Opt Rev (2015) 22:762–778
DOI 10.1007/s10043-015-0123-4

REGULAR PAPER

Multi-projector auto-calibration and placement optimization for non-planar surfaces

Dong Li[1] · Jinghui Xie[1] · Lu Zhao[1] · Lijing Zhou[1] · Dongdong Weng[1]

Received: 19 December 2014 / Accepted: 22 June 2015 / Published online: 21 July 2015

Abstract Non-planar projection has been widely applied in virtual reality and digital entertainment and exhibitions because of its flexible layout and immersive display effects. Compared with planar projection, a non-planar projection is more difficult to achieve because projector calibration and image distortion correction are difficult processes. This paper uses a cylindrical screen as an example to present a new method for automatically calibrating a multi-projector system in a non-planar environment without using 3D reconstruction. This method corrects the geometric calibration error caused by the screen's manufactured imperfections, such as an undulating surface or a slant in the vertical plane. In addition, based on actual projection demand, this paper presents the overall performance evaluation criteria for the multi-projector system. According to these criteria, we determined the optimal placement for the projectors. This method also extends to surfaces that can be parameterized, such as spheres, ellipsoids, and paraboloids, and demonstrates a broad applicability.

Keywords Multi-projector · Geometric registration · Non-planar display · Placement optimization

1 Introduction

Non-planar projection is often used to create immersive, high-resolution displays in the fields of virtual reality and digital media. Two challenges exist for non-planar projection. First, images directly projected onto a non-planar screen distort, and a pre-warp process is required to make the projection image fit the screen's shape. Second, to enhance the display's resolution and expand the illumination's dynamic range, multiple projectors are usually necessary. Therefore, the geometric alignment and illumination fusion of the overlap region between the different projectors are unavoidable problems.

To solve these problems, two projection correction methods have been developed. One method, based on a three-dimensional (3D) reconstruction, recovers the geometric shape and relative positions of the projectors in the scene. The other method avoids this 3D reconstruction, aligning the texture by transforming two-dimensional (2D) images. Gaining all of a scene's information using 3D reconstruction benefits the generation of projection images, especially for projection systems with arbitrary views. However, the cost of reconstruction devices is high, and the process is complex. Methods that do not use reconstruction typically correct the images using a homography or a Bezier deformation and determine the error using a camera or the human eye. This technology is readily available and efficient. However, along with the scene's unknown information, the attached fiducials and other information should be gathered to build a correspondence between the projectors and the display screen. This process requires a substantial amount of manual operation.

Based on the advantages of the two methods, we present an accurate projection calibration method for cylindrical surface. In addition, a set of evaluation criteria is proposed

✉ Dongdong Weng
crgj@bit.edu.cn

[1] School of Optoelectronics, Beijing Institute of Technology, Beijing 100081, China

that assesses the projection effect and optimizes the positions of the projectors to enhance the display performance. Compared with the existing methods, our paper presents several new features.

1. Reconstruction of the scene is unnecessary because the correspondence between the projectors and screen is measured using the calibrated camera.
2. Our proposed constrained calibration method using multiple projectors employs all the projectors' information, and the calibration errors due to the screen's deformation are effectively reduced.
3. Overlap images are aligned using a geometric adjustment process. The mismatch error can be automatically and rapidly corrected without manual operation.
4. The influence of the projector's pixel size and defocus are addressed. An evaluation of the projection provides a reference for the projector's deployment.

2 Related work

Because projectors are not able to directly retrieve information about a scene, the traditional projection calibration method often uses cameras to capture the images. In the field of planar projection, Raij and Pollefeys [1] proposed an auto-calibration method that did not use any artificial markers. A fusion of multi-projector images resulted, but the assumed point positions and other parameters were inconsistent with the real system, which led to some calibration errors. Based on a conformal texture mapping, Raskar et al. [2] produced a self-adaptive projection calibration using calibrated projectors and cameras. Yang et al. [3] and Raij et al. [4] proposed a multi-projector calibration method for planar screens, which quickly recalibrates repositioned projectors. Bhasker et al. [5] used Bezier patches to create a planar projection. Chen et al. [6] implemented multi-projector calibration using a homography matrix. Bhasker et al. [7] presented an asynchronous distributed projection calibration system. His projection unit consists of a projector and a camera, and the number of projectors in the system can be dynamically increased or decreased. In general, planar projection correction technology is not difficult to use, and many mature theories and practical systems have been developed.

Existing non-planar projection calibration methods can be divided into two groups: those that use 3D reconstruction and those that do not. Methods based on 3D reconstruction require multiple projectors and cameras. The relationship between the matching points of the projectors and cameras is found using structured light or feature points to obtain the parameters of the scene and projectors. Raskar et al. [8] calibrated the projectors and cameras using 3D fiducials and retrieved the scene information with structured light. In addition, he proposed a two-step image-rendering method for view-dependent applications. Aliaga [9] and Xu [10] used cross validation to examine the calibrated parameters and model shape, and they achieved a self-calibration of the projection system by applying photometric and geometric theory. Raskar et al. [11] reconstructed the quadric surface using a stereo camera and minimized the pixel alignment error using conformal mapping and a quadric surface transformation.

These methods usually require a pre-calibration of the projectors and cameras to determine their relative positions. Even if a projection system has been calibrated, it needs another recalibration after the devices have been moved. To correct the projection images in accordance with the real-time changing environment, several technologies have been presented. Yang and Welch [12] used feature-point matching of the images and Kalman filtering to produce the screen's shape. Both the projectors and screen can move when the system is working. Cotting et al. [13, 14] embedded invisible fiducials into the projection images and used a digital light-processing (DLP) projector to generate the imperceptible encoded pattern. However, the cost benefit is achieved only with a sacrifice in the scope of the display's dynamic range. Johnson [15, 16] used stereo cameras and structured light to initially calibrate the system. The scene's structure is restored by matching the images of the projector and camera. This process allows all the projectors to move simultaneously, but the projection screen's shape still needs to be constant and known. Zhou's method [17] is similar to Johnson's in that every projector unit was combined with a camera, and it could not be determined whether the screen or the projectors had moved. If the screen moves, it must be planar. Zollmann's method [18] addressed the issue of minor screen deformation and produced a view-dependent calibration. Newcombe et al. [19] presented a tracking method, which collected the space's information using a depth camera. Tan et al. [20] used the depth camera to achieve a self-adaptive calibration for a planar screen.

Calibration methods that do not use complex 3D reconstruction in the non-planar environments have also been proposed. Based on the transform of 2D images, Brown [21, 22] calibrated multiple projectors with one camera. This method's disadvantage is its requirement that the display be rendered from the camera's perspective. Harville et al. [23] and Sun et al. [24] attached artificial fiducials to the screen's top and bottom edges and determined the projectors' and the screen's correspondence using a linear mapping. This method does not provide screen's 3D information and cannot be used for view-dependent applications. Sajadi and Majumder [25] assumed the parallelism of the screen's top and bottom edges, which limited its applicability to smooth, vertical surfaces without severe deformation. He then extended this method to vertical extruded surfaces in [26, 27].

The method presented in this paper avoids 3D reconstruction and parameterizes the camera and the screen.

Sajadi's method assumed that the top and bottom edges had the same shape, which is not the case, and the calibration errors due to this assumption cannot be reduced. Our method allows for some inconsistency between the screen's mathematical expression and its actual shape. Error caused by the screen's deformation is corrected in subsequent steps.

Multiple approaches have been proposed to enhance projection quality. Sajadi et al. [28] attached an extra lens in front of the projector. This increased the projector's pixel density but reduced the temporal resolution, losing 50 % of the illumination efficiency. They improved this method using the shifted images as transcripts to increase the illumination efficiency [29]. However, this method does not address the situation in which multiple projectors are used. Law et al. [30] optimized the projectors' positions and orientations to obtain the required display effect. Based on the minimal projector radiation efficiency, a surface's reflection and an object's distance from the projectors were considered when calculating the projectors' optimal positions. The problem was simplified, but it was assumed that all the projectors were positioned a fixed radius around the object. Moreover, the result's evaluation primarily considered the relationship between the projection illumination and the position of the projectors. The influence on the projection pixel density was not analyzed. Aliaga et al. [31] modeled the correspondence between the projection pixels and the projectors using an ellipse Gaussian distribution. In this paper, we analyze the requirements necessary to produce better projection performance, and we propose a set of standards that evaluates the projection's effect. In addition, we construct an optimization problem to determine the optimal placement for projectors.

3 Geometric calibration of projectors

For projector-based high immersive display systems, a projection can be considered to be a "wallpaper" pasted on the screen for viewpoint-independent applications, and the projectors' intrinsic and extrinsic parameters are necessary to align the projected image with the real objects. Previous methods rely on manual adjustment to produce projection texture alignment, and they require experienced operators. When the screen or projectors are moved, the calibration must be repeated, and this is a time-consuming process. The method proposed by this paper uses a camera as an auxiliary measuring tool, and structured light is applied to establish a pixel mapping between the camera and projectors. The calibration program operates automatically, without human intervention.

Figure 1 displays the process flow of a projection calibration for a cylindrical screen. The screen's shape should be parameterized first to enable a mathematical description of the screen's appearance. To obtain the camera's position, artificial markers or other means are employed to determine the corresponding points between the screen and camera. Once the screen's shape and the camera's position have been determined, a 2D–3D mapping is created from every pixel in the camera to a 3D point on the screen. A pixel mapping between the camera and projectors is generated using structured light. With assistance from the camera, the correspondence between the projector and screen is established, and the projectors' intrinsic and extrinsic parameters (the projection matrix) are calibrated. Because the shape of the screen is not exactly represented by the mathematical description, the initial calibration result is not acceptable. We apply constraints between adjacent projectors to refine the calibration result. This step can reduce error, but the texture misalignment is still visible to the human eye. Therefore, to minimize the texture alignment error, we use a Bezier interpolation-based image-morphing method to adjust the projected image. This renders the texture misalignment imperceptible. During the photometric calibration process, we employ color uniformization and intensity fusion to guarantee that the projected image is appropriately bright. In this paper, we focus on the geometric calibration of the projector; the photometric calibration is not explained in detail.

3.1 Estimating projector position

In this paper, the camera's 3D coordinate is denoted $\boldsymbol{X}_c = [x_c, y_c, z_c]^T$, and the camera's corresponding 2D point is $\boldsymbol{x}_c = [u_c, v_c]^T$. The projector's 3D coordinate is $\boldsymbol{X}_p = \left[x_p, y_p, z_p\right]^T$, and the projector's corresponding 2D point is denoted $\boldsymbol{x}_p = \left[u_p, v_p\right]^T$. The world's 3D point is $\boldsymbol{X}_w = [x, y, z]^T$, and its corresponding 2D point, which is on the surface expanded along the generatrix of the cylinder, is $\boldsymbol{x}_w = [s, t]^T$. The relationship between all of these defined symbols is illustrated in Fig. 2. The width w, height h, and depth d are previously measured, and the cylinder's radius is $R = (4d^2 + w^2)/8d$. The field of vision for the cylindrical screen is determined to be $\theta = 2\sin^{-1}(4wd/(4d^2 + w^2))$, where R, θ, and h are sufficient to parameterize the screen's shape.

Sajadi and Majumder [27] required the following preconditions in his paper.

1. Both the projector and camera form a linear system (pinhole model), and the distortion is not addressed.
2. The screen is a vertically extruded surface.

This study also requires precondition (1) but relaxes the requirement for precondition (2). Because of the

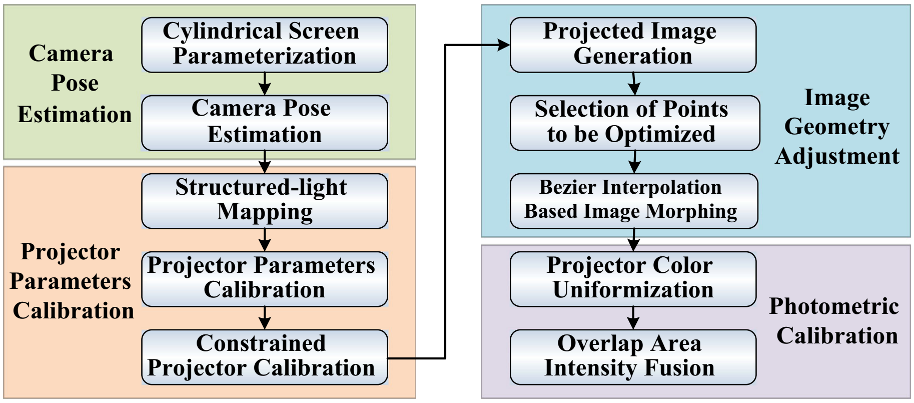

Camera Pose Estimation
Cylindrical Screen Parameterization
Camera Pose Estimation
Projector Parameters Calibration
Structured-light Mapping
Projector Parameters Calibration
Constrained Projector Calibration
Projected Image Generation
Selection of Points to be Optimized
Bezier Interpolation Based Image Morphing
Image Geometry Adjustment
Projector Color Uniformization
Overlap Area Intensity Fusion
Photometric Calibration

Fig. 1 (Color online) Process flow of the projection calibration

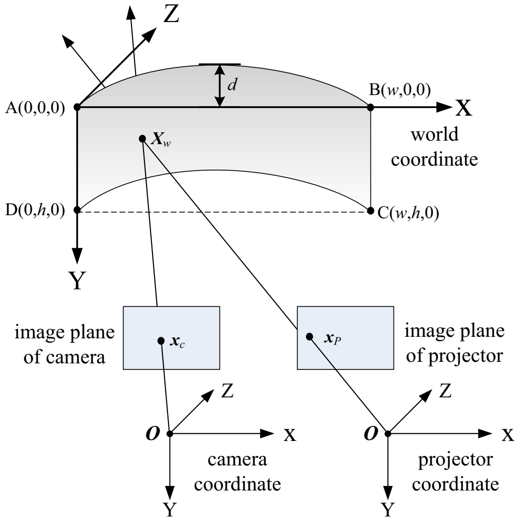


Fig. 2 (Color online) Relationship between points with different coordinates

manufacturing error and installation influence, we cannot guarantee that the screen's upper and lower edges are parallel. In addition, there may be some wavy distortions in the screen, and these factors, which do not follow precondition (2), produce imprecise results using Sajadi's method. Our method does not rely on precondition (2), and it allows some inconsistency between the mathematical expression and the actual shape of the screen. The error caused by the screen's deformation is corrected in subsequent steps, which are specifically described in Sects. 3.1.3 and 3.2.

3.1.1 *Estimating camera position*

The imaging process illustrated in Fig. 2 can be described by the projection formula (1). Here, $\tilde{\boldsymbol{x}}_c = [\boldsymbol{x}_c^T, w]^T$, $\tilde{\boldsymbol{x}}_p = [\boldsymbol{x}_p^T, w]^T$ and $\tilde{\boldsymbol{X}}_w = [\boldsymbol{X}_w^T, w]^T$ are homogeneous coordinates of $\boldsymbol{x}_c$, $\boldsymbol{x}_p$ and $\boldsymbol{X}_w$, respectively. $\boldsymbol{P}_c$ is a 3 × 4 projection matrix of the camera which represents the product of $\boldsymbol{K}_c$ and $[\boldsymbol{R}|\boldsymbol{T}]$. $\boldsymbol{K}_c$ is a 3 × 3 intrinsic matrix of the camera which is pre-calibrated. $[\boldsymbol{R}|\boldsymbol{T}]$ (combination of 3 × 3 rotation matrix $\boldsymbol{R}$ and 3 × 1 translation vector $\boldsymbol{T}$) is a 3 × 4 extrinsic matrix which represents the relative position between the camera and screen. $\boldsymbol{P}_p$ is a 3 × 4 projection matrix of the projector.

To acquire the extrinsic matrix $[\boldsymbol{R}|\boldsymbol{T}]$, the correspondences between the 3D points on the screen and the 2D image points in the camera must be found. Since $[\boldsymbol{R}|\boldsymbol{T}]$ has six degrees of freedom, it requires more than six pairs of correspondent points to solve formula (1) linearly [32]. If all the 3D points are coplanar and the intrinsic matrix $\boldsymbol{K}_c$ is known, then only four pairs of correspondent points are enough [33]. *A*, *B*, *C* and *D* are coplanar points in Fig. 2 and their coordinates are known, since the screen has been parameterized. The correspondent 2D image points of *A*, *B*, *C* and *D* can be obtained through corner extraction of the screen image or manual specification. Using correspondent points of *A*, *B*, *C* and *D*, the extrinsic matrix $[\boldsymbol{R}|\boldsymbol{T}]$ of the camera can be obtained.

$$\begin{cases}\tilde{\boldsymbol{x}}_c = \boldsymbol{P}_c\tilde{\boldsymbol{X}}_w \\ \boldsymbol{P}_c = \boldsymbol{K}_c[\boldsymbol{R}|\boldsymbol{T}] \\ \tilde{\boldsymbol{x}}_p = \boldsymbol{P}_p\tilde{\boldsymbol{X}}_w\end{cases} \tag{1}$$

Sajadi's approach [27] extracted the screen's upper and lower edges in the camera image and used an optimization method to determine the camera's position. His method is feasible under the condition that the screen is a vertically extruded surface but infeasible when the manufacturing error is substantial. We use the four points *A*, *B*, *C* and *D*, which are considered to be coplanar, to calculate the camera's position. Although the result may also be influenced by the manufacturing error, we refine it in the subsequent process.

3.1.2 *Initial projector calibration*

Formula (2), in which $\boldsymbol{L}$ represents the ray passing through point $\boldsymbol{x}_c$ in the camera, is derived from formula (1). Here,

$\boldsymbol{P}_c^+$ is the pseudo-inverse of $\boldsymbol{P}_c$. Using structured light [8], a pixel mapping from the projector coordinates to the camera coordinates which is denoted as $\boldsymbol{x}_c = f(\boldsymbol{x}_p)$ is established. The inverse mapping of $\boldsymbol{x}_c = f(\boldsymbol{x}_p)$ is denoted as $\boldsymbol{x}_p = f^{-1}(\boldsymbol{x}_c)$. Using formula (2) and the mathematical expression of the screen, the corresponding 3D point (denoted as $\boldsymbol{X}_w$) of $\boldsymbol{x}_c$ on the screen can be obtained by calculating the intersection of line $\boldsymbol{L}$ and the screen. Then, function $f_b()$ which represents the mapping from the camera point to its corresponding 3D point on the screen can be established by formula (3). Thus, the mapping from projector's 2D point $\boldsymbol{x}_p$ to its corresponding 3D point $\boldsymbol{X}_w$ is determined in formula (4). Figure 3 shows the corresponding relationship for these points.

$$\boldsymbol{L} = \boldsymbol{P}_c^+\tilde{\boldsymbol{x}}_c \tag{2}$$

$$\boldsymbol{X}_w = f_b(\boldsymbol{x}_c) \tag{3}$$

$$\boldsymbol{X}_w = f_b(\boldsymbol{x}_c) = f_b(f(\boldsymbol{x}_p)) \tag{4}$$

$$\boldsymbol{x}_{src} = f_{s2p}(\boldsymbol{x}_p) \tag{5}$$

Using (4), we sample some of the projector's 2D points and determine their corresponding 3D points. Then, the projector's projection matrix $\boldsymbol{P}_p$ is calculated using (1). Since the texture image is "wallpapered" on the cylindrical screen, $\boldsymbol{X}_w$ has a unique corresponding point $\boldsymbol{x}_{src}$ in the texture image. To create the projection image, the mapping $f_{s2p}()$ which is from projector point $\boldsymbol{x}_p$ to the texture image point $\boldsymbol{x}_{src}$ can be established through (3) and (4) and denoted as formula (5).

3.1.3 Constrained projector calibration

After the initial calibration, projected images for all the projectors have been created. Figure 4 exhibits the display result of an example using three projectors. There are obvious mismatches in the overlap region of the adjacent projectors because the screen's actual shape is not exactly represented by the mathematical model. Figure 4 demonstrates that the mismatch obtained in the right overlap region is more substantial than that resulting in the left overlap region. Further, the mismatch observed in the same overlap region is not uniform. Focusing on the left overlap region, we see that the mismatch in the lower part is more substantial than that in the upper part.

We apply a constrained projector calibration to uniform the mismatch caused by the inaccurate initial projector calibration. This step may not reduce the average mismatch error, but it simplifies the adjustment of the image, as can be seen in Sect. 3.2.

Figure 5 demonstrates the constrained projector calibration process. The screen's 3D point $\boldsymbol{X}_w$ on the screen corresponds to $\boldsymbol{x}_l$ and $\boldsymbol{x}_r$ in different projectors using formula (1). The 2D point $\boldsymbol{x}_{lnew}$, which corresponds to $\boldsymbol{x}_r$ using a structured-light mapping and is displayed in formula (6), is not equivalent to $\boldsymbol{x}_l$ because of the calibration error. There are two ways to acquire the corresponding points necessary to calibrate the projector. The first method uses formula (4), and the other utilizes (6). We select some of the sample points using (4) and the rest using (6). Because some of these sample points are determined using adjacent

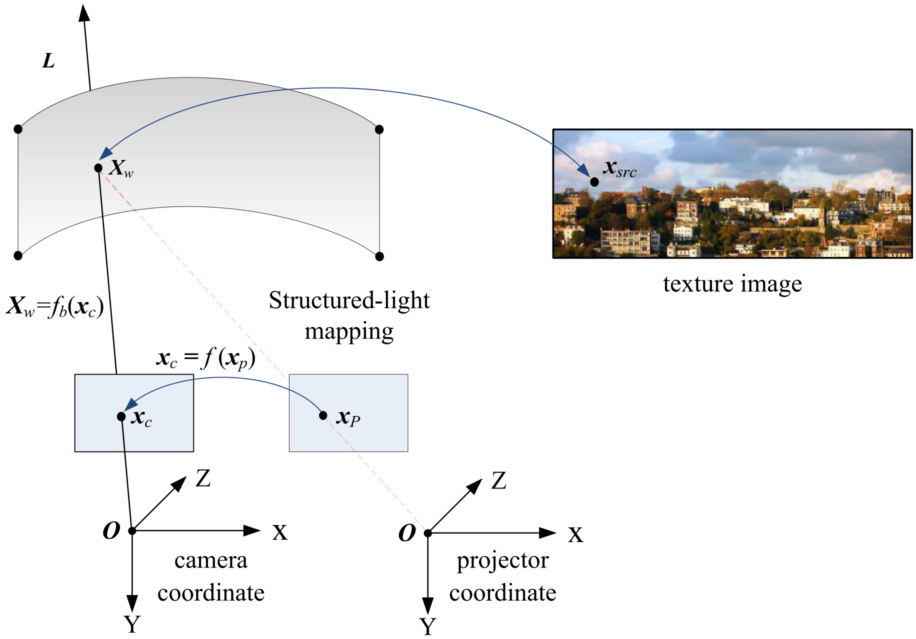


Fig. 3 (Color online) Mapping from 2D point to 3D point

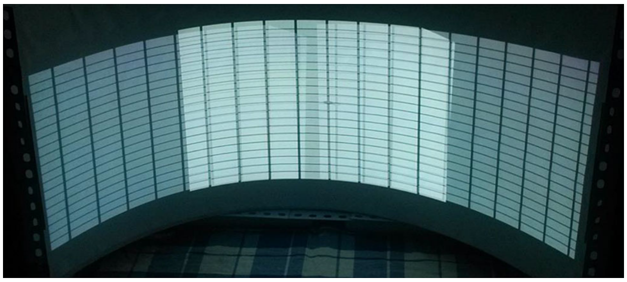


Fig. 4 (Color online) Display result after initial projector calibration

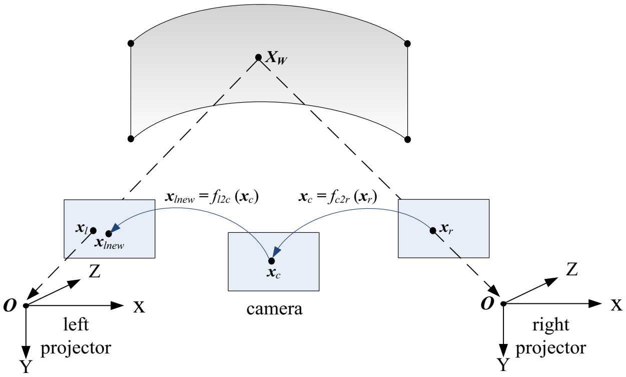


Fig. 5 (Color online) Constrained projector calibration

projectors, the calibration result of one projector is constrained by its adjacent projectors. This constrained method exploits the corresponding points' information, and it is more robust to wrong corresponding points.

$$x_{lnew} = f_{l2c}(x_c) = f_{l2c}(f_{c2r}(x_r)) \tag{6}$$

3.2 Image geometry adjustment

After performing the constrained projector calibration, perceptible texture misalignments in the overlap area still exist, as is shown in Fig. 6. The screen's deformation is very complex, and a non-linear method is required to minimize the mismatches.

Bezier interpolation is a very effective method for adjusting the image in a local patch, and we employ it to construct an optimization function, denoted by formula (7), where (u, v) is the pre-processing image coordinate, $\boldsymbol{x_B}$ is the post-processing image coordinate, and m, n are the control points' respective number of rows and columns in the Bezier function. Here, $\boldsymbol{p}_{j,k}$ is the coordinate of the control point in the jth row and kth column. In addition, $BEZ()$ is the basis function [5].

$$\boldsymbol{x_B} = P(\boldsymbol{u}, \boldsymbol{v}) = \sum_{j=1}^{m}\sum_{k=1}^{n} \boldsymbol{p}_{j,k} BEZ_{j,m}(v) BEZ_{k,n}(u) \tag{7}$$

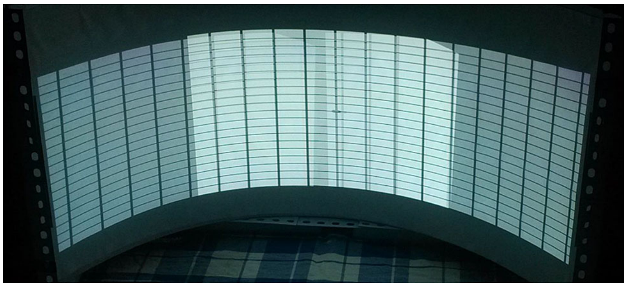


Fig. 6 (Color online) Display result after constrained projector calibration

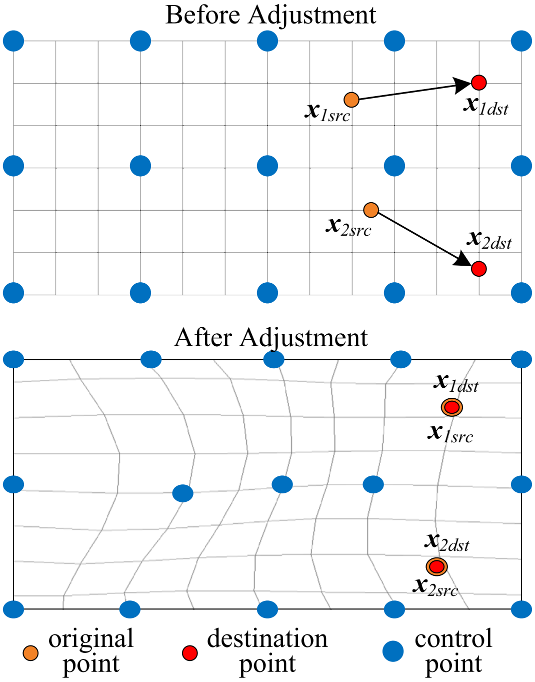


Fig. 7 (Color online) Bezier interpolation-based image adjustment

Previous methods often required manual operation to move the control points, which was time consuming. Because the movement of a control point does not have an intuitive visual effect on the image point, the operation is considered experience dependent. Using (6), we determine the correspondence between two adjacent projectors, which means we can adjust one projector's image to align it with the others. Figure 7 illustrates the process that uses Bezier interpolation to rearrange the mismatched points in the overlap region. Here, $\boldsymbol{x}_{\rm src}$ denotes the projector's original 2D image point, and $\boldsymbol{x}_{\rm dst}$ is the destination position. We define formula (8) to indicate the average differences between all $\boldsymbol{x}_{\rm src}$ and $\boldsymbol{x}_{\rm dst}$, where N is the number of image points in the overlap area.

$$E_{\rm dst} = \frac{1}{N}\sum_{i=1}^{N} \boldsymbol{x}_{i\rm src} - \boldsymbol{x}_{i\rm dst2} \tag{8}$$

To determine the optimal control points' coordinates, we construct the optimization problem shown in formula (9). Here, $\boldsymbol{p}$ denotes the control point's coordinates, and $\boldsymbol{x}_B$ denotes the image coordinates calculated using formula (7).

$$\min f(\boldsymbol{p}) = \frac{1}{N}\sum_{i=1}^{N} \boldsymbol{x}_{i\mathrm{dst}} - \boldsymbol{x}_{iB2}^{2} \tag{9}$$

The optimization problem in (9) has no limitations and could result in an infinite number of solutions. Considering that the range of the control point's movement is limited and only points in the overlap area should be adjusted, we add some constraints to the optimization problem in (9) to obtain formula (10), where $\boldsymbol{p}_0$ is the original position of the control points, and $w(\boldsymbol{p}_i)$ is the range of motion of $\boldsymbol{p}_i$, which is specifically described in Sect. 5.3.

$$\begin{cases} \min f(\boldsymbol{p}) = \frac{1}{N}\sum_{i=1}^{N} \boldsymbol{x}_{i\mathrm{dst}} - \boldsymbol{x}_{iB2}^{2} \\ \text{s.t.} \quad \boldsymbol{p}_i - \boldsymbol{p}_{0i2} \le w(\boldsymbol{p}_i) \end{cases} \tag{10}$$

3.3 Multi-projector expansion

When more than two projectors are used, projectors in the middle may create two overlap regions with their neighbor projectors. Figure 8, taking three projectors as an example, demonstrates the configuration of the projection system. If the image adjustment is successively performed from the first projector to the last one, the image belonging to the middle projector (projector 2 in Fig. 8) will be adjusted twice (one for projector 1 and 2, another for projector 2 and 3). In this case, it is possible that the latter process will conflict with the former process. To avoid this, we design a strategy that processes every projector image no more than once. The first projector's image is not adjusted, and only the region that overlaps the previous projector's image is processed for the remaining projectors' images. That means if the ith projector overlaps with the $(i+1)$th projector, we only adjust the $(i+1)$th projector's image. To give an intuitive description, the adjusted area of the projector's image is painted red in Fig. 8. This strategy produces efficient processing and does not introduce extra error.

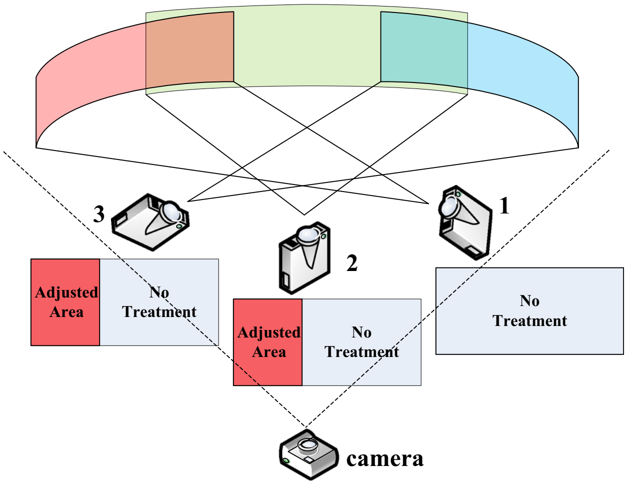


Fig. 8 (Color online) Multi-projector expansion

Our calibration method requires the use of the camera as the measuring tool. Therefore, the available range of the cylindrical screen is limited by the camera's field of view. The view field of a common camera is no more than 100° and can serve four projectors at most. To achieve a 360° projection, the common camera is replaced with a panoramic camera, which has a very high resolution and can cover the entire cylindrical screen. This is the only difference.

3.4 Expansion to other parametric surface

When we use a cylindrical screen, the 3D point $\boldsymbol{X}_w = [x, y, z]^T$ on the screen shown in Fig. 2 must meet the cylinder parametric Eq. (11).

$$\begin{cases} (x - w/2)^2 + (z - d + R)^2 = R^2 \\ R = (4d^2 + w^2)/8d \\ x \in [0, w] \\ y \in [0, h] \\ z \in [0, d] \end{cases} \tag{11}$$

The parametric equation of line $\boldsymbol{L}$ which passes through $\boldsymbol{X}_w$ and $\boldsymbol{x}_c$ in Fig. 3 and has been given by formula (2) can be transformed to (12), in which $[x_o, y_o, z_o]^T$ are the coordinates of the camera's optical center. If the rotation matrix $\boldsymbol{R}$ and translation vector $\boldsymbol{T}$ of the camera are known, $[x_o, y_o, z_o]^T = -\boldsymbol{R}^{-1}\boldsymbol{T}$ is the optical center. Combining (11) and (12), the coordinates of $\boldsymbol{X}_w$ can be solved. That means 3D point $\boldsymbol{X}_w$ on the screen can be determined by its 2D projection point $\boldsymbol{x}_c$ in the camera and formula (3) gives a general description of this solution process.

$$\begin{cases} x = x_o + at \\ y = y_o + bt\,, t \in \boldsymbol{R} \\ z = z_o + ct \\ \boldsymbol{L} = \boldsymbol{P}_c^{+}\tilde{\boldsymbol{x}}_c = \begin{bmatrix} a \\ b \\ c \\ w \end{bmatrix} \end{cases} \tag{12}$$

If the screen has been changed to other parametric surfaces, taking sphere for example, we can define the sphere parametric equation by (13), in which $[x_s, y_s, z_s]^T$ is the center of the sphere and R is the radius. Combining (12) and (13), the 3D point $\boldsymbol{X}_w$ on the spherical screen can also be determined by its 2D projection point $\boldsymbol{x}_c$ in the camera.

$$\begin{cases} (x - x_s)^2 + (y - y_s)^2 + (z - z_s)^2 = R^2 \\ x \in [x_1, x_2] \\ y \in [y_1, y_2] \\ z \in [z_1, z_2] \end{cases} \tag{13}$$

If a screen can be parameterized, we can always achieve its parametric equation $S(\boldsymbol{x}) = 0$. Then, combine formula (12) and the screen's parametric equation, the 3D point's coordinates on the screen can be determined using its projection 2D point in the camera image. In this way, since the pixel mapping from the projector to the camera can be established using structured light, the mapping from projector's image point to the 3D point on the screen [like formula (4)] can be achieved and the projector can be calibrated by taking the camera as a measuring instrument. After the projection matrix calibration of the projector, the remaining correction processes are the same with the cylinder screen, so our method can be extended to other parametric surfaces by just replacing the parametric equations.

4 Projector placement optimization

When the actual projector is not an ideal optical instrument, the projection result is affected by the projector's capability and utilization. After analyzing the requirements for better projection performance, we propose a set of standards to evaluate the projection. Using these standards, we construct an optimization problem to obtain the optimal placement for the projectors.

4.1 Factors affecting projection performance

Resolution and brightness (or illumination) are the most important properties of projection. Although further study has been performed concerning brightness optimization [31], this paper focuses on the effect of projector placement on resolution and performance.

4.1.1 Resolution

There are two types of resolution. The first type is called honey-point resolution, in which the resolution depends on the audience's viewpoint. The other is called absolute resolution, which represents the pixel density on the real screen.

The computation necessary to process all the points on the screen to calculate the resolution or other parameters is excessive. In practice, we select some sample points, which are uniformly distributed on the screen, for every projector and use the sample value to approximate the global value.

In this paper, honey-point resolution is quantified as the size of the sample point's pixel, which is captured by a camera positioned on the honey point. It should be noted that there is no unit for the size of a pixel. The smaller a pixel is, the better the resolution. Figure 9 illustrates the method for calculating a honey-point resolution. Here, $\boldsymbol{X}_{ij}$ is the 3D coordinate of the jth sample point, which corresponds to the ith projector. The calculation process is as follows:

(a) Using formula (1), we determine the image point $\boldsymbol{x}_{ij}$ in the projector and then select four adjacent points A, B, C, and D of $\boldsymbol{x}_{ij}$. Formula (4) produces the corresponding 3D points of A, B, C, and D, which we denote as $\boldsymbol{X}_A$, $\boldsymbol{X}_B$, $\boldsymbol{X}_C$, and $\boldsymbol{X}_D$, respectively.

(b) The 3D points $\boldsymbol{X}_A$, $\boldsymbol{X}_B$, $\boldsymbol{X}_C$, and $\boldsymbol{X}_D$ are captured by the camera, and the corresponding 2D image points are denoted $\boldsymbol{x}_A$, $\boldsymbol{x}_B$, $\boldsymbol{x}_C$, and $\boldsymbol{x}_D$. These four points constitute a quadrilateral with area written as S_{abcd}. Because the four image points approximate a square, the pixel size of sample point $\boldsymbol{X}_{ij}$ can be expressed by formula (14). Here, $\boldsymbol{x}_i = [r_{xi}, r_{yi}, r_{zi}, t_{xi}, t_{yi}, t_{zi}]^T$ is the rotation and translation parameter of the ith projector.

$$A\left(\boldsymbol{x}_i \boldsymbol{X}_{ij}\right) = \frac{1}{2}\sqrt{S_{abcd}} \tag{14}$$

We can derive formula (15) from (14), which signifies the average pixel size of all sample points in the ith projector, where $\boldsymbol{X}_i$ represents all the sample points' coordinates in the ith projector and N_i is the number of sample points.

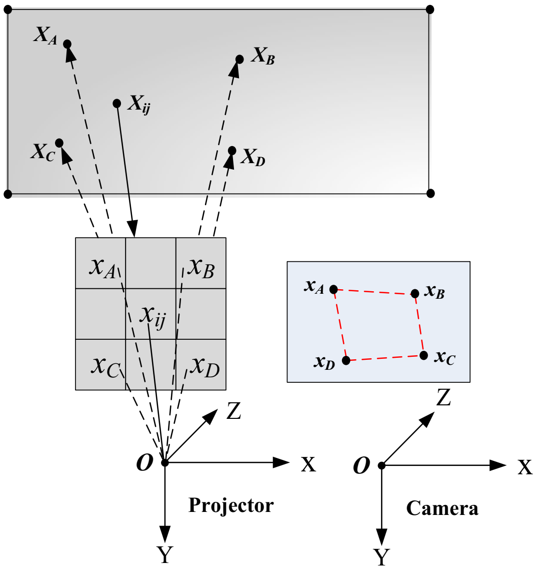


Fig. 9 (Color online) Diagram of honey-point and absolute resolutions

$$A(\boldsymbol{x}_i, \boldsymbol{X}_i) = \frac{1}{N_i}\sum_{j=1}^{N_i} A(\boldsymbol{x}_i, \boldsymbol{X}_{ij}) \tag{15}$$

Absolute resolution is similar to honey-point resolution except that we use the screen's pixel size as the metric (measured in millimeters). The 3D points $\boldsymbol{X}_A, \boldsymbol{X}_B, \boldsymbol{X}_C$, and $\boldsymbol{X}_D$ in Fig. 9 may not be coplanar because the screen is not flat. Because these four 3D points are very close to one another, we can use formula (16) to approximately evaluate the real size of a projector's pixel on the screen. The area of the quadrilateral created by $\boldsymbol{X}_A, \boldsymbol{X}_B, \boldsymbol{X}_C$, and $\boldsymbol{X}_D$ is written as S_{ABCD}. Similar to formula (15), the average absolute pixel size of all sample points for the ith projector is expressed in formula (17).

$$B(\boldsymbol{x}_i, \boldsymbol{X}_{ij}) = \frac{1}{2}\sqrt{S_{ABCD}} \tag{16}$$

$$B(\boldsymbol{x}_i, \boldsymbol{X}_i) = \frac{1}{N_i}\sum_{j=1}^{N_i} B(\boldsymbol{x}_i, \boldsymbol{X}_{ij}) \tag{17}$$

4.1.2 Resolution discrepancy

Resolution discrepancy in a projector represents the resolution differences between the sample points in different areas. A large resolution discrepancy indicates that the shape of the screen has substantially changed in the projector-covered area and the projector's placement should be adjusted, as shown in Fig. 10. The resolution discrepancy of the ith projector is determined using formula (18).

$$C(\boldsymbol{x}_i, \boldsymbol{X}_i) = \frac{1}{N_i}\sum_{j=1}^{N_i} C(\boldsymbol{x}_i, \boldsymbol{X}_{ij}) = \frac{1}{N_i}\sum_{j=1}^{N_i}\frac{\max(\mathrm{B}(\mathbf{x}_\mathrm{i}, \boldsymbol{X}_{\mathrm{ij}}))}{\min(B(\boldsymbol{x}_i, \boldsymbol{X}_{ij}))} \tag{18}$$

4.1.3 Projection distance discrepancy

In practice, a projector has a focal plane, and points far from the plane could blur. The distance discrepancy of the ith projector defines the difference in the distances between the farthest and nearest sample points to the projector's optical center. This can be seen in Fig. 11 and formula (19). The terminology $\mathrm{dist}(\boldsymbol{x}_i, \boldsymbol{X}_i)$ denotes a set of distances from all sample points $\boldsymbol{X}_i$ to the projector's center. To generate a clear projection, the projection distance discrepancy should be as small as possible.

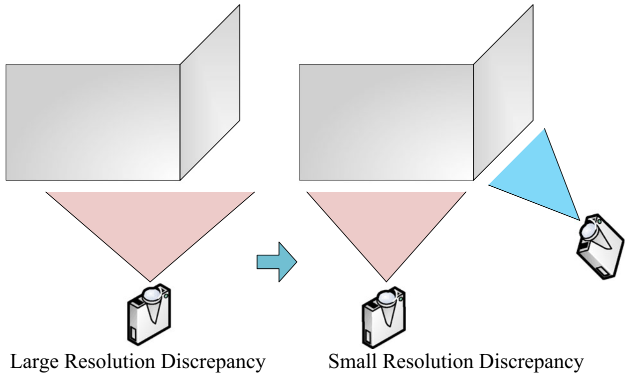


Fig. 10 (Color online) Resolution discrepancy

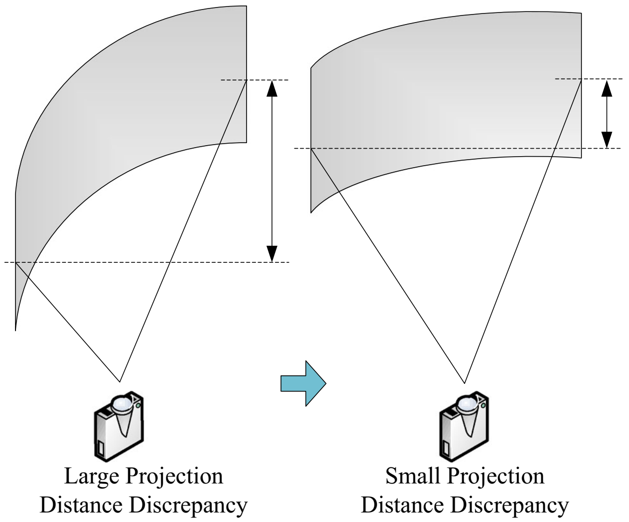

Large Projection Distance Discrepancy
Small Projection Distance Discrepancy

Fig. 11 (Color online) Projection distance discrepancy

$$D(\boldsymbol{x}_i, \boldsymbol{X}_i) = \max(\mathrm{dist}(\boldsymbol{x}_i, \boldsymbol{X}_i)) - \min(\mathrm{dist}(\boldsymbol{x}_i, \boldsymbol{X}_i)) \tag{19}$$

4.2 Projection performance evaluation function

In Sect. 4.1, we proposed four properties that assess the projection's performance. The smaller these properties measure, the better the projector performs. Therefore, we create an evaluation function $f(\boldsymbol{x}, \boldsymbol{X})$ that is the weighted sum of these properties and is shown in formula (20). Here, $\boldsymbol{a} = [a_1, a_2, a_3, a_4]$ denotes the weight coefficient, $\boldsymbol{x} = [\boldsymbol{x}_1; \boldsymbol{x}_2; \ldots; \boldsymbol{x}_M]$ denotes the set of the extrinsic parameters of all the projectors, M denotes the number of projectors, and $\boldsymbol{X} = [\boldsymbol{X}_1; \boldsymbol{X}_2; \ldots; \boldsymbol{X}_M]$ denotes the sample points' 3D coordinates for every projector.

$$\begin{gathered}
A(\boldsymbol{x}, \boldsymbol{X}) = \frac{1}{M}\sum_{i=1}^{M} A(\boldsymbol{x}_i, \boldsymbol{X}_i)\\
B(\boldsymbol{x}, \boldsymbol{X}) = \frac{1}{M}\sum_{i=1}^{M} B(\boldsymbol{x}_i, \boldsymbol{X}_i)\\
C(\boldsymbol{x}, \boldsymbol{X}) = \frac{1}{M}\sum_{i=1}^{M} C(\boldsymbol{x}_i, \boldsymbol{X}_i)\\
D(\boldsymbol{x}, \boldsymbol{X}) = \frac{1}{M}\sum_{i=1}^{M} D(\boldsymbol{x}_i, \boldsymbol{X}_i)\\
f(\boldsymbol{x}, \boldsymbol{X}) = a_1 A(\boldsymbol{x}, \boldsymbol{X}) + a_2 B(\boldsymbol{x}, \boldsymbol{X}) + a_3 C(\boldsymbol{x}, \boldsymbol{X}) + a_4 D(\boldsymbol{x}, \boldsymbol{X})
\end{gathered} \tag{20}$$

Using the evaluation function, we can construct the optimization problem in formula (21), whose solution provides the optimal placement of the projectors. Because a projector's usage is limited by space or other factors, optimal placement cannot always be produced in practice. However, our optimization algorithm is still a useful means for obtaining the result and may prove to be helpful in guiding users through the projector positioning.

$$\mathbf{min}_{\arg x} f(\boldsymbol{x}, \boldsymbol{X}) \tag{21}$$

5 Experiments

Our experiment is designed to verify the accuracy and robustness of our method. The screen we used is handmade and inevitably has some deformation so that the vertically extruded surface requirement is not satisfied, as shown in Fig. 12. We used three projectors with resolutions of 1024 × 768 pixels to illustrate the screen and a camera with a resolution of 640 × 480 pixels as the measuring tool. Cameras with higher resolution will improve the accuracy of the calibration, but they also require more time to process the data. We confirm that the 640 × 480 camera is an acceptable choice by considering both its accuracy and efficiency throughout our experiment.

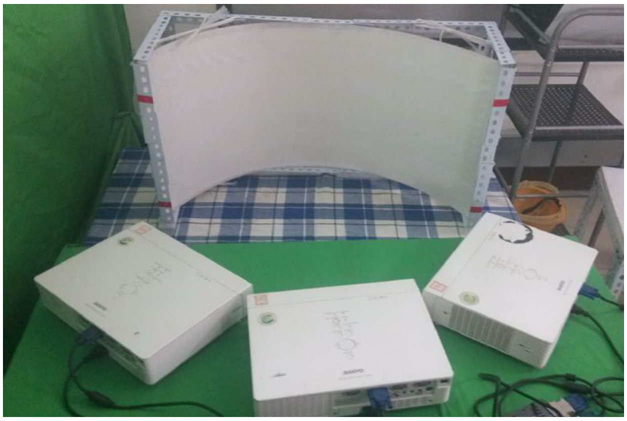


Fig. 12 (Color online) Experiment environment

5.1 Structured-light coding

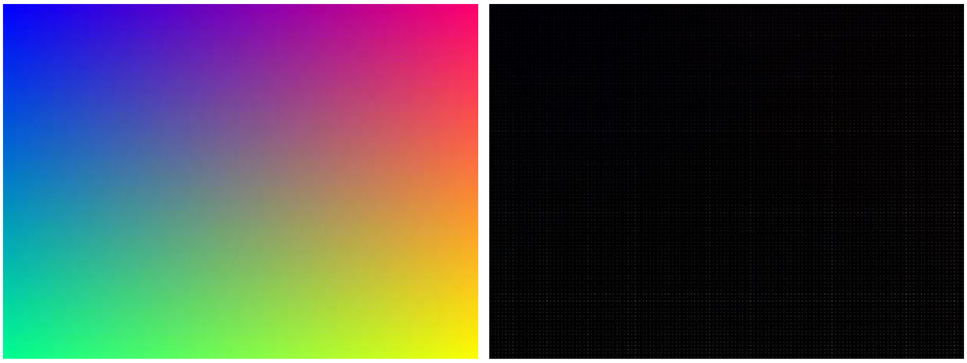
In our experiment, we used Gray code as the structured-light pattern. Because our camera's resolution is low, we used eight Gray-code patterns, which can create $2^8 = 256$ different codes. The number of kinds of codes is less than the number of projector pixels, so one code may correspond to several 2D points in the projector. Before calculating the pixel mapping, we combined the 2D points that had the same code and used their average position as the code's correspondence. To provide an intuitive description, we created a code map, which represents the pixels with different codes, and can be seen in Fig. 13. Every code is illustrated using a unique color, so the per-pixel code image, which uses 11 Gray-code patterns, is denser than the actual code image, which uses only eight Gray-code patterns. The colorful code image can be used to check the result of the structured-light mapping because we can easily determine incorrect matches by comparing the actual image with the original per-pixel code image.

Figure 14 shows the actual code image for the experiment. The original code image has both missing and wrong codes because of the interference due to ambient light. We filtered the wrong codes using epipolar geometric constraints [32] between the camera and projector and then interpolated the points to fill in the "holes" in the code image. This step improves the accuracy of the structured-light mapping.

5.2 Projector calibration

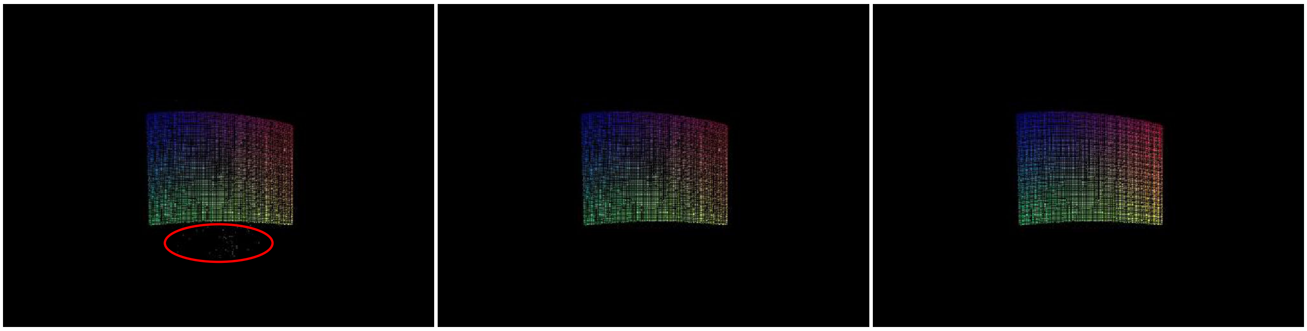
Because too many match points were generated by the structured-light mapping, we only sampled 100 of them to shorten the computation time. The initial calibration result based on (1) is shown in Table 1.

Fig. 13 (Color online) Encoded projector image



Per-pixel code image

Actual code image (zoom to see detail)



Original code image Filtered code image Code image after interpolation

Fig. 14 (Color online) Encoded camera image (using central projector)

Table 1 Result of initial projector calibration

Index of projector	Reprojection error (pixel)
1 (left)	2.72346
2 (central)	2.87783
3 (right)	2.18927

Table 2 Result of constrained projector calibration

Index of projector	Reprojection error (pixel)
1 (left)	2.51407
2 (central)	2.80383
3 (right)	2.41489

In the constrained calibration procedure, the right and left projections are only constrained by the central projection, but the central projection is constrained by the projections on both sides. The results of this constrained calibration are exhibited in Table 2. We see that the reprojection error of projectors 1 and 2 is smaller after performing the constrained calibration, but the reprojection error of projector 3 is larger. The errors of all three projectors do not substantially decrease but are closer to one another, which is the intention of the constrained calibration.

5.3 Image adjustment

After performing the constrained calibration, the mismatch error E_{dst}, which is defined by (8), of two adjacent projectors is shown in Table 3, in which "ratio" denotes the quotient of E_{dst} and the projector's resolution.

To produce a fast computation, we limited our selection to N target points, denoted by $\boldsymbol{x}_{\mathrm{dst}}$, from all the projector's corresponding points and adjusted the image. The image of the corresponding points was separated into $\mathrm{row}_{\mathrm{grid}} \times \mathrm{col}_{\mathrm{grid}}$ regions, and the region that contained corresponding points was called the effective region $\boldsymbol{G}_E$. The corresponding points in every effective region are clustered according to their coordinates, and the target points were selected from these clustered points. Figures 15 and 16 demonstrate the result of the target points selection with $\mathrm{row}_{\mathrm{grid}} = 10$, $\mathrm{col}_{\mathrm{grid}} = 14$, and $N = 250$.

Table 3 Mismatch error (pixel) before image adjustment

Overlap area	Mean	Std	Max	Ratio (%)
1 and 2	6.2169	4.0508	18.601	0.61
2 and 3	12.840	8.7101	36.235	1.25

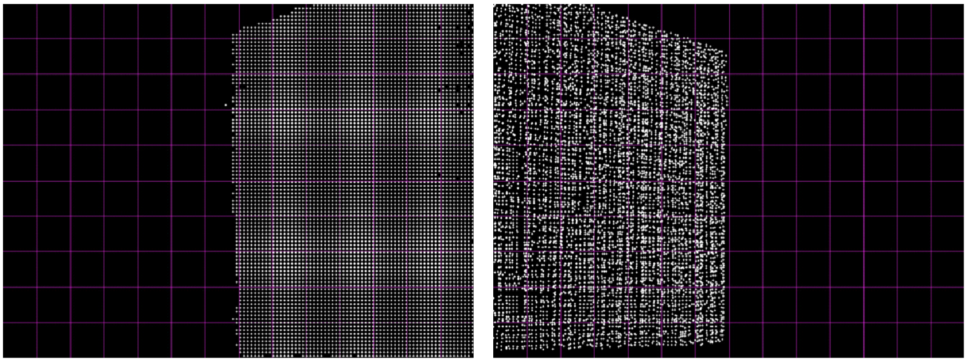
Before performing the image adjustment, we confirm the value of $w(\boldsymbol{p}_i)$ in formula (10). Considering the control points in overlap areas contribute more to the image adjustment, they require a larger moving range than other control points. In practice, the value of $w(\boldsymbol{p}_i)$ is subject to (22), where imagewidth is the width of the projector's image, m denotes the number of rows of the control points, $\boldsymbol{p}_{\mathrm{border}}$ denotes the border coordinates of the effective regions, and C_1 is the calculation coefficient with range [1,5].

$$w(\boldsymbol{p}_i) = \begin{cases} \mathrm{imagewidth}/2m, \boldsymbol{p}_i \in \boldsymbol{G}_E \\ 10/(C_1\boldsymbol{p}_i - \boldsymbol{p}_{\mathrm{border2}}), \boldsymbol{p}_i \notin \boldsymbol{G}_E \end{cases} \tag{22}$$

Figure 17 illustrates the movements of the control points using (9) and (10) after 100 iterations. The image shows that both (9) and (10) move the original points toward the destination points. However, (9) has no constraints for the control points, which will cause undesirable movements for the points not in the overlap area, as shown in Figs. 18 and 19. After 100 iterations of the image adjustment, the mismatch error has obviously reduced. This can be seen in Table 4.

To compare our method with Sajadi's method in accuracy and robustness, we construct a virtual screen which has some bulges to simulate the actual deformation. Figure 20 illustrates the virtual screens with varying

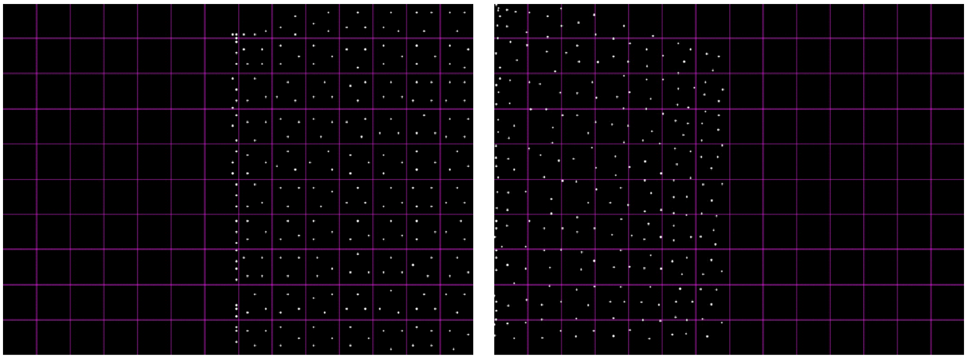
Fig. 15 (Color online) Corresponding points in the overlap area of projectors 2 and 3



Corresponding points of projector 2

Corresponding points of projector 3

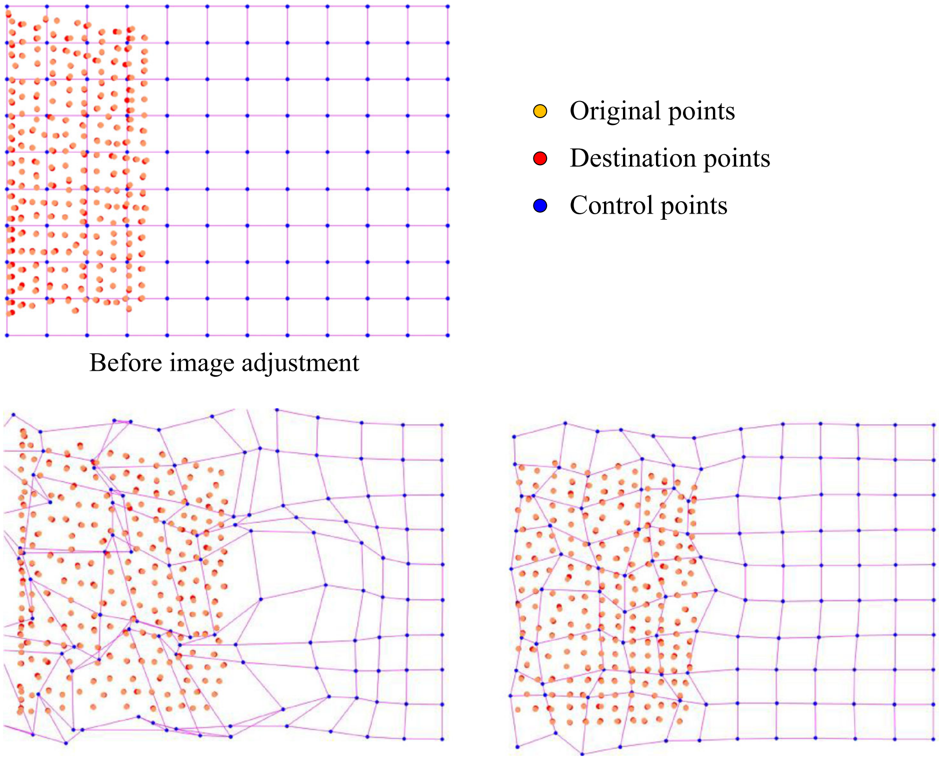
Fig. 16 (Color online) Points selected for image adjustment



Selected corresponding points of projector 2

Selected corresponding points of projector 3

Fig. 17 (Color online) Comparison of unconstrained and constrained adjustments



Before image adjustment

Control points without constraints

Control points with constraints

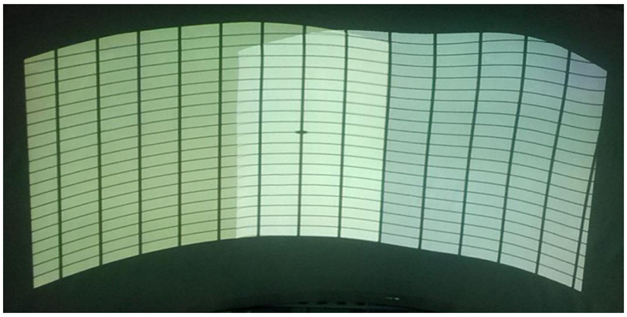


Fig. 18 (Color online) Result of image adjustment with unconstrained control points

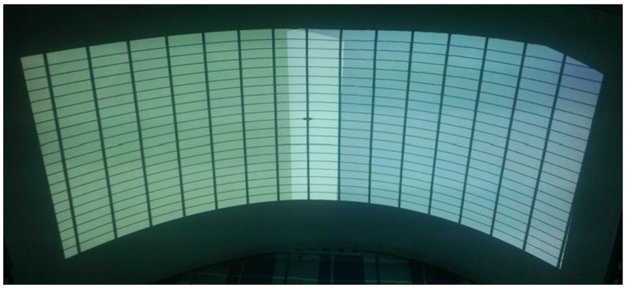


Fig. 19 (Color online) Result of image adjustment with constrained control points

Table 4 Mismatch error (pixel) after image adjustment

Overlap area	Mean	Std	Max	Ratio (%)
1 and 2	2.66275	1.38429	7.58206	0.26
2 and 3	3.11903	1.71371	9.47161	0.30

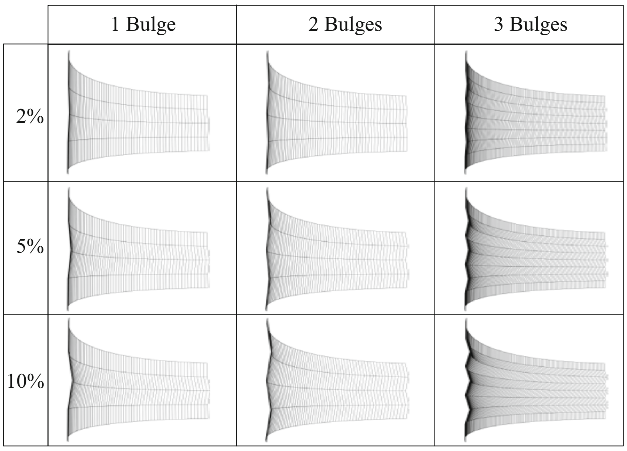


Fig. 20 (Color online) Deformation of the screen

degrees of deformations and the calibration results are exhibited in Table 5. We can see that with the increase of the number and range of the bulges, the mismatch error of

Table 5 Mismatch error (pixel) of our method and Sajadi's method

	Deformation	2 %	5 %	10 %
1 Bulge	Sajadi's method	7.182	8.064	10.621
	Our method	2.639	2.522	2.661
2 Bulges	Sajadi's method	5.324	6.174	9.490
	Our method	2.742	2.636	2.893
3 Bulges	Sajadi's method	8.100	12.080	14.532
	Our method	2.564	2.948	3.167

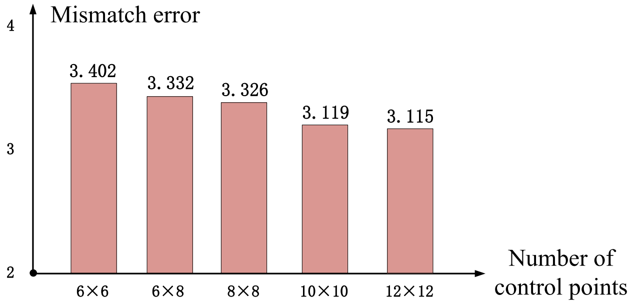


Fig. 21 (Color online) Effect of number of control points on mismatch error

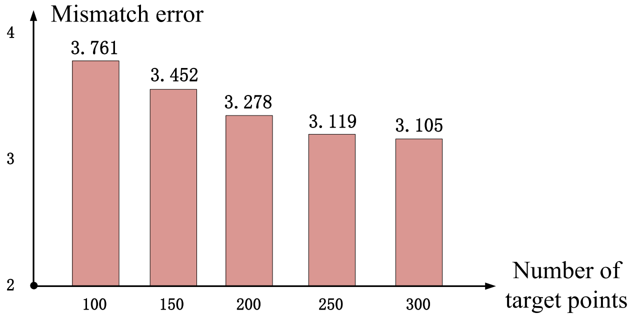


Fig. 22 (Color online) Effect of number of target points on mismatch error

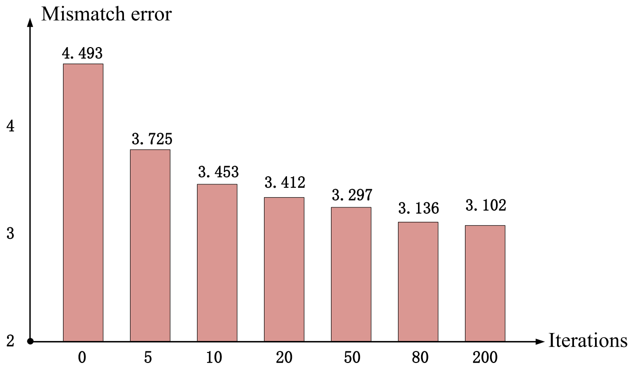


Fig. 23 (Color online) Effect of number of iterations on mismatch error

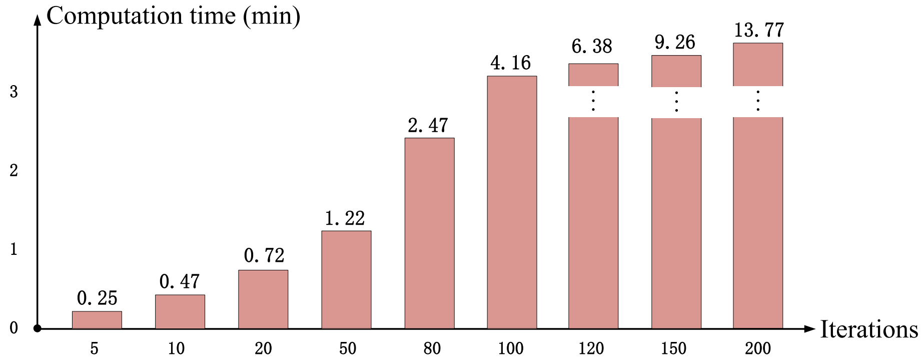


Fig. 24 (Color online) Effect of number of iterations on computation time

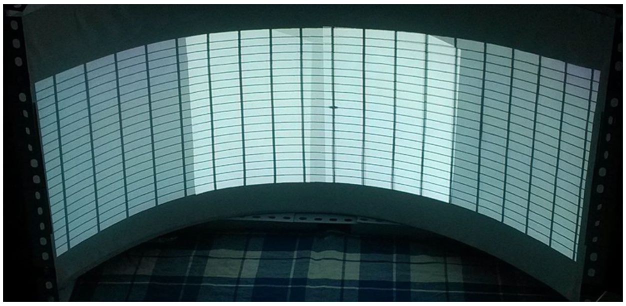


Fig. 25 (Color online) Result after geometric calibration

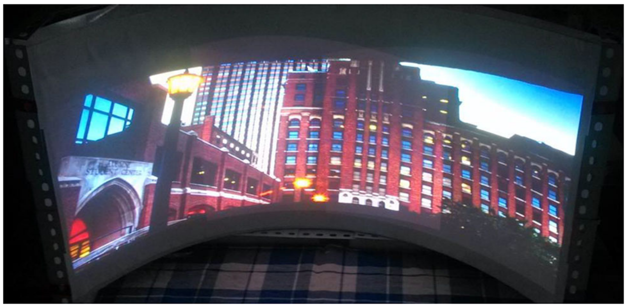


Fig. 26 (Color online) Result after photometric calibration

both methods grows synchronously. However, our method gives better results than Sajadi's method in all conditions.

5.4 Accuracy and efficiency

During the image adjustment procedure, the use of more control and target points produces greater calibration accuracy and a smoother adjustment of the image. However, it also results in a more time-consuming process. Therefore, it is necessary to optimally choose the number of control and target points, which is the topic of this section.

Figures 21 and 22 display the relationship between the number of control points, the number of target points, and the mismatch error produced after 100 iterations. It indicates that the mismatch error will not substantially reduce when the number of control points is greater than 100 or the number of target points exceeds 250.

To solve formula (10), we use the Levenberg–Marquardt algorithm [34]. Figures 23 and 24 demonstrate the relationship between the mismatch error, computation time, and iteration times. They indicate that the mismatch error is not substantially reduced when the number of iterations exceeds 80. We also observe that the computation time significantly increases when the number of iterations is greater than 80.

In summary, we suggest the following parameters: 100 control points, 250 target points, and a maximum of 100 iterations. This set of parameters meets the accuracy requirements and ensures an optimal end time of 5 min.

Figure 25 exhibits the display result after image adjustment. Because this paper does not focus on the photometric calibration, we employed a simple cosine modulation to fuse the intensity of the overlap area [3]. This can be seen in Fig. 26. A more accurate method of photometric calibration was developed by Majumder and Stevens [35].

5.5 Projector placement optimization

Because the actual use of projectors is very time consuming, we validated our method using a software simulation. The projector's resolution and number of projectors were established. Then, we determined the region on the screen illuminated by each projector. Sample points were selected from each of the projector's covered areas, which finalized the preparation.

$$\mathrm{cover}\left(\boldsymbol{x}_i, \boldsymbol{X}_{ij}\right) = \begin{cases} 1, & \text{covered} \\ \varepsilon, & \text{not covered} \end{cases} \tag{23}$$

$$\mathrm{num}(\boldsymbol{x}_i, \boldsymbol{X}_i) = \sum_{j=1}^{N_i} \mathrm{cover}\left(\boldsymbol{x}_i, \boldsymbol{X}_{ij}\right) \tag{24}$$

Optimally, we should not only minimize the target function in (21) but also guarantee that all the sample points are illuminated by their corresponding projectors. To

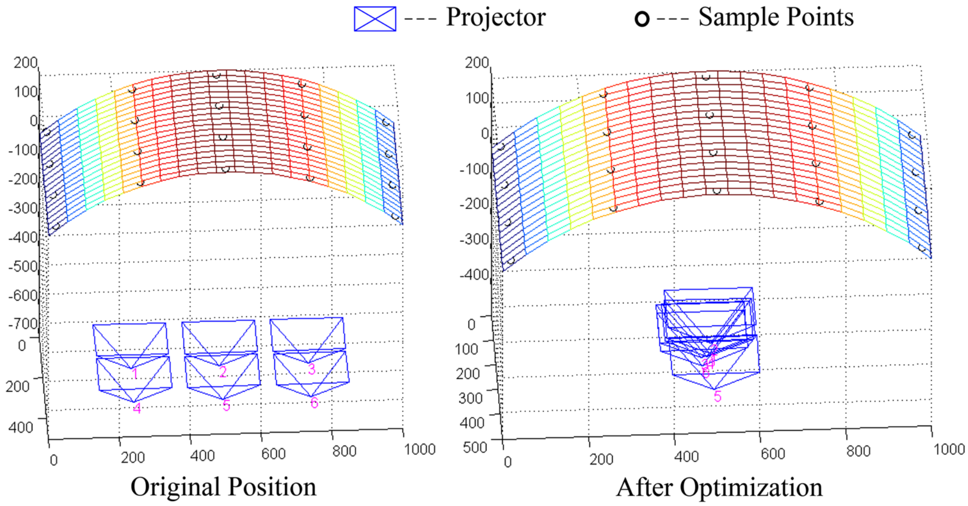


Fig. 27 (Color online) Optimization with different initial values

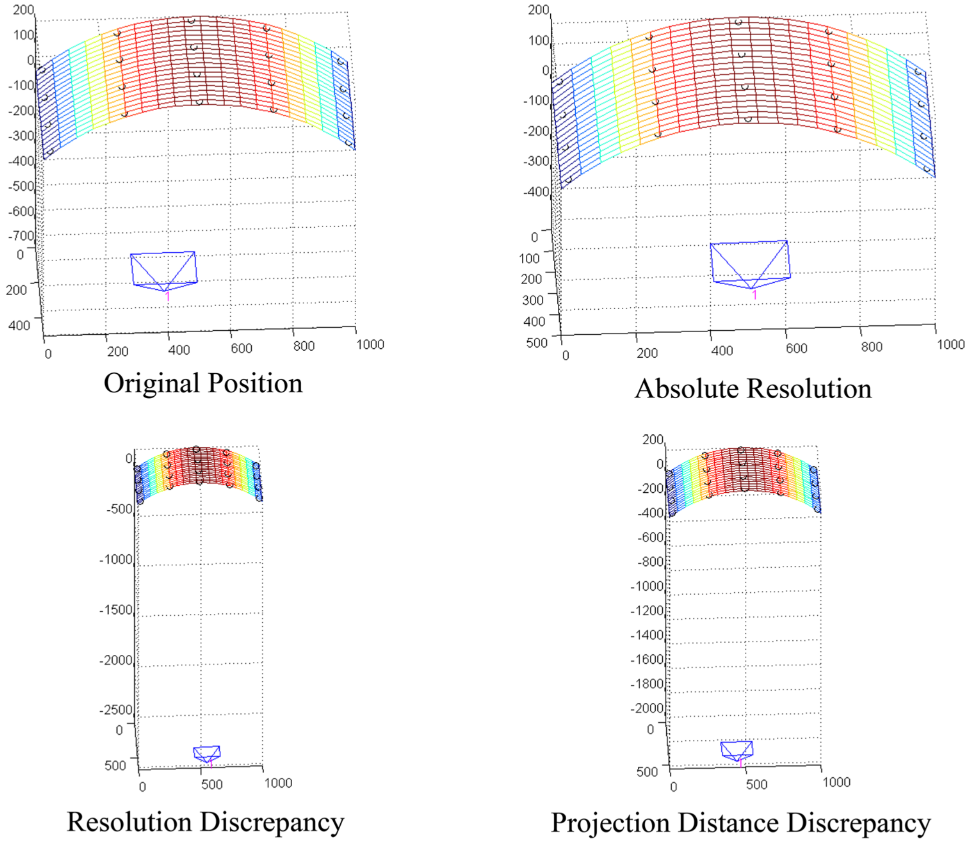
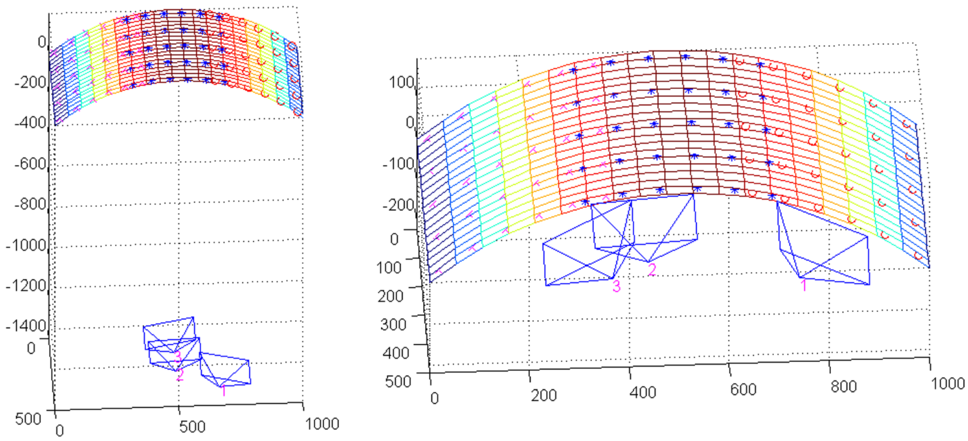


Fig. 28 (Color online) Optimization with different evaluation criteria

do this, we propose the function cover(), which determines whether a sample point $\boldsymbol{X}_{ij}$ is covered by the ith projector. This function can be seen in formula (23), where ε denotes a very small positive number, assigned to be 0.000001 in practice. The integer part of function (24) represents the number of sample points covered by the ith projector. Applying (23), formulas (14), (16), and (19) are replaced by (25), (26) and (27), respectively. Here, L denotes a very large positive number, which we designated to be 999,999 in practice. The output of the newly proposed formulas is significantly large when there is a sample point not illustrated by any projector, and this guarantees that all the sample points are covered when the optimization stops.

$$A'\left(\boldsymbol{x}_i, \boldsymbol{X}_{ij}\right) = \begin{cases} A\left(\boldsymbol{x}_i, \boldsymbol{X}_{ij}\right), & \text{cover}(\boldsymbol{x}_i, \boldsymbol{X}_{ij}) = 1 \\ L, & \text{cover}(\boldsymbol{x}_i, \boldsymbol{X}_{ij}) \neq 1 \end{cases} \qquad (25)$$

Fig. 29 (Color online) Optimization result using three projectors



Original Position

After Optimization

Table 6 Extrinsic parameters before and after optimization

Projector index	Initial position	Initial evaluation $f(\boldsymbol{x}, \boldsymbol{X})$	Optimal position	Optimal evaluation $f(\boldsymbol{x}, \boldsymbol{X})$
No. 1				
r_x	0.050	3.0704	0.0770	0.6341
r_y	−0.250		−0.4879	
r_z	0		−0.0004	
t_x	−990.0		−767.51	
t_y	−440.0		−208.79	
t_z	1171.3		−176.52	
No. 2				
r_x	0.010	3.1734	0.0105	0.6369
r_y	0.050		0.0719	
r_z	0		0.0013	
t_x	−429.47		−438.80	
t_y	−264.11		−246.05	
t_z	1420.7		173.36	
No. 3				
r_x	0.020	3.1544	−0.0213	0.6846
r_y	0.200		0.5088	
r_z	0		−0.0086	
t_x	212.16		−247.26	
t_y	153.78		244.72	
t_z	1468.7		344.44	

$$B'(\boldsymbol{x}_i, \boldsymbol{X}_{ij}) = \begin{cases} B(\boldsymbol{x}_i, \boldsymbol{X}_{ij}), & \text{cover}(\boldsymbol{x}_i, \boldsymbol{X}_{ij}) = 1 \\ L, & \text{cover}(\boldsymbol{x}_i, \boldsymbol{X}_{ij}) \neq 1 \end{cases} \tag{26}$$

$$D'(\boldsymbol{x}_i, \boldsymbol{X}_i) = \begin{cases} D(\boldsymbol{x}_i, \boldsymbol{X}_i), & \text{cover}(\boldsymbol{x}_i, \boldsymbol{X}_{ij}) = 1 \\ L, & \text{cover}(\boldsymbol{x}_i, \boldsymbol{X}_{ij}) \neq 1 \end{cases} \tag{27}$$

Notice that (25), (26) and (27) are not continuously differentiable, and we cannot consistently ensure the optimal attainment of a global minimum. Figure 27 demonstrates the optimization results for one projector with different initial values. We see that the initial position has an obvious influence on the final optimization result. However, an approximate minimum can be determined by selecting the best result from the different initial positions.

Figure 28 shows the optimization results evaluated according to different criteria with the same initial position. As expected, different criteria lead to different results. However, some results are ambivalent to the varying criteria. To minimize the values resulting from (17) and (18), the projector should be as close as possible to the screen, whereas (18) and (19) may have opposite requirements in some cases. To find a compromise to this dilemma, the value of the weight coefficient $\boldsymbol{a}$ must be set according to the actual situation, as shown in (20). In our study, the recommended coefficients are $\boldsymbol{a} = [0, 1, 0.05, 0.5]$, which do not take the honey-point resolution into account (thus $\boldsymbol{a}(1) = 0$).

We employed the optimization using three projectors and exhibit the result in Fig. 29, in which circles with different colors represent sample points from different projectors. To approximate the minimum, we chose $3 \times 5 \times 1$ initial positions, and the optimization result is shown in Table 6. Although the multiple attempts made to approximate the minimum value increase the computation time, when only a few variables need to be optimized, the total run time is still acceptable. We used a common PC with an Intel i5 2.8 GHz CPU to run the optimization. With three projectors having 30 sample points each, the entire computation time was 85 s.

6 Conclusion

This paper presented a fast multi-projector calibration method for a cylindrical screen without using 3D reconstruction. Our method is robust to the error caused by the manufacturing or installation of the screen, and the equipment used is accessible. The calibration procedure does not require manual intervention and can be extended to environments in which the screen or the object to be projected can be parameterized. In addition, this paper also proposed a set of criteria to evaluate the performance of the projection and demonstrated a method for optimizing the projectors' utilization.

Some problems with our method require further study. First, our method does not work for non-parameterized screens because the camera's position is difficult to obtain. The 2D–3D correspondence is easily established in a parameterized environment, but it is difficult to determine in a non-parameterized environment without the help of 3D reconstruction or artificial markers. Developing a method that accurately calculates a camera's position using only the non-parameterized environment's constraints is worth future exploration. Second, our proposed evaluation criteria for the projection are incomplete. There are some important factors, which would increase the complexity of the optimization, we have not considered yet, such as the number of fusion areas and the number of projectors. Constructing a more comprehensive evaluation function and finding a method for determining its minimum are goals for our future work.

Acknowledgments This work is supported by the Key Projects in the National S&T pillar program (No. 2012BAH64F03).

References

1. Raij, A., Pollefeys, M.: Proceedings of 17th International Conference on Pattern Recognition, 2004, p. 14
2. Raskar, R., van Baar, J., Beardsley, P., Willwacher, T., Rao, S., Forlines, C.: Proceedings of SIGGRAPH Courses, 2006, p. 7
3. Yang, R., Gotz, D., Hensley, J., Towles, H., Brown, M.S.: Proceedings of Visualization, 2001, p. 167
4. Raij, A., Gill, G., Majumder, A., Towles, H., Fuchs, H.: Proceedings of International Workshop on Projector-Camera Systems, 2003, p. 203
5. Bhasker, E., Juang, R., Majumder, A.: IEEE Trans. Vis. Comput. Graph. **13**, 1368 (2007)
6. Chen, H., Sukthankar, R., Wallace, G., Li, K.: Proceedings of Visualization, 2002, p. 339
7. Bhasker, E.S., Sinha, P., Majumder, A.: IEEE Trans. Vis. Comput. Graph. **12**, 1101 (2006)
8. Raskar, R., Brown, M.S., Ruigang, Y., Wei-Chao, C., Welch, G., Towles, H., Scales, B., Fuchs, H.: Proceedings of Visualization, 1999, p. 161
9. Aliaga, D.G.: Proceedings of the Symposium Interactive 3D Graphics and Games, 2008, p. 53
10. Aliaga, D.G., Xu, Y.: Proceedings of Conference on Computer Vision and Pattern Recognition, 2008, p. 1
11. Raskar, R., van Baar, J., Willwacher, T., Rao, S.: Comput. Graph. Forum **23**, 451 (2004)
12. Yang, R., Welch, G.: Presented at Proceedings of 9th International Conference on Computer Graphics, Visualization and Computer Vision, 2001
13. Cotting, D., Naef, M., Gross, M., Fuchs, H.: Proceedings of International Symposium on Mixed and Augmented Reality, 2004, p. 100
14. Cotting, D., Ziegler, R., Gross, M., Fuchs, H.: Comput. Graph. Forum **24**, 705 (2005)
15. Johnson, T., Fuchs, H.: Proceedings of Conference on Computer Vision and Pattern Recognition, 2007, p. 1
16. Johnson, T., Welch, G., Fuchs, H., La Force, E., Towles, H.: Proceedings of Conference on Virtual Reality, 2009, p. 35
17. Zhou, J., Wang, L., Akbarzadeh, A., Yang, R.: Proceedings of International Workshop on Projector-Camera Systems, 2008, p. 1
18. Langlotz, T., Zollmann, S., Bimber, O.: J. Virtual Real. Broadcast. **4**, 6 (2007)
19. Newcombe, R.A., Izadi, S., Hilliges, O., Molyneaux, D., Kim, D., Davison, A.J., Kohi, P., Shotton, J., Hodges, S., Fitzgibbon, A.: Proceedings of International Symposium on Mixed and Augmented Reality, 2011, p. 127
20. Tan, M., Xu, W., Weng, D.: Proceedings of International Conference on Computer-Aided Design and Computer Graphics, 2013, p. 425
21. Brown, M.S., Seales, W.B.: Proceedings of Conference on Computer Graphics and Applications, 2002, p. 194
22. Brown, M., Majumder, A., Ruigang, Y.: IEEE Trans. Vis. Comput. Graph. **11**, 193 (2005)
23. Harville, M., Culbertson, B., Sobel, I., Gelb, D., Fitzhugh, A., Tanguay, D.: Proceedings of Conference on Computer Vision and Pattern Recognition Workshops, 2006, p. 5
24. Sun, W., Sobel, I., Culbertson, B., Gelb, D., Robinson, I.: Proceedings of International Workshop on Projector-Camera Systems, 2008, p. 1
25. Sajadi, B., Majumder, A.: IEEE Trans. Vis. Comput. Graph. **15**, 1307 (2009)
26. Sajadi, B., Majumder, A.: Proceedings of Conference on Virtual Reality, 2010, p. 155
27. Sajadi, B., Majumder, A.: IEEE Trans. Vis. Comput. Graph. **17**, 1209 (2011)
28. Sajadi, B., Gopi, M., Majumder, A.: ACM Trans. Graph. **31**, 1 (2012)
29. Sajadi, B., Duy, Q., Ihler, A.H., Gopi, M., Majumder, A.: Proceedings of International Conference on Computational Photography, 2013, p. 1
30. Law, A.J., Aliaga, D.G., Majumder, A.: IEEE Trans. Vis. Comput. Graph. **16**, 1633 (2010)
31. Aliaga, D.G., Yeung, Y.H., Law, A., Sajadi, B., Majumder, A.: ACM Trans. Graph. **31**, 1 (2012)
32. Hartley, R., Zisserman, A.: Multiple View Geometry in Computer Vision. Cambridge University Press, Cambridge (2003)
33. Zhang, Z.: Proceedings of International Conference on Computer Vision, 1999, p. 666
34. Pujol, J.: Geophysics **72**, W1 (2007)
35. Majumder, A., Stevens, R.: ACM Trans. Graph. **24**, 118 (2005)