Automatic Calibration of a Multiple-Projector Spherical Fish Tank VR Display

Qian Zhou Gregor Miller Kai Wu Daniela Correa Sidney Fels
Department of Electrical and Computer Engineering, University of British Columbia, BC, Canada

Abstract

We describe a novel automatic calibration method using a single camera for a multiple-projector spherical Fish Tank Virtual Reality (FTVR) display. Modeling the projector as an inverse camera, we estimate the intrinsic and extrinsic projector parameters automatically using a set of projected images on the spherical screen. A calibrated camera is placed beneath to observe partially visible projected patterns. Using the correspondence between the observed pattern and the projected pattern, we reconstruct the shape of the spherical display and finally recover the 3D position of each projected pixel on the display. Additionally we present a practical calibration evaluation method that estimates on-surface accuracy using the single camera. We use point mismatch as a metric to describe misalignment and line mismatch to describe distortion. We demonstrate our automatic approach can achieve an on-surface point mismatch less than $1mm$ *and line mismatch less than* 1° *on a* $30cm$ *diameter spherical screen. Taken together, our calibration approach and evaluation method are automatic and accurate for a desktop spherical FTVR and can be applied to other multiple-projector displays with curved screens.*

1. Introduction

Spherical displays are useful for many applications, such as planetariums and virtual snowglobes, since they provide an unobstructed view from all angles. This property is ideal for creating FTVR visualizations. Using the spherical nature of the display, viewers can move around the display with head-tracking and observe a perspective-corrected scene on the spherical screen. Providing high resolution, uniformly spaced pixel imagery on the spherical screen is important for constructing spherical FTVR. One approach is to tile multiple projectors on the spherical screen to increase the resolution and make the system scalable. The challenge for this lies in the stitching and blending of images from different projectors to create seamless imagery. This requires geometric and photometric calibration of the multiple-projector system.

Geometric calibration of a multiple-projector system typically uses a camera to record correspondences from the known pattern to the observed pattern. Systems with a planar screen take advantage of 2D homography transformations to linearly establish the projector-to-display correspondences. While there has been substantial previous work on calibration techniques for planar screens [5, 6, 17, 19], automatic calibration for a curved screen has not received as much attention. A few have investigated approximate correspondence through 2D parameterization either with a linear approximation [28] or physical markers on curved screens [12]. Others have attempted to recover the 3D geometry of the display to establish the mapping [1, 20, 21, 31], but this usually requires a substantial amount of manual interaction.

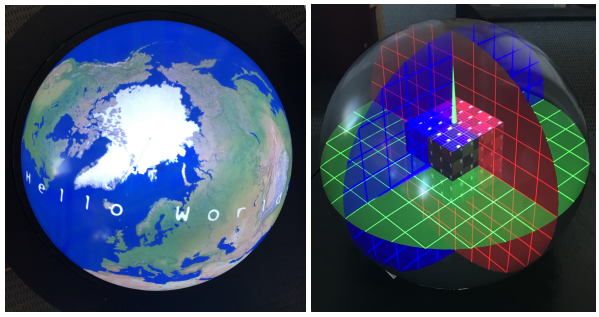

Figure 1. Our goal is to calibrate a multiple-projector spherical display with a single camera to allow for a seamlessly blended image. For spherical rendering (left), blending is the most important issue. We target the application of spherical FTVR (right) which uses single-person perspective-corrected rendering and requires a more accurate calibration method to provide a higher quality experience to the user.

In addition, previous work has primarily targeted large scale immersive displays like domes to create a sense of immersion. These displays consist of multiple front-projecting projectors and cameras with pan-tilt units to cover the entire display. For relatively small scale desktop FTVR, projectors are used in a rear-projection configuration through a small projection hole at the bottom of the spherical screen [27, 3, 31]. Existing calibration methods will not work since the camera's view is mostly blocked by the edge of the projection hole. Applications in FTVR require accurate geometry registration to support perspective-corrected viewpoints and subtle interactions in real-time. Taken together,

the configuration of the spherical FTVR system makes the calibration challenging in the following aspects:

Scale The spherical FTVR is a desktop system with a small-scale screen compared to large-scale immersive displays. Space is quite limited for the camera and projectors beneath the screen.

Visibility The camera view will be occluded by the edge of the small projection hole at the bottom.

Perspective-correction The calibration result will support perspective-corrected imagery in real-time.

This paper intends to provide an automatic calibration method that meets these requirements and supports applications in a desktop spherical FTVR. We start with a semi-automatic approach that solves these problems. This approach begins with a pre-calibration step that outputs the intrinsic parameters of the camera and the projectors. Then each projector is paired with the same camera to form a stereo pair. For each pair, a pattern projected onto the display is captured by the camera to recover extrinsic parameters via the essential matrix. Using intrinsic and extrinsic parameters, we triangulate projected features and compute the sphere's pose via Weighted Least Squares. The parameters of the sphere (pose) and the camera/projector pairs (intrinsics and extrinsics) are further refined via a nonlinear optimization. Finally we recover the 3D position of each pixel on the display surface via sphere-ray intersection for each pixel per projector.

Up to this point this approach is semi-automatic since it requires additional work to calibrate the intrinsics of the projectors. We provide further improvement of this approach by avoiding the separate calibration of projectors: by estimating the fundamental matrix using the projected pattern on the sphere, we recover the absolute dual quadric for each projector, which is then used to recover the intrinsic parameters of projectors.

We also introduce a practical evaluation method using the camera to estimate the accuracy of our approach, using on-screen metrics instead of reprojection error. We can measure the misalignment by matching points and lines between the observed pattern and the expected pattern.

We find that our automatic approach can achieve less than $1mm$ on-surface point error and less than 1° line error using a $30cm$ diameter spherical screen. The result is compared with other work [21, 31] using RMS re-projection error. With a typical $40cm$ viewing distance, this yields no more than 0.14° eye angular error from the viewpoint as in [31] calculated using the viewpoint position and the on-surface point error, making it appropriate for applications of FTVR.

Our contribution includes an automatic calibration method that solves practical problems and supports various applications in spherical FTVR, and an accompanying evaluation method that automatically estimates on-surface calibration error. This method can support displays with various scales even when the camera cannot see the entire display. Our approach is automatic, accurate and practical for multiple-projector spherical FTVR systems, and can be applied to other multiple-projector systems.

2. Related work

We present a contextual overview of the field of fish tank virtual reality, spherical displays and multiple-projector calibration as it relates to our proposed contribution.

2.1. FTVR and Spherical Displays

Fish Tank Virtual Reality [2] is a type of 3D head-tracked display, providing multiple dynamic depth cues like motion parallax to improve user's 3D perception of the 3D virtual scene. Various work [7, 25, 13] have been conducted using this technique. As one of the well-known systems, CAVE [7] extends the traditional FTVR by projecting on multiple screens to form a geometric shape of display. Among different shapes of displays, spherical display has a promising shape as it has no seam between screens. A number of spherical systems have been proposed with different implementations. As an early work, Perspecta Spatial System [10] from Actuality System utilizes an embedded projector to project images onto a rotating flat screen. Despite of the volume-filling imagery it produces, the system is expensive to build and not scalable for system scale and resolution. An alternative approach uses one projector to rear-project directly on a spherical screen. There have been multiple systems using this technique [4, 3, 27, 29]. Being simple and effective, these systems offer limited resolution and lack of scalability. Recent work [27, 31] extend the idea by using multiple pico-projectors to increase the resolution, making the system scalable to spherical screens with different sizes.

Drawing inspiration from previous work, we use the same approach to build our spherical FTVR system. However, the rear-projection through a small projection hole at the bottom of the spherical screen becomes a challenge for the system calibration.

2.2. Multiple-Projector System Calibration

Despite substantial work on calibration techniques for planar screens [19, 17, 6, 5], the non-linearity of curved screen is a challenge for multiple-projector system calibration. An early work [20] presents a calibration approach of non-planar surface using a stereo camera pair to recover intrinsic and extrinsic parameters and reconstruct the non-planar surface. This approach has been further improved by focusing a subset of curved screens called a quadric screen to recover a quadric transformation [21]. Another approach [12] uses physical checkerboard patterns attached on the curved display to provide the camera a composition of 2D-mesh-based mappings. Their approach aims at a class

of curved surface that can be bent or folded from a plane. However, the use of a physical marker on the display causes limitation in the application space.

Recently Majumder et al. proposed a series of automatic calibration approaches for non-planar screens [22, 24, 23]. Using an uncalibrated camera, they compute a rational Bezier patch for a dome screen. This approach works for various shapes such as extruded surfaces, swept surfaces, dome surfaces and CAVE-like surfaces. The camera is mounted on a pan-tilt unit to cover the entire display. Their approaches aim at large-scale immersive displays.

More recently, a calibration library [28] has been developed to automatically calibrate multiple-projector system with different shapes of screens. They use fixed warping to register imagery from projectors to avoid geometry reconstruction of the display. For a spherical shape, they make an implicit linear assumption by using a homography transformation between camera and projector. This causes observable misalignment and distortions in overlapping areas.

In recent work [31], an approach with reconstruction of pixels on a curved screen has been proposed. They use a physical checkerboard pattern attached to an additional planar surface to calibrate intrinsic and extrinsic parameters of the camera and projectors. Using these parameters, they recover the 3D position of pixels on the curved screen. Their approach requires manual calibration of both the intrinsic and extrinsic parameters of projectors, which can be time-consuming and difficult to maintain.

Our work is close to these approaches that intend to create seamless imagery on a curved screen. However, our approach is designed to support applications in a desktop spherical or curved FTVR display, which inherently has a few distinctions from other displays due to scale, visibility and perspective-correction as identified in the previous section. The perspective-correction requirement implies a full reconstruction approach is needed. Fixed warping with linear assumptions will cause distortions and discontinuity across projectors [28] which makes them inappropriate for high quality visualization and interaction. The quadric transformation [21] can handle with the distortion of the screen shape, but the transformation has to be estimated and updated for each frame as the viewer is moving, making it unsuitable for a real-time perspective-corrected application. The visibility problem makes most patch-based and mesh-based approaches [1, 22, 23] inapplicable in our system. Finally the limited space constraints suggests a single camera approach is preferred.

While there are some work [20, 31] that meet the above requirements, they are not easy to use due to the tedious manual work and effort to calibrate the system. In addition, it is time consuming to re-calibrate if there is a disturbance in the calibrated system.

As a result, we propose an automatic approach that determines intrinsic and extrinsic parameters directly using projected patterns on the curved screen. Our approach aims at relatively small-scale desktop spherical displays (i.e $< 1m$ diameter with projection hole cut at the bottom) compared to other large-scale immersive displays ($> 1m$); however, it generalizes to these. In addition, our approach does not require the camera to see the entire display surface. In fact, more than one-third of the display surface is invisible to the camera due to the small projection hole.

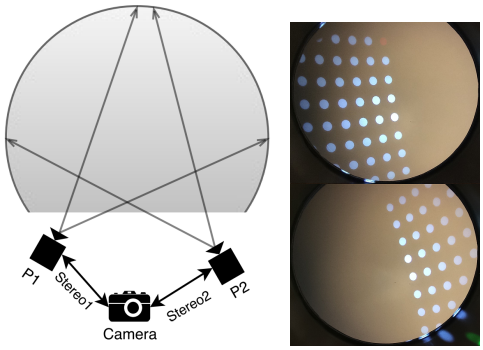

Figure 2. Left: Multiple-projector spherical display layout, showing overlap and approximate layout with respect to the camera. Right: Projected blob patterns on the spherical display surface observed by the camera, for each of two projectors.

3. Calibration of a Multiple-Projector Spherical FTVR System

Our spherical FTVR system consists of multiple projectors and a spherical screen. As shown in Figure 2(Left), the projectors work in rear-projection mode through a small projection hole at the bottom of the spherical screen. Perspective-corrected images are generated on the sphere based on the viewer's position.

The calibration approach we propose is presented in Figure 3. We start with a semi-automatic approach that supports applications in a desktop spherical FTVR. Then we enhance the method to make it automatic by avoiding the separate calibration of projectors. This approach is realized by recovering the absolute dual quadric for each projector using the fundamental matrix. We also explore the possibility of using uncalibrated camera in Section 4.

3.1. Semi-Automatic Calibration Approach

Pre-Calibration The camera and the projectors are pre-calibrated to determine the intrinsic parameters. The camera is calibrated using a checkerboard-based calibration approach [30] and each projector is calibrated using a plane-based calibration approach [9] with the help of the calibrated camera. After this step, the intrinsic parameters of camera and projectors are recovered.

Camera-Projector Calibration Each projector P_i and

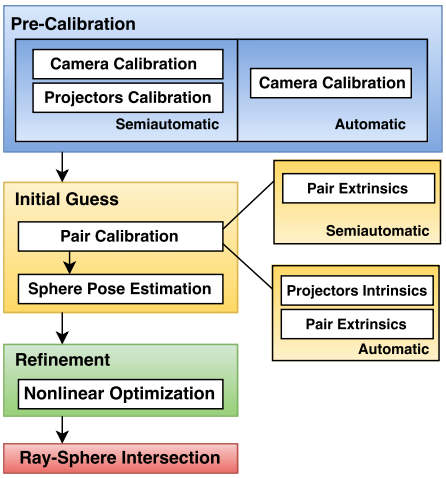

Figure 3. Calibration pipeline of Semi-automatic and Automatic approach for a desktop spherical FTVR. For semi-automatic approach, projectors and camera are calibrated in the pre-calibration step. For automatic approach, only camera is calibrated in the pre-calibration step.

the camera C are paired as a stereo pair S_i as shown in Figure 2(a). Although we have the pre-calibrated intrinsics for each pair, the extrinsics are still unknown. In this step, we project blob patterns onto the spherical screen, detect them as blob features in the camera, and record feature correspondences for each pair as shown in Figure 2(b).

Using these correspondences, the essential matrix E_i can be recovered since we know the intrinsics for the pair S_i. Then we extract the rotation R_i and translation T_i from the essential matrix E_i using SVD. Although there are four possible solutions for the calibrated reconstruction from E_i, only one solution is the correct reconstruction that has 3D points in front of both C and P_i. Thus testing with a single point to determine if it is in front of both C and P_i is sufficient to select the correct solution [11] for S_i. The camera center is chosen to be the origin for all pairs to put them in the same coordinate system.

However, the translation vector T_i is recovered up-to-scale such that the coordinates between pairs are still up-to-scale. To solve this problem, we choose one pair S_0 as the "standard" pair that has translation with norm of 1, then we estimate scale factors for other pairs with respect to S_0. These scale factors are computed using the knowledge that points between pairs are on the same sphere. We first triangulate blob features for each pair using the up-to-scale extrinsics. Then we fit a sphere for each pair, and compute each scale factor using Linear Least Squares based on the recovered sphere poses from S_0 and S_i. After this step, each pair has extrinsics in the same camera-centered coordinates.

Sphere Pose Estimation So far we have intrinsics, extrinsics and 3D points in the camera-centered coordinate system. The sphere pose can be recovered by fitting a sphere with these 3D points using a Weighted Linear Least Squares [26]. The weighting comes from the re-projection error in the triangulation step so that a large re-projection error results in a small weight in determining the sphere pose.

From these steps we have calculated a full set of parameters that affects this system: intrinsics, extrinsics and sphere pose. Unlike iterative methods, the above approach used to determine these parameters are linear and can always generate a result. However, some parameters like extrinsics and sphere pose are roughly estimated. Thus, using these parameters to compute a 3D position for each pixel on the display sphere may cause significant errors. So we use these results as an initial guess for a nonlinear optimization to refine them.

Nonlinear Optimization Parameters are refined using a non-linear optimization with the previous result serving as an initial guess. We now describe the error function we use for the non-linear optimization.

Assume we have 1 camera and N projectors. Camera parameters $\vec{p}_c$ have 9 degree of freedom (DOF): 4 for the focal length and the principle point; 5 for lens distortion [8]. Each projector has parameters $\vec{p}_{p_i}$ with 10 DOF: 4 for the focal length and the principle point; 3 for rotation and 3 for translation. Sphere parameters $\vec{p}_s$ have 4 DOF: 3 for the center position and 1 for radius.

For each pixel $\vec{x}_{p_{ij}}$ in projector P_i, a ray is back-projected and intersects with the sphere at the point $\vec{X}_{ij}$. The back-projection and ray-sphere intersection can be expressed as a function f based on variables $\vec{p}_{p_i}$ and $\vec{p}_s$:

$$\vec{X}_{ij} = f(\vec{x}_{p_{ij}}; \vec{p}_{p_i}, \vec{p}_s) \tag{1}$$

Then the 3D point $\vec{X}_{ij}$ is observed by the camera at pixel $\vec{x}_{c_{ij}}$ on the image plane. This can be expressed as a function g based on variables $\vec{p}_s$ and $\vec{p}_c$:

$$\vec{x}_{c_{ij}} = g(\vec{X}_{ij}; \vec{p}_c, \vec{p}_s) \tag{2}$$

Substituting equation (1) into equation (2), we get a function F that models this whole process:

$$\begin{aligned} \vec{x}_{c_{ij}} &= g(f(\vec{x}_{p_{ij}}; \vec{p}_{p_i}, \vec{p}_s); \vec{p}_c, \vec{p}_s) \\ &= F(\vec{x}_{p_{ij}}; \vec{p}_c, \vec{p}_s, \vec{p}_{p_i}) \end{aligned} \tag{3}$$

Since we know exactly which pixel $\vec{x}_{p_{ij}}$ has been projected from projector P_i, the error function is formulated as the re-projection error in the camera:

$$E = \sum_i \sum_j d(x_{c_{ij}}, F(x_{p_j}; \hat{p}_c, \hat{p}_s, \hat{p}_{p_i}))^2, \tag{4}$$

where $x_{c_{ij}}$ is the detected point from camera and $F(x_{p_j}; \hat{p}_c, \hat{p}_s, \hat{p}_{p_i})$ is the estimated point based on parameters $\hat{p}_c, \hat{p}_s$ and $\hat{p}_{p_i}$. For a system with N projectors, there

are $13 + 10N$ variables to refine. We use a gradient descent method based on trust regions [14] to solve this non-linear least square problem. As indicated before, the solver is initialized using our previous results.

Ray-Sphere Intersection After refining the parameters, we compute the 3D position for each pixel on the display via ray-sphere intersection with rays coming from each projector. The geometric result for each pixel is stored in a look-up table.

To summarize, our semi-automatic method uses projector and camera intrinsics as prior knowledge to estimate the extrinsics and display pose parameters. These estimations are then used as an initial guess in a nonlinear optimization to refine the result. In this process, the camera and projectors are calibrated once and can be used afterwards. If there is disturbance on these devices, the system can be re-calibrated by projecting blob patterns and using the non-linear optimization. However, as described, this semi-automatic approach still requires manual work in the pre-calibration step to calibrate the intrinsics of the projectors. We now describe our automatic approach that recovers projector intrinsics together with the extrinsics directly from projected patterns on the spherical display.

3.2. Automatic Calibration Approach

In this section we revisit the calibration pipeline in Figure 3 and present techniques to make the workflow automatic.

Pre-Calibration As illustrated in Figure 3, only the camera's intrinsics are determined for use as prior information.

Pair Calibration In this step, we first determine the internal projector parameters directly from the uncalibrated images. This is essentially an auto-calibration problem. For each pair S_i, we project the same blob pattern as in the semi-automatic approach. The fundamental matrix F_i is recovered using these correspondences. We obtain a projective reconstruction for each pair by choosing the projection matrices as $P_0 = [I|\mathbf{0}]$ and $P_i = [[e'_i]_\times F_i|e'_i]$, where e'_i is the epipole in the projector view, P_0 for the camera and P_i for the projector. The reconstruction $\{P_i, X_i\}$ for each projector is up to a projective transformation. Our goal is to find the projective transformation H_i such that $\{P_i H_i, H_i^{-1} X_i\}$ is a metric reconstruction which is only up to a similarity transformation. As in [11], H_i can be expressed in the form:

$$H_i = \begin{pmatrix} K_c & \mathbf{0} \\ v^T & 1 \end{pmatrix} \tag{5}$$

where K_c is the intrinsic matrix of the camera. Since K_c is known, the only unknown is the vector v^T with 3 DOF. Since $v^T = -p^T K_c$ where p^T are the coordinates of the plane at infinity, this is essentially a problem to recover the plane at infinity, p^T, for each pair.

With further information such as the vanishing points, p^T can be recovered. However, vanishing points or parallel lines can hardly be observed in our case since the screen is curved. In most projectors today the principal point is vertically offset at the bottom-center of the image with zero-skew between axes so that the projection is not occluded by the table [15, 18]. We use these assumptions[1] to provide additional constraints needed to solve for p^T.

Encoding the infinity plane p^T in a concise fashion, the absolute dual quadric Q^*_∞ for each projector under the transformation H_i can be expressed as:

$$Q^*_\infty = H_i \begin{pmatrix} I_{3\times3} & \mathbf{0} \\ \mathbf{0^T} & 0 \end{pmatrix} H_i^T = \begin{pmatrix} \omega_c^* & -\omega_c^* p \\ -p^T \omega_c^* & p^T \omega_c^* p \end{pmatrix} \tag{6}$$

where $\omega_c^* = K_c K_c^T$ is the dual image of absolute conic (DIAC) of the camera [11].

Each Q^*_∞ is related to the projector intrinsics in the form of:

$$\omega^*_{p_i} = P_i Q^*_\infty P_i^T \tag{7}$$

where $\omega^*_{p_i} = K_{p_i} K_{p_i}^T$ is the DIAC of the projector and P_i is the reconstructed projection matrix of the projector using the fundamental matrix.

In this case, constraints on projector intrinsics can be transferred to constraints on Q^*_∞. Our constraints are the known principal points and zero-skew. This results in three linear constraints for each projector on Q^*_∞:

$$\begin{aligned} \omega^*_{p_i}(1,3) &= pp_x * \omega^*_{p_i}(3,3) \\ \omega^*_{p_i}(2,3) &= pp_y * \omega^*_{p_i}(3,3) \\ \omega^*_{p_i}(1,2) &= pp_x * pp_y * \omega^*_{p_i}(3,3) \end{aligned} \tag{8}$$

where pp_x and pp_y are known principle points. This provides a direct solution for p^T. Once we recover p^T, we can get the DIAC and hence the intrinsics of each projector.

The rest of the method is the same as our semi-automatic approach: estimate the extrinsics using these intrinsics, then triangulate and fit a sphere to find the sphere pose, followed by a nonlinear optimization to refine these parameters.

By doing these steps, we avoid the manual work to calibrate the projectors, and the whole calibration can be implemented by projecting blob patterns and detecting projected features automatically.

4. Evaluation

While many previous papers have proposed calibration methods for curved screens, none of them have provided

[1] For cases in which the principal point is not at the bottom-center, a general camera matrix is necessary. This will be covered by the semi-automatic method in which projector intrinsics are pre-calibrated.

an evaluation for on-surface accuracy. Raskar et al. [21] evaluated their methods using RMS re-projection error, others use simulation to estimate percentage errors of the estimated camera and display parameters [23, 24, 22]. Recent work demonstrated analytically how each source of error can affect the display error [31], but did not provide an estimate of on-screen error accumulated by different error sources. Here, we illustrate an evaluation method that estimates on-surface accuracy empirically. We use this to evaluate our calibration approach and compare variations of our approach. We also include comparison with other work [21, 31] based on RMS re-projection error.

4.1. Metrics for Multiple-Projector Spherical FTVR Systems

We use three metrics to evaluate the result of calibration: *global point error*, *local point error* and *line error*.

Global point error describes the overall misalignment of the display. We define the global point error to be the displacement between the expected and the actual position of a projected point. There are two units that can be applied to point error: one is the RMS re-projection error in pixels; the other is the arc length in, for example, millimeters, directly on the spherical screen. Since the arc length varies with the size of the spherical screen, we use radians of the sphere to describe on-surface misalignment.

Local point error describes the local misalignment between adjacent projectors. The effect of local point error is usually observed as a ghost effect in the overlapping projectors area. We define it as the displacement between a point from one projector and the same point from its adjacent projector. Similar to global point error, we use both RMS reprojection error in pixel and on-surface error in radians to describe local point error.

Line error is used to describe the distortion of the overall display. The effect of line error is observed as distortions (i.e. straight lines appear to be curved). This is important for a FTVR system, since the distortion will cause perceptual discrepancies based on the viewpoint. We define the line error to be the angular difference between the expected lines and the observed lines. Ideally the projected line is collinear with the expected line.

4.2. Error Measurement

To measure error, a camera is introduced into the system that observes projected patterns on the spherical screen. We utilize the same camera used for calibrating the spherical display since the camera has already been calibrated with known intrinsic and pose parameters for the display.

The camera is regarded as a virtual viewer with known pose (relative to the display) that observes certain patterns. Suppose a grid pattern is expected from the viewpoint of the camera; then the color value for each pixel in the projector

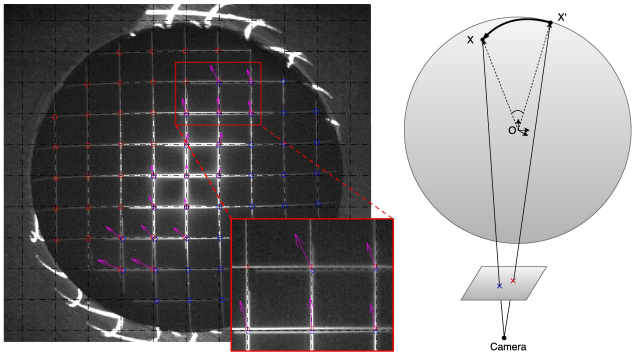

Figure 4. Left: Expected grid pattern (black dash lines) and observed projected pattern (solid white lines); purple arrows illustrate local point error in overlapping area. Right: Back-projection of pixel error to estimate on-surface error

can be determined by projecting its associated 3D position onto the image plane of the camera. Ideally, the camera will observe a grid pattern regardless of the curvature of spherical screen. We use the expected grid pattern as ground truth and compare it with the actual observed pattern. The point error is computed based on the locations of crossing features in the grid pattern, while line error is computed based on the angle between the projected line segment and the expected line segment at each crossing feature.

Figure 4 shows the global point error with dashed black lines as ground truth and solid white lines as actual observations. Crosses and line segments in the image are detected using template matching. The global point error is computed based on the displacement in pixels between black and white crosses. Shown as arrows, the local point error is computed based on the displacement in pixels between crosses from adjacent projectors in the overlapping area. Finally, the line error is computed based on the slope difference of observed white lines and the expected black lines.

Point errors are evaluated in the form of RMS reprojection error in pixels. Despite being simple and effective, the computed pixel error does not directly predict on-surface registration error [21]. To acquire an estimate of on-surface error, we back-project the displacement in the image onto the spherical screen and compute the arc length on the screen as shown in Figure 4. As the on-surface error varies with the size of display, the arc length is computed in radians.

4.3. Implementation

We implement the spherical FTVR system with two pico-projectors and an acrylic spherical screen. The spherical screen has a diameter of 30 cm and a projection hole of 21 cm diameter. The projectors are ASUS P2B with resolution of 1280 x 800. A host computer with a NVDIA Quadro K5200 graphic card sends rendering content to projectors. The rendering contents are created using OpenGL. We em-

Approach	Projectors	Global point error		Line error	Local point error	
		pixel	radian	degree	pixel	radian
Semi-automatic	Projector 1	3.9093	0.0107	1.2610	5.6768	0.0167
	Projector 2	5.7653	0.0149	1.4643		
Semi-automatic with NLO	Projector 1	1.2556	0.0036	0.8024	1.7165	0.0052
	Projector 2	1.4586	0.0053	0.8465		
Automatic	Projector 1	7.1373	0.0241	1.5683	15.7708	0.0453
	Projector 2	10.5067	0.0294	2.0423		
Automatic with NLO	Projector 1	1.6159	0.0051	0.8268	1.8222	0.0056
	Projector 2	1.7965	0.0064	0.9298		

Table 1. Comparing results of our Semi-Automatic and Automatic approach before and after Non Linear Optimization (NLO) in terms of global point error, local point error and line error. Measurements come from our implemented system with two pico-projectors. On-surface error is expressed in radians on the sphere.

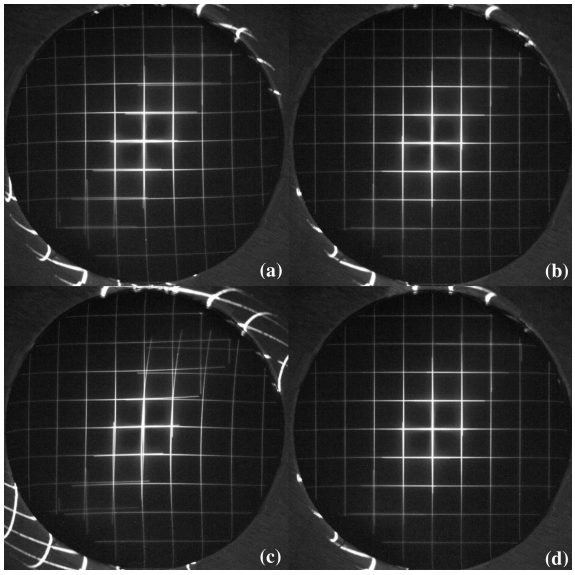

Figure 5. Observed grid pattern by camera using the semi-automatic approach (a) before nonlinear optimization and (b) after nonlinear optimization. Observed grid pattern using the automatic approach (c) before nonlinear optimization and (d) after nonlinear optimization

ploy head-tracking to generate perspective-corrected views for applications in FTVR. The viewer is tracked using Polhemus Fastrak [16]. A two-pass rendering [20] approach based on head position is chosen due to the non-linearity of the curved screen.

The calibration approaches are implemented in C++ using OpenCV and the nonlinear optimization is implemented in Matlab using its optimization Toolbox. The semi-automatic approach takes about 20 mins and automatic approach takes about 10 mins to calibrate a two-projector system. Our approach supports view-dependent and view-independent applications in spherical FTVR. Figure 1(a) shows a view-independent application after calibrated and blended. The earth image is stitched from different projectors seamlessly. Blending is implemented using an alpha mask technique [20]. Figure 1(b) shows a view-dependent application. The viewer is tracked and presented a perspective-corrected images using our calibration result.

Semi+NLO	Auto+NLO	Raskar03	Zhou16
1.3571	1.7062	0.8625	2.0640

Table 2. Comparing results with existing approaches based on RMS re-projection error. The proposed semi-automatic approach (semi+NLO) and automatic approach with Nonlinear Optimization (auto+NLO) is compared with Raskar03[21] and Zhou16[31].

4.4. Results

We compare the results of the semi-automatic and automatic approaches in Table 1. For both, we see substantial improvement after nonlinear optimization. For the semi-automatic approach, the initial guess is computed using a pre-calibrated projector with more accurate intrinsic parameters; hence it has much less error than the initial guess of the automatic approach. This also explains a slightly smaller error after refinement compared with the automatic approach, although the difference is very small. For the automatic approach, although the error before optimization is quite large, the results are largely improved by nonlinear optimization. As a result, the final result of the automatic approach is close to the one from the semi-automatic approach but required no manual interaction.

Figure 6 shows the comparison between algorithms with respect to on-surface error. We convert the radian error in Table 1 to arc length in millimeters on a $30cm$ diameter sphere. Both the semi-automatic and automatic approach can achieve accurate registration: the on-surface point error is less than $1mm$ and line distortion is less than 1°. Our approaches are appropriate for spherical FTVR where viewers are usually $40cm$ to $60cm$ away from the screen, with an eye angular error to be no more than 0.14° [31].

The result is also compared with existing work based on

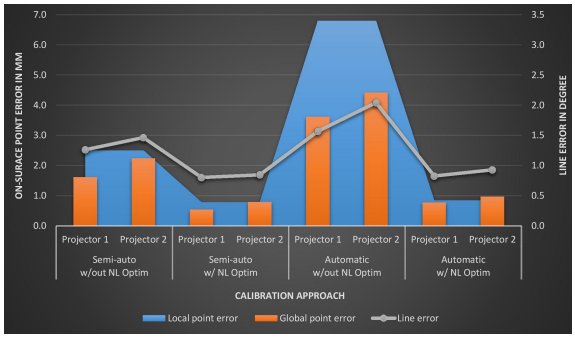

Figure 6. Comparing results of Semi-automatic and Automatic approach before and after nonlinear optimization. On-surface error is estimated as arc distance in millimeters on a 30cm diameter sphere.

RMS re-projection error in pixel. As shown in Table 2, the proposed semi-automatic approach (semi+NLO) and automatic approach with Nonlinear Optimization (auto+NLO) is compared with Raskar03 [21] (mean of four projectors) and Zhou16 [31] (mean of two projectors). We achieve better results than [31] but not as good as [21], however all approaches are within the same scale. Comparisons are made in each group's own setup.

4.5. Discussion

In this section, we discuss factors that are related with these calibration approaches.

Scalability: Though we evaluate the approach with only two projectors, the result generalizes to more than two projectors. To add new projectors into the system requires adding another stereo pair directly registered to the world coordinate system for each projector. Since each projector is registered independently, this does not cause cascading error as the number of projectors increases.

Uniform error: Among all the stereo pairs, there will always be a "standard" pair that has theoretically minimum error. As shown in Figure 6, the error in Projector 1 is always smaller than Projector 2, regardless of the metric used. This is due to the scale ambiguity of projector extrinsic parameters. When recovering extrinsic parameters for each pair, we choose one pair as the "standard" pair that has a translation with norm 1; then we estimate scale factors for other pairs with respect to the "standard" pair. The estimation of scale, however, is not accurate since we use the estimated sphere pose in each pair to get a linear solution of the scale as the initial guess, while those estimated sphere poses already contain errors from the previous step. For future work, we suggest an improved method will be to estimate the sphere pose together with all scale factors using a nonlinear least square solver.

Robustness: While the automatic approach can generate results very close to the semi-automatic one, the former is more sensitive to noise. In our system, if the fundamental matrix has not been estimated correctly due to incorrect feature correspondences or bad lighting, the automatic approach is more likely to fail than the Semi-automatic approach. So a trade-off has been made between robustness and priors, which should influence the decision on which to use.

Uncalibrated camera: It is possible to calibrate the system without calibrating the camera. The initial guess of camera intrinsics like focal length can be acquired via EXIF tags of the capture image [23]. However, due to the small projection hole in our system, we use a camera with strong lens distortion to have a wider view on the spherical screen. The focal length is also adjusted to make sure that the unblocked portion is in focus. So in our case it is not appropriate to calibrate the system without having camera intrinsic parameters as priors even with the help of nonlinear optimization.

Evaluation metrics: Although direct comparison with existing work using proposed metrics is preferable, limitations inherent in the type of display we are targeting differ from the situations covered by other work, which has been identified in Section 1 as scale, visibility and perspective-correction. These practical limitations of our multi-projector system are the primary motivators to create the proposed approach. Meanwhile, we include our additional proposed metrics and evaluation method for use by future researchers that have fewer limitations in their systems.

5. Conclusion

We presented an automatic calibration approach, as well as a practical evaluation method for a spherical FTVR display. We identified practical problems in calibrating spherical FTVR. Our proposed approach solves these problems and achieves less than $1mm$ on-surface point error and less than 1° line error using a $30cm$ diameter spherical screen. Although our calibration approach is for a spherical multiple-projector FTVR system, it can be applied to other multiple-projector system, especially for ones with curved screens of various sizes. The proposed evaluation method can work in different systems to estimate on-surface error. We believe our work has the potential to make curved display technology accessible to various applications.

6. Acknowledgment

We thank B-Con Engineering, NVIDIA and NSERC Canada for providing financial and in-kind support. We also thank Dr. Stavness and his group at University of Saskatchewan for helpful discussions.

References

[1] A. Ahmed, R. Hafiz, M. M. Khan, Y. Cho, and J. Cha. Geometric correction for uneven quadric projection surfaces using recursive subdivision of bézier patches. *ETRI Journal*, 35(6):1115–1125, 2013.

[2] K. W. Arthur, K. S. Booth, and C. Ware. Evaluating 3d task performance for fish tank virtual worlds. *ACM Transactions on Information Systems (TOIS)*, 11(3):239–265, 1993.

[3] H. Benko, A. D. Wilson, and R. Balakrishnan. Sphere: multi-touch interactions on a spherical display. In *Proceedings of the 21st annual ACM symposium on User interface software and technology*, pages 77–86. ACM, 2008.

[4] J. Bolton, K. Kim, and R. Vertegaal. Snowglobe: a spherical fish-tank vr display. In *CHI'11 Extended Abstracts on Human Factors in Computing Systems*, pages 1159–1164. ACM, 2011.

[5] H. Chen, R. Sukthankar, G. Wallace, and K. Li. Scalable alignment of large-format multi-projector displays using camera homography trees. In *Proceedings of the conference on Visualization'02*, pages 339–346. IEEE Computer Society, 2002.

[6] Y. Chen, D. W. Clark, A. Finkelstein, T. C. Housel, and K. Li. Automatic alignment of high-resolution multi-projector display using an un-calibrated camera. In *Proceedings of the conference on Visualization'00*, pages 125–130. IEEE Computer Society Press, 2000.

[7] C. Cruz-Neira, D. J. Sandin, and T. A. DeFanti. Surround-screen projection-based virtual reality: the design and implementation of the cave. In *Proceedings of the 20th annual conference on Computer graphics and interactive techniques*, pages 135–142. ACM, 1993.

[8] C. B. Duane. Close-range camera calibration. *Photogramm. Eng*, 37(8):855–866, 1971.

[9] G. Falcao, N. Hurtos, and J. Massich. Plane-based calibration of a projector-camera system. *VIBOT master*, 9(1):1–12, 2008.

[10] G. E. Favalora. Volumetric 3d displays and application infrastructure. *Computer*, (8):37–44, 2005.

[11] R. Hartley and A. Zisserman. *Multiple view geometry in computer vision*. Cambridge university press, 2003.

[12] M. Harville, B. Culbertson, I. Sobel, D. Gelb, A. Fitzhugh, and D. Tanguay. Practical methods for geometric and photometric correction of tiled projector. In *Computer Vision and Pattern Recognition Workshop, 2006. CVPRW'06. Conference on*, pages 5–5. IEEE, 2006.

[13] B. Lam, Y. Tang, I. Stavness, and S. Fels. A 3d cubic puzzle in pcubee. In *3D User Interfaces (3DUI), 2011 IEEE Symposium on*, pages 135–136. IEEE, 2011.

[14] J. J. Moré and D. C. Sorensen. Computing a trust region step. *SIAM Journal on Scientific and Statistical Computing*, 4(3):553–572, 1983.

[15] D. Moreno and G. Taubin. Simple, accurate, and robust projector-camera calibration. In *2012 Second International Conference on 3D Imaging, Modeling, Processing, Visualization & Transmission*, pages 464–471. IEEE, 2012.

[16] F. Polhemus. 3space fastrak users manual. *F. Polhemus Inc., Colchester, VT*, 1993.

[17] A. Raij, G. Gill, A. Majumder, H. Towles, and H. Fuchs. Pixelflex2: A comprehensive, automatic, casually-aligned multi-projector display. In *IEEE International Workshop on Projector-Camera Systems*, pages 203–211. Nice, France, 2003.

[18] A. Raij and M. Pollefeys. Auto-calibration of multi-projector display walls. In *Pattern Recognition, 2004. ICPR 2004. Proceedings of the 17th International Conference on*, volume 1, pages 14–17. IEEE, 2004.

[19] R. Raskar. Immersive planar display using roughly aligned projectors. In *Virtual Reality, 2000. Proceedings. IEEE*, pages 109–115. IEEE, 2000.

[20] R. Raskar, M. S. Brown, R. Yang, W.-C. Chen, G. Welch, H. Towles, B. Scales, and H. Fuchs. Multi-projector displays using camera-based registration. In *Visualization'99. Proceedings*, pages 161–522. IEEE, 1999.

[21] R. Raskar, J. van Baar, S. Rao, and T. Willwacher. Multi-projector imagery on curved surfaces. *Mitsubishi Electric Research Labs*, pages 1–8, 2003.

[22] B. Sajadi and A. Majumder. Scalable multi-view registration for multi-projector displays on vertically extruded surfaces. In *Computer Graphics Forum*, volume 29, pages 1063–1072. Wiley Online Library, 2010.

[23] B. Sajadi and A. Majumder. Automatic registration of multi-projector domes using a single uncalibrated camera. In *Computer Graphics Forum*, volume 30, pages 1161–1170. Wiley Online Library, 2011.

[24] B. Sajadi and A. Majumder. Autocalibration of multiprojector cave-like immersive environments. *Visualization and Computer Graphics, IEEE Transactions on*, 18(3):381–393, 2012.

[25] I. Stavness, F. Vogt, and S. Fels. Cubee: a cubic 3d display for physics-based interaction. In *ACM SIGGRAPH 2006 Sketches*, page 165. ACM, 2006.

[26] G. Strang. *Introduction to Applied Mathematics*. Wellesley-Cambridge, 1986.

[27] F. Teubl, C. S. Kurashima, M. Cabral, R. D. Lopes, J. C. Anacleto, M. K. Zuffo, and S. Fels. Spheree: An interactive perspective-corrected spherical 3d display. In *3DTV-Conference: The True Vision-Capture, Transmission and Display of 3D Video (3DTV-CON), 2014*, pages 1–4. IEEE, 2014.

[28] F. Teubl Ferreira, C. Kurashima, M. Calixto Cabral, and M. Knorich Zuffo. A small-scale and flexible platform for multi-projector systems development. In *Virtual and Augmented Reality (SVR), 2013 XV Symposium on*, pages 216–219. IEEE, 2013.

[29] A. Wagemakers, D. Fafard, and I. Stavness. Interactive visual calibration of volumetric head-tracked 3d displays. In *2017 SIGCHI Conference on Human Factors in Computing Systems (CHI '17)*. ACM, 2017, to appear.

[30] Z. Zhang. A flexible new technique for camera calibration. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 22(11):1330–1334, 2000.

[31] Q. Zhou, G. Miller, K. Wu, I. Stavness, and S. Fels. Analysis and practical minimization of registration error in a spherical fish tank virtual reality system. In *Asian Conference on Computer Vision*. Springer, 2016, to appear.