applied optics

Method for out-of-focus camera calibration

Tyler Bell,[1] Jing Xu,[2] and Song Zhang[1,*]

[1]School of Mechanical Engineering, Purdue University, West Lafayette, Indiana 47907, USA
[2]Department of Mechanical Engineering, Tsinghua University, Beijing 10084, China
*Corresponding author: szhang15@purdue.edu

Received 5 January 2016; revised 15 February 2016; accepted 18 February 2016; posted 18 February 2016 (Doc. ID 256848); published 17 March 2016

State-of-the-art camera calibration methods assume that the camera is at least nearly in focus and thus fail if the camera is substantially defocused. This paper presents a method which enables the accurate calibration of an out-of-focus camera. Specifically, the proposed method uses a digital display (e.g., liquid crystal display monitor) to generate fringe patterns that encode feature points into the carrier phase; these feature points can be accurately recovered, even if the fringe patterns are substantially blurred (i.e., the camera is substantially defocused). Experiments demonstrated that the proposed method can accurately calibrate a camera regardless of the amount of defocusing: the focal length difference is approximately 0.2% when the camera is focused compared to when the camera is substantially defocused.

OCIS codes: (120.0120) Instrumentation, measurement, and metrology; (120.2650) Fringe analysis; (100.5070) Phase retrieval.

http://dx.doi.org/10.1364/AO.55.002346

1. INTRODUCTION

Two- and three-dimensional vision systems typically use at least one calibrated camera to capture images for information analytics. Measurement accuracy heavily hinges on the accuracy of camera calibration, thus accurate and flexible camera calibration has been extensively studied over the past few decades.

Accurate camera calibration can be carried out by using highly accurate fabricated and measured 3D calibration targets [1,2]; since the 3D dimensions of these targets are known, the transformation from 3D world coordinates to the 2D imaging plane can be estimated through optimization methods. However, fabricating such highly accurate 3D targets is sometimes difficult and usually very expensive. Since using a 3D target is equivalent to moving a 2D planar object perpendicular to its surface, Tsai [3] proposed a method to use 2D calibration targets with rigid out-of-plane movements to accurately calibrate a camera. By using a 2D flat surface, Tsai's method simplifies the target fabrication process since a 2D flat surface is easier to obtain. However, such a method requires the use of a high-precision translation stage, which is often expensive. To further simplify the camera calibration process, Zhang [4] proposed a flexible camera calibration method that allows the use of a planar 2D target with arbitrary poses and orientations, although it requires some known-dimension feature points (e.g., checkerboard or circle patterns) on the target surface. Image processing algorithms are then used to detect those feature points for camera calibration. Zhang's method is, by far, the most extensively adopted method due to its flexibility and ease of use.

Recently, researchers have developed more flexible camera calibration approaches by using unknown feature points or even imperfect calibration targets [5–8]. Instead of fabricating a calibration target, researchers have demonstrated that active targets (e.g., digital displays) can also be used to accurately calibrate cameras [9] and can further improve calibration accuracy [10] since feature points can be more accurately defined and located. Similar active pattern approaches have also been used to calibrate a system using a fish-eye lens [11].

To our knowledge, the state-of-art camera calibration methods were primarily developed for close-range vision systems (i.e., the sensing range is usually rather small such that the calibration target used to calibrate the camera can be accurately and feasibly fabricated). However, long-range vision systems are becoming increasingly used for applications such as navigation and large-scale measurements. Given this, if high levels of accuracy are required for long-range vision systems, a camera calibration method for such systems is required. The challenge then becomes the fabrication of a calibration target that is in the same order of size as the system's range. Obviously, scaling the calibration target to the scale of a long-range system becomes increasingly difficult in terms of fabrication accuracy, feasibility, and cost.

This paper proposes a method that allows the use of a smaller calibration target for large-range vision system calibration. The aforementioned state-of-the-art camera calibration methods assume that the camera is at least nearly focused on the calibration target, and thus they fail if the camera is substantially defocused. This paper presents a method that enables the calibration of an out-of-focus camera to conquer the

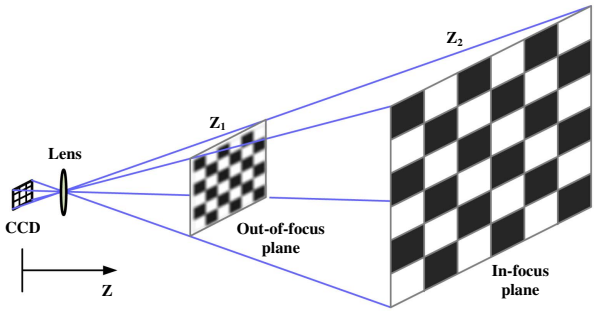

Fig. 1. Illustration of placing the calibration target at different distances from the camera. Compared to putting the calibration target at the focal plane (Z_2), the calibration target dimensions could be substantially smaller if the calibration target is placed at the out-of-focus plane (Z_1). However, if the target is placed at Z_1, the captured images are blurred, failing the current state-of-the-art calibration methods that assume the camera is at least nearly focused.

challenges of calibrating large-range vision systems. By allowing for the placement of the calibration target closer to the camera than the sensing plane, the calibration target size can be substantially smaller, as illustrated in Fig. 1. Similar to the previously proposed calibration methods that use active targets [9–11], we also use an active digital display (e.g., a liquid crystal display monitor) to generate fringe patterns that encode feature points into the carrier phase, which can be accurately recovered even if the fringe patterns are substantially blurred (i.e., camera is substantially defocused). Instead of using phase-shifted sinusoidal patterns, we use phase-shifted square binary patterns for phase generation to enhance fringe contrast when the patterns are substantially defocused and to eliminate the influence of digital display nonlinearity. Experimental results demonstrate that the proposed camera calibration method can accurately calibrate a camera regardless of the amount of defocusing.

Section 2 explains the principles of the proposed out-of-focus camera calibration method. Section 3 shows some simulation results to validate the proposed method. Section 4 presents experimental results to further validate the proposed method. Lastly, Section 5 summarizes the paper.

2. PRINCIPLE

This section thoroughly explains the principle of the proposed method. Specifically, we will present the standard pinhole camera model followed by the feature point encoding and subpixel feature point extraction framework we developed to calibrate a camera regardless of its amount of defocusing.

A. Camera Lens Model

In this research, we use a well-known pinhole model to describe a camera lens. This model essentially describes the relationship between 3D-world coordinates (x^w, y^w, z^w) and its projection onto the 2D-imaging coordinates (u, v). For a linear system, without considering lens distortion, the pinhole model can be mathematically described as

$$s\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} f_u & \gamma & u_0 \\ 0 & f_v & v_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \mathbf{R} & \mathbf{t} \end{bmatrix} \begin{bmatrix} x^w \\ y^w \\ z^w \\ 1 \end{bmatrix}. \tag{1}$$

Here s is the scaling factor; f_u and f_v are, respectively, the effective focal lengths of the camera along u and v directions; γ is the skew factor of u and v axes, for modern cameras $\gamma = 0$; and (u_0, v_0) is the principle point. In this equation,

$$\mathbf{R} = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \tag{2}$$

represents the rotation matrix from the world coordinate system and the camera lens coordinate system; and $\mathbf{t} = [t_1, t_2, t_3]^T$ describes the translation from the world coordinate system and the camera lens coordinate system.

If the camera lens is nonlinear, its distortion can be modeled as

$$\mathbf{D} = \begin{bmatrix} k_1 & k_2 & p_1 & p_2 & k_3 \end{bmatrix}^T, \tag{3}$$

where k_1, k_2, and k_3 are the radial distortion coefficients, which can be rectified by

$$u' = u(1 + k_1 r^2 + k_2 r^4 + k_3 r^6), \tag{4}$$

$$v' = v(1 + k_1 r^2 + k_2 r^4 + k_3 r^6). \tag{5}$$

Here (u', v') are the camera coordinates after nonlinear distortion corrections, and $r = \sqrt{(u - u_0)^2 + (v - v_0)^2}$ represents the absolute distance between the camera point and the origin. Similarly, tangential distortion can be corrected using the following formula:

$$u' = x + [2p_1 uv + p_2(r^2 + 2u^2)], \tag{6}$$

$$y' = y + [p_1(r^2 + 2v^2) + 2p_2 uv]. \tag{7}$$

B. Feature Point Encoding

As introduced in Section 1, one of the most extensively adopted camera calibration approaches uses a planar object with a number of feature points with known dimensions on a flat plane [4]. Typically, a checkerboard or a circle pattern is printed on a flat surface [12]. A method such as this first captures a sequence of images of the calibration object placed at different poses. Then, image processing is performed to detect the known feature points within the sets of images. Lastly, optimization algorithms are used to estimate camera calibration parameters.

Since only a number of known-dimension feature points are needed for calibration, these feature points can be generated digitally by a digital display device (e.g., an LCD monitor). If the monitor is flat, the same calibration approach can be adopted for camera calibration. Furthermore, since the feature points are discretely defined by LCD monitor pixel locations, as long as those pixels can be located, no real circle pattern or checkerboard is necessary for camera calibration. In this research, we use phase information to define such feature points.

Figure 2 illustrates the framework of using phase information to encode desired feature points, and we propose to use such a framework to calibrate an out-of-focus camera. Briefly, the desired feature points for camera calibration are encoded by horizontal and vertical phase maps. These phase maps are then further carried along by phase-shifted fringe patterns, which can be used to recover the original phase using phase-shifting algorithms.

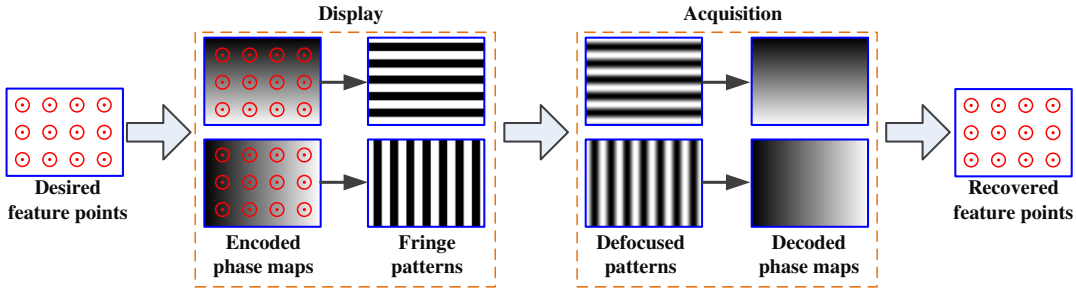

Fig. 2. Proposed framework for out-of-focus camera calibration. The target feature points are carried by the uniquely defined horizontal and vertical phase maps. Each phase map is the resultant of a set of phase-shifted binary structured patterns. These patterns are then displayed on an LCD monitor. The out-of-focus camera captures those structured patterns, which will be blurred according to the camera's level of defocus. These captured patterns are then used to recover horizontal and vertical phase maps from which the encoded feature points are decoded. The out-of-focus camera can then be calibrated using the recovered feature points.

Phase-shifting algorithms are extensively adopted in optical metrology, mainly because of their accuracy and robustness to both noise and ambient lighting effects. In general, for N-equally phase-shifted fringe patterns, the fringe patterns can be mathematically described as

$$I^i(x,y) = I'(x,y) + I''(x,y)\cos(\phi + 2i\pi/N), \tag{8}$$

where $I'(x,y)$ is the average intensity, $I''(x,y)$ is the intensity modulation, $i = 1, 2, \cdots, N$, and $\phi(x,y)$ is the phase to be solved by

$$\phi(x,y) = -\tan^{-1}\left[\frac{\sum_{i=1}^{N} I^i \sin(2i\pi/N)}{\sum_{i=1}^{N} I^i \cos(2i\pi/N)}\right]. \tag{9}$$

This equation produces the wrapped phase ranging from $-\pi$ to $+\pi$. To obtain the continuous phase without 2π discontinuities, one can use a spatial or temporal phase unwrapping method. The essence of phase unwrapping is to find the fringe order $k(x,y)$ for each pixel so that the phase can be unwrapped as

$$\Phi(x,y) = \phi(x,y) + k \times 2\pi. \tag{10}$$

The fundamental difference between spatial phase unwrapping and temporal phase unwrapping is that the spatial phase unwrapping algorithm finds $k(x,y)$ by analyzing the difference between the point to be processed and its neighboring pixels. In other words, the phase obtained using a spatial phase unwrapping algorithm is relative to one point, and thus the unwrapped phase is often called the *relative* phase. Temporal phase unwrapping algorithms, in contrast, uniquely find the phase values for each independent point without referring to the phase information of any neighboring pixel; thus such a method can recover *absolute* phase. Given this, to uniquely carry phase information, using absolute phase is necessary.

In this research, we adopted a temporal phase unwrapping method which uses gray-coded binary patterns to uniquely determine fringe order, $k(x,y)$, and to unwrap the *absolute* phase values. Any subsequent unwrapping artifacts were eliminated by using the computational framework introduced in Ref. [13].

Once absolute phase maps are uniquely defined, they can be used to encode any arbitrary number of points at any location on the monitor since they are encoded in the continuous phase. This differs from the calibration method developed by Li *et al.* [12], where a physical calibration board, with printed circular patterns, was used as feature points. To be properly captured and identified by an in-focus camera, these circles have to be large enough for the camera to capture, so that these feature centers can be determined accurately from the camera images. Therefore, this requirement places a constraint on the total number of feature points that can fit on the board, unlike the proposed method where an arbitrary number of points can be used as feature points. For example, a point (u_0^d, v_0^d) on the monitor can be encoded as

$$u_0^d \leftarrow \Phi_v(u_0^d, v_0^d), \tag{11}$$

$$v_0^d \leftarrow \Phi_h(u_0^d, v_0^d). \tag{12}$$

Here Φ_v represents the vertical phase map that varies along u^d direction, and Φ_h represents the horizontal phase map that varies along v^d direction. These phase values can be further discretely represented as

$$\Phi_v(u_0^d, v_0^d) = 2\pi u_0^d / P_v, \tag{13}$$

$$\Phi_h(u_0^d, v_0^d) = 2\pi v_0^d / P_h. \tag{14}$$

Here P_v and P_h, respectively, represent the vertical and horizontal number of pixels to represent 2π. And thus

$$u_0^d = \frac{\Phi_v(u_0^d, v_0^d)}{2\pi} \times P_v, \tag{15}$$

$$v_0^d = \frac{\Phi_h(u_0^d, v_0^d)}{2\pi} \times P_h. \tag{16}$$

As discussed earlier in this section, the phase can be carried along by N-equally phase-shifted fringe patterns

$$I_v^i(u,v) = 127.5[1 + \cos(\Phi_v + 2i\pi/N], \tag{17}$$

$$I_h^i(u,v) = 127.5[1 + \cos(\Phi_u + 2i\pi/N], \tag{18}$$

where $i = 1, 2, \ldots, N$. Once these phase-shifted fringe patterns and gray-coded binary patterns are captured by a camera, the absolute phase maps $\Phi_v(u^c, v^c) = \Phi_v(u^d, v^d)$ and $\Phi_h(u^c, v^c) = \Phi_h(u^d, v^d)$ can be computed for each camera pixel. These phase maps can be used to uniquely find the corresponding mapped points on the LCD pixel coordinates (u^d, v^d) for any point on the camera (u^c, v^c).

C. Subpixel Feature Point Extraction

In theory, the phase can accurately carry encoded information (e.g., feature points). For example, to encode a feature point, (u, v), we can represent the same information in the phase domain, using horizontal and vertical phase as (Φ_v, Φ_h), and if the phase is unique, the mapping from (u, v) to (Φ_v, Φ_h) is one to one. However, in practice, due to the nature of discrete fringe generation, and the sampling of the camera, unique one-to-one mappings cannot always be guaranteed. The defocusing of the lens further makes this one-to-one mapping a rarity, in practice. In general, without loss of generality, there are two scenarios: 1) the camera pixel is larger than the LCD monitor pixel and 2) the camera pixel is smaller than the LCD pixel. The former corresponds to the case when the camera is far away from the LCD screen. For this case, some pixels on the LCD monitor may not have corresponding sampled pixels on the camera, as illustrated in Fig. 3(a), where the encoded pixel (u_0^c, v_0^c) is not resolved by the camera. The latter corresponds to the case when the camera is very close to the LCD screen, indicating that many camera pixels may correspond to one LCD pixel, as illustrated in Fig. 3(b). For either case, additional processing is required to accurately recover the feature point from the phase maps. For example, previous works have performed bilinear interpolation using pixels surrounding the feature points to establish correspondence between the two phase maps [14]. Although successful, the bilinear interpolation could introduce bias error if the phase value of one or more of these four points has large error.

Instead of using the simple bilinear interpolation, in our research, we assumed that any given point on the camera is locally planar, and thus the plane could be used to accurately determine any corresponding points. To determine subpixel accuracy feature points, we used the following steps

- *Step 1: Camera to LCD mapping creation*. This mapping is generated by the horizontal and vertical phase maps, e.g., $(\Phi_v, \Phi_h) \leftarrow (u^d, v^d)$. Since for each camera point (u^c, v^c), the horizontal and vertical phase values are unique, we can also establish the mapping $(u^c, v^c) \leftarrow (u^d, v^d)$.

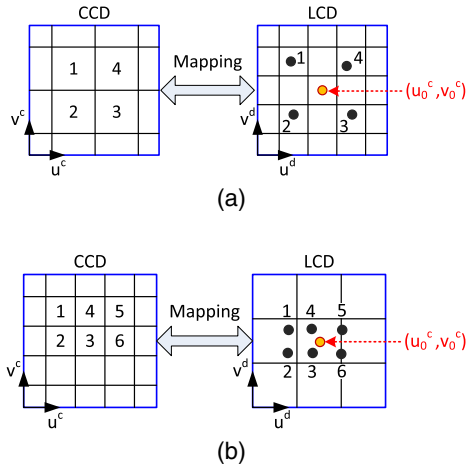

Fig. 3. Mapped feature point establishment through horizontal and vertical phase maps. (a) Camera pixel is larger than LCD pixel and (b) camera pixel is smaller than LCD pixel.

- *Step 2: Local plane fitting*. For any feature point (u_0^d, v_0^d), locally find all of the camera-mapped points (u_k^d, v_k^d). If we assume that those local mapped feature points are on the same plane, we will have

$$u_k^c = a_1 u_k^d + b_1 v_k^d + c_1, \tag{19}$$

$$v_k^c = a_2 u_k^d + b_2 v_k^d + c_2, \tag{20}$$

where a_1, b_1, c_1, a_2, b_2, and c_2 are plane coefficients. These coefficients can be determined by a least-square method using all of the local feature points.

- *Step 3: Subpixel feature point extraction*. Once the local plane functions are estimated, the coordinates for any given feature points (u_0^d, v_0^d) can be computed as

$$u_0^c = a_1 u_0^d + b_1 v_0^d + c_1, \tag{21}$$

$$v_0^c = a_2 u_0^d + b_2 v_0^d + c_2. \tag{22}$$

3. SIMULATION

Li *et al.* [12] thoroughly proved that phase information is preserved regardless of projector lens defocusing. Briefly put, if an optical imaging system is defocused, a point on the object no longer converges to a point on the image plane but rather a blurred circular disk. However, considering the infinite light ray of the optical system, the center of a camera pixel, regardless of the amount of defocusing, corresponds to the peak intensity value of the circular disk. In practice, it is difficult to find the peak intensity value from an image due to surface reflectivity variation, however. In contrast, it is much easier to find the peak value through phase. Using phase, the center point still corresponds to the peak value of the circular disk regardless of the amount of defocusing.

We carried out some simulation to confirm that camera defocusing does not change the resultant phase of the phase-shifted fringe patterns. From diffraction theory, one may know that lens defocusing can be simulated by convolving the image with a Gaussian function

$$G(x, y) = \frac{1}{2\pi\sigma^2} \exp\left[-\frac{(x-\mu_x)^2 + (y-\mu_y)}{2\sigma^2}\right]. \tag{23}$$

Here (μ_x, μ_y) is the position of the focal center, and σ controls the width of the blurred image area.

Figure 4 shows one of the phase-shifted fringe patterns with different amounts of defocusing (e.g., different Gaussian filter sizes). Squared binary patterns, in lieu of sinusoidal patterns, were adopted in this research because 1) squared binary patterns provide the highest possible contrast when the patterns are substantially defocused, 2) binary patterns are not affected by the nonlinear gamma of the LCD monitor, and 3) Esktrand and Zhang [15] have demonstrated that accurate phase can be recovered even if the patterns are ideally squared binary. In this simulation, the squared binary patterns had a period of 42 pixels and 21 equally phase-shifted patterns were used to compute the phase.

The absolute differences between the ideal phase without defocusing and the phase with different amounts of defocusing are shown in Fig. 5. The root-mean-square (rms) errors are all very small, demonstrating that lens defocusing indeed does not alter the phase carried by the fringe patterns. Therefore, it is

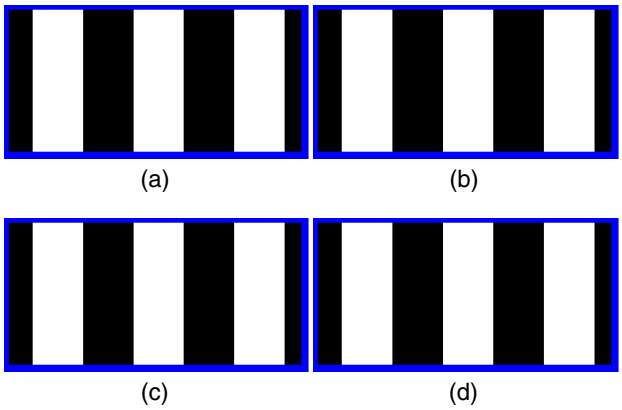

Fig. 4. Example squared binary fringe patterns when the structured patterns are defocused at different degrees. (a) A focused binary pattern, (b) the pattern after applying a Gaussian filter with a size of 9 × 9 pixels, (c) the pattern after applying a Gaussian filter with a size of 17 × 17 pixels, and (d) the pattern after applying a Gaussian filter with a size of 33 × 33 pixels.

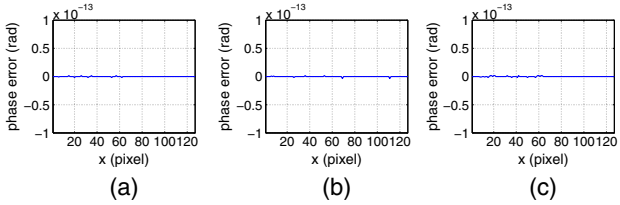

Fig. 5. Phase difference between blurred structured patterns and focused (without blur) patterns. (a) Gaussian filter size of 9 × 9 pixels (phase rms 3.9×10^{-16} rad), (b) Gaussian filter size of 17 × 17 pixels (phase rms 3.3×10^{-16} rad), and (c) Gaussian filter size of 33 × 33 pixels (phase rms 4.6×10^{-16} rad).

theoretically possible to encode the feature point information into phase to avoid the problems caused by camera lens defocusing.

4. EXPERIMENT

To verify the performance of the proposed method, we developed a camera calibration system that includes an LCD monitor (model: HP EliteDisplay E241i 24-inch IPS LED Backlit Monitor) and a charge-coupled device (CCD) camera (model: Jai Pulinx TM-6840CL) with a 12 mm focal length lens (model: Computar M1214-MP2). The LCD monitor has a resolution of 1920 × 1200 and a pixel pitch of 0.270 mm. The camera resolution is 640 × 480 with a pixel size of 7.4 μm × 7.4 μm. The lens is a 2/3-inch, 12 mm lens with an aperture of F/1.4-F/16C. The range of focus is approximately 150 mm to infinity.

For all of the following experiments, the camera focus remained constant and untouched to maintain the camera intrinsic parameters; differing amounts of defocus were realized by changing the distance, D, between the monitor and the camera. In this research, we tested four different amounts of defocusing from the camera being focused to the camera being substantially defocused (i.e., we used four different distances). The active areas of the monitors used for calibration for each distance are summarized in Table 1. This table shows that the calibration target size needs to be proportionally scaled up when the distance between the camera and the object increases, as illustrated in Fig. 1. Therefore, as discussed in Section 1, the conventional method of calibrating a camera lens is to use a larger calibration target when the sensing area is larger.

Table 1. LCD Monitor Pixels Used for Camera Lens Calibration

	Active Monitor Range (Pixels)
$D_1 = 950$ mm	1260 × 960
$D_2 = 540$ mm	720 × 540
$D_3 = 250$ mm	330 × 250
$D_4 = 125$ mm	165 × 125

The point of our research is to prove that it is not necessary to increase target size for camera calibration, but rather one can place the calibration target closer to the camera. Obviously, if the focal plane of the camera does not change, the camera will be out of focus; making the image blurry when the calibration target is placed away from its camera focal plane. Figure 6 shows some example camera images for those four different distances. The same square binary pattern was displayed on the LCD monitor, but the structured pattern was blurred if the camera was not focused [e.g., the image shown in Fig. 6(d)]; note that when the camera is away for the monitor, the patterns appear denser.

As discussed previously, since the feature points are carried along by phase values and not intensity values, the appearance of the structured patterns should not alter the feature points if the phase itself does not change. The horizontal and vertical phase maps were encoded with 12 equally phase-shifted, squared binary patterns with a fringe pitch of 24 pixels [i.e., $N = 12$ and $P_v = P_h = 24$ pixels for Eqs (15)–(18)]. A temporal phase unwrapping algorithm was used to unwrap the phase pixel by pixel, with unwrapping artifacts being removed using the computational framework discussed in Ref. [13].

Since the feature points are encoded in phase maps, they can be determined once the horizontal and vertical phase maps are

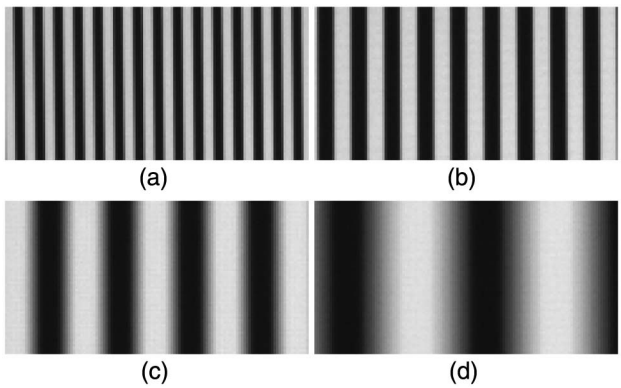

Fig. 6. Example images of a squared binary pattern when the camera is placed at different distances without changing its focus. (a) $D_1 = 950$ mm, (b) $D_2 = 540$ mm, (c) $D_3 = 250$ mm, and (d) $D_4 = 125$ mm.

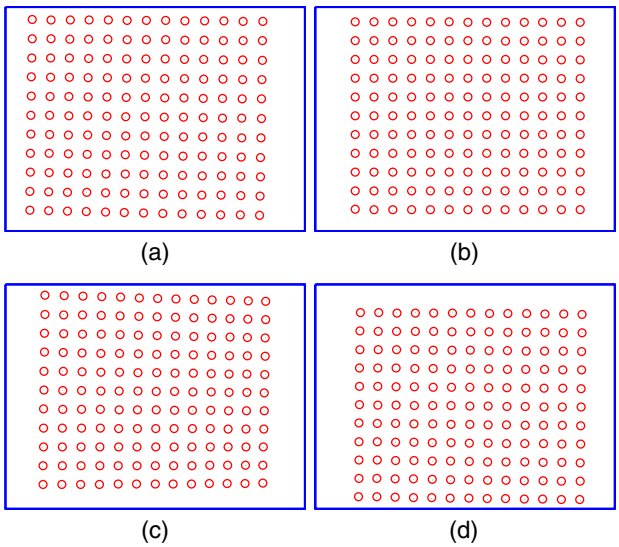

Fig. 7. Example detected feature points for one of the poses when the camera is at different amounts of defocusing (i.e., different distances from the LCD monitor). (a) $D_1 = 950$ mm, (b) $D_2 = 540$ mm, (c) $D_3 = 250$ mm, and (d) $D_4 = 125$ mm.

computed regardless of the amount of camera defocusing by using the proposed method discussed in Subsection C. Figure 7 shows some at different amounts of camera defocusing. It at least visually appears that all of the feature points are properly recovered.

Once the feature points are detected, the camera intrinsic parameters can be estimated using the standard camera calibration approach. In this research, we used the OpenCV camera calibration software package to estimate the intrinsic parameters of the camera lens. Table 2 summarizes the intrinsic parameters estimated from four different levels of defocusing. For each amount of defocusing, we captured 15 different target poses and used 143 feature points for each calibration plane. These experimental results clearly demonstrate that the equivalent focal lengths estimated from different amounts of defocusing are extremely close to each other: less than 0.2% difference. The principle points estimated from different amounts of defocusing are also very close to each other. One may notice the largest principle difference occurs when the camera is substantially defocused, which is still only approximately 1%.

Table 2. Intrinsic Parameters Estimated When the Camera is under Different Amounts of Defocusing

	f_u (mm)	f_v (mm)	u_0 (mm)	v_0 (mm)
$D_1 = 950$ mm	12.17	12.17	2.34	1.88
$D_2 = 540$ mm	12.16	12.16	2.34	1.88
$D_3 = 250$ mm	12.14	12.14	2.34	1.87
$D_4 = 125$ mm	12.16	12.16	2.32	1.89

Table 3. Distortion Coefficients Estimated When the Camera is under Different Amounts of Defocusing

	k_1	k_2
$D_1 = 950$ mm	−0.058	0.288
$D_2 = 540$ mm	−0.071	0.345
$D_3 = 250$ mm	−0.088	0.270
$D_4 = 125$ mm	−0.118	0.133

For our camera lens, we found that keeping k_1 and k_2 for nonlinear distortion is sufficient. Table 3 presents the estimated distortion coefficients, and they are all very small. The reprojection errors are, respectively, 0.033, 0.022, 0.029, and 0.058 pixels for $D_1 = 950$ mm, $D_2 = 540$ mm, $D_3 = 250$ mm, and $D_4 = 125$ mm. As can been seen, all of these reprojection errors are very small.

These experiments confirm that the calibration data are very close to each other when the camera is under different amounts of defocusing. One may notice that there are some slight differences between the different amounts of defocusing, which might be a result of the disparity between calibration poses used at each level of defocusing and/or the different number of pixels used to display the encoded fringe patterns.

5. SUMMARY

This paper has presented an out-of-focus camera calibration approach by encoding the calibration feature points into phase, which are further carried along by phase-shifted fringe patterns displayed by an LCD monitor. Experiments demonstrated that the proposed method can accurately calibrate camera intrinsic parameters (e.g., focal length, principle points) regardless of the amounts of defocusing: the focal length difference is approximately 0.2% when the camera is focused versus substantially defocused. The proposed camera calibration method could significantly simplify the calibration of large-range vision systems.

Funding. Directorate for Engineering (ENG) (CMMI-1521048)

REFERENCES

1. C. B. Duane, "Close-range camera calibration," Photogramm. Eng. **37**, 855–866 (1971).
2. I. Sobel, "On calibrating computer controlled cameras for perceiving 3-d scenes," Artif. Intell. **5**, 185–198 (1974).
3. R. Tsai, "A versatile camera calibration technique for high-accuracy 3d machine vision metrology using off-the-shelf tv cameras and lenses," IEEE J. Rob. Autom. **3**, 323–344 (1987).
4. Z. Zhang, "A flexible new technique for camera calibration," IEEE Trans. Pattern Anal. Mach. Intell. **22**, 1330–1334 (2000).
5. J. Lavest, M. Viala, and M. Dhome, "Do we really need an accurate calibration pattern to achieve a reliable camera calibration?" in *Computer Vision ECCV* (Springer, 1998), pp. 158–174.
6. A. Albarelli, E. Rodolà, and A. Torsello, "Robust camera calibration using inaccurate targets," IEEE Trans. Pattern Anal. Mach. Intell. **31**, 376–383 (2009).
7. K. H. Strobl and G. Hirzinger, "More accurate pinhole camera calibration with imperfect planar target," in *IEEE International Conference on Computer Vision Workshops (ICCV Workshops)* (IEEE, 2011), pp. 1068–1075.
8. L. Huang, Q. Zhang, and A. Asundi, "Flexible camera calibration using not-measured imperfect target," Appl. Opt. **52**, 6278–6286 (2013).
9. C. Schmalz, F. Forster, and E. Angelopoulou, "Camera calibration: active versus passive targets," Opt. Eng. **50**, 113601 (2011).
10. L. Huang, Q. Zhang, and A. Asundi, "Camera calibration with active phase target: improvement on feature detection and optimization," Opt. Lett. **38**, 1446–1448 (2013).
11. W. Li and Y. F. Li, "Single-camera panoramic stereo imaging system with a fisheye lens and a convex mirror," Opt. Express **19**, 5855–5867 (2011).

12. B. Li, N. Karpinsky, and S. Zhang, "Novel calibration method for structured-light system with an out-of-focus projector," Appl. Opt. **53**, 3415–3426 (2014).
13. S. Zhang, "Flexible 3d shape measurement using projector defocusing: extended measurement range," Opt. Lett. **35**, 931–933 (2010).
14. L. Yong, "A correspondence finding method based on space conversion in 3d shape measurement using fringe projection," Opt. Express **23**, 14188–14202 (2015).
15. L. Ekstrand and S. Zhang, "Three-dimensional profilometry with nearly focused binary phase-shifting algorithms," Opt. Lett. **36**, 4518–4520 (2011).