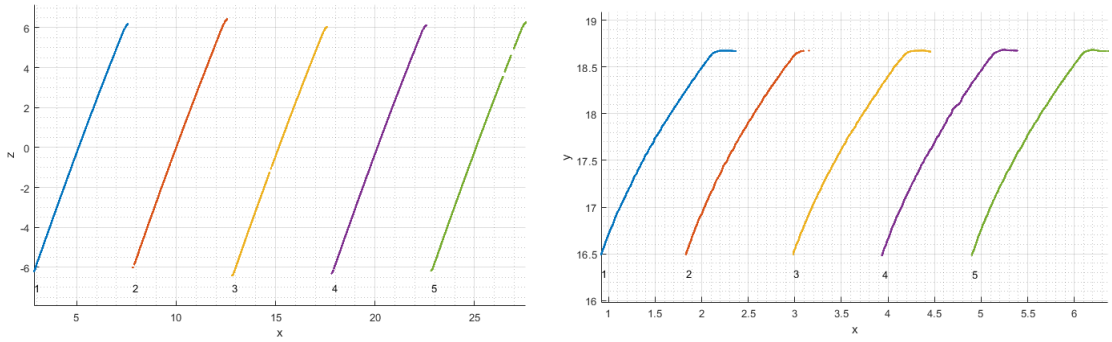


Figure 8.11: Final comparison of teeth profiles obtained intersecting the surface with two planes.

8.5 Conclusions

In this Chapter we proposed a microscopic surface 3D acquisition and reconstruction, with a specific industrial application involving gear's teeth measurements and quality control. The described system is composed by a camera-projector pair, both characterized by orthographic projection. We proposed a calibration process using a spherical object to retrieve the geometrical characteristics of the system with a precision in the order of microns. Moreover, we propose a robust gear axis calibration that is used in junction with the acquisition system to perform some profile measurements over the teeth's surface.

9

A Cultural Heritage Application: the Euporus Sundial

The sundial of Euporus was discovered in 1878 in the Circus of Aquileia (Italy), in an unusual location: the centre of the horse race track. Some studies have tried to demonstrate that Euporus sundial had been built for a more southern location then the one it was found at, although no specific alternative positions have been suggested. This Chapter first presents the workflow designed to fully 3D digitalize such artefact. The final reconstruction offers an accuracy in the order of the millimetre and thus gives the opportunity to analyse small details of its surface and to perform non-trivial measurements. Moreover, we propose a mathematical approach to compute the object's optimal working latitude as well as the gnomon position and orientation. The algorithm is designed as an optimization problem where the sundial's inscriptions and the Sun positions during daytime are considered to obtain the optimal configuration. The complete 3D model of the object is used to get all the geometrical information needed and to validate the results of computations.

9.1 Introduction

Digitalisation and archiving play a fundamental role in cultural heritage field [137, 198]. Indeed, nowadays there are plenty of technological applications specifically designed to support such tasks, allowing for fast and precise results which were usually obtained through manual or analogue tools [146, 183]. In particular, 3D reconstruction offers a wide range of opportunities in terms of recording artefacts geometry, opening new research directions in terms of conservation, restoration and study [122, 158]. The reasons behind such applications are several: digital libraries of high-resolution 3D models bring to the creation of extensive archives, that can be available to a large audience of users, both in research and public communities. Digitalisation also allows for restoration and monitoring of artworks which are often exposed to atmospheric agents. In particular, structured-light scanning techniques have already been employed in some cultural heritage applications, leading to optimal results [8, 9, 193].

This Chapter describes a practical case study showing how structured-light scanning and phase-shift in particular can be applied in a specific cultural heritage application. We explore two main aspects regarding the study of an ancient sundial: the employed digitalization method and the reverse engineering approach determining the gnomon's shape and working latitude. The first part of the Chapter deals with the 3D scanning process preliminary to functional analysis of the sundial, and focuses on the methodological aspects of its high-resolution acquisition aimed at capturing the object's geometry and all the meaningful details of its inscription. For the first time a complete 3D digitisation is attempted on this sundial, opening new possibilities in the analysis of its historical background and providing insights into the level of ancient knowledge of sundial design and construction principles. The complete 3D model enables to perform many non-trivial measurements without physically accessing the object, like the calculation of its overall volume, the planarity of the dial top surface (which affects the shadow curves casted by its gnomon) and the technique used to chisel the details of its inscriptions.

The second part of the Chapter presents the reverse engineering technique adopted to reconstruct the original gnomon's shape. This study first exploits the 3D model to acquire precise measurements on the sundial's surface and create a "synthetic" model of its inscriptions. Then an optimization process is carried out to simultaneously compute the best gnomon configuration and the optimal working latitude of the sundial.

9.2 Artefact Characteristics

The National Archaeological Museum of Aquileia (Italy) hosts a rare type of Karst limestone horizontal plane sundial (see Figure 9.1), known as the *"sundial of Euporus"* from the name of its donor, M. Anstitius Euporus, inscribed within the dial. This object pertains to the Vitruvian type called *"plinthium sive lacunar"* [109], that is an horizontal slab surrounded by a frame, recalling the form of an overturned coffered ceiling [14]. The

Figure 9.1: Left: Inscriptions on the top surface of the Euporus sundial. Right: 3D acquisition of the sundial: on the left the structured light scanner mounted on a tripod.

instrument was likely engraved in the 2nd century AD and was discovered in 1878 in the area of the Roman Circus of Aquileia, in centre of the horse race track [101], probably not in its original position.

The artefact structure may be considered as resembling a 2 m long by 1 m wide table (see Figure 9.1, right) placed on top of two $\sim$ 70 cm high cylindrical columns. The top planar surface measures 100×206 cm, and it is surrounded by a $\sim$ 10 cm frame extending all around its rectangular shape.

On one half of the upper surface a set of inscriptions reveals its usage as sundial (Figure 9.1, left), while on the opposite side of the plane there are three large cracks, repaired in antiquity with three lead patches. Around the main table, on the left and bottom side at a distance of about 30 cm, additional stone blocks are present probably acting as benches.

In this work only the acquisition of the main table was performed, excluding the two columns supporting the structure and the lateral benches. The digitisation was therefore greatly simplified because, as explained in the following sections, the process consists in multiple acquisitions of small partially-overlapping portions of the whole surface. Considering the geometrical constraints and self-occlusions of the specific object's shape, the acquisition of in-between the benches and the supporting columns with the same scanning device would have been impractical. Moreover, the current positioning of all the stone blocks around the main structure is not faithful with respect to the original positioning at a Roman age, so the acquisition of all such structure into a single model is not relevant.

9.3 3D Acquisition Process

The sundial digitisation into a 3D model was accomplished using our custom 3D structured-light scanner (see Chapter 6 for details). Such acquisition device enables to quickly and efficiently modifying its features to one's needs, moreover the complete (software and hardware) device control allows an effortless reconfiguration of its optical characteristics

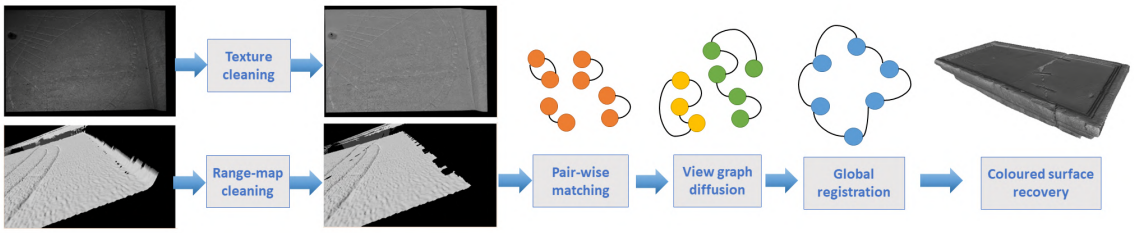


Figure 9.2: Schematic representation of the performed 3D digitisation pipeline.

and the reconstruction of algorithms involved. During the process, the hardware configuration can be adapted to capture different areas of the scanned object at different levels of precision and resolution. We employed multi-period phase shift technique and the unwrapping method described in Chapter 3. After the phase unwrapping step, the spatial 3D position of each observed point is then obtained by triangulating the projector and camera exiting rays, according to the recovered projector's codes. Additionally, the signal amplitude is used to get a high-resolution texture of the surface, i.e it captures the brightness intensity variations of the acquired object surface like in a standard grayscale photograph.

Each scanner acquisition generates a surface composed of a set of connected triangles, with their 3D vertices (the triangulated points) and the corresponding texture map coming from the signal amplitude. This 3D surface is called *range-map*, which is conceptually equivalent to a 3D photograph capturing both the optical and geometrical properties of the surface from a specific point of view.

Considering the object shape and size, the 3D digitisation of the whole artefact cannot be performed with a single acquisition. In fact, the area recorded by the scanner is not large enough to cover the entire object at a reasonable resolution and the convex nature of the object self-shadows the back side of each acquired area. The common procedure involves the acquisition of many overlapping portions of the object from different points of view in order to collect a set of "3D surface patches", corresponding to each acquired range-map. In this way, the entire artefact can be recorded with a high resolution, especially in the engraved sundial zone. The following section describes the specifically-designed pipeline applied in order to obtain the final reconstruction.

9.3.1 Reconstruction Pipeline

For each different point of view, the scanner produces a high-resolution intensity image (texture) and the 3D triangulated surface (range-map) defining the geometrical structure of each observed point. After multiple acquisitions, a sequence of six subsequent operations was designed to reconstruct the final complete coloured surface. Figure 9.2 presents a schematic sequence of the tasks performed during such pipeline. The main tasks are the following: texture and range-map *pre-cleaning*, *pairwise matching*, *view graph diffusion*, *global registration* and textured surface computation. In those operations supervision is

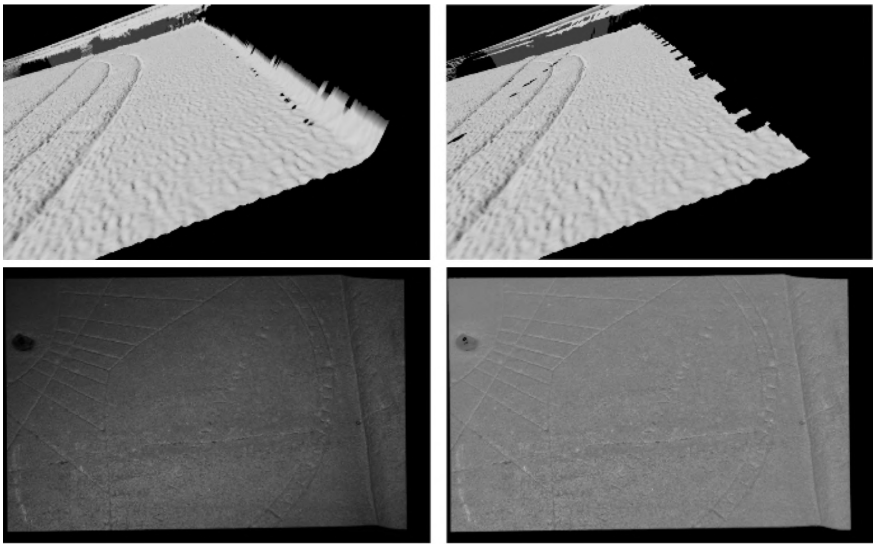


Figure 9.3: Range-map and texture pre-cleaning step before (left) and after (right) applying the described filters.

almost unnecessary: user intervention is limited to the optional configuration of different task-specific parameters. After data pre-processing, the most challenging part of the pipeline consists in merging all the individual range-maps. This procedure is usually referred as *registration* and it has been splitted in two main steps. First, a pair-wise registration operates on every range-map pair to find the best possible rotation and translation that aligns the two of them. Second, all pairwise transformations are averaged into a common consistent 3D space. The latter operation is called global registration, which is the only semi-automatic part the pipeline presented, requiring a human intervention just in its last refinement. Finally, the entire model is recovered by means of an automated algorithm which computes the surface given the registered points and normals.

Texture and 3D Pre-cleaning

The first two tasks entail a preliminary cleaning phase, in which range-maps and textures are processed independently to filter out noisy elements and improve the overall photometric quality of data, which has a direct impact on the final reconstruction accuracy. Regarding the range-maps, the initial cleaning process performs the following steps:

1. the triangles close to the range-map border are deleted;
2. all the connected components of the range-map are identified considering the topological connectivity of vertices induced by triangles (two vertices are adjacent if they share at least one triangle). Then, such components are ordered by cardinality and removed if the number of vertices is below a certain threshold (1000 vertices);
3. per-vertex surface normals are computed by averaging the orientation of each triangle insisting on the same vertex.

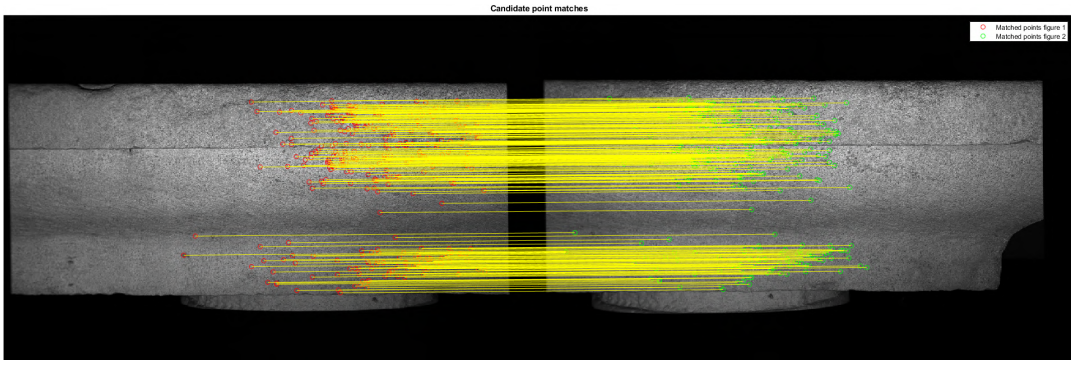


Figure 9.4: Example of image feature matching on two subsequent views. To simplify the visualization, here a small subsample of the actually computed matches is displayed.

The combination of the aforementioned steps improves the quality of the range-maps, which are typically prone to exhibit errors near the boundaries of the illuminated area. A visual example of the effect of the range-map cleaning process is shown in the top row of Figure 9.3.

For what concerns the acquired textures, we faced the problem of normalizing the uneven illumination of the scanned area due to the relative angle and distance between the light source and each surface point. Indeed, since the acquired texture is generated from the amplitude of the projected sinusoid, it is related not only to the intrinsic surface albedo but also to the actual light energy provided by the projector to each object point. In other words, surface areas that are far away from the projector's centre will appear darker than the other ones, with a decay that is typically exponential with respect to the distance. The same effect happens with the local surface angle, since slanted faces distribute the received light energy to a relatively bigger area compared to the faces orthogonal to the projector's optical axis. The effect is visible in Figure 9.3 (bottom-left image). Since the planar surface is angled, we observe an illumination gradient spanning the texture extent from top (darker) to bottom (lighter). To correct this phenomenon, we normalized each texture by means of an high-pass filter designed to remove the low-frequency light variations and thus preserve the high-frequency details of the texture, which are in our interests. This operation was performed with a non-linear top-hat operation using a disc-shaped structuring element of size 51×51. In Figure 9.3 (bottom-right) we show an example of resulting image texture after the correction. We can clearly see that now the texture image shows a uniform shading along its extent, yet maintaining its salient characteristics, such as inscriptions or surface marks.

When all the images have been processed, they can be put near each other (entirely or partially) with no evidence of texture borders or abnormal changes in light. For this reason, image normalisation represents a fundamental step in the pipeline presented, since it allows for a natural range-maps fusion and a surface representation as accurate as possible.

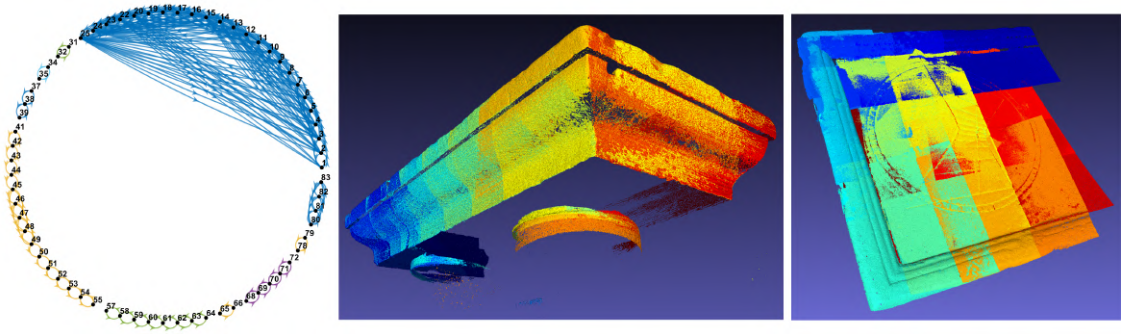


Figure 9.5: Left: the complete view graph. Each connected component is denoted by a different colour. Centre and right: the registered range-maps (after diffusion) from the two largest connected components. The yellow component (nodes from 41 to 55, connected as a chain) forms part of the outer frame; while the blue one (from 1 to 25) connects all the range-maps acquired in the engraved area. Each single view is displayed with a different colour.

Pair-wise Registration

Given the feature-rich optical nature of the object surface, the pair-wise registration step was implemented by matching relevant image feature points between every couple of views. The approach proposed in [132] was followed to provide a set of point-to-point correspondences between each couple of textures, exploiting well-known SIFT features. An example of feature matching is shown in Figure 9.4, where the correspondences between two subsequent acquisitions are shown. It is easy to verify that the majority of such matches is correct, as they connect the same points across different overlapping textures.

Each range-map assign a 3D point coordinate to each 2D point located on the texture image, therefore the computed feature matching produces two sets of corresponding 3D points in space. Since they were acquired separately, such point clouds belong to two independent reference frames. Nevertheless, the correspondences can be used to compute the relative transformation which allows to align the two range-maps. Such transformation is a composition of a rotation and a translation, computed as described in [82]. Since some possible matching errors would significantly affect the alignment precision, a RANSAC-based algorithm was adopted, allowing to select the actual consistent 3D matches and thus computing the best transformation between point clouds. The approach consists in two tasks: first, the alignment is computed by randomly selecting a subset of data points (in this case, 5 points is the minimum required to compute a 3D roto-translation); second, the remaining points express a consensus vote to assess the precision of the computed transformation [65].

The quality of the alignment is finally measured in terms of number of inliers, i.e. couples of 3D points for which distance after the transformation is below a threshold (1 mm for this application). This technique ensured that each registration between couples of range-maps had an acceptable level of precision. All the pairwise transformations which

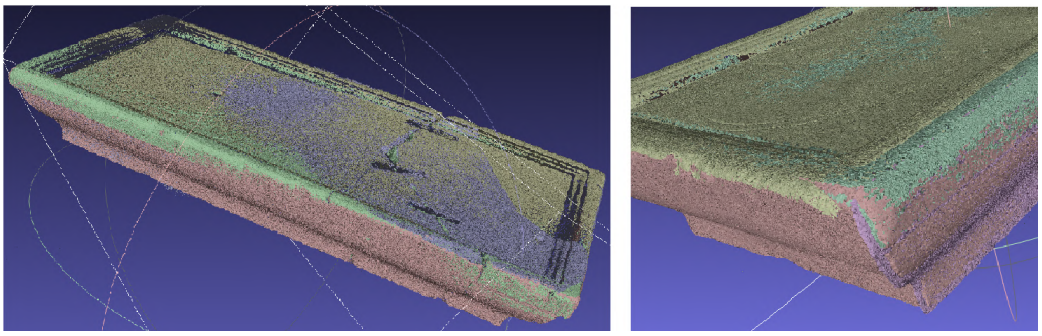


Figure 9.6: Global registration results after applying ICP algorithm. Each connected component is visualized with a different colour.

exhibit at least 20 inlier points were considered valid to build the so-called view graph, described in the next paragraph.

View-Graph Diffusion

After pairwise registration, a view graph is used to group together the computed transformations. The nodes represent the single acquired range-maps, while each edge connecting two nodes denotes the computed transformation between such views. In other words, the complete graph shows all the overlapping views of the object. Once the graph is built, it is possible to identify some subsets of nodes which are internally linked to each other by at least one path. Since an edge denotes a transformation between two views, each connected component of the graph represents a group of overlapping range-maps that can be merged to create a portion of the whole object.

In Figure 9.5 (left) the complete view graph is displayed: each connected component is identified by a different colour. Within a connected component, different paths between two views would result in two inconsistent transformations: this because the computed transformations are still subject to small errors. In order to generate a consistent portion of the object from each connected component, a state-of-the-art diffusion technique was applied [210] allowing for the computation of a consistent set of transformation within each group of connected views. This step also ensures the overall error minimisation for each set. After the diffusion, the single range-maps belonging to a connected component are merged to form a macro-section of the sundial. In Figure 9.5 (centre and right) the merged range-maps coming from the two largest components are shown.

Global Registration and Surface Reconstruction

The global registration step involves the new range-maps coming from each connected component of the graph. First, they are merged in the same scene and a first rough registration is performed in a semi-automatic way by manually selecting some correspondence points. Then, to refine the alignment among all the sections, all the individual components

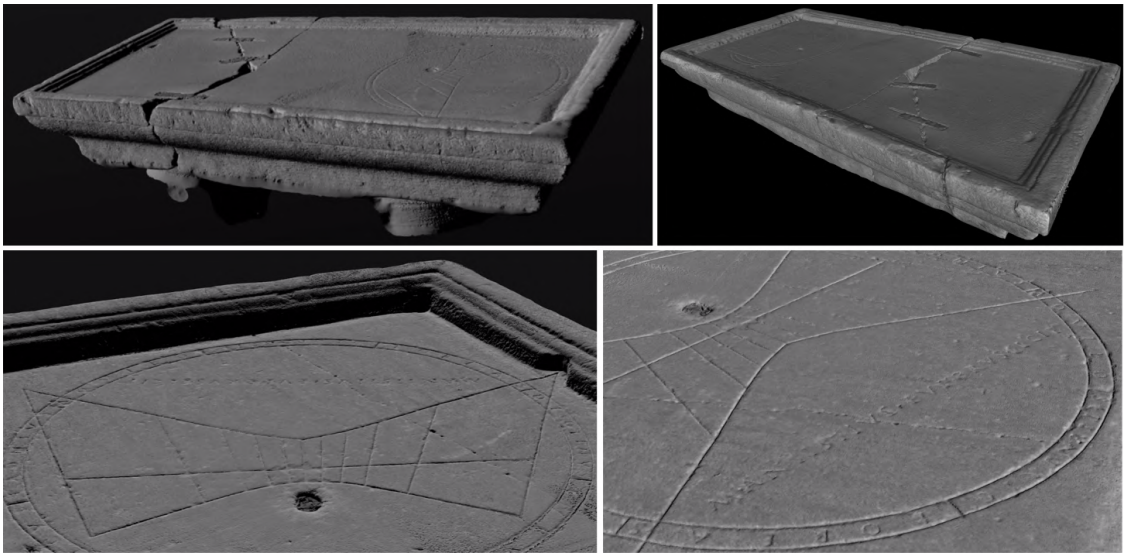


Figure 9.7: Some views of the final coloured model after global registration and Poisson surface reconstruction. Top: the whole scanned artefact. Bottom: details of the sundial engraving.

are registered using ICP (Iterative Closest Point) algorithm [231]. Such technique is designed to minimise the distance between two point clouds, and thus compute the best alignment. One range-map is selected to be the target and each of its point is associated with the closest point in the other cloud; then the transformation which reduces the overall distance between closest points is computed and applied to the second range-map. Such association and transformation is repeated until a required precision is reached. This approach was adopted only in this last phase because, unlike RANSAC, it would be more sensitive to outliers (that we already filtered out through the graph). Moreover, it requires a quite good initial alignment, which is infeasible to perform for each range-map pair. The results of the global registration of the point clouds is shown in Figure 9.6, where each connected component is depicted with different colour.

Starting from the registered point clouds, the final surface of the sundial was computed. Since the acquired points in the range-maps come with their normal vectors, the Screened Poisson Reconstruction method [100] was adopted. This algorithm exploits points and their normals to obtain a closed watertight triangulated surface which interpolates all the reconstructed points. The output is the set of coloured points and triangles reconstructing the entire acquired object.

9.3.2 Reconstruction Results

Since the most relevant part of the sundial is the engraved zone, the acquisition process was planned to capture such area with a higher accuracy level with respect to the rest. Therefore, the acquisition of the whole artefact was performed in two separate sessions.

First, the scanner was configured to acquire an area of approximately 50x40cm, at a distance of 150 cm in order to obtain a high-resolution reconstruction of the main inscriptions of the sundial, exhibiting a thickness in the order of some millimetres. The second session entailed a change of the scanner optics in order to acquire a wider area (approximately 150x200 cm) at a distance of 250 cm. This configuration allowed a good trade-off between the extent of the scanned area and the overall resolution of the acquired point cloud to easily capture the entire artefact.

All acquired views were planned so that the overlapping portion of each range-map with its successive was at least $\frac{1}{3}$ of the total area. The scanner was moved around the artefact to acquire the upper portion of the sundial in a single connected component and the lower sides on another. In this way, the recorded surface was kept as parallel as possible to the scanner image plane to ensure that all the points lie in the depth of field of the scanner and ensure a sharp texture recovery.

During the first session 40 range-maps equally distributed on the left, right and top side of the sundial were acquired. Among them, the first 25 were acquired at 30cm from the surface and the last 15 at 45cm in order to help the global registration of each view. During the second session, 45 range-maps were acquired all around the whole artefact with the exception of an area positioned on the short side of the sundial due to the structure of the sundial itself (placed on two large cylindrical marble supports and surrounded on three sides by unremovable marble benches) that prevented the placement of the scanner in a feasible position for appropriate recording. For each range-map, an average of 6.5 million triangles composed from $\sim$5 million 3D points were acquired. After the global registration, each range-map was resampled with a variable number of points, based on the number of views for each connected component. The goal was to obtain a roughly uniform distribution of the 3D points among all the connected components. Finally, the surface was created using the screened Poisson reconstruction technique with a tree depth set to 11 and 3 samples per node. Figure 9.7 shows the model of the scanned artefact in its entirety (top row) and some details of the sundial engraving (bottom row). The overall resolution, in particular for the engraved portion, is high enough to allow the analysis not only of the whole artefact, but also of the micro-furrows, scratches and unevenness of surface.

9.4 Gnomon Reverse Engineering

After the description of the digitalization process for the entire artefact, we move the discussion on the original purpose of the sundial. It is a common fact that, from ancient times, sundials were manufactured with the purpose of measuring time exploiting the path traced by the sun in the sky during the day. Usually sundials employ an object to cast a shadow on a surface where some marks indicate the time or the current part of the day. In particular, ancient sundials worked by dividing daylight time (i.e. from sunrise to sunset) into equal parts. Daylight has a variable duration, depending on the day and the latitude, so each ”time slot” does not have the same duration throughout the year. On the other

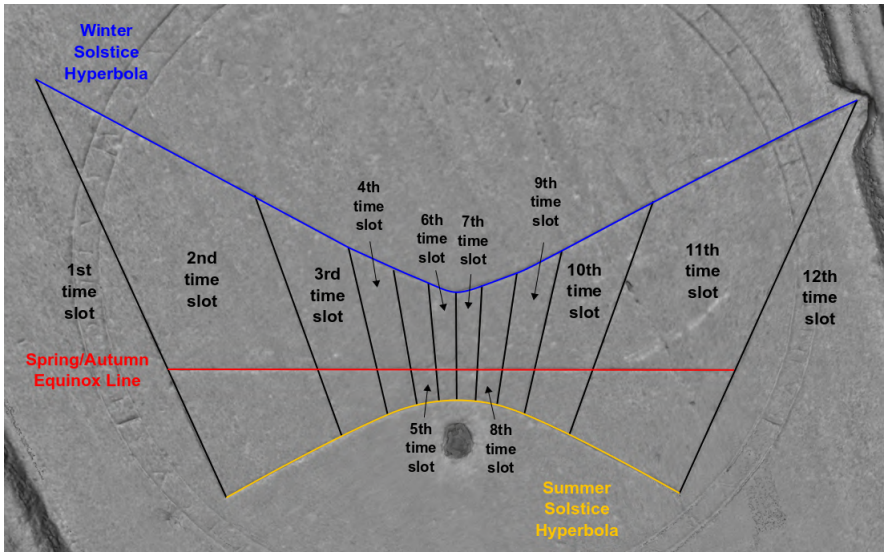


Figure 9.8: Time slots displayed by the sundial, each one denotes an equal subdivision of daylight. Solstice hyperbole and Equinox line are also highlighted.

hand in this way the sundial quadrant construction is made more easy and the system does not need longitude correction, since it does not point the ”clock time” as we are used today.

In this section we describe the optimization process specifically designed to recover the original gnomon’s shape and inclination, together with the optimal latitude in which the sundial should be used.

9.4.1 Model Acquisition

The initial phase of this study consisted in the identification of the salient points laying on the sundial’s surface. The engraved part of the object is characterized by a circle which contains the eleven segments used to mark the time during the day, together with the base of the gnomon (see Figure 9.8). The time marks are not perfectly symmetric with respect to the central line (the sixth), that corresponds to the south-north axis. The segments’ endpoints are connected by two curves, which are the solstice hyperbole (blue and yellow curves in Figure 9.8). Such curves indicate the path traced by the shadow of the gnomon’s tip during two specific days in a year: the Winter and Summer solstices, which are respectively days with the shortest and the longest light-time. The horizontal line between the two hyperbole is the Equinox line, which is traced by the shadow during two days, in Autumn and Spring (red line in Figure 9.8). During equinoxes, light and night times have the same duration.

It is evident that a correct localization of the marks on the sundial’s plane implies a more reliable and precise estimation of both the latitude and gnomon’s inclination. We exploited the 3D reconstruction of the artefact for two main reasons: first, its precision

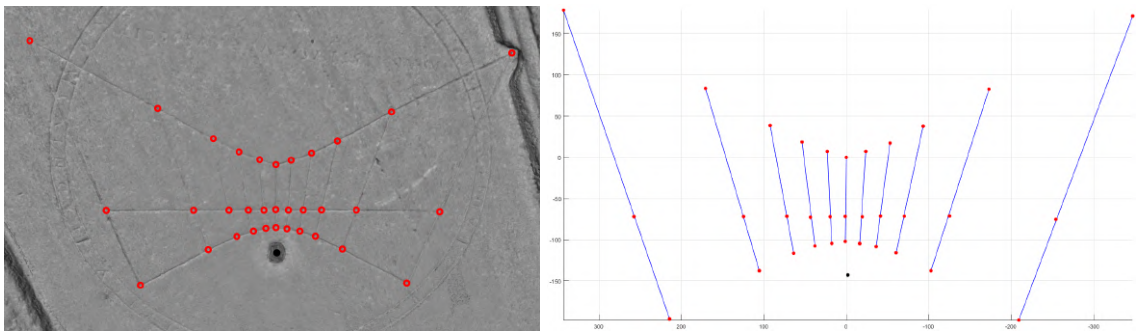


Figure 9.9: On the left, the 3D points acquired from the scanned model; on the right, the corresponding normalized 2D points used for optimization. All measures are displayed in mm.

reaches the millimetre and thus offers an excellent point localization free from errors; second, it allows the recovering of non-trivial measures, such as the object's planarity.

We recovered the coordinates of 34 points in three-dimensional space, marking the intersections between equinox and solstices curves with the 11 time marks of the day. Figure 9.9 (left) shows a detail of the 3D model and the extracted points: three for each time mark (in red) plus the gnomon's base (in black).

The acquired points were normalized to a convenient reference frame. Specifically, a plane was fitted such that the distances from each point to the plane are minimized. Then, the world's coordinate frame is moved on a *geographical plane* such that the y-axis (coinciding with the sundial's main axis) faces north, x-axis faces east and z-axis is oriented upward. The applied transformations do not affect the final result; this because the sundial's axis must be aligned with the North-South axis to work properly.

The measured points are almost coplanar, in fact the maximum distance from the fitted plane is $0.8mm$. Considering this accuracy, we can assume the points to lay on the plane (so that their z coordinate is zero) without loss of precision in the following computations. The result of such normalization is shown in the rightmost part of Figure 9.9.

After the normalization, the point coordinates on the sundial's plane (apart from the gnomon's base) are arranged in a 88-elements column vector:

$$V = (\mathbf{v}_1^1, \mathbf{v}_2^1, \ldots, \mathbf{v}_{11}^1, \mathbf{v}_1^2, \ldots, \mathbf{v}_{11}^2, \mathbf{v}_1^3, \ldots, \mathbf{v}_{11}^3, \mathbf{v}_1^4 \ldots, \mathbf{v}_{11}^4)^T \tag{9.1}$$

where each element $\mathbf{v}_i^j$ is a 2-dimensional row vector which indicates the coordinates of a point on the sundial's plane corresponding to time mark $i = 1, ..., 11$ in the day j (equinox or solstice). They are ordered as follows: Spring Equinox ($\mathbf{v}_i^1$), Summer Solstice ($\mathbf{v}_i^2$), Autumn Equinox ($\mathbf{v}_i^3$) and Winter solstice ($\mathbf{v}_i^4$). Note that elements $\mathbf{v}_i^1$ and $\mathbf{v}_i^3$ are identical, since the projections during the two equinoxes must correspond.

9.4.2 Sun Position and Projection

In order to introduce the notation that will be used in the following parts, we define the following sets:

- $Y \subset \mathbb{Z}$: set of years. It is formed by integers such that negative values represent BC years, and positive AD years.
- Δ: set of dates. Each element $d \in \Delta$ indicates a day in the format (year, month, day).
- T: set of timestamps. Each element $t \in T$ encodes a date and a time, accurate up to seconds. A timestamp can be interpreted as a sequence of values of the kind (year, month, day, hour, minutes, seconds).
- $L = [-90, 90]$: set of latitudes (in degrees). Negative values indicate southern latitudes.

We suppose to have a function which returns the Sun's position given a latitude and a timestamp:

$$S(t, l) : T \times L \to \mathbb{S}^2 \tag{9.2}$$

Where $\mathbb{S}^2$ is the set of 3D vectors belonging to the surface of a unitary sphere. The result of such function is a unit vector which points at the Sun's position during timestamp t and latitude l. To compute the correct Sun's position given a place, time and a date, we used the open-source library Pvlib [199], implementing the Solar Position Algorithm [173]. Such algorithm ensures a precision of ± 0.0003 degrees between the years 2000 BC and 6000 AD.

We fixed the longitude value given to the algorithm in all the computations. This choice has been made for two main reasons: first, as we already discussed, the sundial's marks do not indicate the "clock" time, but the current fraction of light-time (which has been equally splitted). For this reason we are only interested in the light time of each day, and not in the real clock time as we were calibrating a clock. Given a place and a date on Earth, we just need to compute the 12 time slots, without considering the so-called apparent solar time. Consequently, we have no need of longitude corrections and thus our formulation can be restricted to latitude only. Second, the geographic area involved in the research is quite narrow in terms of longitude, but it exhibits a wider latitude range. Considering that the optimal working location of the object is extremely sensitive to latitude changes (especially for what concerns the hyperbole curves and equinox line), in this application the longitude can be considered irrelevant.

Once the Sun's position is determined, the projection of a generic 3D point on the sundial's plane has to be defined. This defines a projective system in which the Sun is the centre, located at infinity, emitting parallel light rays to be projected on the sundial's

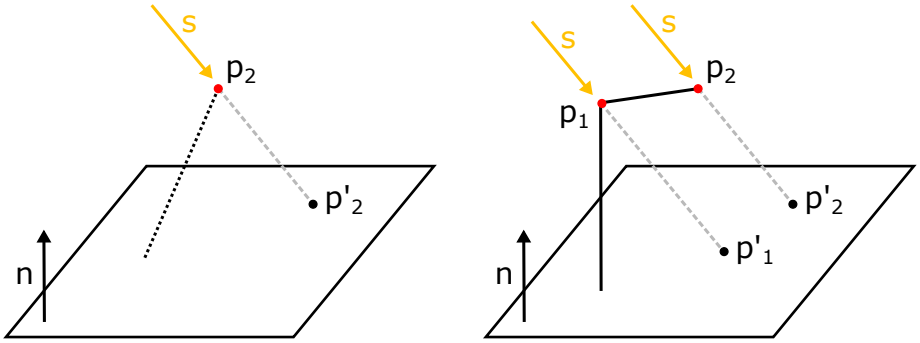


Figure 9.10: Left: punctual projection, with unknown gnomon configuration. This setup is sufficient to estimate p_2 coordinates as well as the optimal latitude. Right: presumed gnomon configuration. Points p_1 and p_2 are projected on sundial's plane, forming a segment.

plane. Therefore, the projection of any point into the sundial's surface is obtained as a parallel projection [79]. The resulting projection $\mathbf{p}'$ of a 3D point $\mathbf{p}$ on a plane is computed as follows:

$$\mathbf{p}' = \begin{pmatrix} \mathbf{r}_1^T \\ \mathbf{r}_2^T \end{pmatrix} \left(\mathbf{p} + \frac{-\mathbf{p} \cdot \mathbf{n}}{\mathbf{n} \cdot \mathbf{s}} \mathbf{s} \right) \tag{9.3}$$

where $\mathbf{s}$ is the vector pointing at Sun's position computed as in (9.2), and $\mathbf{n}$ is a unitary vector representing the normal direction of the plane (i.e. perpendicular to it) in which the shadow is casted. Finally, $\begin{pmatrix} \mathbf{r}_1^T \\ \mathbf{r}_2^T \\ \mathbf{n}^T \end{pmatrix}$ is the rotation matrix transforming the sundial's plane to our geographical plane. Note that we used the first two rows of the matrix so that the final projected point is a 2D point.

From now on, we model the gnomon through its two junction points, characterizing the shadow's length and slope. The end of the vertical part will be identified as $\mathbf{p}_1$, while the endpoint of the horizontal part is $\mathbf{p}_2$. Figure 9.10 displays a schematic representation of the projection process. In particular, points and vectors involved in Equation 9.3 are included (the apostrophe denotes the projection of the point).

9.4.3 Energy Function Minimization

Since the sundial works with 12 equally-spaced time slots during daylight (instead of absolute solar time), we define a utility function H such that $H(d, l) : \Delta \times L \rightarrow T^{11}$. Given a date d and a latitude l, such function computes the 11 timestamps which should correspond to the marks on sundial's plane. Note that each timestamp marks the transition from one slot to the following one, as sketched in Figure 9.8. For example, the first mark on the sundial denotes the end of the first slot and the beginning of the second, and so on.

We estimate the sundial parameters so that the gnomon's shadow is projected as close as possible to the corresponding marks at the timestamps returned by function H.

We divided the operation in two independent tasks:

- the joint optimization of p_2 and the latitude;
- the computation of p_1 and thus the inclination of the gnomon.

This choice is driven by the following observations: the sundial's working latitude is only affected by the projection of point p_2 during equinoxes and solstices. In fact, the endpoint of the shadow casted by the gnomon must fall exactly on that specific curves in four specific days in a year. In this terms, we can forget the gnomon's shape and consider only a punctual projection of p_2 onto the plane (see Figure 9.10, left). During the four relevant days that we mentioned, such projection must overlap with the corresponding curve.

Once the point p_2 is fixed, and hence the latitude is recovered, the shadow's inclination allows to guess a possible shape for the whole gnomon. In particular, we assume a single joint in correspondence of point p_1. This task could be affected by the thickness of the stick, and requires an entirely different criterion to assess the correctness of the projection.

Latitude Optimization

For the sundial to work we need the shadow of the gnomon's tip (identified as p_2) to occur in some predefined points during four days every year. Each of these days is associated with a curve and, in turn, each curve is marked with eleven points (not equally spaced), which identify the time slots in which the day has been divided.

Solstices and equinoxes days slightly change each year, thus we define a utility function $\hat{D}(y)$ which, given a year y, returns four dates from Δ corresponding to equinoxes and solstices. These dates are computed as follows: the Summer and Winter solstices are the dates in which the daylight is respectively maximum and minimum (20-22 June and 20-23 December), while the two equinoxes are days in which daylight is equal to night time (19-21 March and 21-24 September).

The function $P(y, l, \mathbf{p}_2, \mathbf{n})$ computes a 88-elements column vector containing the projections of 3D point $\mathbf{p}_2$ at each of the 11 timestamps of each date in $\hat{D}(y)$. The structure of vector P is analogue to V:

$$P(y, l, \mathbf{p}_2, \mathbf{n}) = (\mathbf{q}_1^1, \mathbf{q}_2^1, \ldots, \mathbf{q}_{11}^1, \mathbf{q}_1^2, \ldots, \mathbf{q}_{11}^2, \mathbf{q}_1^3, \ldots, \mathbf{q}_{11}^3, \mathbf{q}_1^4 \ldots, \mathbf{q}_{11}^4)^T \tag{9.4}$$

in which each projection $\mathbf{q}_j^i$ is a (transposed) 2D vector computed as in Eq.9.3 and depends on the the plane normal $\mathbf{n}$ and the parameters defining the Sun's position (latitude and time). In an ideal configuration, the coordinates of the computed projections in P perfectly overlap with the sundial's points in V. In practice the projection can not be perfect,

so we aim to minimize the squared distance of all the projections from the corresponding sundial's points. Thus, we formulate the following non-linear least square problem:

$$(l^*, \mathbf{p}_2^*, \mathbf{n}^*) = \underset{l,\mathbf{p}_2,\mathbf{n}}{argmin} \quad \sum_{y \in Y} (P - V)^T (P - V) \tag{9.5}$$

where P is a shortcut for $P(y, l, \mathbf{p}_2, \mathbf{n})$, which formulation involves non-linear terms, depending on the Sun's position. The set Y contains a list of years in which the solstices and equinoxes dates are computed for the subsequent shadow optimization. With this formulation we optimize simultaneously the latitude, the gnomon's vertex and the normal of the sundial's plane in such a way that the sum of the distances from the casted shadows to the target points is minimized.

Gnomon's Inclination Optimization

In this second optimization, we keep fixed the coordinates of p_2, the latitude and the plane normal as the optimal values obtained in the previous task, so that the position of p_1 has an impact only on the shadow's shape and inclination. Indeed, in this setup the role of p_1 is simply aesthetic because the oblique part of the gnomon projects a segment which should be aligned with the eleven sundial's marks to ease the time reading.

For this reason, the position of p_1 is optimized such that its punctual projection falls as close as possible to the line of the corresponding time mark, determined by the current time slot. The only parameter we need to optimize in this case is the junction's height, since the original gnomon was supposed to rise vertically from its base. Thus, the coordinates of point $\mathbf{p}_1$ are:

$$\mathbf{p}_1 = (x_0, y_0, h)^T \tag{9.6}$$

where (x_0, y_0) are the coordinates of the gnomon's base (Figure 9.9, the black point) and h is the elevation of the junction point.

Similarly to what we did for p_2, we define a function $P_1(d, l, \mathbf{p}_1, \mathbf{n})$ that gives us the eleven projections of point $\mathbf{p}_1$ during the day d at latitude l. Such projections are computed in correspondence of the eleven timestamps that split the daylight of d into twelve equal time slots.

$$P_1(d, l, \mathbf{p}_1, \mathbf{n}) = (\hat{\mathbf{q}}_1, \hat{\mathbf{q}}_2, \ldots, \hat{\mathbf{q}}_{11})^T \tag{9.7}$$

where $\hat{\mathbf{q}}_i = (\mathbf{q}_i, 1)^T$. Note that, unlike the previous function P (that is computed for four particular days of the year), P_1 is defined for a single day, since we want the projection of p_1 to be close to the engraved segments during all days and not only during solstices and equinoxes.

We also need to express the parameters of the eleven lines containing the segments of the sundial. In a 2D Euclidean plane, a line can be denoted as $\ell = (a, b, c)$, involving three parameters and a point (x, y) lies on such line if $\ell \cdot (x, y, 1) = 0$.

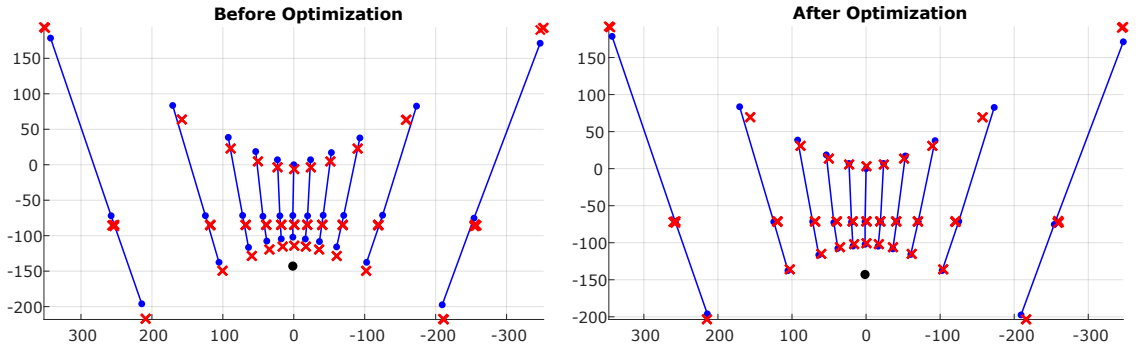


Figure 9.11: Optimization results of latitude and point p_2. The images show the projections of point p_2 (red marks) during equinoxes and solstices of several years before (left) and after (right) optimization. The rightmost configuration exhibits a better overlap between projections and sundial's lines intersections.

We collect these values in a 33-element vector $\mathbf{r}$:

$$\mathbf{r} = \left(\frac{\ell_1}{||\ell_1||}, \frac{\ell_2}{||\ell_2||}, ..., \frac{\ell_{11}}{||\ell_{11}||} \right)^T \tag{9.8}$$

such that $\ell_i = (a_i, b_i, c_i)$ are the three parameters of the i-th line lying on the plane. Each vector had also be normalized with its norm. Then, we minimize the squared distance of such projections from the correspondent line engraved on the sundial's surface.

The energy function to be minimized is the following

$$h^* = \underset{h}{argmin} \quad \sum_{y \in Y} \sum_{d \in D(y)} (P_1 \cdot \mathbf{r})^2. \tag{9.9}$$

The optimization task is performed during a set of years Y, and we defined a set $D(y)$ which returns all the valid dates in a year y. Unlike the previous energy function, that involves non-linear components deriving from the sun's position, this function is linear and thus can be optimized through least squares method.

9.5 Results

To have a robust estimation of the gnomon's configuration and latitude, both optimization tasks were performed over a set of several years. We chose a set of plausible years in which the sundial were in use and optimized the configuration for that specific period.

The location of gnomon's tip p_2 and the latitude optimization (described in section 9.4.3) was performed computing all the solstices and equinoxes from 200 AD to 210 AD and minimizing the energy function (9.5). The Nelder-Mead simplex algorithm for function optimization has been used, as described in [108]. Such method is a numerical algorithm used in multidimensional optimization problems for which derivatives are not

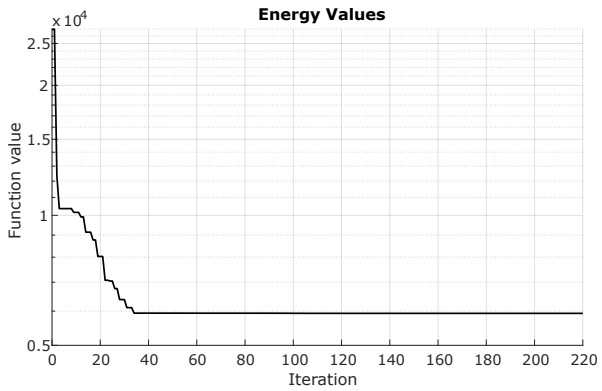


Figure 9.12: Energy values during the joint optimization of the latitude and p_2. For each iteration of the optimization algorithm (on x-axis) the energy value is displayed using logarithmic scale.

known, like in our case in which the sun position function that is not derivable.

The initial latitude was set at $45.46°$ North (Aquileia latitude) and the gnomon's endpoint was positioned in a plausible way such that its shadows fall close to the ideal solstices and equinoxes points. The plane normal was also initialised as $\mathbf{n} = (0, 0, 1)$, that is perfectly parallel with respect to the ground. The initial configuration is displayed in Figure 9.11 (left), in which each red cross marks the shadow projected by the gnomon's endpoint during the selected days of the period.

The optimization took around 250 iterations to converge, with a maximum tolerance over both the energy values and the optimized value equal to 10^{-5}. The energy values during the optimization process are shown in Figure 9.12 against the number of iterations (in x-axis). To assess the process stability, we run several optimizations slightly changing the initial gnomon's endpoint and we observed the same final configuration. After the optimization, the optimal working latitude of the sundial was 44.019 N (in decimal notation), while the endpoint of the gnomon was $p_2 = (23.8225, -0.0007, 50.1031)$. As for the plane normal vector $\mathbf{n}$, each test did not change its orientation, so that we kept the sundial's plane parallel to the ground. The qualitative result obtained after the optimization is shown in Figure 9.11 (right): the shadows casted by the gnomon are now almost coincident with the marks on the sundial, except in some cases.

Regarding the junction point p_1, we optimized its height over all days of the selected years. Since the function (9.9) is linear, we used the optimal latitude from the previous process and analytically minimized it via linear least squares. The optimal value for p_1's height resulted as $h = 47.97$. Figure 9.13 displays the results obtained with the complete optimal configuration of the gnomon. The leftmost plot shows the set of shadows projected during the day respectively in: Winter solstice (in blue), Summer solstice (red) and equinoxes (green). Note that the inclination of the projected segment is almost parallel to the engraved lines and the top part of the shadows points at the two hyperbole and the line.

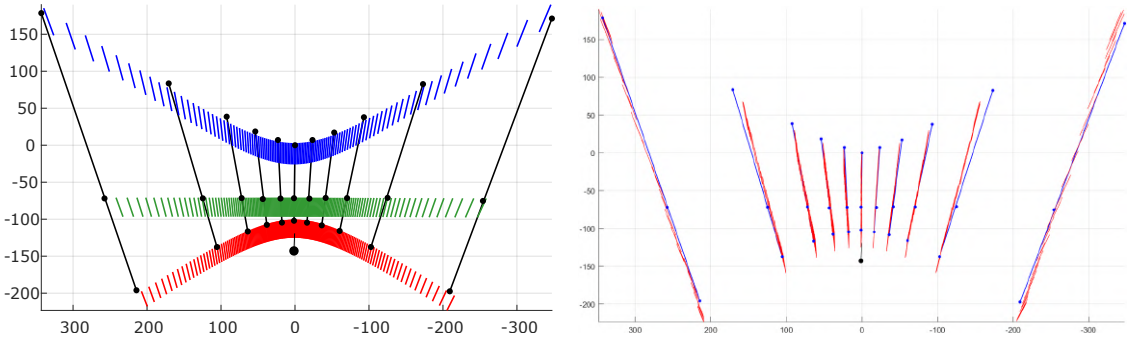


Figure 9.13: Final configuration results. Left: shadows projected during daylight time in Winter solstice (blue), Autumn and Spring equinoxes (green) and Summer solstice (red). Right: shadows projected during a whole year (every 10 days), in correspondence of the eleven sundial's time marks.

The rightmost plot shows the shadows (in red) projected during a whole year in correspondence of the timestamps that denotes the change of timeslots in which each day is divided. Note that the inclination of the shadows is almost always parallel to the sundial segments, thanks to the optimization of p_1.

Finally, a simulation of the shadow advancement over the day during a year was performed. Figure 9.14 shows three rendered pictures with the computed gnomon and the projected shadow during the summer solstice, winter solstice and equinoxes. The gnomon shape and the latitude simulated in the rendering are the optimal values computed during the optimization task.

9.6 Conclusions

The scanning of the Euporus sundial is the first step of a more in-depth analysis to be undertaken on the artefact. The reconstruction provides a high quality 3D shape record

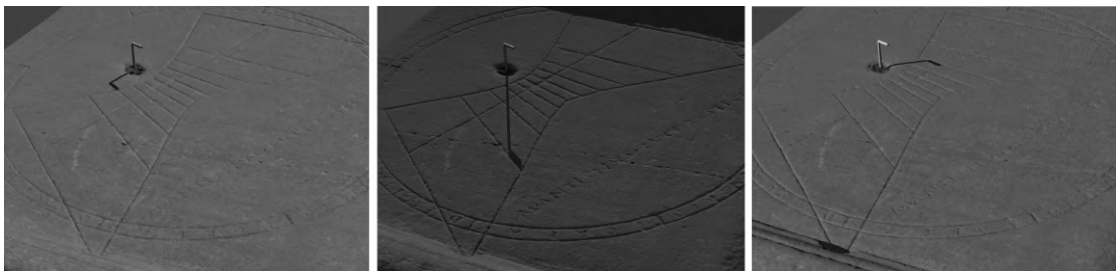


Figure 9.14: Digital rendering of the acquired sundial with the simulated gnomon as computed in our optimization process. The sun position was set according to the optimal latitude. Pictures show the casted shadow during (from left to right): summer solstice, winter solstice and equinox.

of the sundial's surface with measurement accuracy to the millimetre level and serve as a fine 3D representation that can be used for documentation, research and conservation purposes.

Moreover, the 3D model supported a further analysis to determine the geographical location for which the sundial was designed and shaped. In particular, a mathematical analysis of the gnomon's projection and the relative optimization process shown that the optimal working latitude for this kind of configuration is indeed a southern location with respect to the place where the object was discovered (lat 44.019 N, while Aquileia latitude is 45.79 N). A plausible hypothesis is that the Euporus sundial inscription was copied from another object designed for more southern latitudes, and perhaps some calculation errors caused its slightly defective functioning in Aquileia area.

III

Future Works and Conclusions

10

Exploring Surface Detail Transfer via Deep Learning

Almost all the 3D acquisition processes output an oriented point cloud on which a uniform 3D surface needs to be interpolated. Ideally, we would like to capture the object's small surface details together with its geometry, but non-uniformity of acquired data is still an open issue for surface reconstruction algorithms. In this Chapter we propose a preliminary work in the field of surface detail transfer employing learning techniques. Specifically, we first propose a technique which exploits spectral analysis to effectively separate high resolution areas of a surface from the zones exhibiting poor details. After that, a neural network is trained to transfer the desired high frequencies from high resolution patches to low resolution areas, allowing for a realistic surface reconstruction.

10.1 Introduction

Previous chapters mainly focused on 3D acquisition methods and subsequent practical applications, but three-dimensional data acquisition is only the first step for a complete reconstruction. Usually when an object needs to be completely acquired, it is first observed from different points of view so that the union of all the range-maps provides a complete surface coverage. The process of joining all the views in a unique coherent point cloud is called *registration*. Several algorithms perform this task, from the popular ICP (Iterative Closest Point) [27, 50] to other approaches that exploit local 3D features to assess matches between different views [177].

After registration, the final surface must be computed. Popular surface reconstruction techniques compute the so-called *implicit representation* of the surface where the set of acquired 3D points lies. Such representation consists in a function defined in 3D space domain returning negative values for all the points contained inside the object's volume, positive values for external points and zero on the surface. A special case of the general implicit function is the *signed distance function* (SDF), which returns the signed distance from the surface.

In the literature, several approaches propose to estimate this function and to extract the zero iso-surface to compute the final surface. Some methods require normal vectors and employ Radial Basis Functions interpolation [36] or cast the reconstruction as a spatial Poisson problem [99]. The latter proposes a heuristic technique to deal with data non-uniformity, consisting in the estimation of local density to weight the corresponding kernels accordingly. With the implicit representation, Marching Cubes [130] is often used to discretize the 3D space in regular grid cells that are classified as inside or outside the object's surface, performing linear interpolation over the edges resulting in a (usually watertight) triangulated surface. Recently, Deep Learning approaches have been proposed to perform the same task [124] but, despite promising, their spatial resolution is very limited.

In general, surface reconstruction methods suffer from a non-uniform data sampling over the surface. Indeed, the acquisition itself and the subsequent registration process cause the data to be unevenly distributed, generating a variable density over the surface to be recovered. In those cases, surface reconstruction techniques are not able to accurately represent the object geometry at smaller scales, since they tend to smooth out zones with a low point density, implicitly losing the captured details. Moreover, marching cubes algorithm has a fixed grid size, so the final reconstruction needs to balance a trade-off between surface detail fidelity and time complexity.

In many cases the acquired object exhibits a characteristic surface pattern all over its extent, but the non-uniformity of data causes the loss of such detail from the zones in which it should have been recorded. In those cases the overall surface quality does not correspond to the desired outcome, even having a prefect implicit function.

The Europus sundial presented in Chapter 9 is a classical example of this situation: the density of acquired points is not uniform over its surface for two main reasons. First, the final point cloud is denser and more detailed in correspondence of the engraved zone. This was done on purpose, since inscriptions are the most salient details of the whole

artefact, requiring a more accurate acquisition. Second, the object's placement made impossible to evenly acquire the whole surface from some points of view, so we had a lower resolution in the zones occluded by the benches. Moreover, even having a perfectly uniform acquisition all over the surface, the registration process creates inevitable overlaps, causing non uniform data distribution. In that application we adopted the Poisson surface reconstruction and, although we had satisfactory results, the detail loss is visible on the final reconstruction. The low resolution areas result smoother and the high resolution ones are in some way mitigated due to the need of the algorithm to interpolate the whole point cloud and produce a good implicit function. The entire object is made from the same material, so its real surface exhibits anywhere the same kind of "roughness", which unfortunately is not entirely evident on the resulting surface. Ideally, we would like to transfer the high frequencies contained in dense areas to the entire object's surface, where low resolution zones generated unrealistic smooth details.

The same issue is encountered in several fields like computer graphics, where there is the need of generating realistic 3D surfaces by adding high-resolution details to simple models in order to obtain the appearance of many materials. Other applications requiring a surface-level editing while preserving the geometry of the object are free-form deformation [188], fusion [94] or morphing [10].

The described task is well reported in the literature and addressed as *detail transfer*. Several approaches have been proposed, some of them work in 2D domain and are applied to images [129] or perform texture synthesis [69, 218]. Detail transfer on 3D surfaces is also a widely explored topic: some technique propose a pure geometrical approach [195] or learning-based methods [26]. Other similar approaches consist in surface completion, i.e. closing surface holes observing the contextual information given by the rest of the surface [147, 191]. Recently, geometric deep learning have been proposed [31, 144], offering novel techniques to generalize CNN architectures in order to work in non-Euclidean domains, such as graphs and manifolds.

In this Chapter we present a preliminary exploration in the field of surface detail transfer using learning techniques. We are especially interested in cases similar to the sundial example, in which we need to reconstruct objects exhibiting a specific surface pattern. In the majority of the cases, the non-uniform data distribution over the surface and the reconstruction process cause detail loss in low resolution areas, discarding valuable information. The following study is divided in two main parts: first, the separation of surface detail from geometry through spectral analysis is explored in order to identify which areas contain the features to be transferred over the rest of the surface. Second, we propose a learning-based technique to transfer high-frequency details from denser zones of the surface to sparse areas keeping the underlying geometric topology.

In both tasks we adopted a patch-wise approach, as commonly carried out in the literature. This choice is driven by two motivations: (i) detail patterns are locally recognizable in small patches with no need to observe the whole surface, and (ii) the patch-based approach is more tractable in terms of time complexity and offers good scalability in the case of large amounts of data.

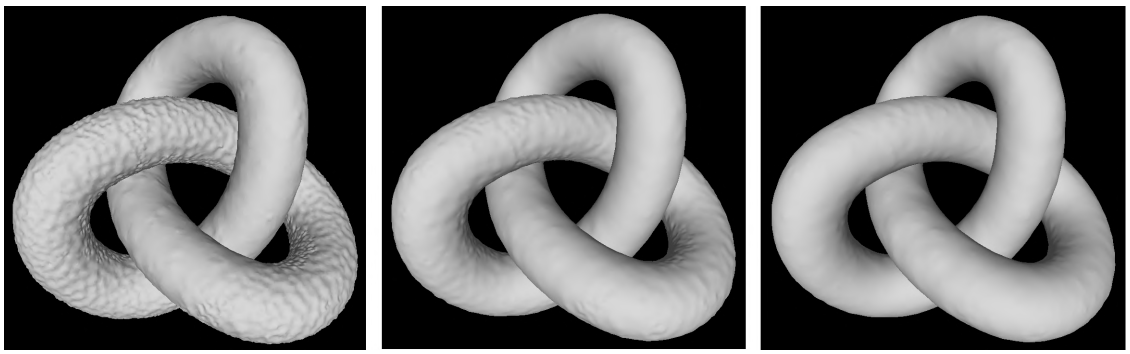


Figure 10.1: Different smoothing levels for the same acquired surface. From left to right: original surface S_0, after applying three times Laplacian filter S_3, and the maximum smoothing level (in this case S_8).

10.2 Separating Surface Detail from Topology

As we already discussed, the aim of this study is to transfer high-frequency details all over the surface extent in order to obtain a final object reconstruction that is as realistic as possible. The main target of the whole process is to accomplish such task so that the underlying object topology stays unaltered and only the relevant details are isolated and transferred in the appropriate patches.

To do so, we first need to define an automatic technique that allows to efficiently and robustly distinguish the "detailed" part of the surface from low resolution areas. Our proposal is to apply several levels of smoothing to the acquired surface and then, for each smoothness level, analyse the Fourier spectrum of extracted patches containing the displacements of acquired points from the surface. The general idea behind this procedure is to identify the appropriate smoothness level which efficiently separates the geometric component of the surface from the details, that have to be isolated. Once such level is established the displacement patches are extracted and processed by the detail transfer technique that will be described in the next Section.

Let's assume we acquired a point cloud P, resulting from a 3D scanning process as described in the previous Chapters. Then, after applying some surface reconstruction technique, we obtain a surface $S = (V, T)$, composed by a set of vertices V interconnected by a set of triangles T. Note that in general $P \neq V$, since the surface reconstruction technique usually generates a new point set lying on the surface.

A smoothing filter (like surface Laplacian) is consecutively applied N times to S, obtaining a sequence $S_0, S_1, ..., S_N$ of different surfaces. To obtain S_i, with $i = 1, ..., N$ the smoothing filter is applied to surface S_{i-1}, then normals are recomputed for the new vertices and the process is repeated until the changes applied by the filter are not visible. In particular, note that $S_0 = S$ is the original surface and S_N represents the surface with the maximum smoothing level. In Figure 10.1 three different levels of smoothing are displayed. We produced a synthetic surface simulating a non-uniform sampling over the

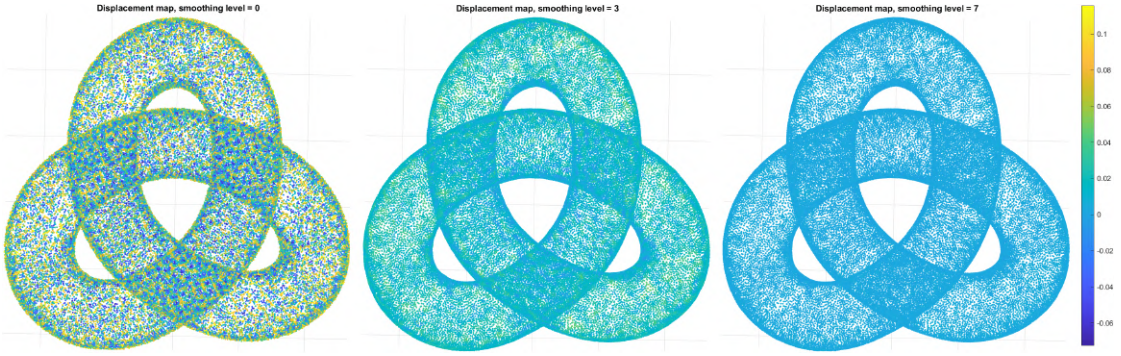


Figure 10.2: Displacement values for each vertex with respect of the maximum smoothed surface S_8. From left to right: S_0 (the initial surface, exhibiting larger displacement values), S_3 and S_7.

original point cloud, resulting in areas of different detail levels. We applied $N = 8$ times Laplacian smoothing filter and display (from left to right): S_0, S_3 and S_8.

For each smoothed surface $S_i = (V_i, T_i)$, its displacement values with respect to the surface S_N are computed. These values are obtained for each point in V_i by taking the length of the segment joining each vertex and the surface S_N along the direction defined by its normal vector. In this way we obtain, for each surface S_i, a displacement map defined over all its vertices and representing the amount of shifting with respect to the surface S_N. Note that surface S_N ideally captures only the topological structure of the original surface, since the details have been removed by repeated smoothing filters. Figure 10.2 displays three displacement maps for surfaces S_0, S_3 and S_7: as the surface approaches the maximum smoothing level, the displacement values are smaller.

Assume we have a set of M local patches uniformly distributed over the surface, each one characterized by a centre and a normal vector defining a tangent plane. Every patch captures a small portion of the surface, and it is associated with a set of surface vertices (and their incident triangles) for which the geodesic distance from the patch centre is smaller than a radius r [1].

Each 3D patch can then be mapped onto its associate tangent plane (defined by the normal vector) to obtain a 2D representation of its features. The vertices belonging to a patch are first projected in the tangent plane, adjusting their position to preserve the correct geodesic distance with respect to the patch centre (along the same projected direction on the planar surface).

For each smoothness level $i = 0, ..., N-1$, the displacement values associated to patches are projected on its tangent space, generating a sparse set of 2D points for each local patch. Such displacement values are then interpolated over a regular grid to compute a dense displacement map, that is an image representation of the local patch displacements. In this way we obtain, for each smoothness level i, a set of M images containing the

[1] Note that, while the centres stay unchanged, the patches themselves change for different smoothing levels and they are designed to partially overlap.

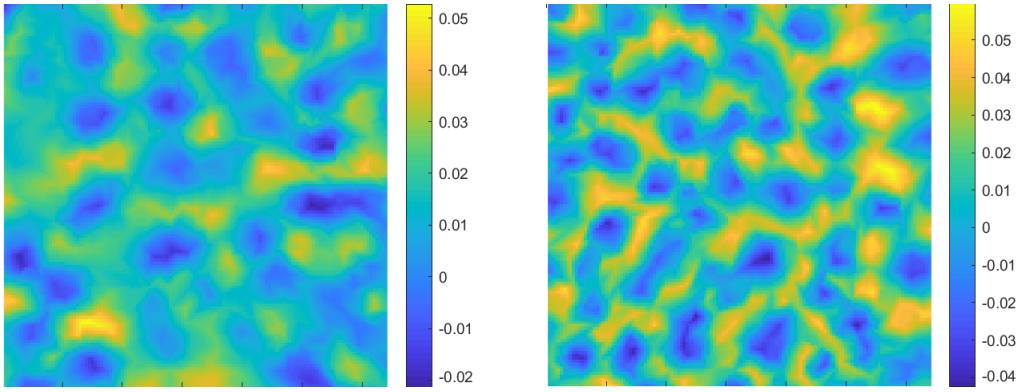


Figure 10.3: Examples of image patches with displacement values coming from low (left) and high (right) resolution zones of the surface.

surface displacement pattern of every extracted patch. In Figure 10.3 two examples of 128×128 image patches coming from low (left) and high (right) resolution areas of the surface are displayed.

As already stated, we are particularly interested in analysing the frequency distribution over the surface patches as the smoothing level increases. Ideally, low frequencies characterize the topological part of the surface, while high frequencies carry information about the detail we would like to isolate. Our goal is to exploit the patch image frequencies to identify an ideal smoothing that allows to separate the surface geometry from the details.

To do so, the 2D Fourier transform is used to compute the spectrum magnitude for each 2D patch image. Note that we are only interested in analysing the amount of low and high frequencies contained in the patches, rather then the harmonics directions. For this reason, we restrict our analysis to the omnidirectional spectrum of each patch. The omnidirectional spectrum for a 2D image is obtained by fixing a number of frequency bins (that can be viewed as concentric annuli) and accumulating the power spectrum values for each bin, normalizing by the bin area. The result is a 1D spectrum with an intensity value corresponding to each bin, with increased frequency. In this approach, the same spatial frequencies (i.e. harmonics with the same period but different directions) are summed together, making the final result totally independent from the patches orientations. The M omnidirectional spectra of all the patches lying on the same surface S_i are grouped together, obtaining N sets of curves.

Figure 10.4 shows omnidirectional spectra for surfaces S_0, S_3 and S_7: we used 18 frequency bins and plotted the logarithm of the power spectrum. Low and high resolution patches are plotted with different colours in order to observe their behaviour at different smoothing levels. We observe that, in general, in correspondence of higher frequencies the two classes of patches tend to separate and group together.

This separation becomes less evident applying a higher smoothing, since the two groups tend to gradually merge as the smoothing level increases. This is expected since the patches extracted from denser areas contain more surface details (i.e. high frequen-

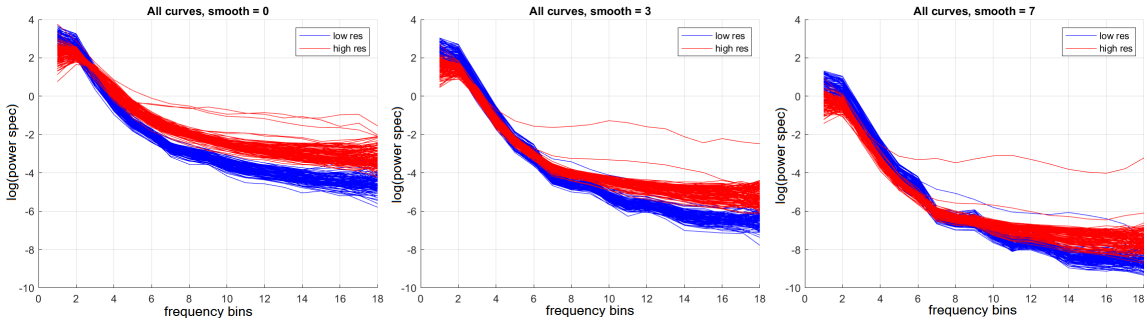


Figure 10.4: Logarithm of omnidirectional spectra (using 18 bins) for all patches at smoothing levels 0, 3 and 7. Each curve corresponds to the omnidirectional spectrum of a patch. Low and high resolution patches are plotted respectively in blue and red.

cies) than the others, and the repeated smoothing filters cause a decreasing in high frequencies. On the other hand, low frequencies in the spectrum represent the geometrical component of the patches. In the leftmost plot we observe a small overlap in the low frequencies area, while in the central and rightmost plots the two sets of curves tend to increasingly overlap between bin 1 and 8. In particular, the curves at smoothing level 3 (central plot) exhibit a significant overlap in low frequencies (from 0 to 6), while maintaining a good separation in the higher frequencies part.

This means that the observed patches at smoothing level 3 preserve the underlying surface topology, but still exhibit different detail levels, as expected. This situation is meaningful since we can exploit this property to (i) isolate low resolution patches and (ii) transfer the detail contained in the high frequency part from the high resolution patches to the others, without altering the object's geometry.

The final objective of this process is to exploit the computed omnidirectional spectra of image patches to automatically devise the correct smoothing level and thus characterize surface detail and separate it from topology. Once we hold this "optimal" smoothed surface, we can perform detail transfer.

10.3 Detail Transfer via Deep Learning

In this Section we propose a learning-based technique that is able to learn high frequency details from high resolution patches and to properly transfer the same kind of roughness to low resolution patches.

Assume we have formerly identified S_D as the optimal surface capturing the object's topology, and thus having a partition over the patches separating detailed ones from low resolution ones. Then, we compute for each patch a *sparse* and a *dense* image: sparse data comes from displacements of original points P with respect to S_D, while dense data from the displacements of S_D with respect to S_0. The main idea is to feed a specifically designed network with acquired sparse data and obtain dense patches as output, which ideally "filled" sparse data with the required level of detail.

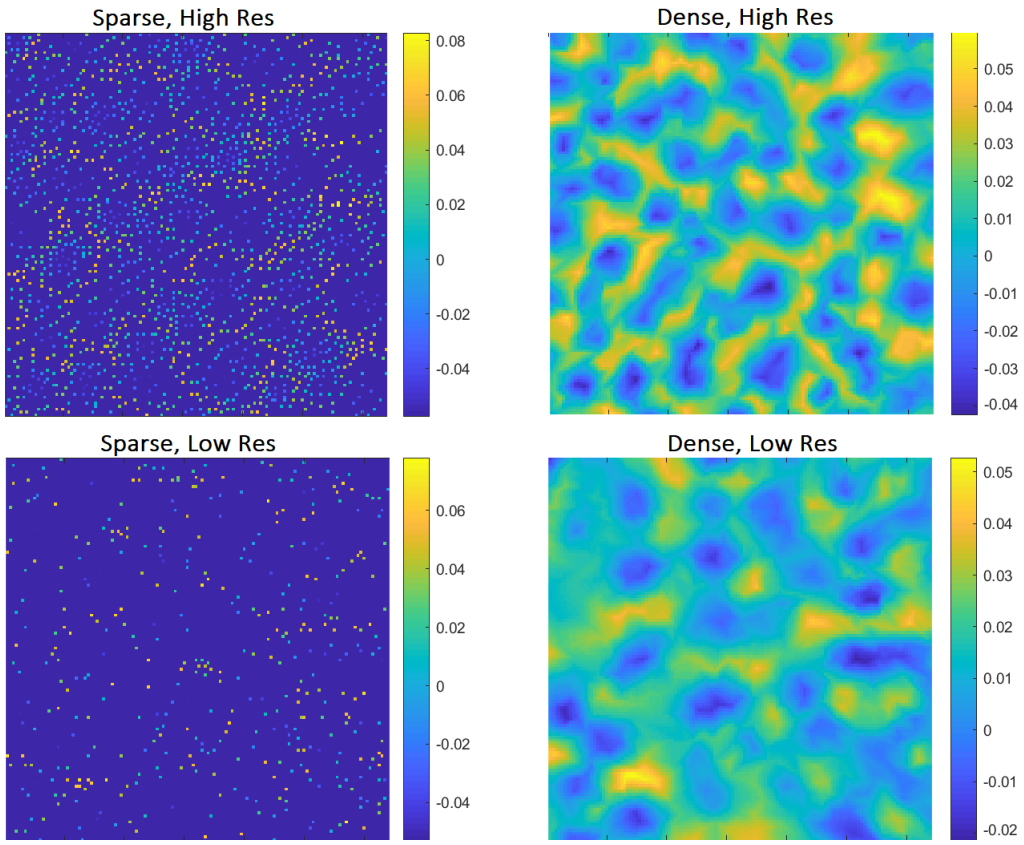


Figure 10.5: Dense and sparse patches generated from the optimal surface S_D. The first row shows the same high resolution patch, second row a low resolution patch.

Specifically, sparse patches contain the displacements of the original point cloud P with respect to surface S_D. Each point $\mathbf{x}$ in P is projected onto the closest point lying on surface S_D, and it is associated with its signed distance from the surface. The projected points are then moved to the patch tangent plane as already described. Sparse patches are originated directly from input data, thus they capture the original point cloud density over the surface. Moreover, they are defined only in correspondence of projections, originating sparse data. Figure 10.5 (left) displays two examples of sparse patches: from a detailed area (high resolution, top) and a low resolution patch (bottom). On the other hand, dense patches images are computed by taking the displacements of S_0 with respect to S_D, in the same way we computed the displacement maps in the previous Section. The computed values represent the amount of displacement to apply to vertices in S_4 to meet the surface S_0. Such values are projected on the tangent plane and interpolated on a regular grid to obtain dense images. Rightmost part of Figure 10.5 shows the dense version (with high and low resolutions) of the two sparse patches in the leftmost part, highlighting the difference between low and high resolution patches.

Our approach consists in using only high resolution patches to generate the training data, simulating different (lower) sampling over sparse patches in such a way that they

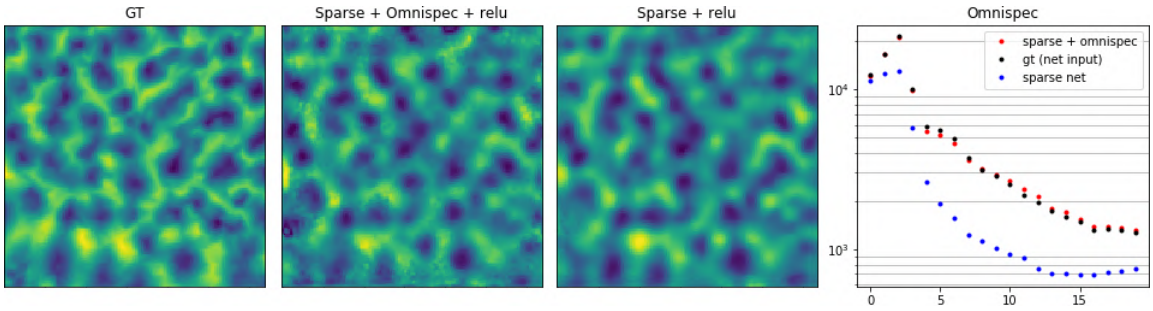


Figure 10.6: Example of patch contained in the test set. Leftmost picture is the ground truth detailed patch. Following the network output with and without omnidirectional spectrum normalization. Rightmost plot shows the three spectra: the target values (black), the output with (red) and without (blue) spectrum normalization.

resemble low resolution data. The expected output are then the detailed dense patches, exhibiting a compatible omnidirectional spectra.

In this way, we can formulate the detail transfer as a particular depth completion problem. In the literature, depth completion is addressed as the task of recovering dense depth maps from sparse data (for example, coming from LIDAR devices). Several learning approaches propose specifically designed neural networks which are able to process sparse input data, provided with a validity mask. Sparsity invariant CNN [215] and its hierarchical multi-scale version HMS-Net [87] define sparsity invariant layers to perform this task, exploiting a boolean validity mask given as input. Other approaches simulate different densities in input images to train an encoder-decoder network and make it sparsity-invariant [92] or employ classical image processing techniques [105].

In our approach we adopt an architecture similar to HMS-Net [87], but we define an additional custom layer which normalizes the omnidirectional spectrum of the output image. Input data are three-channel images, containing the sparse image patch, its validity mask and a high-resolution dense patch used only as target for the output spectrum.

The proposed spectrum normalization layer takes the target image (thus with an ideal spectrum) and the result of sparse net computation. Then, it normalizes the spectrum of the final dense patch such that its omnidirectional spectrum has the same values of the target image. To perform this operation, each frequency bin of the spectrum of predicted image is uniformly scaled according to the ratio between the target and the input image spectra, regardless the direction. In this way the predicted patch is not altered in its pattern, but only the appropriated frequencies are enhanced, accordingly to the target patch. Figure 10.6 shows the effect of omnidirectional spectrum normalization for an image contained in the test set. The leftmost picture is the expected output (the ground truth), while the two central patches represent the network output with and without the normalization layer. It is evident that the former contains sharper details, being more compatible with the target patch. The rightmost plot shows omnidirectional spectra for the three displayed images: the target spectrum (black) is almost overlapped with the net

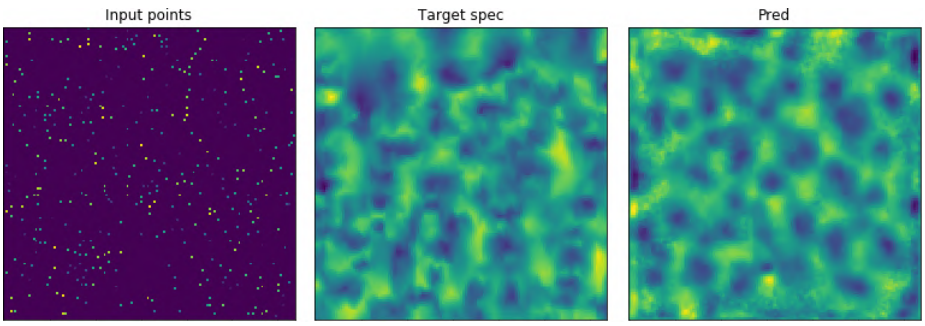


Figure 10.7: Sparse low-resolution patch (left), dense patch used as target spectrum for normalization layer (centre) and predicted dense patch (right).

output employing the additional normalization layer (red), while the result of basic sparse net (blue) exhibit a different shape, having less low frequencies.

Figure 10.7 shows an example of detail generation applied to a low resolution patch. Leftmost picture displays the sparse input points coming from a low detailed area of the surface, the central patch is a high resolution dense patch (from the training set) used only as spectrum target and the rightmost picture represents the final dense patch with added high frequency details.

Finally, we applied the proposed transfer method to all sparse low-resolution patches, employing images from the training set as targets for spectrum normalization. The set of dense images predicted by the network are then applied to the correspondent surface patches by applying the displacement values to the involved vertices of the original surface S_0. The surface with transferred details is shown in Figure 10.8 (right), where it is compared to the original surface S_0 (left).

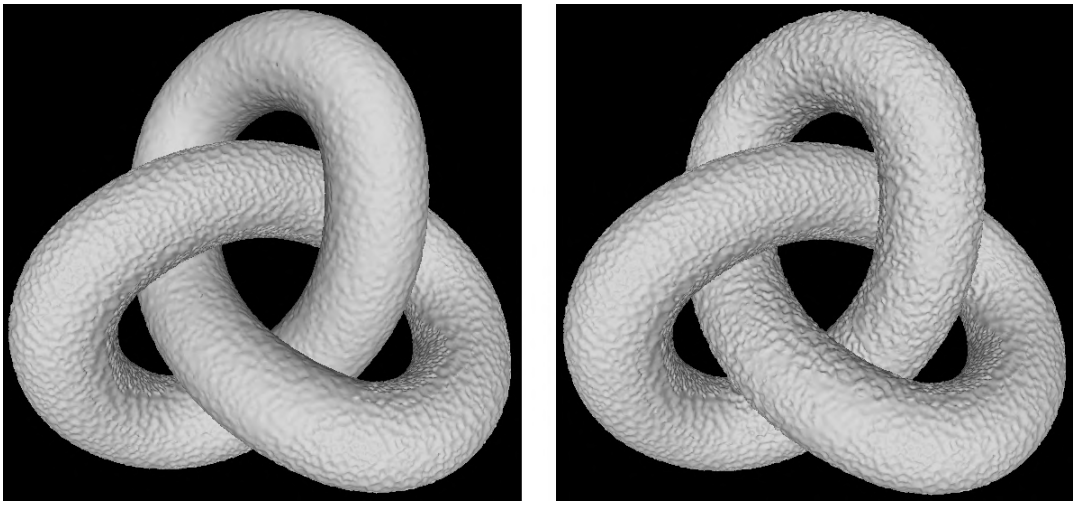


Figure 10.8: Original surface (left) and after the application of patch displacements predicted by the network.

10.4 Conclusions

This Chapter presents a preliminary investigation in the field of surface detail transfer using deep learning techniques. The contribution of this work is twofold: first, an approach that exploits patches omnidirectional spectra to separate detailed from low resolution patches is presented. Second, a specifically designed network takes sparse input data representing the displacements of the original point cloud and predicts the corresponding dense patch, properly adding high frequency details. The proposed network uses a custom layer that is able to normalize the omnidirectional spectra of the processed patch, making it compatible with the high resolution areas of the surface. A further investigation step on the topic could be taken in the direction of geometric deep learning, applying such paradigm directly to 3D data instead of working with planar patches.
The qualitative preliminary results presented throughout this Chapter show good hints about the method's potentialities and represent a promising starting point for a further formalization and analysis.

11
Conclusions

In this thesis we discussed 3D reconstruction techniques and in particular structured-light methods. The main purpose of the thesis is in fact to present novel contributions for phase unwrapping algorithms and correction techniques for phase shift method. Moreover, some practical applications which benefit from the usage of the proposed techniques have been introduced.

Chapter 3 proposed a novel phase unwrapping approach that considers intrinsic signal properties and casts absolute phase computation into a probabilistic framework. Code recovery can be used in conjunction with our probabilistic phase unwrapping method exploiting likelihood function peaks or be applied to any fringe-based techniques independently of the unwrapping algorithm, as shown in Chapter 5. Finally, a phase correction method was proposed in Chapter 4 allowing to fix phase measurements that are corrupted by non-uniform albedo in the acquired surface. In all cases, experimental results confirmed the theoretical assumptions on acquired signal and also exhibit overall improvements with respect to several state-of-the-art techniques.

The second part of the Thesis is devoted to present a set of practical applications related to 3D reconstruction. In Chapter 6 we proposed a practical calibration technique working with structured-light scanner composed by a camera-projector pair, while Chapters 7 and 8 are devoted to industrial applications. We presented a robust cylinder extraction technique working with non-oriented point clouds and a micrometric surface reconstruction using a tailored structured-light approach. A different practical application is presented in Chapter 9, where the scanning process of an archaeological artefact is presented. The acquired 3D model is then used to devise relevant information about its functioning during Roman times. Finally, Chapter 10 described a preliminary exploration in the field of surface detail transfer using learning techniques. Preliminary results shown in this Chapter are promising and will be the base of a subsequent formalization and improvement in this topic.

Overall, the topics covered by the thesis offer some good examples on how 3D data can be successfully exploited in heterogeneous fields and with different goals in mind. Each single application presented in the thesis has its own distinctive objective, that are precise measurements, accurate parameter estimation or realistic surface reconstruction. In all of these cases we proposed a reconstruction approach which is able to satisfy specific needs, depending on the desired outcome.

Bibliography

[7] ADALLAH, H. B., ORTEU, J.-J., DOLIVES, B., AND JOVANČEVIĆ, I. 3d point cloud analysis for automatic inspection of aeronautical mechanical assemblies. In *Fourteenth International Conference on Quality Control by Artificial Vision* (2019), vol. 11172, International Society for Optics and Photonics, p. 111720U.

[8] AKCA, D., REMONDINO, F., NOVÁK, D., HANUSCH, T., SCHROTTER, G., AND GRUEN, A. Recording and modeling of cultural heritage objects with coded structured light projection systems. In *2nd International Conference on Remote Sensing in Archaeology* (2006), Institute of Geodesy and Photogrammetry, ETH Zurich, pp. 375–382.

[9] AKÇA, M. D. 3d modeling of cultural heritage objects with a structured light system.

[10] ALEXA, M. Differential coordinates for local mesh morphing and deformation. *The Visual Computer 19*, 2 (2003), 105–114.

[11] AMIR, Y. M., AND THÖRNBERG, B. High precision laser scanning of metallic surfaces.

[12] AN, Y., HYUN, J.-S., AND ZHANG, S. Pixel-wise absolute phase unwrapping using geometric constraints of structured light system. *Optics express 24*, 16 (2016), 18445–18459.

[13] AN, Y., AND ZHANG, S. Three-dimensional absolute shape measurement by combining binary statistical pattern matching with phase-shifting methods. *Applied optics 56*, 19 (2017), 5418–5426.

[14] AUBER, P. A. L'orologio solare orizzontale del circo di aquileia (ii sec. d.c.). il "plintio di euporus". prima parte. *Atti dell'Istituto Veneto di Scienze, Lettere ed Arti, Classe di scienze fisiche, matematiche e naturali CLXIII (2004-2005) 231-298*.

[15] BAY, H., ESS, A., TUYTELAARS, T., AND VAN GOOL, L. Speeded-up robust features (surf). *Computer vision and image understanding 110*, 3 (2008), 346–359.

[16] BELL, T., XU, J., AND ZHANG, S. Method for out-of-focus camera calibration. *Applied optics 55*, 9 (2016), 2346–2352.

[17] BERALDIN, J.-A., PICARD, M., EL-HAKIM, S. F., GODIN, G., VALZANO, V., AND BANDIERA, A. Combining 3d technologies for cultural heritage interpretation and entertainment. In *Videometrics VIII* (2005), vol. 5665, International Society for Optics and Photonics, p. 56650C.

[18] BERGAMASCO, F., ALBARELLI, A., COSMO, L., RODOLA, E., AND TORSELLO, A. An accurate and robust artificial marker based on cyclic codes. *PAMI 38*, 12 (2016), 2359–2373.

[19] BERGAMASCO, F., ALBARELLI, A., COSMO, L., TORSELLO, A., RODOLÀ, E., AND CREMERS, D. Adopting an unconstrained ray model in light-field cameras for 3d shape reconstruction. In *CVPR 2015* (2015), pp. 3003–3012.

[20] BERGAMASCO, F., ALBARELLI, A., COSMO, L., TORSELLO, A., RODOLÀ, E., AND CREMERS, D. Adopting an unconstrained ray model in light-field cameras for 3d shape reconstruction. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition 07-12-June-2015* (2015), 3003–3012.

[21] BERGAMASCO, F., ALBARELLI, A., AND TORSELLO, A. Image-space marker detection and recognition using projective invariants. pp. 381–388.

[22] BERGAMASCO, F., ALBARELLI, A., AND TORSELLO, A. Image-space marker detection and recognition using projective invariants. In *3DIMPVT 2011* (2011), pp. 381–388.

[23] BERGAMASCO, F., COSMO, L., ALBARELLI, A., AND TORSELLO, A. A robust multi-camera 3d ellipse fitting for contactless measurements. In *3DIMPVT 2012* (2012), pp. 168–175.

[24] BERGAMASCO, F., COSMO, L., ALBARELLI, A., AND TORSELLO, A. Camera calibration from coplanar circles. In *ICPR 2014* (2014), pp. 2137–2142.

[25] BERGSTRÖM, P., ROSENDAHL, S., AND SJÖDAHL, M. Shape verification aimed for manufacturing process control. *Optics and Lasers in Engineering 49*, 3 (2011), 403–409.

[26] BERKITEN, S., HALBER, M., SOLOMON, J., MA, C., LI, H., AND RUSINKIEWICZ, S. Learning detail transfer based on geometric features. In *Computer Graphics Forum* (2017), vol. 36, Wiley Online Library, pp. 361–373.

[27] BESL, P. J., AND MCKAY, N. D. Method for registration of 3-d shapes. In *Sensor fusion IV: control paradigms and data structures* (1992), vol. 1611, International Society for Optics and Photonics, pp. 586–606.

[28] BOEHLER, W., AND MARBS, A. 3d scanning instruments. *Proceedings of the CIPA WG 6* (2002), 9–18.

[29] BOLLES, R. C., AND FISCHLER, M. A. A ransac-based approach to model fitting and its application to finding cylinders in range data. In *IJCAI* (1981), vol. 1981, pp. 637–643.

[30] BRAHM, A., RÖSSLER, C., DIETRICH, P., HEIST, S., KÜHMSTEDT, P., AND NOTNI, G. Non-destructive 3d shape measurement of transparent and black objects with thermal fringes. In *Dimensional Optical Metrology and Inspection for Practical Applications V* (2016), vol. 9868, International Society for Optics and Photonics, p. 98680C.

[31] BRONSTEIN, M. M., BRUNA, J., LECUN, Y., SZLAM, A., AND VANDERGHEYNST, P. Geometric deep learning: going beyond euclidean data. *IEEE Signal Processing Magazine 34*, 4 (2017), 18–42.

[32] BURKE, J., AND ZHONG, L. Suppression of contrast-related artefacts in phase-measuring structured light techniques. *Proceedings of SPIE - The International Society for Optical Engineering 10329* (2017).

[33] BUSÉ, L., GALLIGO, A., AND ZHANG, J. Extraction of cylinders and cones from minimal point sets. *Graphical Models 86* (2016), 1–12.

[34] CALLI, B., SINGH, A., BRUCE, J., WALSMAN, A., KONOLIGE, K., SRINIVASA, S., ABBEEL, P., AND DOLLAR, A. M. Yale-cmu-berkeley dataset for robotic manipulation research. *The International Journal of Robotics Research 36*, 3 (2017), 261–268.

[35] CALLI, B., SINGH, A., WALSMAN, A., SRINIVASA, S., ABBEEL, P., AND DOLLAR, A. M. The ycb object and model set: Towards common benchmarks for manipulation research. In *Advanced Robotics (ICAR), 2015 International Conference on* (2015), IEEE, pp. 510–517.

[36] CARR, J. C., BEATSON, R. K., CHERRIE, J. B., MITCHELL, T. J., FRIGHT, W. R., MCCALLUM, B. C., AND EVANS, T. R. Reconstruction and representation of 3d objects with radial basis functions. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques* (2001), ACM, pp. 67–76.

[37] CARRIHILL, B., AND HUMMEL, R. Experiments with the intensity ratio depth sensor. *Computer vision, graphics, and image processing 32*, 3 (1985), 337–358.

[38] CASPI, D., KIRYATI, N., AND SHAMIR, J. Range imaging with adaptive color structured light. *IEEE Transactions on Pattern analysis and machine intelligence 20*, 5 (1998), 470–480.

[39] CHAPERON, T., AND GOULETTE, F. Extracting cylinders in full 3d data using a random sampling method and the gaussian image.

[40] CHASLES, M. Note sur les proprietes generales du systeme de deux corps semblables entr'eux et places d'une maniere quelconque dans l'espace; et sur le deplacement fini ou infiniment petit d'un corps solide libre. *Bulletin des Sciences Mathematiques, Astronomiques, Physiques et Chimiques 14* (1830), 321–326.

[41] CHAZAN, G., AND KIRYATI, N. *Pyramidal intensity-ratio depth sensor*. Technion-IIT, Department of Electrical Engineering, 1995.

[42] CHEAH, C.-M., CHUA, C.-K., TAN, K.-H., AND TEO, C.-K. Integration of laser surface digitizing with cad/cam techniques for developing facial prostheses. part 1: Design and fabrication of prosthesis replicas. *International Journal of prosthodontics 16*, 4 (2003).

[43] CHEN, C., GAO, N., WANG, X., AND ZHANG, Z. Adaptive pixel-to-pixel projection intensity adjustment for measuring a shiny surface using orthogonal color fringe pattern projection. *Measurement Science and Technology 29*, 5 (2018), 055203.

[44] CHEN, H., AND BISHOP, J. Delaunay triangulation for curved surfaces. *Meshing Roundtable* (1997), 115–127.

[45] CHEN, K., SHI, T., LIU, Q., TANG, Z., AND LIAO, G. Microscopic three-dimensional measurement based on telecentric stereo and speckle projection methods. *Sensors 18*, 11 (2018), 3882.

[46] CHEN, X., LU, C., MA, M., MAO, X., AND MEI, T. Color-coding and phase-shift method for absolute phase measurement. *Optics Communications 298* (2013), 54–58.

[47] CHEN, X., MA, H., WAN, J., LI, B., AND XIA, T. Multi-view 3d object detection network for autonomous driving. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2017), pp. 1907–1915.

[48] CHENG, Y.-Y., AND WYANT, J. C. Two-wavelength phase shifting interferometry. *Applied optics 23*, 24 (1984), 4539–4543.

[49] CHENG, Y.-Y., AND WYANT, J. C. Multiple-wavelength phase-shifting interferometry. *Applied optics 24*, 6 (1985), 804–807.

[50] CHETVERIKOV, D., SVIRKO, D., STEPANOV, D., AND KRSEK, P. The trimmed iterative closest point algorithm. In *Object recognition supported by user interaction for service robots* (2002), vol. 3, IEEE, pp. 545–548.

[51] CLAUS, D., AND FITZGIBBON, A. W. A rational function lens distortion model for general cameras. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)* (2005), vol. 1, IEEE, pp. 213–219.

[52] CLIFFORD, W. K. *Mathematical papers*. Macmillan and Company, 1882.

[53] CUI, Y., SCHUON, S., CHAN, D., THRUN, S., AND THEOBALT, C. 3d shape scanning with a time-of-flight camera. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on* (2010), IEEE, pp. 1173–1180.

[54] DANIILIDIS, K. Hand-eye calibration using dual quaternions. *The International Journal of Robotics Research 18*, 3 (1999), 286–298.

[55] DE REU, J., DE SMEDT, P., HERREMANS, D., VAN MEIRVENNE, M., LALOO, P., AND DE CLERCQ, W. On introducing an image-based 3d reconstruction method in archaeological excavation practice. *Journal of Archaeological Science 41* (2014), 251–262.

[56] DELLEPIANE, M., CALIERI, M., AND DELL'UNTO, N. Monitoring archeological excavation using dense stereo matching techniques. Tech. rep., tech. report CNR-ISTI, Pisa, Italy, 2011.

[57] DELLEPIANE, M., CALLIERI, M., PONCHIO, F., AND SCOPIGNO, R. Mapping highly detailed colour information on extremely dense 3d models: the case of david's restoration. In *Computer Graphics Forum* (2008), vol. 27, Wiley Online Library, pp. 2178–2187.

[58] DER JEUGHT, S. V., AND DIRCKX, J. J. Real-time structured light profilometry: a review. *Optics and Lasers in Engineering 87*, Supplement C (2016), 18 – 31.

[59] DONEUS, M., AND NEUBAUER, W. 3d laser scanners on archaeological excavations. In *Proceedings of the XXth international symposium CIPA, Torino* (2005), pp. 226–231.

[60] DOUILLARD, B., UNDERWOOD, J., KUNTZ, N., VLASKINE, V., QUADROS, A., MORTON, P., AND FRENKEL, A. On the segmentation of 3d lidar point clouds. In *2011 IEEE International Conference on Robotics and Automation* (2011), IEEE, pp. 2798–2805.

[61] DUDA, R. O., AND HART, P. E. Use of the hough transformation to detect lines and curves in pictures. Tech. rep., Sri International Menlo Park Ca Artificial Intelligence Center, 1971.

[62] DURDLE, N. G., THAYYOOR, J., AND RASO, V. An improved structured light technique for surface reconstruction of the human trunk. In *Electrical and Computer Engineering, 1998. IEEE Canadian Conference on* (1998), vol. 2, IEEE, pp. 874–877.

[63] FANELLO, S. R., RHEMANN, C., TANKOVICH, V., KOWDLE, A., ESCOLANO, S. O., KIM, D., AND IZADI, S. Hyperdepth: Learning depth from structured light without matching. 5441–5450.

[64] FIGUEIREDO, R., DEHBAN, A., MORENO, P., BERNARDINO, A., SANTOS-VICTOR, J., AND ARAÚJO, H. A robust and efficient framework for fast cylinder detection. *Robotics and Autonomous Systems 117* (2019), 17–28.

[65] FISCHLER, M. A., AND BOLLES, R. C. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM 24*, 6 (1981), 381–395.

[66] FOREST, J., AND SALVI, J. A review of laser scanning three-dimensional digitisers. In *Intelligent Robots and Systems, 2002. IEEE/RSJ International Conference on* (2002), vol. 1, IEEE, pp. 73–78.

[67] FORLANI, G., NARDINOCCHI, C., SCAIONI, M., AND ZINGARETTI, P. Complete classification of raw lidar data and 3d reconstruction of buildings. *Pattern analysis and applications 8*, 4 (2006), 357–374.

[68] FREDRICKSEN, H. A survey of full length nonlinear shift register cycle algorithms. *SIAM review 24*, 2 (1982), 195–221.

[69] GATYS, L., ECKER, A. S., AND BETHGE, M. Texture synthesis using convolutional neural networks. In *Advances in neural information processing systems* (2015), pp. 262–270.

[70] GENG, Z. J. Rainbow three-dimensional camera: new concept of high-speed three-dimensional vision systems. *Optical Engineering 35*, 2 (1996), 376–384.

[71] GODIN, G., BERALDIN, J.-A., PICARD, M., TAYLOR, J., COURNOYER, L., RIOUX, M., EL-HAKIM, S., BARIBEAU, R., BLAIS, F., BOULANGER, P., ET AL. Active optical 3d imaging for heritage applications. *IEEE Computer Graphics and Applications*, 5 (2002), 24–36.

[72] GORON, L. C., MARTON, Z.-C., LAZEA, G., AND BEETZ, M. Robustly segmenting cylindrical and box-like objects in cluttered scenes using depth cameras. In *ROBOTIK 2012; 7th German Conference on Robotics* (2012), VDE, pp. 1–6.

[73] HAGGAG, H., HOSSNY, M., NAHAVANDI, S., AND HAGGAG, O. An adaptable system for rgb-d based human body detection and pose estimation: Incorporating attached props. In *2016 IEEE International Conference on Systems, Man, and Cybernetics (SMC)* (2016), IEEE, pp. 001544–001549.

[74] HAKALA, D., HILLYARD, R., AND MALRAISON, P. J. Natural quadrics in mechanical design. In *CAD/CAM VIII part of Autofact West Anaheim* (California, 1980).

[75] HALIR, R., AND FLUSSER, J. Numerically stable direct least squares fitting of ellipses. In *Proc. 6th International Conference in Central Europe on Computer Graphics and Visualization. WSCG* (1998), vol. 98, Citeseer, pp. 125–132.

[76] HAN, J., SHAO, L., XU, D., AND SHOTTON, J. Enhanced computer vision with microsoft kinect sensor: A review. *IEEE Transactions on Cybernetics 43*, 5 (2013), 1318–1334.

[77] HANSARD, M., LEE, S., CHOI, O., AND HORAUD, R. P. *Time-of-flight cameras: principles, methods and applications*. Springer Science & Business Media, 2012.

[78] HANSEN, K., PEDERSEN, J., SØLUND, T., AANÆS, H., AND KRAFT, D. A structured light scanner for hyper flexible industrial automation. In *2014 2nd International Conference on 3D Vision* (2014), vol. 1, IEEE, pp. 401–408.

[79] HARTLEY, R., AND ZISSERMAN, A. *Multiple view geometry in computer vision*. Cambridge university press, 2003.

[80] HEIKKILA, J. Geometric camera calibration using circular control points. *IEEE Transactions on pattern analysis and machine intelligence 22*, 10 (2000), 1066–1077.

[81] HOCKEN, R. J., AND PEREIRA, P. H. *Coordinate measuring machines and systems*. CRC Press, 2016.

[82] HORN, B. K. Closed-form solution of absolute orientation using unit quaternions. *Josa a 4*, 4 (1987), 629–642.

[83] HORN, E., AND KIRYATI, N. Toward optimal structured light patterns1. *Image and Vision Computing 17*, 2 (1999), 87–97.

[84] HU, Q., HUANG, P. S., FU, Q., AND CHIANG, F.-P. Calibration of a three-dimensional shape measurement system. *Optical Engineering 42*, 2 (2003), 487–494.

[85] HUANG, L., CHUA, P. S., AND ASUNDI, A. Least-squares calibration method for fringe projection profilometry considering camera lens distortion. *Applied optics 49*, 9 (2010), 1539–1548.

[86] HUANG, Q.-X., FLÖRY, S., GELFAND, N., HOFER, M., AND POTTMANN, H. Reassembling fractured objects by geometric matching. *ACM Transactions on Graphics (TOG) 25*, 3 (2006), 569–578.

[87] HUANG, Z., FAN, J., YI, S., WANG, X., AND LI, H. Hms-net: Hierarchical multi-scale sparsity-invariant network for sparse depth completion. *arXiv preprint arXiv:1808.08685* (2018).

[88] HUGLI, H., AND MAITRE, G. Generation and use of color pseudo random sequences for coding structured light in active ranging. In *Industrial Inspection* (1989), vol. 1010, International Society for Optics and Photonics, pp. 75–83.

[89] HUNTLEY, J. M., AND SALDNER, H. Temporal phase-unwrapping algorithm for automated interferogram analysis. *Applied Optics 32*, 17 (1993), 3047–3052.

[90] INOKUCHI, S. Range-imaging system for 3d object recognition. In *Proc. of 7th International Conference on Pattern Recognition, 1984* (1984).

[91] ISHIYAMA, R., SAKAMOTO, S., TAJIMA, J., OKATANI, T., AND DEGUCHI, K. Absolute phase measurements using geometric constraints between multiple cameras and projectors. *Applied optics 46*, 17 (2007), 3528–3538.

[92] JARITZ, M., DE CHARETTE, R., WIRBEL, E., PERROTTON, X., AND NASHASHIBI, F. Sparse and dense data with cnns: Depth completion and semantic segmentation. In *2018 International Conference on 3D Vision (3DV)* (2018), IEEE, pp. 52–60.

[93] KAISER, A., YBANEZ ZEPEDA, J. A., AND BOUBEKEUR, T. A survey of simple geometric primitives detection methods for captured 3d data. In *Computer Graphics Forum* (2019), vol. 38, Wiley Online Library, pp. 167–196.

[94] KANAI, T., SUZUKI, H., MITANI, J., AND KIMURA, F. Interactive mesh fusion based on local 3d metamorphosis. In *Graphics interface* (1999), vol. 99, pp. 148–156.

[95] KAVAN, L., COLLINS, S., O'SULLIVAN, C., AND ZARA, J. Dual quaternions for rigid transformation blending. *Trinity College Dublin, Tech. Rep. TCD-CS-2006–46* (2006).

[96] KAVAN, L., COLLINS, S., ŽÁRA, J., AND O'SULLIVAN, C. Geometric skinning with approximate dual quaternion blending. *ACM Trans. Graph. 27*, 4 (Nov. 2008), 105:1–105:23.

[97] KAVAN, L., COLLINS, S., AND ZARA, J. Dual quaternions for rigid transformation blending. Tech. rep., 2006.

[98] KAWASAKI, H., FURUKAWA, R., SAGAWA, R., AND YAGI, Y. Dynamic scene shape reconstruction using a single structured light pattern. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on* (2008), IEEE, pp. 1–8.

[99] KAZHDAN, M., BOLITHO, M., AND HOPPE, H. Poisson surface reconstruction. In *Proceedings of the fourth Eurographics symposium on Geometry processing* (2006), vol. 7.

[100] KAZHDAN, M., AND HOPPE, H. Screened poisson surface reconstruction. *ACM Transactions on Graphics (ToG) 32*, 3 (2013), 29.

[101] KENNER, F. Sonnenuhren aus aquileia, mittheilungen der k.k. central-commission zur erfortschung und erhaltung der kunst und historischen denkmale. *VI Jahrgang, Neue Folge, Wien* (1880).

[102] KOBAYASHI, T., HIGO, T., YAMASAKI, M., KOBAYASHI, K., AND KATAYAMA, A. Accurate and practical 3d measurement for translucent objects by dashed lines and complementary gray code projection. In *2015 International Conference on 3D Vision* (2015), IEEE, pp. 189–197.

[103] KOKKU, R., AND BROOKSBY, G. Improving 3d surface measurement accuracy on metallic surfaces. In *Optical Measurement Systems for Industrial Inspection IV* (2005), vol. 5856, International Society for Optics and Photonics, pp. 618–624.

[104] KOUTSOUDIS, A., ARNAOUTOGLOU, F., AND CHAMZAS, C. On 3d reconstruction of the old city of xanthi. a minimum budget approach to virtual touring based on photogrammetry. *Journal of Cultural Heritage 8*, 1 (2007), 26–31.

[105] KU, J., HARAKEH, A., AND WASLANDER, S. L. In defense of classical image processing: Fast depth completion on the cpu. In *2018 15th Conference on Computer and Robot Vision (CRV)* (2018), IEEE, pp. 16–22.

[106] KUMAR, A. Computer-vision-based fabric defect detection: A survey. *IEEE transactions on industrial electronics 55*, 1 (2008), 348–363.

[107] KURZ, G., GILITSCHENSKI, I., AND HANEBECK, U. D. Efficient evaluation of the probability density function of a wrapped normal distribution. 1–5.

[108] LAGARIAS, J. C., REEDS, J. A., WRIGHT, M. H., AND WRIGHT, P. E. Convergence properties of the nelder–mead simplex method in low dimensions. *SIAM Journal on optimization 9*, 1 (1998), 112–147.

[109] LANCASTER, L. C. *Concrete vaulted construction in Imperial Rome: innovations in context*. Cambridge University Press, 2005.

[110] LANGE, R., AND SEITZ, P. Solid-state time-of-flight range camera. *IEEE Journal of quantum electronics 37*, 3 (2001), 390–397.

[111] LAZAROS, N., SIRAKOULIS, G. C., AND GASTERATOS, A. Review of stereo vision algorithms: from software to hardware. *International Journal of Optomechatronics 2*, 4 (2008), 435–462.

[112] LE MOIGNE, J., AND WAXMAN, A. M. Structured light patterns for robot mobility. *IEEE Journal on Robotics and Automation 4*, 5 (1988), 541–548.

[113] LEE, T.-C., KASHYAP, R. L., AND CHU, C.-N. Building skeleton models via 3-d medial surface axis thinning algorithms. *CVGIP: Graphical Models and Image Processing 56*, 6 (1994), 462–478.

[114] LEI, S., AND ZHANG, S. Flexible 3-d shape measurement using projector defocusing. *Optics letters 34*, 20 (2009), 3080–3082.

[115] LI, B., AN, Y., CAPPELLERI, D., XU, J., AND ZHANG, S. High-accuracy, high-speed 3d structured light imaging techniques and potential applications to intelligent robotics. *International Journal of Intelligent Robotics and Applications 1*, 1 (2017), 86–103.

[116] LI, B., BELL, T., AND ZHANG, S. Computer-aided-design-model-assisted absolute three-dimensional shape measurement. *Applied optics 56*, 24 (2017), 6770–6776.

[117] LI, B., KARPINSKY, N., AND ZHANG, S. Novel calibration method for structured-light system with an out-of-focus projector. *Applied optics 53*, 16 (2014), 3415–3426.

[118] LI, B., AND ZHANG, S. Flexible calibration method for microscopic structured light system using telecentric lens. *Optics express 23*, 20 (2015), 25795–25803.

[119] LI, B., AND ZHANG, S. Microscopic structured light 3d profilometry: Binary defocusing technique vs. sinusoidal fringe projection. *Optics and Lasers in Engineering 96* (2017), 117–123.

[120] LI, D., LIU, C., AND TIAN, J. Telecentric 3d profilometry based on phase-shifting fringe projection. *Optics express 22*, 26 (2014), 31826–31835.

[121] LI, L., SUNG, M., DUBROVINA, A., YI, L., AND GUIBAS, L. J. Supervised fitting of geometric primitives to 3d point clouds. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2019), pp. 2652–2660.

[122] LI, R., LUO, T., AND ZHA, H. 3d digitization and its applications in cultural heritage. In *Euro-Mediterranean Conference* (2010), Springer, pp. 381–388.

[123] LI, Z., ZHONG, K., LI, Y. F., ZHOU, X., AND SHI, Y. Multiview phase shifting: a full-resolution and high-speed 3d measurement framework for arbitrary shape dynamic objects. *Optics letters 38*, 9 (2013), 1389–1391.

[124] LIAO, Y., DONNE, S., AND GEIGER, A. Deep marching cubes: Learning explicit surface representations. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2018), pp. 2916–2925.

[125] LILIENBLUM, E., AND MICHAELIS, B. Optical 3d surface reconstruction by a multi-period phase shift method. *JCP 2*, 2 (2007), 73–83.

[126] LIU, C., CHEN, L., HE, X., THANG, V. D., AND KOFIDIS, T. Coaxial projection profilometry based on speckle and fringe projection. *Optics Communications 341* (2015), 228–236.

[127] LIU, H., LIN, H., AND YAO, L. Calibration method for projector-camera-based telecentric fringe projection profilometry system. *Optics express 25*, 25 (2017), 31492–31508.

[128] LIU, Y.-J., ZHANG, J.-B., HOU, J.-C., REN, J.-C., AND TANG, W.-Q. Cylinder detection in large-scale point cloud of pipeline plant. *IEEE transactions on visualization and computer graphics 19*, 10 (2013), 1700–1707.

[129] LIU, Z., ZHANG, Z., AND SHAN, Y. Image-based surface detail transfer. *IEEE Computer Graphics and Applications 24*, 3 (2004), 30–35.

[130] LORENSEN, W. E., AND CLINE, H. E. Marching cubes: A high resolution 3d surface construction algorithm. In *ACM siggraph computer graphics* (1987), vol. 21, ACM, pp. 163–169.

[131] LOWE, D. G. Object recognition from local scale-invariant features. In *Computer vision, 1999. The proceedings of the seventh IEEE international conference on* (1999), vol. 2, Ieee, pp. 1150–1157.

[132] LOWE, D. G. Distinctive image features from scale-invariant keypoints. *International journal of computer vision 60*, 2 (2004), 91–110.

[133] LUHMANN, T. Close range photogrammetry for industrial applications. *ISPRS Journal of Photogrammetry and Remote Sensing 65*, 6 (2010), 558–569.

[134] LUHMANN, T. Close range photogrammetry for industrial applications. *ISPRS Journal of Photogrammetry and Remote Sensing 65*, 6 (2010), 558–569.

[135] LUKÁCS, G., MARSHALL, A. D., AND MARTIN, R. R. Geometric least-squares fitting of spheres , cylinders , cones and tori.

[136] LUKÁCS, G., MARTIN, R., AND MARSHALL, D. Faithful least-squares fitting of spheres, cylinders, cones and tori for reliable segmentation. In *Computer Vision - ECCV'98* (1998), pp. 671–686.

[137] LYNCH, C. Digital collections, digital libraries & the digitization of cultural heritage information. *Microform & imaging review 31*, 4 (2002), 131–145.

[138] MAALEK, R., LICHTI, D. D., WALKER, R., BHAVNANI, A., AND RUWANPURA, J. Y. Extraction of pipes and flanges from point clouds for automated verification of pre-fabricated modules in oil and gas refinery projects. *Automation in Construction 103* (2019), 150–167.

[139] MACLEOD, M. D. Fast nearly ml estimation of the parameters of real or complex single tones or resolved multiple tones. *IEEE Transactions on Signal Processing 46*, 1 (Jan 1998), 141–148.

[140] MCCARTHY, J. M. *Introduction to Theoretical Kinematics*. MIT Press, Cambridge, MA, USA, 1990.

[141] MIGDAL, A., PETROV, M., LEBEDEV, A., SHELYEKHOVA, V., ABADJEV, V., BERNSTEIN, V., AND AFANASSENKOV, A. Structured-light, triangulation-based three-dimensional digitizer, Apr. 15 2003. US Patent 6,549,288.

[142] MILLER, A. T., KNOOP, S., CHRISTENSEN, H. I., AND ALLEN, P. K. Automatic grasp planning using shape primitives.

[143] MIYASAKA, T., KURODA, K., HIROSE, M., AND ARAKI, K. High speed 3-d measurement system using incoherent light source for human performance analysis. *INTERNATIONAL ARCHIVES OF PHOTOGRAMMETRY AND REMOTE SENSING 33*, B5/2; PART 5 (2000), 547–551.

[144] MONTI, F., BOSCAINI, D., MASCI, J., RODOLA, E., SVOBODA, J., AND BRONSTEIN, M. M. Geometric deep learning on graphs and manifolds using mixture model cnns. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2017), pp. 5115–5124.

[145] MORENO, D., AND TAUBIN, G. Simple, accurate, and robust projector-camera calibration. In *3DIMPVT 2012* (2012), pp. 464–471.

[146] MÜLLER, G., BENDELS, G. H., AND KLEIN, R. Rapid synchronous acquisition of geometry and appearance of cultural heritage artefacts. In *VAST* (2005), pp. 13–20.

[147] NGUYEN, M. X., YUAN, X., AND CHEN, B. Geometry completion and detail generation by texture synthesis. *The Visual Computer 21*, 8-10 (2005), 669–678.

[148] NURUNNABI, A., SADAHIRO, Y., LINDENBERGH, R., AND BELTON, D. Robust cylinder fitting in laser scanning point cloud data. *Measurement 138* (2019), 632–651.

[149] OIKONOMIDIS, I., KYRIAZIS, N., AND ARGYROS, A. A. Efficient model-based 3d tracking of hand articulations using kinect. In *BmVC* (2011), vol. 1, p. 3.

[150] OSTEN, W., NADEBORN, W., AND ANDRAE, P. General hierarchical approach in absolute phase measurement. In *Laser Interferometry VIII: Techniques and Analysis* (1996), vol. 2860, International Society for Optics and Photonics, pp. 2–14.

[151] PALÁNCZ, B., AWANGE, J., SOMOGYI, A., REHÁNY, N., LOVAS, T., MOLNÁR, B., AND FUKUDA, Y. A robust cylindrical fitting to point cloud data. *Australian Journal of Earth Sciences 63*, 5 (2016), 665–673.

[152] PATIL, A. K., HOLI, P., LEE, S. K., AND CHAI, Y. H. An adaptive approach for the reconstruction and modeling of as-built 3d pipelines from point clouds. *Automation in construction 75* (2017), 65–78.

[153] PAVLIDIS, G., KOUTSOUDIS, A., ARNAOUTOGLOU, F., TSIOUKAS, V., AND CHAMZAS, C. Methods for 3d digitization of cultural heritage. *Journal of cultural heritage 8*, 1 (2007), 93–98.

[154] PELLOT, C., HERMENT, A., SIGELLE, M., HORAIN, P., MAÎTRE, H., AND PERONNEAU, P. A 3d reconstruction of vascular structures from two x-ray angiograms using an adapted simulated annealing algorithm. *IEEE transactions on medical imaging 13*, 1 (1994), 48–60.

[155] PÉREZ, L., RODRÍGUEZ, Í., RODRÍGUEZ, N., USAMENTIAGA, R., AND GARCÍA, D. Robot guidance using machine vision techniques in industrial environments: A comparative review. *Sensors 16*, 3 (2016), 335.

[156] PETRIU, E. M., BIESEMAN, T., TRIF, N., MCMATH, W. S., AND YEUNG, S. Visual object recognition using pseudo-random grid encoding. In *Intelligent Robots and Systems, 1992., Proceedings of the 1992 lEEE/RSJ International Conference on* (1992), vol. 3, IEEE, pp. 1617–1624.

[157] PFEIFER, N., GORTE, B., AND WINTERHALDER, D. Automatic reconstruction of single trees from terrestrial laser scanner data.

[158] PIERACCINI, M., GUIDI, G., AND ATZENI, C. 3d digitizing of cultural heritage. *Journal of Cultural Heritage 2*, 1 (2001), 63–70.

[159] PISTELLATO, M., ALBARELLI, A., BERGAMASCO, F., AND TORSELLO, A. Robust joint selection of camera orientations and feature projections over multiple views. In *ICPR 2016* (2017), pp. 3703–3708.

[160] PISTELLATO, M., BERGAMASCO, F., ALBARELLI, A., COSMO, L., GASPARETTO, A., AND TORSELLO, A. Robust phase unwrapping by probabilistic consensus. *Optics and Lasers in Engineering 121* (2019), 428–440.

[161] PISTELLATO, M., BERGAMASCO, F., ALBARELLI, A., COSMO, L., GASPARETTO, A., AND TORSELLO, A. Stochastic phase estimation and unwrapping.

[162] PISTELLATO, M., BERGAMASCO, F., ALBARELLI, A., AND TORSELLO, A. Robust cylinder estimation in point clouds from pairwise axes similarities.

[163] PISTELLATO, M., BERGAMASCO, F., COSMO, L., GASPARETTO, A., RESSI, D., AND ALBARELLI, A. Neighborhood-based recovery of phase unwrapping faults. In *2018 24th International Conference on Pattern Recognition (ICPR)* (2018), IEEE, pp. 2462–2467.

[164] PISTELLATO, M., COSMO, L., BERGAMASCO, F., GASPARETTO, A., AND ALBARELLI, A. Adaptive albedo compensation for accurate phase-shift coding. In *2018 24th International Conference on Pattern Recognition (ICPR)* (2018), IEEE, pp. 2450–2455.

[165] POLEWSKI, P., YAO, W., HEURICH, M., KRZYSTEK, P., AND STILLA, U. A voting-based statistical cylinder detection framework applied to fallen tree mapping in terrestrial laser scanning point clouds. *ISPRS Journal of Photogrammetry and Remote Sensing 129* (2017), 118–130.

[166] POLLEFEYS, M., VAN GOOL, L., AND OOSTERLINCK, A. The modulus constraint: a new constraint self-calibration. In *Proceedings of 13th International Conference on Pattern Recognition* (1996), vol. 1, IEEE, pp. 349–353.

[167] PRITT, M. D., AND GHIGLIA, D. C. *Two-dimensional phase unwrapping: theory, algorithms, and software*. Wiley, 1998.

[168] QI, C. R., SU, H., MO, K., AND GUIBAS, L. J. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2017), pp. 652–660.

[169] QIU, R., ZHOU, Q.-Y., AND NEUMANN, U. Pipe-run extraction and reconstruction from point clouds. In *Computer Vision – ECCV 2014* (2014), pp. 17–30.

[170] RABBANI, T., AND DEN HEUVEL, F. A. V. Efficient hough transform for automatic detection of cylinders in point clouds.

[171] RAHAYEM, M., WERGHI, N., AND KJELLANDER, J. Best ellipse and cylinder parameters estimation from laser profile scan sections. *Optics and Lasers in Engineering 50*, 9 (2012), 1242 – 1259.

[172] RAHAYEM, M. R., AND KJELLANDER, J. A. P. Quadric segmentation and fitting of data captured by a laser profile scanner mounted on an industrial robot. *The International Journal of Advanced Manufacturing Technology 52*, 1 (Jan 2011), 155–169.

[173] REDA, I., AND ANDREAS, A. Solar position algorithm for solar radiation applications. *Solar energy 76*, 5 (2004), 577–589.

[174] REMONDINO, F., AND EL-HAKIM, S. Image-based 3d modelling: A review. *The Photogrammetric Record 21*, 115 (2006), 269–291.

[175] RIFE, D., AND BOORSTYN, R. Single tone parameter estimation from discrete-time observations. *IEEE Transactions on Information Theory 20*, 5 (Sep 1974), 591–598.

[176] ROSENDAHL, S., HÄLLSTIG, E., GREN, P., AND SJÖDAHL, M. Shape measurement with one fringe pattern recording including a digital master. *Applied Optics 49*, 14 (2010), 2622–2629.

[177] RUSU, R. B., BLODOW, N., AND BEETZ, M. Fast point feature histograms (fpfh) for 3d registration. In *2009 IEEE International Conference on Robotics and Automation* (2009), IEEE, pp. 3212–3217.

[178] SAEED, G., AND ZHANG, Y. Weld pool surface depth measurement using a calibrated camera and structured light. *Measurement Science and Technology 18*, 8 (2007), 2570.

[179] SALVI, J., ARMANGUÉ, X., AND BATLLE, J. A comparative review of camera calibrating methods with accuracy evaluation. *Pattern recognition 35*, 7 (2002), 1617–1635.

[180] SALVI, J., FERNANDEZ, S., PRIBANIC, T., AND LLADO, X. A state of the art in structured light patterns for surface profilometry. *Pattern recognition 43*, 8 (2010), 2666–2680.

[181] SALVI, J., PAGES, J., AND BATLLE, J. Pattern codification strategies in structured light systems. *Pattern recognition 37*, 4 (2004), 827–849.

[182] SANSONI, G., CAROCCI, M., AND RODELLA, R. Three-dimensional vision based on a combination of gray-code and phase-shift light projection: analysis and compensation of the systematic errors. *Applied optics 38*, 31 (1999), 6565–6573.

[183] SANTAGATI, C., INZERILLO, L., AND DI PAOLA, F. Image-based modeling techniques for architectural heritage 3d digitalization: limits and potentialities. *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences 5*, w2 (2013), 555–560.

[184] SATO, T. Multispectral pattern projection range finder. In *Three-Dimensional Image Capture and Applications II* (1999), vol. 3640, International Society for Optics and Photonics, pp. 28–37.

[185] SAVAGE, C. A survey of combinatorial gray codes. *SIAM review 39*, 4 (1997), 605–629.

[186] SCHARSTEIN, D., AND SZELISKI, R. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *International journal of computer vision 47*, 1-3 (2002), 7–42.

[187] SCHNABEL, R., WAHL, R., AND KLEIN, R. Efficient ransac for point-cloud shape detection. In *Computer graphics forum* (2007), vol. 26, Wiley Online Library, pp. 214–226.

[188] SEDERBERG, T. W., AND PARRY, S. R. Free-form deformation of solid geometric models. *ACM SIGGRAPH computer graphics 20*, 4 (1986), 151–160.

[189] SHAH, S., AND AGGARWAL, J. Intrinsic parameter calibration procedure for a (high-distortion) fish-eye lens camera with distortion model and accuracy estimation. *Pattern Recognition 29*, 11 (1996), 1775–1788.

[190] SHAHPASKI, M., SAPAICO, L. R., CHEVASSUS, G., AND SÜSSTRUNK, S. Simultaneous geometric and radiometric calibration of a projector-camera pair. In *CVPR 2017* (2017).

[191] SHARF, A., ALEXA, M., AND COHEN-OR, D. Context-based surface completion. In *ACM Transactions on Graphics (TOG)* (2004), vol. 23, ACM, pp. 878–887.

[192] SHIH, N.-J., WANG, H.-J., LIN, C.-Y., AND LIAU, C.-Y. 3d scan for the digital preservation of a historical temple in taiwan. *Advances in engineering software 38*, 7 (2007), 501–512.

[193] SITNIK, R., KRZESLOWSKI, J. F., AND MACZKOWSKI, G. Archiving shape and appearance of cultural heritage objects using structured light projection and multispectral imaging. *Optical Engineering 51*, 2 (2012), 021115.

[194] SKOCAJ, D., AND LEONARDIS, A. Range image acquisition of objects with non-uniform albedo using structured light range sensor. In *Proceedings 15th International Conference on Pattern Recognition. ICPR-2000* (2000), vol. 1, IEEE, pp. 778–781.

[195] SORKINE, O., COHEN-OR, D., LIPMAN, Y., ALEXA, M., RÖSSL, C., AND SEIDEL, H.-P. Laplacian surface editing. In *Proceedings of the 2004 Eurographics/ACM SIGGRAPH symposium on Geometry processing* (2004), ACM, pp. 175–184.

[196] SPECIAN, A., MUCCHIANI, C., YIM, M., AND SEO, J. Robotic edge-rolling manipulation: A grasp planning approach. *IEEE Robotics and Automation Letters 3*, 4 (Oct 2018), 3137–3144.

[197] SRINIVASAN, V., LIU, H. C., AND HALIOUA, M. Automated phase-measuring profilometry of 3-d diffuse objects. *Appl. Opt. 23*, 18 (Sep 1984), 3105–3108.

[198] STANCO, F., BATTIATO, S., AND GALLO, G. *Digital imaging for cultural heritage preservation: Analysis, restoration, and reconstruction of ancient artworks*. CRC Press, 2011.

[199] STEIN, J. S., HOLMGREN, W. F., FORBESS, J., AND HANSEN, C. W. Pvlib: Open source photovoltaic performance modeling functions for matlab and python. In *2016 ieee 43rd photovoltaic specialists conference (pvsc)* (2016), IEEE, pp. 3425–3430.

[200] SU, X., AND CHEN, W. Reliability-guided phase unwrapping algorithm: a review. *Optics and Lasers in Engineering 42*, 3 (2004), 245–261.

[201] SURREL, Y. Design of algorithms for phase measurements by the use of phase stepping. *Applied optics 35*, 1 (1996), 51–60.

[202] TAJIMA, J., AND IWAKAWA, M. 3-d data acquisition by rainbow range finder. In *[1990] Proceedings. 10th International Conference on Pattern Recognition* (1990), vol. 1, IEEE, pp. 309–313.

[203] TAKEDA, M. Fourier fringe analysis and its application to metrology of extreme physical phenomena: a review. *Applied optics 52*, 1 (2013), 20–29.

[204] TAYLOR, P. D., AND JONKER, L. B. Evolutionarily stable strategies and game dynamics.

[205] TCHAPMI, L., CHOY, C., ARMENI, I., GWAK, J., AND SAVARESE, S. Segcloud: Semantic segmentation of 3d point clouds. In *2017 International Conference on 3D Vision (3DV)* (2017), IEEE, pp. 537–547.

[206] TEICHMAN, A., LEVINSON, J., AND THRUN, S. Towards 3d object recognition via classification of arbitrary object tracks. In *2011 IEEE International Conference on Robotics and Automation* (2011), IEEE, pp. 4034–4041.

[207] TIKUISIS, P., MEUNIER, P., AND JUBENVILLE, C. Human body surface area: measurement and prediction using three dimensional body scans. *European journal of applied physiology 85*, 3 (2001), 264–271.

[208] TOLDO, R., AND FUSIELLO, A. Robust multiple structures estimation with j-linkage. In *European conference on computer vision* (2008), Springer, pp. 537–547.

[209] TORR, P. H. S., AND ZISSERMAN, A. MLESAC: A new robust estimator with application to estimating image geometry. *CVIU 78* (2000), 138–156.

[210] TORSELLO, A., RODOLA, E., AND ALBARELLI, A. Multiview registration via graph diffusion of dual quaternions. In *CVPR 2011* (2011), IEEE, pp. 2441–2448.

[211] TRAN, T.-T., CAO, V.-T., AND LAURENDEAU, D. Extraction of cylinders and estimation of their parameters from point clouds. *Computers & Graphics 46* (2015), 345–357.

[212] TRIGGS, B. Autocalibration from planar scenes. In *European conference on computer vision* (1998), Springer, pp. 89–105.

[213] TROBINA, M. Error model of a coded-light range sensor. *Technical report* (1995).

[214] TSAI, M.-J., AND HUNG, C.-C. Development of a high-precision surface metrology system using structured light projection. *Measurement 38*, 3 (2005), 236–247.

[215] UHRIG, J., SCHNEIDER, N., SCHNEIDER, L., FRANKE, U., BROX, T., AND GEIGER, A. Sparsity invariant cnns. In *2017 International Conference on 3D Vision (3DV)* (2017), IEEE, pp. 11–20.

[216] VO, M., NARASIMHAN, S. G., AND SHEIKH, Y. Texture illumination separation for single-shot structured light reconstruction. *IEEE Trans. Pattern Anal. Mach. Intell. 38*, 2 (Feb. 2016), 390–404.

[217] WECKENMANN, A., PEGGS, G., AND HOFFMANN, J. Probing systems for dimensional micro-and nano-metrology. *Measurement Science and Technology 17*, 3 (2006), 504.

[218] WEI, L.-Y., LEFEBVRE, S., KWATRA, V., AND TURK, G. State of the art in example-based texture synthesis.

[219] WEIBULL, J. *Evolutionary game theory*. MIT Press, Cambridge, Mass. [u.a.], 1995.

[220] WENG, J., COHEN, P., AND HERNIOU, M. Camera calibration with distortion models and accuracy evaluation. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, 10 (1992), 965–980.

[221] WINTERHALTER, W., FLECKENSTEIN, F., STEDER, B., SPINELLO, L., AND BURGARD, W. Accurate indoor localization for rgb-d smartphones and tablets given 2d floor plans. vol. 2015-December, pp. 3138–3143.

[222] XU, J., XI, N., ZHANG, C., SHI, Q., AND GREGORY, J. Real-time 3d shape inspection system of automotive parts based on structured light pattern. *Optics & Laser Technology 43*, 1 (2011), 1–8.

[223] XU, J., XI, N., ZHANG, C., ZHAO, J., GAO, B., AND SHI, Q. Rapid 3d surface profile measurement of industrial parts using two-level structured light patterns. *Optics and Lasers in Engineering 49*, 7 (2011), 907–914.

[224] XU, X., WANG, Y., AND LIU, K. Error reduction for the high-contrast regions in phase measuring profilometry. In *IEEE China Summit and Int. Conf. on Signal and Information Processing* (July 2013), pp. 72–76.

[225] YAO, M., AND XU, B. Evaluating wrinkles on laminated plastic sheets using 3d laser scanning. *Measurement Science and Technology 18*, 12 (2007), 3724.

[226] ZAPPA, E., AND BUSCA, G. Fourier-transform profilometry calibration based on an exhaustive geometric model of the system. *Optics and Lasers in Engineering 47*, 7-8 (2009), 754–767.

[227] ZHANG, S. Comparative study on passive and active projector nonlinear gamma calibration. *Appl. Optics 54*, 13 (2015), 3834–3841.

[228] ZHANG, S. Absolute phase retrieval methods for digital fringe projection profilometry: a review. *Optics and Lasers in Engineering 107* (2018), 28–37.

[229] ZHANG, S. High-speed 3d shape measurement with structured light methods: A review. *Optics and Lasers in Engineering 106* (2018), 119–131.

[230] ZHANG, S., AND HUANG, P. S. Novel method for structured light system calibration. *Optical Engineering 45*, 8 (2006), 083601.

[231] ZHANG, Z. Iterative point matching for registration of free-form curves and surfaces. *International journal of computer vision 13*, 2 (1994), 119–152.

[232] ZHANG, Z. A flexible new technique for camera calibration. *IEEE Transactions on pattern analysis and machine intelligence 22* (2000).

[233] ZHAO, M., HUANG, L., ZHANG, Q., SU, X., ASUNDI, A., AND KEMAO, Q. Quality-guided phase unwrapping technique: comparison of quality maps and guiding strategies. *Applied optics 50*, 33 (2011), 6214–6224.

[234] ZHENG, D., AND DA, F. Self-correction phase unwrapping method based on gray-code light. *Optics and Lasers in Engineering 50*, 8 (2012), 1130 – 1139.

[235] ZHENG, D., DA, F., KEMAO, Q., AND SEAH, H. S. Phase-shifting profilometry combined with gray-code patterns projection: unwrapping error removal by an adaptive median filter. *Opt. Express 25*, 5 (Mar 2017), 4700–4713.

[236] ZHONG, J., AND WANG, M. Phase unwrapping by lookup table method: application to phase map with singular points. *Optical Engineering 38*, 12 (1999), 2075–2081.

[237] ZHOU, P., ZHU, J., SU, X., JING, H., AND ZHANG, X. Three-dimensional shape measurement using color random binary encoding pattern projection. *Optical Engineering 56*, 10 (2017), 104102.

[238] ZUO, C., FENG, S., HUANG, L., TAO, T., YIN, W., AND CHEN, Q. Phase shifting algorithms for fringe projection profilometry: A review. *Optics and Lasers in Engineering 109* (2018), 23–59.

[239] ZUO, C., HUANG, L., ZHANG, M., CHEN, Q., AND ASUNDI, A. Temporal phase unwrapping algorithms for fringe projection profilometry: A comparative review. *Optics and Lasers in Engineering 85* (2016), 84–103.