

Fast Projection Defocus Correction for Multiple Projection Surface Types

Zaixing He , Member, IEEE, Peilong Li , Xinyue Zhao , Shuyou Zhang , and Jianrong Tan 

Abstract—A major obstacle in digital projector technology is that images projected onto nonideal surfaces with large depth variances can easily become blurred. In this article, present a method to overcome projection defocus for projection surfaces that inevitably have complex shapes and large depth variances. The proposed method has two main advantages over traditional methods. First, an edge-intensification-based defocus compensation algorithm is proposed to manipulate the input image to compensate for its projection defocus before blurring occurs. Unlike previous time-consuming compensation algorithms, the proposed algorithm has very high efficiency, as it is noniterative and open loop. Second, a sinusoidal-projection-based estimation method is proposed to reduce kernel estimation errors on complex surface types. Unlike previous methods limited to specific surface types, the proposed method can provide consistently good kernel estimation results even for discontinuous and textured (nonpure white) projection surfaces. Hence, the proposed method can be applied to a wider range of applicable surfaces. These two contributions are demonstrated through extensive experiments and compared with the state-of-the-art methods.¹

Index Terms—Display enhancement, image composition, projector defocus.

I. INTRODUCTION

RECENT research studies into digital projection technology, such as shape acquisition [1]–[6], immersive virtual reality (VR) [7]–[9], [19], [21], [24], and appearance control [10]–[12], have greatly increased the consumer, commercial, and scientific applications of video projectors. For example, a user can shade historical artifacts [12] and the human face [19] with virtual contents by projection. Additionally, a user can visualize the deformation of a nonrigid surface by superimposing graphics directly onto the surface by projection [24]. In such applications, images are projected on nonplanar surfaces with large depth variances to enhance an immersive viewing experience.

Manuscript received December 23, 2019; revised May 13, 2020; accepted June 11, 2020. Date of publication June 17, 2020; date of current version February 22, 2021. This work was supported in part by the National Natural Science Foundation of China under Grants 51775497 and 51775498. Paper no. TII-19-5442. (Corresponding author: Xinyue Zhao.)

The authors are with the State Key Lab of Fluid Power & Mechatronic Systems, Zhejiang University, Hangzhou 310027, China (e-mail: zaixinghe@zju.edu.cn; lplzju18@zju.edu.cn; zhaoxinyue@zju.edu.cn; zsy@zju.edu.cn; egi@zju.edu.cn).

Color versions of one or more of the figures in this article are available online at <https://ieeexplore.ieee.org>.

Digital Object Identifier 10.1109/TII.2020.3003110

¹The MATALB code and experimental data for this article are available at <https://github.com/lpl-code/FastProjDefocusComp>

A major obstacle in these applications is that images projected onto surfaces with large depth variances can easily become blurred. For the best performance, all these applications desire the projectors to be focused on any part of the projection surface. However, in practice, almost all projectors have a significantly narrow depth-of-field (DOF) [13], [15], due to their optical design (that projectors are fabricated with large apertures to maximize their brightness). Only a single planar surface perpendicular to the projector's optical axis can produce a completely focused projection. Such an ideal surface is almost impossible to find in current VR/augmented reality(AR) applications.

To address the projector defocus issue, previous studies have developed image preconditioning techniques to sharpen projected results. These techniques attempt to manipulate the input images to compensate for their projection defocus before blurring occurs. The manipulated input image is also called the compensation image. To compute the compensation image, the first step is to estimate the defocus kernel, and the second step is to generate the compensation image by inputting the estimated defocus kernel into the defocus compensation algorithm. Concerning these two steps, previous preconditioning techniques have two major limitations.

- 1) *The computational efficiency is prohibitively low:* To compute the compensation images, the state-of-the-art methods [14], [15], [18] cast this as a constrained optimization problem. However, as this method requires an iterative and steepest descent algorithm, it consumes prohibitive computation time, especially for high-resolution images [15]. Therefore, the current defocus compensation technique is far from a real-time (frame-rate) blur compensation for moving images (films). Iwai *et al.* [15] hope to increase speed by graphics processing unit (GPU) parallel computing rather than by reducing the algorithm's complexity. Efficiency problem seems to be ignored in comparison with the benefits obtained from the good defocus correction results for a single static image.
- 2) *The kernel estimation results have large errors on complex surface types:* As an important input of a compensation algorithm, incorrect kernel estimation leads to compensation failure. However, previous methods can be applied to limited surface types and experience large estimation errors outside these types. For example, the method presented in [16] is for a tilted-planar surface, in [17], a method is presented for a dual-planar surface, in [18], a method is presented for a smooth curved surface, and in [23], a method is presented for a pure white

projection surface. Previous surface-type-dependent methods lead to large estimation errors on complex surfaces, such as on a discontinuous surface (e.g., v-groove) or on a nonpure white surface (e.g., textured). A universal kernel estimation method applied to a wide range of complex surface types is currently lacking.

In this article, we aim to address the above-mentioned two limitations of previous preconditioning techniques for reducing the projection defocus. *First*, we propose an edge-intensification-based method to boost the efficiency of computing compensation images. *Second*, we propose a surface-type-independent sinusoidal-projection-based kernel estimation method to reduce the kernel estimation error on complex surfaces.

The main contributions of this work are as follows.

- 1) A new edge-intensification-based defocus compensation algorithm is proposed to compute the compensation image for projection defocus correction. The algorithm has very high efficiency because it is noniterative and open loop, whereas the state-of-the-art compensation algorithms presented in [15] and [18] require time-consuming iterative optimization for the same task. Hence, the proposed method can solve the compensation image in less than 1/10 000 of the state-of-the-art method's time with a comparable compensation effect.
- 2) A new sinusoidal-projection-based kernel estimation method is presented to estimate defocus kernels required for the defocus compensation algorithm. Unlike previous methods that can be used only for a specific surface type, the proposed method provides consistently good kernel estimation results even at discontinuous corners and textured regions (such as black stains) on the projection surface. Hence, the proposed method can be applied to a wide range of complex projection surfaces, such as a discontinuous surface, for e.g., a v-groove, and even a nonpure white surface with black stains.

The remainder of this article is organized as follows. Section II presents the proposed edge-intensification-based method for generating the compensation image. Section III discusses the proposed sinusoidal-projection-based method for estimating the defocus kernel. Next, in Section IV, extensive experiments are conducted to demonstrate the two contributions of the proposed method. Efficiency is evaluated by the computation time. Five challenging surface types with complex shapes and reflectivity (e.g., texture) are used in our experiments. We experimentally and numerically show that the state-of-the-art methods presented in [18] and [23] may fail for some surface types, whereas the proposed methods can handle all of these surface types. Finally, Section V concludes this article.

II. EDGE-INTENSIFICATION-BASED DEFOCUS COMPENSATION ALGORITHM

Here, we give the details of the proposed edge-intensification-based method for generating the compensation image. We manipulate the input image to generate a compensation image for reducing its projection defocus. First, we analyze the defocus process with an optical model (see Section II-A). Second, we

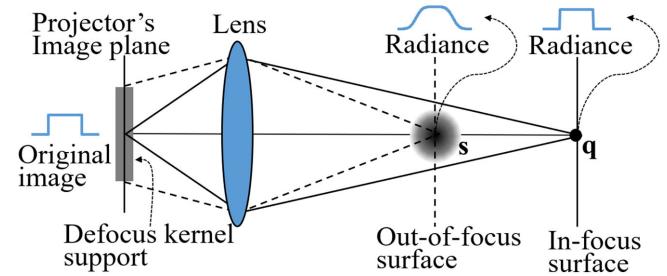


Fig. 1. Principle of projection defocus. Radiance projected at different distances from the projector exhibits different amounts of blur. For a point q that is in focus, rays of light emitted from a single projector pixel converge, creating a sharp radiance. For another point s , that is defocused, rays of light are distributed in a small area, creating a blurred radiance.

explain how the proposed algorithm works (see Section II-B). We then numerically evaluate the significant efficiency improvement of the proposed method, compared with the state-of-the-art iterative optimization algorithm (see Section II-C).

A. Projection Defocus Model

Consider a scene illuminated by a binary periodic pattern from a projector that is focused behind the scene (see Fig. 1). The binary periodic pattern contains only 0 and 1; hence, we simplify it as a 1-D square wave. For a point q that is in focus, light rays emitted from a single projector pixel converge, creating a sharp radiance. For another point s that is defocused, light rays are distributed in circle-of-confusion, creating a blurred radiance.

We use the modeling presented in [14]. The irradiance at point s is equal to the convolution of its defocus kernel point spread function (PSF) with the light rays from the projector. Assume that the surface is Lambertian, then the radiance I_o of s can be written as

$$I_o = \alpha f(\mathbf{x}) * P(\mathbf{x}) + \Gamma \quad (1)$$

where \mathbf{x} is shorthand for the projector pixel coordinate pixel (x_p, y_p) , $*$ denotes convolution, α depends on the surface albedo bidirectional reflectance distribution function (BRDF), $f(\mathbf{x})$ is defocus kernel, $P(\mathbf{x})$ is the original image input into the projector, and Γ is the ambient radiance.

We analyze the process of defocus blur. Generally, an original image can be divided into *edge* areas and *non-edge* areas. The *edge* areas are where abrupt intensity change occurs, whereas the *nonedge* areas have slight or no intensity change. When αf is applied to the original image (as a spatially weighted average), it weakens the intensity change in the edge areas. Hence, the abrupt edge areas become smooth and diffused, so they may be confused with the nonedge areas, and blurring occurs. In Fig. 1, as s is more defocused than q , its radiance profile is more blurred, and the sharp square edges are degraded into rounded.

B. Edge-Intensification-Based Algorithm

Now we describe the methodology of the proposed method. We compare the change between two images before and after

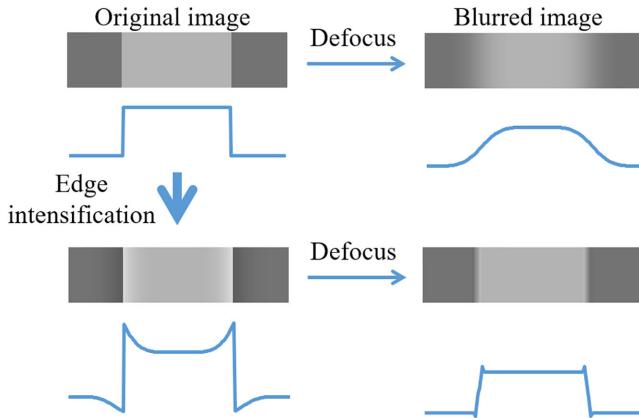


Fig. 2. The principle of the proposed edge-intensification-based algorithm in the presence of projection defocus. In the top row, after defocus, the 90° sharp edge diffuses to a smooth edge (>90°). Hence, in the second row, we sharpen the edge (<90°) and obtain a defocus result that is close to a 90° sharp edge.

projection defocus, to guide what should be preconditioned in the original image to combat the effects of projection defocus. Before defocus, in the original image (see Fig. 2, top left), the edge areas represent high-frequency components, and the nonedge areas represent low-frequency components. After defocus, in the blurred image (see Fig. 2, top right), as the convolution of kernel αf resembles a low-pass filter, the sharp edge areas become diffused and smooth to resemble the nonedge areas, and the projection looks blurred. Moreover, the sharp edge areas in the original image are much more vulnerable to defocus, whereas the nonedge areas are nearly invariant during the defocus process.

Based on this, the task to compensate for defocus is to intensify the edge areas (high-frequency components) in the original image. More concretely, after blurring without compensation, the 90° square edges (=90°) become obtuse (>90°). Accordingly, we preconditioned a concave acute-angled edge (<90°) in the original image. As a result, we obtain a near right-angled edge ($\approx 90^\circ$) after blurring, which is almost as sharp as the original image (the second row in Fig. 2).

To determine what areas should be intensified in the original image, we divide the original image P by its blurred image $I_o = \alpha f * P + \Gamma$ [see (1)] in pixelwise. By this means, the perceived edge spread function (ESF) can be represented as follows:

$$\text{ESF}(\mathbf{x}) = \begin{cases} \frac{P(\mathbf{x})}{\alpha f(\mathbf{x}) * P(\mathbf{x}) + \Gamma} & \text{if } \alpha f(\mathbf{x}) * P(\mathbf{x}) \neq 0 \\ 1 & \text{if } \alpha f(\mathbf{x}) * P(\mathbf{x}) = 0 \end{cases}. \quad (2)$$

Based on the defocus model in (1), the denominator, $\alpha f * P + \Gamma$, is the defocused version of P . Also, to avoid a denominator that equals 0, if $\alpha f * P + \Gamma = 0$, we define the ESF = 1, which will be explained in detail later. In practice, we obtain the kernel αf and Γ for a given scene by using a kernel estimation procedure, which will be discussed in Section III.

As the proposed method aims to compute the compensation image before blurring, the defocused version of P ($\alpha f * P + \Gamma$) is not captured by a camera after blurring. Instead, it is computationally generated applying the estimation results of αf ,

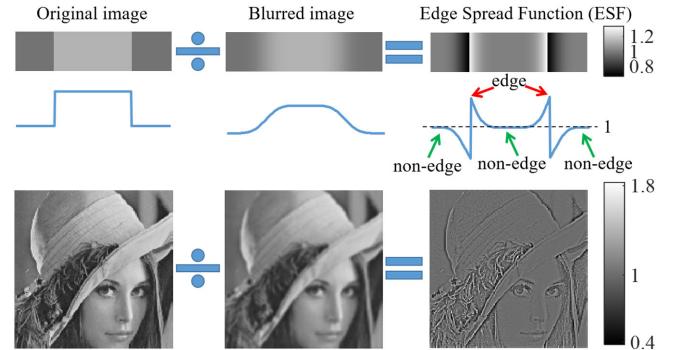


Fig. 3. Acquisition and behavior of the ESF. The ESF shows the position of the edge area in the original image. An abrupt jump in ESF means an edge area is present, whereas $\text{ESF} = 1$ indicates a nonedge area. The jump in ESF can be used to strengthen the original image.

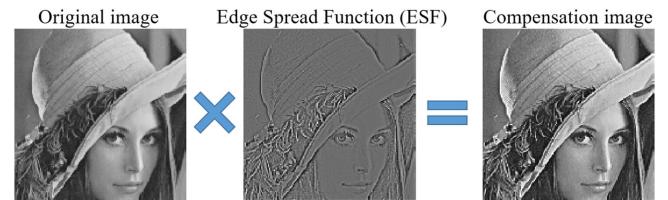


Fig. 4. Edge-intensification-based algorithm for the fast computation of the compensation image by multiplying the original image with the ESF.

Γ to P . Hence, the proposed method is open loop; it does not require after-blurring feedback but can compensate for blurring before it occurs. Besides, incorrect estimation results will result in an inability to generate correct compensation images. Details of the kernel estimation are given in Section III.

Next, we present some insights about the ESF behavior. We first take a 1-D square-wave edge model as an example (top of Fig. 3). If an area contains a sharp edge, the changes before and after blurring must be significant due to the convolution of kernel αf ; hence, the ESF will have an abrupt jump across 1, and we use this transition to highlight the high-frequency components to contour the defocus. Otherwise, if an area has gentle change (nonedge area) that is invariant to the convolution of kernel αf , the ESF will be 1, which means no compensation is needed. This is also true for the 2-D image. As shown in the second row in Fig. 3, the ESF result highlights the location of the edge areas that should be intensified. Hence, in Fig. 4, we simply multiply the original image by the ESF to compute the *compensation image*, P^* , using the following two steps:

$$P^* = \text{ESF} \cdot P, \quad (3)$$

$$P^* = \text{CLAMP}(P^*; 0, 255) \quad (4)$$

where \cdot is an elementwise multiplication, P^* is the compensation image for defocus compensation, and **CLAMP** is a pixelwise clamping operation. Because P^* computed by (3) may have a value greater than 255, CLAMP is to restrict P^* to a given range [0, 255], as images are stored in 8 bit for each of R, G, and B channel. If $P^*(\mathbf{x})$ obtained by (3) is less than 0 or larger than 255, then $P^*(\mathbf{x}) = 0$ or $P^*(\mathbf{x}) = 255$, respectively. Equations (3)

and (4) find the compensation image, P^* , with all its intensity values within the projector's dynamic range (0–255), which after defocus blurring more closely matches the original image P . If a projector has a wider dynamic range than 0–255, there is no need to perform the CLAMP instruction in (4).

According to (3), if an area has $\text{ESF} > 1$, it means this area should be intensified in compensation image P^* ; if $\text{ESF} = 1$, we need to do nothing; and if $\text{ESF} < 1$, we should dilute this area to highlight its *adjacent* edge areas (increase contrast). Hence, in (2), to avoid a denominator being 0, if $\alpha f * P + \Gamma = 0$ (generally due to the loss of estimation results of αf and Γ), we define $\text{ESF} = 1$, which means the underlying area remains unchanged before and after the proposed compensation algorithm.

As shown in (3) and (4), the compensation image can be efficiently computed by simply multiplying the original image by the ESF. Compared with previous time-consuming iterative optimization algorithms [14], [15], [18], the proposed method is simple and noniterative, requiring much less computation time. Further experiments in Section IV numerically prove the significant efficiency improvement (through computation time).

III. KERNEL ESTIMATION FOR MULTIPLE SURFACE TYPES USING A SINUSOIDAL-PROJECTION-BASED METHOD

In this section, we describe a framework to estimate the parameters of αf and Γ in (2), which are required for the defocus compensation algorithm. These parameters are implicit in the defocus response of the projection surface. Hence, three projection patterns were used to test the defocus response, including an all-black pattern, a dot pattern, and a sinusoidal pattern. The *all-black pattern* and *dot pattern* are used as a standardized input (impulse) to produce output defocus responses on the projection surface. These output responses are captured by the camera. To compare the output in the camera coordinate frame with the input in the projector's coordinate frame, the *sinusoidal patterns* are used to warp the captured output image from the camera to the projector's coordinate frame. Through comparison, the kernel estimation results are obtained. The details for kernel estimation include two steps, as discussed below. The data and code required for implementing the procedures discussed in this section are given in [20].

A. Testing the Defocus Response on the Projection Surface

To estimate Γ and αf , we use a projector to generate standardized input (dot or all-black) and a camera to observe the output defocus response on the projection surface.

To make it convenient to compare the captured output response with the input, we always transfer the image from the camera to the projector's coordinate frame. Here, for convenience of expression, we assume that the output and input have been aligned in the same projector (input) coordinate frame. The details for the transfer will be discussed later.

First, the ambient radiance Γ is estimated by turning off the projector and taking an image with the camera. The all-black pattern indicates an “off” projector, so that no projector pixels give off light, and only ambient radiance is captured by camera.

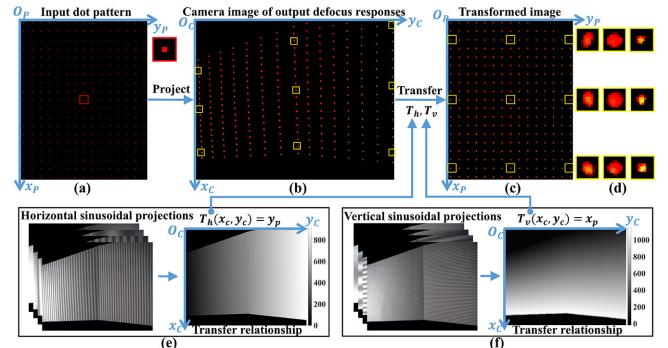


Fig. 5. Estimation of spatially varying defocus kernel of the red display channel. (a) Projected dot pattern (input). (b) Camera image. (c) Transformed image. (d) Transformed defocus responses. (d) and (e) Sinusoidal projections for computing the transfer relationship. A similar estimation of the ambient radiance Γ can be obtained by replacing the dot pattern with the all-black pattern.

Second, the defocus kernel αf is estimated by projecting a binary dot pattern [see Fig. 5(a)] onto the projection surface and then taking another image with the camera [see Fig. 5(b)]. When defocus occurs, the dot light rays emitting from a single projector pixel do not converge onto a single point but are instead distributed in a small *circle-of-confusion*. The *output* radiance profile (*circle-of-confusion*) captured by the camera [see Fig. 5(d)] is approximated as the defocus kernel αf for each *input* binary dot that is “1” in projection patterns, as presented in [15] and [18]. From these kernels, the kernels for other pixels that are “0” are estimated using bilinear interpolation. Besides, given that the intensity responses across the projection surface may be not uniform, we also normalize the output radiance for each *circle-of-confusion*. In our implementation, the distance between neighboring dots is 12 pixels. This method is effective for a projection surface with intricate geometry such as in Section IV.

B. Transferring the Output to the Input Coordinate Frame

As mentioned above, the transfer [see Fig. 5(b) and (c)] is vital for estimating αf and Γ . Inaccurate or incorrect transfer results will lead to missing kernel estimation results, which lead to the failure of the defocus compensation algorithm.

Previous methods to perform this transfer are applicable to only specific surface types and have large transfer errors for other surface types. A universal method applied to a wide range of complex surface types is lacking. For example, the method presented in [17] is designed for a dual-planar surface, but fails for a curved surface; the method presented in [18] is designed for a smooth curved surface, but fails for a discontinuous v-groove; and the method presented in [23] is designed for multiple surface shapes; however, it fails on a nonpure white surface with dark stains. These surface-type-dependent methods have large kernel estimation errors on complex surfaces, such as on a discontinuous v-groove or a nonpure white (textured) surface.

To minimize kernel estimation errors on complex surfaces and make the defocus compensation applicable to a wide range of

surface types, we employ sinusoidal projections to compute the transfer. The details are given below.

The coordinate information of each projector pixel (x_p, y_p) is encoded in the projected sinusoidal patterns. These patterns are projected onto the surface and captured by the camera. Assuming a linear projection process and given horizontal sinusoidal patterns (see Fig. 5), captured images can be written as

$$\begin{cases} I_{kn}(x_c, y_c) = A(x_c, y_c) + B_{kn}\cos(\phi(y_p) + \frac{2\pi n}{N}) \\ \phi(y_p) = 2\pi f_k y_p \end{cases} \quad (5)$$

where $A(x_c, y_c)$ is the albedo corresponding to a surface pixel (x_c, y_c) , B_{kn} is the intensity of the (k, n) th sinusoidal projection, f_k is the known frequency used to generate sinusoidal projections, and $\phi(y_p)$ is a phase. N is the number of sinusoidal projections for each frequency, and $n = 1, \dots, N$. We use $N = 4$ and project sinusoidal projections at 4 ($k = 1, 2, 3, 4$) different frequencies 1, 4, 16, and 64 in both horizontal and vertical directions. Sixteen sinusoidal projections are projected totally.

Assume that we have only a single frequency f_k , and let $N = 4$. The $\phi(y_p)$ can be recovered from captured images using the four-step phase-shifting algorithm [22] as

$$\phi(y_p) = -\arctan \frac{I_{k1}(x_c, y_c) - I_{k3}(x_c, y_c)}{I_{k4}(x_c, y_c) - I_{k2}(x_c, y_c)}. \quad (6)$$

The $\phi(y_p)$ recovered by (6) has a phase wrapping problem; hence, we repeat the recovery of $\phi(y_p)$ for four different frequencies f_1-f_4 and use all the results to disambiguate $\phi(y_p)$, based on the multifrequency algorithm [22]. In the end, substituting the disambiguated $\phi(y_p)$ into the second line of (5), we can recover the y_p for each surface pixel (x_c, y_c) as

$$T_h(x_c, y_c) = y_p, \text{ and } y_p = \frac{\phi(y_p)}{2\pi f_k} \quad (7)$$

where T_h indicates the transfer relationship from the camera's coordinate (x_c, y_c) to the projector's coordinate y_p . A similar equation $T_v(x_c, y_c) = x_p$ can be obtained for vertical sinusoidal projections by replacing y_p with x_p in (5)–(7).

Using T_h and T_v , we can transfer the output defocus response to the same projector coordinate frame as the input. Comparing the transferred output and the input, we can obtain the estimation result for defocus kernels that accounts for spatially varying defocus effects. We repeat the above-mentioned kernel estimation procedures for each of the three projector color channels.

C. Advantages of the Sinusoidal-Projection-Based Transfer

In summary, the sinusoidal-projection-based transfer has the following three advantages in practical applications.

First, the proposed transfer provides consistently good kernel estimation results for different surface shapes, whereas previous methods, such as the one presented in [18] is designed for a smooth curved surface, will fail on a discontinuous surface, such as a v-groove. This will be proven in Section IV (see Fig. 11 and Table III).

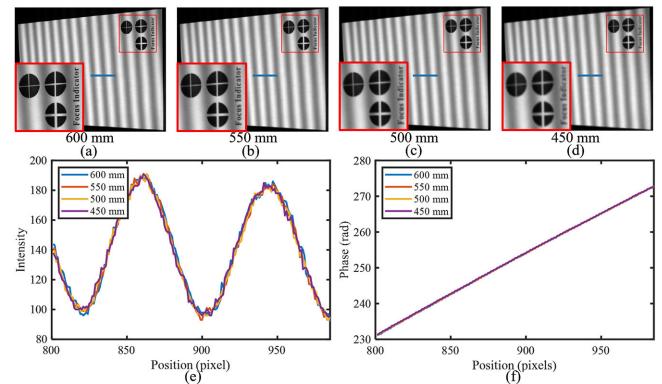


Fig. 6. Influence of defocus on projected sinusoidal projections. (a)–(d) Captured sinusoidal patterns with scene depths of 600–450 mm. (e) 1-D intensity profile and (f) the extracted phase value at the same selected location.

Second, the proposed transfer can be used for a challenging nonpure white surface, such as one with dark stains. Although the state-of-the-art method presented in [23] can also be used for arbitrary surface shapes in theory, it will fail on a nonpure white surface. This will be demonstrated in Section IV (Fig. 12 and Table III).

Here, we also give a theoretical explanation of how the sinusoidal-projection-based transfer can be applied to a nonpure white surface. As shown in (7), the transfer relationship depends on the phase $\phi(x_p)$. The phase $\phi(x_p)$ recovered by (6) uses four different sinusoidal images $I_{k1}-I_{k4}$ with the same frequency f_k . Among the four images $I_{k1}-I_{k4}$, the same nonuniform interference exists. These same nonuniform interferences can be canceled out in the subtraction operation in the numerator and denominator of (6). Hence, the nonpure white surface will not affect the proposed transfer algorithm.

Third, the sinusoidal projections are naturally insensitive to defocus. One may wonder whether the defocus of sinusoidal patterns will affect solving the transfer relationship. In fact, we find that the sinusoidal projections are nearly invariant to defocus. In Fig. 6, we experimentally and numerically proved this and studied the effect of different defocus levels.

We projected a sinusoidal pattern on a planar surface at four different depths (i.e., 600, 550, 500, and 450 mm) in Fig. 6. The focusing distance was fixed at 600 mm. Fig. 6(a)–(d) shows the captured images. From the ‘‘focus indicator’’ enlarged in the lower-left corner of Fig. 6(a)–(d), we can distinguish the extent of defocus. Fig. 6(c) shows the intensity profile of the radiance, and Fig. 6(d) shows the extracted phase value. From the results, we confirm that the sinusoidal projection is nearly invariant to the depths (defocus). Compared with intensity, phase $\phi(x_p)$ is more robust to defocus, which shows a smaller fluctuation (less than 0.8%) with depth variances. Hence, we use phase to recover the transfer relationship, as shown in (7).

The explanation of the third advantage is as follows. Unlike binary patterns having sharp edges, the sinusoidal projections belong to gentle nonedged. The sharp edge areas are much more prone to defocus than gentle nonedge areas (see Section II). Hence, defocus has less effect on sinusoidal projections.

IV. EXPERIMENTS AND RESULTS

Following the experimental sequence, the proposed defocus compensation method includes two main steps, as discussed in Sections II and III. The first step is kernel estimation. In this step, the main contribution is we use sinusoidal projections to achieve flexible kernel estimation on multiple surface types. The second step is to substitute the estimated kernels into the proposed defocus compensation algorithm. In this step, the main contribution is we significantly boost the efficiency of computing compensation images, compared with the previous defocus compensation algorithm. Experiments were carried out regarding the two main contributions of the proposed method.

A. Efficiency Evaluation of Computing Compensation Images

Here, we experimentally and numerically prove the efficiency improvement of compensation image computation. We compared the proposed edge intensification-based algorithm with the state-of-the-art iterative optimization algorithm used in [15] and [18]. In addition to the efficiency, we also present the image quality after compensation in both images and metrics to demonstrate that the proposed compensation algorithm can significantly improve efficiency while providing almost the same compensation quality.

To fairly compare the efficiency of different methods, we assume that the kernel estimation results substituted into different methods are known and the same. We used the 2-D spatial circular Gaussian filter to emulate the defocus kernel [18]. The Gaussian parameter was set as $\sigma = 1$. The original image is 150×150 pixels. When implementing the iterative optimization algorithm to reduce the memory footprint and avoid a local optimization, we divide the original image into 9 nonoverlapping blocks, each of which is 50×50 pixels. The algorithm is applied in turn to these blocks. The iteration number is set to 10 to ensure an acceptable time consumption.

The efficiency is evaluated by computation time on MATLAB R2018b with a laptop (CPU: Intel Core i7-8750H 2.21 GHz, RAM: 16 GB). To optimize efficiency of iterative optimization algorithm, we implemented the MATLAB optimization toolbox. To facilitate future comparisons, we publish the code for both the two methods in [20].

The efficiency comparison of different compensation algorithms is shown in Table I. Three original images, Lena, Baby, and House, were used to test the different algorithms. The algorithms are all implemented in unoptimized MATLAB code. On average, the computation time declines dramatically from 565.3989 to 0.0345 s, compared with the iterative optimization algorithm. In other words, the proposed compensation algorithm takes only 1/16 000 of the time of the previous iterative optimization algorithm, with our current unoptimized code. Hence, the proposed method dramatically boosts the efficiency of computing the compensation image.

We also found a study [15] that reported the computation time of the previous iterative compensation algorithm. In their implementation, for a static 300×300 pixels original image, the previous optimization algorithm requires approximately 50

TABLE I
BLUR COMPENSATION METHOD EVALUATION RESULTS

Image	Metric	Blurred Original Image	Iterative Optimization Algorithm [15], [18]	Proposed
Lena	Time	-	581.6766 s	0.0351 s
	SSIM	0.6442	0.7584	0.7388
	PSNR	24.1245	25.8198	26.1590
	RMSE	13.8175	11.3674	10.9322
Baby	Time	-	567.1190 s	0.0343 s
	SSIM	0.7353	0.8140	0.8133
	PSNR	27.6941	29.1437	30.7395
	RMSE	10.5157	8.8993	7.4057
House	Time	-	547.4010 s	0.0342 s
	SSIM	0.6510	0.7696	0.7552
	PSNR	20.7986	22.8663	22.9188
	RMSE	23.2534	18.3273	18.2169

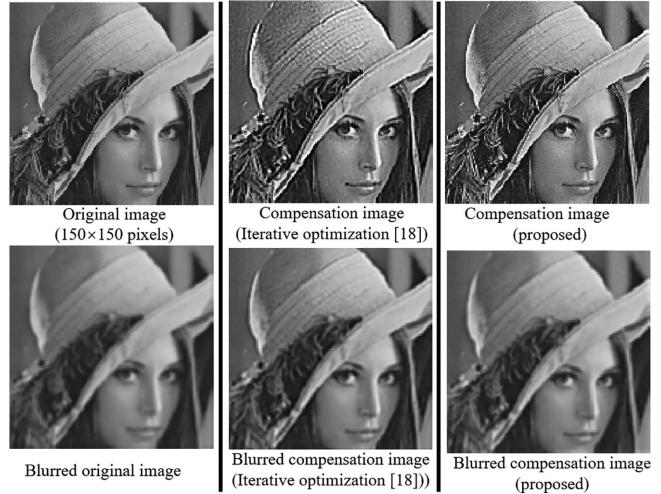


Fig. 7. Comparison of different defocus compensation algorithms. The proposed method gives good image quality, nearly equivalent to the state-of-the-art iterative optimization algorithm.

min, which is similar to our implementation of the iterative compensation algorithm. In contrast, the proposed method takes only 0.0379 s on average in our implementation for the same task. Hence, the previous iterative optimization algorithm has a prohibitive time cost and is far from a real-time defocus compensation method.

Fig. 7 shows one of the compensation results for the original image “Lena.” Visually, the compensation images computed by the two algorithms are almost the same, and both appear to have stronger edges like high-pass filtered versions of the original image. This is expected, since the compensation algorithms boost high-frequency components to compensate for the defocus. The second row shows the blurred results of the first row by applying the known αf to the compensation images. Using the proposed algorithm, the blurred results are less blurry and more closely resemble the original image.

Table I numerically shows the compensation quality, which is evaluated by comparing the blurred compensation image with the original image. Three metrics were used to describe the compensation quality: structural similarity (SSIM), peak signal-to-noise ratio (PSNR), and root-mean-square error (RMSE). Larger SSIM and PSNR results and smaller RMSE results indicate better compensation algorithm results. From the SSIM results,

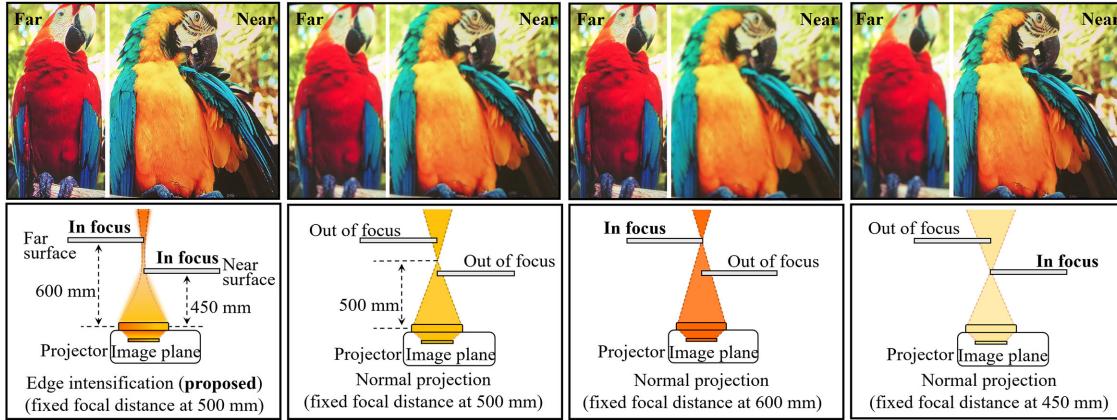


Fig. 8. Comparison of the proposed method and normal projection on a dual-planar surface placed at depths of 450 and 600 mm from the projector.

the proposed method provides a 13.8% improvement in image quality on average, which is close but slightly inferior to the iterative optimization algorithm. The PSNR and RMSE results show that the proposed algorithm is better than the iterative optimization algorithm. This difference is due to the different metric emphases. In general, the proposed algorithm provides comparable defocus compensation quality to the iterative optimization algorithm, with 1/16 000 of the time consumption.

B. Adaptability Evaluation for Multiple Surface Types

Here, we demonstrate the proposed method's flexible application to multiple projection surfaces types. The surface type includes *surface shape* and *reflectivity* (e.g., *texture*). Five typical surface types, including two surfaces with different depths (see Fig. 8), a tilted surface (see Fig. 9), a smooth curved surface (see Fig. 10), a challenging v-groove surface (see Fig. 11), and a more challenging textured (nonpure white) surface (see Fig. 12), were used in our experiments. We also compared the proposed method with two state-of-the-art methods in [18] and [23]. The method presented in [18] fails on the discontinuous v-groove surface. The method presented in [23] fails on the textured surface. However, the proposed method can handle all these complex surface types.

A camera-projector system was built to implement the proposed defocus compensation method. Our system consists of a projector (TI DLP4500), an industrial camera (Flea FL3-U3-13E4C), and a laptop (CPU: Intel Core i7-8750H 2.21 GHz, RAM: 16 GB). To minimize the camera hardware's errors, a high-precision industrial camera was used for kernel estimation. We used another mobile camera (an iPhone 7) as a proxy for the human eye to observe the projected images. All hardware used in our experiments is off-the-shelf. The data and MATLAB code are given in [20]. As the kernel estimation method is the reason for limiting the adaptability to different surface types, for a fair comparison with methods presented in [18] and [23], all methods use the same algorithm to generate compensation images, but the kernel estimation methods are different.

The first experiment was conducted on a dual-planar surface placed at different depths from the projector (see Fig. 8). Two

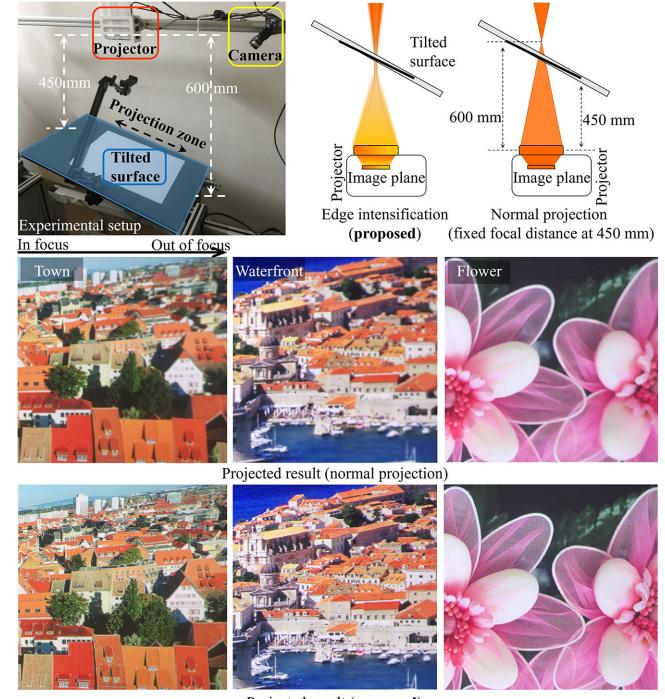


Fig. 9. Demonstration of the proposed projection defocus correction method with a tilted surface.

parrot images were projected onto the far-surface (600 mm from the projector) and the near-surface (450 mm from the projector). Due to the narrow DOF of the projector, normal projection can focus only on either the near or far surface (focused at 450 mm and 600 mm), or were out-of-focus on both surfaces (focused at 500 mm). In contrast, only the proposed method provided a focused image on both planar surfaces simultaneously. This experiment clearly illustrates that the proposed method can effectively extend the projector's DOF.

In the tilted surface experiment, the projection surface was placed so that the left and right edge of the projection zone was 600 mm and 450 mm from the projector, respectively (see Fig. 9). We used three natural images as original images.

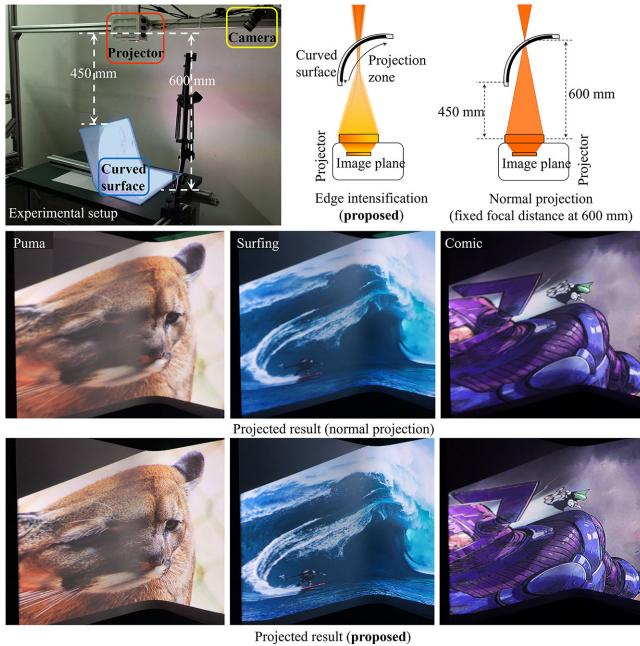


Fig. 10. Demonstration of the proposed projection defocus correction method with a curved surface.

The projector was focused on the left edge with a fixed focal distance of 450 mm. Hence, in normal projection, the right side of the normal projected results are more blurred than the left side. In contrast, the proposed method produced a nearly completely focused result on both the left and right sides.

In the experiment conducted on a smooth curved surface, the projection surface has large depth variances from 450 to 600 mm. Three images with rich edge details were used as original images. The projector was focused at the right edge of the projection zone with a fixed focal distance of 600 mm. Hence, in normal projection, the left side of the projected results are out-of-focus and blurred. In contrast, the proposed method provides more in-focus projected results, as the puma's beard, the waves, and the textures on the comic characters are clearer.

A challenging discontinuous surface, the v-groove surface, was used to compare the proposed method with the normal projection and the state-of-the-art method presented in [18], as shown in Fig. 11. Method presented in [18] uses a quadric transfer to realize the kernel estimation on a smooth curved surface. When the surface has complex discontinuities such as the v-groove, the kernel estimation using [18] fails at the discontinuous concave edges of the v-groove. To more clearly show the failure of the method presented in [18], we selected and enlarged three regions A, B, and C at the discontinuous edges (see Fig. 11, bottom left). In these three groups of enlarged images, the method presented in [18] has basically no compensation effect, and its projection results are almost the same as normal projection. In contrast, the proposed method shows good defocus compensation results for these discontinuous regions.

In Fig. 11, we also selected and enlarged a region D that is not in the discontinuous region. Although the method presented in [18] improves the projection quality compared with normal

TABLE II
COMPARISON OF ADAPTABILITY TO DIFFERENT SURFACE TYPES

	Surface type	Normal projection	Ref. [18]	Ref. [23]	Proposed
White surface	Tilted planar (Fig. 9)	×	○	○	○
	Continuous curved (Fig. 10)	×	○	○	○
	Discontinuous (Fig. 11)	×	×	○	○
White surface with black stains (Textured surface, Fig. 12)		×	×	×	○

Note. ○ means applicable, and × means not applicable.

projection, the proposed method gives a better compensation effect than the method in [18]. This is because the method presented in [18] requires nine point pairs to estimate the quadric transfer. For complex surface (v-groove), it is difficult to select right point pairs. Hence, in region D, incorrect kernel estimation makes the method presented in [18] has a poor defocus compensation effect.

The above-mentioned experiments prove the adaptability of the proposed method on multiple *surface shapes*. In addition, a more challenging experiment was conducted on a *textured (nonpure white)* curved surface. The binary-structured-based method presented in [23] can also be used for the kernel estimation on multiple surface shapes, like the proposed method. However, this method [23] fails on the textured curved surface (see Fig. 12). This is because the textured surface can interfere with the decoding of binary projections. The transfers solved by the method presented in [23] and the proposed method are shown in the upper-right corner of Fig. 12. Obviously, the proposed method obtains a more accurate transfer relationship on the textured surface, for reasons explained in Section II-C. However, the binary projections [23] obtain the wrong transfer results (with large errors) in textured places that have black stripes. The kernel estimation results in these places will be lost, leading to invalid defocus compensation in these places. The projected results are shown in the middle of Fig. 12. To clearly show the failure of method [23], we selected and enlarged four regions, A, B, C, and D, where the projection surface has black stripes. In these areas, the method presented in [23] has no compensation effect. In contrast, the proposed method still shows good defocus compensation effects.

A comparison of the adaptability to different surface types of different methods is given in Table II. The proposed method has the highest adaptability to the projection surface types.

C. Defocus Compensation Quality Evaluation

To evaluate the defocus compensation quality, we first borrowed four metrics used in the previous works to achieve a comprehensive evaluation. Then, we conducted a subjective evaluation and proposed a binary-decoding-based evaluation method to further gauge the performance of the proposed method. Corresponding code and data were provided in [20].

The four metrics include SSIM [15], RMSE [18], PSNR [7], and the number of edge pixels detected in the images captured by the camera (Edge numbers) [17]. To compute the first three metrics, SSIM, RMSE, and PSNR, we should have compared

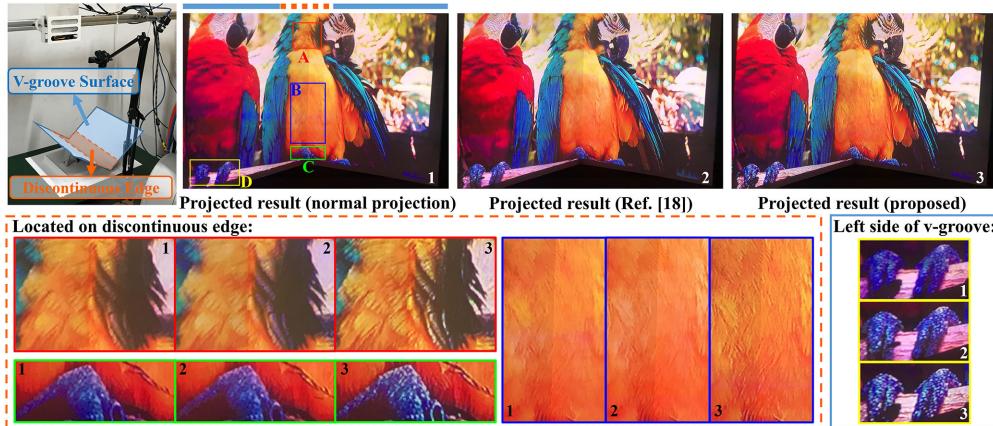


Fig. 11. Comparison of the proposed method with the normal projection and the method presented in [18] on a discontinuous v-groove surface.

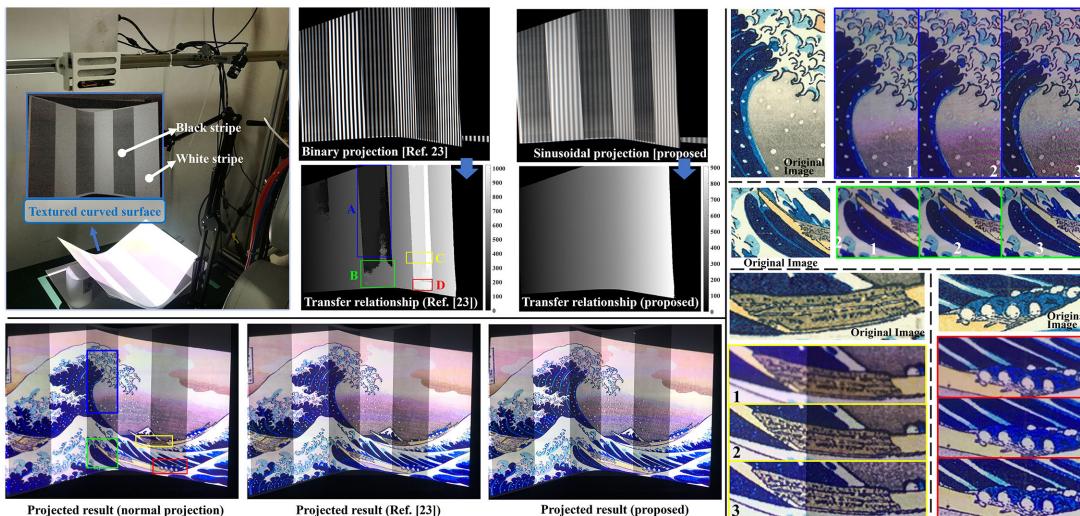


Fig. 12. Comparison of the proposed method with the normal projection and the method presented in [23] on a textured (nonpure white) curved surface.

the defocused image with the input original image. However, deformed by the shape of the projection surface, the defocused image captured by the camera is distorted and cannot be compared. Hence, as presented in [13] and [18], we synthesized the defocused image by applying the kernel estimation results to the input image. To compute the fourth metrics, Edge numbers, Harris detector was used. This metric is inspired by the defocus principle that high-frequency components are mainly lost in the defocus process. Therefore, a better defocus compensation result should be sharper, retaining more detected edge pixels. The results of the four metrics are listed in Table III. We preliminarily observed that the proposed method provides better defocus compensation quality than normal projection and the two methods presented in [18] and [23].

Although the above-mentioned four metrics are widely used in the literature, they may be affected by camera errors. Hence, another two methods that are free from camera errors were proposed to further evaluate the proposed method.

First, we conducted a subjective evaluation of the proposed method. The subjective evaluation method is widely used in the

TABLE III
COMPARISON OF COMPENSATION QUALITY USING FOUR METRICS

Scene	Metric	Normal projection	Ref. [18]	Proposed
V-groove (Fig. 11)	SSIM	0.9560	0.9555	0.9569
	PSNR	29.5525	29.9091	29.9612
	RMSE	8.4902	8.1487	8.0999
	Edge numbers	4058	4458	6083
Scene	Metric	Normal projection	Ref. [23]	Proposed
Textured (Fig. 12)	SSIM	0.8719	0.8850	0.8993
	PSNR	23.1102	23.5654	24.0151
	RMSE	17.6853	16.7832	15.9355
	Edge numbers	4117	12722	15546

TABLE IV
PROFILES OF THE VOLUNTEERS

Number of volunteers (male/female)	20 (18/2)
Nationality (number)	Pakistan (1), India (2), China (17)
Ages	22–48

evaluation of audio quality, image quality, or video quality [27], [28]. We recruited 20 volunteers to evaluate the quality of the projections with naked eyes. Table IV shows the profiles of the

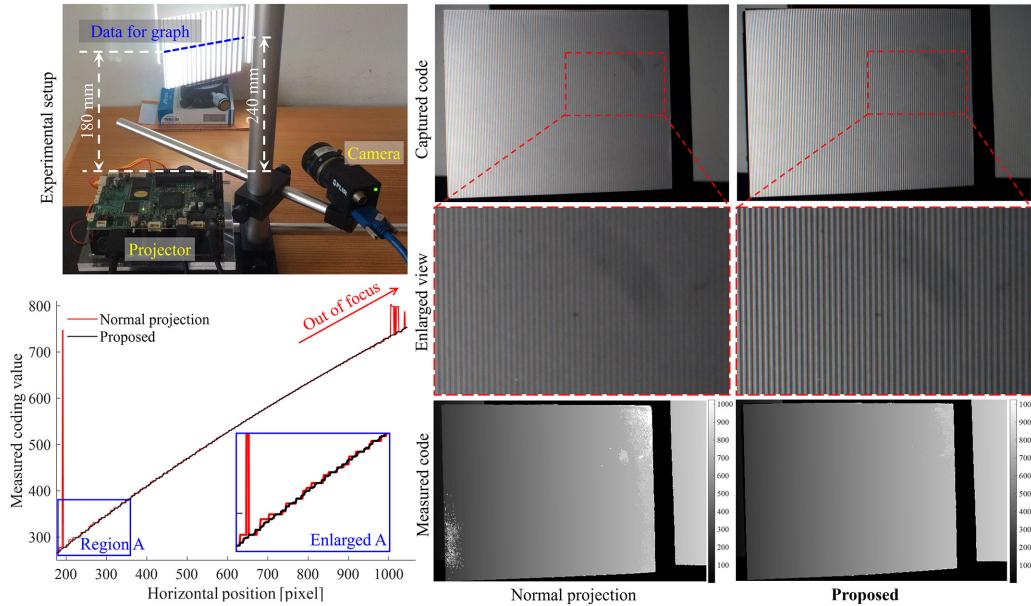


Fig. 13. Experiment of the binary-decoding-based evaluation method on a tilted surface. Top-left: experimental setup. Bottom-left: profile comparison of the measured coding values extracted from the two projections (see the blue dashed line in experimental setup). Top-right and middle-right: comparison of captured raw images used to capture the finest gray-code pattern (two pixels wide); for normal projection, the right side of the captured images was not clear, whereas for the proposed method, fringes were clear and sharp. Bottom-right: extracted coding maps; for normal projection, approximately a quarter of codes were not measured correctly; by contrast, codes were successfully measured by the proposed method.

volunteers. In the evaluation, two image groups were projected on the v-groove surface (see Fig. 11), and on the textured surface (see Fig. 12), respectively. Each group in a random and blind order included the original image, the compensation image of the method presented in [18] or [25], and the compensation image of the proposed method. The steps of the subjective evaluation were as follows.

- 1) Volunteers watched the original image (projector input) attached to each group as a reference for the subsequent evaluation of the projection quality.
 - 2) Volunteers watched the projection results of each image group on the projection surface. The three included images in each group were projected in a random and blind order.
 - 3) Volunteers rated the projection quality of the three projection results in each image group. The projection quality depends on how well the projection results match the original image. Specifically, the rating is on a scale from low-, middle- to high-grade. High grade indicates the best quality.
 - 4) Procedures 1–3 were repeated for the two image groups.
- In the subjective evaluation, we counted the number of votes as an evaluation metric. Table V shows the voting results in three grades. From the results, the proposed method has the highest projection quality, compared with the normal projection and the two state-of-the-art methods presented in [18] and [23].

Second, we launched a binary-decoding-based evaluation method and analyzed why it is not disturbed by camera Errors.

The binary decoding is vital for structured-light three-dimensional measurement. In the structured-light measurement,

TABLE V
RATING RESULTS OF SUBJECTIVE EVALUATION

Scene	Rating	Normal projection	Ref. [18]	Proposed
V-groove (Fig. 11)	High	0	0	20
	Middle	0	20	0
	Low	20	0	0
Scene	Rating	Normal projection	Ref. [23]	Proposed
Textured (Fig. 12)	High	0	0	20
	Middle	1	19	0
	Low	19	1	0

encoded structured-light patterns are projected onto the object, and its reflections are captured by the camera. Extracted from the captured images, decoding information can be obtained, which are then used in shape computation through triangulation [1]. Structured-light-based measurement suffers from the projector defocus [15]. Once a part of the object surface is not within the projector's DOF, the projected structured-light patterns become blurred. Consequently, decoding results cannot be acquired accurately. In other words, the lower the extent of defocus, the more accurate the decoding values. Hence, the defocus compensation quality can be evaluated by the decoding results.

A binary-decoding-based evaluation experiment is shown in Fig. 13. Ten binary gray-code patterns were sequentially projected, which separated the surface into $2^{10} = 1024$ subregions in the horizontal direction.

We measured the coding value on the tilted surface by normal projection and by the proposed method. The projector was focused on the left side (180 mm from the projector). Consequently, defocus occurs on the right side. A profile of the measured coding values (see the blue dashed line at the top left of Fig. 13) is shown at the bottom left of Fig. 13.

TABLE VI
CAMERA ERROR LEVEL OF THE RAW CAPTURED IMAGE

Image No.	1	2	3	4	5
Noise Level	5.7566×10^{-4}	4.5137×10^{-4}	4.6773×10^{-4}	4.9054×10^{-4}	5.5982×10^{-4}
Image No.	6	7	8	9	10
Noise Level	6.4467×10^{-4}	8.2077×10^{-4}	1.0320×10^{-3}	1.0669×10^{-3}	5.5521×10^{-4}

From the decoding results, we observed that decoding errors were smaller with the proposed method than normal projection. To be specific, decoding results became noisy where the projected images were blurred by using normal projection. To numerically evaluate the error due to defocus, a line was fitted to the measured profile using a least square regression, and we calculated that the mean difference was 2.4065 in normal projection and 0.0125 in the proposed method. In summary, we confirmed that the proposed method provides better decoding results with fewer errors and less defocus than normal projections.

Next, we analyzed that the binary decoding results are free from camera errors, based on the Poissonian–Gaussian noise modeling for image raw-data in [25] and [26]. In this model, the camera noise e is formulated as follows:

$$e(X) = \pm 255 \cdot \sigma \cdot \xi, \text{ where } \xi \sim N(0, 1). \quad (8)$$

In (8), e is the camera noise of the captured raw image X . σ is a standard deviation. $\xi \sim N(0, 1)$ means ξ is zero-mean independent random noise with a standard deviation equal to 1 (normal distribution). Utilizing the code provided in [26], the σ values of all the ten raw images captured in the previous binary-decoding-based evaluation (see Fig. 13) are listed in Table VI. The maximum σ estimated was 1.0669×10^{-3} . According to (8) and the three-sigma rule, the camera noises in our experiments less than $\pm 255 \times 3\sigma \approx \pm 0.8162$ accounted for a probability of 99.73%. In contrast, in captured raw images, the differences in gray values to distinguish the binary codeword 0 and 1 were greater than 30. As 30 is much greater than 0.8162, camera errors will not affect the binarization and decoding from captured images. Thus, the binary-decoding-based evaluation method is free from camera errors.

V. CONCLUSION

In this article, we presented an effective method to overcome projection defocus. The proposed method has two major advantages over traditional methods.

First, the efficiency of computing compensation images is significantly improved by using an edge intensification-based defocus compensation algorithm. In our implementation, the proposed algorithm is over 16 000 times faster than the iterative optimization algorithm [18], with our current unoptimized code, yet still delivers a comparable compensation effect.

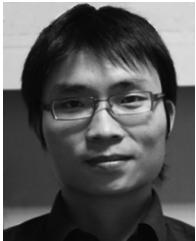
Second, the adaptability of defocus compensation is broadened by using a sinusoidal-projection-based kernel estimation method. The kernel estimation results are vital for guiding the defocus compensation algorithm. The proposed method can provide consistently good kernel estimation results, even on complex surfaces, whereas previous kernel estimation methods may fail on complex surfaces, leading to defocus compensation

failure. Five challenging surface types were used to demonstrate this. The proposed method can handle all these surface types. However, the latest methods presented in [18] and [23] fail on the discontinuous v-groove surface and the textured (nonpure white) surface, respectively.

REFERENCES

- [1] X. Feng and L. Gao, "Robust structured-light depth mapping via recursive decomposition of binary codes," *Opt. Eng.*, vol. 58, no. 6, Jun. 2019, Art. no. 060501.
- [2] T. Kreis, "Application of digital holography for nondestructive testing and metrology: A review," *IEEE Trans. Ind. Informat.*, vol. 12, no. 1, pp. 240–247, Feb. 2016.
- [3] H. Rashidizad, M. Sheikhi, and G. Akbarizadeh, "Efficient point cloud lossless data compression method based on an embedded gray code structured light pattern sequence," *Appl. Opt.*, vol. 57, no. 29, pp. 8766–8774, Oct. 2018.
- [4] B. Tayebi, F. Sharif, A. Karimi and J. Han, "Stable extended imaging area sensing without mechanical movement based on spatial frequency multiplexing," *IEEE Trans. Ind. Informat.*, vol. 65, no. 10, pp. 8195–8203, Oct. 2018.
- [5] R. Kelner and J. Rosen, "Methods of single-channel digital holography for three-dimensional imaging," *IEEE Trans. Ind. Informat.*, vol. 12, no. 1, pp. 220–230, Feb. 2016.
- [6] J. Lei, J. Sun, Z. Pan, S. Kwong, J. Duan, and C. Hou, "Fast mode decision using inter-view and inter-component correlations for multiview depth video coding," *IEEE Trans. Ind. Informat.*, vol. 11, no. 4, pp. 978–986, Aug. 2015.
- [7] R. Konrad, N. Padmanaban, K. Molner, E. Cooper, and G. Wetzstei, "Accommodation-invariant computational near-eye displays," *ACM Trans. Graph.*, vol. 36, no. 4, pp. 1–12, Jul. 2017.
- [8] K. Akşit, W. Lopes, J. Kim, P. Shirley, and D. Luebke, "Near-eye varifocal augmented reality display using see-through screens," *ACM Trans. Graph.*, vol. 36, no. 6, Nov. 2017, Art. no. 189.
- [9] K. Rathinavel, H. Wang, A. Blate, and H. Fuchs, "An extended depth-at-field volumetric near-eye augmented reality display," *IEEE Trans. Vision Comput. Graph.*, vol. 24, no. 11, pp. 2857–2866, Nov. 2018.
- [10] J. Ye, Y. Ji, M. Zhou, S. B. Kang, and J. Yu, "Content aware image pre-compensation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 41, no. 7, pp. 1545–1558, Jul. 2019.
- [11] H. Asayama, D. Iwai, and K. Sato, "Fabricating diminishable visual markers for geometric registration in projection mapping," *IEEE Trans. Vision Comput. Graph.*, vol. 24, no. 2, pp. 1091–1102, Feb. 2018.
- [12] Y. Y. Lee, J. H. Lee, B. Ahmed, M. G. Son, and K. H. Lee, "A new projection-based exhibition system for a museum," *ACM J. Comput. Cultural Heritage*, vol. 12, no. 2, May 2019, Art. no. 10.
- [13] M. S. Brown, P. Song, and T. J. Cham, "Image pre-conditioning for out-of-focus projector blur," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2006, pp. 1956–1963.
- [14] L. Zhang and S. Nayar, "Projection defocus analysis for scene capture and image display," *ACM Trans. Graph.*, vol. 25, no. 3, pp. 907–915, Aug. 2006.
- [15] D. Iwai, S. Miura and K. Sato, "Extended depth-of-field projector by fast focal sweep projection," *IEEE Trans. Vision Comput. Graph.*, vol. 21, no. 4, pp. 462–470, Apr. 2015.
- [16] Y. Oyamada and H. Saito, "Defocus blur correcting projector-camera system," in *Proc. Int. Conf. Adv. Concepts Intell. Vis. Syst.*, Juan-les-Pins, France, 2008, pp. 453–464.
- [17] S. Ladha, K. Smith-Miles, and S. Chandran, "Projection defocus correction using adaptive kernel sampling and geometric correction in dual-planar environments," in *Proc. IEEE Conf. Comput. Vision Pattern Recognit.*, Colorado Springs, CO, USA, 2011, pp. 9–14.
- [18] J. Park and B. Lee, "Defocus and geometric distortion correction for projected images on a curved surface," *Appl. Opt.*, vol. 55, no. 4, pp. 896–902, Feb. 2016.
- [19] A. Grundhöfer and D. Iwai, "Recent advances in projection mapping algorithm, hardware and applications," *Comput. Graph. Forum*, vol. 37, no. 2, pp. 653–675, 2018.
- [20] Project webpage, 2020. [Online]. Available: <https://github.com/lpl-code/FastProjDefocusComp>
- [21] M. Grosse, G. Wetzstein, A. Grundhoefer, and O. Bimber, "Coded aperture projection," *ACM Trans. Graph.*, vol. 29, no. 3, pp. 1–12, Jun. 2010.

- [22] C. Zuo, L. Huang, M. Zhang, Q. Chen, and A. Asundi, "Temporal phase unwrapping algorithms for fringe projection profilometry: A comparative review," *Opt. Lasers Eng.*, vol. 85, pp. 84–103, Oct. 2016.
- [23] F. Li *et al.*, "Simultaneous projector-camera self-calibration for three-dimensional reconstruction and projection mapping," *IEEE Trans. Comput. Imag.*, vol. 3, no. 1, pp. 74–83, Mar. 2017.
- [24] P. Punpongsanon and D. Iwai, "Projection-based visualization of tangential deformation of nonrigid surface by deformation estimation using infrared texture," *Virtual Reality*, vol. 19, no. 1, pp. 45–56, Jan. 2015.
- [25] A. Foi, M. Trimeche, V. Katkovnik, and K. Egiazarian, "Practical Poissonian-Gaussian noise modeling and fitting for single-image raw-data," *IEEE Trans. Image Process.*, vol. 17, no. 10, pp. 1737–1754, Oct. 2008.
- [26] W. Zhao, Q. Liu, Y. Lv, and B. Qin, "Texture variation adaptive image denoising with nonlocal PCA," *IEEE Trans. Image Process.*, vol. 28, no. 11, pp. 5537–5551, Nov. 2019.
- [27] H. Ohkushi, T. Ogawa, and M. Haseyama, "Music recommendation according to human motion based on kernel CCA-based relationship," *EURASIP J. Adv. Signal Process.*, vol. 2011, May 2011, Art. no. 121.
- [28] Y. Li *et al.*, "No-reference video quality assessment with 3D shearlet transform and convolutional neural networks," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 26, no. 6, pp. 1044–1057, Jun. 2016.



Zaixing He (Member, IEEE) received the B.Sc. and M.Sc. degrees in mechanical engineering from Zhejiang University, Hangzhou, China, in 2006 and 2008, respectively, and the Ph.D. degree in information science from the Graduate School of Information Science and Technology, Hokkaido University, Hangzhou, Japan, in 2012.

He is currently an Associate Professor with the Department of Mechanical Engineering, Zhejiang University. His current research interests include robotic vision, visual intelligence of manufacturing equipment, and image understanding.



Peilong Li received the B.S. degree in mechanical engineering from Huazhong Agricultural University, Wuhan, China, in 2018. He is currently working toward the M.S. degree in mechanical engineering at Zhejiang University, Hangzhou, China.

His current research interests include structured light-based sensing and machine vision.



Xinyue Zhao received the M.S. degree in mechanical engineering from Zhejiang University, Hangzhou, China, in 2008, and the Ph.D. degree in information science from the Graduate School of Information Science and Technology, Hokkaido University, Japan, in 2012.

She is currently an Associate Professor with the Department of Mechanical Engineering, Zhejiang University. Her current research interests include machine vision and image processing.



Shuyou Zhang received the M.S. degree in mechanical engineering and the Ph.D. degree in computer science from the State Key Lab of CAD&CG, Zhejiang University, China, in 1991 and 1999, respectively.

He is currently a Professor with the Department of Mechanical Engineering, Zhejiang University. He is also the Administrator of the Institute of Design Engineering, Zhejiang University, the Assistant Director of Computer Graphics Professional Committee for China Engineering Graphic Society, a Member of Product Digital Design Professional Committee, and the Chairman of Zhejiang Engineering Graphic Society. His current research interests include product digital design, design and stimulation for complex equipment, and engineering and computer graphics.



Jianrong Tan received the B.S. and M.S. degrees in mechanical engineer and electronic engineering from the Open University of China, Beijing, China, in 1982, the M.S. degree in engineering from the Huazhong University of Science and Technology, Wuhan, China, in 1985, and the Ph.D. degree in mathematics from Zhejiang University, Hangzhou, China, in 1987.

He is currently a Professor with the CAD&CG State Key Laboratory and School of Mechanical Engineering, Zhejiang University. He is also an Academician of the Chinese Academy of Science. Using his 15 years of working experience in manufacturing and theory in science, he proposed the technique in combination of batch and customization, which is used for multitudinous customization, status in engineering transition, fuzzy status, technology simulation of random status modeling and digitalized prototype integration, combination of figures and geometry in multicomponent correlation in intricate equipment, analysis of multi-level disposing, and multiparameter matching. He has 12 copyrights in computer software, which achieved a lot in manufacturing enterprises. He has authored eight pieces of monograph and 142 articles, which has been cited 1600 times in the abovementioned research areas. His current research interests include mechanical designing and theory and digitalized designing and manufactory.

Prof. Tan was the recipient of the National Pride Award four times, which includes Second Prize in National Technology Progress twice and the First Prize in National Excellent Education Achievement. He was also the recipient of the First Prize in Provincial Level Technology Progress award six times.