

A Platform for Spatial Augmented Reality

Julio Lucio Martin
IMPA
juliolucio@gmail.com

Luiz Velho
IMPA
lvelho@impa.br

Abstract

This document describes the development and first experiences of the S.A.R. platform ("Spatial Augmented Reality"). This document also describes the first two pieces of software developed for this platform. The first of them generates a set of transformation matrices that let us to transform geometries from 3D virtual space to the 3D real space. The second allow us to design tests with simple geometries onto de real geometries. We also show the configuration and devices used in the first experiments and a few images of the results.

1 Introduction

We developed a project in virtual augmented reality, and did some experiments in this area, through the development of a system that is able to lighting and texturing dynamically, the surfaces of convex white objects.

To this end, we used a projector that will place textures, lights and shadows on the object's surfaces. To calibrate the projections we used an algorithm developed at VISGRAF Laboratory. This algorithm allow us to use a reference object, to obtain information about the transformations that we should apply the image to be mapped on the projector space.

Once we have calculated these transformations in the projector space, we are able to do the projection on the reference object.

2 Previous Works

Our research is based on previous work in S.A.R., where different techniques were used for lighting and texturing environments and objects dynamically. The technique of spatially augmented reality has its origins in the seminal work "Office of the Future" by Henry Fuchs at North Carolina University [7].

The first work by Ramesh Raska and colleges [6][8][1] describes a new way of visualisation of 3D computer graphics, which involves light projectors and physical objects to

generate images rich in detail directly in the user's world. Although the method is limited when compared to traditional graphics rendered on computer's screens, it offers a new way to interact with synthetic images. They presented new lighting techniques based on images of non-trivial objects. A rendering process involves essentially the user's point of view, the shape of graphic objects, the refraction and lighting properties. The traditional computer graphics or traditional head augmented reality generates the results for all these elements with a temporal reduction rate (frame rate) or spatial resolution (pixel). With shader lamps, they try to keep the point of view and the surface at the highest resolution, and only the additional colour information stays at a limited resolution.

The second work from Deepak Bandyopadhyay and coauthors, presents a new 3D painting augmented reality painting system onto real objects. It requires two projectors facing each other, a tracker attached to the object (on the brush), both projectors to be calibrated to the reference framework of the tracker, the object to be fairly coloured and diffuse, and that the object's geometry and its texture coordinates are pre-computed. More references on this and other works can be found in course materials and books about the subject[5] [3] [4].

Our initial inspiration for this project was also the work of artist Pablo Valbuena, who developed an artwork, using a projector to illuminate some square base prisms. He achieved these results through the building of the geometry in the real world and then he modelled geometry in the computer and rendered them with a few lights, using a 3D software. The resulting rendered movie was projected on the geometries creating the illusion that the light was flying around them. [2]

3 Description

Our platform is composed of two parts. A set of hardware components and various software applications. At first, the hardware configuration consists of a computer with high end graphics capability, a projector, a base space and support objects.

In this section we describe both the hardware and software aspects of the system.

3.1 Devices

In the laboratory, we built a platform made of planar surfaces that form an angle of ninety degrees, which is serving as a basis for projection.

We put the projector hanging from the ceiling oriented in the direction of the base, allowing it to illuminate the platform with a lateral projection.

Then we construct a cubic object which when placed on the platform, allows us to capture the positions of some specific points of the cubic geometry for calibration.

- Base Space or projection platform. This base space (figure. 1), is a coordinate space in the real world. That means that has a position and orientation. It is the space where the projector illuminates the geometries that are in it. For our first experiments, we have built over a table at the lab planes of 90cm x 60cm foam white paper, and marked on it the origin of the base space coordinates system.

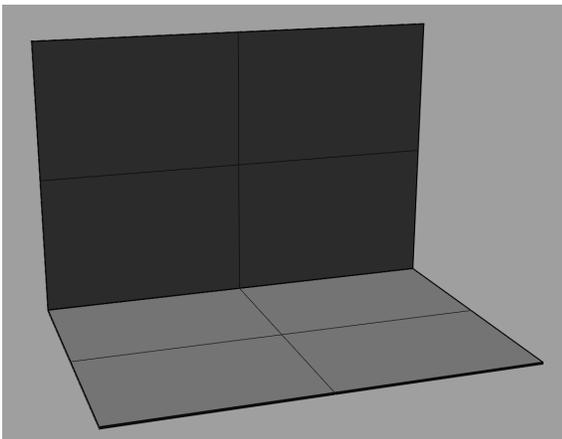


Figure 1: Base space.

- Calibration Object. We built a cubic calibration object of dimensions 30cm x 30cm x 30cm and we marked 17 points on its edges and faces, points that the calibration application will use as input information. (figure 2)

With this object placed at position (30, 0, 0) on the projection space, we measure the 3D positions of the marked points, and generate a file with these points positions in the real 3D world.

- Projector. The projector is placed hanging from the ceiling with the brackets included in the lab, and oriented in such a way that is able to illuminate both the

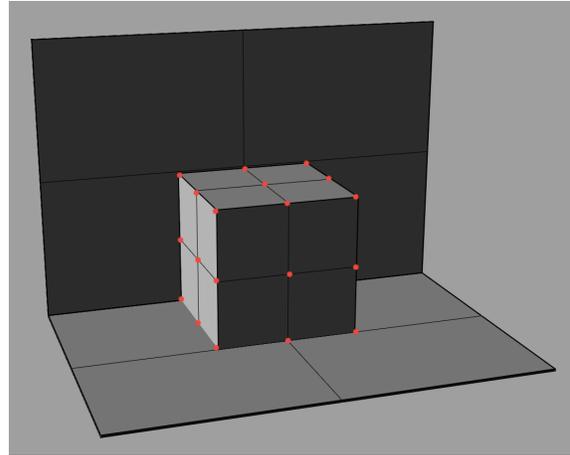


Figure 2: Calibration object.

space-based projection as the top face, left and front of the cube. (figure 3)

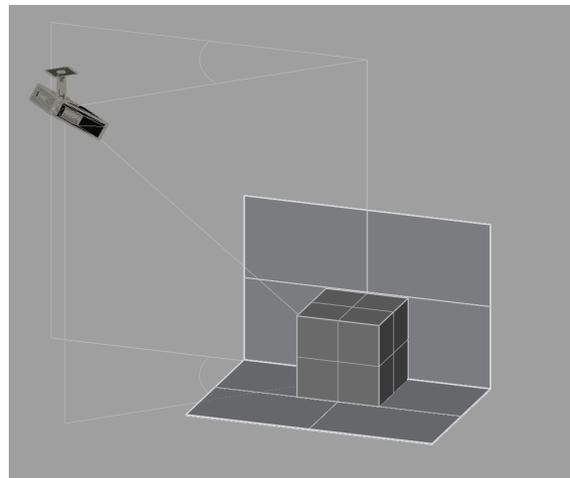


Figure 3: Projector placement.

- Computer. We connect the projector to the extra video output that has the computer where we run applications.

3.2 Applications

With this geometry information, a calibration application is able to calculate projective transformation, which will be used later to project the geometries transformed for the projection platform for this projector.

Rendering applications, will draw some textures on the cubic object, and will allows us to establish if the results of

calculations of the matrices are correct.

Both applications use the libraries openFrameworks.cc and mathematical GSL library for matrix calculations.

4 Calibration Application

Figure 4 describes the structure and components of the calibration program.

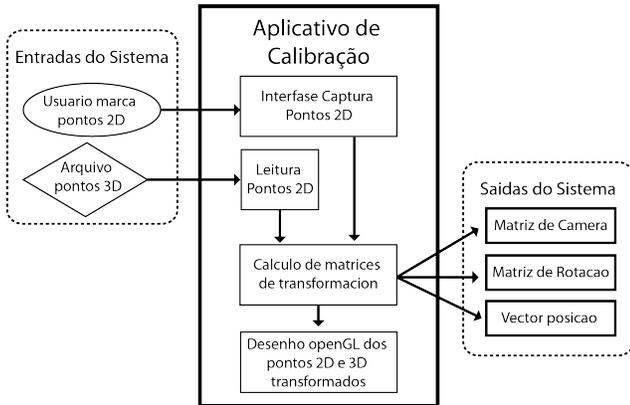


Figure 4: Architecture of the calibration program.

4.1 Input

Once we have placed the object on the base calibration space and we have decided and marked the points on the object being used to calibrate, we generate a file that contains the 3D positions of points marked on the object in the real world.

When running the application, it loads this point file and start a 3D projected user interface in the base space that allows the user to draw points on the geometry and move them to the positions of marked points on the object. (Figure 5). These two sets of points are those that the system uses as input to the calculations of the matrices.

4.2 Calibration

For the calculation of the calibration matrices we use an algorithm developed at VISGRAF Laboratory. This algorithm calculates the transformation matrix that will transform from the 3D computer scene space (virtual) to the real space of the base space. This algorithm was developed by Bruno Madeira and built into Windows XP. Using combinations of 2D and 3D points, the algorithm generates two matrices and a vector:

- Camera Transformation. Transformation matrix, which deforms the scene in the same way that makes the projector lens.

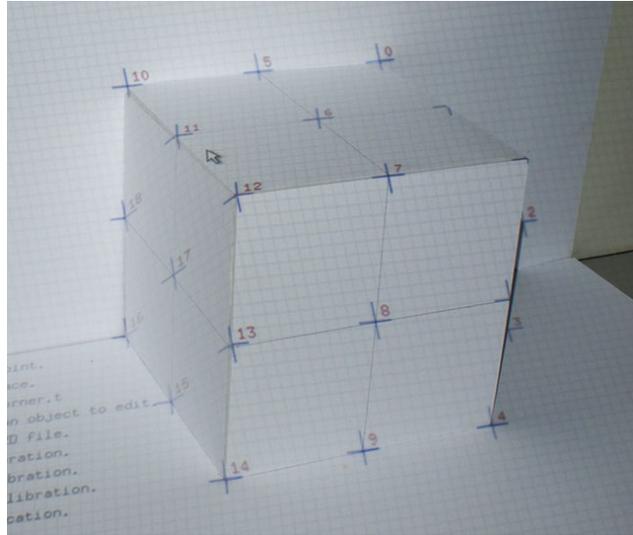


Figure 5: Interface for specifying point correspondence in the calibration program.

- Rotation. Matrix representing the rotation between the base space and projector.
- Position vector. Represents the vector of relative position between the base space and the projector.

4.3 Application output

The matrices generated by the application are stored on computer disk for further use.

After calculating the matrices, the application will sketch 2D points added by the user and also the 3D points transformed using the transformation matrices.

Mapping the two sets of points we have a proof of the correct projector calibration, as a consequence of the coincidence of the position of points 2D and 3D transformed.(Figure 6).

Repeating this process with various projectors, we can calculate transformation matrices for each of them, so we are able to illuminate objects from several different positions, which allows us to cover all areas of the objects.

5 Texture design application

This application is able to load any 3D scene in "3ds" file format and draw the objects transformed in the scene, for them to be designed correctly on the base space. For testing , the application also designs a light that rotates around the object, and illuminate it.

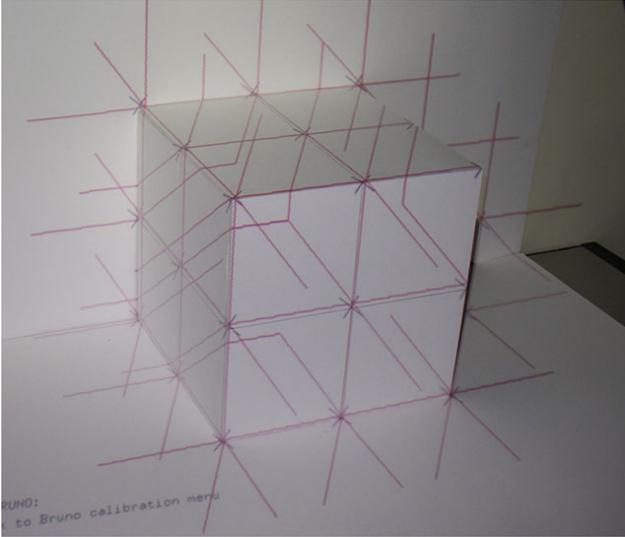
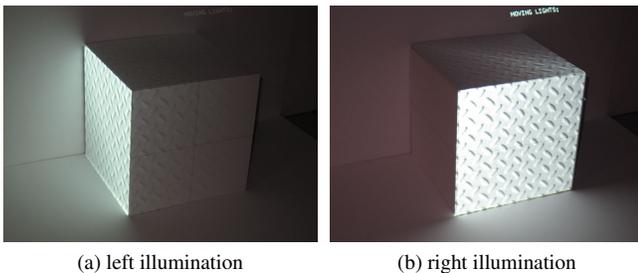


Figure 6: Proof of calibration.

6 Results

Generating different 3D scenes on the computer with cubes of different textures and projecting them on the object cubic calibration, we obtained the following results. See figures 7 8 e 9



(a) left illumination (b) right illumination

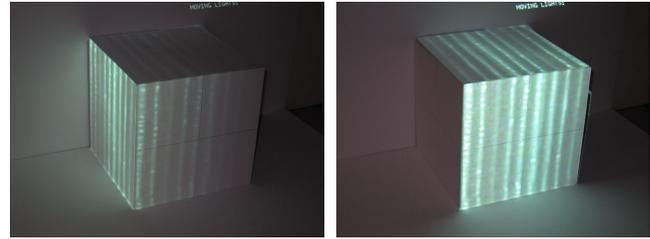
Figure 7: Aluminium texture

7 Ongoing work

We have set up the bases for the experimentation with an object and the projector calibrated for the space where the object is placed.

As previously mentioned, the projector lights only (textured) one side of the object.

To cover all sides of an object, it would be necessary to add more projectors (the number depending on the complexity of the object).



(a) left illumination (b) frontal illumination

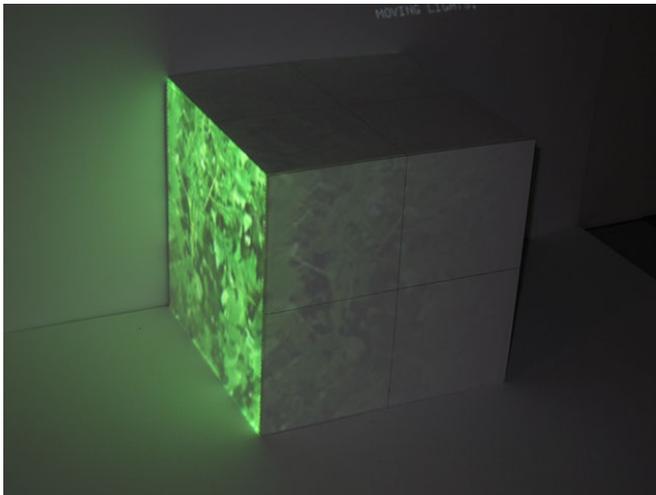
Figure 8: Semitransparent glass texture

We could also use cameras and other sensors to determine both the geometry of the object, how to detect the head position of the user, enabling us to add the specular component to render.

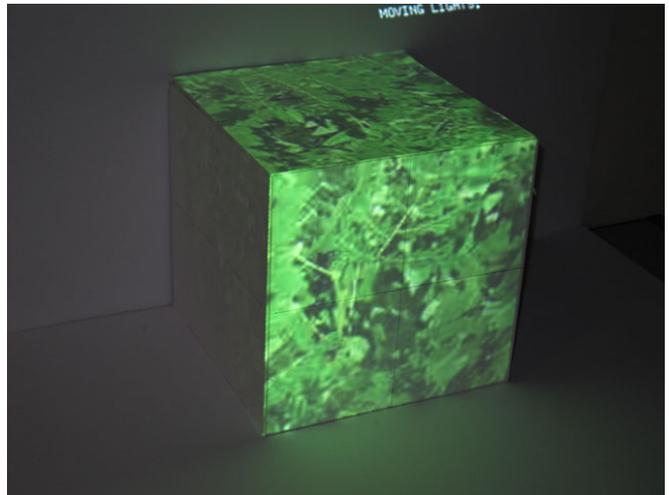
Though depending on the angle at which focuses the light from the projector in the geometry, the reflection is higher or lower for the same colour, we should do some adjustments in projector illumination, independently for each object face. In the case of several projectors, would have to also make a blending that would eliminate discontinuities in the light of a projector.

References

- [1] Dynamic shader lamps: Painting on movable objects. In *ISAR '01: Proceedings of the IEEE and ACM International Symposium on Augmented Reality (ISAR'01)*, page 207, Washington, DC, USA, 2001. IEEE Computer Society.
- [2] Pablo valbuena, spatial augmented reality, 2007. <http://www.pablovalbuena.com/>.
- [3] O. Bimber, F. Coriand, A. Kleppe, E. Bruns, S. Zollmann, and T. Langlotz. Superimposing pictorial artwork with projected imagery. In *SIGGRAPH '06: ACM SIGGRAPH 2006 Courses*, page 10, New York, NY, USA, 2006. ACM.
- [4] J. C. Lee, P. H. Dietz, D. Maynes-Aminzade, R. Raskar, and S. E. Hudson. Automatic projector calibration with embedded light sensors. In *UIST '04: Proceedings of the 17th annual ACM symposium on User interface software and technology*, pages 123–126, New York, NY, USA, 2004. ACM.
- [5] R. Raskar. Projectors: advanced graphics and vision techniques. In *SIGGRAPH '04: ACM SIGGRAPH 2004 Course Notes*, page 23, New York, NY, USA, 2004. ACM.
- [6] R. Raskar, K. Low, and G. Welch. Shader lamps: Animating real objects with image-based illumination. Technical report, Chapel Hill, NC, USA, 2000.
- [7] R. Raskar, G. Welch, M. Cutts, A. Lake, L. Stesin, and H. Fuchs. The office of the future: a unified approach to image-based modeling and spatially immersive displays. In *SIGGRAPH '98: Proceedings of the 25th annual conference on Computer graphics and interactive techniques*, pages 179–188, New York, NY, USA, 1998. ACM.



(a) left illumination



(b) frontal illumination

Figure 9: Plant texture

- [8] R. Raskar, R. Ziegler, and T. Willwacher. Cartoon dioramas in motion. In *NPAR '02: Proceedings of the 2nd international symposium on Non-photorealistic animation and rendering*, pages 7–ff, New York, NY, USA, 2002. ACM.