Fully automatic multi-projector calibration with an uncalibrated camera

Ignacio Garcia-Dorado
McGill University, Montreal, Canada
ignacio.garcia.dorado@gmail.com

Jeremy Cooperstock
McGill University, Montreal, Canada
http://www.cim.mcgill.ca/~jer

Abstract

Multiple video projectors can be used to provide a seamless, undistorted image or video over one or more display surfaces. Correct rendering requires calibration of the projectors with respect to these surface(s) and an efficient mechanism to distribute and warp the frame buffer data to the projectors. Typically, the calibration process involves some degree of manual intervention or embedding of optical sensors in the display surface itself, neither of which is practical for general deployment by non-technical users. We show that an effective result can in fact be achieved without such intervention or hardware augmentation, allowing for a fully automatic multi-projector calibration that requires nothing more than a low-cost uncalibrated camera and the placement of paper markers to delimit the boundaries of the desired display region. Both geometric and intensity calibration are performed by projection of gray-coded binary patterns, observed by the camera. Finally, the frame buffer contents for display are distributed in real time by a remote desktop transport to multiple rendering machines, connected to the various projectors.[1]

1. Introduction

The ubiquity of small, low-cost, high resolution projectors has enabled wide-scale deployment of large display environments that were previously limited to the screen space of desktop CRTs and LCDs. By default, the location and orientation of a projector must be defined carefully and maintained to ensure that the optical axis is perpendicular to a single, planar display surface. Otherwise, the resulting image suffers from keystone effects and appears distorted. Although many projectors now include keystone correction controls, these are often limited by optical constraints. More problematically, the physical realities of mechanical mountings and thermal variations [16] and vibration necessitate periodic adjustment and realignment. Often, users simply accept a certain amount of misalignment in order to avoid the need for manual recalibration.

Of particular interest is the use of multiple commodity projectors in parallel to achieve even higher resolutions in display walls or immersive (CAVE-like) environments, whether for information visualization, education, or entertainment. In these settings, projector calibration is more important, since misalignment results in visible overlap or gaps between images, and variation in projector intensities and color response produces seams at the junctions. Typically, multi-projector alignment is a complex, sensitive process that requires the services of a skilled, experienced professional.

One fully automated and several semi-automated methods have been proposed for this problem, ranging from structured light calibration to other computer vision techniques, but none allow for a fast, pixel-accurate solution, with a simple deployment by a non-technical user, using commodity hardware alone. This motivated our development of a robust, easy-to-use, COTS-based, fully automated calibration system that can generate a seamless, aligned, multi-projector display on any reasonable configuration of planar display surfaces. The only requirements are the use of simple markers to define the borders of the target display area and the placement of an uncalibrated camera at a position from which the entire display surface is visible. From there, our system self-calibrates to create a uniform intensity, undistorted, continuous projected display that fills the target area, insofar as is possible. Significantly, the solution does not require human intervention to manipulate the equipment and avoids the often painful step of camera calibration. Moreover, the system has been demonstrated as scalable to a distributed architecture, with individual clients connected to the multiple projectors, rendering the frame buffer computed at the server.[2]

The remainder of the paper is organized as follows. First, Section 2 summarizes previous solutions from the literature. Next, our solution is presented, beginning with a description

[1] A demonstration of the system is available from http://www.youtube.com/watch?v=G1X3G1j9mus.

[2] However, a bug in the third-party (NoMachine) rendering code presently results in significantly reduced frame rate when multiple clients connect to the server in *shadowing* mode. This is anticipated to be corrected in an upcoming release.

of the system layout in Section 3, including the computer vision techniques used for geometric calibration and brightness correction, and the implementation of a distributed architecture for real-world deployment, in Section 4. Finally, Section 5 summarizes future work and concludes the paper.

2. Literature Review

Relevant prior work in the area of projector calibration consists primarily of computer vision techniques in association with one or more cameras that view the projected output. The various approaches differ mostly in their camera calibration requirements and the type of projection pattern used during the calibration process. Regardless of the method, all approaches calculate the intrinsic and extrinsic parameters of the projector, then pre-warp the projector frame buffer so that it appears correctly on the display surface.

2.1. Calibrated Camera Approaches

In general, methods that require a calibrated camera cannot by fully automated, given the need for manual intervention in this process. Most such systems perform camera calibration using Zhang's method [24], which consists of image matching using geometric constraints and a least median of squares. Typically, a checkerboard pattern is displayed in at least two different positions to provide the necessary inputs. The method then calculates the homographies for the intrinsic and extrinsic parameters. For this step, the display must appear rectangular on the target surface from the perspective of the camera and, in some cases, to be able to determine the projector-camera homography where the intrinsic projector parameters are known. As an alternative to the traditional checkerboard pattern, it is possible to use patterns that contain more information in each code, and thus, reduce the number of required calibration patterns to obtain the projector parameters. For example, Fiala [7] employed augmented reality tags (ARTags) viewed by a calibrated camera, which allowed him to use only three ARTag patterns rather than the 18 Gray code images previously required, resulting in significant speed-up.

Raskar et al. [18] used two cameras to recover a 3D representation of the display environment, allowing for the more general case of an arbitrarily complex display surface geometry. The cameras were first calibrated using a pattern printed on a cube, rather than the traditional checkerboard. This allows two different homographies to be obtained simultaneously. Next, the display geometry was determined using structured light projection, resulting in the generation of a 3D mesh. Alpha masking was used to improve transitions in regions of overlap. This process took anywhere from one to fifteen minutes per projector, depending on the sampling density used. Raskar also presented a method that uses a projector-camera pair, with an attached tilt sensor, to calibrate the system and render keystone-corrected rectangular images [17]. This approach required only a single camera and a blank wall as a target display surface for the calibration pattern thereby obtaining improved accuracy at the cost of requiring additional correspondences to obtain the homography.

Raij et al. [15] proposed using relative pose estimation and structured light to find the correspondences between the defined display area and a calibrated camera. To avoid the need for fiducial markers to define the display area, they assume that the x-axis of the image is parallel to the ground; this permits determination of the vertical plane by calculation of a cross product. Since the method involves an iterative minimization, reasonable initial values are required in order for the system to converge to a solution.

2.2. Uncalibrated Camera Approaches

Camera calibration, of course, generally requires manual involvement. For the more interesting case of systems that avoid the need for a calibrated camera, several approaches have been proposed. However, these require either considerable calibration time, uniform color display areas, only a single planar display, or non-trivial augmentation of the environment, e.g., the addition of fiber optics [10], to achieve high accuracy.

For the special case of a single, moving user, a single-pass rendering process was described [16]. This used the collineation between points of the screen and the projector pixels to display images that appear correct from the viewer's perspective. Collineation is a one-to-one mapping from one projective space to another that can be expressed as a matrix transformation. In this case, the first projector is used to describe the transformation of the projected image, so that it is seen correctly from the user's point of view. This is calculated using eight points. The mapping between the first and the remaining projectors is computed observing individual pixels. This allows for creation of larger displays using a two-pass rendering process. For such an approach, the first step involves determining the regions of overlapping projection as necessary to compute the appropriate transformation matrices. In an immersive environment such as a CAVE, this simply requires determining which projector pixels illuminate the corner markers, a process that can be performed by activating each projector in turn and observing the designated screen positions with an uncalibrated camera. Such a system, of course, requires integration of an accurate 3D tracker and is not appropriate for rendering a display suitable for multi-user viewing.

For their multi-projector system, Chen et al. [4] began with an assumption of approximately aligned projectors, with no more than a ten-pixel misalignment between them. Next, they projected a series of grid patterns and points to measure the misalignment between projectors. A global

minimization algorithm was then applied to obtain a corrective transformation that aligns the overlapping projectors. Their system achieved pixel precision with only a 0.1% error. Subsequently, they decreased the calibration time to a few minutes, using an approach similar to the one we present here, also involving use of PTZ camera [3]. However, their method requires a single, uniform, planar display surface and the projectors must be positioned such that the display misalignment between them is less than ten pixels, neither of which is an acceptable constraint for our purposes.

Sukthankar et al. [22] calibrated a single projector using an un-calibrated camera in just few seconds to within a maximum error of 0.8%. Their technique further assumed that the screens appear in the camera image as uniformly illuminated objects with edges that are visible against the planar background. The projector-camera mapping was first estimated using a bright, projected rectangle, whose corners, as seen from the camera, were used as features to recover the homography. The calibration was subsequently refined using a display of white circles whose centers were estimated and compared to the predicted locations in the camera image. Finally, the maximum available projection area was defined by a greedy algorithm that segmented the camera image, grouping adjacent pixels with similar intensities. Rehg et al. [19] presented a similar system, although not intended for a single, large display, but instead, designed to project a fully overlapping region that was illuminated by all projectors, stable even when a projector, camera or screen was slightly moved. Moreover, their system was able to remove shadows cast by the user without requiring 3D localization of the components and users. This was achieved by applying an alpha mask to each projector, computed from the reference and observed images. Similar to us, they used four fiducials to define the desired display area. However, unlike our approach, they imposed additional constraints of a uniformly illuminated background and all markers simultaneously visible to the camera, resulting in a non-trivial error of 2.5% (tens of pixels).

Lee et al. [10] used embedded optical sensors in the corners of the display surface in conjunction with a projected Gray code pattern to determine the display geometry and to improve the speed of the projector calibration process. This data was then used to pre-warp the projected image so that it fit the target area delimited by the sensors. The entire calibration process was extremely fast, limited only by the display refresh rate and the response time of the optical sensors. However, since the optical sensors must be embedded into the screen, this system is impractical for display walls. Their system achieved an error less than 0.3%.

To determine the intrinsic parameters, Draréni et al. [5], moved the projector between at least four different positions and orientations, with the projected checkerboard display observed by the camera. This provided many highly accurate point matches and allowed them to achieve a wall-projector calibration accuracy on par with a direct method based on a printed planar grid fixed to the wall.

Draréni et al. [6] also presented a method that did not require a physical calibration board or full knowledge of the camera parameters to determine the wall-camera homography. Instead, they explored the space of all possible orientations of the hemisphere and selected the direction that minimized the reprojection error as the correct plane orientation. This also obviated the need for markers to define the projection surface. This approach satisfies almost of our requirements, but the need for manual involvement in the projector calibration could prove problematic for many installations, e.g., where projectors are mounted semi-rigidly.

Okatani et al. [13] described the calculation of a projector-screen-camera homography from multiple images when the projectors and the projected patterns are known. In more recent work [14], they used this approach to calibrate a multi-projector display system using a hand-held six-megapixel camera. In their case, the projected area was required to be a unique planar screen. To calculate the unique solution, either four projectors, placed in a perpendicular grid layout or three projectors and a partially calibrated camera were required. In the latter case, camera calibration involved both intrinsic and extrinsic parameters, apart from the focal length, for which only an initial estimation was needed. Their method is fast and accurate, obtaining an average error of 0.3%. However, some prior knowledge is required concerning the layout of the projectors and the desired image rotation. A further drawback is that this approach does not solve the situation where the projected image exceeds the available area.

3. System Overview

Our solution to remove the human from the loop and obtain faster calibration begins with affixing visible markers at the corners of the projected display that limit the desired boundaries of the display region. This one simple addition allows us to employ an uncalibrated camera and still achieve pixel precision of the multi-projector wall display. Geometry information obtained by the camera is then used to pre-warp the contents of the graphics display according to the projector-wall homography. Finally, intensity correction is performed to create an undistorted, uniformly illuminated output. This process relies on two assumptions: the camera and projector optics can be modeled by a perspective transformation and the display area corresponding to the output of each projector is planar or can be defined as a geometric figure, drawn in OpenGL.[3]

[3]Our initial implementation simply uses the rectangular texture from the framebuffer, transformed by the projector-wall homography.

This implementation has been evaluated with four Sanyo PLC-EF30 projectors of 1280x1024 pixel resolution, illuminating a three-wall environment, and an NTSC Sony EVI-D30 camera, arranged as illustrated in Fig. 1.

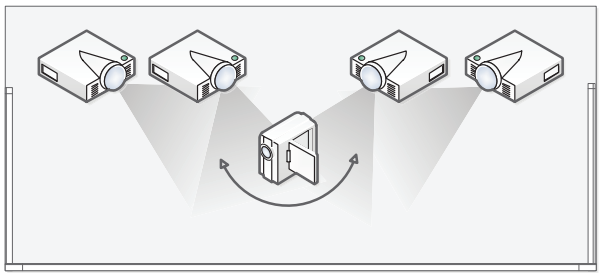


Figure 1. Our multi-projector layout

3.1. Marker placement and detection

We manually place the fiducials, which must be within the region "reachable" by the projectors, to delimit the desired display area. The accuracy of their placement affects only the definition of the boundaries of this region, which, for a horizontal arrangement of projectors, is generally relevant only for those projectors responsible for the left and right edges of the overall display.

The position and pose of fiduciary markers [1] relative to the camera can be tracked with various libraries such as ARToolkit.[4] Because of the complexity of the associated visual patterns, the markers need to occupy a minimum 14 × 14 pixel region in the camera frame to be detected and decoded [23]. For our configuration, with a fixed camera position viewing the entire scene from a distance of approximately 3 m, this implies a marker size of 10×10 cm. Although this can be reduced by approximately 20% if we forgo decoding of the marker ID, this is still awkwardly large. Instead, we opted to employ a trivially simple marker, consisting of two black squares, which can be detected using basic pattern matching, applied to the output of a Canny edge detector [2], at a resolution of 6×6 pixels or approximately 4×4 cm.

We use an uncalibrated camera to locate the markers, placed around the boundaries of the desired projection region. Since the field of view of our layout is considerably wider than 90^o, we opted to use a pan-tilt-zoom (PTZ) camera to locate each of the markers progressively, choosing a focal length such that at least two of the markers are contained within the field of view at sufficient resolution as described above. Our current implementation requires approximately three minutes to perform the calibration for a four-projector system. Increased calibration accuracy can be obtained through the use of a higher zoom factor at the expense of greater calibration time.

[4] http://www.hitl.washington.edu/artoolkit

3.2. Projector-wall calibration

Once the relationship between display geometry and camera parameters is established, our system uses a structured light calibration approach to determine the projector-wall homography, that is, the mapping between pixels in projector space (x, y) and the corresponding positions (X, Y) on the target surface. For this process, we set the camera to a minimum focal length to maximize the visible area. As a compromise between speed and resolution at which the markers are viewed under these conditions, we require that two markers are visible to the camera at each position, thus allowing for calibration of two points simultaneously. If necessary, zoom control can be used to increase the viewed resolution of each marker in order to achieve pixel accuracy, at the cost of a longer calibration time.

For the camera viewing this display, averaging or median filtering of multiple frames of video input is required to cope with the transition between successive patterns, which are not registered by the camera as instantaneous. This has the additional benefit of reducing the effects of camera noise.

Various options exist for structured light patterns to identify the correspondences between surface points and pixels of the projected pattern, as described in the survey by Salvi et al. [20]. Time-multiplexing techniques, using Gray code and/or phase shifting, offer the benefits of easy implementation, high spatial resolution, and the ability to achieve highly accurate measurements. To cope with moving surfaces, spatial neighborhood techniques can be considered, as these concentrate the pixels needed to identify the given codeword into a local region. However, for our system, which assumes a static surface, this is unnecessary. Finally, direct coding techniques are useful for achieving high spatial resolution with few projected patterns. However, the limited bandwidth of LCD projectors and noise limitations of the imaging system result in integration of intensity values, complicating the correct identification of every projected pixel value.

Technique	Steps	Lights ON	Lights OFF
Gray-coded Pattern	17	0%	0%
Sweeping Line 1Px	2304	0.99%	0.07%
Sweeping Line 2Px	1152	0.10%	0.12%
Pattern and Sweep	29	0.08%	0.15%

Table 1. Camera error as a function of ambient light.

In our configuration, the camera resolution is approximately one quarter that of the projectors, suggesting that without zoom control, we might be limited to a best-case error of four (projector) pixels. However, our calibration process overcomes this limitation and reduces the error as required, at least to an accuracy of two pixels using the line-sweep method described below, and to single pixel accuracy

using zoom control. We selected our calibration technique based on the results of a single-marker test, with five repetitions per technique, shown in Table 1. Gray coding was the only method that achieved zero error, and did so using the fewest number of necessary projected patterns.

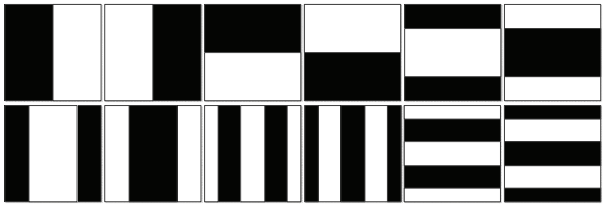


Figure 2. The first twelve gray-coded binary patterns

In order to calibrate at the full projector resolution independently of camera resolution and zoom value, we employ a combination of Gray code binary patterns, shown in Fig. 2, and a line sweeping approach, inspired by phase shifting. In theory, we could simply determine a suitable average intensity for "on" values of the pixels in the displayed patterns. However, given the variance of ambient illumination, we found that in some cases, the difference between states "on" and "off" states could be as small as 2%.[5] This made determination of a suitable threshold impractical. Instead, we first display each pattern, followed by its inverse. In this manner, the system does not require prior knowledge of the intensity value of an illuminated pixel, but can simply use differencing to determine the state of each pixel.

After displaying the Gray code binary patterns, we take advantage of our multiple projector configuration to increase the accuracy of the marker position estimation. This is done by having both projectors sweep a line in their overlapping corners, first vertically and then horizontally, delimited by the information obtained through the Gray code pattern detection. The position that yields the greatest luminance intensity as measured by the camera sensor, as shown in Fig. 3, corresponds to the overlapping location between the two lines. This achieves a position estimation closer to the full resolution of the projectors, even though the camera has lower resolution.[6] This improves upon the raw results from the Gray code pattern detection while avoiding the need for slower phase shifting techniques.

3.3. Homography and Compiz Plug-in

The projector-wall homography can be expressed by a single and unitary projective transform with eight degrees

[5]We explain this surprising result by the placement of the projectors (at varying distances from the walls), which required physical keystone correction, differences in luminance output, intensity variation across the projected frame, and the camera's position relative to the displays.

[6]This relies on the assumption that neither the camera pixels nor the pixels of the two projectors are perfectly aligned with each other. Otherwise, maximum intensity would also be observed at a single camera pixel even if the two projectors displayed lines offset by a single projector pixel.

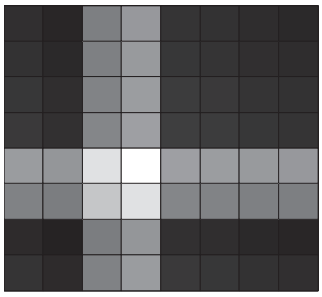


Figure 3. Luminance measurement along the sweeping line

of freedom $\vec{p} = (p_1 ... p_9)$:

$$(x, y) = \left(\frac{p_1 X + p_2 Y + p_3}{p_7 X + p_8 Y + p_9}, \frac{p_4 X + p_5 Y + p_6}{p_7 X + p_8 Y + p_9} \right) \quad (1)$$

or in homogeneous coordinates:

$$\begin{pmatrix} xw \\ yw \\ w \end{pmatrix} = \begin{pmatrix} p_1 & p_2 & p_3 \\ p_4 & p_5 & p_6 \\ p_7 & p_8 & p_9 \end{pmatrix} \begin{pmatrix} X \\ Y \\ 1 \end{pmatrix} \quad (2)$$

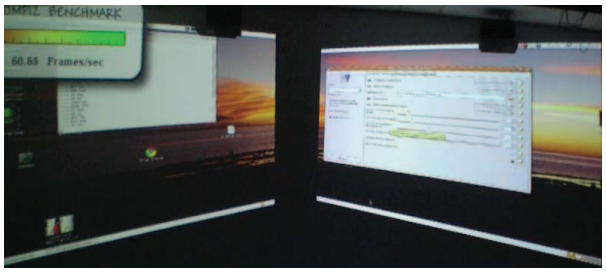
These equations can be solved through Gaussian Elimination using as few as four pixel correspondences, e.g., the corners of the projector output and their corresponding coordinates in the structured light pattern. We implement the homography as a matrix transformation in OpenGL through the Compiz plug-in[7] that warps the complete desktop to render the calibrated, projected display as shown in Fig. 4. This operation can be performed in real-time on low-cost hardware, provided only that the graphics card memory is at least the double the size of the frame buffer.

3.4. Overlap correction

Since the calibration process is imperfect, in part due to the limited resolution of the camera, some display regions are likely to be illuminated by multiple overlapping projectors, resulting in obviously brighter contents, or worse, not illuminated by any projector, resulting in a visible gap.[8] Fortunately, edge attenuation and brightness equalization techniques have been reported to allow an error of as much as 5–6 pixel in the display as tolerable [12]. We employ an alpha blending technique to achieve an effectively seamless overlap region, assigning to each projector an alpha mask with intensity in $[0, 1]$. The blending is then performed in real time as part of our Compiz plug-in. We found that an

[7]Compiz is the default OpenGL compositing manager on many Linux distributions, which uses GLX_EXT_texture_from_pixmap for binding redirected top-level windows to texture objects. It offers a flexible plug-in architecture and runs on most graphics hardware.

[8]We assume that initial positioning of the projectors can resolve this latter problem, ensuring that all necessary display locations are illuminated by at least one projector.



(a) Before correction.

(b) After correction.

Figure 4. Keystone correction achieved by application of the projector-wall homography through the Compiz plugin.

overlap of ten pixels between projectors is sufficient to ensure an imperceptible transition, provided that the vertical misalignment does not exceed three pixels.

3.5. Non-uniform brightness correction

The final issue to address is variation of intensity and color. These effects are typically evident as inter-projector variations, but also manifest as intra-projector variations, that is, across the pixels of a single projector, often due to lamp design and lens optics. Correction of such variations has been investigated by previous research [12, 11, 21]. Since human viewers are most sensitive to luminance variation at boundaries [9], we confine our efforts to inter- and intra-luminance normalization to those regions, both in cases for which a single projector is responsible and in areas of overlap between multiple projectors.

Inspired by the approach of Majumder and Stevens [12] and borrowing from their terminology, let $L_d(x_d, y_d)$ be the luminance response of the maximum input, i.e., the measured brightness at pixel (x_d, y_d) in display coordinates, with all projectors responsible for that pixel illuminating it at maximum intensity, and L_{min} be the Common Achievable Response, i.e., the minimum of $L_d(x_d, y_d)$.

However, to avoid unduly limiting the dynamic range of our display, we balance brightness *after* the system is calibrated. This ensures that we ignore regions adjacent to the display area, such as the ceiling and lower walls, which would otherwise affect the calculations. Moreover, we assume a semi-Lambertian projection surface, allowing for some specularities, but also assume that some locations will reflect very little light due to surface defects. This suggests that attenuating projector outputs based on L_{min} would be excessive. Instead, we replace L_{min} with $L_{10\%}$, the first decile value of $L_d(x_d, y_d)$. Thus, the Luminance Attenuation Map, $A_d(x_d, y_d)$, which provides the attenuation value of L_d for each pixel, is calculated as:

$$A_d(x_d, y_d) = \begin{cases} 1, & L_d(x_d, y_d) \leq L_{10\%} \\ \frac{L_{10\%}}{L_d(x_d, y_d)}, & L_d(x_d, y_d) > L_{10\%} \end{cases} \quad (3)$$

This represents a small sacrifice in intensity equalization for the objective of increasing the overall display brightness. An example result from this equalization process is provided in Fig. 5.

(a) Original, non-uniform output.

(b) Equalized illumination intensity output display.

Figure 5. Brightness correction.

Using the positions obtained from the geometric calibration step, we project full white at the corners and average the observed intensity over a few frames. Images are acquired at minimum aperture over a long time exposure time to reduce noise while avoiding saturating the sensor. The results are then used to map to the projector frame buffer, applied as an alpha channel texture, following Eq. (3).

4. Distributed architecture

For scalability, dedicated hardware solutions, such as the Christie Vista Spyder, exist for a fixed maximum number of projectors. Another solution built with once-high-end SGI Onyx2 hardware using a customized wire OpenGL protocol was described [8]. While this was suitable for GL-rendered graphics, our distributed rendering architecture is instead intended to support generic display content. The architecture is deployed with multiple client computers, each connected to a single projector, as illustrated in Fig. 6. A server performs all the graphic computations at the target resolution and communicates its output to a set of clients via remote desktop software. The clients receive the full desktop

frame buffer and make use of another Compiz plug-in we designed to crop and transform the portion of the remote desktop window for which each is responsible. The server need not remain agnostic of the client-projector geometry, and thus, could, in theory, transmit to each client only that portion of its frame buffer that the client requires. This would significantly reduce communication requirements. Unfortunately, this is not feasible, either for reasons of architecture or performance, with current distributed display protocols. For example, the `Xdmx` protocol provides both control over distributed machines and remote rendering of (portions of) the display buffer, but it runs extremely slow when rendering anything other than simple text consoles. The effort of redesigning the server and client of such a protocol for our purpose remains beyond the scope of our work.

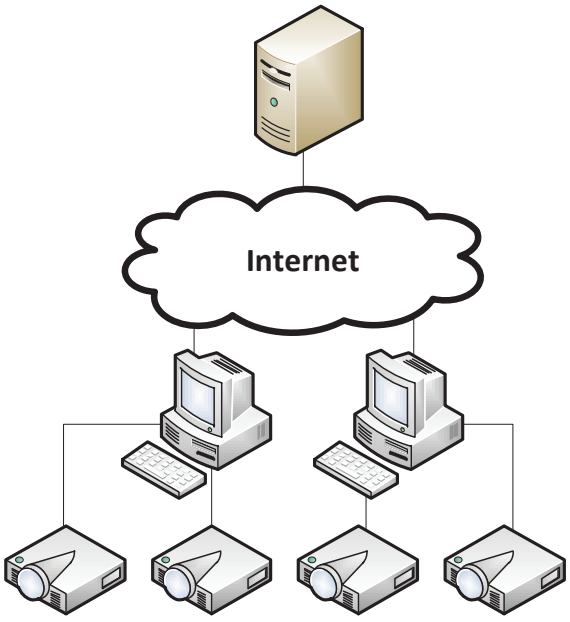


Figure 6. A possible network layout

Possibilities included X11 combined with SSH forwarding, Sun Microsystems' ALP, for which no working implementations could be found, Microsoft's proprietary RDP, and the simple RFB protocol, used by the popular VNC software. These protocols all operate at the frame buffer level, i.e., they share the *entire* frame buffer with any connected client(s). The most promising solution we found was NX, which compresses X11 data using a differential technique derived from DXPC,[9] which exploits various spatial and temporal redundancies to improve bandwidth efficiency and maximize responsiveness. A side-by-side comparison of the relevant features for each of the remote desktop protocols is provided in Table 2, based on results on a 100 Mb/s LAN environment using a 5120×1024 target display resolution, produced at the server side.

The solution should ideally be open source, or offer sufficient SDK support to facilitate further extensions as necessary. Our choice of remote desktop software for this purpose was constrained largely by questions of support for simultaneous connection of multiple clients and encoding efficiency. Specifically, it had to provide a sufficiently streamlined data transfer pipeline to operate at reasonable frame rates, at modest bandwidth, even for high resolutions. Most of the systems we investigated were unsuitable for these reasons. The only viable solution that satisfied our requirements was NX, as its "shadow" feature allows multiple simultaneous client connections to the same session.[10]

Software	**Protocol**	**License**	**>25 fps**
xpra	Bencode-based	GPL	×
NeatX	NX	GPL	√
FreeNX	NX,RDP,RFB	GPL	√
rdesktop	RDP	GPL	×
xrdp	RDP	GPL	×
RealVNC	RFB(VNC)	closed	×
TightVNC	RFB(VNC)	GPL	×
X11VNC	RFB(VNC)	GPL	×
SSH + X-forwarding	X11	BSD	×
XDMCP	X11	MIT	×

Table 2. Remote Desktop options for 5120×1024 resolution

5. Conclusions and Future work

We have described a fully automated brightness and geometric calibration system for a multi-projector display using an uncalibrated camera. Our approach to brightness correction achieves acceptable photometric uniformity in such displays using a practical and automatic process that solves all different types of inter- and intra-projector luminance variation without the use of any special device such as spectrometers [12]. The geometric calibration can achieve, within minutes, pixel-accurate calibration using an inexpensive camera and vision-based techniques. This entire process is completely automatic, unlike other methods that require some degree of human intervention. This allows for re-calibration of the system for each use, if required, without imposing any additional burden on the user. Moreover, our method does not make any assumptions concerning the number of projectors or their configuration, the accurate

[9] Differential X Protocol Compressor project

[10] Unfortunately, this "shadow" feature is only available in the non-free NX server from NoMachine. Their most recent version (3.4.0) suffers from a bug that results in dramatic drop in frame rate when more than one client is connected to the same server session.

placement of markers, or the use of a single display surface to create a seamless display.

Our implementation is integrated in the Linux Compiz plug-in. This transforms the frame buffer in real time to obtain the desired transformation, based on the geometric calibration and the brightness map. Although the current implementation uses a rectangular region as the display area, arbitrarily shaped projected regions could be achieved by modification of the OpenGL code that draws the region. However, it is worth noting that when the warping is performed digitally, the resulting projected image quality decreases as a result of the reduced effective projected area. Image filtering and anti-aliasing techniques can reduce this effect.

References

[1] D. F. Abawi, J. Bienwald, and R. Dorner. Accuracy in optical tracking with fiducial markers: An accuracy function for artoolkit. In *Intl. Symposium on Mixed and Augmented Reality*, pages 260–261. IEEE/ACM, Nov. 2004. 4

[2] J. Canny. A computational approach to edge detection. *IEEE Trans. Pattern Anal. Mach. Intell.*, 8(6):679–698, 1986. 4

[3] H. Chen, R. Sukthankar, G. Wallace, and K. Li. Scalable alignment of large-format multi-projector displays using camera homography trees. In *VIS '02: Proceedings of the conference on Visualization*, pages 339 –346, 2002. 3

[4] 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 *Conference on Visualization*, pages 125–130, Oct. 2000. 2

[5] J. Drareni, S. Roy, and P. Sturm. Geometric video projector auto-calibration. In *Computer Vision and Pattern Recognition Workshop*, pages 39–46, June 2009. 3

[6] J. Drareni, P. Sturm, and S. Roy. Projector calibration using a markerless plane. In *Intl. Conference on Computer Vision Theory and Applications*, volume 2, pages 377–382, Feb. 2009. 3

[7] M. Fiala. Automatic projector calibration using self-identifying patterns. In *Computer Vision and Pattern Recognition Workshop*, page 113, June 2005. 2

[8] G. Humphreys and P. Hanrahan. A distributed graphics system for large tiled displays. In *VIS '99: Proceedings of the conference on Visualization*, pages 215–223, 1999. 6

[9] J. Laycocka and R. Chorleyb. Human factors considerations for the interface between an electro-optical display and the human visual system. *Displays*, 2(6):304–314, 1981. 6

[10] J. C. Lee, P. H. Dietz, D. Maynes-Aminzade, R. Raskar, and S. E. Hudson. Automatic projector calibration with embedded light sensors. In *Symposium on User Interface Software and Technology*, pages 123–126. ACM, Oct. 2004. 2, 3

[11] A. Majumder, Z. He, H. Towles, and G. Welch. Achieving color uniformity across multi-projector displays. In *VIS '00: Proceedings of the conference on Visualization*, pages 117–124, Oct. 2000. 6

[12] A. Majumder and R. Stevens. Color nonuniformity in projection-based displays: Analysis and solutions. *Transactions on Visualization and Computer Graphics*, 10(2):177–188, 2004. 5, 6, 7

[13] T. Okatani and K. Deguchi. Autocalibration of a projector-screen-camera system: Theory and algorithm for screen-to-camera homography estimation. In *Intl. Conference on Computer Vision*, volume 1, pages 774–781, 2003. 3

[14] T. Okatani and K. Deguchi. Easy calibration of a multi-projector display system. *IJCV '09: International Journal of Computer Vision*, 85:1–18, Oct. 2009. 3

[15] A. Raij and M. Pollefeys. Auto-calibration of multi-projector display walls. In *Intl. Conference on Pattern Recognition*, volume 2, pages 14–17, 2004. 2

[16] R. Raskar. Immersive planar display using roughly aligned projectors. In *Virtual Reality Conference*, pages 109–116. IEEE, March 2000. 1, 2

[17] R. Raskar and P. Beardsley. A self-correcting projector. In *Computer Vision and Pattern Recognition*, pages 504–508, Dec. 2001. 2

[18] R. Raskar, M. S. Brown, R. Yang, W.-C. Chen, G. Welch, H. Towles, B. Seales, and H. Fuchs. Multi-projector displays using camera-based registration. In *Conference on Visualization*, pages 161–168, Oct. 1999. 2

[19] J. M. Rehg, M. Flagg, T.-J. Cham, R. Sukthankar, and G. Sukthankar. Projected light displays using visual feedback. In *Control, Automation, Robotics and Vision*, pages 926–932, 2003. 3

[20] J. Salvi, J. Pags, and J. Batlle. Pattern codification strategies in structured light systems. *Pattern Recognition*, 37:827–849, 2004. 4

[21] M. C. Stone. Color and brightness appearance issues in tiled displays. *IEEE Comput. Graph. Appl.*, 21(5):58–66, 2001. 6

[22] R. Sukthankar, R. G. Stockton, and M. D. Mullin. Smarter presentations: Exploiting homography in camera-projector systems. In *Intl. Conference on Computer Vision*, pages 247–253, July 2001. 3

[23] X. Zhang, S. Fronz, and N. Navab. Visual marker detection and decoding in ar systems: A comparative study. In *Intl. Symposium on Mixed and Augmented Reality*, page 97, Sep. 2002. 4

[24] Z. Zhang, R. Deriche, O. Faugeras, and Q.-T. Luong. A robust technique for matching two uncalibrated images through the recovery of the unknown epipolar geometry. *Artificial Intelligence*, 78:87–119, October 1995. 2