Camera-based Calibration for Scalable Immersive Rendering

William C. Thibault [*]
California State University, East Bay

Immersive multi-projector displays with dozens of projectors are becoming easier to build as projection technology proliferates. We envision scenarios such as a classroom of students with individual projectors, and informal groups of people with projector-equipped mobile devices, in which computer-generated imagery can be generated and displayed in real time. Ideally, the resolution of the multi-projector display should grow with the number of projectors, and support arbitrarily wide displays (to 4π steradians).

Our approach attempts to minimize the resampling error possible in Raskar's two-pass algorithm [Raskar et al. 1999] by fitting tightly the frustum used to render the scene in the algorithm's first pass to the portion of the field-of-view for which the projector is responsible. (We assume a fixed viewpoint, at the location of the calibrating camera. Each projector's "responsibility" is to render the part of the scene visible from the viewpoint, which can have an arbitrary extent if no assumptions are made about projector placement or display surface shape. We assume each projector's responsibility is small enough to be rendered with a conventional perspective projection, although this can be extended to arbitrary fields-of-view using non-linear projection[Yuen and Thibault 2008].)

Care must be taken when using two-pass rendering for immersive displays if resampling errors are to be minimized. The simplest approach to immersive rendering is to render to a cube-map texture in the first pass, and apply the texture to a model of the screen geometry in the second pass. However, cube-map faces are limited to the maximum texture supported by the GPU. For extremely high-resolution displays, using cube-maps will not scale beyond a certain number of pixels, while the portion of the field-of-view contributed by each projector decreases as more projectors are added. With our technique, each projector frustum is automatically computed to match the associated projector's contribution to the overall field-of-view of the display. For scenarios such as "gigapixel displays" created with large numbers of projectors, cubemaps fail to provide the resolution needed. Our technique, together with the scalable rendering architecture of Equalizer[Eilemann et al. 2009], allows for creating immersive displays with extremely high resolution and good interactive performance, without requiring precise physical alignment of projectors. Equalizer and similar systems typically handle displays built from tiles of flat screens (walls and CAVEs) well, but Equalizer's display configuration specifications provide added generality that makes possible the creation of arbitrarily oriented, overlapping display segments. We use this flexibility to create Equalizer configurations from our camera-based projector calibration data. Physical projectors need only casual alignment, and can be placed at arbitrary locations.

Our camera-based approach is applicable to immersive displays of arbitrary field of view. During calibration, temporally-coded structured light patterns are used to establish a sparse set of projector-camera correspondences. If the field of view exceeds that of the camera, the calibration can be repeated with a rotated camera, and the correspondences used to solve for the rotation. The camera coordinates of each set of per-projector correspondences are then mapped to direction vectors based on a fisheye lens calibration[Yuen and Thibault 2008]. We use the results of the projector calibration to create the frustum used for each projector's rendering, as well as a floating-point texture to be used for the second-pass warp. The frustum is determined as the smallest field-of-view centered at the mean camera direction of the projected features that includes all of the correspondences for the given projector. We use the Equalizer framework [Eilemann et al. 2009] to handle cluster synchronization and rendering: frusta specifications are written to an Equalizer configuration file, and the framework is extended to transparently implement the details of the two-pass algorithm. The first pass renders to an FBO, and a fragment program is used in the second pass to resample the rendered frame.

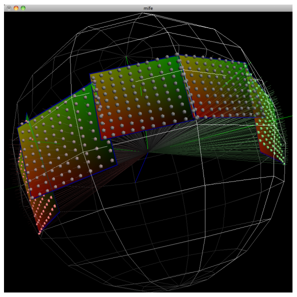

Figure 1: Per-projector camera frusta.

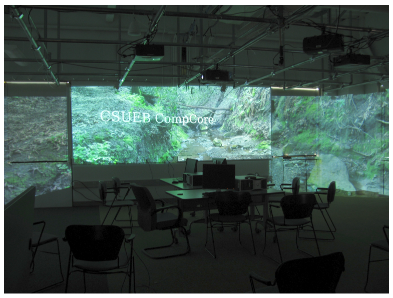

Figure 2: A panoramic image displayed on our system.

References

EILEMANN, S., MAKHINYA, M., AND PAJAROLA, R. 2009. Equalizer: A scalable parallel rendering framework. *IEEE Transactions on Visualization and Computer Graphics 15*, 436–452.

RASKAR, R., BROWN, M., YANG, R., CHEN, W., WELCH, G., TOWLES, H., SEALES, B., AND FUCHS, H. 1999. Multi-projector displays using camera-based registration. In *IEEE Visualization 99*.

YUEN, N. P. Y., AND THIBAULT, W. C. 2008. Inexpensive immersive projection. In *Proceedings of IEEE Virtual Reality 2008*, 237–240.

[*]e-mail: william.thibault@csueastbay.edu

Copyright is held by the author / owner(s).
SIGGRAPH 2010, Los Angeles, California, July 25 – 29, 2010.
ISBN 978-1-4503-0210-4/10/0007