

# Automatic Color Calibration for Commodity Multi-projection Display Walls

Luciano Pereira Soares  
Instituto Superior Técnico  
Lisboa - Portugal  
Isoares@immi.inesc-id.pt

Ricardo Jota Costa  
Instituto Superior Técnico  
Lisboa - Portugal  
jota@immi.inesc-id.pt

Joaquim Armando Jorge  
Instituto Superior Técnico  
Lisboa - Portugal  
jaj@immi.inesc-id.pt

## ABSTRACT

Multi-projection display walls are often used in advanced virtual reality applications. However, most dedicated hardware available for these systems is very expensive. Our approach focuses on developing alternative solutions using inexpensive commodity projectors and screens driven by a commodity PC cluster. Unfortunately, using inexpensive projectors raises interesting problems both in terms of color and intensity matching, which need to be tackled to ensure reasonable image quality and precision. Indeed, commodity projectors do not have good color or brightness inter-projection stability or control. This means that two "identical" projectors from the same manufacturer, model, lamp life, operating at the same temperature, can present widely different color and brightness output given the same input signal. To alleviate this, our technique uses graphics card resources to control the output video signal before it reaches the projectors. We use an inexpensive web camera to capture the display wall image. In this way we can identify color variations in the projected image and then adapt the graphics card's gamma curve to achieve good color and brightness balance amongst tiles. Visual inspection shows good results, which can be improved by careful choice of commodity devices. Furthermore the solution is compatible with existing applications which can run unchanged.

## Categories and Subject Descriptors

C.3 [Special-Purpose and Application-Based Systems]: Real-time and embedded systems; I.4.9 [Image Processing and Computer Vision]: Applications

## Keywords

Virtual Reality, Multi-projection

## 1. INTRODUCTION

In recent years, projection walls were used by many research centers for large scale display applications in virtual environments. However, such settings still pose considerable prob-

lems regarding brightness and color balance. Unfortunately the human eye is very sensitive to variations in brightness and color nuances and can detect subtle differences in these characteristics. Since each projector has a different color gamut, in order to produce a homogeneous multi-projection display, it is necessary to balance brightness and match colors across different projector devices in a display wall, in the biggest color gamut available among all projectors.

Photometric correction in projection walls is a common task, which is usually performed manually by using spectroradiometers and visual inspection. However, this is a time consuming and difficult task which often requires periodic re-calibrations, due to normal drift in projector output. Our approach uses simple low-resolution web cameras to do this task and achieve satisfactory results, close to those achievable by manual operation. Furthermore, web cameras are about one thousand times cheaper than the more expensive, and sophisticated spectroradiometers. By collecting display images using the web camera, it is possible to assess both color and brightness errors and reduce these, in a few steps to achieve an homogeneous appearance across the display mosaic. Since our approach is geared to displays driven by computers in a cluster, the correction for a given projector is achieved by changing the gamma curve stored in the graphics card of each node that drives a projector. Moreover, as the calibration is configured directly on the graphics card, applications running on the display wall can use this calibration without any changes or reduction in performance, since the application is already ported to a distributed environment.

## 2. RELATED WORK

Many publications dealing with photometric calibration are very recent [2] [8] and most of them focus on intra-projector calibration. Although this approach creates a smoother image, it also requires extensive modifications to the image displaying software which entail severe reductions in frame rate. Other research [6] on tiled projection systems addresses gamut matching. This technique also uses cameras for image capture but sets the calibration in the projectors. Furthermore, it is recommended to do a first step color calibration in the projectors although this creates a device dependent solution which limits its utilization. Some more recent work [9] resorts to feeding light inside the projectors, through fiber optics, using one single lamp and also sharing the diachronic filter. While this solution is optimal concerning output matching, using a single lamp creates scalability

problems. Moreover small visible differences remain in the color and brightness of each projection image.

A non parametric full gamut color matching algorithm was developed [11][7] with good results, but this technique uses data set from colorimeters which are cheaper than most spectroradiometers. However these are not a commodity device and they are still much more expensive than commodity web cameras.

Optimized color gamut equations were developed to work around the color and brightness non-linear characteristics of projectors [1]. While these might provide good approximations to the color space, the resulting higher order equations are not very easy to solve. Furthermore, the quality of the results was not clear since the algorithm was not implemented.

The research project described in this paper does not cover intra-projection calibrations (the variation that occurs across the field-of-view of a single projector). This is well covered in the related work discussed above. Our approach focuses on compensating inter-projector fluctuations, this does not avoid boundary artifacts between the projection tiles, although it enables several users at the same time and no need for a user tracking system.

### 3. PROBLEM TACKLED

The main problem in multi-projection walls lies in color and brightness differences across projection tiles. Even from a fixed user position it is possible to notice large differences amongst projected areas, which negatively impairs the user experience.

As we have seen, different projectors will have different lamp characteristics both in terms of color temperature and brightness. Also, lamps age differently in both color and brightness. This means that neighbor projectors will show different colors and brightnesses given the same input in the same conditions and furthermore that even for matched projectors colors will shift differently. Also, color filter optical properties can vary for each projector, producing different spectral colors.

Moreover, the screen has some specular amount of light. To reduce this effect, projection screens usually have several layers of the same material to increase diffusion and try to achieve a unity gain. This leads to a prism effect, where refraction of incident light depends on its wavelength and the relative positions of user and projector. All these issues create chrominance variations in the resulting image which are difficult to overcome.

The projectors also use an additional white filter in the color wheel inside the projectors to enhance brightness. This lead to a complex calibration that does not fit in a linear equation. Usually more expensive projectors for virtual reality applications do not use this white filter in the color wheel, or they uses dichroic mirrors or a three DLP chip setup with static filters - one for each primary color. Some more specialized high-end projectors [4] or [3] come with a color and brightness detector inside them. Some manufacturers look for dichroic mirrors and lamps with the same behavior and

physical characteristics, to reduce the photometric calibration problem.

### 3.1 Setup

The setup used for the experiments [5] is a high-resolution 4x3 tiled back-projection wall, with a single screen and low cost projectors (HP vp6100 series digital projector). Each projector is connected to the video output (nVidia Quadro FX 3000) of a cluster node also composed of HP workstations (xw4100 workstations). A server (HP xw6200, Linux also) coordinates the cluster basic operations, such as wakeup and sleep, dispatch and control applications. Our setup is connected through a dedicated Gigabit Ethernet network controlled by the cluster server. Figure 1 shows an overview of the projector support structure. We developed special geometric calibration tools to ensure projector alignment and correct positioning of the resulting images on the screen. The projection images do not overlap on the screen to yield an image size of 4096x2304 corresponding to an effective resolution of 1 pixel/mm over a surface of almost 10 square meters.

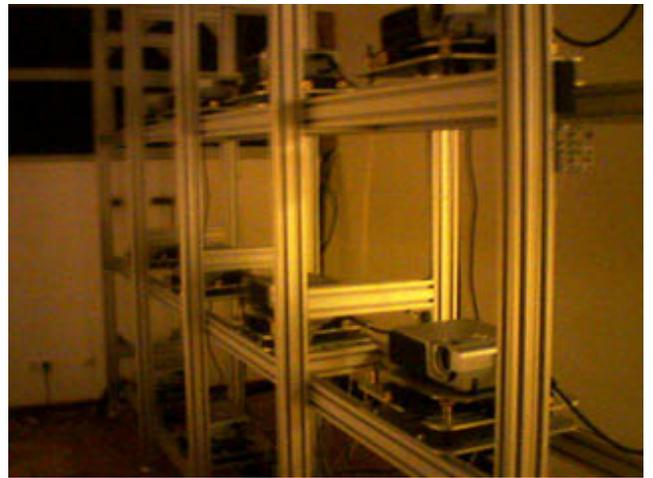


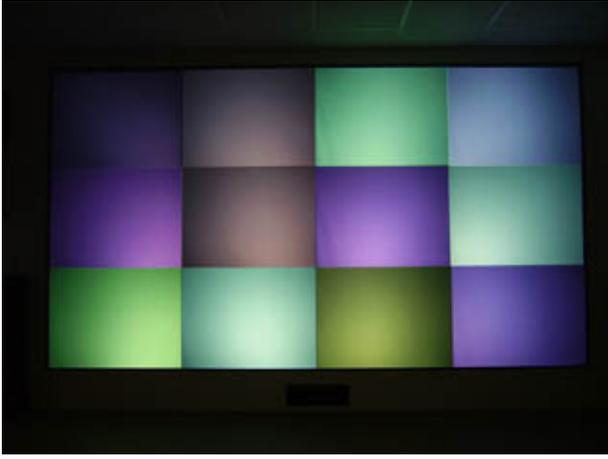
Figure 1: Projectors setup

The calibration application uses a commodity camera connected to the server to retrieve raw image information from the projection grid.

One important issue regarding the web camera is that it must support an exposure control to specify a scan frequency lower than the color wheel cycle, otherwise images will show color artifacts like those shown in Figure 2 in which each surface in the mosaic shows a different color. This is because the colors of the projectors are perceived as white due to integration in the human eye, but because each color wheel was in a different position when the picture was taken, the colors in the image are not those expected. On the other hand longer exposure times, lead to saturation problems. Another aspect of the camera is that it is more sensitive to the non lambertian effects of the screen than human eyes, increasing the hot spot effect.

### 4. MATCHING BRIGHTNESS AND COLOR

Although gamma correction control has been long available in supercomputer operating systems at device-driver level,



**Figure 2: Multi-Projection Wall**

in Linux it has only been fully available since XFree86 release 4.3.0 . Now it is possible to control the gamma curve of the graphics card by using look-up-tables. Therefore it is possible to change color values for the pixels in the graphic card, before it is sent to the display. These look-up-tables are available for each of the three primary colors (RGB).

The first step in calibrating color and brightness is to get a projection sample in order to fix the projection variations. To gather projection samples a snapshot of the projection screen is taken to be analyzed. Since there is a big difference in brightness, and sometimes in color, inside each projection area depending on the camera point of view it is necessary to provide a good measurement in order to average the colors of each tile in the display mosaic. Also, as the samples of each projection area are taken at the same time, since the web camera captures all the projection tiles in one shot, each region is not aligned with the direction of projection. Thus the brightness changes drastically along the field of view for each tile. This makes it necessary to estimate the distortion between the projectors, the screen and the camera. This approach creates a great flexibility, otherwise having to place the camera in front of each projection area in the mosaic would make the operation more complex and cumbersome.

A way used to discover the deformation is by using the image brightest spot as the focus ( $\varphi$ ), and then use it as a parameter to apply over each pixel. As it is also easy to discover the proportional width of the projection area in the image ( $\Delta w$ ), we can use the  $\cos^4(\alpha)$  rule [10] to decrease the influence as the pixels move away from the focus. We decided by this approach, because otherwise we had to know the projection, screen and camera relative positions before the calibration, making the system less flexible. With our approach the camera can be used almost immediately, the only information required being the lens aspect ratio. As our system uses a 1:1.2 lens we can use this information to discover the angular variation. For other systems this constant must be changed. The equation 1, 2 and 3 provide a good approximation for the characteristic color and brightness values of each projection surface in the mosaic.

$$\Delta\varphi = \sqrt{(x - \varphi_x)^2 + (y - \varphi_y)^2} \quad (1)$$

$$V = \sum_{i=0}^{size} RGB * \cos^4(\arctan(\Delta\varphi/(\Delta w * 1.2))) \quad (2)$$

$$v = V/size \quad (3)$$

The next step is to display the three primary colors, as shown in the Figure 3 in the display wall. The camera stores the tri-stimulus values of each individual projection tile. Initial versions used solid white images to get the tri-stimulus values, but using the primary colors yields a better result for the camera, at the cost of three, instead of one, image projection and processing steps. This is due the greater color saturation in solid white than in three separate color projections. Indeed sampling solid white images inserted greater measurement errors that threw off our calibration efforts.



**Figure 3: Primary Colors and White Mask.**

As we calibrate the brightness, the colors shift irregularly, because the projector does not have a linear response to the input color signals. This error requires more than a simple 1-dimensional lookup table for each color to properly match color spaces between projectors. Thus, changing the primary colors does not affect the brightness values as expected. This behavior leads to a difficult situation. To overcome this the calibration algorithm tries to maintain the same proportional level of each color in each projector. As digital projectors do not have a linear behavior in the primary colors, there is no practical method to find a possible color gamut common to all projectors, we decided to proceed with a iterative method looking for the best color and brightness balance amongst projectors. Since the brightness levels of projectors are already set to their maximum levels, it is not possible to increase the brightness for calibration. Our system then starts to decrease the intensity of the saturated color in each projector, and then the projectors begin to show uniformity in their color gamut. In the next calibration step, the brightness of each projection area

is averaged and decreased as necessary. Since the displayed colors do not behave linearly as we change the brightness, it is necessary to re-calibrate at each step. We usually need two or three steps to achieve good results. Special projection systems, use the opposite approach, first adjusting the brightness and then the color values. In our case, as this procedure severely reduces the projection brightness, we opted to use the opposite strategy.

As we have seen, reducing brightness creates a color shift, which makes it necessary to perform a calibration in the intermediate color input levels. Our tests suggested that discretizing color intensities in up to eight levels provides good results for the calibration and does not saturate the look up tables. Using more levels will create problems and prevent possible user manipulation of the curves. One issue that usually presents lots of problems is the darkest image setting (absolute black). This is because digital light projectors leak some light even in the absence of signal, since the DLP chip set-up scatters some of the incident light. In these cases we need increase the brightness of the darker projectors in order to maintain the same black setting across all displays. It is also important to maintain the lowest overall intensity level possible, otherwise this setting can result in too bright an image at the darkest setting.

Applying the above steps reduces the dynamic range of the projectors, and this significantly impairs the projection brightness and quality. A good way to mitigate this is to use intensity-matched projectors, since low-brightness projectors can impair the whole system. Our experiments showed that one of the twelve projectors had lower brightness ranges, as can be seen from Figure 3. In order to maintain the overall dynamic range of the projection wall we gave up calibrating that specific projector.

## 5. SOFTWARE APPROACH

The applications developed are based on a client/server architecture using sockets for the communication and XML as protocol. Each node runs a C++ client whose main purpose is to deal with color and pattern commands received from the server. The server gathers information using the camera and runs their color convergence algorithm. The server was implemented in Java. Both client and server use very distinct libraries mainly because C++ APIs are better suited for interfacing the graphics card drivers while Java Runtime Environments provide for better camera frameworks and faster user interface development.

The client receives messages to display solid colors or patterns which are read by the camera and used to compute the modifications in the color curve for each graphics card. All colors and patterns are generated by OpenGL. The client also understands message to change the graphics card behavior and settings. It implements a look-up-table used for the modifications, but the messages received by each client contain only the curve modification parameters, instead of the whole curve specification, making the procedure quite fast by avoiding to transmit redundant data.

The server is an application that receives connection requests from the clients running in each node, it is connected to the camera and has a user interface, presented in figure 4,

that enables dispatching automatic commands for each node in the cluster or manual commands for a user-defined calibration. The application also incorporates pattern commands to use in a manual geometric adjustment, that can be very useful. The patterns we commonly use include grid, color-bars, jitter (half-tone), degrade (color-ramp). The system also allows users to choose the number of grid divisions, making it possible to adapt the application to different mosaic configurations.

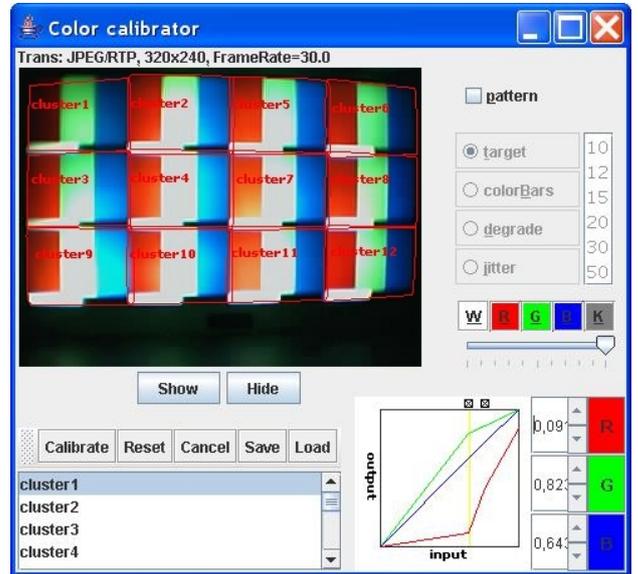


Figure 4: Interface Snapshot

## 6. FINDING PROJECTED REGIONS

Each projector is responsible for a portion of the total displayed image (mosaic). This portion needs to be correctly identified. Any failure in finding the right projected regions might yield significant errors in the following algorithm steps.

To identify each projection region two snapshots are taken for each display surface. The first projects solid black and the second projects solid white in the intended region. Subtracting both snapshots results in a picture that easily identifies the projected region. Unfortunately this approach is not bullet-proof, due to the white color interference with the environment. For example, our setup is sensitive to wall reflections and some camera lens artifacts when the solid white pattern is shown. Figure 5 shows the ceiling reflection after the image subtraction operation.

As previously reported, the algorithm does not handle erroneous regions very well. This is because the computed region is used to find the RGB values presented by the projection area. If pixels outside the projection area are taken into account, the algorithm might not converge. To extract each projected region correctly, we try to find the two vertical borders of the region. The vertical borders are obtained by computing image rows one by one and then analyzing the neighboring pixels. Each intensity value pair that exceeds a threshold is taken into account. If the difference between intensity values is low, then the pair is tagged as a false pos-



Figure 5: Subtraction Operation

itive. Figure 6 shows a false positive between rows 20 and 50.

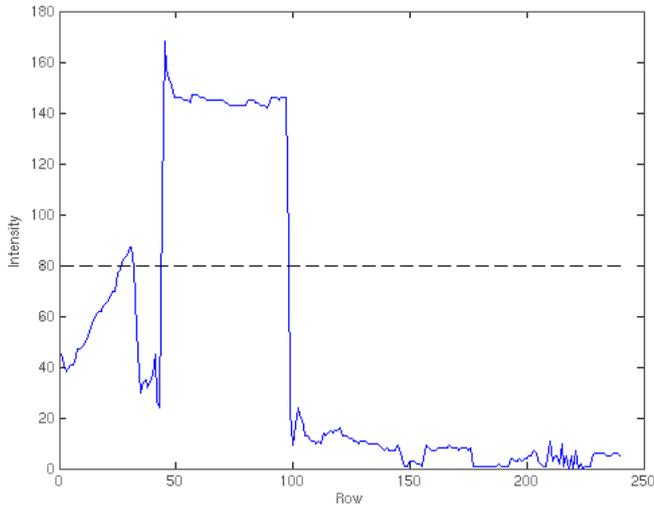


Figure 6: Single Row Intensity Value

A problem we have to deal with, is that the workaround to remove false positives may cause the image region to shrink in size. However, this artifact is more desirable than having to deal with false positives. Indeed, by using the smaller size region we are assured that every pixel included into the region is actually illuminated by the same projector we are trying to find. This allows us to use any pixel included in the region to compute the corrected RGB values. Figure 7 presents the final results of the detection algorithm. Notice how the estimated top left region is smaller than the real projected region. This is due to the fact that the corresponding projector is not working properly, thus the false positive workaround which we devised to adapt the lower intensity values. As we can see from the previous figures, the difference in brightness among projectors is quite noticeable and this adversely affects the performance of our calibration procedure.

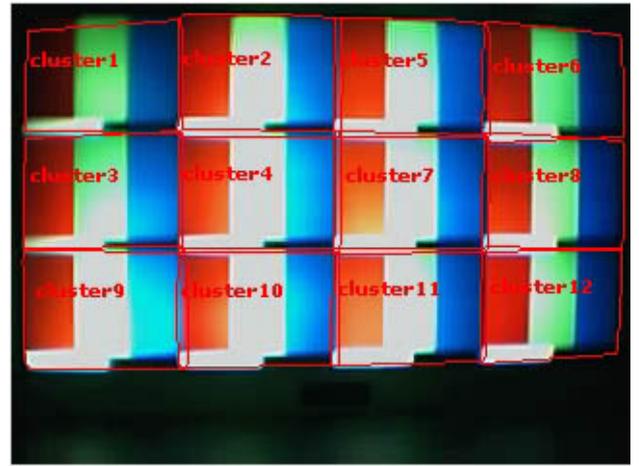


Figure 7: Image Division Obtained by the Application

To complete the detection process we apply a scan-line algorithm to identify the pixels inside the borders of each tile in the projection mosaic. These pixels are stored for further use in order to compute color values in order to calibrate future primary color images.

## 7. PROCEDURE

To start the calibration procedure it is important to wait until the system reaches a stable temperature environment, since it can change the properties of the projectors and even the screen. After that the camera must be positioned in such a way as to cover all the projection area. Usually we get better results if the camera is in the same position as the user point of view. The Java application must be started and the client application must be dispatched to each node. Then just one click in the interface starts the calibration process. Our method first detects the projector boundaries and starts to display the patterns to identify the color variation between projectors. Usually after a short number of steps (three to four) the automatic procedure stops, letting the user make final adjustments manipulating the gamma curve of each projection as desired, usually necessary if some projector is quite different from the others. Finally the calibration can be saved for future use.

A complete flow diagram of the system is presented in Figure 8. Current results point to effective use of commodity hardware to build comparatively low cost multi-projector display solutions.

## 8. RESULTS

There are some problems that prevent our solution from achieving a perfect calibration. One of the biggest issues was the fact that the projectors used have a white area in the color wheel that creates problems in finding a good equation to deal with intensities. Also, as we decrease the brightness of each projector we observed big shifts in color values which create problems in converging to a common stable setting. Furthermore, typical web cameras have very limited workable frame rates, which sometimes makes it difficult to capture a clean image of the projected area because of dis-

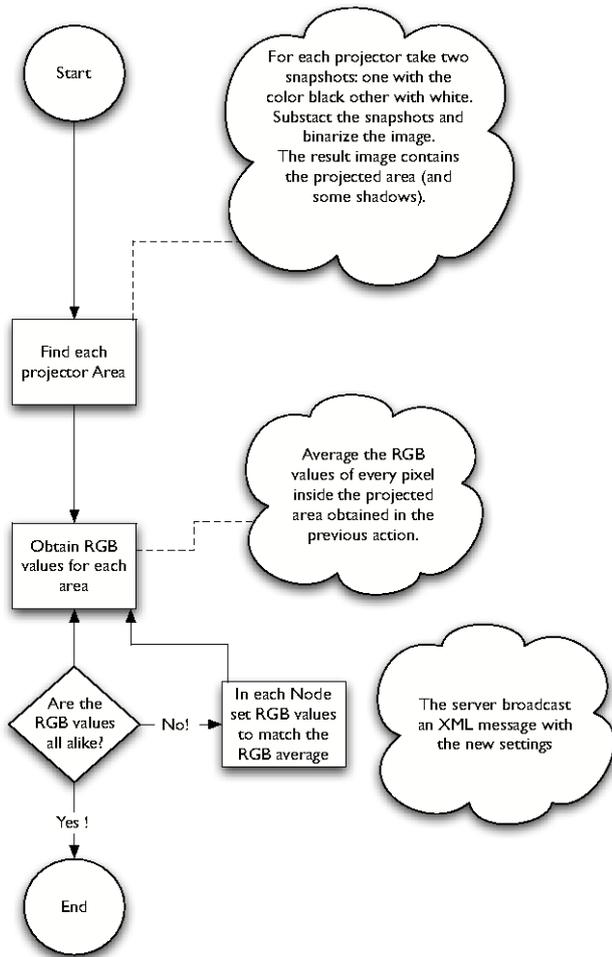


Figure 8: Complete Algorithm Flow

parities in display redraw and camera capture frequencies. Sometimes this results in white strips appearing in the captured image, which create noise. To solve this problem, as we can not genlock the camera and projector devices, since we are using commodity components that does not support this features, our software captures three images at different instants and merges them. This is fundamental for our edge detection technique to work.

Since each calibration takes some time it is possible to save the configuration and load it to individual graphics cards. This is preferable to running the calibration procedure each time it is necessary to restart the projection wall. This preload feature solves problems when it is not possible to calibrate all the system, if there are objects obstructing the camera view of the screen, such as chairs or tables. Furthermore, this setup also helps to detect changes in image quality due to lamp degradation which usually results in color and brightness shifts.

Figure 9 shows a cultural heritage scene, depicting *Mosteiro da Batalha*, a famous XIV century Portuguese monument. The image displayed depicts its main facade which was laser-

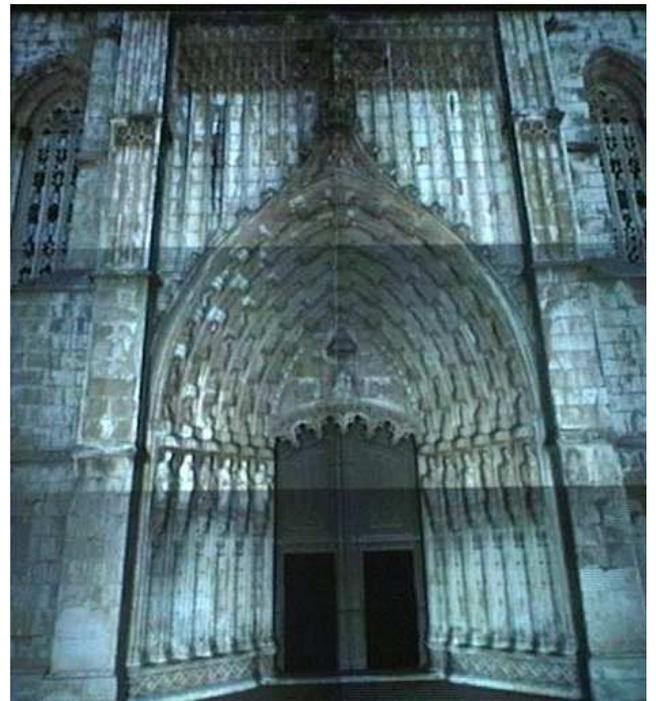
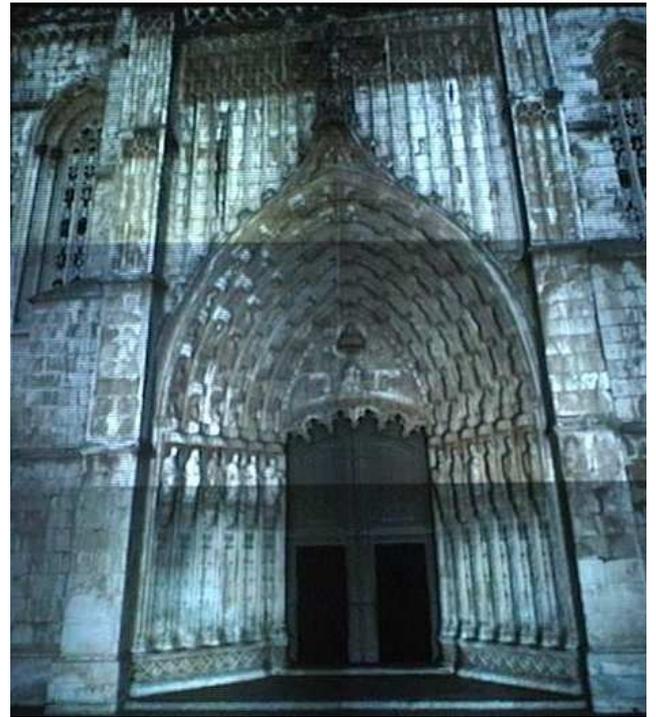


Figure 9: Cultural Heritage Display showing colors and intensities before (above) and after applying brightness and color correction (below)

scanned at high resolution – 3mm/pixel. On the topmost and bottommost pictures we can see the projection screen before and after the calibration, respectively. Since there is no overlap between adjacent projection screens in the mosaic, it is quite clear from look at the top picture where each individual screen ends. This makes it necessary to provide as good a color matching as possible. To make calibration results easier to see, both top and bottom figures zoom on the central tiles to better show the differences in color and brightness. It is possible to see that such differences are considerably mitigated in the bottom image which exhibits better homogeneity between individual projectors both in color and brightness.

## 9. CONCLUSIONS

The research work described herein allowed us to improve the color and intensity calibration between different display tiles in a multi-projection wall. Since we are driving the display wall without any overlapping region between tiles and we use a high gain screen, no technique can completely eliminate the boundary artifacts. However this more narrowly focused solution did show good results as can be seen from the images we have presented. In a different approach from pixel-based correction algorithms, this technique does not have impact in application performance. Furthermore it is possible to use it with existing applications without changing any code.

Unfortunately, it was not possible to perform a better system evaluation, and compare other algorithms or techniques to ours, since our laboratory does not currently have access to spectroradiometers or similar devices that accurately measure the quality of a calibration. Also, as shown in Figure 1 and Figure 2, the projector in the top left corner, had very different brightness and color properties, which posed considerable problems in achieving a good inter-projector calibration. We are currently replacing the projectors with less quality to achieve better results. Further research steps include testing with cameras of different quality, to assess camera influence in the calibration, as well as using different kinds of projectors and display screens. Another reasonable step is to plug-in additional software components to allow using projector controls in the calibration.

Finally as we also intend to increase overlap and apply edge blending in the near future using a precise analogical linear blending, we can predict the color variation between two neighboring projectors and program the corresponding attenuation functions directly on each graphics card.

## 10. ACKNOWLEDGMENTS

The authors wish to thank Tiago Guerreiro for his insights and help this project. This research was partially funded by the Portuguese branch of Hewlett Packard, Fundação para a Ciência e a Tecnologia (FCT) through individual grants SFRH/BPD/20572/2004 and SFRH/BD/17574/2004 and by the EU project IMPROVE (IST-2003-004785).

## 11. ADDITIONAL AUTHORS

Bruno Rodrigues de Araújo Instituto Superior Técnico - Lisboa - Portugal, email: [brar@immi.inesc-id.pt](mailto:brar@immi.inesc-id.pt)

## 12. REFERENCES

- [1] M. Bern and D. Eppstein. Optimized color gamuts for tiled displays. In *Proceedings of the nineteenth annual symposium on Computational geometry*, pages 274–281. ACM Press, 2003.
- [2] M. Brown, A. Majumder, and R. Yang. Camera-based calibration techniques for seamless multiprojector displays. *IEEE Transactions on Visualization and Computer Graphics*, vol. 11, no. 2, pages 193–206, March/April 2005.
- [3] Christie Digital. *Edge Blending User’s Manual*, 2003.
- [4] R. M. Clodfelter, D. Sadler, and J. Blondelle. High resolution display systems via tiling of projectors. *white paper, Barco Simulation Products*, 2003.
- [5] B. R. de Araujo, T. Guerreiro, R. Jota, J. A. P. Jorge, and J. A. M. Pereira. Leme wall: Desenvolvendo um sistema de multi-projecção. *13 Encontro Português de Computação Gráfica*, October 2005.
- [6] M. Hereld, I. R. Judson, J. Paris, and R. L. Stevens. Developing tiled projection display systems. In *Proceedings of Fourth Immersive Projection Technology Workshop*, June 2000.
- [7] W. Kresse, D. Reiners, and C. Knöpfle. Color consistency for digital multi-projector stereo display systems: the heyewall and the digital cave. In *EGVE ’03: Proceedings of the workshop on Virtual environments 2003*, pages 271–279, New York, NY, USA, 2003. ACM Press.
- [8] A. Majumder, Z. He, H. Towles, and G. Welch. Achieving color uniformity across multi-projector displays. In *Proceedings of the 11th IEEE Visualization*, page 17, October 2000.
- [9] J. L. Moreland and S. Reinsch. A single light-source uniform tiled display. *SDSC White Paper*, 2003.
- [10] M. C. Stone. Color and brightness appearance issues in tiled displays. *Computer Graphics and Applications, IEEE*, 21:58–66, Sep/Oct 2001.
- [11] G. Wallace and K. L. Han Chen. Color gamut matching for tiled display walls. In *Proceedings of the workshop on Virtual environments 2003 EGVE ’03*, pages 293–302. ACM Press, 2003.