DottyToto: A measurement engine for aligning multi-projector display systems

Mark Hereld[*ab], Ivan R. Judson[a], and Rick Stevens[abc]
[a]Argonne National Laboratory;
[b]Computation Institute, Argonne National Laboratory, and The University of Chicago;
[c]Dept. of Computer Science, The University of Chicago

ABSTRACT

Tiled displays systems built by combining the images from arrays of projectors can provide huge numbers of pixel elements to applications needing to visually represent lots of information. Such applications are already coming into wide usage and include large scientific visualizations, collaborative virtual environments, and rich multimedia spaces. It is, however, difficult to create the illusion of a unified seamless display for a variety of reasons including optical distortion of the individual projector images due to imperfections in the lenses and basic alignment of the projectors. In this paper we describe an efficient and optimized measurement process using inexpensive components that is tolerant of a wide range of imperfections in components and measurement setup (lighting conditions, camera optics, etc.). Our method nonetheless is capable of accurate and detailed measurement of the layout of all projector images, including the generation of a detailed model of the distortions in each projector optical system. It performs these measurements on the entire array of projectors at once. Once the detailed mapping between projector pixels and mural pixels is measured, the resulting relations can be used in any of a number of ways to improve the appearance of images projected on the display.

Keywords: Multi-projector displays, tiled displays, calibration, alignment, projector arrays.

1. THE PROBLEM

High-resolution display systems enable deep and information-rich scientific visualizations. When they are implemented in wall- and room-sized form factors, they promote presence in group-to-group collaborative environments. Their development has been an intense area of research for years[1-3], fueled first by developments in high-performance graphics, later by the spread of commodity computing and graphics, and more recently by the evolution of high-performance commodity cluster computing. Tiling multiple subsystems into a seamless whole is *the* scalable road to highest resolution at any given point in time.

One obstacle to creating very large high-resolution imaging systems by tiling projectors is creating a seamless transition from one projector to a neighbor. Put another way, a goal for such aggregated systems is to present large numbers of pixels (proportional to the number of projectors in the system) in such a way as to hide evidence of the individual contributing projected images.

1.1. Seamless Image Geometry

Of the several important aspects to this problem, among them luminosity and color uniformity, geometrical continuity across the projector-to-projector transitions is the subject of this paper. The user of a tiled display system is meant to imagine the large display as a single resource for placing rendered image content at certain locations on the display surface in the usual way. In current conventional display systems the model is that of a regular grid of pixels on the display surface that are mapped to by memory locations in a frame buffer. More generally one can refer to some set of world coordinates on the display system. The problem of guaranteeing geometrical continuity can be thought of as generating coordinate maps between individual projected frame buffers and this set of display system coordinates.

[*] hereld@mcs.anl.gov; phone: 630.252.4170

In this paper we present a method for measuring the mapping between projector frame buffer coordinates and display coordinates, a method that we call DottyToto. The method can be used to quickly measure these mappings for all projectors in a tiled array at once with a single, inexpensive, uncalibrated digital camera. The result of the measurement can be used to correct the alignment of projectors in the array and/or to correctly place image content into the frame buffers feeding the individual projectors so that overlapping and neighboring regions in the projected images appear to be smooth and continuous.

1.2. The Main Issues

The properties of projectors and aggregates of projectors that can contribute to geometrical mismatch in the image plane are imperfect projector optics and misalignment in six degrees of freedom of the projector with respect to the display surface.

In this paper we discuss the problem in terms of three different coordinate systems. The first is the set of 2D coordinate systems for each image tile. It is natural and convenient to use the pixel addressing scheme of the frame buffer as a coordinate system for this image tile, taking on values in the range (0..1023, 0..767) for pixels that can be illuminated. The second is the global coordinate system, which labels physical points on the entire flat display surface. From the standpoint of an application using the tiled display, it is often convenient to imagine a virtual frame buffer with virtual pixel coordinates in the form of row and column values, analogous to the frame buffer coordinates of the individual image tiles. The third coordinate system is the one used by the digital camera to make the measurements, the pixel coordinates of the detector in the camera.

The goal of DottyToto is to use the camera to accurately determine mappings between the tile coordinates (for each tile) and the global coordinates. The measurement and correction method we describe in the rest of the paper is designed to be robust for a range of measurement conditions: intensity variation across mural and individual projection tiles, camera pose, small changes in camera pose from image to image, imperfect camera optics, different tiled arrays, camera auto-gain, detector saturation, and ambient lighting.

The result of the measurement and analysis is a detailed and accurate mapping of pixels coordinates for each projector in the array to a unified coordinate system for the entire mural. These can be used in a number of ways:

- *Simple offset*. Determine best 2-parameter fit; that is, find the simple offset that places the tile in the mural coordinates. We have used this method to dice up large images (or movies) for display.

- *Affine map*. Determine pose parameters for each projector that can in turn be used to correct image rendering using an affine mapping, often represented as a 3 x 3 matrix transformation on 2D homogeneous coordinates. This application can be implemented efficiently with the addition of a simple matrix operation in the graphics pipeline.

- *Image warping*. Warp the image content into the frame buffers of the projectors to correct for projector pose and distortion in the projector optics; this is sometimes called rubber sheeting. This method produces the best results and can be used as part of inline processing of display content, or can be used to configure "zone corrections" in high-end projectors or specialized intermediate hardware.

- *Automated alignment*. Compute corrections to be applied to projector alignment motors affecting position, orientation, and zoom of each projector in the array. This approach is well suited to automatic alignment of large arrays of projectors followed by a second measurement to extract final mappings for image warping.

- *Manual alignment*. Iterate by interleaving manual alignment adjustments of the projectors with new measurements, using the measurements to guide the adjustment process.

1.3. Previous Work in This Area

A common approach to the problem of achieving a unified and uniform geometrical basis for image presentation on tiled displays is to carefully align the projectors in the tiled array[4]. Indeed, this is very often the starting place even when additional methods are employed. This involves adjusting the position, orientation, and zoom for each projector until the desired pixel-to-pixel relationships are obtained. In simplest terms, and assuming that projector optics do not introduce any distortion, one can aim to align pixels in each tile so that they are aligned with the nearest pixels of neighboring projected tiles and introduce no tile-to-tile gap.

To aid in this process, several groups have designed mechanical positioning systems that allow fine reproducible control of as many as six degrees of freedom in position and orientation. These misalignments manifest themselves as tile-to-tile coordinate axes orientation and scaling differences, as well as keystoning of the tile image on the display surface. Such precision positioning jigs can be built for a fraction of the cost of the projector, are accurate to a fraction of a pixel, and are very stable. Without such devices, aligning tiled displays would be unwieldy for even the smallest arrays. They are amenable to motorization and therefore automatic control. With this addition, precision positioning systems offer a scalable solution to array alignment, though not without limitations. We know of no published work on the application of motorized alignment of tiled display systems, though many of the ingredients of such systems have been in use in other applications for a long time.

The chief deficiency not addressed by these positioning systems is image distortion by the projector optics. Even without optical distortion, achieving perfect alignment of all of the elements of a tiled array can be very difficult if not effectively impossible. Coupling between the many degrees of freedom of this system makes removing the last pixel of misalignment all but impossible for large arrays.

Another path to solving these problems has been pursued by projector manufacturers, at least in terms of correction if not measurement of alignment issues. Over time, features once only available in high-end projectors are becoming available in commodity units. Digital keystone correction has been available for a couple of years. Motorized lens shift, helpful in adjusting left-right and up-down adjustments on the screen, is now available on some models. We expect that some form of 2D image warping might become available in the near future. Integration of these features into a computer controllable interface with a well-defined API could help with the problems of tiling large numbers of projectors into single display systems.

Researchers at Princeton have devised a method that relies only on local comparative measurements[5]. An uncalibrated camera on a pan-tilt stage measures the pixel position and coordinate grid direction of neighboring projection tiles in their region of overlap. A complete circuit of such measurements, covering the entire mural, is fed to a simulated annealing algorithm to find a unifying mapping between the tile coordinates of each projector and some global coordinate system. The parameters of the mapping are the affine parameters equivalent to the effective pose of each projector. They also consider but don't extend their method to include a nonlinear model of the spherical aberrations in the projector lens. They express concern that the method may not always converge.

Researchers at Princeton and Carnegie Mellon University (CMU) have improved on this method by introducing a hierarchy of homographies relating pairs and groups of nearby projector coordinates to one another through the coordinate system of the uncalibrated camera[6]. The method is computationally less intensive than the simulated annealing, has no convergence problems, and scales better.

In another approach, one can completely sidestep the alignment process and simply measure the positions of the projectors "where they lie". Researchers at the University of North Carolina (UNC) have studied this approach. They have developed a measurement and correction system that works well and targets casually aligned and reconfigurable display systems[7,8]. They use a fixed calibrated camera to measure the location and mapping for each projector serially. Their method measures the projected image of a pattern of around ten Gaussian blobs positioned variously through the framebuffer. From this they calculate the coordinate mapping between the projected frame buffer coordinates and the calibrated camera coordinates. They use these measurements in two corrective schemes. In the first they extract a 3x3 affine projection model and use it in the geometry pipeline of their rendering engine to correct scenes. In the second, they use the measured coordinate map to warp the image using a suitably fine mesh and the 2D texture hardware of the rendering engine.

Related work at Honeywell[9] and the University of Maryland[10] has been done. Both of these approaches generate affine transformations of each tile in the array, similar to the Princeton and CMU correction techniques.

2. THE DOTTYTOTO WAY

Our goal in this section is to present the key ideas incorporated into DottyToto. We will present an overview of the several key features of the method. We then describe the implementation and the interesting technical parts of the process. This will expose the basic architecture of DottyToto, the key data products that drive the process, and some specifics of our current implementation of that architecture.

2.1. Overview of Salient Points

The key features and assumptions of our method are as follows.

Many dots. We densely sample the mapping between tile coordinates (labeled by pixel coordinates in the frame buffer driving the projector) and the camera coordinates with high-contrast features (in this case “dots”). This approach minimizes our sensitivity to photometric issues. We also benefit from the good statistics of the sheer number of features measured to determine a small number of model parameters. With DottyToto we are able to determine to subpixel accuracy the mapping of individual projector tile pixels to the unified coordinate system of the tiled array mural. Moreover, we are able to do this with a camera that can have far fewer pixels than the mural. For our experiments we have used both the Nikon CoolPix 950 (1600 x 1200) and the Fujifilm MX-2900 Zoom (1800 x 1200) to calibrate the ActiveMural, which has approximately 4800 x 2100 pixels.

Single uncalibrated camera. Using an uncalibrated camera allows us to use relatively inexpensive digital cameras with little regard to setup. We can, for example, zoom the camera casually to frame the display without needing to measure the relationship between ray angles and camera pixel position. Except for arranging that the image is unsaturated, little else needs to be considered.

Single pose. A single camera pose suffices for a large number of tiles; we have tested it on arrays of XGA projectors as large as 8 x 5. This circumvents the possible need for accurate camera pan-tilt mechanisms. Each frame contains tile-to-tile relative position information.

Few exposures. The total number of images exposed is minimized in our approach, the ideal being to capture all of the necessary measurement data in a single exposure of the entire tiled array. For unblended or minimally overlapped tiled displays (where tiles do not overlap with their neighbors), the number of exposures needed is two in the current implementation. This could be reduced to a single image with any of the following: somewhat more clever tile arrangement discovery methods, more a priori information passed to the program, better test pattern encoding, human intervention (to be avoided!), or a combination of any of these. For tiled displays with significant tile-to-tile overlap (10%), it is possible to generate excellent results with five images.

Direct solution. The image analysis is direct, involving no iteration either with the measurement process or internally as a part of converging to a solution. The solution comes from a least-squares minimization of the measurements made from the input images against a simple parameterized model of the display that accounts in a general way for projector and camera alignment and optical distortion in their lenses.

Aligned array. One assumption that we make in our measurement of the tile geometry is that the array of projectors has been aligned, though not necessarily perfectly. We make this assumption for several reasons. Pixels aren’t entirely free yet, so we attempt to preserve the underlying resolution of the individual projectors. Zoom, keystoning, and overlap can all degrade the final resolution. Variations in zoom from projector to projector and keystoning introduce pixel-scale variations across the mural. Overlap simply throws pixels away, and so we use it in a controlled way to manage blending. Our assumption of array alignment is used when we discover the general layout of the tiles in the array and again when we create a mural coordinate system.

Smooth and slowly varying. In order to get the highest accuracy in our pixel mappings, our method relies on the assumption that misalignment and distortions introduce only smooth and slowly varying changes to the coordinates. This ensures that combining the measurement of many features can be combined to reduce the measurement noise effectively without forcing a particular parameterized model.

2.2. Detailed Description of the Method

We use Matlab to implement our method as a series of processing steps each of which can be modified or replaced as superior techniques become available (illustrated in Figure 1). The analysis requires at most five snapshots of the tiled array with arrangements of two test patterns on the individual tiles. The test features of the two test patterns are described to the program to allow for a range of different patterns optimized for different situations. An example of such a description is shown in Figure 2. The output of the analysis is a parameterized mapping from each tile coordinate system (frame buffer pixel coordinates) to a unified coordinate system for the entire mural display.

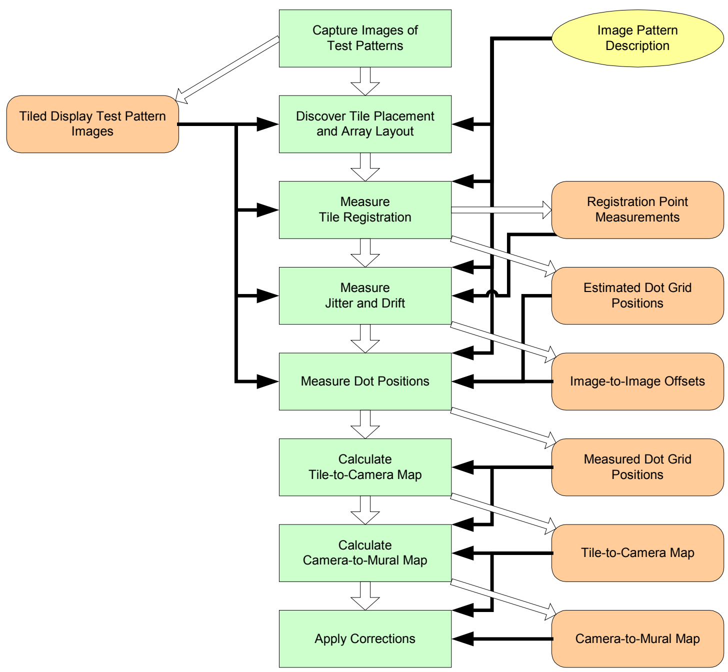

Figure 1 Schematic of the processing steps. The steps in the process are shown in the middle column and are represented by rectangles. Input to the process in the form of a geometrical description of the test pattern (see Figure 2) is shown as an oval. Rounded rectangles in the left and right columns are data products created and consumed in the process. Wide white arrows indicate process flow. Thinner white arrows indicate data creation. And, narrow black arrows indicate data consumption.

Tile layout discovery. We first determine the overall layout of the tiled array: total number of tiles, number of rows and columns in the array, and rough position of each tile in the camera coordinate system. A single image provides tile discovery, corner registration, and drift/jitter reference for correction in subsequent images (left panel of Figure 3, and Figure 5). The large monolithic rectangular region at the center of each projected area is bright enough to qualify as "bigger and brighter than anything else in the image", thereby simplifying the tile discovery process immensely. The image is thresholded and separated into disjoint objects. These tile blobs rise to the top of a list of objects sorted from largest (by pixel area) to smallest. After legitimate tiles in this list come objects that are discarded by a simple threshold based on size. This method is reasonably tolerant of oblique camera poses, which introduce size skew – objects further away will appear smaller and therefore sort to a lower position in the object list. It is fairly easy to extract total tile count with no a priori knowledge of the tiled array configuration. With a little more effort, it is possible to sort these tile objects by x and y coordinates to deduce the number of rows and columns. This is most easily achieved (as in our implementation) by assuming that the array is regular and that the camera is aligned with rows and columns along vertical and horizontal pixel coordinates. The brightness of both images in Figure 3 are arranged to be approximately the same to coerce the autogain on some cameras into the same range for all exposed images using any combination of these tile images across the array (e.g., Figures 5 and 6).

```
% Describe test pattern image features.
%
%       x increases from left to right
%       y increases from top to bottom
%   1. rectangular tile marker in tile coordinates
%       a. upper left corner (x=col, y=row)
%       b. size in pixels (width=cols, height=rows)
%   2. registration marks in tile coordinates
%       a. upper left corner (x=col, y=row) OF EACH, L->R, T->B
%       b. size in pixels (width=cols, height=rows)
%   3. vast array of dots in tile coordinates
%       a. upper left corner (x=col, y=row)
%       b. size in pixels (width=cols, height=rows)
%       c. dot repeat spacing (delx,dely)
%       d. valid range (for each tile if different)

    % general
img.tile.size = [1024,768];     % pix: width, height

    % discovery pattern
img.rect.ulc  = [250, 250];     % pix: col, row
img.rect.size = [500, 300];     % pix: width, height
img.reg.ulc   = [200, 200;      % pix: left-to-right,
                 800, 200;      %      top-to-bottom
                 200, 600;
                 800, 600];
img.reg.size  = [10, 10];       % pix: width, height

    % dot pattern
img.dots.ulc  = [0, 0];         % pix: col, row
img.dots.size = [10, 10];       % pix: width, height
img.dots.repeat = [20, 20];     % pix: horiz, vert
img.dots.num  = [51,39];         % dot: across, down

    % range of dots to include in the fit
img.dots.first = [1,1];    % dot: index of UL dot to include
img.dots.last =  [51,39];  % dot: index of LR dot to include
```

Figure 2 An example of a testpattern image description. Included is placement and size information for all of the features of the testpattern.

Tile registration. Bootstrapping from the general layout measurements, we measure registration points in each tile that will be used later to seed the detailed measurement process. Using the discovered tiles, identified by the large central blob, we find the coordinates of the corners of the rectangular blob. These corners are used to estimate the position of the four individual dots, which serve as registration for later steps in the process. A measured position for each of the four corner registration dots is recorded for each of the N tiles. The measurement is made by first constructing a template with known shape parameterized by scale factors determined from the estimated corner feature positions. The template is used locally, convolved with the image to find the peak measured to an accuracy of a fraction of a camera pixel.

Drift and jitter correction. For applications requiring a series of images, it is important to account for possible small image-to-image movements of the camera. For this purpose we use an average of the registration feature measurements, available in each image, as an indicator of overall image offset with respect to the first image.

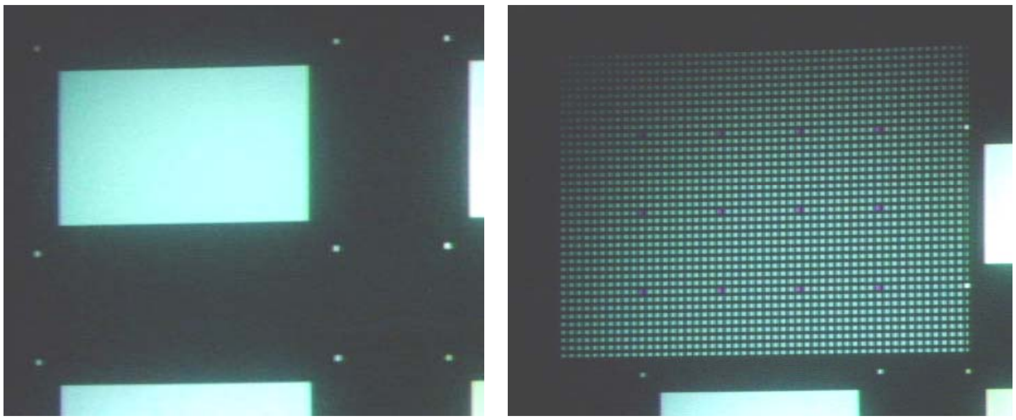

Figure 3 Examples of the two test pattern tile images used by DottyToto as seen on our 15-projector ActiveMural. The image on the left is designed to make identification of tile layout easy. The image on the right is used in the final measurement of the coordinate mapping. Figures 5 and 6 illustrate how these component patterns are arranged on the tiled array.

Dot measurement. The detailed mapping from tile frame buffer coordinates to camera pixel coordinates is accomplished by measuring a large number of features with known tile coordinates. The image (right panel of Figure 3) we use includes fiducial pixels marked distinctively, which are far enough from the edge of the tile to avoid blending regions. Corner registration marks from the tile discovery phase correspond to these special pixels in the dot grid. Bilinear interpolation and extrapolation provide estimates of the coordinates of all the dot features in the image. Armed with these estimates, we measure the individual dot positions in the same way that the corner registration features were measured: by autocorrelation with a constructed mask.

Parameterized tile-to-camera model. The dot measurements are fit to a model. In doing so, we gain signal to noise by smoothing over individual dot measurements, and we gain the ability to extrapolate coordinates beyond the available domain of measurable dots. This latter is useful when measuring blended overlapping tiles. All measurable dot features are included in least squares fit to a 10-parameter third-order model (for each of X and Y as in Eq. 1, below) of the mapping. Any other suitable model is acceptable so long as it is able to represent the distortion from both the projector and camera optics. The chosen model is simple to implement, computationally efficient, and sufficiently accurate. In this step, the statistical errors in the measurement are smoothed, resulting in a significantly improved estimate of the mapping between any tile coordinate and the camera coordinate system. This is due to the very large number of features measured compared with the small number of parameters needed to model the mapping. The third-order mapping from the k^{th} tile coordinates to camera coordinates, in each of X and Y, is represented by the following equation, where index pair *ij* takes on 10 values. For each tile *k* the 10 coefficients for the X mapping are given by A_k^{ij}, and the 10 coefficients for the Y mapping are given by B_k^{ij}.

$$\left(X,Y\right)_{camera} = \sum_{i=0}^{3}\sum_{j=0}^{i}\left(A_k^{ij}, B_k^{ij}\right)\cdot X_k^{i-j}\cdot Y_k^{j} \qquad (1)$$

Parameterized camera-to-mural model. We derive a mural coordinate system from the measured dots in the following way. We first estimate the relative offsets between neighboring pairs of projectors by comparing coordinates along their shared borders. From this simplified model of tile-to-tile coordinate transformation we create an approximate camera-to-mural mapping. This method assumes that the tiles are essentially undistorted, have been aligned, and have similar zoom; deviations from these assumptions are smoothed out in the next steps. We use the approximate mapping to project measured tile dot positions through camera coordinates to this estimated mural coordinate system. We then use the resulting multitude of dots to find a best fit in the least-squares sense of the measured dots in camera coordinates to their estimated mural coordinates. The fit produces a pair of 10-parameter third-order equations relating camera pixel coordinates to our derived mural coordinate system. The form is identical to the tile-to-camera mappings and is shown in Equation 2.

$$(X,Y)_{mural} = \sum_{i=0}^{3}\sum_{j=0}^{i}\left(A_{camera}^{ij}, B_{camera}^{ij}\right)\cdot X_{camera}^{i-j}\cdot Y_{camera}^{j} \qquad (2)$$

For improved immunity to numerical precision errors, the A and B parameters in both equations (1) and (2) can be referred to centered and normalized coordinates. That is, X_{camera} can be replaced by $(X_{camera} - X_o) / X_{norm}$, to range between 0 and 1, and similarly for Y_{camera}. The A_{camera} and B_{camera} coefficients will be generally better behaved.

Final output. The final results from the analysis are a set of third-order transformations, one for each of the tiles, and one taking camera coordinates into the unified coordinate system of the mural. The possible applications of these mappings have been enumerated earlier. Transformations from projector frame buffer pixel coordinates into the mural coordinate space are affected by composition of the two mappings.

3. RESULTS

In this section we first describe a few of the sources of error with which we are most concerned. Then we present results of a proof-of-concept test that demonstrates the accuracy of the techniques used in DottyToto. Finally we present results from a typical end-to-end application of DottyToto, including a sample of a corrected image.

3.1. Possible Error Sources

A naïve estimate tells us that DottyToto has a good chance of successfully measuring the coordinate mapping between projector frame buffer and mural coordinate system. Assuming random errors are responsible for the uncertainty in measurements of the position of a single dot, then many dots can be measured to reduce the error to a fraction of a projector pixel. For example, the Nikon CoolPix 950 with 1600 x 1200 pixels can image the entire ActiveMural with a scale of roughly three projector pixels to one camera pixel. If we can measure the position of a dot to better than one camera pixel, then by measuring the positions of ten dots we improve our estimate by the square root of 10, or to better than one-third of a camera pixel. This corresponds to better than a mural pixel. With many hundreds of dots per tile at our disposal it should be possible to accurately determine many more of the projector-to-camera coordinate mapping parameters than just the relative position.

Sampling the mapping densely with fine grain test pattern features improves measurement accuracy in many ways by reducing a number of potential sources of error. Among these advantages are easily measured high-contrast features, statistical leverage on low-order mapping, low camera resolution compared with display being measured, camera pixel errors (Bayer filter, etc.), pixel saturation and bleeding, insensitivity to camera lens distortion details, and insensitivity to electrical noise.

Limiting the number of measurement images reduces susceptibility to errors induced by frame-to-frame changes in camera position, camera zoom setting, and ambient lighting. Brightness-balanced test patterns with registration features (Figure 3) on all tiles for all images reduce frame-to-frame changes in exposure and enable accurate jitter and drift corrections.

3.2. Accuracy Test

The first question that we tried to answer was whether we had any chance of achieving the necessary accuracy with this method. In other words, can we easily detect subpixel displacements on the mural? To test this we took two pictures: a reference shot with the test pattern displayed in each of the fifteen projector tiles, and a shot with the test pattern shifted by small amounts in a few of the tiles. In this test we moved tiles 2, 6, 8, and 12, as summarized in Table 1. The second two columns of the table list the shift introduced in each tile of the test shot: most are unshifted.

Table 1 Summary of results from our initial test of DottyToto accuracy using the 5 x 3 projector array of the ActiveMural. Test patterns on four of fifteen projectors were shifted by the amounts shown.

Tile	***Actual Shift***		***Measured Shift***		***Pix Scale***
	X	Y	X	Y	Proj/Cam
1	0	0	0.0	0.0	2.81
2	0	1	-0.1	0.9	2.75
3	0	0	-0.2	-0.3	2.73
4	0	0	-0.1	-0.4	2.76
5	0	0	0.0	-0.5	2.84
6	8	0	8.0	0.2	2.76
7	0	0	0.1	-0.1	2.72
8	1	0	1.0	-0.2	2.71
9	0	0	0.1	-0.4	2.74
10	0	0	0.1	-0.6	2.87
11	0	0	0.1	0.0	2.74
12	1	2	1.1	1.9	2.72
13	0	0	0.1	-0.2	2.73
14	0	0	0.1	-0.4	2.80
15	0	0	0.2	-0.3	2.93

Each dot in the image is a 10 pixel by 10 pixel square in the projector image, separated from its neighboring dot by a 10 pixel black gutter. Figure 4 shows a detail of a representative portion of the dot swarm annotated with crosses (red) at the detected peaks.

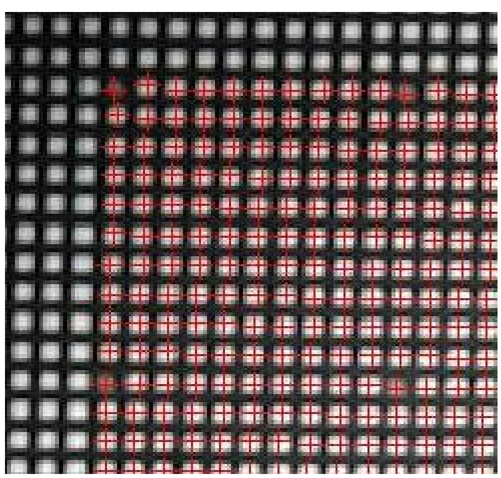

Figure 4 A detail of the dot pattern for one tile captured in a measurement image and adorned with crosses (red) indicating automatically measured positions.

We used as our measure of the center of each tile the average of all of the dot positions measured for that tile – 1271 for each of the 15 tiles in both the reference shot and the test shot. We corrected for image drift by referring tile centers to tile 0. We converted the measured shifts in camera coordinates into tile coordinates by applying a locally estimated conversion factor, shown in the last column of the table. Columns 4 and 5 summarize the measured shift for each tile. For example, we shifted the test pattern displayed in tile 2 by 1 pixel in the Y direction with respect to its reference position. We measured the shift to be –0.1 pixel in X and 0.9 pixel in Y.

Finally, we estimate the overall uncertainty in our measurement technique using the mean and the standard deviation of the difference between the measured and the actual shift. We find that DX = 0.04 +/- 0.11 pixels and DY = -0.24 +/- 0.21 pixels. Despite the fact that at the time this test was performed DottyToto employed a simple peak detector for its dot measurements, errors are well below a single pixel.

3.3. End-to-End

Having demonstrated that the ideas embedded in DottyToto have at least the hope of performing well on this difficult problem, we now present typical results for DottyToto measurements of the ActiveMural. For these results the ActiveMural was configured with hardware blending and approximately 10% overlap between neighboring projectors.

The tile discovery image is shown in Figure 5, and the four dot pattern images are shown in Figure 6.

Processing the five images with the Matlab implementation of DottyToto for this ActiveMural reduction on a 1.2 GHz Pentium III laptop took under 2.5 minutes. Details of the dot measurement are shown in Figure 7 for one of the tiles in the middle of the array.

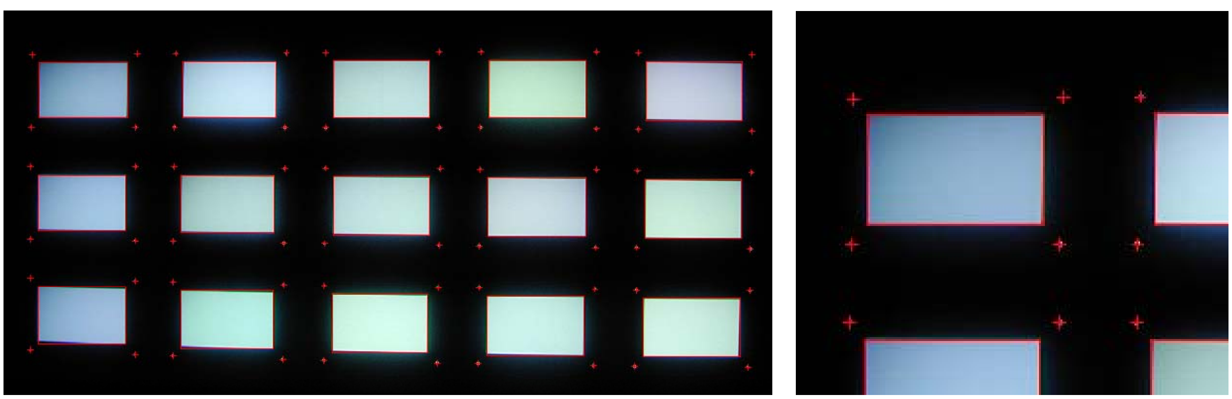

Figure 5 The tile discovery test pattern with tile and registration marks annotated by DottyToto. The discovered tiles are marked with bounding box rectangles (red). The subsequently measured registration dots are marked with crosses (red). A close-up of the upper left corner is shown in the right panel.

The resulting coordinate mappings can be explored by showing the distortions they illuminate with respect to a suitable center. Figure 8 shows one such representation. The 'x' grid is the undistorted uniform grid of pixel coordinates in the frame buffer of one of the projectors. The 'o' grid illustrates the mapping onto mural coordinates. It represents the shape of the projector illumination pattern on the mural. In this image, the mapping has been exaggerated by a factor of 10 to make the distortions easily discernable.

In addition, we have used the measured coordinate mappings to fill the frame buffers driving our projector wall with appropriately warped data. In Figure 9 we show a small portion of the ActiveMural with uncorrected and corrected image content. We have incorporated this technique into a real-time image correction solution[11] that also includes a method for correcting color and luminosity variations[12].

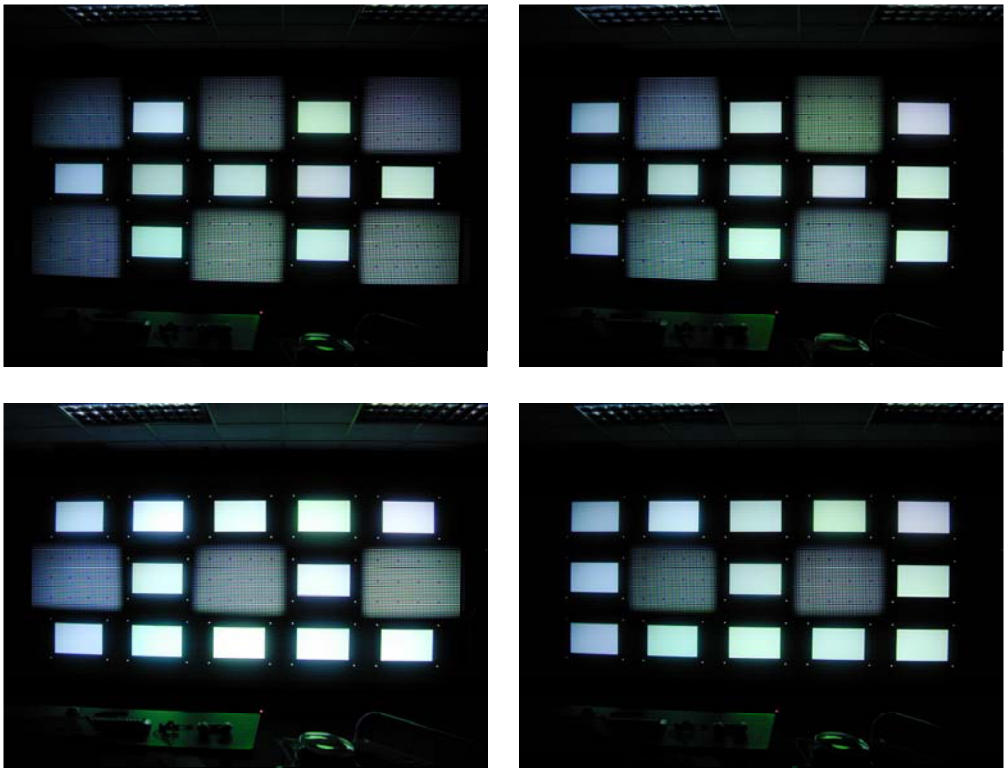

Figure 6 The four dot pattern images used by DottyToto to measure the fifteen projectors in the ActiveMural. Dense feature-rich single tile test patterns are alternated with simple white rectangular images to avoid confusion in the overlap region. Each tile is represented by a dense image in one of these four images. See Figure 7 for an annotated close-up of the dense pattern.

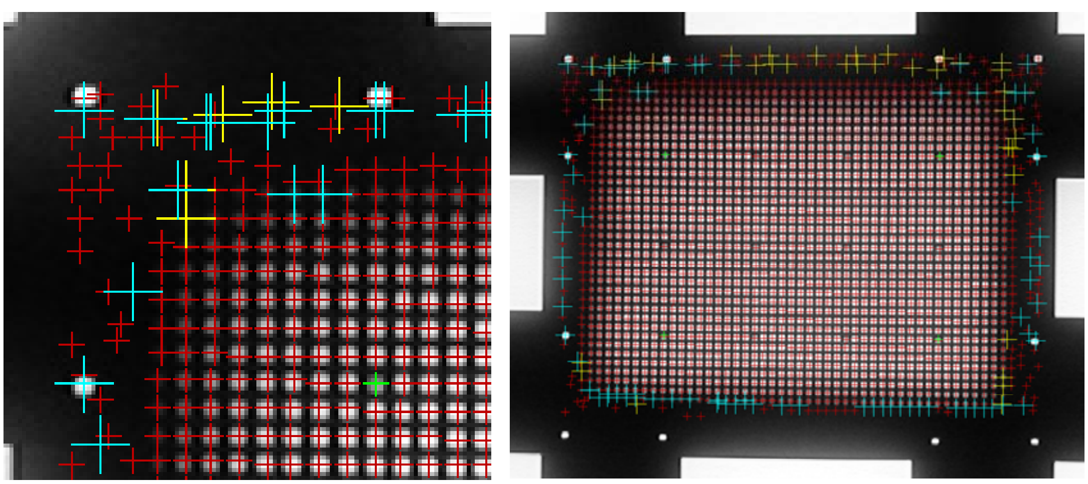

Figure 7 Automatic measurement of one of the tiles in the array: full tile on the right, detail of upper left corner on the left. For these measurements the ActiveMural was configured with hardware blending masks, which account for the fading of the dot pattern near the edge of this tile. The surrounding tiles are projecting the tile discovery pattern. The smaller crosses (red) are instances of successful dot detection and measurement; the larger crosses (yellow and blue) are instances of automatically determined anomalies, excluded from the fit.

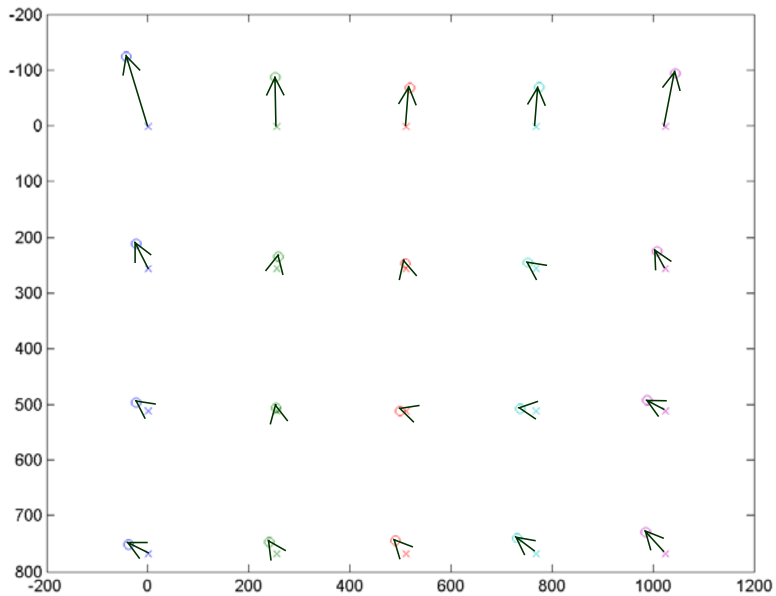

Figure 8 A representation of the warping measured between the pixels in a tile frame buffer and the pixels on the mural to which they map. A sampling of points in the frame buffer on a 5x4 grid is marked with ‘x’. The axes are labeled with frame buffer pixel coordinates (the display is 1024 x 768). The arrow shows the direction that the grid point is displaced by the combination of projector alignment and optical distortion. The ‘o’ marks the relative position in mural coordinates of the corresponding frame buffer point. The length of the arrows has been multiplied by a factor of 10 to make the effect more visible. One can easily see the effect of pincushion distortion stretching the upper left and upper right corners away from the optical center.

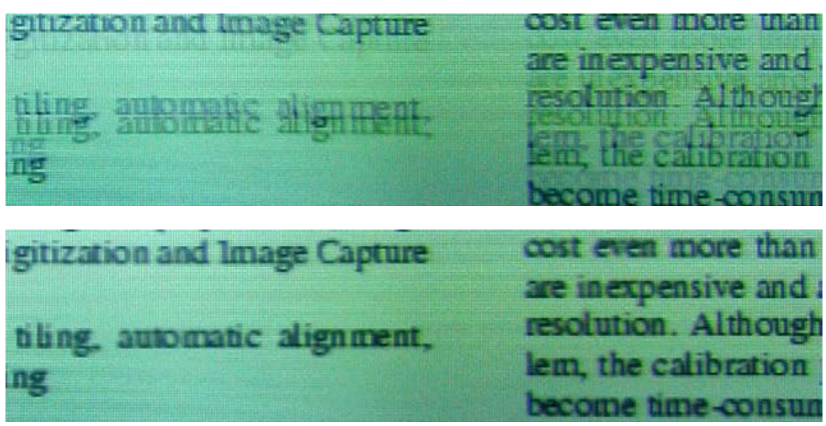

Figure 9 An example of the results of correcting images using DottyToto measured coordinate transformations. The top segment shows a 4” x 3/4” (120 pixel x 22 pixel) portion of the ActiveMural screen taken of an uncorrected image in an area of overlap between four projectors. The bottom segment shows the same portion after correction. The ActiveMural is 8 feet tall by 16 feet wide. The camera was approximately 16 feet from the screen.

4. CONCLUSIONS

We have described a technique capable of measuring to sub-pixel accuracy the positions of all pixels in all tiles of large multi-projector display system using as few as one image from a commodity digital camera. We demonstrated its application using a 1600 x 1200 pixel camera to measure and correct a 4800 x 2100 pixel display system, made from the slightly overlapping images tiles of 15 projectors.

The current implementation works well for our present needs. There are a few areas that would benefit from additional experiment and development work. Among these are:

- Optimize the image acquisition and analysis for speed. While drift in tiled displays is slow and therefore places only modest demands on turnaround time for calibration systems, we anticipate that for some applications real-time measurement, analysis, and correction would be useful.

- Integrate implementation of real time image warp corrections into a wider range of existing applications and development tools. The goal here is complete transparency to all applications that use the large pixel plane of a tiled display.

- Develop a tool to help in the initial alignment of tiled displays along the lines of the *automated alignment* and *manual alignment* applications discussed in the paper.

ACKNOWLEDGEMENTS

We acknowledge useful discussions with the groups at Princeton and the University of North Carolina. This work was supported by the Mathematical, Information, and Computational Sciences Division subprogram of the Office of Advanced Scientific Computing Research, U.S. Department of Energy, under Contract W-31-109-Eng-38 and in part by DOE ASCI VIEWS program.

REFERENCES

1. Theo Mayer, "Design considerations and applications for innovative display options using projector arrays", *Proceedings of SPIE Projection Displays II*, Vol. 2650, pp. 131-139, 1996.

2. Ramesh Raskar, Greg Welch, Matt Cutts, Adam Lake, Lev Stesin, and Henry Fuchs, "The Office of the Future: A unified approach to image-based modeling and spatially immersive displays", *Computer Graphics*, 32:179-188, 1998.

3. Rajeev J. Surati,. "A scalable self-calibrating technology for seamless large-scale displays", PhD thesis, Department of Electrical Engineering and Computer Sceince, Massachussetts Institute of Technology, 1999.

4. Mark Hereld, Ivan R. Judson, and Rick L. Stevens, "Introduction to building projection-based tiled display systems," *IEEE Computer Graphics & Applications*, vol. 20, pp. 22 - 28, 2000.

5. Yuqun Chen, Douglas W. Clark, Adam Finkelstein, Timothy C. Housel, and Kai Li, "Automatic alignment of high resolution multiprojector displays using an un-calibrated camera", in *Proceedings of IEEE Visualization Conference 2000,* pp. 125-130.

6. Han Chen, Rahul Sukthankar, Grant Wallace, and Kai Li, "Scalable alignment of large-format multi-projector displays using camera homography trees", to appear in *Proceedings of IEEE Visualization 2002*.

7. Ramesh Raskar, Michael S. Brown, Ruigang Yang, Wei-Chao Chen, Greg Welch, Herman Towles, Brent Seales, and Henry Fuchs, “Multi-projector displays using camera-based registration", *Proceedings of IEEE Visualization 99,* San Francisco, CA, pp.161-168, 1999.

8. Ruigang Yang, David Gotz, Justin Hensley, Herman Towles, and Mike Brown, “PixelFlex: A reconfigurable multi-projector display system", *Proceedings of IEEE Visualization 2001*, San Diego, CA, USA, October 2001.

9. C. J. Chen and Mike Johnson, “Fundamentals of scalable high resolution seamlessly tiled projection system”, in *Proceedings of SPIE Projection Displays VII*, Vol. 4294, pp. 67-74, 2001.

10. Zhiyun Li and Amitabh Varshney, “A real-time seamless tiled display system for 3D graphics”, presented at the *Immersive Projection Technology Symposium of the IEEE Virtual Reality 2002 Conference (VR2002 IPT)*, March 2002.

11. J. Binns, G. Gill, M. Hereld, D. Jones, I. Judson, T. Leggett, A. Majumder, M. McCrory, M. E. Papka, and R. Stevens, "Applying geometry and color correction to tiled display walls," a poster presented at *Visualization 2002*, Boston, MA, October 2002.

12. A. Majumder and R. Stevens, "LAM: Luminance Attenuation Map for photometric uniformity in projection based displays," Argonne National Laboratory, Argonne, Preprint ANL/MCS-P992-0902, September 2002.