

# Solving Rendering Issues in Realistic 3D Immersion for Visual Rehabilitation

Tristan Carrier-Baudouin, Claude Chapdelaine, Marc Lalonde,  
Philippe Quinn, and Samuel Foucher<sup>(✉)</sup>

Computer Research Institute of Montréal (CRIM), Montréal, QC, Canada  
{tristan.carrier-baudouin,claudc.chapdelaine,marc.lalonde,philippe.quinn,  
samuel.foucher}@crim.ca

**Abstract.** When a person becomes visually impaired, intensive rehabilitation is required to learn the skills necessary to accurately interpret the sensory cues. 3D virtual immersion (3D VI) can provide a safe and rich world for rehabilitation by rendering a complex environment. 3D VI can stage challenges that can be faced with less stress than the real outdoor world. In order for 3D VI to be used efficiently, the visual rendering for people with low vision and the audio for both the blind and people with low vision must be as realistic as possible. Building a 3D environment at low cost imposes constraints both on the installation and the technical aspects. This paper describes our current work to solve projections issues for a realistic rendering when images are not aligned and deformed.

**Keywords:** Visual rehabilitation aid · 3D immersion · Multi-projector calibration

## 1 Introduction

For people with blindness, knowledge of the environment depends mainly on the interpretation of auditory information available to ensure a safe and efficient travel. Likewise, people with low vision have to learn strategic use of their residual vision while learning how to process appropriate auditory information according to the environment. The research results on the use of 3D VI in visual rehabilitation suggest that the gains made in virtual environments are partly observable in real world settings [1,2]. Blind people gain skills in orientation and mobility (O&M) when they move in 3D VI guided only by auditory cues. This enables them to effectively correct their trajectory in case of deviation and improve their ability to focus on useful cues.

This article presents the constraints related to a low-cost realistic 3D virtual immersion installation that renders a simulation of an urban street designed to assist visual rehabilitation. The fact that the immersion installation is made of a mobile wall that can be rearranged requires a flexible solution with simple equipment and an easy calibration procedure. The paper describes the physical and technical issues that were addressed to solve misalignment and deformation

problems related to the 3D projection. It details a successful solution based on intra/inter geometric calibrations with a single webcam and discusses future work.

### 1.1 Description of 3D VI for Vision Rehabilitation

We built a simulated scene (Fig. 1) to assist the learning of how to align to traffic, which is the first step in an O&M outdoor rehabilitation training. This simulation does not cover all aspects of O&M outdoor training but enables four tasks to be mastered indoor before going out into the real world: (1) detecting the direction of incoming/outgoing sounds of vehicles (cars, motorcycles), (2) evaluating the distance between oneself and the vehicles, (3) positioning oneself parallel or perpendicular to traffic and (4) approaching traffic safely.



**Fig. 1.** Actual urban street simulation for visual rehabilitation in 3D VI

### 1.2 Challenges and Installed Solutions

Rendering realistic sounds for proper sound interpretation implies taking into account reverberation, distance and direction of objects emitting the sounds [3]. The existing sound databases available for gaming development were not realistic enough since sounds from vehicles are mostly the engine sounds heard from the inside of the vehicle. We needed not only engine sounds from the outside but also the sound of the tires on pavement and the sound of wind on the vehicles body at various speeds. Sound databases used by audiologists are realistic but not in a format that could be fed to a game engine. We recorded the needed sounds with the assistance of an audiologist and our acoustical surrounding sounds were combined in Unity <sup>1</sup> with *Fmod* <sup>2</sup> for a more accurate range of frequencies. But in order to render these sounds accurately in the 3D VI, the loudspeakers had to be placed at ear level (i.e. mid-height of the structure). This added another constraint on the environment since we needed to hide the loudspeakers in order to get a realistic visual rendering.

For rendering realistic and accurate urban settings for people with low vision at the proper scale, we used guidelines that were produced by a group of O&M

<sup>1</sup> <https://unity3d.com/>.

<sup>2</sup> <http://www.fmod.org/>.

specialists [4]. The perceived length and depth were correctly reproduced and had to be properly projected without deformations or discontinuities. The scene contains a realistic street bordered with trees, houses, commercial buildings and parking lots. The street is approximately one kilometer long. The vehicles can pass by at various realistic speeds, ranging from 35 km/h to 70 km/h. Parked cars can also be added. The user experiences the scene through a first-person view. The person may rotate and take a few steps to align oneself and can also take a few steps to move toward traffic. Furthermore, the simulation can be started from a dozen available points of view.

After visiting and working with specialists in a few Rehabilitation Centers, it became evident that they did not have the space and/or the budget to afford a large and expensive virtual environment. This added the constraint of building the lowest possible cost for 3D environment without jeopardizing efficiency. The lab 3D VI room is a four by three meters with a projection height of 2.44 m in a room of 4 m height. The INLB<sup>3</sup> installation is 4.2 by 4.2 m with a projection height of 2.22 m in a room of 3 m height. It is also equipped with three projectors to cover the left, right and back walls of the room.

The 3D VI consists of a cube-shaped metallic structure assembled from off-the-shelf galvanized steel square tubes and angles screwed together. A white and flexible vinyl fabric is affixed to the four walls of the structure. There is a small entrance and only the middle, eastern and western walls are being projected onto, creating a field of view of approximately 250°. It should be underlined that the projectors are short-throw video projectors fixed to the top of the metal structure. All three projectors are connected to a single computer.

Our first installation with O&M specialists revealed that the visual accuracy was inadequate for the envisioned training with low vision people. Since loud-speakers had to be hidden by curved corners of flexible vinyl sheets to provide the best acoustic possible, it caused deformation to the image projected unto those curved surfaces. Also, the limited height of the room did not allow for optimal installation of the projectors and the projection on the eastern wall had to be tilted, thus creating a deformation at the lower left corner. We saw these constraints as challenges in need of a low-cost solution. This article describes an automatic method capable of successfully calibrating the projectors using a low-cost webcam mounted on a tripod.

### 1.3 Projections in 3D VI

Unless the projector's axis is perfectly perpendicular to a planar wall, projected images will be distorted in a various ways [5]. Displays are often non-planar, either to improve immersivity [6] or for other reasons such as in our case with the curved corners. Moreover, two projections can be distortion free, but might not be well aligned with each other. In order to avoid black gaps, projectors are usually positioned so that there is a slight overlap between the views. This

---

<sup>3</sup> Institut Nazareth et Louis-Braille.

overlap zone is more luminous and detracts from the rendering quality. Projector calibration aims at correcting these visual issues [7].

Visual correctness of the image is important, because the visually impaired person is expected to take his or her training acquired within the 3D VI to the real world. Being trained with flawed images might reduce the rehabilitation potential of the immersive experience.

Brown et al. [7] describe many approaches that we will review. There are two types of geometric calibrations that need to be performed for each projector: (1) Intra-projector calibration refers to correcting the deformations introduced by non-planar screens or imperfect alignment of the projector with respect to the planar screen [7]; and (2) Inter-projector calibration means adjusting the images of two neighboring projections so they are well aligned with each other (i.e. there are no discontinuities in a line passing through the display [7]). Blending is also part of the calibration process and consists of uniforming the luminosity across the entire projection field [7]. The presented method includes intra and inter geometric calibration, but not blending.

A method capable of calibrating projectors in the context of a low-cost 3D immersion environment is presented. This method relies on a computer vision approach and only needs an inexpensive webcam, even though the field of view of the immersion environment is close to  $250^\circ$ . The method preserves the perspective effect and is based on Alberti's procedure for drawing geometrically correct tiled floors [8].

## 2 Related Work

The purpose of projector calibration is to obtain a seamless undistorted image with uniform luminosity, regardless of the number of projectors and the shape of the screen surface. A large array of methods are available to obtain such a result. Calibration can be performed through manual adjustments by an expert using special equipment [6]. However, this option is costly because of the regular maintenance, equipment and personnel necessary [7]. It must also be carried out every time the projector or the screen are displaced. In the case of this work, the fact that there is a mobile wall precluded the use of the manual method.

Brown et al. propose another approach based on 3D modeling of the screen including projector positions and parameters [7]. The projection can then be simulated on a computer, with the objective of calculating the appropriate distortion for each projector. This method is convenient when the whole installation is unmovable and the shape of the screen is fixed. Unfortunately, it is not our case.

The third general class of methods consists of using a computer vision approach to calibrate projectors. A grid or a set of features ordered in a rectangular manner are projected on the screen. As explained in Brown et al. [7], the calibration process follows two main steps. During the first step, the camera acquires images of the screen and establishes a correspondence between camera coordinates and projectors coordinates. The second step consists of using this information to warp and blend the images sent to the projectors. Warping means

deforming an image in order to obtain spatial continuity between the projections as well as compensating for the curvature of the screen. Blending means adjusting the luminosity in order to make overlaps indistinguishable. Blending typically relies on knowing the correspondence between camera coordinates and projectors coordinates [9]. For the remainder of this section, methods relying on computer vision will be discussed.

In Brown et al., a rectangular array of points is displayed on the screen for each projector [7]. The correspondence between the points viewed by the camera and the projected points is then established. From the camera point of view, the grids are distorted and unaligned with each other. From the projectors point of view, each grid is perfectly square. Both the distorted and undistorted grids are triangulated in order to create a model for texture mapping. The desired image from the simulation is then input to the texture buffer of the distorted grids provided by the camera. Next, a texture mapping between the distorted grids and the undistorted grids given by the simulation is created. The undistorted grids are textured and the result is a warped image that, when projected on the curved display, appears undeformed and well aligned. This method produces excellent results. However, the entirety of the image needs to be visible to the camera, i.e. the camera field of view needs to cover the whole scene. In our case, in order to respect the low-cost constraints, inexpensive cameras were used during this work. Webcams have limited fields of view and a single camera cannot observe the whole immersive environment at the same time. As said earlier, the environment covers an angle of approximately  $250^\circ$ . Moreover, this described method may remove too many deformations depending on the camera location. Only the deformations caused by the curved corners should be removed, not the natural deformations caused by the perspective effect. Let's assume that a person looks at a standard straight corner between two walls. If square grids are superimposed on the walls, the person will observe that the horizontal lines are straight, except in the middle of the corner where there is an abrupt change of direction. Such deformation is caused by perspective and should not be removed.

Garcia-Dorado and Cooperstock make use of homography techniques in order to calibrate projectors [5]. The walls can be planar or curved. If curved, then they must be modeled with OpenGL. The method relies on a motorized pan-tilt camera in order to cover a very wide field of view. Excellent results inside immersive environments can be obtained. In particular, perspective effects are not removed. However, as it was said earlier, the OpenGL modeling of a wall is not very practical when the display undergoes slight modifications from time to time. Moreover, it would be difficult to use this method for an immersive environment when a low-cost webcam is used.

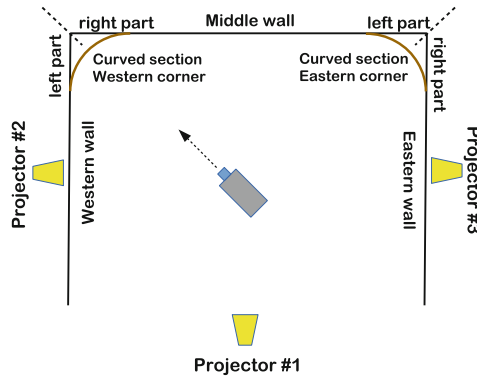
Sajadi and Majumder have designed a method specifically for immersive environments. It assumes that the display is a swept surface and requires the user to input the rotation angle of the profile curve [10]. While this method produces excellent results, a pan-tilt camera is again necessary when the whole screen covers a wide angle. This method cannot be applied to immersive rooms when only a low-cost webcam is provided.

Van Baar et al. have published a method designed for quadric displays [6]. It is capable of calibrating multiple projectors and it can produce distortion free images on curved surfaces, assuming they are quadric. It requires one camera by projector, in other words a standard camera is attached to each projector. This method produces very good results, but the displays are limited to quadric surfaces.

None of the methods described in this section seems to satisfy all the installation and cost constraints mentioned in the Introduction. It has therefore been necessary to develop a new procedure based on computer vision.

### 3 Requirements and System Design

As shown in Fig. 2, our 3D immersion environment bears certain similarities with classical CAVEs. One difference is the presence of curved sections in the corners.



**Fig. 2.** The 3D immersive environment with the camera aiming at the Western corner

In Fig. 2, projector #2 displays an image on the eastern wall, projector #3 displays an image on the western wall and projector #1 displays an image on the middle wall. The camera can only observe one corner at a time. In order to increase the vertical field of view, the webcam is rotated 90° (portrait mode). The left and right parts of each corner are illuminated by different projectors.

The simulation used to train visually impaired people has been created with the game engine Unity. The Unity code contains three virtual cameras, each one sending the images they acquire to a projector.

Unity can be programmed to correct the distortions caused by the curved corners. The first step consists in accurately modeling the immersive display in 3D. The next step is to render the scene on the 3D model. Three virtual cameras, each one located exactly where there should be a projector in the scaled model, film the modeled display and send what they see to the actual physical projectors. As mentioned in Sect. 2, the main hurdle is the 3D modeling

of the display. The display surface may undergo changes and it is flexible. Also, it does not follow any well defined geometrical surface.

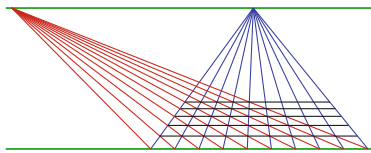
Another approach is to model the immersive display without the curved corners. The walls are assumed to be entirely planar. A mesh is applied to each of the three walls. Displacing mesh vertices warps the rendered scene on the wall in a predictable manner. In fact, the scene rendering is used as a texture mapped to the mesh. Unity could then read a file containing the new vertex positions necessary to correct the distortions caused by the curved corners<sup>4</sup>. The scene would appear warped from the projectors' point of view, but undistorted and well-aligned from the user point of view. This approach was implemented and tested. A method capable of creating such a file will be presented.

## 4 Projector Calibration

In this section, an algorithm capable of aligning the images created by the three projectors in the immersive environment and compensating for the two curved corners is presented. The algorithm currently does not perform blending, i.e. luminosity correction in the overlap regions. However, warping and geometric calibration is an essential step before blending [9].

### 4.1 Alberti's Method

Alberti's method refers to a very old technique used by painters to create a geometrically correct tiling from a perspective viewpoint [8]. A tiled floor is composed of straight lines. This method relies on the fact that a straight line remains straight regardless of the viewpoint. Of course, two parallel lines may intersect if viewed from elsewhere.



**Fig. 3.** Alberti's method for tiled floor drawing (Color figure online)

In Fig. 3, the top horizontal line (in green) is the horizon line. The bottom horizontal line is parallel to the first and is divided into equal spacing corresponding to the floor tiles. The blue lines intersect at the vanishing point and the red lines intersect at the secondary vanishing point. The horizontal black lines are the desired floor lines and go through the intersections between the red lines and the blue ones.

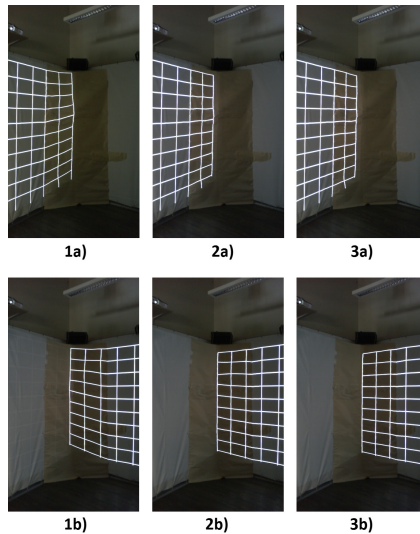
<sup>4</sup> More specifically, it is the *vertices* property of the Unity *Mesh* that is modified. The three meshes are observed by three Unity *Cameras*.

## 4.2 Iterative Algorithm

This section discusses intra- and inter-projector calibration. Both consist of iterative processes that use the camera feedback to improve the visual aspect of the projected grids. If some grids can be correctly displayed, then the more complex rendered images of real-world scenes will also be correctly displayed without misalignment and distortion.

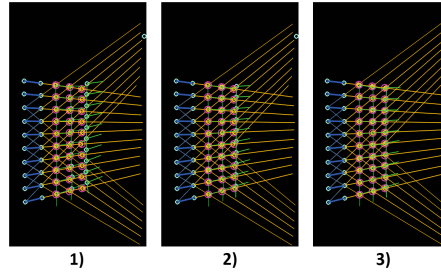
**Intra-projector Calibration.** At the beginning, a grid is projected on one side of the corner, the left or the right. Figure 4(1a) and (1b) show a grid being displayed on the left and right parts of the corner respectively. It is to be noted that they are rendered by two distinct projectors. They are also as wide as the projectors allow. The grid vertices are detected using functions from the OpenCV computer vision library [11]. All the grids used for this work have a size of  $11 \times 11$ .

Figure 5 illustrates the perspective correction process. The grid vertices that fall on the planar wall are called unmovable vertices. Those that fall on the curved section are called movable. There are two columns of unmovable vertices and three columns of movable vertices. Both are pale blue in the figure. These numbers are specified by the experimenter based on his or her observations. The camera does not see the entirety of the grid on the left or right side, but this is not an issue as long as there are two columns of unmovable vertices or more.



**Fig. 4.** Grid projected on the left part of the corner, from not corrected (1a) to most corrected (3a). Not optimal iteration in progress (2a) and 2(b). Grid projected on the right part of the corner, from not corrected (1b) to most corrected (3b)





**Fig. 5.** The iterative process based on Alberti's method, from not corrected (1) to most corrected (3) (Color figure online)

Two points are necessary to determine a line and two columns are necessary to determine the vanishing lines.

The curved sections of the immersive room used at INLB are made of white vinyl. In our lab in order to create a more challenging situation, we used brown paper to mimic the rounded corners. In Fig. 4, one can observe that there are three columns of vertices on the curved section made of brown paper, and two columns of vertices on the planar wall made of white vinyl.

The unmovable vertices are assumed to be correct from a perspective viewpoint. They are used to evaluate where the movable vertices would be if there was no curved section. The goal is to displace the movable vertices at these locations, in order to make the distortion caused by the curved section disappear. As mentioned before, the whole operation is iterative, which implies moving the vertices, detecting them with the use of the camera and computing the new displacements.

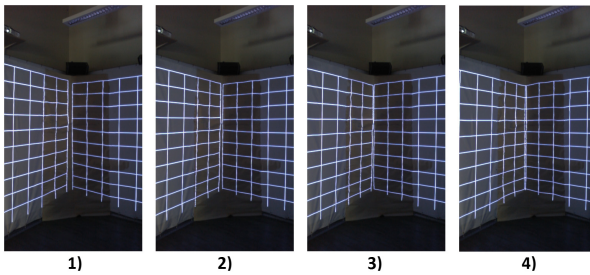
Vanishing lines (the red and blue lines in Fig. 3) are retrieved based on the unmovable vertices. These lines are extended into the curved section of the corner and their intersections represent the desired locations of the movable vertices. In Fig. 5, the yellow lines are the vanishing lines from Alberti's method. They are traced based on the two columns of unmovable vertices. Two is the minimum number of columns necessary to determine the vanishing lines, but more can be used if the camera sees them. The yellow points are the intersection points of the vanishing lines. Each column of yellow points should form a perfectly straight line. However, because of the irregularities in the vinyl screen and the imprecision in the OpenCV vertex detection, it is not always the case. Therefore, a least squares method is applied to find the best fit line that goes through each column of yellow points. The big red points are found by intersecting the best fit lines and the horizontal yellow vanishing lines. The red points represent the ideal locations. Movable vertices are iteratively displaced to these locations. The changes are made iteratively because of the non-linearity of the problem. The displacement vectors are transformed from the camera system of coordinates to the projector system of coordinates using a change of basis matrix based on the local vertical and horizontal segments of a grid cell. These segments are green in

Fig. 5. There is a relaxation factor of 0.5. In other words, a vertex is not moved to the ideal location, but halfway of it. The intent is to facilitate convergence.

Figure 4(2a) and (2b) show the resulting grids after 2 iterations. Figure 4(3a) and (3b) show the resulting grids after 6 iterations. Three iterations are sufficient, because they will be repeated many times during the inter-projector calibration process.

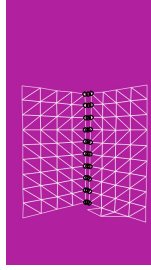
**Inter-projector Calibration.** The intra-projector calibration is embedded in the inter-projector calibration and the whole process must be repeated several times. The left and right sides of the corner are corrected in alternation. After three iterations on each side, the grids on the left and right are moved either closer together or farther from each other depending on whether there is a gap or an overlap. For example, 5 cycles were necessary to produce the results presented later in this article. Each cycle is composed of 3 Alberti's iterations for the left side (intra), 3 Alberti's iterations for the right side (intra) and one step of moving the whole grids (inter). These numbers were found by trial and error to give visually acceptable results. The whole process hopefully converge toward an acceptable solution. After a certain number of iterations and cycles, improvements become non-existent. At this point, grid vertices only oscillate around a certain position. More iterations would only consume more time.

Figure 6(1) shows a case where there is gap instead of an overlap. The gap is reduced until the two grids collide, as illustrated in Fig. 6(3).



**Fig. 6.** Inter-projector calibration process, from the beginning (1) to the end (4)

The camera coordinates of the left and right border vertices that should coincide are known. It is therefore possible to compute the inter-projector calibration with the middle ground between these two locations. The middle point minus the actual location of the border vertex forms the displacement vector in camera coordinates. The displacement vector in projector coordinates can be computed for each of the border vertices using a change of basis. These displacement vectors are then averaged. The whole grids are moved according to these average displacement vectors until they collide without overlapping. In an attempt to improve convergence, a relaxation factor of 0.9 is used, i.e. the displacement



**Fig. 7.** The left and right grids in the process of being connected together

vector in projector coordinates is multiplied by 0.9. Figure 7 shows the left and right border vertices along with the middle points.

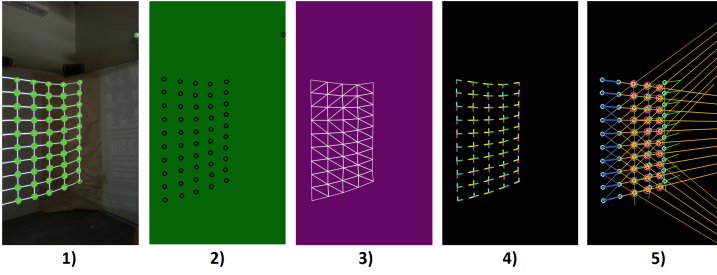
Because the border vertices of the left and right grids may not form perfectly parallel lines, the final step consists in connecting the corresponding border vertices two by two, ignoring perspective which needs to be repeated three times. Distortion files are written immediately after. This step makes any gap disappear even though the perspective may be slightly inexact, as shown in Fig. 6(4).

Each of the two corners is corrected separately, creating four distortion files to be read by the Unity simulation. The distortion files for the right part of the western corner (WR) and the left part of the eastern corner (EL) both describe how to warp the image of the middle projector. They contain the new locations of the grid vertices in the projector coordinate system. The two files are therefore merged together. Some grid cells from the planar section may need to be stretched, especially if the two grids have been moved toward the corners, but the visual impact is not very noticeable. Also, the grids projected on the middle wall are never moved vertically during the correction process. Only the grids on the eastern and western walls move vertically. This is necessary in order to be able to connect the grid from the WR file with the one from the EL file.

In the event that a projector is tilted, an additional parameter can be introduced to rotate the grid by a certain amount of degrees. The parameter is chosen by the user. The rotation origin is the upper left vertex for a grid displayed on the right part of the corner, and the upper right vertex for a grid displayed on the left. In the current implementation, upper vertices should always be visible, because they are used to determine the left and right neighbors of each vertex in the grid. In return, knowing the right and left neighbor of each vertex allows the program to draw the vanishing lines of Alberti's method.

### 4.3 Implementation Details

Figure 8(1) shows the detected grid vertices. Each vertex is a corner created by the intersection of a vertical and horizontal grid segment of non-zero width. The OpenCV method *goodFeaturesToTrack* was used to detect these intersections. In Fig. 8(2), the points are aggregated in order to have only one vertex by intersection. In Fig. 8(3), the vertices are triangulated following Delaunay's method. For



**Fig. 8.** Detected features (1), vertices after aggregation (2), triangulation (3), graph of neighbors (4), Alberti's method (5)

each vertex, the top, bottom, left and right neighbors are identified in Fig. 8(4). Alberti's method is displayed in Fig. 8(5).

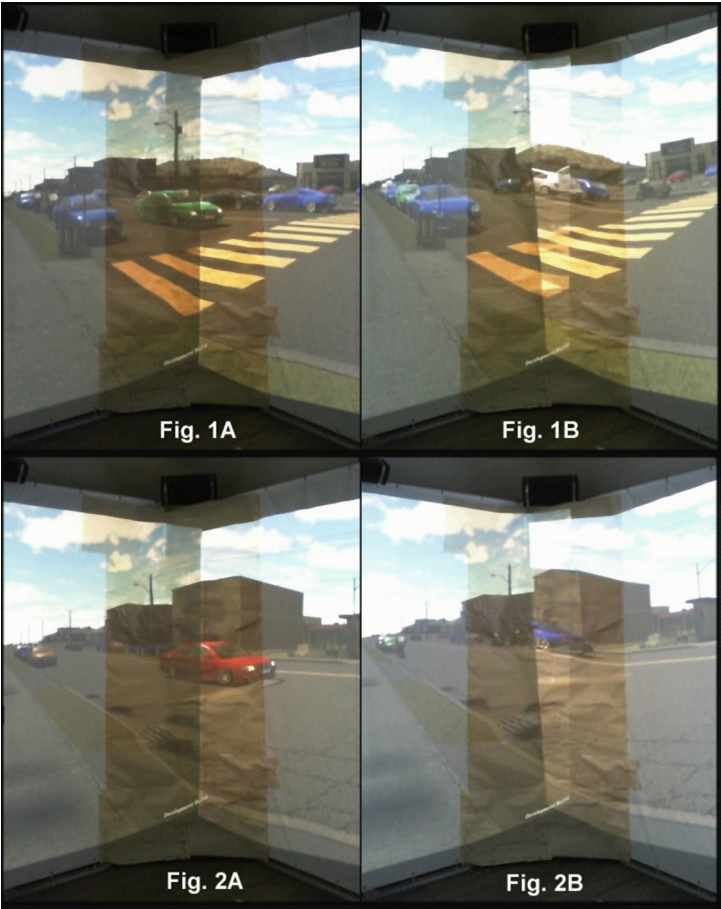
The calibration algorithm was implemented in C++ and uses the FreeGlut graphical library for displaying the grids [12]. Calibration time is approximately 7 min per corner. The camera does not need to be calibrated, but should at best be level with the floor.

## 5 Experiments and Results

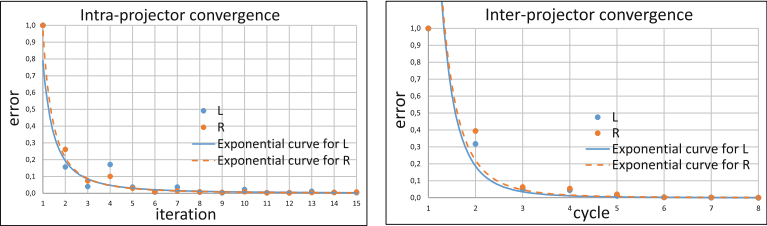
Distortion files for the 3D immersion environment were created by the implementation discussed in this article. In this section, screen captures of the scene generated by the Unity simulation after correction are shown in Fig. 9. Corrected and uncorrected versions of the same point of view are displayed side by side in order to facilitate comparison. As seen in Fig. 9(1B) and (2B), uncorrected images are afflicted by excessive overlap, misalignment of the left and right parts as well as distortions induced by the screen curvature. A slight rotation of the projection on the right part of the corner is also visible. On the contrary, Fig. 9(1A) and (2A) contain minimal overlap. The alignment was also restored. While the distortion created by the screen curvature was reduced, smaller distortions generated by the wobbliness of the paper were not eliminated. This mishap could possibly be due to the coarseness of the  $11 \times 11$  grid.

The luminosity on the darker paper is much lower than on the white vinyl. Such an issue is not present in the actual immersive environment used for rehabilitation. Five cycles of 6 iterations (3 left and 3 right) were necessary in order to obtain these images.

Figure 10 shows two plots of the error as a function of the number of iterations/cycles. For intra-projector calibration, error is defined as the sum of the squared vertex displacements required to comply with the perspective retrieved by Alberti's method. For inter-projector calibration, the error is defined as the sum of the squared vertex displacements necessary to avoid an overlap or a gap between the left and right grids. As seen from the plots, both the intra- and inter-projector calibration processes converge. Three iterations for the left and right sides were performed for 5 cycles, which means 15 iterations, in addition to three final cycles that join the border vertices together while ignoring perspective.



**Fig. 9.** Two corrected (1A, 2A) and uncorrected views (1B, 2B) of the Unity scene



**Fig. 10.** Convergence plots: Normalized error as a function of the number of iterations (left) and cycles (right)

## 6 Conclusions

A 3D immersive environment has been presented. Both the visual and the audio components of a street with moving cars were simulated. The objective was to offer a safe, realistic and efficient setting to train blind and low-vision people in urban orientation and mobility. The rehabilitation of low-vision people can be maximized if the projected image is free from distortions and misalignments. A method capable of calibrating projectors while respecting cost constraints was devised. This method uses camera feedback to correct projected grids iteratively. The desired positions of the grid vertices are given by Alberti's perspective technique. The warping used to obtain visually correct grids is applied to the simulation, thus yielding an accurate representation of a street with traffic.

The current method has some limitations. In particular, the luminosity is not entirely uniform, especially in the thin overlapping regions. Blending would be part of a future work and would consist in attenuating the zones of higher luminosity. Additional tests in other immersive rooms would also be welcome in order to evaluate the robustness of the algorithm.

**Acknowledgment.** We thank the O&M specialists of INLB for their feedback. Our gratitude to Dr. Tony Leroux (audiologist), Pr. Agathe Ratelle (O&M specialist trainer) and Dany Vaillancourt for their creative and helpful skills. This work is supported by CRIM, ISED (Innovation, Science and Economic Development Canada), and le MESI (Ministère de l'Économie, de la Science et de l'Innovation du Québec).

## References

1. Merabet, L., Sánchez, J.: Audio-based navigation using virtual environments: combining technology and neuroscience. *AER J. Res. Pract. Vis. Impairment Blindness* **2**(3), 128–137 (2009)
2. Sánchez, J., Saenz, M., Garrido, J.M.: Usability of a multimodal video game to improve the navigational skills for blind children. *ACM Trans. Accessible Comput.* **3**(2), 7 (2010)
3. LaGrow, S.J.: Improving perception for orientation and mobility. In: Wiener, W.R., Welsh, R.L., Blasch, B.B. (eds.) *Foundations of Orientation and Mobility*, vol. 2, 3rd edn, pp. 3–26. AFB Press, New York (2010)
4. Couturier, J., Ratelle, A.: *Manuel d'Intervention en Orientation et Mobilité*. Université de Montréal and INLB, Québec, Canada. Course Notes (2012)
5. Garcia-Dorado, I., Cooperstock, J.R.: Fully automatic multi-projector calibration with an uncalibrated camera. In: *Proceedings of the Computer Vision and Pattern Recognition Workshops*, Colorado Springs, pp. 29–36. IEEE, June 2011
6. Baar, J.V., Willwacher, T., Rao, S., Raskar, R.: Seamless multi-projector display on curved screens. In: *Proceedings of the Workshop on Virtual Environments*, Zurich, pp. 281–286. ACM (2003)
7. Brown, M., Majumder, A., Yang, R.: Camera-based calibration techniques for seamless multiprojector displays. *IEEE Trans. Visual Comput. Graphics* **11**(2), 193–206 (2005)
8. Stillwell, J.: *Mathematics and its History*. Springer, New York (2010)

9. May, B.B., Cahill, N.D., Rosen, M.R.: Calibration of a multi-projector system for display on a cylindrical surface. In: Proceedings of the Image Processing Workshop, Rochester, New York, pp. 6–9. IEEE (2010)
10. Sajadi, B., Majumder, A.: Auto-calibration of multi-projector cave-like immersive environments. *IEEE Trans. Visual Comput. Graphics* **18**(3), 381–393 (2011)
11. Bradski, G.: The opencv library. *Dr. Dobb's Journal of Software Tools* (2000)
12. Olszta, P.W., Umbach, A., Baker, S., Fay, J.: The freeglut library (version 3.0.0). (2015). [freeglut.sourceforge.net](http://freeglut.sourceforge.net)