Int J Interact Des Manuf (2013) 7:13–25
DOI 10.1007/s12008-012-0161-0

ORIGINAL PAPER

Optimized methods for multi-projector display correction

C. Zoido · J. Maroto · G. Romero · J. Felez

Received: 20 January 2012 / Accepted: 27 March 2012 / Published online: 7 April 2012

Abstract Multi-projector displays are commonly used for a wide range of applications such as virtual reality systems, simulators or data visualization where a high resolution image over a large projection surface is required. Such systems are cheap for the resolutions they can provide, can be configured to project images on almost any kind of screen shapes and are easily scalable, but in order to provide a seamless image with no photometric discontinuities they require a precise geometric and colour correction. In this paper, we propose a series of optimization techniques for large projection displays which make the adjustment procedure simpler and faster. Calibration method uses commercial off-the-shelf equipment such as a webcam and an intermediate performance graphics card. Multiple views are used if one camera image can't cover the entire display with enough resolution. The effect of these optimizations in the calibration of a simulated display formed by 48 projectors are shown.

Keywords Multi-projector display · Simulator · Calibration · Virtual reality

1 Introduction

Multi-projector systems are a common choice when a high resolution image has to be represented on a projection surface which has not a fixed aspect ratio or simple geometry. Nevertheless, configuring a multi-projector display has traditionally been an expensive task involving investment in specialized hardware and high maintenance costs. Advances in graphics hardware technology due to the expansion of the computer games industry and the increasing availability of low cost digital capture devices provide new tools to design and calibrate a multi-projector display at a fraction of the price.

Design of these kinds of displays has to confront various challenges in order to obtain a seamless output image. Projectors can be casually positioned, so in order to project an aligned, perfectly delimited image, the partial output from each projector has to be distorted. Also, colour discontinuities arise due to overlapping of projectors, lamp degradation or the use of different kinds of projectors so colour correction has to be applied to each projector image.

Years ago Hardware-based solutions were used to solve these problems. Special projector mounts helped to get a precise projector alignment. Photometric correction was achieved putting metal plates physically interfering with the beam of light in order to tone down the brilliant areas in the overlap zones. This approach doesn't usually provide optimal results and is extremely expensive and slow. Nowadays, using camera-based calibration techniques, displays can be set up in a cheaper and more accurate manner. Cameras give the chance to capture projecting structured light of a known geometrical pattern, like a checkerboard or temporal coded light patterns. Through this pattern, camera–projector correspondences are established in order to relate all the projectors to a common reference frame. This reference frame is usually set along the display surface using a "fiducial border" or taking an arbitrary reference frame. These cameras are also used to capture colour information in order to estimate the projector colour response curve. Once all the data captured by the camera has been collected and processed, geometric and photometric correction data is generated and image correction is achieved in real-time using GPU's through a custom built 3D rendering graphics engine.

C. Zoido (✉) · J. Maroto · G. Romero · J. Felez
CITEF, Universidad Politécnica de Madrid, Madrid, Spain
e-mail: czoido@etsii.upm.es
URL: http://www.citef.etsii.upm.es/

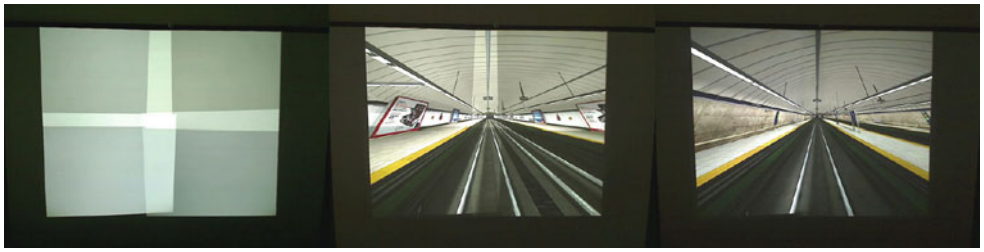

Fig. 1 Initial projector positions, geometrically corrected display and display corrected in geometry and colour

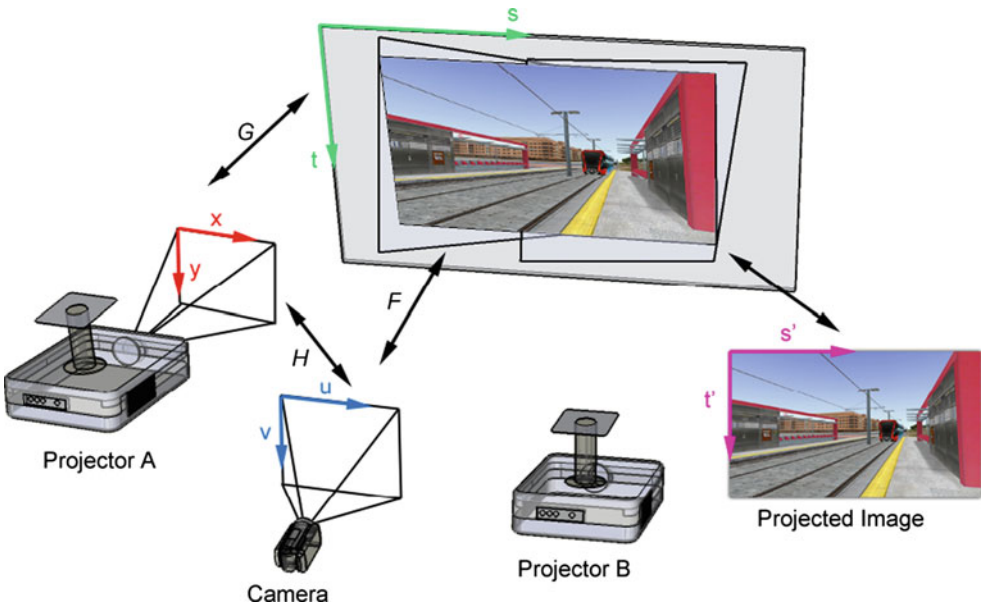

Fig. 2 Planar projection surface with all the coordinate systems involved

Several methods have been developed to correct tiled projection systems but most of them do not put enough emphasis on calibration speed. Some of them use slow equipment like spectroradiometers or digital cameras to capture huge amounts of data in a calibration step previous to geometric correction. Optimizing calibration speed is crucial for large-scale projection displays such as a 48 projector system or when the calibration is carried out by an embedded platform that cant perform intensive calculations.

Almost every previous correction method centralises the correction process in a single PC. Here, a decentralized correction method with a set of optimizations applied is presented. The proposed implementation estimates also each projector's transfer function as a part of the calibration process. The proposed implementation can achieve automatic geometric and photometric seamlessness in few seconds for small-scale displays (Fig. 1) and in minutes for a large-scale display formed by 48 projectors. All of this can be achieved making use of one single conventional webcam taking advantage of graphics hardware.

The most popular state-of-the-art multi-projector displays correction methods are listed and analyzed next. Then, the proposed implementation and its optimizations are explained.

2 Geometric correction of tiled displays

Several methods have been developed for automatic geometric correction of multi-projector displays. All of them make use of a camera to find out the relationship between the camera itself, the projection surface and each projector's coordinate system. The coordinate systems are shown in Fig. 2. The notation adopted by [1] is followed.

The transformation that has to be found is given by

$$G_{(x_i,y_i)\to(s,t)} = F_{(u,v)\to(s,t)} \cdot H_{(x_i,y_i)\to(u,v)} \tag{1}$$

The different elements that constitute the multi-projector system are listed below:

- Surface projection S, with horizontal and vertical coordinates (s, t). The image I, projected on the surface is parameterized by (s', t'). The relation between (s', t') and

(s, t) in planar projection surfaces is a scale plus a translation so they can be considered equivalents.
- Each projector P_i that projects images with coordinates (x_i, y_i).
- A camera that captures images with coordinate system $C(u, v)$.

The transformation that relates each projector to the projection surface $G_{(x_i,y_i)\to(s,t)}$ (Eq. 1) has to be found. This transformation can be calculated as the composition of the function that relates the camera coordinates $C(u, v)$ to the display coordinates $F_{(u,v)\to(s,t)}$ and the one that relates the camera coordinates to the projector coordinates $H_{(x_i,y_i)\to(u,v)}$.

The most used methods to establish these mappings between coordinate systems are the following:

- Linear methods. Geometric linearity is assumed in both the camera and the projector. If this assumption is true, or if the camera and the projector have non-linearities, but they can be corrected (to make them behave as linear devices), then a linear transformation called homography can be used to make the mapping between all the coordinate systems [2].
- Piecewise linear methods. Instead of having a model for the transformation functions, a dense correspondence between camera and projector features is established [3]. Linear methods are used for interpolation between these features. This interpolation can be done with three points using barycentric coordinates or with four points defining a homography matrix.
- Non-linear models. When non-linearities are present, cubic polynomials can be used to relate the camera and projector coordinate systems [4]. This model can deal with projector's lens distortion but it requires iterative algorithms to estimate all the parameters and a dense correspondence of features to establish the model.

3 Photometric correction of tiled displays

One of the challenges when designing multi-projector displays is to solve the colour variations that may appear on the projection surface. These variations break the projected image continuity. In this section, the causes of colour variations in tiled display systems and the different methods for avoiding them are discussed.

3.1 Colour discontinuities in tiled projection displays

Colour variations can be produced in greater or lesser degree depending on factors such as the optics of the projectors, their spatial distribution, their technology or the type of projection surface which displays the image. The causes of colour variations over the projection surface are listed below:

- Intra-Projector variations. Such variations are those that can be seen within the area of influence of a single projector. Several studies have determined that projectors show colour variations mostly due to changes in luminance. Chrominance variation inside the area of the projector is minimal [5]. Also, the normalized transfer function of the projector remains constant throughout the projection surface [3].
- Inter-Projector variations. If the projection system is composed from projectors of different makes and models colour discontinuities will occur due to the different colour outputs. Sometimes, even if the projection system is composed of projectors from the same make and model these variations can happen if they have different lamp age or are configured with different settings.
- Overlapping areas. The areas where two or more projectors overlap show a significant increase in brightness. These areas have to be corrected through an attenuation factor that mitigates this effect.

3.2 Colour correction methods

In recent years various colour correction methods have been developed in order to achieve photometric uniformity across the display, or at least smooth colour variations. Some of these methods are analyzed next:

- Luminance attenuation maps. This method corrects the inter-projector, intra-projector and overlapping regions luminance variations [6]. It uses per channel maps that scale the luminance response in order to produce smooth transitions between projectors. Since it cannot address chrominance variations it doesn't perform very well for flat colours when projectors have large colour shifts.
- Gamut matching. Gamut matching based methods [7] only address the inter-projector colour variations. They usually use a sensing device such as a spectroradiometer to measure the 3D gamut of each projector. This gamut is a parallelepiped for three primary projectors. Then a common gamut is defined resulting from the intersection of all the parallelepipeds. Finally, the colour output of all the projectors is mapped to this common gamut. This method is only valid for three primary projectors where linear transformations can be applied. This method was expanded to handle four primary devices such as DLP projectors [8]. Making a dense sampling of the colour output to measure the gamut with nonlinear primaries dependencies. This procedure is expensive and slow as it requires an spectroradiometer and dense colour sampling.

- Gamut morphing. This method addresses spatial colour variations in both chrominance and luminance [9]. It morphs the gamut across the display in order to make smooth chrominance and luminance variations. This morphing is constrained by perceptual parameters producing a seamless display.

4 Proposed implementation

The aim of these optimizations is to drastically reduce the calibration time for large-scale projection displays. This is usually not a problem for commercial grade applications on small-scale displays but can be extremely time consuming as the size of the projection display increases. Recalibration is more likely to be needed for large displays as the chance of fail or misalignment of the system increases with the number of projectors. Previous correction methods [10] for large-scale displays take 4 h to correct a 48 projector display that has to be recalibrated about once a month. Our method could adjust the display in minutes. It should be noted also, that if anything goes wrong with the calibration procedure, repeating a task that takes hours is not the same as repeating one that takes minutes.

Also, an interest for display systems formed by pico-projectors connected to single-board computers such as the BeagleBoard [11] or Raspberry Pi has grown in recent years. This kind of platforms can't usually perform intensive tasks on its CPU but they come with a GPU, making it possible to implement the proposed techniques on them.

The described implementation requires no human intervention and can achieve both geometric and photometric correction for each projector in seconds. This calibration method is implemented in C++ and offloads some of the most intensive calculations to the graphics card GPU. Also, the online image correction part is carried out by the GPU making use of shaders. This allows us to make fast corrections that don't alter normal operation when the system needs to be recalibrated.

The display correction process is divided into three different steps, all of them have optimizations applied that will be described further:

- Camera data collection. A single webcam is used for capturing the geometry of the display. As a webcam resolution is not enough for large displays, data collection is divided into multiple views.
- Correction mask calculation. Camera data is used to model display geometry and masks are generated to correct the overlapping regions and luminance variations between the projectors. All the calculations are parallelized distributing them over the network taking advantage of the nature of these kinds of systems.
- Online output image correction. All the data generated in the previous step is saved in textures that shader programs can load directly. Some optimizations in the rendering pipeline are also applied.

Specific details of each steps and its optimizations will be analyzed next.

4.1 Camera data collection

The application of this method is shown for a simulated display formed by 48 projectors. This simulation is created to prove the scalability of the set of optimization techniques proposed and show its impact in a large-scale projection system. All the collection of camera images are generated using the real camera projection model and previously captured photometric data from projectors. The generated set of images are processed by the distributed calibration system in order to generate correction data. This way, a set of input images similar to real ones can be used to test the algorithm. Images from a real display and a simulated one are shown in Fig. 3.

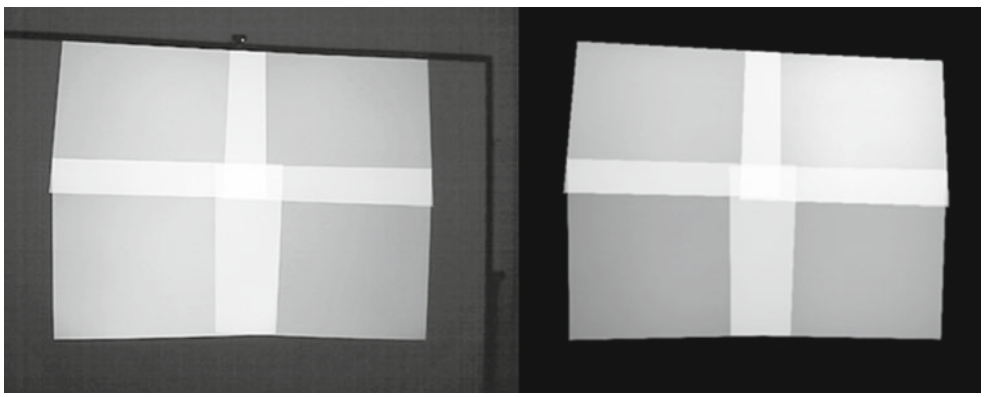

Fig. 3 A real 4 projector system (*left*) compared to the simulated one (*right*)

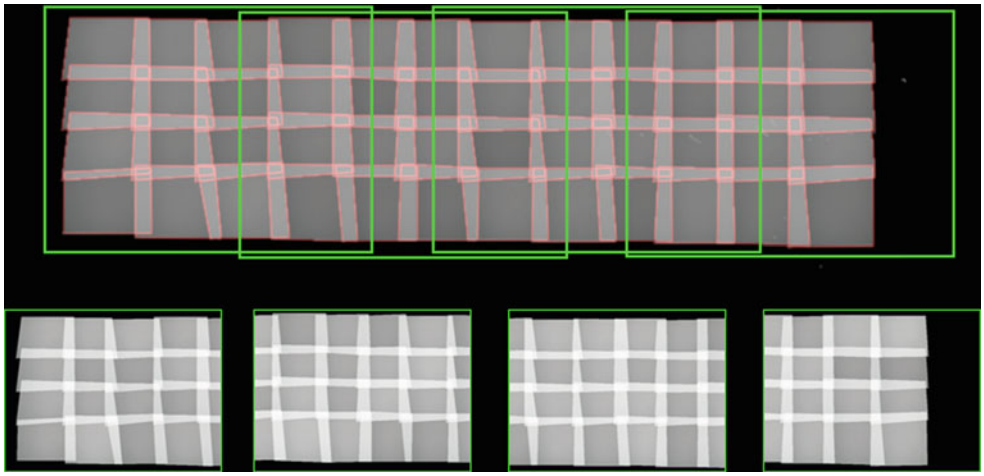

Fig. 4 Simulated display formed by 48 projectors and the partial views used to calibrate it

As previously said, our implementation makes use of a single conventional webcam for calibration. Obviously, one single view of the webcam doesn't provide enough accuracy for the calibration of a large projection display. Our experience shows that a maximum of 4×4 projectors array can be corrected with enough accuracy with one single view at a 640×480 resolution using the structured light technique explained further. For larger displays the data from multiple views of the same camera is used in a similar way to [2], using multiple views is equivalent to using multiple cameras. The camera is positioned capturing points shared between views so adjacent views can be related through homography transformations. Then all of the partial views can be combined in an arbitrary reference frame. Figure 4 shows a simulated display formed by 48 projectors that has been reconstructed from four partial views represented in it.

The proposed implementation uses structured light techniques to establish the correspondences between each camera view, projector and projection surface. Temporal coded patterns with gray codes are projected. Gray codes reduce the probability of detection errors between consecutive patterns [12]. These patterns are captured by a conventional webcam that captures at a rate of 10 patterns per second. Usually no more than 5 patterns are needed for precise geometric alignment. This means that it takes about 2 s to retrieve the pattern information for each projector.

Once all the images have been captured they are processed in order to get correspondent points between reference frames. The intensity levels of the horizontal $I_h(u, v)$ and vertical $I_v(u, v)$ images are processed in Eq. 2 to produce a checkerboard pattern image (Fig. 5).

$$I_{\text{checker board}}(u, v) = |I_h(u, v) - I_v(u, v)| \tag{2}$$

The checkerboard corners are extracted using computer vision techniques (Fig. 6). Only the inner points of the checkerboard are kept. All the points lying on the outside contour

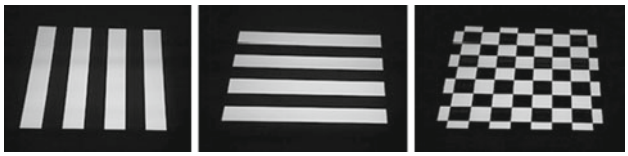

Fig. 5 Vertical and horizontal *gray codes* combined to produce a checkerboard pattern

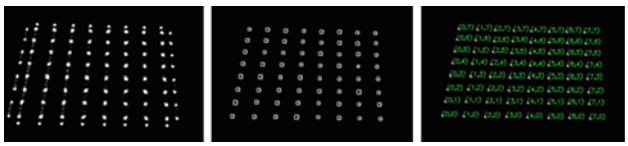

Fig. 6 Inner points are detected and labelled

are discarded. Gray codes are used to label all the points. The point labelling procedure is shown in Fig. 7.

The found location of the checkerboard corners are refined using sub-pixel accuracy algorithms like the ones explained in [13], resulting in the points that will establish the correspondences between reference frames.

Once all the points are detected and labelled in camera space (u, v) for each view, they are related to the projector (x, y) and the surface (s, t) to find the transformations between systems (Fig. 8). As the projection display is planar and the webcam's lens distortion has been corrected using the method in [14], this relation is a homography transformation.

If images from multiple views have been captured, calculated transformations for each camera view are related to a common reference frame. This reference frame is usually coincident with one camera's view reference frame.

The final projection surface reference is chosen via an automatic algorithm to use all the available space for projection. The only measure that has to be provided is the aspect ratio of the final projected image. For the sake of simplicity, the method is explained for a two projector display, but can be scaled for any number of projectors.

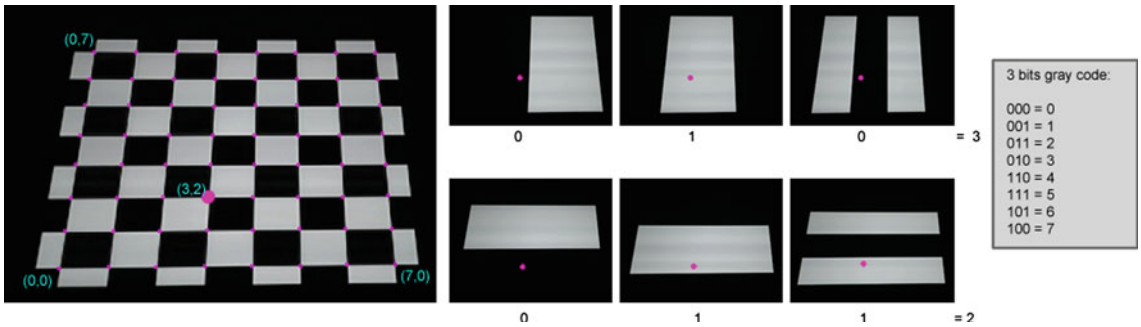

Fig. 7 Labelling the point at column 3, row 2 using gray codes

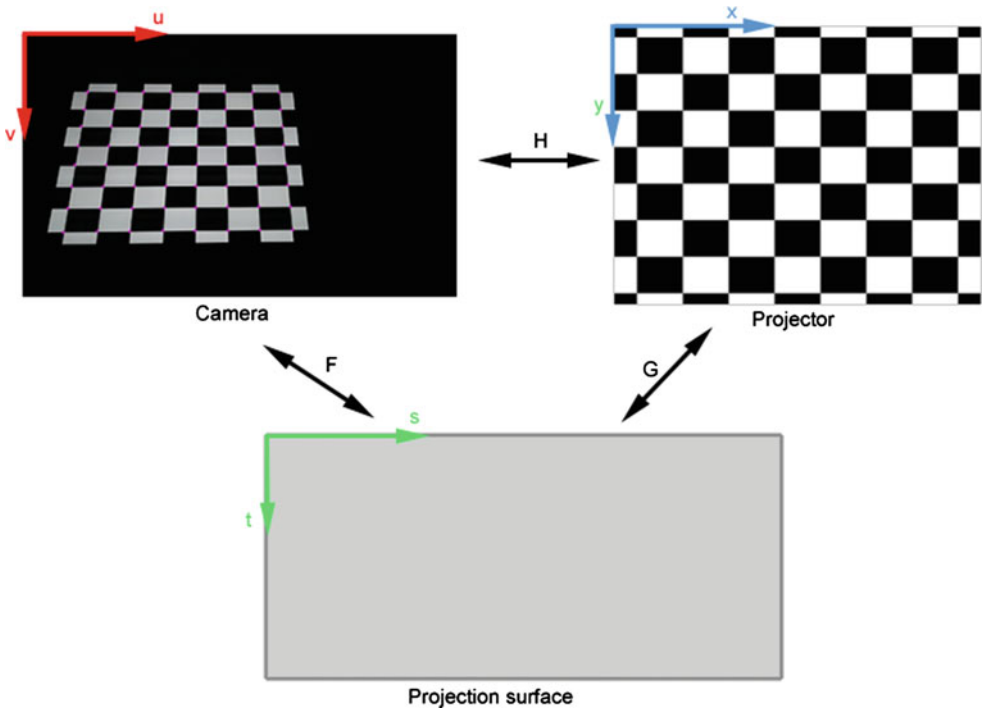

Fig. 8 Relations between reference frames of one projector and the projection surface

The area where the final image is going to be projected is a trapezoid. To calculate this trapezoid the outer contour of each projector in camera space Q_i is determined. These contours are then joined (Eq. 3), obtaining the total area of influence of all the projectors E.

$$E = \bigcup_{i=0}^{n} Q_i \tag{3}$$

The polygon E is taken to an arbitrary coordinate frame where the projected image of one reference projector is perfectly aligned. In this reference frame the algorithm finds the maximum inscribed polygon that will form the final projection trapezoid E'. For each projector the intersection of Q_i and E' is calculated (Eq. 4). This intersection Q'_i gives the transformation that must be applied for each projector (Fig. 9).

$$Q'_i = E' \cap Q_i \tag{4}$$

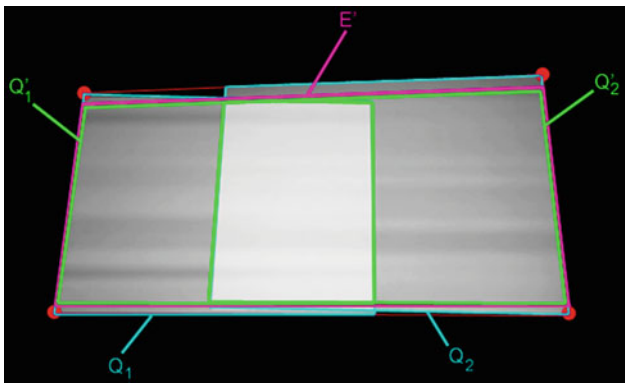

Fig. 9 Determining the final projection area

Once the geometry of the projection display is known the transfer function for each projector is calculated. This function must be estimated in order to model the non-linear behaviour of the projectors.

Fig. 10 Four projector display with overcompensation (*left*) and with the inverse transfer function correction applied (*right*)

All photometric correction masks described further are calculated with the assumption that corrections will take place in a linear space so if they were applied directly an undesirable overcompensation effect would arise (Fig. 10). In order to avoid the overcompensation effect the projector's intensity transfer function $f_i\,(l)$, where l is the projector input level, should be estimated.

Projector transfer function estimation usually involves the use of a spectroradiometer or a digital camera that has been calibrated in colour. Spectroradiometers are precise colour measuring devices but they are extremely expensive and slow, what makes them unusable for our purposes. Calibrating the camera, in order to model its non lineal response, with an algorithm such as that proposed in [15] could be an option, but in our implementation this step will be bypassed using a webcam with "raw" capture mode. This is a mode that some recent consumer level webcams include (DSRL's have included this mode for several years) so that the camera image pixel values are a linear transformation of the sensor chip values. Using a webcam instead of a Digital Camera is chosen because of its lower price and faster image capture rate. For transfer function calculation image resolution is not crucial so a view that covers the entire projection display can be used both for small and large projection displays.

The transfer function is estimated measuring the projector's output for each gray input level l. The response function of all the pixels of the display is identical independently of the spatial location [16] so a region near the centre of each projector is chosen for measuring.

However, as the camera can't cover the entire intensity output from the projectors, it must capture projector data at different exposure levels. The capture algorithm is configured in order to capture as few images as possible reducing the capture time. The total data capture time t_i for estimating each projector's transfer function is given by Eq. 5:

$$t_i = \left(\frac{1}{fps}\right) \cdot \sum_{n=0}^{n_{\text{exp}}} l_{\text{exp}} \tag{5}$$

Where fps is the frame rate of the camera, n_{exp} is the number of different exposure levels needed to cover the entire projector dynamic range and l_{exp} is the number of input levels measured for each exposure level.

The frame rate of the camera used for this implementation is around 10 frames per second. For each exposure, a number l_{exp} from a uniform sampling of inputs is projected. The number of inputs l_{exp} and different exposures n_{exp} depends on the configuration of the camera itself, the dynamic range of the projector and the shape of its transfer function. In this case, exposure levels are changed after captured intensity exceeds a certain level l_{max} avoiding saturated images. Each band of data, captured at different exposure levels is shown in Fig. 11. Overlapping regions of data for different exposures are needed in order to obtain the merged result, that's why the first input chosen for the next exposure level is bellow the last input level.

After all the images have been captured, the data of the different exposure levels is merged knowing that the same inputs for different exposure levels represent the same real projector's luminosity level. As previously said, in this case the camera's response is linear so the data at different exposure levels for certain input levels can be related through a proportional factor. Merged data is normalized to the [0, 1] domain.

Functions can be estimated in a few seconds. In the case of Fig. 11, it took 4 different exposures to capture the projector's transfer function for both projectors. First projector had $l_0 = 11$, $l_{11} = 9$, $l_2 = 17$ and $l_3 = 29$ so using Eq. 5, the total time to estimate the projector's transfer function is approximately 6.6 s. Estimating the function for the second, with values $l_0 = 19$, $l_{11} = 12$, $l_2 = 11$ and $l_3 = 21$ took about 6.3 s. This means that collecting images and estimating the projectors transfer functions for a display made up of 48 projectors would take around 7 min. It must be noted that this step is usually performed previous to display geometry correction and is not normally showed for commercial systems.

Once the data from different exposures is scaled, merged and normalized the projector's response function $f_i\,(l)$ is inverted and the fitted to a Bezier curve which will be denoted as $f_i^{-1}\,(l)$.

The estimated transfer function model is not the only information that's used from the captured images; also, the different maximum luminance levels L_i of each projector at its centre region are recovered. Those maximum levels

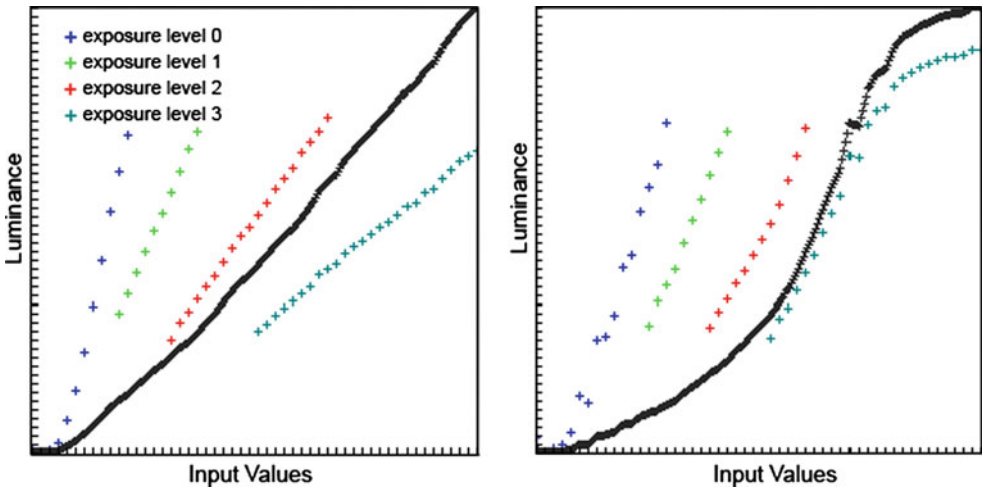

Fig. 11 Bands of luminance levels for different exposures, captured for two different projectors

are used to balance different brightness levels between projectors, multiplying by a factor $\frac{L_i}{L_{\max}}$ reducing the brightest projectors to match the dimmest one. This data will be used in Eq. 7.

4.2 Photometric correction mask calculation

The photometric correction method makes some assumptions like the black level of the projectors is negligible and that there are not intra-projector colour variations. The inter-projector variations are addressed balancing luminance levels between projectors but chrominance variation for the projectors is considered negligible. This assumption holds almost true for displays with projectors from the same make and model with the same settings where only variations in the white point and maximum luminance could appear due to different lamp ages. All the optimizations can still be used in the case that the assumptions are not true, where projectors show colour shifts, but the final result could show some undesirable perceptible colour differences.

The most noticeable photometric discontinuities appear in the overlap regions. This method generates correction masks for these areas in a few seconds. Masks are generated with data derived entirely from geometric calibration and then modified using an estimated projector transfer function and inter-projector luminance balancing factors. The masks multiply the output image in order to make brightness variations disappear.

The correction function $G(\vec{p}_i)$ presented in [17] for mask calculation is used in this implementation. The function is shown in Eq. 6

$$G(\vec{p}_i) = \frac{\xi_i}{\sum_{j=0}^{N} \xi_j}; \quad \xi_i = \prod_{k=1}^{4} d_{i,k} \tag{6}$$

where $d_{i,k}$ is the distance from one point of projector i to the edge of each projector k with which it overlaps.

The final transformed compensation mask $A(\vec{p}_i)$ with the inverse transfer function and luminance level factors applied is shown in Eq. 7.

$$A(\vec{p}_i) = f_i^{-1}(E(\vec{p}_i)); E(\vec{p}_i) = \frac{L_i}{L_{\max}} \cdot G(\vec{p}_i) \tag{7}$$

Mask calculation time depends on the complexity of chosen blending function, although the optimization described bellow is applicable to any kind of blending function it has to be noted that all the results shown are for the blending function from Eq. 6.

Typical calculation procedures for these masks involve a number of operations that increase with the projector resolution. In a SXGA projector (1,280 × 1,024 pixels) 1,310,720 operations are needed to calculate the mask for each projector with standard procedures.

This implementation takes advantage of graphics hardware in order to speed up the calculation of these masks and reduce the number of operations.

First, a triangle mesh is generated in the projector reference frame.

Each triangle vertex of this mesh has an associated colour that contains the information of the mask correction factor, calculated with the function $G(\vec{p}_i)$. As can be seen in Fig. 12, this mesh is not regular. Some of the vertex will contain the same information than others and fine triangle subdivision is not generated in those areas.

The parameter η is defined. This parameter represents what we call the "maximum partition level". This level defines the number of divisions of the mesh. A bigger η will produce more precise masks, but will require more calculation time.

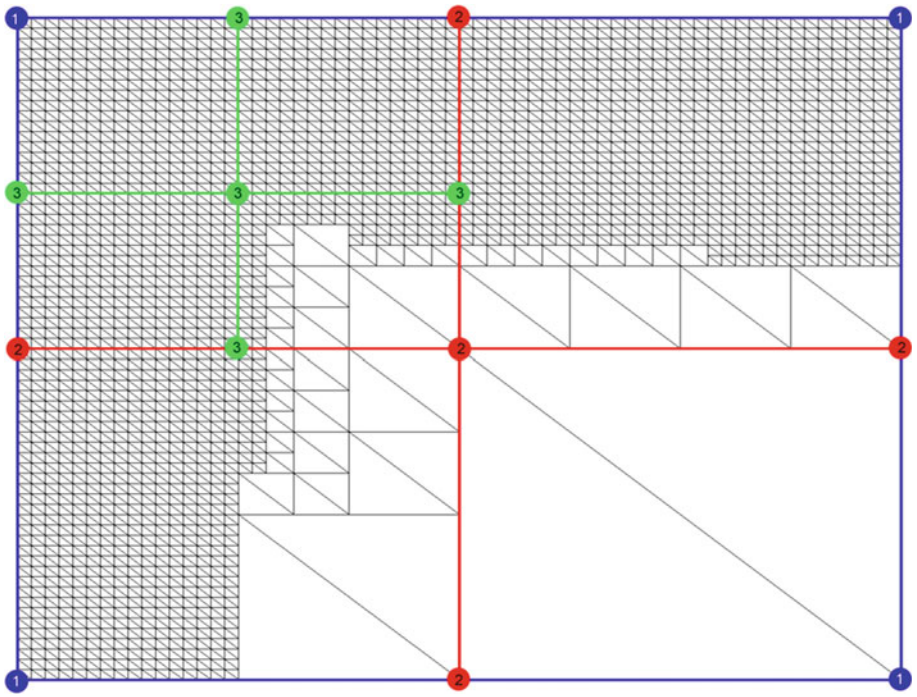

Fig. 12 Triangle mesh for mask calculation

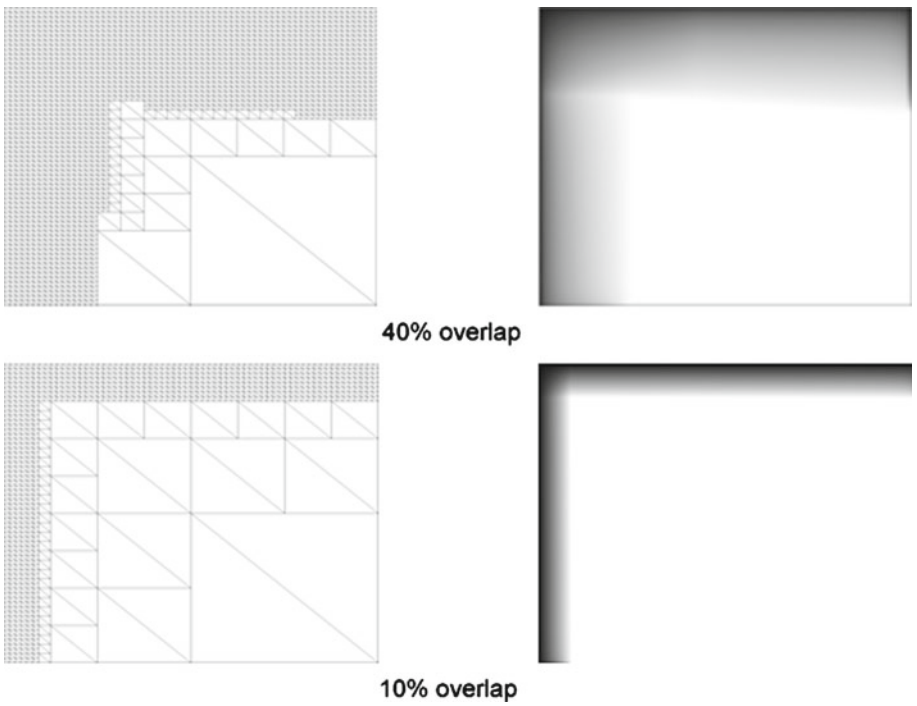

Fig. 13 Rendered triangle meshes for 40 and 10 % overlap

Once all the vertex colours are assigned, the mesh is rendered to a texture. The GPU makes all the interpolation between triangle vertex producing the correction factors for every pixel in the projector space. In Fig. 13 the rendered texture is shown for 10 and a 40 % overlap regions.

The influence of η in the rendered texture quality is shown in Fig. 14. Differences beyond $\eta = 32$ are imperceptible so 32 is chosen as the default value.

The results of the optimization for one projector are presented in Table 1 (time in seconds). The following notation is used:

- η : Maximum partition level. Number of subdivisions of the mesh.
- t_{TR} : Total time to generate a correction masks through a regular triangle mesh for each η.

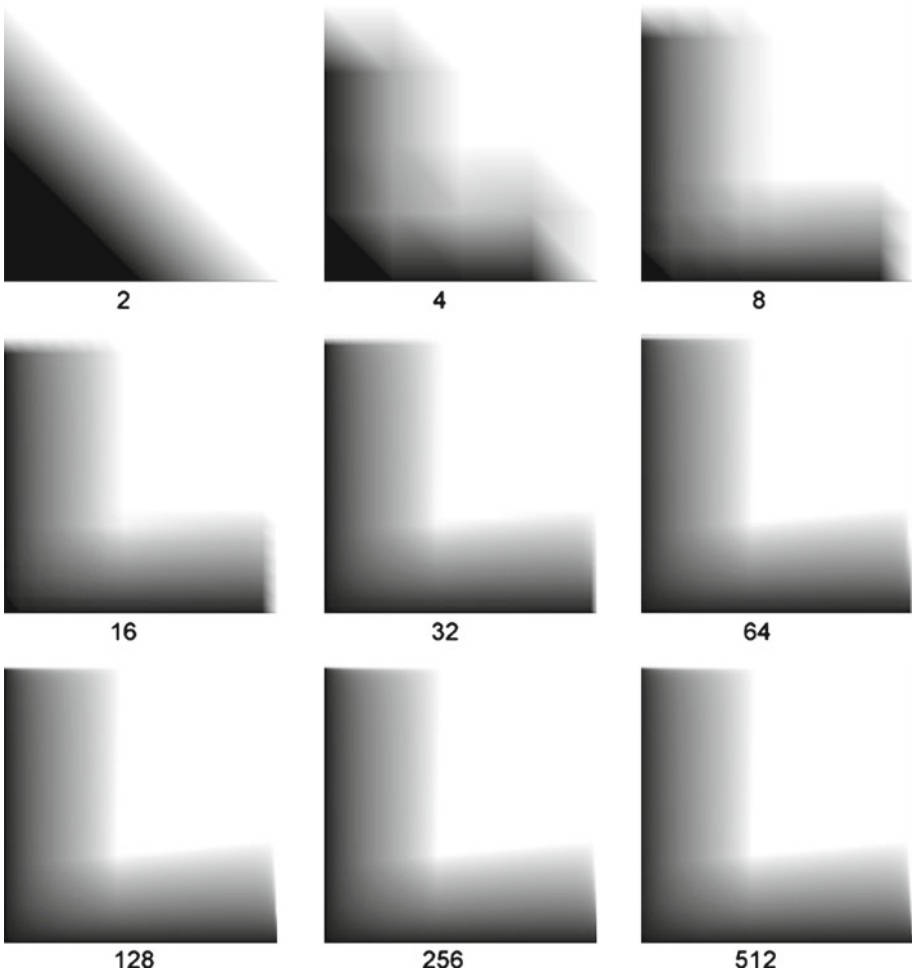

Fig. 14 The same mask generated with maximum partition factor η from $\eta = 2$ to $\eta = 512$

Table 1 Time results for different η values

η	t_{TR}	t_{TI50}	t_{TI10}	t_{CPU}	R
2	0.977	0.977	0.978	204.3	208.93
4	0.992	0.993	0.993	204.3	205.64
8	1.086	1.058	1.029	204.3	198.62
16	1.307	1.279	1.115	204.3	183.16
32	2.896	2.581	1.705	204.3	119.79
64	10.876	7.704	3.939	204.3	51.86
128	36.562	27.796	13.734	204.3	14.87
256	126.919	97.656	43.315	204.3	4.71
512	508.548	483.713	187.399	204.3	1.09

- t_{TI50} and t_{TI10}: Total time to generate the correction masks with the proposed irregular mesh for 50 and 10 % overlaps.
- t_{CPU} : Total time in generating the correction mask calculated by the CPU without any optimization.
- R : Measures the improvement in time using the proposed method for a 10 % overlap. It is calculated as $\frac{t_{\text{CPU}}}{t_{TI10}}$.

These results are for one projector. Table 1 shows that the mask for one projector is generated in about 3 s for the default partition while it would take more than 3 min generating the mask described in Eq. 6 using a standard procedure without the optimizations. Our method performs 120 times faster ($\eta = 32$) than standard CPU calculation for the mask shown in Eq. 6. This results are for the algorithm running on an Intel Xeon 3.06 GHz and 2 GB of RAM with a NVIDIA GForce 6800 GT which can be considered rather outdated.

This implementation is used for multi-projector displays used to show 3D real-time content. The end system comprises a cluster of computers (Fig. 15) where each computer renders the part corresponding to a projector.

We take advantage of this distributed topology in order to parallelize all the correction mask calculations. The computer that carries out most of the calibration process acts as a coordinator and sends to each computer the geometric information of the display needed to calculate the mask that corrects its projector. It also represents a saving in transfer time as a couple of strings are sent over the network instead of sending an image. This way, the mask calculation step is almost independent from the number of projectors, so gen-

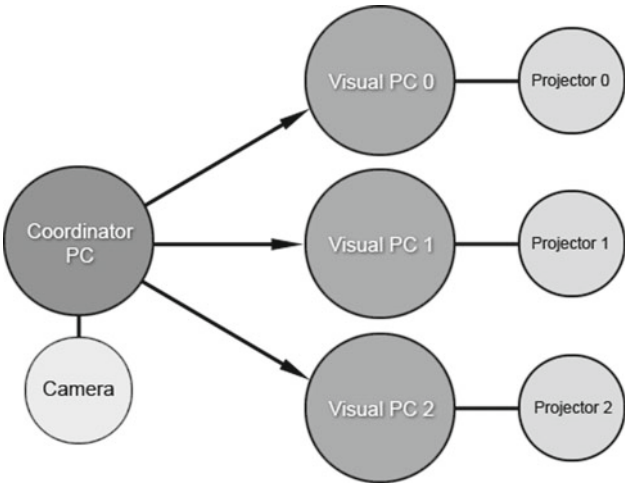

Fig. 15 Coordinator PC sends correction data to each rendering station so that they calculate their own correction mask

erating all the masks for a 48 projector display (Fig. 16), if every rendering node had a hardware similar to the one already mentioned, would take about 3 s for mask generation instead of about 3 min.

This optimization could be especially useful if the projectors are connected to single-board computers with reduced CPU calculation power but with a GPU included.

4.3 Online output image correction

Our implementation works with a tiled projector display aimed at rendering 3D content. Instead of using a tool like Chromium [18] for rendering OpenGL content through a cluster of computers we have developed our own 3D rendering system. The visual engine of this simulator is based on OpenScenegraph, an open source 3D graphics application programming interface.

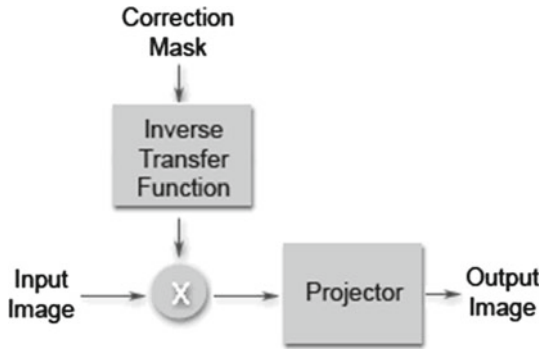

Fig. 18 Proposed photometric correction operation sequence

Each rendering node in the display generates its correction mask in the form of an image. This image is loaded by a correction shader in the 3D rendering engine. The series of operations followed by the shader in this implementation are different from other previous approaches like [9] and [17]. In those, the normal sequence of operations involves linearizing the projector's input image transforming it with a gamma function, then applying the linear calculated correction masks for each projector and transforming again with the inverse transfer function as shown in Fig. 17.

The proposed approach Fig. 18 differs from the previous one in that the input image isn't linearized, only the correction mask is transformed.

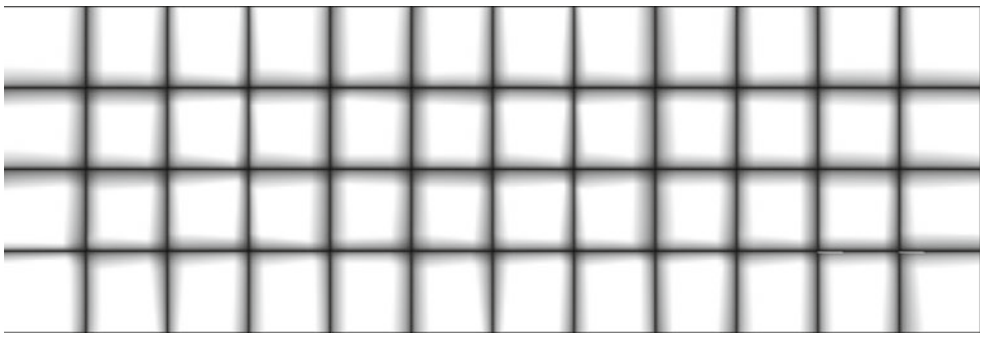

Fig. 16 Calculated mask for each rendering station in a 48 projectors display

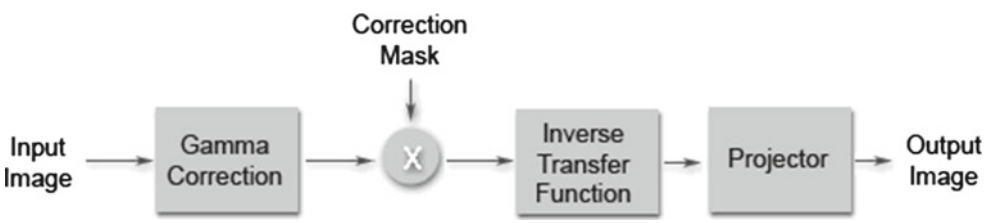

Fig. 17 Typical photometric correction operation sequence

Fig. 19 Results for a railway simulator and a 4 projector display system

There are two advantages in correcting this way. The first one is that it simplifies the operation sequence, decreasing the risk of degrading the input image in the colour transforming operations due to rounding errors. The second one is that all the photometric correction information that has to be passed to the 3D rendering engine is contained in the correction mask. That means that there's no need to apply the inverse function via 1D look-up-tables to the output image inside the 3D graphics pipeline.

4.4 Results

The final results of our implementation are showed in Fig. 19 for a real 4 projection system dedicated to rendering 3D content for a training simulator. As can be seen on the images, there are no noticeable photometric or geometric discontinuities.

Comparing the presented implementation to commercial ones is not an easy task. This implementation is aimed at low-

budget tiled projection systems and proposes a set of techniques that become more useful when the rendering nodes are low-performance but equipped with a GPU or as the number of projectors increases. One single projector enabled with blending technology from a known commercial brand could cost more than the entire hardware for setting up a 24 projector display with our techniques. Although there are commercial systems that produce great results they usually use high-end projectors with a stable known colour output gamut.

5 Conclusions and future work

In this paper, a series of optimization methods for multi-projector displays have been presented. These methods have proved to be not only useful for shortening calibration time needed but also to provide an alternative, simplified and more efficient way of applying the colour correction in the rendering graphics pipeline.

At last, the calibration process takes a few seconds per projector and can be achieved with inexpensive hardware such as a webcam and an intermediate performance graphics card.

Future work include implementing the set of techniques to arbitrary shaped surfaces as these optimizations have been only applied to planar surfaces. Also some the application of further colour correction techniques without penalising time performance should be investigated.

References

1. Majumder, A., Brown, M.S.: Practical Multi-Projector Display Design. A.K. Peters, Ltd, Natick (2007)
2. Chen, H., Sukthankar, R., Wallace, G., Li, K.: Scalable alignment of large-format multi-projector displays using camera homography trees. In: Proceedings of the Conference on Visualization '02 (VIS '02), pp. 339–346. IEEE Computer Society, Washington (2002)
3. Brown, M., Majumder, A., Yang, R.: Camera-based calibration techniques for seamless multiprojector displays. IEEE Trans. Vis. Comput. Graphics **11**(2), 193–206 (2005). doi:10.1109/TVCG.2005.27
4. Hereld, M., Judson, I.R., Stevens, R.: Dottytoto: a measurement engine for aligning multi-projector display systems. Argonne National Laboratory. Preprint ANL/MCS-P958-0502 (2002)
5. Majumder, A.: Properties of color variation across multi-projector displays. In: Proceedings of SID Eurodisplay, pp. 807–810 (2002)
6. Majumder, A., Stevens, R.: LAM: luminance attenuation map for photometric uniformity in projection based displays. In: Proceedings of the ACM Symposium on Virtual Reality Software and Technology (VRST '02), pp. 147–154. ACM, New York (2002). doi:10.1145/585740.585765
7. Wallace, G., Chen, H., Li, K.: Color gamut matching for tiled display walls. In: Proceedings of the Workshop on Virtual Environments 2003 (EGVE '03). ACM, New York, 293–302 (2003). doi:10.1145/769953.769988
8. Stone, M.C.: Color balancing experimental projection displays. In: 9th IS&T/SID Color Imaging Conference, pp. 342–347 (2001)
9. Sajadi, B., Lazarov, M., Gopi, M., Majumder, A.: Color seamlessness in multi-projector displays using constrained gamut morphing. IEEE Trans. Vis. Comput. Graphics **15**(6), 1317–1326 (2009). doi:10.1109/TVCG.2009.124
10. Pagani, A., Stricker, D.: Spatially uniform colors for projectors and tiled displays. J. Soc. Inf. Display **15**(9), 679 (2007). ISSN:10710922
11. Amiri, K., Yang, S.-H., Kurdahi, F., El Zarki, M., Majumder, A.: Collaborative video playback on a federation of tiled mobile projectors enabled by visual feedback. In: Proceedings of the 3rd Multimedia Systems Conference (MMSys '12), pp. 113–118. ACM, New York (2012). doi:10.1145/2155555.2155575
12. Horn, E., Kiryati, N.: Toward optimal structured light patterns. In: Proceedings of the International Conference on Recent Advances in 3D Digital Imaging and Modeling (NRC '97). IEEE Computer Society, Washington (1997)
13. Bradski, G., Kaehler, A.: Learning OpenCV. Computer Vision with the OpenCV Library. O'Reilly Media (2008)
14. Zhang, Z.: A flexible new technique for camera calibration. IEEE Trans. Pattern Anal. Mach. Intell. **22**(11), 1330–1334 (2000). doi:10.1109/34.888718
15. Debevec, P.E., Malik, J.: Recovering high dynamic range radiance maps from photographs. In: Proceedings of the 24th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '97), pp. 369–378. ACM Press/Addison-Wesley, New York (1997)
16. Majumder, A.: A practical framework to achieve perceptually seamless multi-projector displays. Ph.D. Dissertation, The University of North Carolina at Chapel Hill (2003)
17. Harville, M., Culbertson, B., Sobel, I., Gelb, D., Fitzhugh, A., Tanguay, D.: Practical methods for geometric and photometric correction of tiled projector. In: Proceedings of the 2006 Conference on Computer Vision and Pattern Recognition Workshop (CVPRW '06). IEEE Computer Society, Washington (2006). doi:10.1109/CVPRW.2006.161
18. Humphreys, G., Houston, M., Ng, R., Frank, R., Ahern, S., Kirchner, P.D., Klosowski, J.T.: Chromium: a stream-processing framework for interactive rendering on clusters. ACM Trans. Graph. **21**(3), 693–702 (2002). doi:10.1145/566654.566639