Image and Vision Computing 28 (2010) 1255–1266

Contents lists available at ScienceDirect

Image and Vision Computing

journal homepage: www.elsevier.com/locate/imavis

Efficient multiple phase shift patterns for dense 3D acquisition in structured light scanning

Tomislav Pribanić [a,*], Saša Mrvoš [a], Joaquim Salvi [b]

[a] Faculty of Electrical Engineering and Computing, University of Zagreb, Unska 3, HR-10000, Croatia
[b] Institute of Informatics and Applications, University of Girona, Av. Lluis Santalo s/n, 17071 Girona, Spain

ARTICLE INFO

Article history:
Received 1 May 2008
Received in revised form 17 December 2009
Accepted 12 January 2010

Keywords:
3D acquisition
Structured light
Pattern projection
Multiple phase shifting
Phase unwrapping

ABSTRACT

Although phase shifts (PS) are frequently used to acquire colored surfaces of static objects, especially when acquisition time is not critical, the periodic nature of relative (wrapped) PS maps makes it necessary to deal with the issue of phase unwrapping. Consequently, multiple phase shifts (MPS) have been widely used as an alternative, but this usually involves a large number of different PS maps to unwrap an absolute (unique) phase. In this paper we propose a new MPS method to unwrap a phase and accurately perform the dense 3D acquisition of neutral and colored objects using only two PS maps. Accuracy is reported including a quantitative and qualitative evaluation of the results.

1. Introduction

A powerful way to produce a dense 3D reconstruction is to introduce structured light (SL) techniques in a stereovision system [1,2]. SL assumes a projection of controlled illumination of the scene through one or more projected patterns, commonly using DLP or LCD video projectors. Projected light patterns have a characteristic structure (appearance) because projector image pixels are coded in a certain way. Detecting the same code on the pixels from multiple cameras, i.e., a minimum of one camera and one projecting device, quickly solves the correspondence problem for a large number of points and leads to dense 3D surface acquisition. The simplest classification of the projection strategies is based on the patterns used for static scenes and those used for dynamic scenes. Our interest in the acquisition of static scenes is largely motivated by the growing demand for it from a variety of areas and applications, e.g., the inspection of manufactured parts for quality control purposes, and reverse engineering 3D digitation in the art and cultural heritage preservation and souvenir industries.

During the course of our work we have set out a number of conditions that optimal SL patterns should comply with:

1. Every pixel of the projected pattern should contain the entire code. This will create the conditions for high-resolution 3D reconstruction.
2. There should be high distance between the codewords of neighboring pixels. This will allow high sensitivity to spatial depth resolution.
3. There should be robustness to object color/albedo reflectance properties. Any sort of color (calibration) particular adjustments to or restrictions on the ambient light should be avoided. This will assure almost immediate system use and applicability to various types of object surfaces and ambient light scenarios.
4. There should be robustness to objects with sharp discontinuities and depth changes. This will guard against the problem of possible code perturbation and its misinterpretation during the de-codification stage.
5. The patterns should assure full 3D reconstruction where all three spatial coordinates of the object shape are attainable. In other words, ultimately computing only depth coordinate with respect to one reference plane will not be satisfactory.
6. Simple image processing of the acquired patterns, ideally the only image processing used, should consist of addition, multiplication, comparison and look-up table indexing. More complex image processing, such as finding edges, corners, detecting various shapes, centers of stripes, and color thresholding, should be avoided. Simple image processing will allow relatively easy implementation from a software point of view and fast processing, hopefully comparable with state-of-the-art commercial products.
7. Only off-the-shelf components should be used. No special devices such as special lights sources or colorimeters should be part of the system. This will allow easy and affordable implementation from a hardware point of view.

* Corresponding author. Tel.: +385 1 6129 867; fax: +385 1 6129 652.
E-mail addresses: tomislav.pribanic@fer.hr (T. Pribanić), sasa.mrvos@fer.hr (S. Mrvoš), qsalvi@eia.udg.es (J. Salvi).

0262-8856/$ - see front matter © 2010 Elsevier B.V. All rights reserved.
doi:10.1016/j.imavis.2010.01.003

In fact, there are very few SL methods that can meet all of the above conditions, even for static scenarios where we are allowed to project more than one pattern. One of the very few is based on the so-called phase-shift (PS) method, which can be categorized as one of the time multiplexing strategies [3]. The PS method projects a sequence of periodic intensity patterns, each of which is offset by a fraction of its period from the previous one, so that the entire period is covered. As a result, one obtains a so-called relative phase map, which is also of a periodic nature: values readily available from the relative (wrapped) phase map are said to be wrapped in the range modulo 2π. This raises the problem known as a phase unwrapping procedure, i.e. a computation of the absolute (unwrapped) phase map. The simplest unwrapping solution is called spatial unwrapping, which basically only works well on smooth surfaces, since it assumes that during the unwrapping procedure the phase difference between neighboring pixels is less than π. Evidently, in the case of sharp changes in depths, shadows, occlusion, etc. an extension of this rather basic technique is needed. In [4] an additional color-coded pattern is projected, giving a rough estimate to be used during spatial unwrapping, at the expense of the color pattern being basically restricted to color neutral objects. Specially made hardware can be used to tackle the problem of unwrapping [5]. Another part of the proposed extension consists of temporal phase unwrapping [6], using more than one relative phase map, which is occasionally referred to as the multiple phase-shift (MPS) method. The simplest MPS unwrapping principle comes down to totaling the wrapped phase differences between individual relative phase maps. Unfortunately, in order to achieve very good 3D reconstruction accuracy the total number of relative phase maps is typically fairly large, e.g., 20–30 maps, as shown in [6]. More specifically, to unwrap the finest phase map with N periods one needs a total of N phase maps. Improvements have been devised whereby the number of relative phase maps has been cut down from N to $\log_2 N$ [7,8], during which the periods of the various relative maps follow an exponential sequence. Additionally, there is an approach in which an arbitrary sequence, although still lengthy, can be used [9]. There is also an MPS alternative where one can use even fewer relative phase maps (ideally only two) and where they all have a rather large number of periods (note that the number of periods has a direct effect on condition (2)). This is the so-called number theoretic approach, based on the properties of the relative prime numbers [10,11], where the condition is that individual maps' periods have to be relative primes. The core of the idea can be explained through a consideration of the famous Chinese remainder theorem [12]. In this case, depending on the various system parameters, measurements show that in practice at least three relative maps are often needed to reliably unwrap the phase [10]. Finally, unwrapping a single PS can be realized if combined with Gray-code patterns [13,14]. Gray code uniquely codes various image areas belonging to the individual periods of a single relative phase map. The problem here is that on the borders between different Gray coded image areas it is common to see pixels incorrectly Gray-coded. Such image regions can be excluded from processing, which obviously decreases the ultimate number of reconstructed points. For completeness, we mention a composite pattern based on the PS approach to reconstruct dynamic scenes in real time [15]. Unfortunately, this method requires a careful spatial alignment of the camera and projector. In addition, it blurs depth acquisition due to its use of band-pass and low-pass filters.

Apart from the references mentioned, and to the best of our knowledge, there are relatively few recently published works concerning MPS unwrapping despite its advantages in creating an extremely effective SL strategy in terms of the above list of conditions for optimal SL patterns. In fact, recent developments in other SL categories – spatial neighborhood, direct codification [3] and the somewhat special category of Fourier transform profilometry (FTP) – strongly suggest that MPS is still superior for static scenes. Spatial neighborhood concentrates, in the majority of cases, all the coding schemes in a unique pattern, which is a clear advantage for the acquisition of dynamic scenes. As the name suggests, the codeword that labels a certain point of the pattern is obtained from a neighborhood of the points around it. Commonly used cues for (de)codification can be different single or multiple shapes, e.g., stripes or slits [16,17], lines [18,19], circles [20,21], squares [22], which are either colored or have different gray-level intensities. The main problem here is a high sensitivity to colored surfaces and/or code misinterpretation due to sharp changes in the object's depth. Up to a certain extent, both problems can be lessened through color reflectance calibration [23,24] and dynamic programming [25,26]. Direct codification methods encode points on the pattern using a spectrum of gray-level intensity [27] or a wide spectrum of colors [28] in which case the method is typically restricted to color neutral objects. Besides the problems already mentioned when using color, sometimes complicated hardware is needed to project a color spectrum [29]. There have been some attempts at improvements [30], but the biggest obstacle in this category is high sensitivity to noise because the distance between adjacent codewords is near zero [31]. Finally, FTP, introduced in [32], ideally projects only a single sinusoidal pattern projection. The depth information of the object is encoded into a phase of an imaged pattern. FTP assumes Fourier transform computation, filtration in spatial frequency domain and inverse Fourier transform calculation [33]. The attainable depth range is highly dependent on the system geometric parameters, which is obviously not very convenient. Using modified π-phase shifting FTP, the depth range is extended three times [34]. However, it requires the projection of an additional pattern that, as a rule, makes the method inapplicable for dynamic scenes, unless there is prior knowledge about the object speed [35]. There have been attempts to perform π-phase shifting FTP by constructing a single composite pattern either with or without color codification [36,37], but a major drawback remains: similarly as any other (multi)frequency method, FTP faces a problem of blurred depth acquisition, largely due to not trivial choice of carrier frequencies and related to the filtering step [15].

In this work we propose an MPS approach that provides satisfactory results using not more than two relative phase maps. In brief, for a given pixel, we start from the tuple obtained from relative phase values and a possible set of the full number of periods needed to reach any value on the absolute (unwrapped) phase axis. Adding the relative phase value of one relative phase map to all the possible full numbers of periods gives us candidate solutions for absolute phase values. Doing the same thing for other relative phase maps gives us other candidate solutions. Finding the minimum difference between two sets of candidate solutions solves the ambiguity of the absolute phase value. In Section 2 we give a brief outline of a phase shift projection strategy, i.e., a computation of a relative phase map. In Section 3 we propose our method to solve the ambiguity, already mentioned, caused by the periodic nature of a relative phase map, and to compute an absolute phase map from two or more relative phase maps. In Section 4 we briefly explain the design, components and calibration of our system. Related to this, and as an additional contribution of this work, we then present our method of computing image correspondences between cameras. We then proceed in Section 5 to show our experimental results and use them to quantitatively and qualitatively evaluate our method. In the same section we discuss the experiments and a few other implementation issues. In Section 6 we state our conclusions.

2. Brief overview of the phase-shift method

Consider a projection pattern where the pixels in the same column all have equal intensity, and where that intensity varies along the horizontal axis according to a sine function. The phase-shift method typically assumes a projection of periodic sinusoidal patterns several times, where a periodic sine pattern is shifted between projections. Put in more formal terms, a periodic pattern is actually shifted N times by an amount of φ_i, where shifts are equally distributed to cover the entire period:

$$\varphi_i = \frac{2\cdot\pi}{N}\cdot i, \quad i = 0, 1, \ldots, N-1 \tag{1}$$

For a camera image pixel, the detected gray-level intensity, I_i, obtained as a result of a projected periodic pattern in context and for a shift i can be modeled as:

$$I_i = I_0 + A\cdot\sin(\varphi_R - \varphi_i) \tag{2}$$

where I_0 is the intensity when using no source of projection, A reflects the amplitude of a projected (i.e., detected in image) sine signal and φ_R is the so-called relative phase map value we are looking for. Note that the value of A implicitly includes the effect of the albedo/reflectance variation of a surface patch in the space which is projected onto a particular pixel. After projecting the full sequence of N consecutive shifted images, a common approach to computing φ_R is to minimize the following sum:

$$\varepsilon = \sum_{i=0}^{N-1}[I_i - (I_0 + A\cdot\sin(\varphi_R - \varphi_i))]^2 \tag{3}$$

Expression (2), showing the camera pixel detected gray level pixel intensity I_i as a function of projection pattern parameters, is a substantial idealization and subject to various sources of errors in practice. A convenient way to minimize (3) is simply to use the least square method (see Appendix A for more details). It can be shown that for $N \geqslant 3$ computation of the relative phase, φ_R, is reduced to:

$$\varphi_R = \operatorname{atan}\left(-\sum_{i=0}^{N-1} I_i\cdot\cos(\varphi_i),\ \sum_{i=0}^{N-1} I_i\cdot\sin(\varphi_i)\right) \tag{4}$$

where atan is the four-quadrant inverse tangent function, yielding angle φ_R in the interval $[-\pi, \pi]$.

This typical phase shift approach can be upgraded to MPS as follows. Let us suppose that we have obtained the set of relative phases $\varphi_{R,k}$ from $k \geqslant 2$ relative phase maps, each characterized by its number of periods p_k. We want to identify the pixel position on the abscissa axis based solely on the $\varphi_{R,k}$ set of values. That leaves us with the problem of the unwrapping of the absolute (unique) phase Φ_{ABS}. In other words, due to the periodic nature of a given periodic pattern, a single $\varphi_{R,k}$ value by itself is not, generally speaking, a unique representative that we can use to solve the correspondence problem between the image pixels of two or more cameras, i.e., between a single camera and the source of projection (e.g. a common video projector).

3. A proposal for unwrapping the absolute phase map

A pattern period can be defined either directly by the number of requested periods that pattern must have, or by the length of a single period, bearing in mind the total available pattern width in the context. For simplicity of explanation, we can start by representing our approach by considering two periodic sine patterns, defined by integer length periods λ_1 and λ_2. Fig. 1 shows the appearance of two sine patterns (each pattern column has an intensity according to the sine value of its position on the abscissa axis) and the propagation of the absolute phase axis along the width of the pattern. A pair of relative phases $(\varphi_{R,1}, \varphi_{R,2})$ is indicated for the arbitrary absolute phase value Φ_{ABS} and the following equations hold:

$$\Phi_{ABS} = k_1\cdot\lambda_1 + \varphi_{R,1} = k_2\cdot\lambda_2 + \varphi_{R,2} \tag{5}$$

The core of most MPS approaches, including ours, relies on the fact that for suitable chosen period lengths λ_1 and λ_2, one obtains, propagating along the absolute phase axis, a unique set of pairs $(\varphi_{R,1}, \varphi_{R,2})$, all the way up to a certain value of Φ_{ABS}. "Suitable" means that for typical MPS strategies, pattern periods (lengths) have to unconditionally be relative primes to produce any kind of usable output. When this is the case, a product of pattern period lengths $\lambda_1\cdot\lambda_2$ determines the range on the absolute phase axis within which we acquire the unique pairs for relative phase values. However, our method can successfully cope even with sets of period lengths which are not necessarily relative primes. More specifically, our method correctly unwraps the phase up to the value in the absolute phase which is equal to $\operatorname{lcm}(\lambda_1, \lambda_2)$. Here lcm() represents a function whose output is the *l*east *c*ommon *m*ultiple for input parameters, which is a minimal number divisible by the period lengths used. If for some predefined pattern width W there is a case where inputs λ_1 and λ_2 cannot be chosen as relative primes, then our method adjusts the corresponding pattern periods p_i $(i = 1, 2)$. More specifically, instead of just saying $p_i = W/\lambda_i$, our method uses the expression $p_i = \operatorname{lcm}(\lambda_1, \lambda_2)/(\lambda_i)$, where $\operatorname{lcm}(\lambda_1, \lambda_2)$ is rescaled to cover full width W. It should be plain to see that in cases where period lengths are prime numbers, then the lcm() function simply boils down to multiplication of the lengths. In other cases, we simply rely on an efficient Euclid algorithm to compute the greatest common divisor $\gcd(\lambda_1, \lambda_2)$ and ultimately compute $\operatorname{lcm}(\lambda_1, \lambda_2) = (\lambda_1\cdot\lambda_2)/\gcd(\lambda_1, \lambda_2)$ [38].

If "suitable" period lengths are chosen, then one can gather unique pairs of $(\varphi_{R,1}, \varphi_{R,2})$ while propagating along the absolute phase axis. This originates from number theory and the divisibility properties of integers [12]. As rule, any formal mathematical proof is left out of published work presenting MPS strategies; occasionally there are references to a previous work and/or materials specifically discussing a topic of number theory. Therefore, as an additional contribution of this work and before we proceed with further explanation of our method, we will try to first give here a less formal proof of the above.

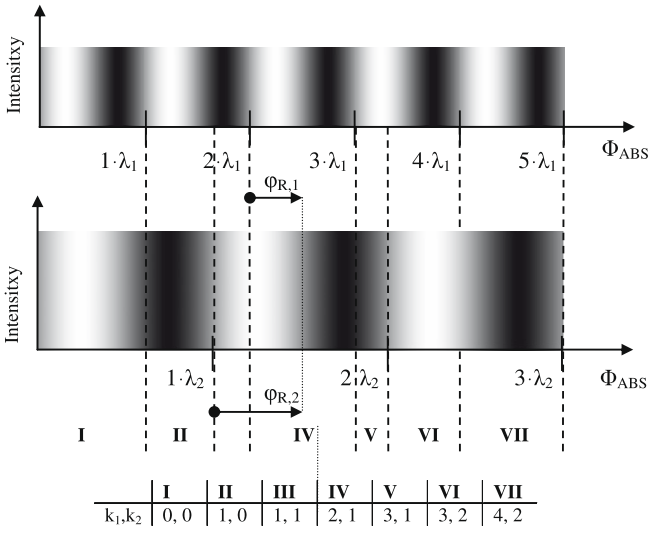

Fig. 1. A propagation of the absolute phase value Φ_{ABS} along two different sine intensity patterns. For the absolute phase position a pair of relative phases $(\varphi_{R,1}, \varphi_{R,2})$ is indicated as an example.

Consider two values on the absolute phase axis $\Phi_{ABS} \neq \Phi'_{ABS}$. For point Φ_{ABS} we recall the derivation of Eq. (5) and for Φ'_{ABS} we write accordingly:

$$\begin{aligned} \Phi_{ABS} &= k_1 \cdot \lambda_1 + \varphi_{R,1} = k_2 \cdot \lambda_2 + \varphi_{R,2} \\ \Phi'_{ABS} &= k'_1 \cdot \lambda_1 + \varphi'_{R,1} = k'_2 \cdot \lambda_2 + \varphi'_{R,2} \end{aligned} \tag{6}$$

Next, assume the situation where $\varphi_{R,1} = \varphi'_{R,1}$ and $\varphi_{R,2} = \varphi'_{R,2}$ or, equivalently, the situation where the necessary condition to obtain a unique pair of relative phase values is violated. Since $\Phi_{ABS} \neq \Phi'_{ABS}$, we now search for a minimal distance between two absolute phase values when the assumed situation (violation) arises. Subtracting the absolute phase values in (6) yields:

$$\begin{aligned} (k'_1 - k_1) \cdot \lambda_1 &= (k'_2 - k_2) \cdot \lambda_2 \\ n_1 \cdot \lambda_1 &= n_2 \cdot \lambda_2 \end{aligned} \tag{7}$$

where we emphasize that n_1 and n_2 are integer numbers. Additionally, it is useful to note that the product of $n_1 \cdot \lambda_1$, i.e., $n_2 \cdot \lambda_2$, should be as small as possible, since we are interested in the minimal distance $|\Phi'_{ABS} - \Phi_{ABS}|$. It almost leads to the natural conclusion that this product should be a least common multiple of period lengths in the contexts λ_1 and λ_2. Hence, there will again be the very same $(\varphi_{R,1}, \varphi_{R,2})$ pair on the absolute phase axis after distance determined by $\mathrm{lcm}(\lambda_1, \lambda_2)$. We now proceed with a further explanation of our method.

Once the extent $\mathrm{lcm}(\lambda_1, \lambda_2)$ on the absolute axis value is determined, it is a simple matter to also compute the range of possible k_1 and k_2 integer values (5) (Fig. 1), as follows:

$$k_1 \in \left[0 \quad \frac{\mathrm{lcm}(\lambda_1,\lambda_2)}{\lambda_1}\right], \quad k_2 \in \left[0 \quad \frac{\mathrm{lcm}(\lambda_1,\lambda_2)}{\lambda_2}\right] \tag{8}$$

Unfortunately, given a $(\varphi_{R,1}, \varphi_{R,2})$ pair, the values k_1 and k_2 are not, in principle, readily known. In addition, with realistic relative phase data, a $\Phi_{ABS,2}$ value computed from $\varphi_{R,2}$, k_2 and λ_2 data will differ from a $\Phi_{ABS,1}$ value obtained on the basis of $\varphi_{R,1}$, k_1 and λ_1. Nevertheless, it is reasonable to assume that a difference due to noise in computed φ_R will be smaller than a difference obtained by a completely wrong combination of integers k_1 and k_2 involved in the computation of $\Phi_{ABS,1}$ and $\Phi_{ABS,2}$. Thus, the proposed method, given $\varphi_{R,1}$ and $\varphi_{R,2}$, first computes $\Phi_{ABS,1}$ and $\Phi_{ABS,2}$ for all possible combinations of k_1 and k_2 (recall that λ_1 and λ_2 are defined through the definition of the requested number of periods). The ambiguity which is the correct value for Φ_{ABS} is solved by finding a (k_1, k_2) pair which gives the smallest difference $|\Phi_{ABS,1} - \Phi_{ABS,2}|$. Note that our method acknowledges the fact that the possible combinations of (k_1, k_2) pair sets do not include all theoretical pair combinations from the ranges of k_1 and k_2, as determined by (8). For instance, Fig. 1 shows changes of two relative phase maps along the absolute phase axis where the first relative phase map has $p_1 = 5$ periods and the second has $p_2 = 3$ periods. Careful consideration reveals that certain intervals on the absolute phase axis are characterized by unique pairs of (k_1, k_2). Therefore in order to speed up data processing, all these unique pairs of (k_1, k_2) can be pre-computed, once the number of periods (i.e. period lengths) for specific periodic intensity patterns are chosen.

Besides the periodic intensity images needed for phase shifting itself, we also acquire two reference images: one 'black' and one 'white'. The former assumes a shut-off projector and imaging of the scene using only ambient light. The later assumes imaging of the scene with the projection of a full white pattern. Computing the image difference of those two images allows us to detect shadow/occluded areas and normalize ambient light. All pixels below a certain, manually set threshold are excluded from further processing.

Having explained the theoretical aspects of our proposed method, it would be useful to reflect on the fulfillment of optimal SL patterns, as stated in the introduction. It is evident that our method codes every projected pixel, which promises a dense 3D reconstruction, and we may recall that this is typically not the case for the spatial neighborhood family of methods. Furthermore, periodic patterns allow a high distance between codewords of adjacent pixels on the abscissa axis. Consequently, high sensitivity to spatial depth resolution is guaranteed, unlike for the direct codification category methods. Next, our patterns use gray-level intensities that, using PS methods (as shown in Appendix A), cancel out the negative influence of varying albedo/reflectance properties. Hence, our method is also a suitable design for colored surfaces, something that would be very hard to expect from any method using color patterns. Our method does not suffer from code perturbations due to sharp depth change, which is common with the spatial neighborhood or even Gray code alone. In addition, it imposes no restriction on the magnitude of depth change that is present in FTP methods. Moreover, our method is invariant to the problems caused by the improper band pass filtering, and it gives code that, during a subsequent triangulation, will provide all three spatial coordinates, and not only the depth coordinate with respect to the reference plane, as is typical of some FTP methods. Finally, apart from the commercially available projector and camera(s), no particular hardware pieces are needed for pattern generation/projection or system color calibration. For adjustments to the environment our method uses just two reference images, as explained in the previous paragraph.

In fact, it appears that the only two methods that could theoretically fulfill the same conditions as ours are two time-multiplexing methods: MPS-based on the number theoretic approach and Gray code + PS. However, as will be shown in the experimental results, a straightforward implementation of those two methods is inferior in terms of correctly computed pixel codes, i.e., available 3D resolution.

4. System set up description

We implemented our SL method using two different system versions: System A is composed of an uncalibrated video projector and a pair of calibrated cameras. System B consists of a calibrated video projector and a single camera.

In more detail, our 3D SL System A consisted of a pair of Basler A601fc FireWire cameras, calibrated and rigidly attached to a bar. An InFocus DLP video projector (Infocus IN26+EP) was positioned between the cameras and served only as a means of pattern projection. The camera-grabbed images had a resolution of 640×480 pixels. Our own routines, written in Matlab and using the Image Acquisition toolbox, allowed communication with the cameras and all other data processing, including the generation of projected patterns and their projection with a DLP video projector. The cameras were calibrated using a 2D calibration pattern with 11×8 black circles on a white background. The diameter of the circles was 15 mm and they were 10 mm apart. The calibration algorithm used closely resembled the one explained in [39], although assuming a specific hardware system design there are even more user-friendly methods [40]. The calibration volume was approximately 400 mm $\times$ 400 mm $\times$ 500 mm. The distance from the cameras to the center of the calibration volume was nearly 1000 mm and the camera baseline was almost 500 mm. The angle between the cameras' optical axes was approximately 30°. The mean error and standard deviation between the calibration point positions detected on the images and the positions provided by the calibration model using computed calibration parameters were for one camera 0.097 pixels and 0.078 pixels, respectively. For the second camera, the mean error obtained and the standard deviation were 0.093 pixels 0.075 pixels, respectively.

The realization of System B involved a spatial readjustment of one of the cameras with respect to the projector in such a manner

that the calibration volume size, base distance and angle between the optical axes mentioned above for two cameras were now basically the same for only one camera and a projector. Of course, a completely new set of calibration images was acquired to calibrate the camera, but the same 2D calibration pattern and algorithm were used as already explained [39]. To calibrate the projector, the SL method proposed in this work was used to come up with the projector image coordinates which corresponds to the calibration points of 2D calibration pattern. More specifically, for every spatial position of the 2D calibration pattern (during camera calibration) we projected the proposed patterns in both a vertical and a horizontal direction with respect to the projected image axes. Finding from our patterns the unique codes (in both projected image axes) of centroids on the 2D pattern, and knowing their 3D spatial counterparts, enabled us to use essentially the same calibration algorithm as the one for a camera [39]. For the camera the mean error and standard deviation between the detected calibration point positions in the images and the positions provided by the calibration model using computed calibration parameters were 0.090 pixels and 0.066 pixels respectively. Similarly for the projector, the mean error obtained and the standard deviation were 0.149 pixels 0.114 pixels, respectively.

It seems that most of the 3D structured light systems have hardware designed along the lines of System B. In fact, even a quasi projector calibration is possible if we, for example, laser scan the object at a constant scanning speed [41]. However, the truth is that both System A and System B designs offer certain pros and cons. In the case of System B, another advantage besides the obvious simpler hardware configuration is that after the de-codification stage is finished, it is a relatively simple matter to undertake 3D triangulation between the camera and the projector. Unfortunately, on the projector side typically only one axis image coordinate is known, which makes image coordinate undistortion (normally undertaken during 3D triangulation) more liable to error. Another drawback of System B is the need to calibrate a projector. Projector characteristics (primarily optics) are typically behind those of cameras, which puts an additional burden on successful calibration. Moreover, the projector is typically calibrated further than the camera, leading to error propagation that introduces inaccuracies in the projector parameters. Hence, ultimate 3D reconstruction accuracy is likely to be impaired. Besides, some high-end professional video projectors project stripes which are slightly bent and therefore make projector calibration even more demanding. Considering all this, a 3D acquisition system designed so that it does not explicitly depend on a (certain type of) calibrated projector, but rather is capable of working elegantly with and switching to almost any source of light as needed, requiring only projector features to be satisfactory for pattern projection and application in the context (e.g., depth ratio, image resolution, refresh rate and luminance), appears to be quite attractive. These were some of the major arguments that led us to test our proposed SL strategy using System A as well as the more common System B design.

On the other hand, a price to pay for having the 3D System A design is that correspondence between pixels that have the same code has to be carried out between cameras, prior to 3D triangulation. It is possible to design projection patterns so that correspondence between pixels in multiple cameras is readily available, but in most cases additional computation effort is needed. For the sake of the completeness of this work, we briefly explain below our proposal for how to accurately find correspondences between two cameras.

We pick one camera as the reference one. For every pixel in the reference camera and its code $\Phi_{ABS,1}$ we first find its undistorted image position. Based on this undistorted value, we compute the epipolar line in the image of a second camera. Next we consider only the pixels on the epipolar line, i.e. its codes. From that set we choose a pixel that has the code $\Phi_{ABS,2}$ closest to the code $\Phi_{ABS,1}$ of our pixel from the reference camera image. That is our initial correspondent solution and we can now compute the initial 3D position. Next we proceed with the refinement phase. Based on the 3D initial solution, we can compute depth w_1, which is the Euclidean distance between the reference camera's optical center and the 3D initial solution. Note that the difference initially found between absolute phase values ε:

$$\varepsilon = \Phi_{ABS,1} - \Phi_{ABS,2} \tag{9}$$

is normally different from zero and either positive or negative. We need to find a pixel position on the second camera's image with sub-pixel precision and a code $\Phi_{ABS,2}$ as close as possible to $\Phi_{ABS,1}$, i.e. $\varepsilon \to 0$. Let us suppose that our initial w_1 gave us $\varepsilon > 0$. Next we look for a new depth w_2 along the initial back projected line from the pixel on the reference camera image, which would give us a new 3D position that, in turn, when projected on the second camera's image, would give $\varepsilon < 0$. Evidently the difference ε is a function of depth w. Therefore, we strive to bracket our final solution for depth w between values w_1 and w_2. Once we find w_2 it is a simple matter, using Brent's method [42], for instance, to find an iteratively refined solution for a depth w for which holds:

$$\Phi_{ABS,1} = \Phi_{ABS,2} \iff \varepsilon = 0 \tag{10}$$

Finding the depth w is equivalent to determining the refined 3D position. Additionally, note that for each new candidate of depth in the Brent iterative algorithm we project a new 3D candidate point on the image of the second camera. After projection of the 3D point on the second image we apply a distortion model to find the final image position. Since the projected and distorted image coordinates will almost certainly be non-integer values, we compute a corresponding code $\Phi_{ABS,2}$ using bilinear interpolation of four neighborhood pixels.

5. Experimental evaluation

The first series of experiment concerns the System A design.

One of the key issues with the SL principle is, without doubt, the necessary number of patterns to be projected. To compute a single PS map we need at least three images according to (4). In Fig. 2 we show typical examples of the absolute phase map's appearance computed from two relative phase maps, in which at each instance a different number of shifted images was used to compute the individual PS maps. It is clear that the absolute map as shown in Fig. 2a is totally useless and, at least in the case of our proposed method, the theoretical minimum of projecting only three patterns does not apply. The situation changes considerably as we move from three to five and eight, in Fig. 2b and c, respectively. In fact, our experiment shows that we obtain a smooth absolute phase map every time we project and shift six patterns for each individual relative PS map. That holds for various pattern periods and period combinations. However, to increase the redundancy and robustness of our method, and in accordance with the assumption that for static scenes a few more images will not make much difference, we normally recommend using eight shifted images to compute a single relative PS map.

The next important issue is to discover how many individual PS maps are needed to find a reliable absolute phase map and eventually perform accurate 3D acquisition. In Table 1 (left and middle part) we show the reconstruction results when using two and three PS maps, respectively. In both cases we projected patterns on the flat surface, positioned at various places in the volume, where subsequently a 3D cloud of points was acquired. Then we fitted a plane through the reconstructed points (ideally all lying in a space on the plane) and computed the mean error and the standard deviation of

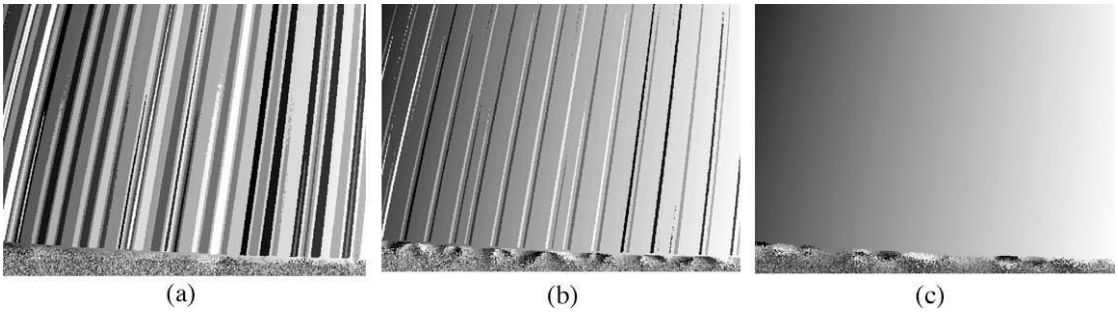

Fig. 2. The impact of a number of shifts/projected patterns on the computation of relative phase maps, and ultimately on the absolute phase map. (a–c) Show the appearance of an absolute phase map computed from two relative phase maps. In (a) the relative maps were found based on three shifts (projections of periodic patterns) and in s (b) and (c) there were five and eight shifts, respectively.

Table 1
Mean error and standard deviation of distances for the total of N reconstructed points with respect to the fitted plane through those N points. p_1, p_2 and p_3 show the chosen number of periods for periodic patterns used to construct two or three relative PS maps in the context. Note that for the pale plane, the width of the reconstructed flat surface ranged approximately between 300 mm and 370 mm and the height between 210 mm and 270 mm. Besides, for the colorful plane the dimensions of the reconstructed flat surface were the ones of an A4 paper, i.e. 297 mm $\times$ 210 mm.

System A, pale plane, two maps					System A, pale plane, three maps						System A, colorful plane, two maps $p_1 = 15$, $p_2 = 19$		
p_1	p_2	N	Mean (mm)	Std. (mm)	p_1	p_2	p_3	N	Mean (mm)	Std. (mm)	N	Mean (mm)	Std. (mm)
1	3	70887	0.269	0.214	4	7	9	69009	0.143	0.120	62264	0.142	0.126
3	5	72355	0.145	0.118	8	11	13	69319	0.131	0.110	42820	0.149	0.129
5	7	72395	0.132	0.109	10	13	15	69241	0.130	0.108	38896	0.143	0.126
7	11	72185	0.130	0.107	12	15	17	69110	0.129	0.109	43744	0.138	0.119
11	15	72066	0.131	0.109	16	19	21	69231	0.130	0.114	55076	0.146	0.129
15	19	71531	0.131	0.109	20	23	25	69229	0.128	0.111	41584	0.145	0.127
17	21	71374	0.132	0.109	24	27	29	69314	0.131	0.116	49314	0.143	0.124
17	23	71385	0.131	0.109	28	31	33	69292	0.130	0.113	59292	0.148	0.130
19	25	71194	0.132	0.109	32	35	37	69333	0.136	0.118	38324	0.139	0.120
20	27	71081	0.132	0.109	36	39	41	69494	0.133	0.123	45494	0.146	0.127
25	27	70396	0.132	0.110	44	47	49	68667	0.137	0.144	58667	0.149	0.131

the distances between the reconstructed points and the fitted plane. The figures shown in Table 1 reveal basically the same level of accuracy, given our experimental conditions. However, the number of projected patterns did increase by roughly 50% in the case of three PS maps, which is strong evidence in support of using our method and relying only on two relative PS maps. This is certainly favorable to the traditional MPS-based methods, categorized as temporal phase unwrapping, and the number theoretic approach, which usually requires more than two PS maps.

Finally, a natural question arises: how does the increase in pattern periods contribute to ultimate accuracy, which itself is related to the question of whether there is an optimal combination of pattern periods. The fact is that the dynamic range of projected/detected (sinusoidal) gray levels is finite. Therefore, to decrease sensitivity to noise between adjacent code words, i.e., to increase the distance between code words of adjacent pixels and consequently increase spatial resolution as well, it is certainly beneficial to project multiple period patterns. Increasing the number of periods helps, in principle, at the very beginning. However, at a certain point shifting the pattern with an extra large number of periods does not provide significantly different intensities of light between adjacent code words either. The bottom line is that the shifts are too small with respect to the actual period lengths (looked at from the perspective of a projector coordinate system). On the other hand, too few shifts may make it difficult to compute relative phase values and eventually compute the absolute phase map (as shown previously in Fig. 2). Our method can certainly deal with patterns with periods over 30, 40 or even more. But in such cases we would need to use at least three PS maps to come up with a correct absolute phase map, since the noise would be too great to rely on only two PS maps. Those experimental results with pattern periods over 30 or 40 are not shown in this paper since they did not contribute to an increase in accuracy. In fact, it is useful to note that the accuracy results shown (Table 1) are pretty constant for any combination of periods where individual patterns are greater than five. This is largely due to the fact that ultimate accuracy also depends on numerous other factors, e.g., the quality of cameras (optics) and the calibration method used, the ambient light, the camera projection geometry, and the size of the calibration volume. Given the other parameters of our system, we have achieved an available level of accuracy that is fairly immune to the specific combination of pattern periods used to construct the PS maps.

In the previous sections we have emphasized the fairly well-known advantage of the PS method, namely that it is robust to colored surfaces and even to surfaces with abrupt changes in albedo. To test our method against it we generated a random pattern of 32×24 squares of various colors (Fig. 3). We show here the representative 3D acquisition accuracy results when using two relative phase maps constructed from periodic patterns with 15 and 19 periods, respectively. The color pattern was printed out on an A4 sheet of paper and stucked onto the same flat pale object we had previously used in our experiments. We also reconstructed a cloud of points, fitted a plane through it and, finally, computed a mean error and standard deviation of distances between the reconstructed points and the fitted plane. The figures shown in reveal a slight decrease in accuracy as compared to results when we reconstructed a pale flat surface (Table 1). This had been expected to a certain extent since the theoretical assumption of full robustness to a colored surface is only a substantial idealization, which is particularly hard to fulfill in areas where abrupt changes in albedo take place. Besides, a laser printed paper (sticker), although glued with the greatest care onto a planar surface and without any

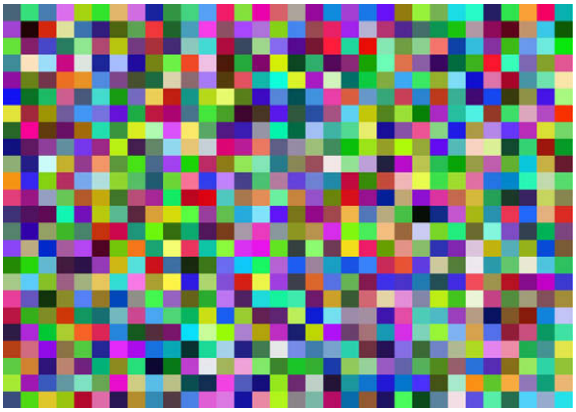

Fig. 3. Method evaluation in the case of colored surfaces: an appearance of a randomly generated colored pattern, printed out on a flat surface that is subsequently reconstructed.

apparent defects, was probably less perfectly flat than the underlying surface.

As part of the qualitative evaluation of our method we chose three rather demanding objects: a manikin head, a human foot (sole) and a yogurt bottle. In these cases we used only two relative phase maps to unwrap the relative phase, constructed from periodic patterns with 17 and 21 periods, respectively. Figs. 4 and 5a and b show details of the manikin's head scan, which exhibits sharp transitions from dark to bright intensities along its surface. Nevertheless, a close qualitative examination of the reconstructed 3D surface with the corresponding image (Fig. 5a and b) reveals that fine details visible on the manikin image are preserved on the accompanying reconstructed 3D surface. Finally, a profile view of the generated 3D mesh is depicted to enhance the shape of the object (Fig. 7a top left). Note that the bounding box encompassed by the reconstructed volume was 154 mm (width) × 231 mm (height) × 87 mm (depth).

Our second test object was a human foot (Figs. 6a and 7b). The corresponding bounding box encompassed by the reconstructed volume was 119 mm (width) × 227 mm (height) × 43 mm (depth). This experiment was demanding primarily with respect to a relatively dubious assumption about how stationary the human body can be during the process of imaging. Note that the pair of cameras and the video projector were unsynchronized. In addition, our MATLAB image acquisition software required several seconds to acquire the 18 images (8 images per PS map, plus two reference images) needed for this experiment. In a more ideal situation, we would have been able to acquire the images using specialized hardware where the camera(s) and video projector were synchronized, thereby reducing acquisition time to less than a second for the 18 images. In spite of this, even with the hardware set up described and an image acquisition duration of several seconds, the 3D surface of the foot that was obtained resembled very closely the fine details visible on the image next to it (Fig. 6a). The less detailed reconstructed area around the toes is simply due to a greater surface complexity, shadowing and occlusions of those parts with respect to either the camera(s) or the projector or both. Hence, just like with any other more complex surface, one needs to scan and reconstruct the same object from multiple views and ultimately perform surface registration; however, that issue is beyond the scope of this work [43].

Our third test object was explicitly chosen as one that had a colored surface (Fig. 6b). Once again, for a qualitative evaluation we relied on the image provided and a reconstructed 3D cloud of points. It is clear that many surface details from the image, primarily in terms of surface color (texture), are successfully preserved on the surface of the reconstructed 3D cloud of points that is shown. In this case the generated mesh of 3D reconstructed data without the original object texture is depicted in Fig. 7c and the bounding box encompassed by the reconstructed volume was of 44 mm (width) × 153 mm (height) × 42 mm (depth). All the profile views considered so far in Fig. 7 show a very good visual match of the reconstructed meshes to the objects' silhouettes.

Next we show performance results using the System B design, performing similar experiments to those with the System A design. In addition, because the System B design features more frequently in other published works, we also show System B's comparative results with respect to two other commonly-used SL methods: the number theoretic approach and Gray code combined with PS. However, it should be clear that similar conclusions could be drawn using System A. As explained in previous sections in the paper, the two common SL methods used to compare our method were specifically chosen since from a theoretical point of view they appear to fulfill set conditions about the optimal SL projection strategy we defined at the very beginning. Table 2 shows the 3D reconstruction accuracy results of the pale and colored flat planes (Fig. 3) when implementing System B, a number theoretic approach and Gray code + PS, respectively. From the figures in the tables there does not appear to be a great difference in accuracy. However, there is a substantial difference in the number of correctly reconstructed points, i.e., the acquired 3D resolution. Specifically, as we increase the number of the pattern periods used and/

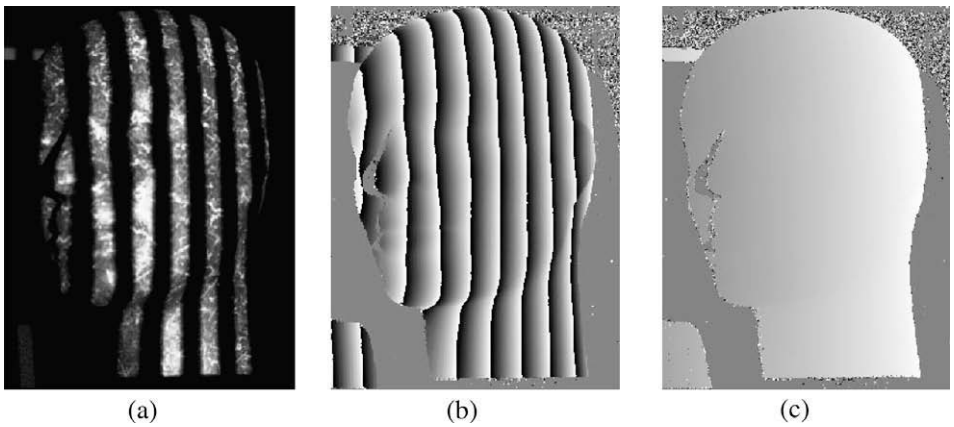

Fig. 4. Manikin head: (a) one of the eight shifted and projected periodic patterns; (b) one of the relative phase maps computed from eight shifted and projected periodic patterns; (c) an absolute phase map computed from two relative phase maps.

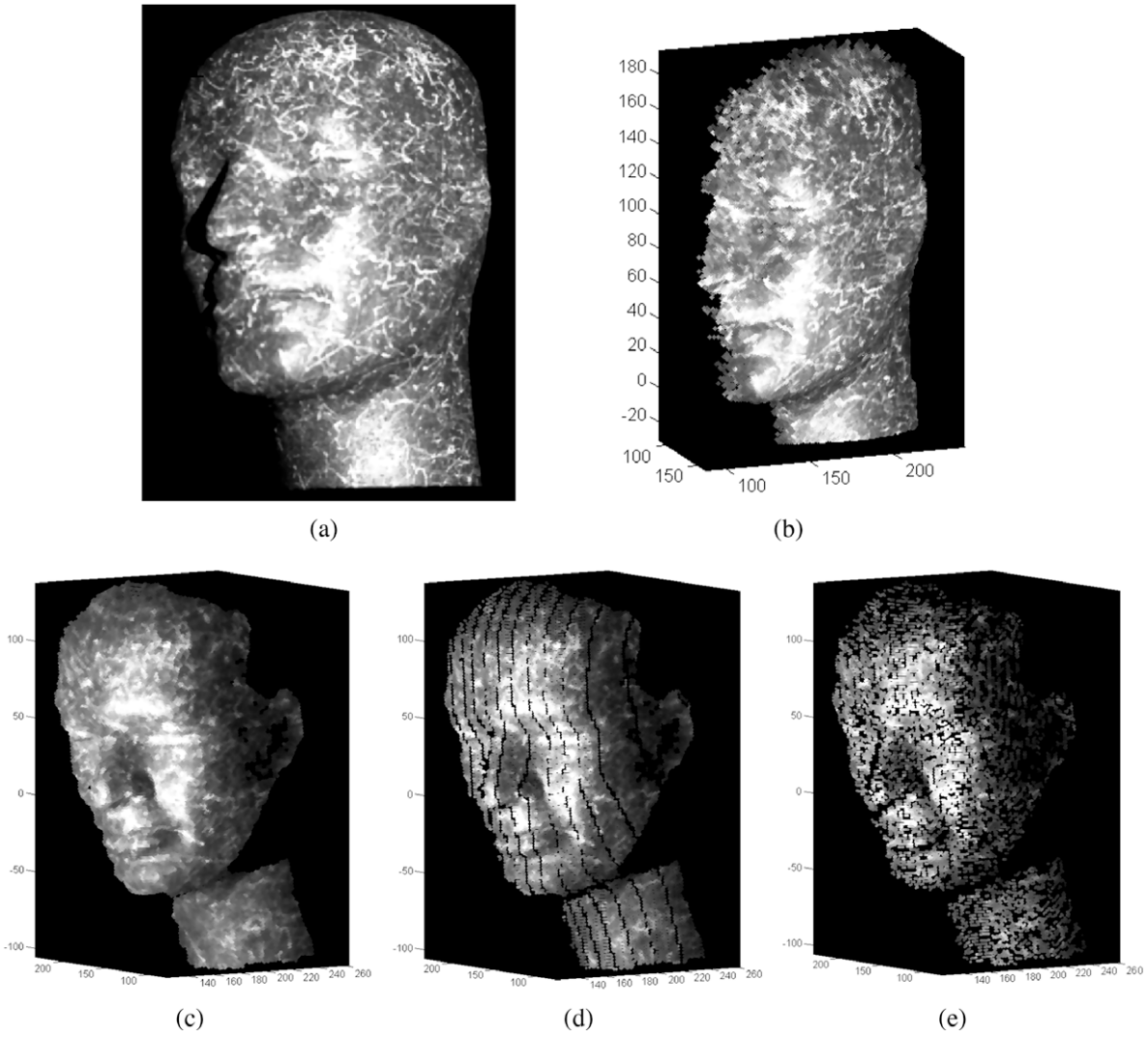

Fig. 5. Manikin head scan. (a) Image and reconstructed surfaces: (b) System A, (c) System B, (d) Gray code + PS, (e) number theoretic approach.

or the length of the Gray code word, the number of wrongly decoded image points increases and the attainable 3D resolution drops. Fig. 8 shows this negative effect for the number theoretic approach and Gray code + PS, respectively. Ideally, we would expect an unwrapped absolute map with a smooth appearance, as shown in Fig. 2c. The number theoretic approach uses, essentially, a linear equation where inputs are PS values. Consequently, it depends heavily on the computational accuracy of the PS values. A further increase in the number of PS maps used or more careful system parameter optimization could ease the problem [10,44]. Both solutions impose additional constraints not present in our proposed method. In the case of Gray code + PS the worse situation is in the finest Gray-code pattern, where the number of Gray-code labeled regions is equal to the number of PS map periods involved. Evidently, incorrectly Gray-code labeled image points combined even with an error-free PS map will eventually give an erroneous absolute phase code. A straightforward solution would be to detect (discard) incorrectly labeled image points (detection would not necessarily be trivial if there were too many of those) where a compromise would have to be made between the size of the Gray code word (i.e., direct impact on the depth resolution) and the total number of incorrectly labeled image points on the finest Gray-code pattern. However, it is important to note that even if the reduced resolution were acceptable, it takes a substantial image processing to automatically detect those invalid pixels having incorrect absolute values. Particularly, in the case number of invalid pixels becomes excessive, it is very hard to detect all of invalid pixels and at the same time not to discard some correct ones, which brings an additional resolution decrease. Besides, it is possible that some of the invalid pixels will remain undetected. To detect invalid pixels we have simply used a scanning mask of size $n = 3$ pixels, where we have computed for every pixel the abrupt changes of unwrapped values in its neighborhood of n pixels. It worked reasonably well, nevertheless recall that according to our idea about an optimal SL strategy such additional image processing should be certainly avoided and a full projector resolution should be utilized. To further visualize the effect of decreased 3D resolution, we show in Fig. 5 the appearance of the manikin head using our approach, the Gray code + PS and the number theoretic approach. Obviously, the number theoretic approach gives the worst result. This experiment is completed showing profiles of the generated mesh of 3D reconstructed points (Fig. 7a). Our proposed method shows a very good match of the reconstruction to the original object's silhouette. For completeness, we reveal that the bounding box encompassed by the reconstructed volume for the proposed method (System B option, Gray code + PS and number theoretic approach) was 142 mm (width) $\times$ 248 mm (height) $\times$ 157 mm (depth).

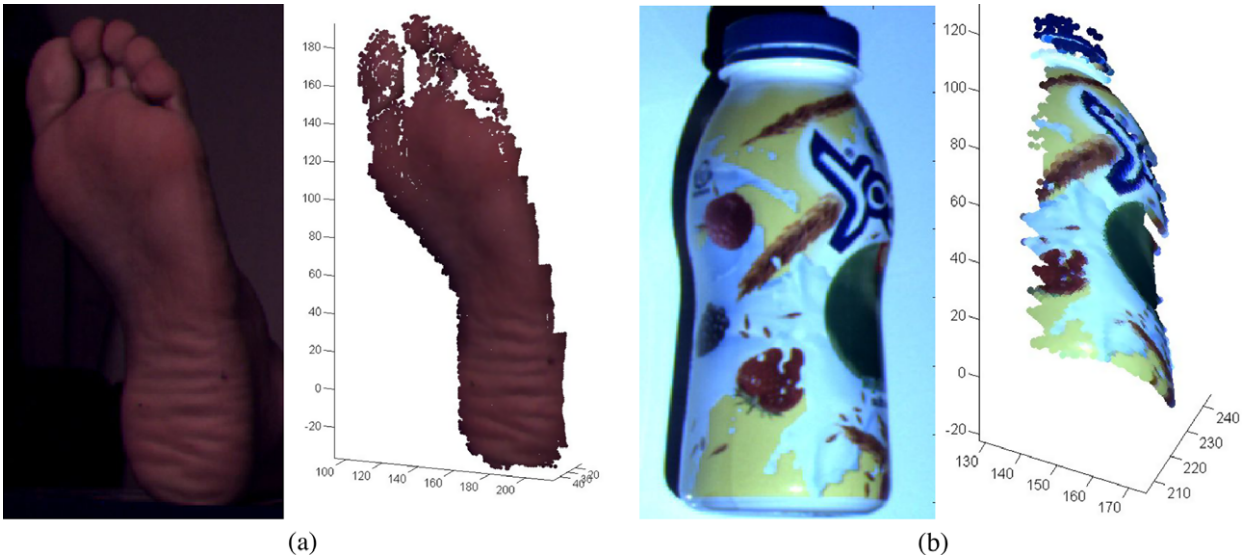

Fig. 6. System A. (a) 3D scan of the foot: image and reconstructed surface. (b) 3D scan of the yogurt bottle: image and reconstructed surface.

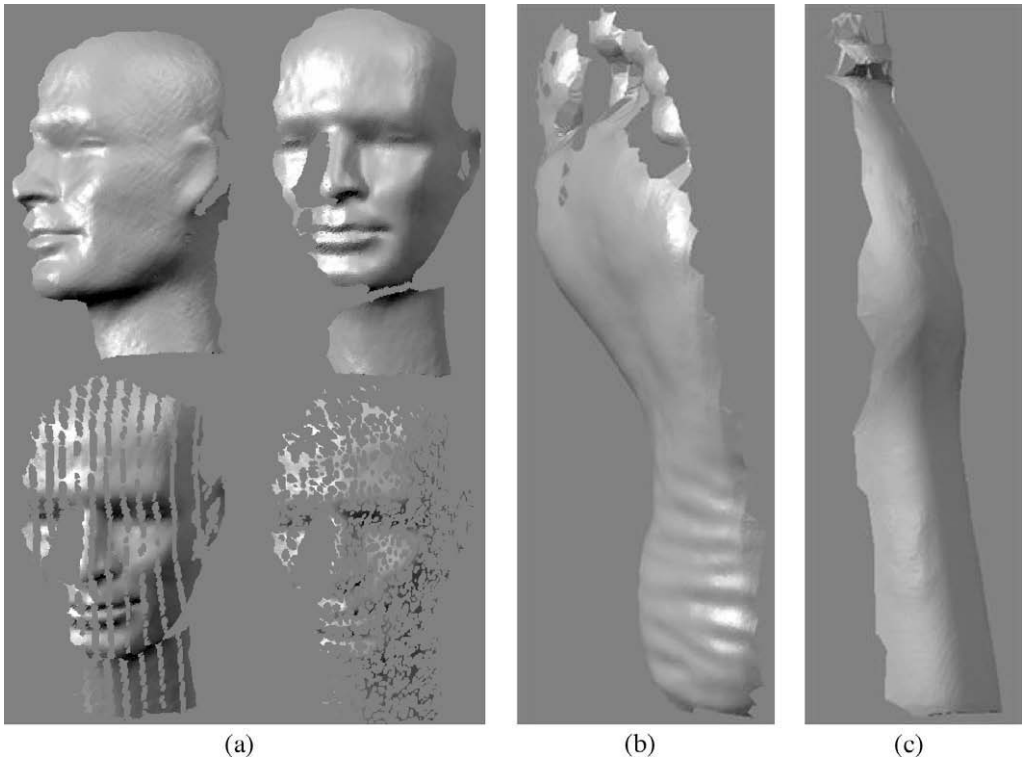

Fig. 7. Profile views of the reconstructed objects: (a) manikin head: System A (top left); System B (top right); Gray code + PS (bottom left); theoretic approach (bottom right). (b) Foot: System A. (c) Yogurt bottle: System A.

Finally, we compare the performance of System A and System B. The 3D reconstruction accuracy figures (Table 1 vs. Table 2) give a slight advantage to System A, which is perfectly in accordance with the theoretical expectations outlined in Section 4. Whether this will actually make a difference will most likely vary in practice from one application to another. In fact, the difference could increase or decrease given other system parameters (e.g., the camera quality, the projector, the size of the calibration volume, or the complexity of the calibration/reconstruction algorithm). We believe that in our case the major cause is the different ways the camera and projector image coordinates are undistorted (prior to triangulation) when using the System A and System B designs. Lens (un)distortion is usually modeled with a polynomial expression where the distance r from an image point (u, v) to what is believed to be the center of distortion plays one of the key roles [45,46]. System A triangulates 3D points using image points from two cameras where both image coordinate components u and v are available, and r is computable. System B, however, only has available the abscissa component u on the side of the projector, i.e., triangulation is based on the plane (projector) and the line (camera). Fortunately, with fairly good quality lenses it is safe to assume that the magnitude of (un)distortion for (u, v) will not change significantly in its

Table 2
Mean error and standard deviation of distances for total of N reconstructed points with respect to the fitted plane through those N points. p_1 and p_2 show the chosen number of periods for periodic patterns used to construct PS maps in the context. n is the Gray code word length. Note that for the pale plane, the width of the reconstructed flat surface ranged approximately between 300 mm and 370 mm and the height between 210 mm and 270 mm. Besides, for the colorful plane the dimensions of the reconstructed flat surface were the ones of an A4 paper, i.e. 297 mm × 210 mm.

p_1	p_2	Pale plane			Colorful plane		
		N	Mean (mm)	Std. (mm)	N	Mean (mm)	Std. (mm)
System B							
1	3	70887	0.273	0.228	69678	0.347	0.303
3	5	72355	0.203	0.178	72355	0.257	0.244
5	7	72395	0.154	0.139	72395	0.245	0.236
7	11	72185	0.159	0.144	72185	0.233	0.221
11	15	72066	0.155	0.140	72066	0.220	0.199
15	19	71531	0.152	0.137	71531	0.227	0.209
17	21	71374	0.158	0.145	71374	0.225	0.210
17	23	71385	0.155	0.137	71385	0.222	0.213
19	25	71194	0.153	0.137	71194	0.219	0.208
20	27	71081	0.149	0.139	71081	0.213	0.207
25	27	70396	0.156	0.140	70396	0.210	0.195
Number theoretic approach							
1	3	69678	0.250	0.209	67543	0.389	0.362
3	5	50675	0.207	0.172	53721	0.271	0.258
5	7	43218	0.160	0.145	47392	0.239	0.228
7	11	29660	0.144	0.129	28411	0.244	0.232
11	15	31145	0.157	0.148	26567	0.231	0.204
15	19	25800	0.158	0.141	27123	0.220	0.199
17	21	14009	0.150	0.140	15871	0.234	0.229
17	23	9472	0.148	0.132	11457	0.218	0.209
19	25	10848	0.151	0.136	9563	0.201	0.196
20	27	8456	0.148	0.135	9321	0.223	0.215
25	27	7807	0.152	0.138	8267	0.205	0.192
n	p_1	Pale plane			Colorful plane		
		N	Mean (mm)	Std. (mm)	N	Mean (mm)	Std. (mm)
Gray code + PS							
2	4	70803	0.281	0.233	71345	0.338	0.297
3	8	54089	0.165	0.151	52974	0.251	0.221
4	16	43123	0.158	0.142	40985	0.231	0.219
5	32	39213	0.148	0.137	39011	0.210	0.197
6	64	30213	0.149	0.133	31117	0.213	0.204

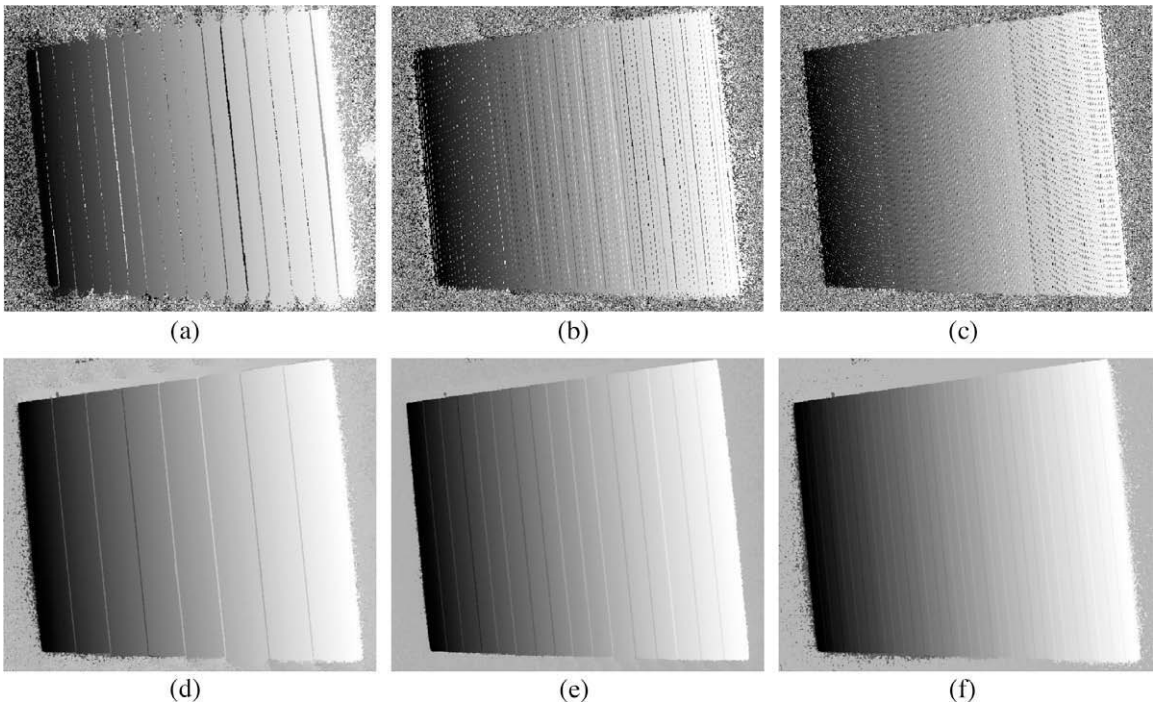

Fig. 8. Top row: number theoretic approach. The effect on the appearance of the absolute phase map using different periods (p_1, p_2) for two relative phase maps: (a) (3, 5), (b) (7, 11), (c) (15, 19). Bottom row: Gray code + PS. The effect on the appearance of the absolute phase map using different Gray code word lengths: (d) three bits (e) four bits (f) five bits.

vicinity. Thus, if we could at least approximate the v component on the side of the projector, we could use this value to compute distance r and subsequently undistort the u component, which after de-codification we do know accurately. First we triangulate an approximate position of a 3D point using the distorted u projection coordinate component. Then, we simply project this 3D point back to the projector image plane, finding the approximate value for v. In turn, we are able compute distance r and ultimately undistort the u component. A straightforward alternative would be, at the expense of doubling the number of projected patterns, to project patterns in the directions of both projector image axes. In fact, when this is done the accuracy results (not shown in the paper) hardly differ between System A and System B. We may also recall that we did this during the projector calibration. However, given that the projector and camera(s) are firmly attached, over a longer period of time calibration is generally considered to be a one-time-effort.

6. Conclusion

We have considered numerous well-known SL strategies and concluded that MPS is one of the very few that fulfils many of the desired conditions, as laid out in the introductory section, for an optimal SL strategy. For experimental comparison with our proposal we have identified and implemented two popular SL solutions which also satisfy set conditions and in theory require a relatively small number of patterns: Gray code + single PS and a number theoretic approach. However a simple, straightforward implementation of these two solutions is inferior, compared to our proposed MPS method, in terms of attainable reconstruction resolution. Related to it, these two solutions would also require an additional image processing (time consumption) to detect pixels with the incorrect absolute phase values.

Our MPS method successfully unwraps relative phases and accurately performs dense 3D acquisition of various objects, using no more than two PS maps. We construct each PS map using eight shifted images. An additional two images are used as references to facilitate the detection of shadows and occluded regions. Eighteen patterns is a reasonable compromise, particularly in cases where a pair of cameras is synchronized to the projector, since then the entire pattern projection/image grabbing should not last more than a second. The theoretical background of our method is quite simple and is based on the observation that given a certain range of values along the abscissa axis of absolute values, one encounters unique pairs of relative phase values. We have also offered straightforward proof of the uniqueness mentioned, which is typically left out of published works. Through experimenting, we have shown a respectable degree of accuracy when reconstructing planar surfaces, even when the surface is particularly colorful. In addition, a qualitative inspection of the acquisition of non-coplanar surfaces has proven to be very good, in cases where both monochromatic surfaces and colored surfaces are reconstructed.

We have demonstrated the proposed method with two different hardware designs, System A (an uncalibrated video projector and a pair of calibrated cameras) and System B (a calibrated video projector and a single camera). Besides experimental comparison, we have also pointed out the major arguments in favor of using each. In addition, we have contributed to the issue of how to accurately find correspondences between cameras using the System A design.

Appendix A

We give here some details for deriving an expression to compute the relative phase value φ_R from a least square minimization of Eq. (3) – in other words, minimizing the following sum of squares, over the parameter φ_R:

$$\varepsilon = \sum_{i=0}^{N-1} [I_i - (I_0 + A \cdot \sin(\varphi_R - \varphi_i))]^2 \tag{A.1}$$

The minimum of Eq. (A.1) is easily found by setting a derivative of ε over φ_R to zero:

$$\frac{\partial \varepsilon}{\partial \varphi_R} = \sum_{i=0}^{N-1} I_i \cdot \cos(\varphi_R - \varphi_i) - I_0 \cdot \sum_{i=0}^{N-1} \cos(\varphi_R - \varphi_i) - A \cdot \sum_{i=0}^{N-1} \cos(\varphi_R - \varphi_i) \cdot \sin(\varphi_R - \varphi_i) = 0 \tag{A.2}$$

It is useful to recall that the following equations hold, assuming sampling of φ_i as shown in (1) and for $N \geqslant 3$:

$$\begin{aligned} &\cos(\varphi_R - \varphi_i) = \cos(\varphi_R) \cdot \cos(\varphi_i) + \sin(\varphi_R) \cdot \sin(\varphi_i) \\ &\sin(\varphi_R - \varphi_i) = \sin(\varphi_R) \cdot \cos(\varphi_i) - \sin(\varphi_i) \cdot \cos(\varphi_R) \\ &\sum_{i=0}^{N-1} \cos(\varphi_i) = \sum_{i=0}^{N-1} \sin(\varphi_i) = \sum_{i=0}^{N-1} \sin(\varphi_i) \cdot \cos(\varphi_i) = 0 \\ &\sum_{i=0}^{N-1} \cos^2(\varphi_i) = \sum_{i=0}^{N-1} \sin^2(\varphi_i) \end{aligned} \tag{A.3}$$

Eq. (A.3) allows a substantial simplification of Eq. (A.2). After some algebraic manipulation it becomes clear that second and third summands in (A.2) vanish and we obtain:

$$\begin{aligned} &\frac{\partial \varepsilon}{\partial \varphi_R} = \sum_{i=0}^{N-1} I_i \cdot \cos(\varphi_R - \varphi_i) - I_0 \cdot 0 - A \cdot 0 = 0 \\ &\cos(\varphi_R) \cdot \sum_{i=0}^{N-1} I_i \cdot \cos(\varphi_i) + \sin(\varphi_R) \cdot \sum_{i=0}^{N-1} I_i \cdot \sin(\varphi_i) = 0 \end{aligned} \tag{A.4}$$

Finally, we are able to write the same expression as already shown in (4):

$$\varphi_R = \operatorname{atan}\left(-\sum_{i=0}^{N-1} I_i \cdot \cos(\varphi_i),\ \sum_{i=0}^{N-1} I_i \cdot \sin(\varphi_i) \right) \tag{A.5}$$

where atan is the four-quadrant inverse tangent function, yielding angle φ_R in the interval $[-\pi, \pi]$.

References

[1] R. Klette, K. Schluns, A. Koschan, Computer Vision: Three-Dimensional Data from Images, Springer Verlag, 1998.
[2] N.L. Chang, Efficient dense correspondences using temporally encoded light patterns, in: IEEE International Workshop on Projector-Camera Systems in Conjunction with ICCV, Nice, France, 2003, pp. 1–8.
[3] J. Salvi, J. Pagés, J. Batlle, Pattern codification strategies in structured light systems, Pattern Recognition 37 (4) (2004) 827–849.
[4] S. Gorthi, K. Lolla, A new approach for simple and rapid shape measurement of objects with surface discontinuities, Proceedings of SPIE 5856 (2005) 184–194.
[5] N. Ono, T. Shimizu, T. Kurihara, S. Ando, Real-time 3-D imager based on spatio-temporal phase unwrapping, in: SICE Annual Conference, Sapporo, Japan, 2004, pp. 2544–2547.
[6] H. Saldner, J. Huntley, Temporal phase unwrapping: application to surface profiling of discontinuous objects, Applied Optics 36 (13) (1997) 2770–2775.
[7] J. Huntley, H. Saldner, Shape measurement by temporal phase unwrapping: comparison of unwrapping algorithms, Measurement Science and Technology 8 (9) (1997) 986–992.
[8] W. Nadeborn, P. Andrä, W. Osten, A robust procedure for absolute phase measurement, Optics and Lasers in Engineering 24 (2–3) (1996) 245–260.
[9] T. Jin-dong, P. Xiang, Z. Xiao-bo, A pitch-variation moiré fringes method of temporal phase unwrapping profilometry, Optoelectronics Letters 3 (3) (2007) 215–218.
[10] J. Burke, T. Bothe, W. Osten, C. Hess, Reverse engineering by fringe projection, Proceedings of SPIE 4778 (2002) 312–324.
[11] V.I. Gushov, Y.N. Solodkin, Automatic processing of fringe patterns in integer interferometers, Optics and Lasers in Engineering 14 (4) (1991) 311–324.
[12] P. Ribenboim, Algebraic Numbers, John Wiley and Sons Inc., New York, 1972.

[13] G. Sansoni, A. Patrioli, F. Docchio, OPL-3D: a novel, portable optical digitizer for fast acquisition of free-form surfaces, Review of Scientific Instruments 74 (4) (2003) 2593–2603.
[14] G. Wiora, High resolution measurement of phase-shift amplitude and numeric object phase calculation, in: Proceedings of SPIE Vision Geometry IX, vol. 4117, 2000, pp. 289–299.
[15] C. Guan, L. Hassebrook, D. Lau, Composite structured light pattern for three-dimensional video, Optics Express 11 (5) (2003) 406–417.
[16] M. Tehrani, A. Saghaeian, O. Mohajerani, A new approach to 3D modeling using structured light pattern, in: 3rd ICCTA, Damascus, Syria, 2008, pp. 1–5.
[17] J. Pagès, J. Salvi, J. Forest, Optimized De Bruijn patterns for one-shot shape acquisition, Image and Vision Computing 23 (8) (2005) 707–720.
[18] H. Kawasaki, R. Furukawa, R. Sagawa, Y. Yagi, Dynamic scene shape reconstruction using a single structured light pattern, in: CVPR, Anchorage, AK, USA, 2008, pp. 1–8.
[19] T. Koninckx, L. Van Gool, Real-time range acquisition by adaptive structured light, TPAMI 28 (3) (2006) 432–445.
[20] P. Griffin, L. Narasimhan, S. Yee, Generation of uniquely encoded light patterns for range data acquisition, Pattern Recognition 25 (6) (1992) 609–616.
[21] C. Albitar, P. Graebling, C. Doignon, Design of a monochromatic pattern for a robust structured light coding, in: IEEE International Conference on Image Processing, San Antonio, TX, USA, 2007, pp. 529–532.
[22] M. Ito, A. Ishii, A three-level checkerboard pattern (TCP) projection method for curved surface measurement, Pattern Recognition 28 (1) (1995) 27–40.
[23] D. Caspi, N. Kiryati, J. Shamir, Range imaging with adaptive color structured light, TPAMI 20 (5) (1998) 470–480.
[24] F. Forster, A high-resolution and high accuracy real-time 3D sensor based on structured light, in: 3rd International Symposium on 3D Data Processing, Visualization, and Transmission, University of North Carolina, Chapel Hill, USA, 2006, pp. 208–215.
[25] P. Fechteler, P. Eisert, Adaptive color classification for structured light systems, IET Computer Vision 3 (2) (2009) 49–59.
[26] L. Zhang, B. Curless, S. Seitz, Rapid shape acquisition using color structured light and multi-pass dynamic programming, in: Proceedings of the 1st International Symposium on 3D Data Processing, Visualization, and Transmission, Padova, Italy, 2002, pp. 24–36.
[27] B. Carrihill, R. Hummel, Experiments with the intensity ratio depth sensor, Computer Vision, Graphics, and Image Processing 32 (3) (1985) 337–358.
[28] C. Wust, D. Capson, Surface profile measurement using color fringe projection, Machine Vision and Applications 4 (3) (1991) 193–203.
[29] J. Tajima, M. Iwakawa, 3-D data acquisition by rainbow range finder, in: 10th International Conference on Pattern Recognition, Atlantic City, NJ, USA, 1990, pp. 393–313.
[30] T. Sato, Multispectral pattern projection range finder, in: Proceedings of SPIE, Conference on Three-Dimensional Image Capture and Applications II, San Jose, California, vol. 3640, 1999, pp. 28–37.
[31] G. Chazan, N. Kiryati, Pyramidal Intensity-Ratio Depth Sensor, Technical Report 121, Center for Communication and Information Technologies, Department of Electrical Engineering, Technion, Haifa, Israel, 1995, pp. 1–22.
[32] M. Takeda, M. Mutoh, Fourier transform profilometry for the automatic measurement of 3-D object shapes, Applied Optics 22 (1983) 3977–3982.
[33] X. Su, W. Chen, Fourier transform profilometry: a review, Optics and Lasers in Engineering 35 (5) (2001) 263–284.
[34] J. Li, X. Su, L. Gou, An improved Fourier transform profilometry for automatic measurement of 3-D object shapes, Optical Engineering 29 (12) (1990) 1439–1444.
[35] E. Hu, Y. He, Surface profile measurement of moving objects by using an improved π phase-shifting Fourier transform profilometry, Optics and Lasers in Engineering 47 (1) (2009) 57–61.
[36] W. Chen, P. Bu, S. Zheng, X. Su, Study on Fourier transforms profilometry based on bi-color projecting, Optics and Laser Technology 39 (4) (2007) 821–827.
[37] H. Yue, X. Su, Y. Liu, Fourier transform profilometry based on composite structured light pattern, Optics and Laser Technology 39 (6) (2007) 1170–1175.
[38] R. Sedgewick, Algorithms in C, Addison-Wesley, New York, USA, 1998.
[39] Z. Zhang, A flexible new technique for camera calibration, TPAMI 22 (11) (2000) 1330–1334.
[40] T. Pribanić, P. Sturm, M. Cifrek, Calibration of 3D kinematic systems using orthogonality constraints, Machine Vision and Applications 18 (6) (2007) 367–381.
[41] T. Pribanić, M. Cifrek, S. Peharec, Simplified light plane determination during structured light scanning, Automatika Journal 47 (3–4) (2006) 141–147.
[42] W.H. Press, S.A. Teukolsky, W.T. Vetterling, B.P. Flannery, Numerical Recipes in C, Cambridge University Press, Cambridge, 1997.
[43] J. Salvi, C. Matabosch, D. Fofi, J. Forest, A review of recent range image registration methods with accuracy evaluation, Image and Vision Computing 25 (5) (2007) 578–596.
[44] C. Coggrave, J. Huntley, Optimization of a shape measurement system based on spatial light modulators, Optical Engineering 39 (1) (2000) 91–98.
[45] G.Q. Wei, S.D. Ma, Implicit and explicit camera calibration: theory and experiment, TPAMI 16 (5) (1994) 469–480.
[46] J. Salvi, X. Armangue, J. Batlle, A comparative review of camera calibrating methods with accuracy evaluation, Pattern Recognition 35 (7) (2002) 1617–1635.