Robust Phase Unwrapping by Probabilistic Consensus

Mara Pistellato[a,*], Filippo Bergamasco[a], Andrea Albarelli[a], Luca Cosmo[a], Andrea Gasparetto[a], Andrea Torsello[a]

[a] *DAIS, Università Ca'Foscari, Via Torino 155, 30172 Venice, Italy*

Abstract

Structured light scanning works by projecting a supplement of controlled information on the scene: the captured signal is processed to provide a unique label (namely a code) for each observed point, and then proceed to geometrical triangulation. In phase shift profilometry sinusoidal patterns are projected so that each point is labelled according to the observed phase. Given the periodic nature of the signal, a disambiguation method (known as phase unwrapping) must be provided. Several unwrapping techniques have been proposed in the literature, since noisy signals lead to inaccuracies in phase estimation. This paper presents a novel phase unwrapping approach based on a probabilistic framework: such method involves the projection of multiple sinusoids with distinct period lengths, so that in each point different phase values are observed. In the proposed technique phase values are 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 compute codes mutual support in a spatial neighbourhood in order to identify and correct faulty unwrappings. An extensive experimental assessment validates the Gaussian distribution hypothesis and verifies the improvements

*Corresponding author

Email addresses: mara.pistellato@unive.it (Mara Pistellato), filippo.bergamasco@unive.it (Filippo Bergamasco), albarelli@unive.it (Andrea Albarelli), luca.cosmo@unive.it (Luca Cosmo), andrea.gasparetto@unive.it (Andrea Gasparetto), andrea.torsello@unive.it (Andrea Torsello)

offered by our method in terms of coding accuracy when compared to other classical unwrapping techniques.

Keywords: Fringe Projection, Phase Unwrapping, Maximum Likelihood Estimation, 3D reconstruction, Structured Light

---

1. Introduction

Structured light scanning employs a projector to cast additional controlled information to the scene: specific light patterns are designed to encode a unique label in each observed point, that can be decoded when captured by one or more cameras. Such codes are uniquely mapped to pixel coordinates in projector's image plane, so they are used as correspondences to perform 3D reconstruction via geometric triangulation.

During the years, different coding approaches have been developed. As described in [1], we can classify structured light techniques by the way in which codes are embedded in the patterns, namely: neighbourhood, direct and time-multiplexing codification.

In neighbourhood based techniques all labels are encoded in a single pattern, and each code is computed as a function of the observed values 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, like [2], while others employ specific pseudo-random structures like De Bruijn sequences [3] or M-arrays [4]. Since a unique fixed pattern is projected on the scene, these approaches allow a fast acquisition, even in case of moving objects. Nevertheless, their spatial resolution is low if compared to other methods, errors are more frequent and code recovery could fail in some points if their neighbourhood is partially occluded.

Another class of encoding strategies consist in direct codification: in this technique each pixel independently encodes its own label by using different possibly repeated colours. Several approaches have been proposed, based both on greyscale [5] or coloured patterns [6]. In practice, these approaches are

extremely sensitive to noise and scene features (like illumination or albedo) and require a proper tuning step or multiple acquisitions. For these reasons, they are not suited for moving scenes or accurate measurements. Other single-pattern approaches, like the one proposed in [7] exploit geometrical properties of a single grid pattern to compute 3D data.

In time-multiplexing approaches, coded data are acquired through a sequence of patterns projected over time: the intensity values observed at each point during the projection are used to compute the final code. Graycoding [8] is a classical example of this technique, it consists in projecting a series of binary striped patterns, and each pixel is labelled with a binary code computed from the sequence of observed intensity values. Adjacent codes are designed to have a Hamming distance equal to one, preventing coding errors. This approach yields 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. Other similar temporal coding approaches employ n-ary codes with coloured patterns [9] or multiple grey levels [10].

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 [11]. This method is widely adopted because of its intrinsic robustness to noise and its ability to recover codes with sub-pixel accuracy, but since phase values are repeated across the scene, 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. Among all different techniques that have been proposed [12], we can distinguish two main categories: spatial and temporal phase unwrapping [1]

In spatial phase unwrapping the relative fringe shifts are computed by look-

[1] Note that there are also other kinds of approaches that exploit pre-known scene information, additional hardware or combinations with other methods.

ing at neighbour values in a single phase map [13]. 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 [14] and [15] a number of quality-guided algorithms and strategies are presented. Despite 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 could 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 [16]. 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 propose different kinds of additional signal, for instance a binary random pattern to perform stereo matching [17], or a coloured striped pattern [18]. Other 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 (see [19] for an extensive review and comparison). In literature, one of the first proposals consists in multi-wavelength phase unwrapping [20] [21]. This technique is also called phase difference and exploits two close frequencies to generate another phase map from their difference such as there are no ambiguities. Another technique is the multi-frequency or hier-

archical phase unwrapping [22] [23]: 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, that are denser. The technique described in [24] and [25] introduces a number-theoretical unwrapping algorithm which exploits look up tables that allow a fast computation. This algorithm offers good results in optimal conditions, but it is very sensitive to small phase inaccuracies and outliers, that lead to a table miss. Other phase unwrapping methods involve the usage of additional hardware like cameras [26], pre-knowledge of the scene such as CAD models [27] or the geometrical configuration of the system. In particular, the minimum phase method proposed in [28] exploits geometrical constraints to generate a virtual phase map at a known depth and exploits it to compute the unwrapped phase values.

As we discussed, in case of noisy acquisitions, many phase unwrapping approaches discard the measurements that do not fulfil some precision level. For instance, in number-theoretical methods the discrete look up table approach results in errors or missing codes, while in Graycoding all phases are used but only one (possibly noisy) phase map is exploited to devise the final unwrapped values. The technique introduced in this paper is designed to work in the opposite direction: we propose a robust multiple-period method that does not discard any acquired phase value. Our basic idea 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. 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. Moreover, we propose a simple yet effective recovery approach that exploits Likelihood values in combination with spatial neighbours to perform outlier detection and correction.

The rest of the paper is organized as follows: Section 2 introduces the basics

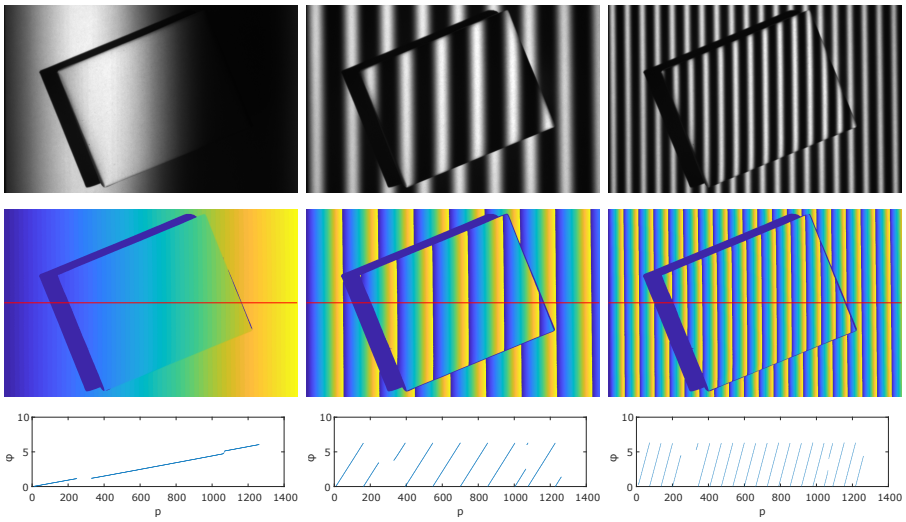

Figure 1: Acquisition process and different period lengths. First row shows one of the shifted light pattern projected on the scene, each one with a different period length (1300, 160 and 60 pixels). In the second row the correspondent phase maps are shown. Third row shows the computed phase values along the red line.

of phase shift, the error model we adopted and the notation. In Sections 3 and 4 we present our proposed method and finally in Section 5 an extensive both synthetic and real-world experiments are displayed.

2. Phase Shift

In phase shift the projected pattern is composed by vertical or horizontal sinusoidal fringes: these patterns are periodic in one direction and are characterized by a period length (in pixels). During acquisition the signal is shifted over time and intensity values are grabbed for each shift. The first row in Figure 1 shows three pictures of the scene while projecting one of the shifted light patterns. The different examples 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 which values consist in

the computed phase at that image coordinate.

In the following sections we briefly cover the basic Phase Shift method and introduce a well-defined set of fundamental definitions and symbols that will be used throughout the paper.

2.1. 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. In 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 of I_i at pixel coordinate (u, v). The phase value $\varphi(u, v)$ is computed as follows:

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

with

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

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

The result of this process are the phase values computed in each pixel, as shown in second row of Figure 1. 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 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 monotone. 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.

2.2. Projector Codes and Phases

Computing projector codes from an observed phase is indeed 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. (1) 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:

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

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 in ξ. We denote the function identifying the fringe number as $\eta(\xi)$:

$$\eta(\xi) = \left\lfloor \frac{\xi}{\lambda} \right\rfloor \in \mathbb{N} \tag{3}$$

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:

$$\xi = (\eta(\xi) + \varphi(\xi))\lambda. \tag{4}$$

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.

2.3. 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:

$$\varepsilon_\varphi = d_c(\varphi^*(\xi), \bar{\varphi}(\xi)) \tag{5}$$

where $\bar{\varphi}(\xi)$ is the correct phase in ξ, defined in Eq. (2), 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:

$$d_c(\varphi_1, \varphi_2) = sign(\varphi_1 - \varphi_2)\, min\{|\varphi_1 - \varphi_2|, 1 - |\varphi_1 - \varphi_2|\}. \tag{6}$$

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 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 acquiring procedure, and most of the time such elements are not predictable, like 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:

$$\varepsilon_\varphi \sim N(0, \sigma_\varphi). \tag{7}$$

We modelled ε_φ as a Normal Distribution with zero mean and standard deviation equal to ${\sigma_\varphi}^2$. 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, only $\sim 10^{-6}$ 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 4 as:

$$\begin{aligned} \xi &= (\eta(\xi) + (\varphi^*(\xi) + \varepsilon_\varphi))\lambda \\ &= \lambda\eta(\xi) + \lambda\varphi^*(\xi) + \lambda\varepsilon_\varphi. \end{aligned} \tag{8}$$

It is evident that phase error ε_φ has a significant impact on final code value

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

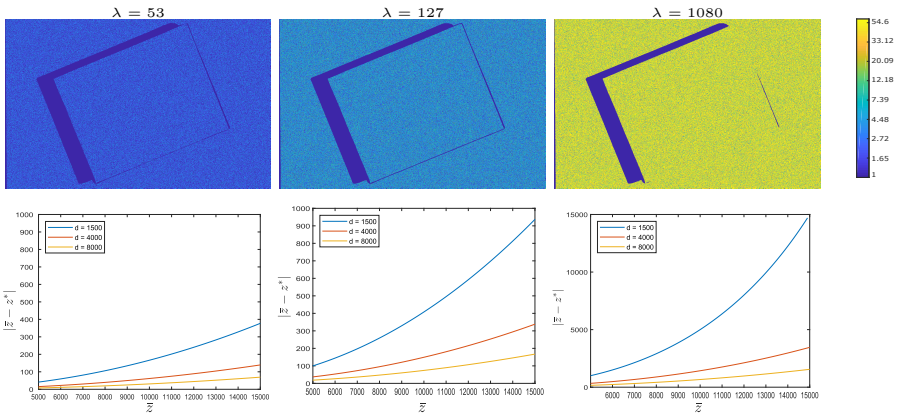

Figure 2: Coding errors using different period lengths (from left to right $\lambda = 53$, 127, 1080 pixels) and constant error standard deviation $\sigma_\varphi = 0.05$ rad. In second row the same parameters are used to measure depth errors for different disparity values (identified with d in three distinct curves). Plots show in x-axes the actual object depth, while in y-axes the depth error after triangulation is plotted.

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:

$$\varepsilon_\xi = \lambda \varepsilon_\varphi. \tag{9}$$

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

$$\varepsilon_\xi \sim N(0, \lambda \sigma_\varphi) \tag{10}$$

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 2 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,

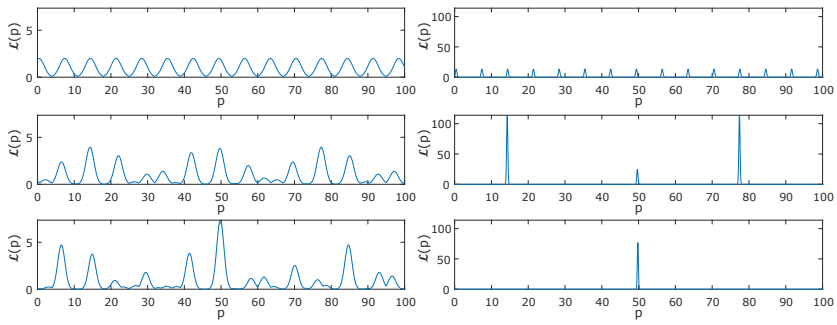

Figure 3: Examples of likelihood functions for different pattern configurations and phase standard deviations. First row shows $n = 1$ and $\lambda_1 = 7$, second row $n = 2$ and $\lambda_1 = 7, \lambda_2 = 9$, third row $n = 3$ and $\lambda_1 = 7, \lambda_2 = 9, \lambda_3 = 11$. We set $\sigma_\varphi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

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

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:

$$\bar{\boldsymbol{\varphi}} = (\bar{\varphi}_1, ..., \bar{\varphi}_n) = (\varphi_1(\xi), ..., \varphi_n(\xi)) \tag{11}$$

Where each $\varphi_i(\xi)$ is defined as in Equation 2. The observed phases computed from intensity images are estimations of true phase values under different noise sources. We denote these values as

$$\boldsymbol{\varphi^*} = (\varphi_1^*, ..., \varphi_n^*). \tag{12}$$

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 $\bar{\varphi}$:

$$\varphi_i^* \sim N_w(\bar{\varphi}_i, \sigma_i) \tag{13}$$

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 [29], and the Probability Density Function relative to observed phases in the i-th pattern is defined as

$$f_i(\varphi_i^*; \xi, \sigma_i) = \frac{1}{\sqrt{2\pi}\sigma_i} e^{-\frac{d_c(\varphi_i^*, \bar{\varphi}_i)^2}{2\sigma_i^2}} \tag{14}$$

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 (13) of unknown mean. Therefore, we can switch the roles of phase value and projector code in Equation (14) considering φ^* as constant value and ξ as the variable. The Likelihood function $\mathcal{L} : [0, \xi_{max}) \to \mathbb{R}^+$ is defined as the product of all individual PDFs and it is defined over the space of projector codes:

$$\mathcal{L}(\xi; \boldsymbol{\varphi^*}, \boldsymbol{\sigma}) = \prod_{i=1}^{n} f_i(\xi; \varphi_i^*, \sigma_i) = \prod_{i=1}^{n} \frac{1}{\sqrt{2\pi}\sigma_i} e^{-\frac{d_c\left(\varphi_i^*, \left(\frac{\xi}{\lambda_i} - \left\lfloor \frac{\xi}{\lambda_i} \right\rfloor\right)\right)^2}{2\sigma_i}} \tag{15}$$

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:

$$\xi^* = argmax_{\xi \in [0, \xi_{max})} \quad \mathcal{L}(\xi; \boldsymbol{\varphi^*}, \boldsymbol{\sigma}). \tag{16}$$

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 in the best possible way.

The choice of $\boldsymbol{\sigma}$ is a fundamental part in this process, since it controls the degree of selectivity of likelihood function. In general, the correct values can not be directly measured, so we consider an approximation $\boldsymbol{\sigma}^*$, in which each σ_i^* to be as close as possible to the real σ_φ which generated the data. 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 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 the phase maps are acquired, we have a vector of observed phases $(\varphi_1^*, ..., \varphi_n^*)$ for each camera pixel. The proposed phase unwrapping approach can be summarized in the following steps: first, an initial integer approximation of the maximum likelihood is computed by choosing the maximum value of $\mathcal{L}(\xi; \boldsymbol{\varphi^*}, \boldsymbol{\sigma^*})$ with $\xi \in \{0, 1, ..., \xi_{max} - 1\} \subset \mathbb{N}$. Then, we restrict the optimization to a real neighbourhood $[\xi - \delta, \xi + \delta]$ of the identified integer code, such that the codes do not wrap. Since the logarithm of likelihood function is a parabola, we can compute the analytical maximum by sampling it in three points, as in

$$\xi^* = \frac{\mathcal{L}(\xi + \delta; \boldsymbol{\varphi^*}, \boldsymbol{\sigma^*}) - \mathcal{L}(\xi - \delta; \boldsymbol{\varphi^*}, \boldsymbol{\sigma^*})}{4\mathcal{L}(\xi; \boldsymbol{\varphi^*}, \boldsymbol{\sigma^*}) - 2\big(\mathcal{L}(\xi + \delta; \boldsymbol{\varphi^*}, \boldsymbol{\sigma^*}) + \mathcal{L}(\xi - \delta; \boldsymbol{\varphi^*}, \boldsymbol{\sigma^*})\big)} \tag{17}$$

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.

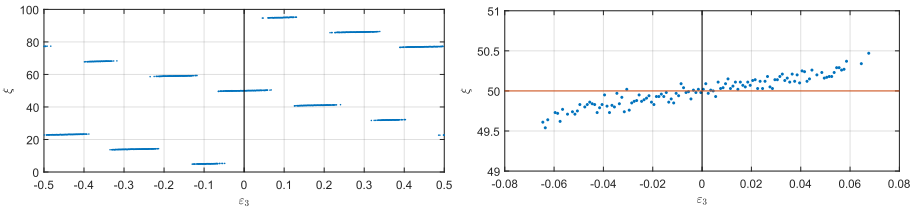

Figure 4: Code estimation in the case $n = 3$ with one erroneous phase value. The first two phases φ_1, φ_2 are slightly perturbed with a random zero-mean Normal noise with $\sigma_\varphi = 0.01$ rad. Error ε_3 on the last phase value ranges from -0.5 to 0.5 (shown in x-axis). The $argmax$ of each phase 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.

4. 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.

4.1. 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, 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 4 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 5 shows an

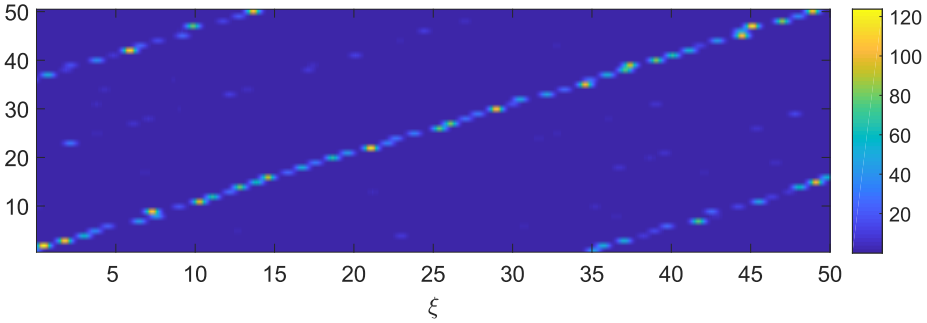

Figure 5: Likelihood values (plotted as intensity values in a colour scale) over the code space (x-axis) for each observed pixel (y-axis). Ideally, values lying on the diagonal should correspond to maximum values in each row.

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 far from the main diagonal. 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.

4.2. 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 15. We then compute the first K local maxima to obtain a set of code candidates with a high likelihood. Such set is denoted as

$$C(u, v) = \{\xi_1^*, \xi_2^*, ..., \xi_K^*\} \tag{18}$$

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

$$V(\xi_i^*, (u', v')) = \sum_{(u,v)} \omega(u' - u, v' - v) S(\xi_i^*, C(u, v)) \tag{19}$$

where $\omega_{(u,v)} : \mathbb{Z}^2 \to \mathbb{R}^+$ 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.

$$S(\xi_i^*, C) = \begin{cases} 0 & \text{if } \nexists\, \xi' \in C \text{ st } |\xi_i^* - \xi'| < \varepsilon \\ \frac{\mathcal{L}(\xi')}{\mathcal{L}_{max}} : \xi' = min(\xi_i^*, C) & \text{otherwise} \end{cases} \tag{20}$$

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

$$min(a, C) = \underset{c \in C}{argmin} \, |a - c| \tag{21}$$

We return likelihood value of the closest code among its neighbours because it is a good indicator on 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:

$$\xi_{max}^* = \underset{\xi_i^* \in C(u', v')}{argmax} \, V(\xi_i^*, (u', v')) \tag{22}$$

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

5. 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.

5.1. 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

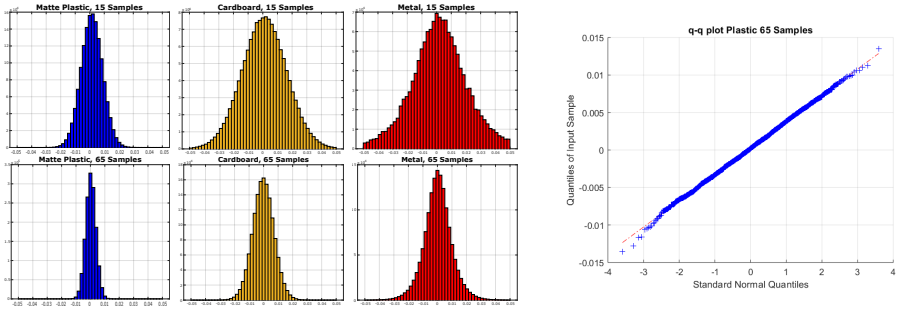

Figure 6: Empirical validation of Gaussian noise assumption: in these acquisitions all expected phases were equal to zero. Left: distributions of measured phase values for different materials (from left to right: matte plastic, cardboard and metal) and samples (1st row 15 samples, 2nd row 65 samples). 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.

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. The experiments were performed using a single camera-projector setup composed by a 5Mpx camera and a 1080p projector, mounted on a rigid structure with 150mm of disparity. Acquired objects were located at about 1m away from the projector.

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 using three surfaces: 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 6 display

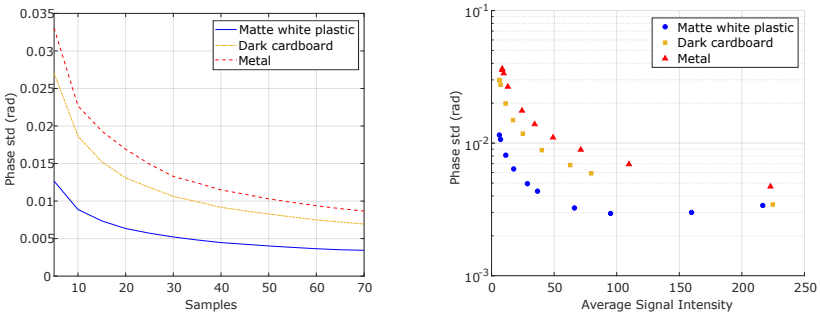

Figure 7: Left: Standard deviation observed on phase values varying the number n of pattern samples. Right: phase standard deviation measured with respect to average signal intensity of acquired images. In both cases, three different materials had been tested.

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 6. 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 7 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 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

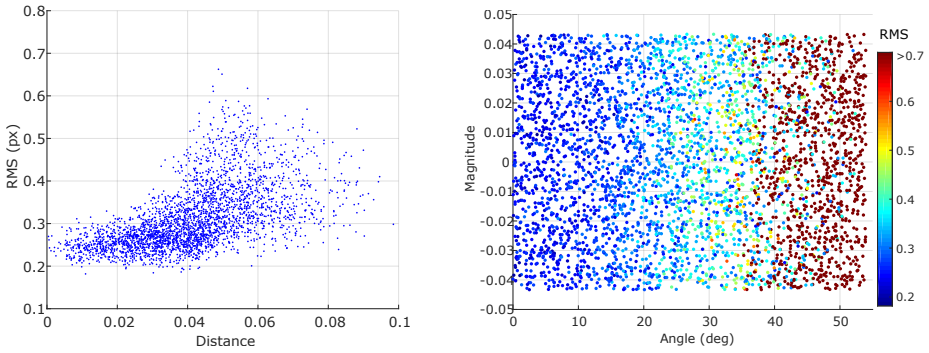

Figure 8: Impact of different sigma estimations on the final RMS (in pixels). Left: effect of distance $|\boldsymbol{\sigma} - \boldsymbol{\sigma}^*|$ on the final coding error. Right: the same experiment, separating vector relative orientation (on x axis) and magnitude of estimated sigma vector (better viewed in colour).

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 7 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 8 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 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 8 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.

5.2. 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 [25] and a classical Graycoding disambiguation. Spatial

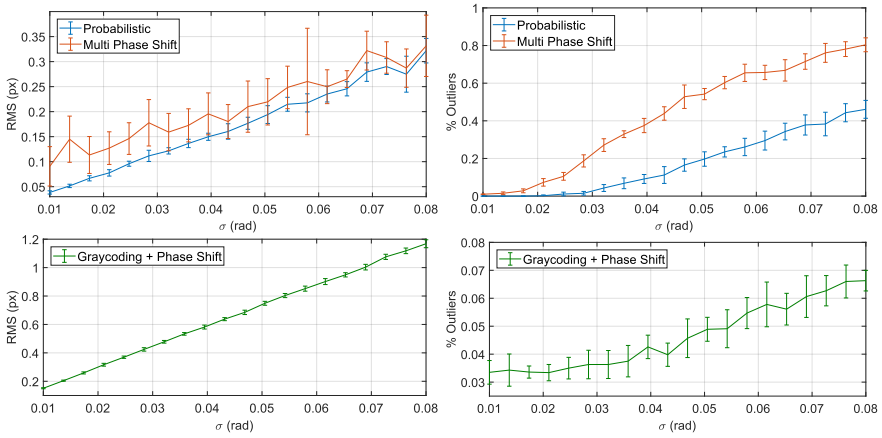

Figure 9: Comparison of Probabilistic approach with respect to number theoretical method [25] (1st row) and Graycoding disambiguation technique combined with just the first phase pattern (2nd row). Left: coding RMS is plotted (on y-axis) against increasing phase error $\boldsymbol{\sigma}$. Right: percentage 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.

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 measurements 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 [28] 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 9 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-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 9 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.

5.3. 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 10, 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 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.

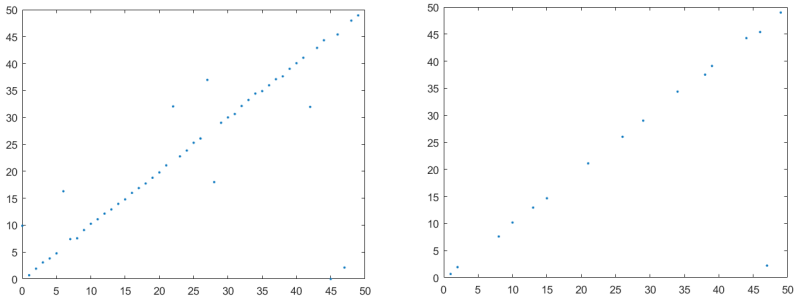

Figure 10: Code recovery comparison. Left: codes computed with our probabilistic method choosing maximum likelihood values. Right: codes computed from the same phase images using a number-theoretical approach for disambiguation.

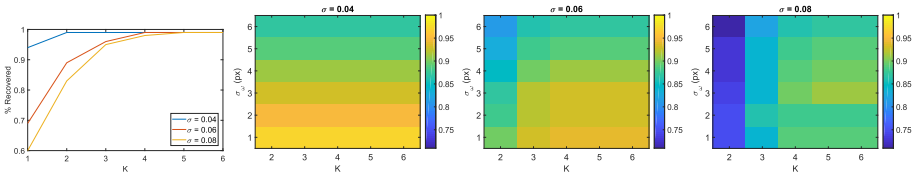

Figure 11: Left: ratio of correctly recovered points wrt the number of peaks extracted from the likelihood function. Other figures show the same ratio combining number of peaks K and different window size.

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 11: 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 4. 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 11 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

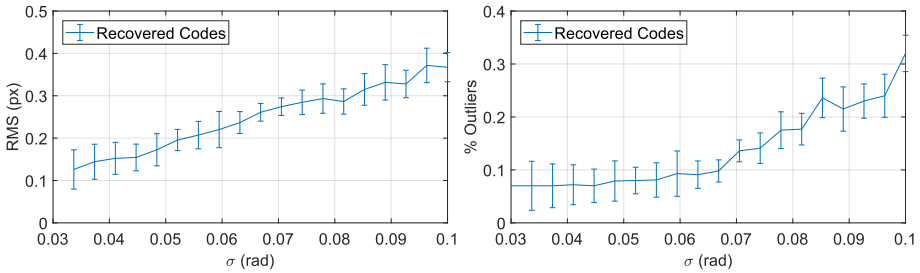

Figure 12: Code RMS (left) and outlier percentage (right) for points recovered through the proposed neighbourhood-based technique.

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.

Finally, we measured error RMS and outlier percentage on recovered codes. This experiment can be compared with Figure 9, 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 12. 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.

5.4. 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

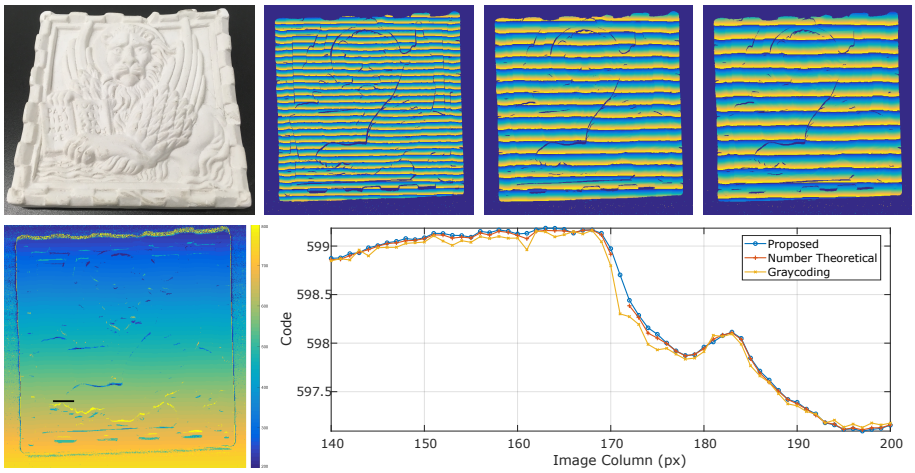

Figure 13: First row: the scanned object and the three acquired phase maps (period length 17, 21 and 33). Second row: computed projector codes using our method, the black segment denotes the selected codes plotted on the right as example. The rightmost plot shows the code values from the three techniques (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.

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. 13, 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 13 on bottom-left corner. The bottom-right plot of Figure 13 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 proba-

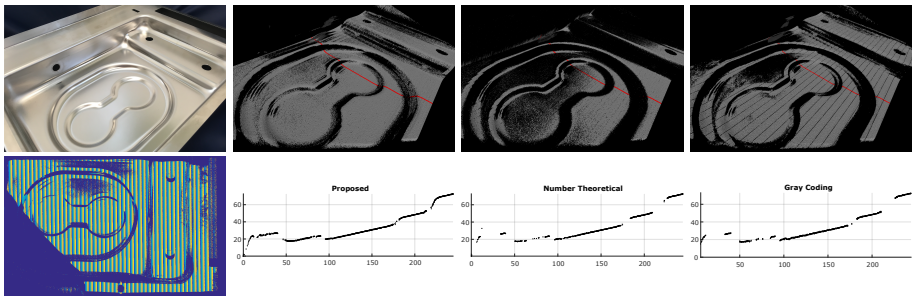

Figure 14: First row: picture of the scanned object and the 3D range-maps obtained applying respectively (from left to right): proposed, number-theoretical and Graycoding. Second row: one of the three phase maps used for Graycoding and the cross sections of 3D data obtained by slicing the scene in correspondence of the red line.

bilistic approaches are similar but the proposed method appears more stable and exhibits no discontinuities (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 15 (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.14, 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 14 (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

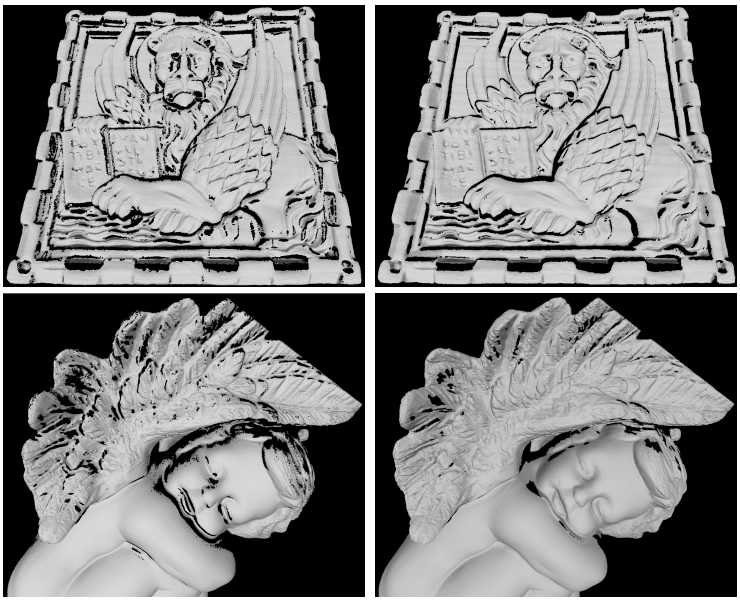

Figure 15: Qualitative comparisons of reconstructed scenes. Left: number theoretical approach, right: proposed probabilistic technique. We display two simple objects acquired using three fringe patterns of different period lengths. Our technique offers a better reconstruction even in simple scenarios, especially in areas where the signal results noisy due to the surface geometrical characteristics.

the acquired surface. Finally, Figure 15 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.

6. Conclusions

In this paper 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. 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). 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. The estimation of a suitable value for $\boldsymbol{\sigma}$ (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 in an empirical way or through a calibration step performed at the beginning of an acquisition session. As future work we plan to investigate the possibility to automatically calibrate the $\boldsymbol{\sigma}$ parameters in a semi-automated way.

References

[1] J. Salvi, J. Pages, J. Batlle, Pattern codification strategies in structured light systems, Pattern recognition 37 (4) (2004) 827–849.

[2] N. G. Durdle, J. Thayyoor, V. Raso, An improved structured light technique for surface reconstruction of the human trunk, in: Electrical and Computer Engineering, 1998. IEEE Canadian Conference on, Vol. 2, IEEE, 1998, pp. 874–877.

[3] H. Hugli, G. Maitre, Generation and use of color pseudo random sequences for coding structured light in active ranging, in: Industrial Inspection, Vol. 1010, International Society for Optics and Photonics, 1989, pp. 75–83.

[4] E. M. Petriu, T. Bieseman, N. Trif, W. S. McMath, S. Yeung, Visual object recognition using pseudo-random grid encoding, in: Intelligent Robots and Systems, 1992., Proceedings of the 1992 lEEE/RSJ International Conference on, Vol. 3, IEEE, 1992, pp. 1617–1624.

[5] T. Miyasaka, K. Kuroda, M. Hirose, K. Araki, 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.

[6] Z. J. Geng, Rainbow three-dimensional camera: new concept of high-speed three-dimensional vision systems, Optical Engineering 35 (2) (1996) 376–384.

[7] H. Kawasaki, R. Furukawa, R. Sagawa, Y. Yagi, Dynamic scene shape reconstruction using a single structured light pattern, in: Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on, IEEE, 2008, pp. 1–8.

[8] S. Inokuchi, Range-imaging system for 3d object recognition, in: Proc. of 7th International Conference on Pattern Recognition, 1984, 1984.

[9] D. Caspi, N. Kiryati, J. Shamir, Range imaging with adaptive color structured light, IEEE Transactions on Pattern analysis and machine intelligence 20 (5) (1998) 470–480.

[10] E. Horn, N. Kiryati, Toward optimal structured light patterns1, Image and Vision Computing 17 (2) (1999) 87–97.

[11] C. Zuo, S. Feng, L. Huang, T. Tao, W. Yin, Q. Chen, Phase shifting algorithms for fringe projection profilometry: A review, Optics and Lasers in Engineering 109 (2018) 23–59.

[12] S. Zhang, Absolute phase retrieval methods for digital fringe projection profilometry: a review, Optics and Lasers in Engineering 107 (2018) 28–37.

[13] M. D. Pritt, D. C. Ghiglia, Two-dimensional phase unwrapping: theory, algorithms, and software, Wiley, 1998.

[14] X. Su, W. Chen, Reliability-guided phase unwrapping algorithm: a review, Optics and Lasers in Engineering 42 (3) (2004) 245–261.

[15] M. Zhao, L. Huang, Q. Zhang, X. Su, A. Asundi, Q. Kemao, Quality-guided phase unwrapping technique: comparison of quality maps and guiding strategies, Applied optics 50 (33) (2011) 6214–6224.

[16] G. Sansoni, M. Carocci, R. Rodella, 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.

[17] Y. An, S. Zhang, Three-dimensional absolute shape measurement by combining binary statistical pattern matching with phase-shifting methods, Applied optics 56 (19) (2017) 5418–5426.

[18] X. Chen, C. Lu, M. Ma, X. Mao, T. Mei, Color-coding and phase-shift method for absolute phase measurement, Optics Communications 298 (2013) 54–58.

[19] C. Zuo, L. Huang, M. Zhang, Q. Chen, A. Asundi, Temporal phase unwrapping algorithms for fringe projection profilometry: A comparative review, Optics and Lasers in Engineering 85 (2016) 84–103.

[20] Y.-Y. Cheng, J. C. Wyant, Two-wavelength phase shifting interferometry, Applied optics 23 (24) (1984) 4539–4543.

[21] Y.-Y. Cheng, J. C. Wyant, Multiple-wavelength phase-shifting interferometry, Applied optics 24 (6) (1985) 804–807.

[22] J. M. Huntley, H. Saldner, Temporal phase-unwrapping algorithm for automated interferogram analysis, Applied Optics 32 (17) (1993) 3047–3052.

[23] W. Osten, W. Nadeborn, P. Andrae, General hierarchical approach in absolute phase measurement, in: Laser Interferometry VIII: Techniques and Analysis, Vol. 2860, International Society for Optics and Photonics, 1996, pp. 2–14.

[24] J. Zhong, M. Wang, Phase unwrapping by lookup table method: application to phase map with singular points, Optical Engineering 38 (12) (1999) 2075–2081.

[25] E. Lilienblum, B. Michaelis, Optical 3d surface reconstruction by a multiperiod phase shift method, JCP 2 (2) (2007) 73–83.

[26] Z. Li, K. Zhong, Y. F. Li, X. Zhou, Y. Shi, Multiview phase shifting: a full-resolution and high-speed 3d measurement framework for arbitrary shape dynamic objects, Optics letters 38 (9) (2013) 1389–1391.

[27] B. Li, T. Bell, S. Zhang, Computer-aided-design-model-assisted absolute three-dimensional shape measurement, Applied optics 56 (24) (2017) 6770–6776.

[28] Y. An, J.-S. Hyun, S. Zhang, Pixel-wise absolute phase unwrapping using geometric constraints of structured light system, Optics express 24 (16) (2016) 18445–18459.

[29] G. Kurz, I. Gilitschenski, U. D. Hanebeck, Efficient evaluation of the probability density function of a wrapped normal distribution (2014) 1–5.