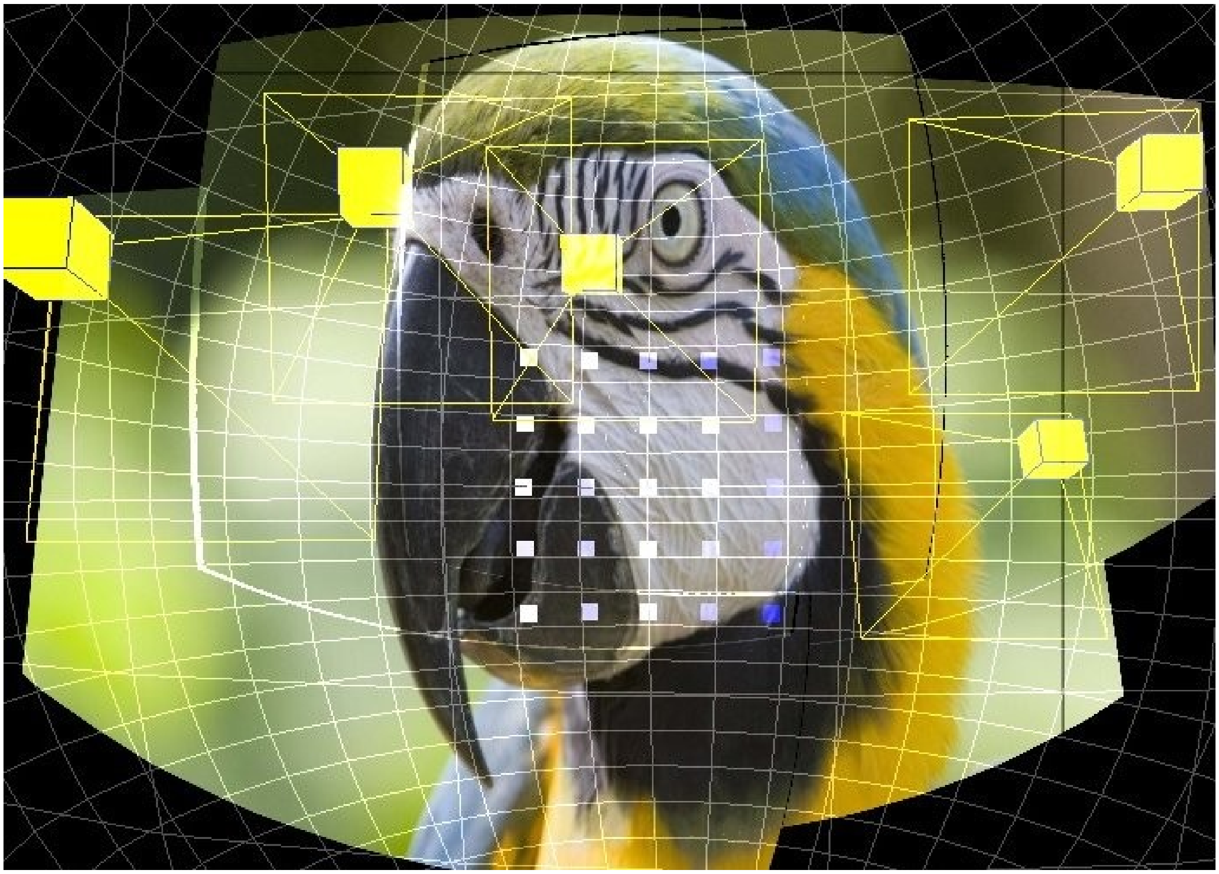
Automated Calibration Method for Multi-Projector Curved Displays

Cha Lee
University of California Santa Barbara
Santa Barbara, CA
chalee21@cs.ucsb.edu

Svetlin Bostandjiev
University of California Santa Barbara
Santa Barbara, CA
alex@cs.ucsb.edu



Abstract

We implement a method for automatically calibrating a multi-projector curved display for seamless projection of a virtual scene. Our proposed method is intended to be used in the Allosphere, a 3-story high spherical space, in which fully immersive and interactive environments can be experienced, but should work for any uniform and concave display surface. Our approach utilizes a single camera, which views all projections from all projectors. With some basic assumptions, our method computes the camera pose, projector poses, and projection regions for each projector.

1. Introduction

Large displays offer a variety of advantages over normal desktop display. While generally more expensive, large displays provide a level of immersion and freedom unmatched by a single-user head-mounted display or desktop stereoscopic display. Large displays also allow for an immersive experience for groups of users instead of a single individual. With the cost of hardware dropping, maturity of development software, and greater interest from universities and the public we have seen a rise in the number of large displays (275 large scale theater displays in universities, planetariums, and science centers worldwide as of 2006).

Large scale displays fall into three basic categories: large tiled displays, curved screen displays, and full dome displays. The scale of these displays may vary, although most are intended for the use of several to hundreds of users. While tiled displays consisting of many smaller flat screen displays are easier to implement, the advantages of a seamless and versatile display system make multi-projector curved displays appealing.

In the past fully immersive large scale displays have been mostly driven by industry and proprietary technologies. The high cost and large space requirements have made these types of displays mostly not accessible to the research community. Recently some of these challenges have been met by the increasing power of graphics hardware, decreasing cost and increasing performance of off-the-shelf projectors. Our proposed method follows this trend and is intended to be used with an off-the-shelf digital camera and a set of low cost projectors.

2. The Problem

A large scale curved display presents many problems such as photometric calibration for color and brightness, geometric calibration, distributed rendering, correct stereo rendering. Large non-planar displays are even more challenging because the mapping between user views and projection views is no longer a simple plane homography. Our method approximates a solution to the geometric calibration problem on a large spherical display. Other problems are beyond the scope of this project due to time constraints. We consider photometric calibration to be future work.

For a planar display surface, a 2D planar homography can be computed for each projector onto the display surface [3]. Nonplanar displays are more problematic, but Raskar et al. [7] demonstrates a clean method,s which utilizes a two pass rendering approach. Three assumptions are made: 1) the display surface geometry is known, 2) the projector’s poses are known with respect to the display surface, and 3) the user position is known with respect to the display surface.

3. Related Literature

In Brown et al [2], the authors give a very good overview of existing camera based approaches to multi-projector displays. The paper touches on both geometric and photometric calibration algorithms. Although much of the paper focuses on the photometric problems on planar displays, the authors give great clues to solving the same photometric problems for spherical displays.

Raskar et al [7] first introduces an elegant method for geometric calibration technique to create a low-cost multi-projector curved screen display, and then in [6] uses a more efficient parametric approach called the quadric transfer to solve for a smaller set of 3D surfaces. Their approach uses camera pairs, while we use a single camera but the main idea remains the same and our approach makes the same assumptions.

In [2], Latnz gives a good survey of existing large scale displays and their advantages. His paper focuses on full dome displays which are of interest to us since our method is intended for the allosphere. Although not directly relevant to the problem we were trying to solve, the author provides good motivation for research in the area of large displays.

4. Approach

In our approach there are three basic steps: approximate positioning, projector calibration, and projection masking. During the calibration process the camera and projectors are assumed to be rigid with respect to the spherical display.

4.1 Approximate Positioning

As with any existing algorithm the first step for calibration is a good initial setup. The user should position the camera in such a way that all the physical feature points (known as calibration points for the rest of this paper) are viewable. Ideally the calibration points should take up a large portion of the camera image and also take up a large portion of the display surface for best results. After positioning the camera, the projectors should also be placed such that their projected images overlap a good portion of the calibration points.

4.2 Projector Calibration

In the first step in the projector calibration phase we acquire the corresponding camera image points for the physical calibration points. In our simulation we assume the calibration points can be activated or “lit” one at a time. By lighting one calibration point at a time in a predefined order we acquire the point correspondences for all of the calibration points. We store both the 3D positions and the corresponding 2D image points in two vectors (P_{3D} and P_{2D_Cam} respectively).

In the second step we obtain the calibration point correspondences for the projectors. We create an image, in which every pixel's red value is based on the X coordinate of the pixel's position and the green value is based on the Y coordinate of the pixel's position. For each projector, we project this color encoded image and capture the view from the camera. We then check the pixel values in the camera view at every image point stored in P_{2D_Cam}. The color of these pixels determines the calibration points' 2D positions in the projector view. We store these 2D points in another vector ($P_{2D\_Proj\_i}$) (Figure 1).

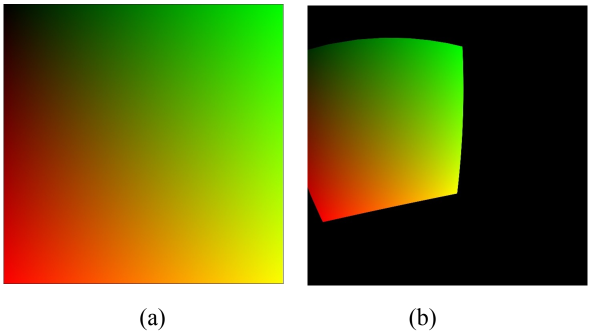

(a) (b)

Figure 1: Color encoded image. (a) Viewed from a projector. (b) Viewed from the camera.

After acquiring all point correspondences, we can now estimate the projector poses. We use OpenCV's *cvFindExtrinsicCameraParams2* method, which estimates the camera pose from planar or non-planar world points. For the planar case, it computes a homography and for the non-planar case it uses a direct linear transform. It then optimizes using an iterative method. More documentation can be found on the OpenCV website [5].

4.3 Projection Masking

The last step in our method is to compute an image mask to approximate an edge blending solution. Without this any areas with overlapping projections would result in overly bright pixels (Figure 2).

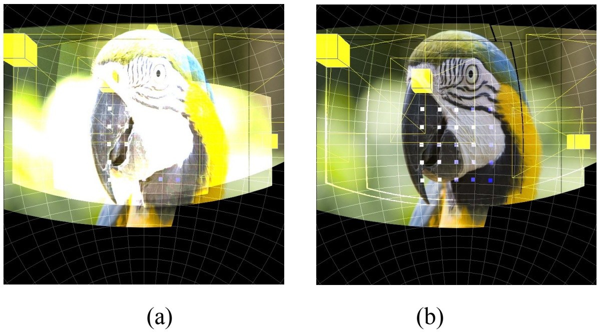

(a) (b)

Figure 2: Projection masking. (a) Image without projector masks. (b) Image with projector masks.

We use a naive approach by a system of elimination. We begin with the first projector and allow it to use all the area it can (from the perspective of the camera). Then we mark that area as unusable and mask it for any following projector whose projection image may overlap with previous projectors' areas. We repeat this until all projectors have been used (Figure 3).

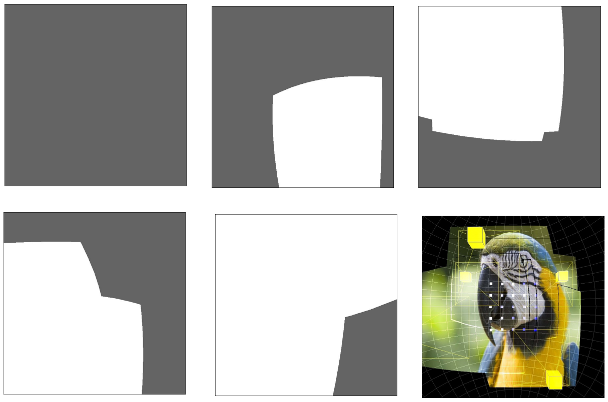


Figure 3: Projector masks for five projectors in order of masking and the resulting global image from top left to bottom right. Areas in white are masked areas.

5. Application

Since we did not have access to more than two projectors and could not access the Allosphere directly, we built a simulation of the environment in OpenGL. The spherical display is modeled as a perfect sphere and the projectors and camera are modeled as pinhole cameras. For our application we used the same intrinsic parameters for both the projectors and camera. The assumption is that a real setup would have intrinsically calibrated projectors and cameras to begin with. We then use projective texturing onto the sphere to simulate the projection of the real projector images.

The simulation application allows the user to move both cameras and projectors in real-time and is a good way to test the calibration algorithm under a controlled environment. We create a set of “real” projectors, with known poses. We use these parameters to project virtual images onto the sphere. Then, we calibrate and compute the poses for a set of “calibrated” projectors. The application then renders a virtual 3D scene based on a user position. The resulting 2D image is projected onto the virtual sphere and a final image for each real projector is acquired based on the calibrated projectors. This final image for each projector is just what it would see if it were a camera instead. The images are then projected by the real projectors. This accurately simulates what a physical setup using our calibration method would have to implement.

6. Results

In our approach errors can accumulate in two places: 2D feature detection and projector pose estimation. Because OpenGL interpolates pixel values when rendering primitives the final 2D points acquired during 2D feature search could have some error. This error is carried along to the pose estimation phase and results in a projector pose error. Since for all intents and purposes we only care about the user noticeable errors, we are only interested in the misalignment of the real projector image as compared to the calibrated projector image when viewed from the camera (Figure 4).

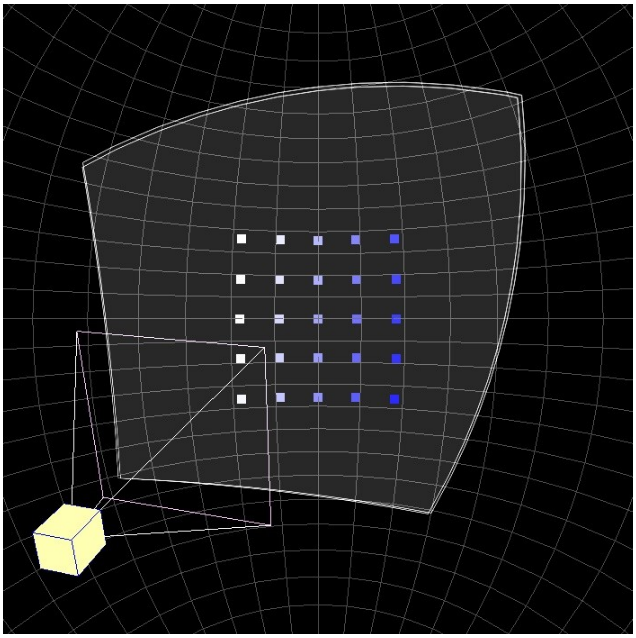


Figure 4: Calibrated projector image region superimposed on real projector image region. 25 calibration points were used to calibrate this projector.

We define the error as the pixel distance in the camera's perspective divided by the pixel size of the image. We use this percentage as our metric. We project *n* points using the real projector's pose and again using the calibrated projector's pose. The error is then equal to the average absolute distance between every two corresponding points (from real image (a) to calibrated image (b) for all *n* points) divided by the size of the camera image (Equation 1).

$$Error = \frac{\frac{\sum_{i=1}^{n} \left| P_a^i - P_b^i \right|}{n}}{imageSize}$$

Equation 1: Alignment error metric.

Although this metric is sensitive to projector positioning, we feel it is a good approximation for evaluating our calibration method. Projectors at a larger projection angle and distance would naturally show larger errors due to the more extreme warping. Good initial projector placement is required for a good calibration. Fig. 5 shows the plot of the average error as a function of the number of calibration points used in calibrating each projector. There were five projectors used in the experiment. Each projector was manually placed such that most calibration points were overlapped by its projected region. We started with 25 points, the least amount where all five projectors could still be successfully calibrated without too much overlapping, and ended with 225 points, after which the calibration time began to exceed a minute.

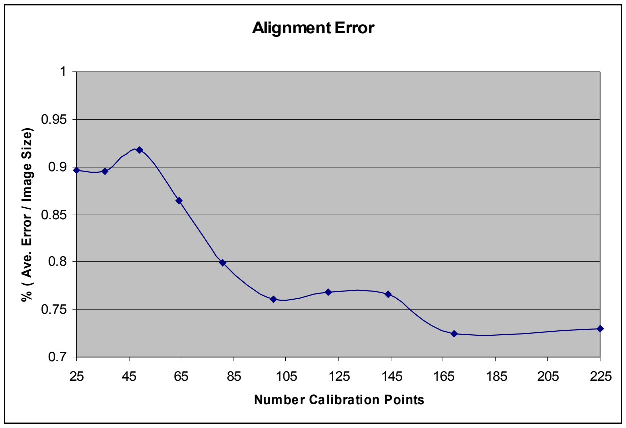


Figure 5: Average distance error as a percentage of the image size over 25, 36, 49, 64, 81, 100, 121, 144, 169, and 225 calibration points. Five projectors were used.

With 64 points or less the error was generally between 0.8 and 1.0 percent given a decent placement of the projectors. For our experiments this translates to an alignment error of 6 to 8 pixels for an image size of 800 by 800 pixels. This error can be seen in Fig. 4, where the true projector region is superimposed on the calibrated projector's region. In our experiments we were never able to get the error below 0.7 percent. We believe the biggest error factor is due to the inaccuracy of our 2D points in the projector images. Because we use a color coded image (as seen in Fig. 1) for acquiring the 2D points, we lose some resolution due interpolation in the OpenGL rendering pipeline.

7. Conclusion

In conclusion we have presented a fast approach to calibrating multiple projectors on a spherical display. We realize that our approach does not stand up to the state of the art, but again this is an initial solution that needs further work before completion.

For future work we need to improve both the pose estimation and the edge blending approach. For better pose estimation results we consider using bundle adjustment algorithm after all projectors have been calibrated. This would be an iterative process, using feedback from the real projectors to correct for errors in the projector pose.

For our edge blending technique, we need to consider lighting environment and projector intensity differences. One approach is to clamp projector intensities based on a global minimum (the dimmest lit pixel). Although this reduces the overall image intensity, it may be enough for our purposes. Another problem a spherical display would need to address is ambient lighting coming from the reflection off of the curved surface. We are still considering possible approaches to correct for this.

8. References

[1] A. Raij and M. Pollefeys. Auto-Calibration of Multi-Projector Walls. In *Proc. of the 17th Int. Conf. on Pattern Recognition* (ICPR'04). 2004.

[2] Brown, M., Majumder. A., and Yang, R. Camera-Based Calibration Techniques for Seamless Multiprojector Displays. *IEEE Transactions on Visualization and Computer Graphics*, Vol. 11- 2, 2005.

[3] Hartley, R.I. and Zisserman, A. Multiple View Geometry in Computer Vision. Cambrige Univ. Press, 2000.

[4] Lantz, E. 2007. A survey of large-scale immersive displays. In *Proceedings of the 2007 Workshop on Emerging Displays Technologies: Images and Beyond: the Future of Displays and interacton* (San Diego, California, August 04 - 04, 2007). EDT '07, vol. 252. ACM, New York, NY, 1. DOI= http://doi.acm.org/10.1145/1278240.1278241.

[5] OpenCV. http://opencvlibrary.sourceforge.net

[6] Raskar, R.; van Baar, Jeroen, "Low-Cost Multi-Projector Curved Screen Displays", *International Symposium Scoiety for Information Display (SID)*, May 2005 (SID 2005, TR2005-079).

[7] Raskar, R., Brown, M.S., Yang, R., Chen, W., Welch, G., Towles, H., Seales, B., and Fuchs, H. Multiprojector Displays Using Camera Bases Registration. *In Proc. IEEE Visualization* 1999, pp. 161-168, 1999.

[8] Z. Zhang. A Flexible New Technique for Camera Calibration. *IEEE Trans. on Pattern Recognition and Machine Intelligence*. Vol. 22 – 11, 2000.