Appears in: Technical Sketches, Computer Vision and Pattern Recognition, 2001

Calibrating Scalable Multi-Projector Displays Using Camera Homography Trees

Han Chen[1], Rahul Sukthankar[2], Grant Wallace[1] Tat-Jen Cham[2]

{chenhan,gwallace}@cs.princeton.edu
{rahul.sukthankar,tat-jen.cham}@compaq.com

[1]Department of Computer Science; Princeton University; Princeton, NJ 08544
[2] Compaq Research (CRL); One Cambridge Center; Cambridge, MA 02142

Abstract

We present a practical vision-based calibration system for large format multi-projector displays. A spanning tree of homographies, automatically constructed from several camera images, accurately registers arbitrarily-mounted projectors to a global reference frame. Experiments on the 18'×8' Princeton Display Wall (a 24 projector array with 6000×3000 resolution) demonstrate that our algorithm achieves sub-pixel accuracy even on large display surfaces. A direct comparison with the previous best algorithm shows that our technique is significantly more accurate, requires far fewer camera images, and runs faster by an order of magnitude.

1. Introduction

Large format high-resolution display devices are becoming increasingly important for scientific visualization, industrial design and entertainment applications. A popular approach to building such displays is the *projector array*, where several commercially-available projectors are tiled to create a seamless, high-resolution display surface.

The projectors in the array require precise geometric calibration to prevent artifacts in the final display. This is typically done by manually aligning the projectors so that their projection areas tile the desired space, and then pre-warping the projected image to eliminate keystoning and discontinuities. Unfortunately, this process is labor-intensive and time-consuming; furthermore, the display wall requires frequent recalibration since the projectors shift slightly due to vibration and thermal flexing in the mounts. Given the increasing demand for large format display walls, calibration solutions must scale well to multi-projector arrays of arbitrary size.

Several ideas for camera-based automation of projector array alignment have recently been proposed. Surati [6] builds lookup tables that map pixels from each projector to points on the display surface; this is done by physically attaching a calibration grid (printed by a high-precision plotter) onto the surface. While this approach is adequate for a 2×2 array of projectors, it scales poorly for larger displays since creating and accurately mounting an absolute measurement grid onto the display surface is infeasible. Raskar *et al.* [4] employ two calibrated cameras in conjunction with projected patterns to recover a 3-D model of a (possibly non-planar) projection surface. However, this method requires that the entire display surface be small enough to be completely visible in the cameras' field of view. As display walls become larger, capturing a single camera image of the entire display surface becomes increasingly impractical. This motivates approaches that can integrate information about the projector geometry from a set of camera images, each of which observe a small portion of the display surface. Chen *et al.* [2] use a pan-tilt camera to observe the individual overlap regions in the projector array. Information about local discontinuities (point-matches and line matches across the seam) is acquired using an iterative process, and a large global optimization problem is constructed using this data. Simulated annealing is used to find a set of pre-warps that minimizes discontinuity errors. The primary advantage of their algorithm (referred to as SA-Align in the remainder of this paper) is that, in principle, it scales well to large display walls since the uncalibrated camera can easily scan the overlap regions. In practice, SA-Align is very slow and often fails to converge to the correct solution unless the initial (manual) alignment between projectors is good.

This paper presents a scalable approach to display wall calibration that is both more accurate and faster than existing approaches. It is motivated by the single-projector keystone correction system described in [5], adapted to employ images taken from multiple, uncalibrated cameras. Our system efficiently scales to projector arrays of arbitrary size without sacrificing local or global alignment accuracy. The experiments described in this paper were performed on the Princeton Scalable Display Wall [3], an 18'×8' 24 projec-

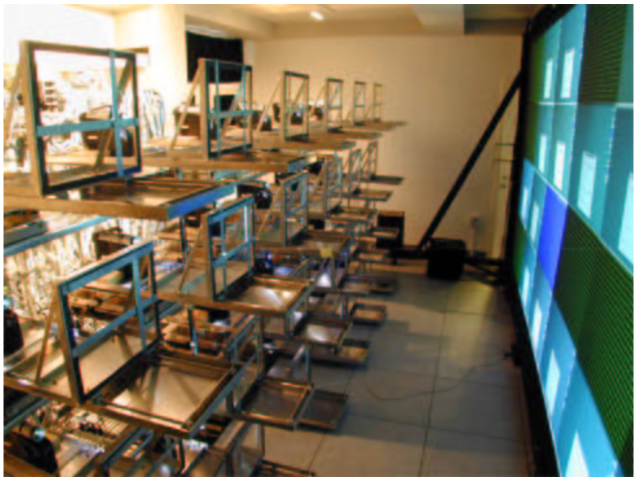

Figure 1: The Princeton Scalable Display Wall is an 18'×8' display (the largest in academia), with an effective resolution of 6000×3000 pixels. This photograph was taken from behind the rear-projection screen and shows the array of 24 Compaq MP-1800 projectors.

tor display with an effective resolution of 6000×3000 pixels (see Figure 1).

2. Display Wall Calibration

We assume that: the positions, orientations and optical parameters of the cameras and projectors are unknown; camera and projector optics can be modeled by perspective transforms; the projection surface is flat. Therefore, the various transforms between cameras, screen and projectors can all be modeled as 2-D planar homographies:

$$\begin{pmatrix} xw \\ yw \\ w \end{pmatrix} = \begin{pmatrix} h_1 & h_2 & h_3 \\ h_4 & h_5 & h_6 \\ h_7 & h_8 & h_9 \end{pmatrix} \begin{pmatrix} X \\ Y \\ 1 \end{pmatrix},$$

where (x, y) and (X, Y) are corresponding points in two frames of reference, and $\vec{h} = (h_1 \ldots\ h_9)^T$ (constrained by $|\vec{h}| = 1$) are the parameters specifying the homography. These parameters can be determined from as few as four point correspondences, using the closed-form solution described in [5].

Our system employs the above technique to compute two types of homographies: camera-to-camera and projector-to-camera. Each is described in greater detail below, and illustrated in Figure 2.

First, camera-to-camera homographies capture the relationship between different camera views of the display surface. Although each view typically observes only four or fewer projectors, the system combines these views to generate a reference frame for the entire display surface.

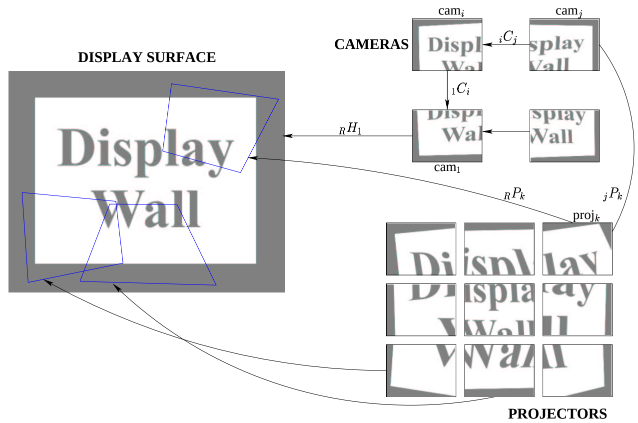

Figure 2: This diagram shows the relationship between the various homographies described in the text. Our system's goal is to recover the homography (${}_RP_k$) mapping each projector k, to the global reference frame. Although ${}_RP_k$ cannot directly be observed, it can be derived by composing ${}_jP_k$, the homography between that projector and some camera, and the chain of homographies connecting that camera to the root node of the homography tree. The geometric distortion for images projected by k's can then be corrected using a pre-warp of ${}_RP_k^{-1}$.

Conceptually, this is equivalent to automatically building a panoramic mosaic from a set of photographs. One cannot directly compute a homography between two camera views that do not overlap since they share no point correspondences. Therefore, our system builds a tree of homography relationships between adjacent views that spans the complete set of cameras[1]; the mapping from any given camera to the panoramic reference frame is determined by compounding the homographies along the path to the reference view at the root of the tree:

$${}_RH_j = {}_RH_1 \times {}_1C_i \times \cdots \times {}_iC_j,$$

where ${}_RH_j$ is the homography mapping points from camera j to the global reference frame, ${}_sC_t$ are homographies connecting adjacent camera views and ${}_RH_1$ maps the root camera view to the global reference frame.[2]

Second, the projector-to-camera homographies transform each projector's area of projection into some camera's coordinate system. These homographies are determined as follows. A white rectangle is displayed by projector k; this appears as some quadrilateral on the display surface (due to keystoning), and as some (other) quadrilateral in some camera view j. The corners of the projected rectangle are

[1]The homography tree is refined in a second pass as described in [1].

[2]The transform ${}_RH_1$ ensures that the global frame axes are aligned with the display surface rather than the root camera; ${}_RH_1$ is computed by observing a reference rectangle on the display wall from any camera.

known in the projector frame of reference and the corresponding corners of the quadrilateral are determined (to sub-pixel accuracy) in the camera image. This gives us the projector-to-camera homography ${}_jP_k$. Since we know the mapping between any camera j and the reference frame, this enables us to compute ${}_RP_k$, the transform from projector k to the reference frame: ${}_RP_k = {}_RH_j \times {}_jP_k$. Note that ${}_RP_k$ captures the geometric distortion induced by the projector's off-center placement. This distortion can be removed by pre-warping each projector k's output by ${}_RP_k^{-1}$. Results are shown in Figure 3 and discussed below.

3. Results

This section presents evaluations of our algorithm on three important metrics: local alignment error, global alignment error and running time. Results were obtained for a variety of multi-projector arrays under three experimental conditions: (1) uncalibrated projector array; (2) array calibrated using the previous best solution, SA-Align [2]; (3) array calibrated using our algorithm. All experiments were performed on the same hardware setup: the 24-projector display wall at Princeton University [3]. Three multi-projector array configurations were used in our experiments: 2×2, 3×3, and 6×4 (complete wall). The experiments are detailed below.

3.1. Local alignment error

To appear seamless, a display wall should minimize alignment error between adjacent projectors. Qualitatively, misalignment creates artifacts such as discontinuities and double-images in the overlap region (see Figure 3, top row). Quantitatively, the error can be characterized by the maximum displacement between a point shown on one projector and the same point displayed by an adjacent projector. Since the SA-Align algorithm explicitly observes point- and line-mismatches in the overlap regions and optimizes over warp parameters to minimize the total error, one would expect it to do well on this measure. Our technique aims to independently register each projector to the (panoramic) reference frame as accurately as possible, only incidentally minimizing local errors. Nevertheless, as can be seen from Table 1, our algorithm achieves sub-pixel registration accuracy, even on large display walls — demonstrating conclusive improvements over SA-Align under all configurations.

3.2. Global alignment error

Global alignment error measures the displacement between pixels in the projected image and their desired locations, as measured in the reference frame. Note that a projector array with excellent local alignment may still exhibit large global alignment errors for two reasons: (1) the projected image may be globally warped so that its edges are not parallel with the sides of the display surface; (2) small errors in local alignment can accumulate as homographies are chained, resulting in non-linear distortions in the projected image (see Figure 3, bottom row).

Table 1: Average local errors: horizontal & vertical (pixels).

Proj.	Uncalibrated		SA-Align		Our system	
array	X-err	Y-err	X-err	Y-err	X-err	Y-err
2×2	153.8	150.3	6.5	6.8	**0.4**	**1.0**
3×3	108.3	140.9	10.7	9.1	**0.6**	**0.6**
6×4	60.2	74.1	18.9	18.3	**0.4**	**0.5**

Table 2: Average global errors: horizontal & vertical (pixels).

Proj.	Uncalibrated		SA-Align		Our system	
array	X-err	Y-err	X-err	Y-err	X-err	Y-err
2×2	102.8	70.5	6.7	1.9	**1.4**	**0.5**
3×3	177.5	161.6	9.0	16.5	**2.2**	**1.0**
6×4	44.5	62.8	30.3	16.8	**2.7**	**0.8**

To evaluate global alignment error, we projected a coarse grid on the display wall, and used the grid intersections as features. We manually measured the distance from each feature to two reference points on the display surface to determine each feature's absolute location. The global alignment error is computed as the average displacement between a feature's desired and actual locations. As shown in Table 2, our method has a substantially smaller global alignment error than the previous methods.

3.3. Running time

To be of practical value, a calibration algorithm must be fast as well as accurate. There are two components to running time: the time taken to acquire images, and the time required for computation. Table 3 compares the running time for our algorithm with SA-Align on identical hardware. Images were automatically acquired using a 640×480 pan-tilt camera and computation performed on a Pentium-III 866 MHz machine. SA-Align was configured to use 6 point and 6 line features per seam, and 50,000 iterations of simulated annealing (as described in [2]). Our system acquired camera images for each 2×2 sub-array of adjacent projectors and the algorithm was implemented in unoptimized Matlab code; our algorithm is faster than SA-Align by an order of magnitude.

For completeness, we ran SA-Align using a variety of iteration parameter settings on the 6×4 configuration. More iterations achieved slightly better solutions at the expense of running time. However, even after 500,000 iterations (912 minutes of computation time!), SA-Align's accuracy (X-err=9.7, Y-err=8.5) remains substantially inferior.

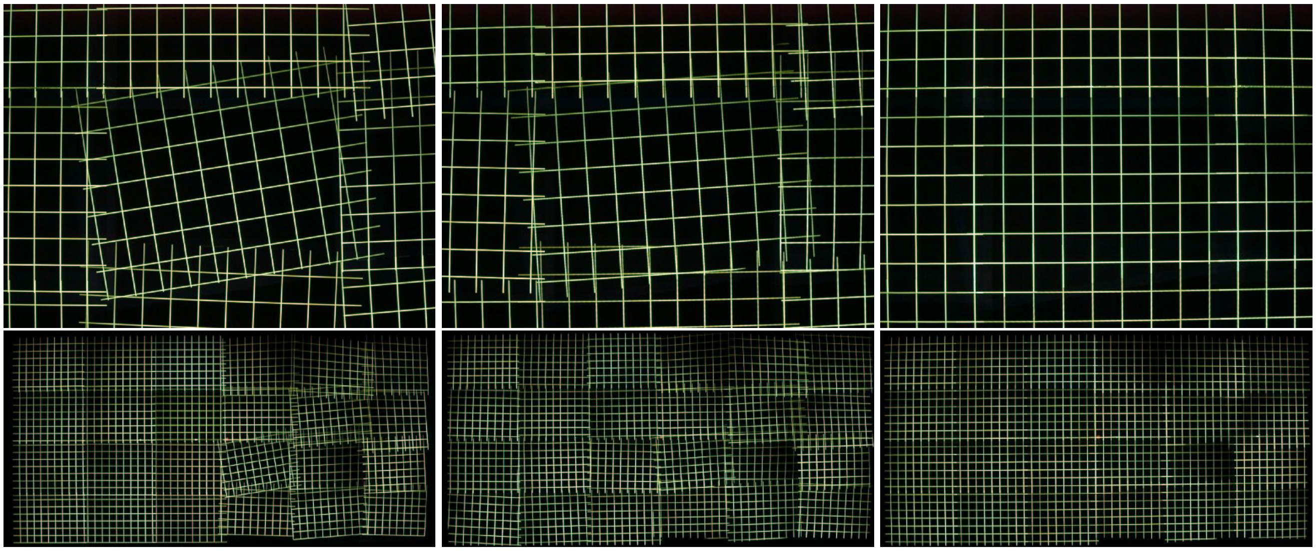

Figure 3: Local and global alignment errors on the Princeton 6×4 projector display wall under three conditions: uncalibrated (left); SA-Align [2] calibration (middle); and our technique (right). The top row shows enlarged views of the regions with the worst local errors. While SA-Align improves upon the uncalibrated case it still displays significant misalignment. By contrast, our algorithm's worst local errors are barely visible. The bottom row shows photographs of the entire 18'×8' multi-projector display surface. In this experiment, the 12 projectors on the left side were mounted carefully (manually) while the 12 projectors on the right were placed haphazardly. Here, SA-Align converges to a bad solution where *both* sides of the display are distorted. Our technique corrects for all of the distortions and displays an image with very little global error.

Table 3: Running time (minutes)

Proj.	Chen *et al.*		Our system	
array	Camera	Compute	Camera	Compute
2×2	12	71	**1**	<**1**
3×3	37	84	**2**	**2**
6×4	130	91	**9**	**6**

4. Conclusion

This paper describes a practical vision-based system for automatically calibrating large format multi-projector displays. Our algorithm is substantially more accurate than previous solutions (in both local and global metrics) and runs an order of magnitude faster. It is now used regularly to calibrate the Princeton Display Wall.

Acknowledgments

We would like to thank Professor Kai Li and the Princeton Scalable Display Wall group. The Princeton Display Wall project is supported in part by Department of Energy grants ANI-9906704 and DE-FC02-99ER25387, and Intel Research Council and Intel Technology 2000 equipment grant.

References

[1] H. Chen, R. Sukthankar, G. Wallace, and T.-J. Cham. Accurate calculation of camera homography trees for calibration of scalable multi-projector displays. Technical Report TR-639-01, Princeton University, September 2001.

[2] Y. Chen, D. Clark, A. Finkelstein, T. Housel, and K. Li. Automatic alignment of high-resolution multi-projector display using an uncalibrated camera. In *Proceedings of IEEE Visualization*, 2000.

[3] K. Li, H. Chen, Y. Chen, D. Clark, P. Cook, S. Daminakis, G. Essl, A. Finkelstein, T. Funkhouser, A. Klein, Z. Liu, E. Praun, R. Samanta, B. Shedd, J. Singh, G. Tzanetakis, and J. Zheng. Building and using a scalable display wall system. *Computer Graphics and Applications*, 20(4), 2000.

[4] R. Raskar, M. Brown, R. Yang, W. Chen, G. Welch, H. Towles, B. Seales, and H. Fuchs. Multi-projector displays using camera-based registration. In *Proceedings of IEEE Visualization*, 1999.

[5] R. Sukthankar, R. Stockton, and M. Mullin. Smarter presentations: Exploiting homography in camera-projector systems. In *Proceedings of International Conference on Computer Vision*, 2001.

[6] R. Surati. *A Scalable Self-Calibrating Technology for Seamless Large-Scale Displays*. PhD thesis, Department of Electrical Engineering and Computer Science, Massachussetts Institute of Technology, 1999.