# Contribution to the study of visualization of real and virtual environments with an extended field-of-view

Jérôme Ardouin

Thèse

THÈSE INSA Rennes
*sous le sceau de l'Université Européenne de Bretagne*
pour obtenir le grade de
DOCTEUR DE L'INSA DE RENNES
*Spécialité : Informatique*


présentée par
Jérôme ARDOUIN
ÉCOLE DOCTORALE : MATISSE
LABORATOIRE : IRISA


# Contribution to the study of visualization of real and virtual environments with an extended field-of-view

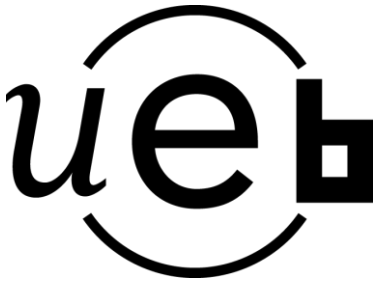
**Eric Marchand**
Professeur, Université Rennes 1 / *Co-encadrant, Membre invité*

**Thèse soutenue le 17 décembre 2014**
devant le jury composé de :

**Bruno Arnaldi**
Professeur, INSA de Rennes / *Président*
**Sabine Coquillart**
Directrice de Recherche, Inria Grenoble / *Rapporteur*
**Guillaume Moreau**
Professeur, Ecole Centrale de Nantes / *Rapporteur*
**George Drettakis**
Directeur de Recherche, Inria Sophia-Antipolis / *Examinateur*
**Maud Marchal**
Maître de Conférences, INSA de Rennes / *Co-encadrante*
**Anatole Lécuyer**
Directeur de Recherche, Inria Rennes / *Directeur de thèse*

# Contribution to the study of visualization of real and virtual environments with an extended field-of-view

Jérôme Ardouin



# Remerciements

Je tiens à remercier tout d'abord mes encadrants Dr. Anatole Lécuyer, Dr. Maud Marchal et Pr. Eric Marchand, qui m'ont guidé durant cette thèse, fait découvrir et apprécier le monde de la recherche.

Je remercie également les membres du jury qui ont accepté dévaluer mon travail. Je remercie ainsi Dr. Sabine Coquillart et Pr. Guillaume Moreau d'avoir accepté d'être rapporteurs de mon manuscrit, Dr. George Drettakis d'avoir accepté d'être examinateur, et Pr. Bruno Arnaldi pour avoir présidé le jury. Merci pour vos commentaires et pour vos conseils.

Je souhaite également remercier mes anciens collègues de l'Esiea, Franck, Marc, Nicolas, Emmanuelle, Aurélien..., qui ont tantôt joué les rôles de cobayes, assistants ou encore cameramen sur les différentes expériences. Je remercie aussi Clément Riant, pour son implication dans le développement du premier prototype de FlyVIZ.

Je remercie également tous ceux qui m'ont soutenu ces quatre dernières années, amis et famille. J'ai une pensée particulière pour les personnes qui m'étaient chères et qui ne sont plus là aujourd'hui. Je tiens à leur dédier ce travail.

# Contents

# List of Figures

# List of Tables

# Introduction

THIS doctoral thesis proposes to study the extension of the human field-of-view (FoV) in both real and virtual environments.

Species evolution has given birth to a wide panel of animals with different fields-of-view. Preys tend to get wider fields to locate potential predator earlier, with smaller head movements. Instead, predators have reduced field-of-view with high acuity, making them able to locate a prey from distance, by searching their surroundings [Sherstyuk 12]. The environment of life has turned eyesight into the major sense for almost all faunas. Thus, this has been a source of inspiration for a lot of science fiction authors imagining superpowers related to vision. Superman has a superhuman vision that grants him to see further than humanly possible, using telescopic or microscopic vision. Cyborg heroes are not in rest, the Six Million Dollar Man features a bionic eye with a 20.2:1 zoom lens and night vision. Nowadays, it is possible to augment someone's vision artificially using various instruments. For instance night vision is realized by goggles amplifying light or high zoom, increasing acuity, is realized by advanced optical instrument. These instruments have improved so much that it is even possible for mankind to watch the edge of the universe through telescopes. In the meantime extending the field-of-view did not benefit of much improvement: no optical device is able to override the natural human FoV.

Virtual reality (VR) has given the possibility to explore senses and interactions with environments in ways which overcome the physical limits of humans. Telekinesis is a common interaction technique allowing grabbing object in distances in virtual environments. Flying through virtual environments (VE) is a so common navigation technique that has become a standard for people comfortable with virtual reality. Many human's characteristics can be augmented through virtual reality, but it is not common to increase the field-of-view. Of course, several visualization devices claim to be "wide field-of-view" but these aim generally to cover the natural human FoV (about 200° horizontally), avoiding decreasing it. However they do not increase it. Therefore, one can ask: what would it feel like to extend our field-of-view, even up to gather visual information from behind us?

There are several motivations and potential applications in increasing the field-of-view. In healthcare for disabled people with neck injuries, the wide field-of-view would give them the perception of their environment without performing painful head's movements. For surveillance, a wide field-of-view would allow acquiring more visual information. In hazardous situations, an extended vision would allow localizing potential dangers faster. Firemen, policemen or soldiers are persons who could benefit from such extended vision. For research in neurosciences or psychology, a vision with increased FoV would be used as a new tool to study human vision, sensory motor control or locomotion. Finally, even entertainment could take advantage of an extended FoV, creating a new sensorial experience for new gameplay in video games or augmented reality games. Besides,

exploration and navigation of VE could benefit from a wide FoV. As more information is available instantaneously to the user, rotations of the user's point of view to scan the surroundings are not required anymore. This decrease the searching time of target and could increase overall performances when completing VR tasks.

Figure 1 describes the components and the architecture of systems required for generating a wide FoV vision. This figure presents the closed loops established when a user interacts with the real world or with virtual environments. In the case of VE (right part of the Figure 1), the user interacts with the surroundings by using interaction devices, then a new representation of the virtual environment is computed with a rendering algorithm, reflecting the changes made by the user (navigation, interaction with virtual objects, etc.). Finally, this new representation is proposed to the user through a display device. Noticeably, the whole process has to be performed fast enough to conform to real-time constraints.

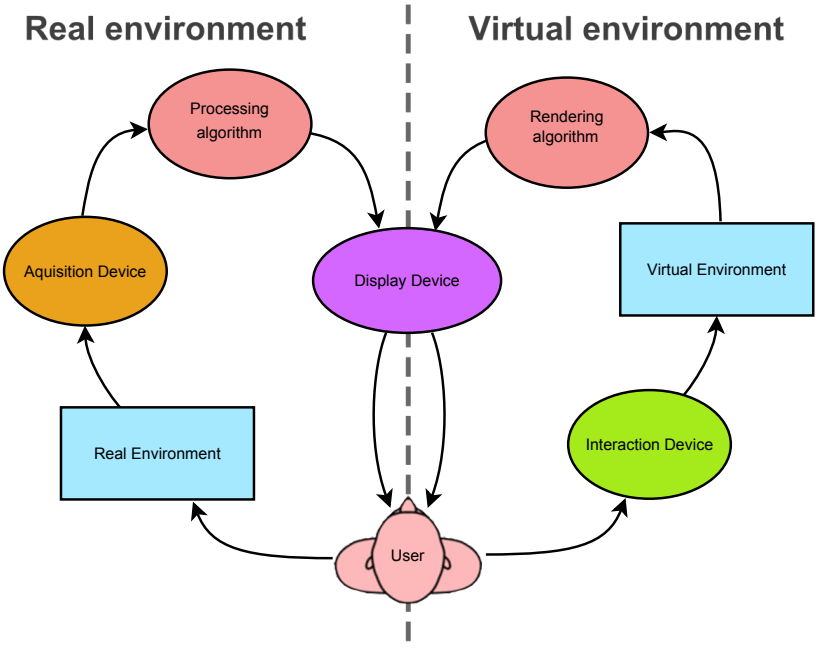



**Figure 1 –** Framework for the extension of user's FoV in real environments (left) or virtual environments (right).

The context of real environments (RE) is illustrated in the left part of the Figure 1. The approach is similar to augmented reality. The user interacts with the real environment; the visual information of the environment is then captured by an acquisition device like a camera. Subsequently, the acquired visual information is processed to generate the visual feedback. Finally, the result is presented to the user with a display device. As for the virtual environment counterpart, the process needs to comply with real-time requirements.

These two approaches present similarities and share a common theoretical and technical background. The challenges raised by extending the human field-of-view in VE and RE are discussed hereafter.

## Challenges in extending human field-of-view

Challenges to increase the human field-of-view are covered in three main topics: software, hardware considerations and human factors. Some challenges are cross context (not depending on specificities of real or virtual environments). They are illustrated in Figure 2.

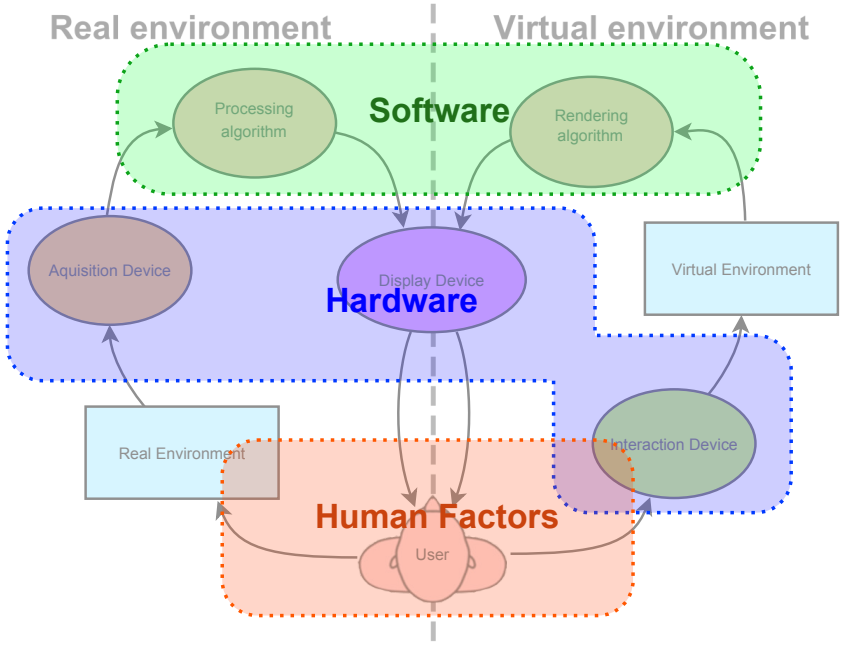



**Figure 2 –** Challenges involved by the extension of human FoV: hardware, software and human factors issues.

Software challenges mainly consist in the processing of data provided by an acquisition device (RE) or virtual models (VE) to generate a meaningful representation. This covers the modeling of wide FoV and omnidirectional cameras as well as algorithms needed to render VE. The fundamental mathematical tools related to this challenge are non-planar projections. In this context, a non-planar projection can be seen as a mathematical relation between a plane and a point of the 3D space. This issue has been studied for years and even centuries in the field of cartography. However numerous problems remain to be solved when they have to be used by algorithms to process images from panoramic sensors or to render omnidirectional views of VE in real-time.

Hardware challenges consist in the design of systems and devices needed to provide an extended FoV to somebody. Three sorts of devices are involved in the framework described above: acquisition device, display devices and interaction devices. A "wide FoV" or even omnidirectional acquisition device needs to capture visual information coming from very wide angles. In such conditions, conventional approaches do not suit: special devices need to be designed. These designs are very effective for driving robots in their environment. For that reason this is an active topic of robotic researches. Wide FoV display devices also need specific designs and rise problems related to the shape of the display surface, the technology to be used or the pixel bandwidth needed to maintain a suitable

image definition. Finally, the interaction devices used need a working area compatible with the wide FoV display used.

Human factors challenges consist in understanding how users perceive and interact in presence of a wide FoV. The human visual system is complex and the way information coming from the eyes is processed by the brain is still an active research topic of neurosciences. Like the other senses, the vision is a capacity developed since birth. Heavy modification of the FoV, one of main characteristics of vision, rise several questions: will the user be able to use a wider FoV than his natural FoV? Can he learn to use this extended vision? Is he able to take advantage of this new amount of visual information? Are there side effects in using such a vision? What are the limits of the shape/distortion/compression of the field-of-view the brain can integrate to interact with the real or virtual environment? To navigate through it?

In relation with these challenges, the contributions proposed in this manuscript are presented hereafter.

## Approach and contributions

The Figure 3 shows the different contributions of this thesis.

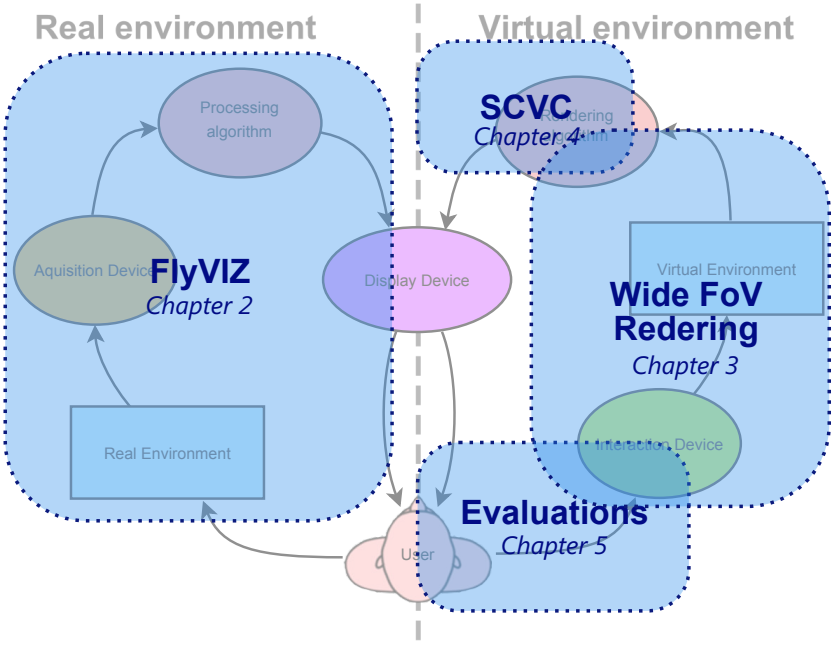



**Figure 3 –** Main contributions of our doctoral thesis

### First contribution: design of a device able to extend the human FoV in real environments

Considering the limitations of current optical or visualization devices available to augment the human vision, a first objective is to design a device able to increase the natural human field-of-view. The design of such device faces several major problems: pure optical approach does not seem realistic. Even with the actual state of the art technique in

lenses design, it seems difficult to build glasses or goggles able to collect, route and focalize correctly the light coming from behind a user. Of course such a device needs to be wearable. This imposes a strong constraint to the hardware. It has to be lightweight and highly integrated.

The device we have designed to reach this objective is called **FlyVIZ** in reference to the vision of flies, able to capture visual information coming from behind them.

### Second contribution: design of rendering methods to display virtual environments with wide FoV

Our second contribution concerns the extension of the FoV when visualizing virtual environments. As explained earlier in the section discussing the challenges, wide FoV rendering of VE needs to take advantage of non-planar projections. Current rendering algorithms (rasterization) and hardware (graphic processing units) are designed together in such a way that they are tightly linked to planar projection. This has made possible to optimize the computation times, a critical aspect to keep the algorithms real-time compatible. The mathematics equations involved in non-planar projection are more complex than the perspective projection used for decade. Therefore, using non-planar projection with current algorithms and graphic hardware is not straightforward. To handle this issue, we proposed a **method for real-time rendering of VE with a wide FoV**. The proposed method is also compatible with stereoscopic rendering.

Stereoscopy is another major feature for rendering of virtual environments. Stereoscopy, by providing different images to left and right eye, is able to augment significantly the depths perception and the feeling of immersion. Rendering wide FoV images in spectroscopy is raising several issues which are also faced in standard FoV rendering with perspective projection. Notably, perception problems like depth cue conflicts can occur. More specifically, a depth cue conflict appears when an object displayed in front of the screen and is partially occluded by the screen borders. We propose an innovative method to solve this problem, called **Stereo Compatible Volume Clipping** (SCVC) as it consists in discarding the rendering of geometry outside of a volume that is free of conflict (the Stereo Compatible Volume).

### Third contribution: human factors of wide field-of-view and preliminary evaluations in virtual environments

Numerous questions raises when considering human factors related to the extension of the field-of-view. How does an omnidirectional vision influence the user navigation in a virtual environment? Is the performance of the user for a specific task modified by the extension of the field-of-view? Does a particular non-planar projection suit to a specific scenario better? Our third contribution addresses the perception of wide FoV by **conducting an evaluation in the specific context of the navigation in virtual environments**. We have evaluated the user performance in a task of object collection. Classical perspective projection was confronted to several non-planar projection methods.

The manuscript is organized as follows: **Chapter 1** provides an overview of the related work in the extension of the FoV. The physiological aspects of the human field-of-view are described. The mathematical aspects cover the non-planar projection models involved

in the modeling of wide field-of-view images. For completeness, devices related to wide field-of-view are presented from both the acquisition and the display sides. Finally, algorithms and methods for rendering wide field-of-view image from virtual models are exposed.

**Chapter 2** presents the design of FlyVIZ, a display device we have created to increase the human field-of-view, providing up to 360° horizontal coverage. Hardware aspects are presented, as well as the software developed to process images coming from the acquisition device. Various scenarii of usage are described, placing the user in several different situations where he needs to interact with his environment, taking advantage of panoramic vision.

**Chapter 3** considers the rendering of wide FoV images, with the real-time and stereoscopic constraints. The chapter describes step by step the design of a new method to render wide FoV images in real-time. Our method is notably capable to render polygons spanning across a projection discontinuity. Stereoscopic rendering is also addressed. Concrete implementation is exposed afterwards, as well as the performance obtained.

**Chapter 4** proposes a new approach to solve the frame cancellation problem. Frame cancellation occurs due to conflict between two depth hints: the stereo disparity of a virtual object and the occlusion of this object by the border of the display surface. Such a problem can occur with the method proposed in Chapter 3, but it can occur more generally in all stereoscopic rendered images and is thus not limited to wide FoV context. The proposed approach is based on the clipping of the rendered geometry against the volume not subject to depth cues conflict. The method is presented and then, the technique is evaluated through a user study.

**Chapter 5** proposes an experimental evaluation of several non-planar projections suitable to render 360° images in real-time. The methods, inspired from cartography field, are described as well as their implementation. The evaluation procedure is described afterwards. It consists in a navigation task in a virtual environment where the user is asked to reach targets appearing in his neighborhood. Subjective evaluation of the users' preference is evaluated as well as the performance to complete the task. Results of these evaluations are given and discussed.

The manuscript ends with **conclusions and perspectives** of the work presented.

# Perception, acquisition and synthesis of wide field-of-view images

# 1

Contents



This chapter describes the related work in the field of the extension of the human field-of-view (FoV). In section 1.1, the physiological aspects of human FoV are addressed. Then in section 1.2 mathematical aspects of the projection models for wide FoV are described. The acquisition and display devices involved for wide FoV are then explored in the sections 1.3 and 1.4. Finally, rendering methods designed for wide FoV images of virtual environments (VE) are analyzed in section 1.5.

## 1.1 Short overview of the physiological aspects of human field-of-view

The field-of-view (or field of vision) is a well-known characteristic of the human visual system. A definition is proposed by Spector [Spector 90]:

*"The field of vision is that portion of space in which objects are visible at the same moment during steady fixation of gaze in one direction. The monocular visual field consists of central vision, which includes the inner 30 degrees of vision and central fixation, and the peripheral visual field, which extends 100 degrees laterally, 60 degrees medially, 60 degrees upward, and 75 degrees downward"*

The field-of-view can be measured using a campimeter (sometimes called perimeter), a tool that enables to establish a perimeter chart illustrating the whole field of vision of each eye (see Figure 1.1). To evaluate the field of vision of the patient, a target is displaced across the FoV while the patient is asked to keep looking at the center of the device. Ophthalmologists have interest in perimeter charts for a long time to diagnose diseases like glaucoma or scotoma, consequently these tools are largely discussed by authors like Fushs and Wright in manuals and textbooks [Fuchs 13][Wright 96].

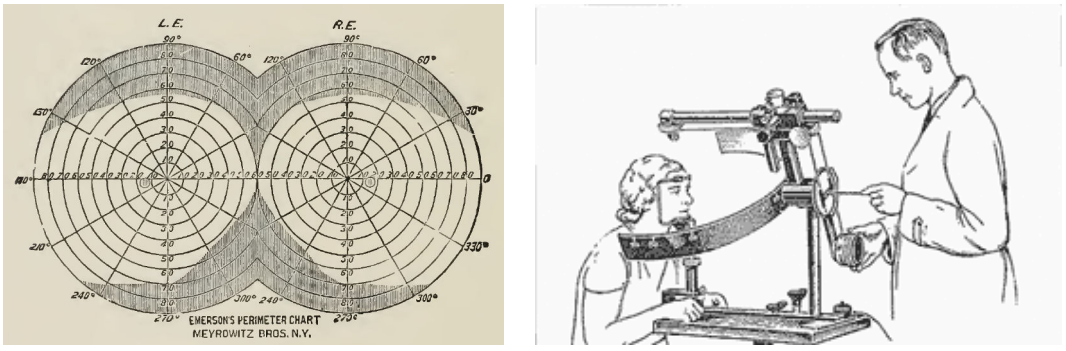



**Figure 1.1 –** Perimeter charts are used to figure the field-of-view covered by each eye. Left: chart for the two eyes, has described in ophthalmology book (1896). Right: Maggiore's perimeter. [Wright 96]

A measurement campaign of a significant sample of the worldwide population is missing. Therefore, statistically reliable data is not available but it is generally admitted that the visual FoV of a non-disabled person extends as described above. These values are considered to be accurate enough to drive engineering designs. For example, the US department of defense provides the measurements of Figure 1.2 as a standard for display design [ofDefense 99].

Across the FoV, several sub areas can be distinguished depending on their acuity. The ability to perceive the intensity of light comes from special photosensitive cells called rods and cones according to their shapes. Specificities of these two kinds of cells make differences in the way they respond to the light stimulation. Rods are very sensitive to only one, large, band of frequencies. Instead, cones have three types of response peaks,

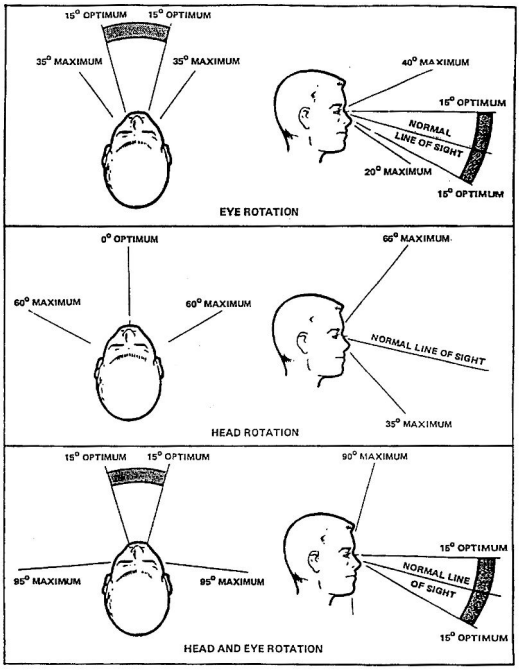



**Figure 1.2 –** Angular field-of-view covered by rotation of the head and the eyes [ofDefense 99]. From top to bottom: eye rotation coverage, head rotation coverage and field covered by the combination of the two. These value and figures are recommended as design guideline by US dept. of defense (design of control panel for example).

centered on red green and blue wavelength, making human eyes able to distinguish colors, but they are less sensitive than rod under low light intensity [Kandel 00].

The density variation of these photoreceptors across the retina leads to several sub-regions in the FoV where acuity varies (see figure 1.3). According to the density variation, three areas are distinguished [Osterberg 35]:

- **Peripheral area**: area with the lowest density of cones, populated with almost only rods and with low density. The vision in this area is efficient in low light and at detecting motion, but it is unable to distinguish colors.
- **Fovea**: area of highest density of cones making sharp vision possible. Human abilities such as reading rely on this part of the eyes.
- **Blind spot**: area where nerve fibers leave the eye. No cone or rods are exposed to the incoming light in this region. In everyday life, the brain "fills" the hole but experiences can highlight that this area is truly blind.

In everyday use, human cannot distinguish these areas precisely. The perceived images are colored in the peripheral vision (although there is no cone able to perceive colors in this area) and the blind spot does not impair the vision most of the time (the exception is experiments on vision that are designed to use the blind spot [Palmer 99a]). The partial visual information is processed and completed through complex mechanisms in the brain which are an active domain of neuroscience researches [Palmer 99b].

The concept of field-of-view looks straightforward; however it is a complex physiological characteristic of the human eye. The relative high field-of-view of human visual system needs to be balanced with the acuity drop in peripheral vision.

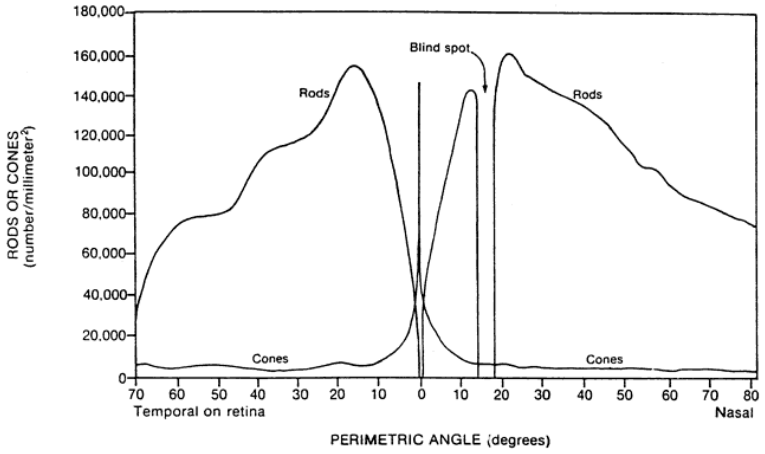



**Figure 1.3 –** Angular distribution of cones and rods [Osterberg 35]. The density variation determines three different areas where the acuity changes.

## 1.2 Projection models for wide field-of-view

Human vision, computer vision or image rendering are processes building a two dimensional representation of a three dimensional world, whether this world is the true world or some virtual environment. In the case of the vision, the optical system has the function to project the directions of the space to the retina (animals) or sensor array (camera). This is achieved thanks to the cornea and optical lenses. The mathematical tools used to model this process are projections. The state of the art in this topic has been highly influenced by cartographers who were facing the problem of visualizing the surface of the globe on a planar map. For ages, projections are used to map the earth, projecting points from the spherical surface of the earth to a planar one, easier to use, to draw or to carry on during sailboat travels [Snyder 87]. Depending on the use, the direct equation or its inverse function can be required. This aspect will be detailed deeper in section 1.5.

The projections of interest are those mapping $\mathbb{R}^3$ to $\mathbb{R}^2$, so a point $(X,Y,Z)$ in $\mathbb{R}^3$ is mapped to $(x,y)$ in $\mathbb{R}^2$.

$$\begin{array}{ccc} \mathbb{R}^3 & \xrightarrow{P} & \mathbb{R}^2 \\ (X,Y,Z) & \xrightarrow{P} & (x,y) \end{array} \tag{1.1}$$

In the following paragraphs the projections discussed are the perspective projection (planar projection), and several non-planar projections: the cylindrical equirectangular projection, the Hammer projection, the azimuthal equidistant projection and the Albers conic projection.

### 1.2.1 Perspective projection

A perspective projection, also known as a planar projection is a central projection on a plane. From mathematical point of view, the central projection onto the screen can be

described by equation (1.2).

$$\begin{cases} x &= f\frac{X}{Z} \\ y &= f\frac{Y}{Z} \end{cases} \tag{1.2}$$

where $f$, is the focal distance, i.e. the distance between the projection plane and the center of projection (see Figure 1.4). $f$ can be adjusted to cover the desired FoV, but $f$ has to be kept strictly positive. The theoretical maximum FoV ($h_{FoV}$) covered is:

$$h_{FoV} = 2\arctan\left(\frac{1}{f}\right) \tag{1.3}$$

$$\lim_{f\to 0} h_{FoV} = 180^\circ \tag{1.4}$$

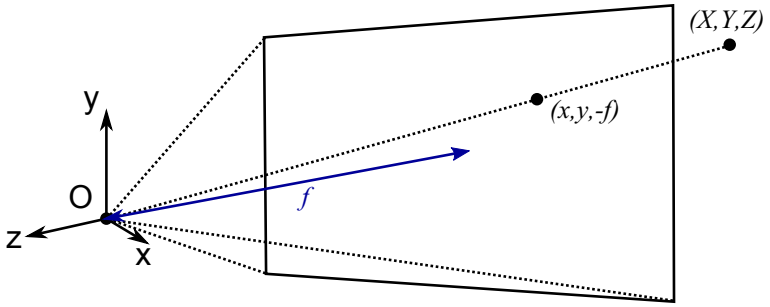



**Figure 1.4 –** Perspective projection. A point $(X,Y,Z)$ is projected to $(x,y,-f)$ on the plane, located at a distance $f$ from the center of projection $O$ in the $-z$ direction.

Computing the horizontal FoV ($h_{FoV}$) from the vertical FoV ($v_{FoV}$) is not direct (the vertical FoV cannot by directly multiplied by the aspect ratio). For example, with an image having $r$ for aspect ratio, the following formula gives the horizontal FoV available:

$$h_{FoV} = 2\arctan\left(r\tan\left(\frac{v_{FoV}}{2}\right)\right) \tag{1.5}$$

This formulation is suitable when, depending on the aspect ratio of the targeted display, the horizontal FoV is modified while the vertical FoV remains fixed. For example, if the vertical FoV is set to 75° (equal to the upper limit found in video games) and the aspect ratio to 16:10 the computed horizontal FoV is about 102°. If the monitor used has 16:9 ratio the horizontal FoV would be around 107°.

### 1.2.2 Non-planar projections

Perspective projection is limited in the amount of FoV it can cover. To alleviate this limitation, non-planar projections need to be used.

The mathematical formulation of the problem consists in mapping all the space directions onto a plane. Mapping one direction of 3D space is equivalent to mapping a point on a unit sphere onto this plane (due to the unique representation of a direction vector in 3D space and of the points lying on the unit sphere). This mapping problem has been widely studied by mathematicians and cartographers [Snyder 97]. In the cartography literature, these projection equations are expressed as functions which map a point from

the unit sphere to one point onto the plane. These points are usually not expressed as a cartesian vector but as spherical coordinates $(\lambda, \phi)$, the third coordinate $\rho$ being ignored (constant and equals to the earth radius). Equations (1.6) and (1.7) establish the relationship between the Cartesian coordinate $(X,Y,Z)$ of any direction in space, the Cartesian coordinate of this direction after normalization $(X',Y',Z')$ and the associated spherical coordinates $(\lambda, \phi)$. Figure 1.5 illustrates these correspondences.

$$\begin{cases} \rho &= \sqrt{X^2+Y^2+Z^2} \\ \lambda &= \arctan\frac{X}{-Z} \\ \phi &= \arcsin\frac{Y}{\rho} \end{cases} \tag{1.6}$$

$$\begin{cases} X' &= \sin\phi\cos\lambda \\ Y' &= \sin\lambda \\ Z' &= -\cos\phi\cos\lambda \end{cases} \tag{1.7}$$

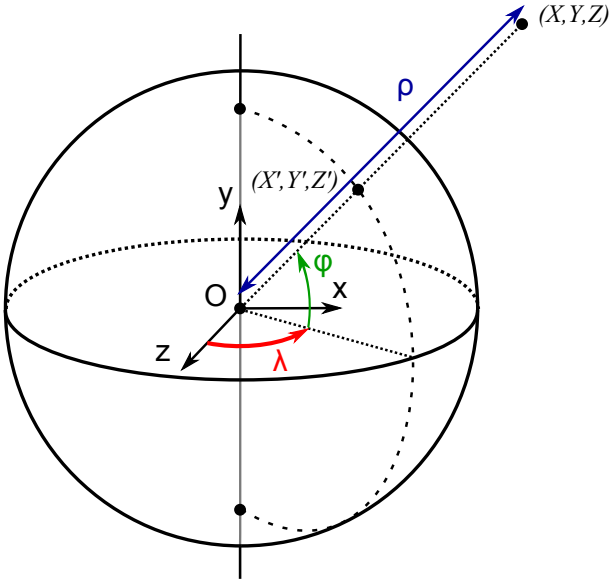



**Figure 1.5 –** Unit sphere and correspondence between coordinates systems.

A way to classify non-planar projections is to sort them based on their primary unfoldable shape. If a sphere is considered as a starting point, a non-planar projection will require an intermediary 3D surface to project the visual information. Candidate for this shape is an unfordable shape, which can then be mapped to the plane. The common unfoldable shapes are:

- Plane, leading to azimuthal projections family
- Cone, leading to conical projections family, and
- Cylinder, leading to cylindrical projections family.

Furthermore, the non-planar projections can be classified according to different geometrical properties. These properties are significant as they can influence the user's

perception of the projected medium. Depending on the final usage of the map, one can target [Snyder 87]:

- Conformal projection, that preserves angle locally,
- Equidistant projection, that preserves distances along the mains axis, or
- Equal Area projection, preserving locally areas.

When used in algorithms, the inverse formulation of the transformation may be required. This is notably the case for image-based rendering methods (see section 1.5). Also, depending on the projection considered, the inverse computation can be straightforward, complex, and it even may have no explicit form.

Sometimes, multiple centers of projection are considered [Rademacher 98]. Concretely, all the projection parameters can vary along the image: center of projection, position or orientation of the projection plane, etc. As the method associates only one projection per pixel, it suits particularly well image-based rendering (see section 1.5). The final rendering can be seen as a composite of multiple images rendered with several single center projections.

The projection methods discussed hereafter are able to provide a full lateral 360° FoV. They are suitable to provide the maximum visual information, at least in the lateral direction. If required, these projection methods can be tuned to match a 180° field-of-view, which would match the physiological characteristics of human vision, or any other value.

#### 1.2.2.1 Cylindrical equirectangular projection

From a geometric point of view (see Figure 1.6), a cylindrical projection first projects a point $S$ from the unit sphere onto a cylinder (point $P$). In the equirectangular case, the cylinder has an infinite height. The cylinder is then unfolded and stretched to a 2:1 rectangle.

The equirectangular projection (Figure 1.7) is one of the most widespread projections, used for instance in panoramic photography. In the cartography field, it is appreciated for its clarity and simplicity. For full representation of surroundings, it leads to a rectangle of 2:1 aspect ratio that fits well modern screens with 16:10 or 16:9 aspect ratio (see Figure 1.7).

It has the equidistant property: distances can be measured from the equator. Mathematical formulation is very simple, spherical coordinates are interpreted as Cartesian coordinates multiplied by a scale factor:

$$\begin{cases} x &= \frac{2}{h_{FoV}}\lambda \\ y &= \frac{2}{v_{FoV}}\phi \end{cases} \tag{1.8}$$

$h_{FoV}$ and $h_{FoV}$ (horizontal and vertical field-of-view) allow to restrict the coverage of the projection. By taking $h_{FoV} = 2\pi$ and $v_{FoV} = pi$, the full field-of-view is covered. The reciprocal formulation is straightforward:

$$\begin{cases} \lambda &= \frac{h_{FoV}}{2}x \\ \phi &= \frac{v_{FoV}}{2}y \end{cases} \tag{1.9}$$

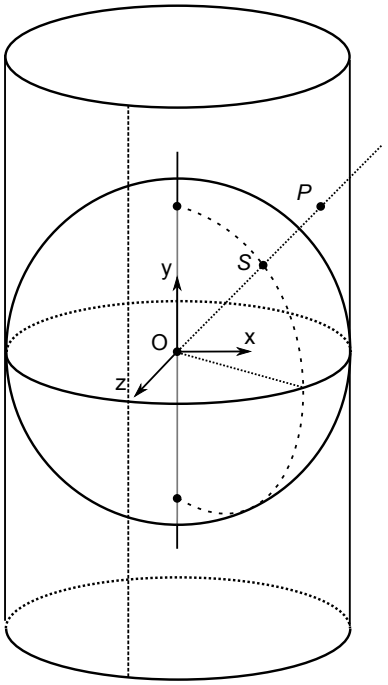



**Figure 1.6 –** Cylindrical projection uses a cylinder as a primary unfoldable shape. A point $S$ on the spherical surface is projected to $P$ on the cylinder. The cylinder is then unfolded and stretched to a 2:1 rectangle.

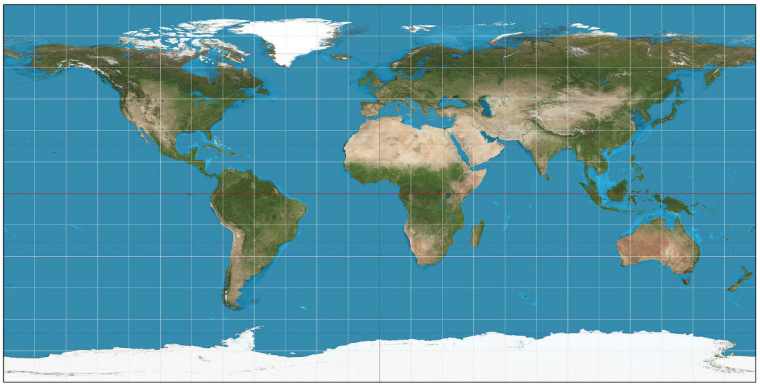


**Figure 1.7 –** Equirectangular projection.

#### 1.2.2.2 Hammer projection

The Hammer projection (Figure 1.8) is a cylindrical projection (see Figure 1.6) which maps the result to an ellipse with 2:1 proportion. This additional deformation is chosen to give the equal area properties to the projection. Compared to the equirectangular projection, it is therefore less subject to area stretching.

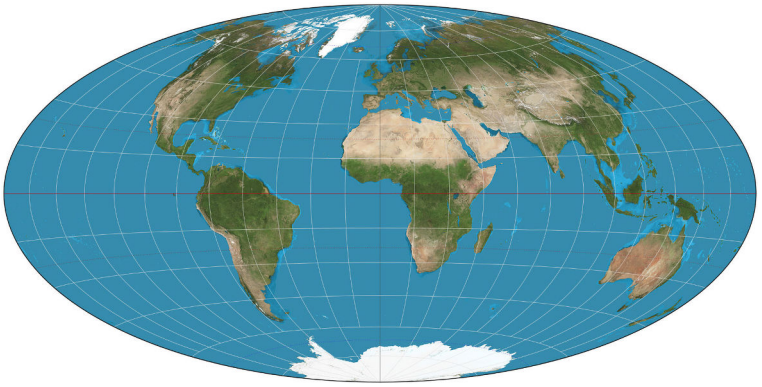


**Figure 1.8 –** Hammer projection.

The projection equation (1.8) is modified into equation (1.10), leading to an ellipse shaped view:

$$\begin{cases} x &= \frac{2}{h_{FoV}} \frac{2\sqrt{2}\cos\phi\sin(\lambda/2)}{\sqrt{1+\cos\phi\cos(\lambda/2)}} \\ y &= \frac{2}{v_{FoV}} \frac{\sqrt{2}\sin\phi}{\sqrt{1+\cos\phi\cos(\lambda/2)}} \end{cases} \tag{1.10}$$

The reciprocal computation is given by:

$$\begin{cases} \lambda &= 2\arctan\frac{zx}{2(2z^2-1)} \\ \phi &= \arcsin(yz) \end{cases} \tag{1.11}$$

with

$$z = \sqrt{1-\left(\frac{1}{4}x\right)^2-\left(\frac{1}{2}y\right)^2}$$

#### 1.2.2.3 Azimuthal equidistant projection

Azimuthal projection family encompasses the projections that directly map point on the unit sphere to a plane. To build such a mapping, the intermediate projection can be perspective. This is the case for orthographic projection, stereographic projection (no to be confused with stereoscopic images or rendering which build depth perception by proposing different images for left and right eyes) or gnomonic projection. Figure 1.9 illustrates from a geometric point of view how these projections are built.

Unfortunately, these projections are not able to cover the full earth (or the full space directions). To build azimuthal projections able to cover a wider field, the intermediate perspective projection has to be replaced by another mapping. Among them, the azimuthal equidistant projection is remarkable as it can be found in real world (Figure 1.10). Image formation for a type of fisheye lenses has this mapping function [Kingslake 89]. The

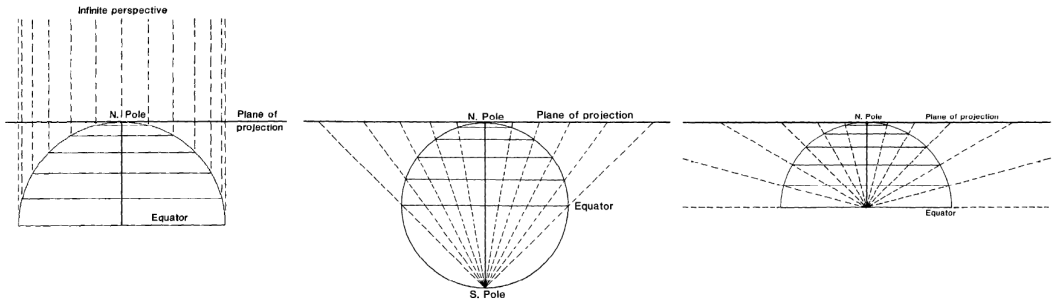



**Figure 1.9 –** Azimuthal perspective projections. Depending on where the center of projection is placed, the final built projection is orthographic (left), stereographic (middle) or gnomonic (right) [Snyder 87].

projection maintains a constant radial scale from the central point. The resulting images are free of distortions at the central point, but a large deformation occurs at the borders.

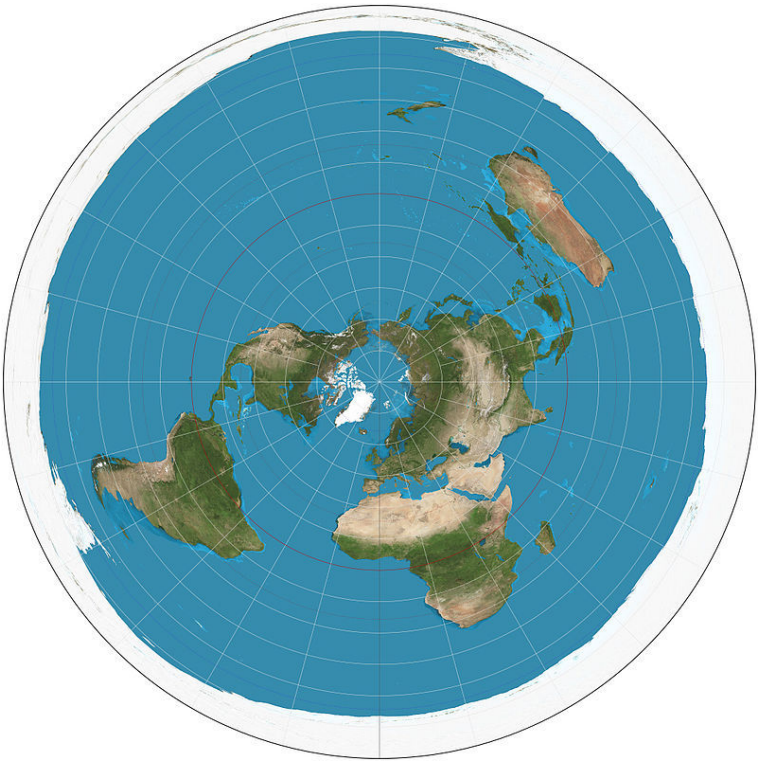


**Figure 1.10 –** Azimuthal equisdistant projection.

Direct mapping is computed as follow:

$$\begin{aligned} x &= k' \cos\phi \sin(\lambda - \lambda_0) \\ y &= k' \left(\cos\phi_1 \sin\phi - \sin\phi_1 \cos\phi \cos(\lambda - \lambda_0)\right) \end{aligned} \tag{1.12}$$

with

$$\begin{aligned} \cos c &= \sin\phi_1 \sin\phi + \cos\phi_1 \cos\phi \cos(\lambda - \lambda_0) \\ k' &= \frac{c}{\sin c} \end{aligned} \tag{1.13}$$

Geometrical mean of $c$ is the angular distance from the center of the projection defined by

$\phi_1$ and $\lambda_0$. Reciprocal form is the following:

$$\begin{aligned} \phi &= \arcsin\left(\cos c \sin\phi_1 + \frac{y \sin c \cos\phi_1}{\rho}\right) \\ \lambda &= \lambda_0 + \arctan\left(\frac{x \sin c}{\rho \cos\phi_1 \cos c - y \sin\phi_1 \sin c}\right) \end{aligned} \tag{1.14}$$

with

$$\begin{aligned} \rho &= \sqrt{x^2 + y^2} \\ c &= \lambda_0 + \arctan\left(\frac{x \sin c}{\rho \cos\phi_1 \cos c - y \sin\phi_1 \sin c}\right) \end{aligned} \tag{1.15}$$

#### 1.2.2.4 Albers conic projection

Conical projections use a cone as an intermediate unfoldable shape. In Figure 1.11, a point $S$ on the unit sphere projects onto point $P$ on the cone. The surface of the cone can then be unfolded to a plane to obtain the final view. The intermediate cone can intersect the sphere or it can be tangent to it. These define one (tangent case) or two (secant case) special parallels on the sphere, which control the final projection. These parallels are the standards parallels.

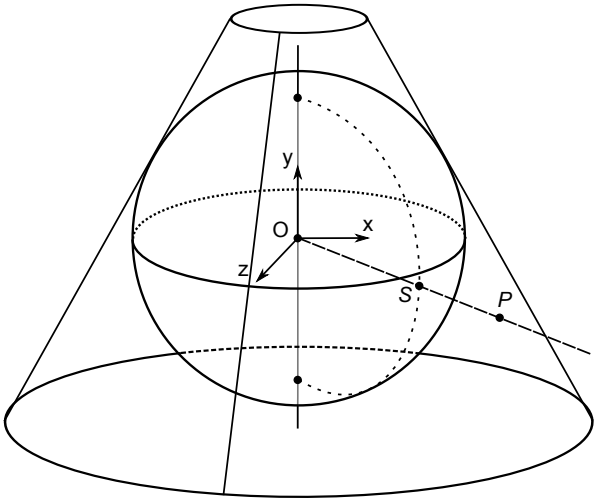



**Figure 1.11 –** Conical projection. Projections of this group use a cone as a primary unfoldable shape. A point $S$ on the spherical surface projected to $P$ on the cone

Among the different conic projection methods, the Albers Conic projection (Figure 1.12) has the property to preserve area. Shapes between the two parameterized parallels suffer low distortion. It can be noticed that it is not possible to represent the full surroundings as the projection of one of the pole necessarily fails. However, a full 360° horizontal FoV can still be obtained.
The projection is formulated as follow:

$$\begin{cases} x &= \rho \sin\theta \\ y &= \rho_0 - \rho \cos\theta \end{cases} \tag{1.16}$$

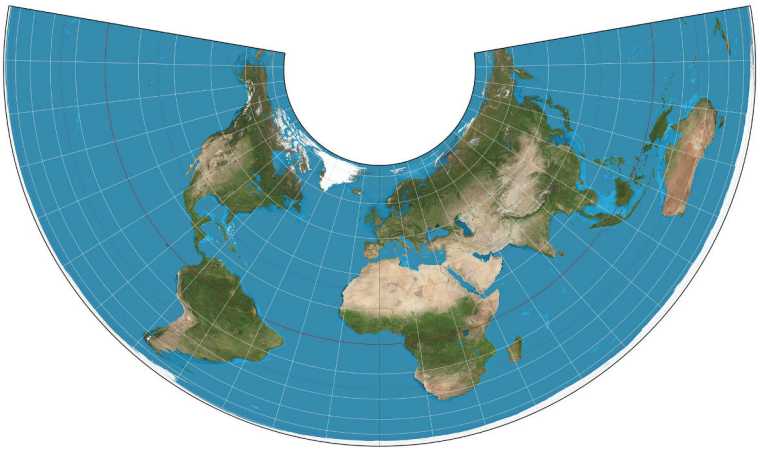


**Figure 1.12 –** Albers conic projection.

with the following parameters:

$$\begin{array}{rcl} n & = & \frac{1}{2}(\sin\phi_1 + \sin\phi_2) \\ \theta & = & n(\lambda - \lambda_0) \\ C & = & \cos^2\phi_1 + 2n\sin\phi_1 \\ \rho & = & \frac{\sqrt{C-2n\sin\phi}}{n} \\ \rho_0 & = & \frac{\sqrt{C-2n\sin\phi_0}}{n} \end{array} \tag{1.17}$$

Where $\phi_1$ and $\phi_2$ are the standards parallels, and $\lambda_0$ and $\phi_0$ the origin of the projection. The reciprocal is given by:

$$\left\{ \begin{array}{rcl} \phi & = & \arcsin\left(\frac{C-\rho^2 n^2}{2n}\right) \\ \lambda & = & \lambda_0 + \frac{\theta}{n} \end{array} \right. \tag{1.18}$$

with

$$\begin{array}{rcl} \rho & = & \sqrt{x^2 + (\rho_0 - y)^2} \\ \theta & = & \arctan\left(\frac{x}{\rho_0 - y}\right) \end{array} \tag{1.19}$$

With the conical projections, the principal families of projections used by cartography are covered. In the next paragraphs, discussing hardware aspects of wide FoV, the link between mathematical equations and real world will be established with the help of models and calibration procedure.

## 1.3 Acquisition devices for wide field-of-view

The capture of real world images with a wide field-of-view requires the design of specific hardware. Such devices have been developed by opticians and researchers motivated by scientific fields such as robotics. We propose a classification for these devices, where four kinds of approaches can be distinguished:

- Hardware using special lenses, often called "fisheye" or "fullsky" lenses [Kingslake 89];

- Hardware combining mirror and lenses, called catadioptric devices [Nayar 97];
- Hardware relying on an array of classical cameras, recombining the images from each sensor to build the final view;
- Hardware relying on a hybrid approach, array of camera used with planar mirrors [Nalwa 96].

A fifth, very trivial, approach can be mentioned. It consists in rotating a camera around a pod, images being acquired sequentially in time. Then they are used to build the final view. As the method relies on images acquired sequentially in time, there is no way to get a video stream from such an approach. From a theoretical point of view, the method is also very similar to camera array approach, where images from several cameras are combined to obtain the final, wide FoV image.

The four main approaches to capture images with a wide field-of-view are described in the following sections.

### 1.3.1 Fisheye lenses

Initially called full sky lenses, fisheye lenses were developed by scientist to study the climate by taking pictures of clouds along the whole sky. They are generally costly to produce due to the significant number of sub lenses they include [Kingslake 89]. Depending on the purpose of the lens, the mapping obtained can have different properties: equidistant or conformal for example. As an example of the performance reached by this lens design, the Nikon fisheye 6mm lens is capable of 220° of field-of-view (see figure 1.13). It is built with 12 sub lenses assembled in a final package which weighs 5.2kg.

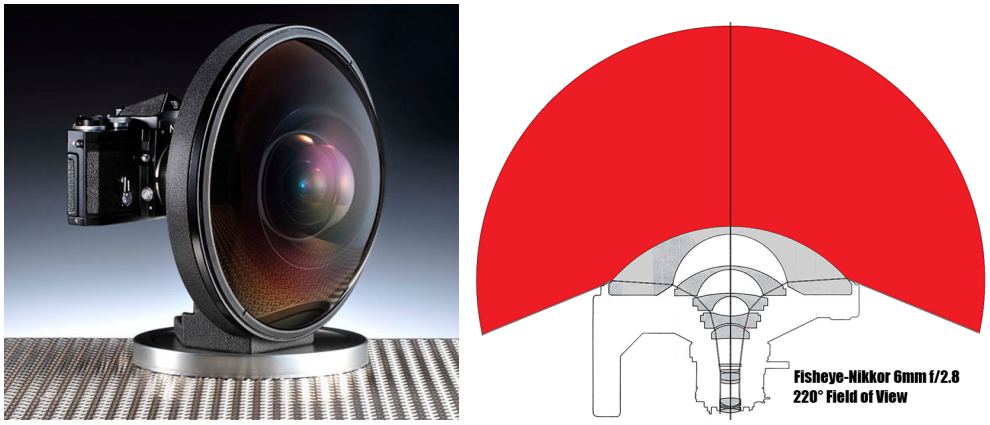



**Figure 1.13 –** Fisheye lens. The model shown is a Nikon 6mm fisheye lens. It weighs 5.2kg and captures field-of-view of 220°. The schematic on the left shows the lens arrangement used to collect light ray coming from behind the lens

The direct limitation of these optical designs is the distortion of the final image. This leads to non-uniform angular definition (the borders have far less pixels per degree than the center). Also the final image can suffer optical specific problems like chromatic aberrations and internal reflections. Lenses use the refraction effect to collect the light. Due to the refraction law, the lens refracts the light with different angle according to its wavelength. This can be problematic for light coming with high angle of incidence. In imaging

systems, it leads to artifacts known as chromatic aberrations: blue and red components of the image are not superposed anymore on the borders of the image. To correct chromatic aberrations, the inner lenses need complex arrangements, 2 or 3 concave and convex elements glued together. To correct inner reflections, the solution is to use special treatments like coating. Unfortunately, this tends to increase even more the overall price of the lens. Simple single lens designs, like the low cost Dot from Kogeto on Figure 1.14, are subject to these problems and give poor image quality.


**Figure 1.14 –** Kogeto Dot. This fisheye lens is a mass market product. It makes possible to capture 360° views with a smartphone.

### 1.3.2 Catadioptric lenses

A catadioptric lens consists of an assembly of both a mirror and a traditional lens. The mirror can take various shapes. Optical systems are successfully built with spherical, conical or parabolic mirror for example [Nayar 97] (see Figure 1.15).

**Figure 1.15 –** Various mirrors suitable to build catadioptric sensors. The picture does not show the additional lens needed to build a functional optical system.

Depending on the shape, there are some restrictions on the lens that can be used. Some combinations may result in an optical system that is not able to focus (due to multiple light paths) and that is not usable for imaging. Typically, when using a conical or a spherical shape for the mirror, the lens needs to provide an orthographic projection. Such lenses are called telecentric.

A catadioptric lens uses the shape of its mirror to collect the light coming from high wide incident angles. This prevents several of problems encountered with fisheye lenses design, such as chromatic aberrations or inner reflections. As for fisheye lenses, low cost versions of catadioptric lenses are available to get wide angle picture and video with smartphones (Figure 1.16).


**Figure 1.16 –** EyeSee360 GoPano. The GoPano is a mass market products based on a catadioptric sensor. It makes possible to capture 360° views with a smartphone.

### 1.3.3 Camera arrays

Camera array is a solution to maintain high resolution on all the field-of-view of the system. The wide FoV is obtained by merging the images coming from several standard cameras. Mechanical constraints lead to optical misalignments between each camera and so the different units composing the array do not share a unique center of projection. In this situation, image with object captured at low distances can fail to blend, and visual artifacts can occur. To be operational, the array needs to be precisely calibrated. That includes the parameters describing each lens (distortions, optical center etc., known as intrinsic parameters of the lens) and how each camera is positioned relatively to others. Another drawback of this kind of approaches is the bandwidth it requires, saturating easily several data transmission technologies. For example, the Ladybug 5 from Point Grey (Figure 1.17.b), requires $6 \times 2048 \times 2448 \times 10 \times 12 \approx 3.6$Gb/s (6 sensors with $2048 \times 2448$ pixels at 10 frames per second with 12 bits per pixel) at full sensor resolution. A last critical feature is the requirement of the synchronization of the sensor composing the array to avoid artifacts during image blending.

Figure 1.17 presents a panel of devices which target different markets. Immersive Media's Dodeca aims at capturing 360° movies (Figure 1.17.a). Point Grey's Ladybug 5 target the robotic market and computer vision sector (Figure 1.17.b) and the Panono ball is crowd-founded project aiming at developing a mass market device able to capture 360° picture by throwing the device in the air (Figure 1.17.c).

(a) (b) (c)

**Figure 1.17 –** Various sensors based on camera array. (a) Immersive Media Dodeca, (b) Point Grey Ladybug 5 and (c) Panono GmbH Panonoball.

### 1.3.4 Hybrid systems: camera array and planar mirrors

To overcome optical problems of the previous approaches, Nalva et al. have developed the first hybrid devices [Nalwa 96]. Their motivation for the hardware design was to minimize the distance between the centers of projection of each camera composing the array. To do so, planar mirrors are used in combination of classical lenses (see Figure 1.18).

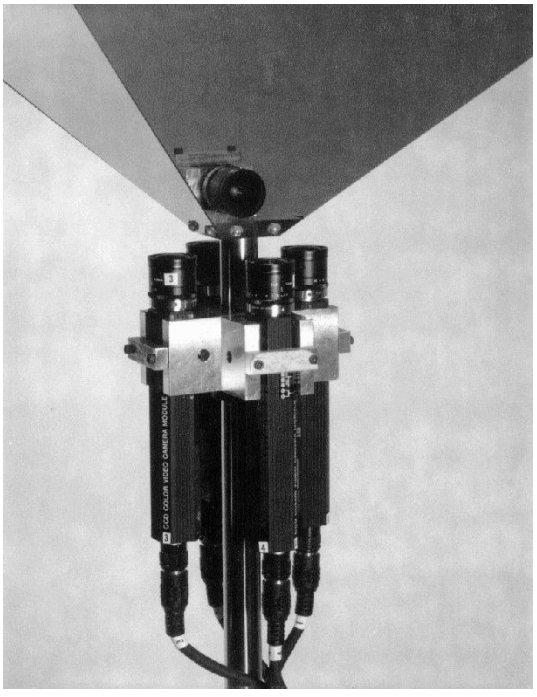


**Figure 1.18 –** Hybrid omnidirectional device. The design associates a camera array with mirrors to maintain all centers of projection at the same location [Nalwa 96].

With calibration, the whole system can provide panoramic images free of blending artifacts. At this time, devices available on the market are successfully used in applications like sports or video surveillance systems.

### 1.3.5 Modeling and calibration of wide field-of-view acquisition devices

Modeling and calibration make possible to establish a relationship between an image acquired from a device and a direction in space. Generic models have to be used and for a chosen device, characterization of the parameters of this model has to be performed through a calibration procedure. Here, classical models and calibration algorithms relying on pinhole camera model cannot be employed; even if a two dimensional polynomial is employed to model the distortions, as in [Zhang 00] or [Hartley 03]. Instead, a model such as the ones proposed by Barreto et al. or Nayar et al. needs to be used [Barreto 01] [Nayar 97]. This model is illustrated in Figure 1.19.

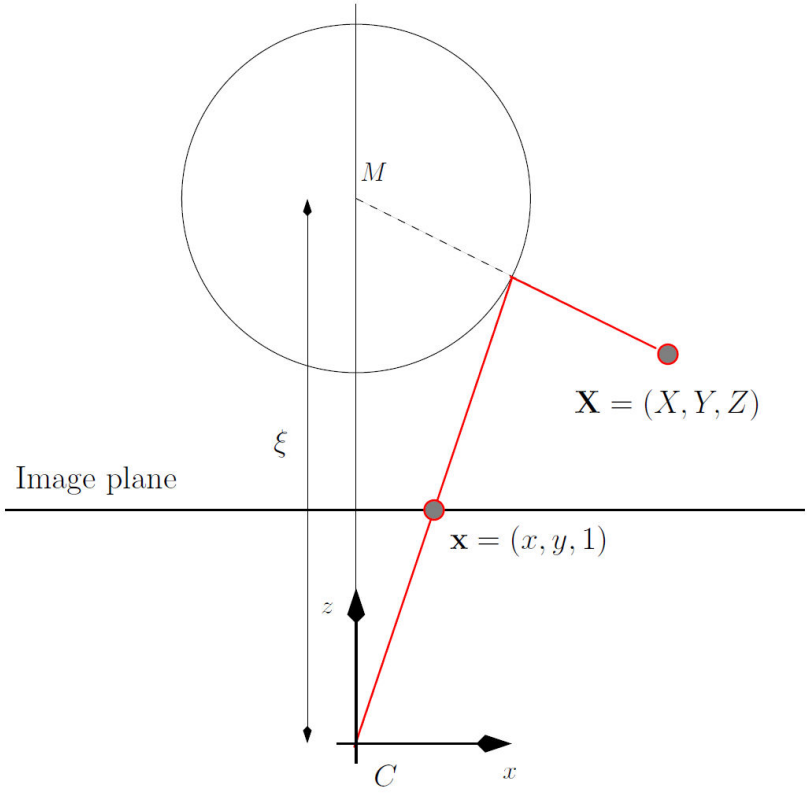



**Figure 1.19 –** Central catadioptric vision system [Marchand 07].

A point $\mathbf{X}$ with coordinates $(X,Y,Z)$ in space is mapped to the point $\mathbf{x}$ of coordinates $(x,y,1)$ in image plane. In the model, the point $\mathbf{X}$ is first projected on a sphere centered in $(0,0,\xi)$. Then a perspective projection is applied to finally map $\mathbf{X}$ to $\mathbf{x}$. The model is defined using the parameter $\xi$ which depends of the mirror parameter used for the catadioptric camera. The formulation gives:

$$\begin{aligned} x &= \frac{X}{Z+\xi\sqrt{X^2+Y^2+Z^2}} \\ y &= \frac{Y}{Z+\xi\sqrt{X^2+Y^2+Z^2}} \end{aligned} \tag{1.20}$$

Historically, research in robotic has pushed forward this topic. For practical use, software tools are available providing a solution to easily calibrate (identify the parameters of

the model) the camera. Scaramuzza proposes a matlab toolbox dedicated to the calibration of omnidirectional camera [Scaramuzza 06b]. HYSCAS is an alternative that is also capable to calibrate rigs composed of classical camera mixed with omnidirectional (fish-eye or catadioptric) camera [Caron 11]. These tools operate as follows: first they gather images of a known pattern (often a chessboard). Then for each image, an estimate of the position of the pattern is performed using an arbitrary initial estimation of the camera parameters. At this time correspondence between 3D points and their image is obtained. The parameters of the camera model are then estimated by using linear algebra methods, in order to minimize the error between the 3D points projected by the model and their real images. The algorithm can then loop to obtain a better guess.

As described above, hardware to capture wide FoV images can adopt several different designs. The next section covers the counterpart of acquisition devices: the wide FoV display devices.

## 1.4 Display devices providing wide field-of-view

There are several kinds of display device that can be qualified of "wide field-of-view". Two main approaches can be distinguished, based on the mechanical link between the user and the display. The first approach is related to head mounted displays: the display surface is mechanically linked to the head of the user. The second approach is employed by surround-screen displays: a room is set with one or several display surfaces surrounding the user. Among the features of the different devices, the amount of field-of-view that is available instantaneously to the user is to be distinguished from the amount of FoV provided by the device if the user rotates his head to scan his surroundings. This is especially significant for head mounted display as seen hereafter.

### 1.4.1 Head-mounted displays

A Head-Mounted Display (HMD) is a display device, worn on the head or as part of a helmet, which has one (monocular HMD) or two (binocular HMD) small display(s) in front of user's eyes. The typical horizontal FoV of research or commercial HMDs ranges from 24° to 187°. Usually, increasing FoV results in an augmentation of the sense of presence [Lin 02], so efforts are regularly made to extend the FoV of HMDs. As an example, Sensic company notably proposed an approach based on a mosaic of small screens (see Figure 1.20). Therefore, the device is able to extend the FoV while maintaining good resolution at the same time. A more affordable device, the SONY HMZ T1, provides 45° of horizontal field-of-view. If the rendering software used to create the images displayed in the HMD respects the geometrical configuration of the display, the vision of the virtual environment will cover exactly 45° too. It is common to artificially increase the visualized FoV by using a virtual camera covering a wider FoV [Bolte 10]. For example, a view of 60° of the virtual environment can be remapped to the 45° FoV of the HMD.

A HMD is mechanically linked to the user's head. To be coherent with respect to the head's rotations, a motion sensor has to be embedded [Bowman 04b]. It provides data corresponding to the head orientation so the visual feedback can be adjusted. There are

**Figure 1.20 –** The PiSight HMD from Sensic. Each screen is composed of several OLED matrixes, providing large FoV without sacrificing the definition of the displayed images.

many requirements on the data provided by a head tracker: low latency, high refresh rate and high precision. If these requirements are not fulfilled, the feeling of immersion is degraded and even motion sickness can occur [Allison 01]. However, such configuration has a benefit: even a cheap HMD, providing low FoV, allows visualizing 360° visual information when used with a head tracker. In this situation, the visual information is accessed sequentially.

### 1.4.2 Surround-screen displays

Surround-screen displays use another strategy to visually immerse the user in virtual environments. The rule is to cover the user's FoV by surrounding him with one or more display surfaces. One of the most famous design is the CAVE [CruzNeira 93, Bowman 04a], shown on Figure 1.21.

To create the visual immersion, the images are computed according to the position of the user's head in respect to the display surfaces. To do so, the user's head position is measured by a tracking device. An alternative to planar projection screens is curved surfaces [Simon 04], leading to wide screens with cylindrical or dome shape [Bourke 05] (Figure 1.22).

These display devices are able to cover up to the entire FoV of the user. As every display device, they need some source to display images. In the next section, the rendering methods to compute wide FoV images are reviewed.

**Figure 1.21 –** The CAVE (Cave Automatic Virtual Environment). The user is surrounded by planar screens where images are projected. The views are computed dynamically with respect to perspective projections computed taking the position of the user's eyes into account [CruzNeira 93].

**Figure 1.22 –** An IMAX Dome theater. Images are projected on a hemispherical screen with a fish-eye lens.

## 1.5 Rendering methods for wide field-of-view

These last decades, the research in computer graphics has designed several algorithms to create synthetic views of virtual environments. These algorithms can be classified into two main groups. These two main families are ray-tracing and rasterization. Rendering of wide FoV can be performed with all of these algorithms, but the approaches, the strengths and flaws can be very different from one to another [Foley 82, Shirley 09].

### Ray-tracing

Ray-tracing algorithms are based on the computation of the inverse path of the light. The image is rendered pixel by pixel. For each pixel, a ray direction is computed. Intersections between objects in the scene and this ray are then computed. Based on the nearest intersection and material properties of the object intersected, the final color for the pixel is computed. The projection equation operates in a decoupled manner during the computation of the initial direction of the ray. Therefore it is straightforward to use more complex equations such as those presented in section 1.2.2. However, a recurrent point of discussion about ray-tracing is the suitability of the algorithm for real-time rendering. Ray-tracing is very expensive in terms of processing power and is generally used to compute off-line images such as for animation movies or special effects. Therefore, this approach is not deeply discussed here.

### Rasterization

Instead, rasterization is the largely adopted approach for real-time rendering. It consists in converting the primitives composing the virtual world into blocs of pixels, one primitive after another. The primitive set is generally restricted to few simple primitives (points, lines, triangles). In modern graphic computing, the rasterization algorithm is executed on dedicated processor (Graphic Processing Unit) and part of the algorithm is even hardware implemented. An extreme simplification of the process could be seen as:

**Algorithm 1** Rasterization (simplified)

```
Input: Primitives composing a scene
Output: Image representing a view of the scene
for each primitive do                          ▷ loop over all the primitives in the scene
    express the primitive coordinates in the image coordinate frame
    convert primitive in group of pixels
    for each pixel of the primitive do
        compute pixel color
        if pixel in front of previously written pixel or no pixel written in this place then
            write the pixel in the image
        end if
    end for
end for
```

Modern rasterization algorithms are tightly coupled to hardware. Almost all real-time rendering software uses API (Application Programming Interface) such as OpenGL[1] or Direct3D[2]. The requirements in high quality rendering for mass consumers market (especially video games) has driven the evolution in graphic hardware design, notably the graphic processing units (GPU). The rasterization pipeline exposed by these API reflects this evolution (Figure 1.23). The purpose of the processing pipeline is to execute sequentially different stages to draw a primitive (point, line triangle, quad or polygon). Each stage owns a set of accessible primitive's data and functionality. The whole process is highly optimized, targeting high-end efficiency (thanks to hardware implementation) but few *programmable stages* (grey stages on the figure) let developers implement specific processing with high level programming languages such as GLSL (the shading language adopted by OpenGL). The remainder of the pipeline is not programmable but *state-configurable* (white stages on the Figure 1.23).

The pipeline is executed each time the main program calls the drawing of a primitive, a triangle for example (the pipeline is the same for other primitives, ie. points, lines or quads). The primitives are processed sequentially as follow:

- **Vertex shader**: each vertex composing the primitive is processed through a vertex shader. A vertex shader is small program that aims to process the vertex specific data: it transforms coordinates to change expression referential, computes color at vertex etc.

- **Tesselation stages**: the primitive data can then be processed by an optional group of stages able to amplify the geometry. These stages allow the generation of new vertices (several thousands) based on the primitive being rendered. Concretely, it allows subdividing the input primitive to refine the geometry by adding details while maintaining the original models lightweight.

- **Geometry Shader**: in this optional stage, the processing has access to all the vertices composing the primitive being rendered and as the tessellation shader, it can generate new vertices (1024 in general, far less than the tessellation shader since it is not its purpose). It also has several specific features, such as the property to write to special buffers, to store the result of the computation done up to this stage, thus avoiding the next stages and rasterization. This can be useful to perform general purpose computation on GPU instead of rendering images. Furthermore, it has the possibility to change on the fly the type of the primitive rendered (switching from a triangle to 3 lines for example). It can be used to route the geometry to several different layers in the frame-buffer (layered rendering). This feature is often used to generate a cube map in only one scene draw, the six perspective projections being applied by the geometry shader (generating the six views: up, down, right, left, forward, backward).

- **clipping stage**: here, parts of the primitive outside of the visible volume are simply discarded by clipping then. The process can generate new primitives or change the

[1]Mark Segal, Kurt Akeley - Opengl 4.2 core profile specification (updated april 27, 2012) - www.opengl.org/registry/

[2]Microsoft Direct3d 11 features - msdn.microsoft.com/enus/library/windows/desktop/ff476342

type of the current rendered one if needed. It is to be noticed that this part is not programmable. It can only be tuned by few parameters.

- **Rasterization stage**: finally, the primitive can be converted to a group (fragment) of pixels.
- **Fragment shader**: each pixel of the fragment is processed individually. This last programmable stage allows computing the colors for the pixel but also the value that can be written in depth buffer.
- **Tests and framebuffer writes**: the pipeline ends with optional tests such as depth test for hidden surface removal. If these tests are passed, the fragment can be written in the framebuffer to be displayed on screen.

This design gives enough versatility to target a wide range of rendering scenarii without sacrificing performances. Whatever the considered API, OpenGL or Direct3D, implementation is very similar. An algorithm that targets real-time performances needs to conform to or fit in this pipeline. This is the case for the algorithms presented hereafter.

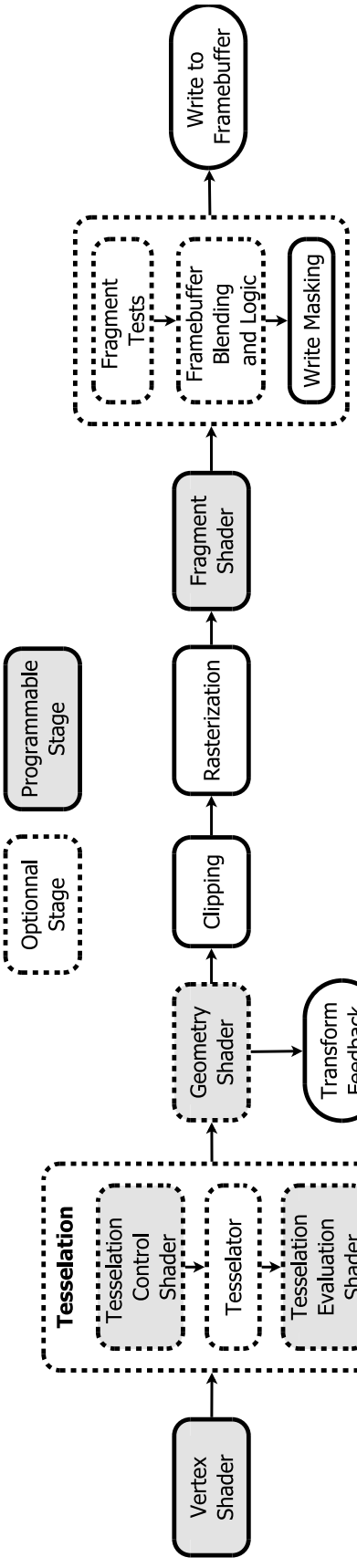



**Figure 1.23 –** Modern rasterization pipeline (OpenGL taxonomy). Graphic data are processed successively in different stages. Dashed stages are optional (can be bypassed) and grayed ones are programmable through shaders.

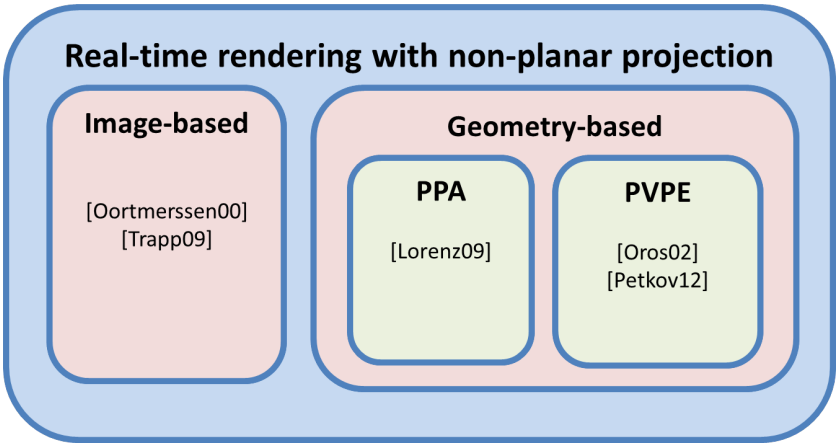



**Figure 1.24 –** Overview of real-time rendering methods for non-planar projections.

We can classify the methods for wide FoV rendering with rasterization as follows (see Figure 1.24):

- **Image-based methods:** An image-based method does not evaluate the non-planar projection during scene rendering. Instead, the scene is first rendered with perspective projections generating the six views: up, down, right, left, forward, backward. Then, the non-planar projection is used to address the images previously generated;
- **Geometry-based methods:** The other group consists of the geometry-based methods. Here, the polygons composing the scene are rasterized directly to the final view. However two different approaches can be chosen;
  - **Perspective projection approximation:** The former tries to approximate the non-planar projection equation with several perspective projections: Perspective Projection Approximation (PPA);
  - **Per Vertex projection evaluation:** With the later, the non-planar projection equation is evaluated for each vertex of the rendered geometry. This approach will then be called Per Vertex Projection Evaluation (PVPE).

The following sections propose an overview of these different approaches. Then, section1.5.3 will discuss different strength and weakness of these approaches.

### 1.5.1 Image-based methods for wide field-of-view rendering

In the case of image-based method, the VE is first rendered to six offline buffers with six standard planar projections generating the six views: up, down, right, left, forward, backward [Greene 86] (Figure 1.25). These offline buffers are often organized as cube map texture. Addressing of cube map textures is efficient: given a direction in space, the graphic hardware is able to return the corresponding color value from one of the six textures.

Although a projection is not necessarily bijective, the ones considered here have to. Since the final image is built by processing one pixel after another from the image frame, the inverse functions need to be used: the points of coordinate $(x,y)$ in the final image plane is mapped to its corresponding direction/point on the unit sphere with the help of

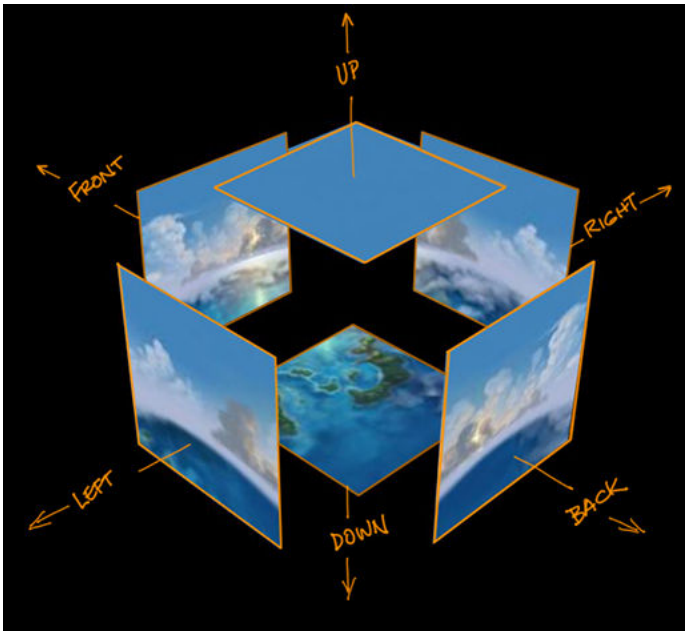



**Figure 1.25 –** Cube map are built to cover all the direction in space with six perspective projections sharing the same center.

the inverse function of the non-planar projection desired, see section 1.2.2. This direction is then used to address the cube map texture.

Van Oortmerssen used this approach in two modified versions of the famous first-person shooter game Quake: FisheyeQuake and PanQuake[3] (Figure 1.26).

**Figure 1.26 –** Fisheye quake is a modification of the ID software's game Quake. Van Oortmerssen uses image based approach to generate the wide FoV rendering[3].

This rendering approach has some limitations. First, it does not benefit from improvements of modern hardware rasterizers such as anti-aliasing or anisotropic filtering [Lorenz 09] and so it suffers sampling artifacts. Moreover, image-based method is not able to generate coherent stereo pairs [Trapp 09a] [Trapp 08]. Details about this aspect are given in section 1.5.3.

[3]W. van Oortmerssen - Fisheyequake - http://strlen.com/gfxengine/fisheyequake/

### 1.5.2 Geometry-based methods for wide field-of-view rendering

As explained above, we can classify geometry-based approaches into two groups. The first group considers an approximation of non-planar projection with several perspective projections (also known as **Perspective Pieces Approximation** or PPA). The second group computes the non-planar projection equation per vertex (called **Per Vertex Projection Evaluation** or PVPE in the remainder of this document).

A common process to perform PPA rendering is to render the scene geometry $n$ times according to the $n$ perspective projections used to approximate the desired non-planar projection, see Figure 1.27. In [Simon 04] this approach is used to adapt single user stereoscopic displays to multi-users scenario. Implementations perform efficiently with a small number of projection pieces [Bourke 06]. To alleviate saturation of vertex bandwidth when the number of pieces increases, Lorentz et al. [Lorenz 09] proposed a 3-pass rendering method. This method replicates the rendered primitive on the GPU and determines which projection pieces the primitive is covering. This approach is characterized by an extensive use of geometry shader but an alternative implementation using the tessellation unit of modern GPU can also be considered [Lorenz 09].

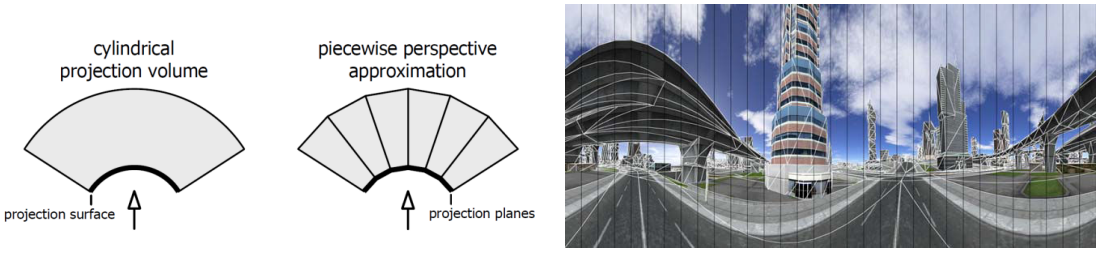



**Figure 1.27 –** Perspective Projection Approximation. The cylindrical projection is subdivided into several perspective projections (left). The resulting rendering approximates the cylindrical projection (right) [Lorenz 09].


**Figure 1.28 –** Curvilinear rendering is a PVPE method developed by Oros on the first generation of programmable GPU [Oros 02].

PVPE methods could appear conceptually simpler to design. Indeed PVPE is compliant even with early programmable GPUs. Oros proposed a vertex shader implementation that supports FoV up to 180° [Oros 02] (see Figure 1.28). Alternatively, more complex projections can be used. In [Brosz 07], a flexible projection can model arbitrary curved

viewing volumes. Moreover, the desired non-planar projection can be specifically computed to match spatial arrangement of display surfaces such as in [Petkov 12]. However, in all cases, two limitations remain. The former is the post-projection error on coarse geometry. The later limitation concerns the rendering of primitives crossing a discontinuity of the projection used [Petkov 12]. These limitations will be further discussed in section 3.1.

### 1.5.3 Synthetic analysis of existing methods

As presented in the paragraph just above, the different rendering methods can have various strengths and weaknesses. The limitations explained in details hereafter are summarized in the Table 1.1.

**Table 1.1 –** Comparison of the limitations of the possible approaches to render in real-time wide FoV images.

| | **Image-based** | **Geometry-based** | |
|---|---|---|---|
| | | **PPA** | **PVPE** |
| Scalability (vertex bandwidth impact) | Yes | No | Yes |
| Invariant to scene tessellation | Yes | Yes | No |
| Compatible with raster enhancements (AA filter) | No | Yes | Yes |
| Stereo compatible | No | Yes | Yes |
| Compatible with projection discontinuities | Yes | Yes | No |

#### Impact on vertex bandwidth

A first issue is how the method impacts the amount of vertices that needs to be processed. Given a scene with $n$ vertices, an image based approach processes $6 \times n$ vertices to render the six texture buffers (in the worst case without optimization). PVPE processes simply $n$ vertices. PPA instead needs to process $n$ vertices for each perspective projection used to approximate the non-planar projection.

#### Requirement on scene tessellation

When evaluating the projection at vertex with PVPE methods, a high tessellated geometry is required. For optimization and performance purposes, rasterizer implementations only propose to render flat, planar primitives delimited by straight edges, such as triangles, lines or polygons. This approach imposes to use transformations (projection, rotation, translation, scale,...) which preserve the shape of the primitives (after transformation a line remains a line, a triangle remains a triangle,...). Non-planar projections cannot guarantee this property. Therefore with PVPE, significant error between the rasterized result and the theoretical result can occur (see Figure 1.29). To keep this error visually acceptable, the PVPE method has to be applied on a geometry which is detailed enough in regard of the distortion of the non-planar projection used. This is a problem specific

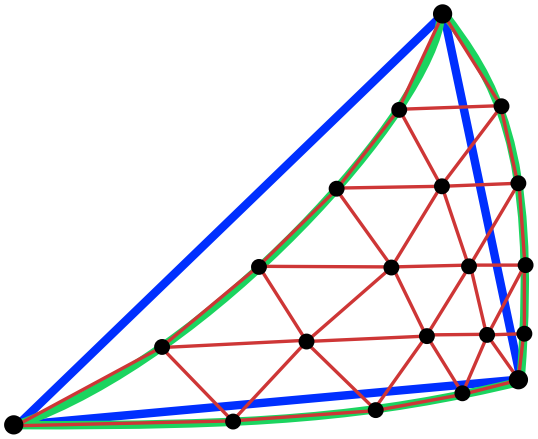


**Figure 1.29 –** Surface error after non-planar projection. Green: theoretical result after an arbitrary non-planar projection. Blue: result after applying non-planar projection on coarse geometry (simple triangle). Red: result on a tessellated triangle.

to PVPE approaches. The two other methods (image-based and PPA) do not suffer such problem because they perform the rasterization with perspective projections .

### Rasterization enhancements

Enhancements like anti-aliasing gather multiple samples for one pixel [Haeberli 90] from slightly different positions. This supposes that the rasterization of the geometry is performed in the final image space at the final image resolution. As image-based methods first render the scene into off-screen buffers, the benefit of such effect is lost and sampling artifacts happen [Lorenz 09].

### Stereoscopic rendering

When using image based method to render wide FoV images, the intermediate storage in cube map texture sets definitively the center of projection. To create a second center of projection the naive approach is to duplicate the process for a second center of projection. This leads to correct result for forward direction, but lateral directions, where the centers of projection are aligned, will result in zero parallax stereoscopy. For the part of the view corresponding to the backward direction, the parallax is swapped. The problem is described in [Trapp 09b]. To be consistent and meaningful, the generated view needs a parallax that is function of the distance of the rendered objects. This will match human's natural stereovision. Only geometry-based methods are able to provide such consistence for the stereo pair rendered. Both PPA and PVPE approaches (detailed in the sections just above) are suitable. Examples are notably provided in [Petkov 12] and [Trapp 09b]. With PPA approach, stereoscopic rendering can be achieved by rendering the scene two times, translating the projection center for each perspective piece according to the user's eyes position.

### Discontinuities of projection

A non-planar projection necessarily has discontinuities in the sense that the continuous surface of the unit sphere is unfolded to a plane (see [Snyder 87]). For example, in a cylin-

drical projection, there are discontinuities at the left and right borders of the plane (along the meridian where the cylinder was "cut") and at poles (see section 1.2.2). A primitive lying across a discontinuity of projection will be wrongly rasterized with a PVPE approach. Instead, image-based and PPA approaches perform the rasterization of the scene geometry by using several perspective (planar) projections which do not suffer discontinuity. This specific aspect will be discussed deeper in chapter 3.

Lastly, it seems there is no definitive method to achieve a wide FoV, stereoscopic and real-time rendering of VE. Various drawbacks exist, therefore it is difficult to find a method that is either compliant with projection discontinuity, well scalable, easy to implement (GPU-friendly) and efficient.

## 1.6 Conclusion

This chapter covered the state of the art of aspects related to wide FoV vision. First the physiological characteristics of the human's FoV were presented. Measurements show a coverage of up to 200° horizontally and 135° vertically. However the acuity is less sharp in the peripheral area.

To model wide FoV, the mathematical formulation relies on non-planar projections. The previous work about this topic were notably highly influenced by cartography, where mapping the points of the earth surface to a plane is equivalent to mapping directions in space to a plane. Therefore, proven methods developed for mapping the earth can be used for wide FoV images.

Hardware aspects encompass acquisition and display devices able to work with wide FoV. To handle these high FoV values, these devices need to rely on specific designs. Specific systems were notably developed, combining classical lenses and curved mirrors to create catadioptric sensors. For display devices, two different approaches are considered: making display wearable (HMD), or surrounding the user with display surfaces (immersive rooms).

The rendering methods able to provide wide FoV images and being real-time compatibles fall into two main families: image-based methods and geometry-based methods. Both of them reveal strengths and weaknesses.

# FlyVIZ: a display device to increase the human field-of-view in real environments

# 2

Contents



As for today, sophisticated optical devices can be used to adapt vision for various situations. For instance, microscopes and telescopes are classical optical devices that can magnify small or distant objects. These devices map a small part of the field-of-view (FoV) to a larger one. Such mapping process decreases the effective user's FoV. Increasing the natural FoV is however very difficult to achieve with traditional optical devices.

This chapter describes the design and prototype of a display device, which can enhance the human FoV, and enable a 360° horizontal FoV (80° vertically with our system). This new device (see Figure 2.1), called FlyVIZ, is based on three components: (1) an acquisition system on top of the user's head, for capturing 360° image), (2) a Head-Mounted Display (HMD) to display the processed image to the end-user in real-time, and (3) a computer vision algorithm to map the captured 360° images to the shape and dimensions of HMD screen.

In the remainder of this chapter, we first describe the global concept of FlyVIZ and its main components. Then, we present a proof-of-concept prototype and detail the proposed hardware and the real-time image processing algorithm. Finally, we discuss the performances and the potential uses of our novel approach.

## 2.1 The FlyVIZ concept

The objective of FlyVIZ is to enhance the natural human field-of-view. It is intended to reach a full 360° FoV of the user's surroundings, in real-time. This would fulfill a dream of humans: to be able to see behind their back or to see like some animals, such as horses or


**Figure 2.1 –** FlyVIZ prototype and proof-of-concept.

flies, with a wider FoV, even reaching a fully panoramic vision. This objective is difficult to achieve using simple and traditional optical devices. However, an image processing approach connecting an image acquisition system with a head-mounted display could suit the purpose.

As illustrated in Figure 2.2, three main functions should drive the design of such a system. First the system has to capture omnidirectional information/images from the user's environment (image acquisition). Second, the system has to transform this view into a meaningful representation (image transformation). Finally, the system has to display this view to the user (image presentation).

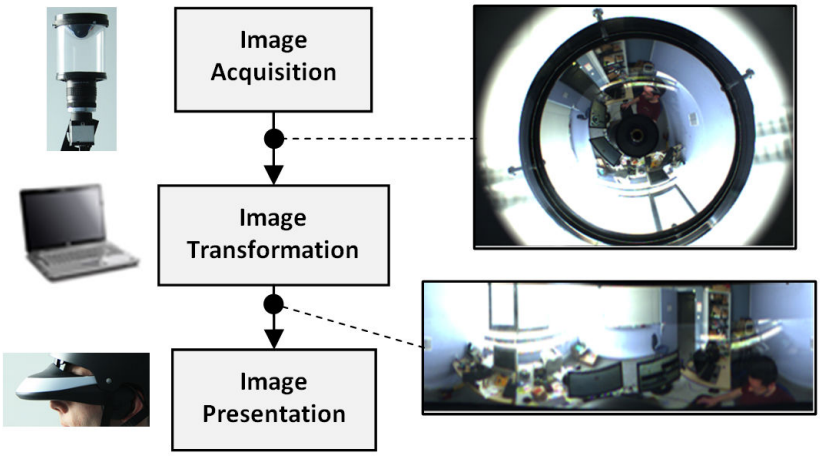



**Figure 2.2 –** FlyVIZ components

The image acquisition can be performed by a catadioptric sensor, i.e., the combina-

tion of a camera with a mirror. Various shapes of mirrors could be used such as parabolic, hyperbolic, or spherical mirrors. Composite sensors could also be used, i.e., a set of multiple cameras assembled in a circle or sphere to cover the 360° FoV. The image transformation requires a processing unit to carry out the necessary computations. This can be a laptop, a netbook, a wearable computer or dedicated hardware. Smartphones with enough processing power could also fit. The Image presentation can be made using a Head-Mounted Display, but specific video glasses could also be suitable. As a complementary requirement, the whole system must preferably be wearable and compatible with the user's locomotion

## 2.2 Proof of concept / hardware prototype

We describe a first prototype of the FlyVIZ concept displayed in Figure 2.1. The image acquisition step is achieved with a catadioptric sensor [Baker 98] made up of a hyperbolic mirror and a traditional 6mm-lens mounted on a CCD camera (IDS uEye 2210). The image acquisition system is mechanically attached on a helmet. Figure 2.3 displays a typical image acquired by this system. The panoramic images are acquired in the head reference frame (and so, it does not require head tracking). The image transformation is done on a laptop computer worn by the user in a backpack. The image presentation is done by means of a HMD (SONY HMZ-T1).

**Figure 2.3 –** Image provided by the catadioptric system

## 2.3 Image processing

In this section we further detail the image transformation process that is a key component of our system. The purpose of the image transformation algorithm is to transform the acquired image (Figure 2.3) into an image that can be displayed in the HMD. A comprehensible representation of the environment is targeted, i.e., a projection which can be

effectively perceived by the user. With our approach, each pixel displayed in the HMD has to be mapped to its corresponding location in the acquired image. The image transformation is achieved in two successive steps, which correspond to two different projections:

**1** Projection between a location on the final displayed image and a direction of the space;

**2** Projection between a direction of the space and its respective location in the acquired image.

For the first projection, the mathematical formulation of the problem consists in mapping all the space directions onto a plane. Mapping one direction of the 3D space is equivalent to map a point from a unit sphere onto a plane [Baker 98] . This mapping problem (and its inverse problem) has been widely studied by mathematicians and cartographers. We use the equirectangular projection (see section 1.2.2). Although the equirectangular projection generates some distortions at poles, it preserves the shapes along the 0° parallel. This helps the overall interpretation of the final view.

The second projection corresponds to the calibration of our optical system. The equations map a 3D space vector to its corresponding location in the acquired image. Details on the modeling and the calibration of catadioptric sensors can be found in section 1.3.5. In our setup we use a dedicated sensor calibration algorithm available in the OCamCalib toolbox [Scaramuzza 06a]. The procedure takes about 15 minutes, and does not require to be fulfilled again, as soon as the optical alignment of the camera lens and the mirror are preserved. The calibration parameters are then stored and can be used later, at runtime.

Mathematically, the whole process can be described as follows:

$$(x,y) \xrightarrow{P} (\lambda,\phi) \xrightarrow{S} (X,Y,Z) \xrightarrow{M} (x',y') \tag{2.1}$$

Thus, in the first step of our image transformation algorithm, a pixel with coordinates $(x,y)$ is transformed to $(\lambda,\phi)$ with the equirectangular inverse projection $P$ which is a simple affine transformation that remaps the $x$ coordinates to $[-\pi,\pi]$ and y to $[-\pi/2,\pi/2]$. Then, the 3D space vector $(X,Y,Z)$ is deduced from the parameterized form of the unit sphere equation $S$.

$$S\begin{cases} X &= \sin\phi\cos\lambda \\ Y &= \sin\lambda \\ Z &= -\cos\phi\cos\lambda \end{cases} \tag{2.2}$$

Finally, the associated coordinates in the input image $(x',y')$, are computed from $(X,Y,Z)$, using the camera model $(M)$ [Scaramuzza 06a].

At runtime, each raw image is acquired by the catadioptric sensor and uploaded into the video memory as a texture buffer. A full-screen quad is then rasterized by using a dedicated fragment program on GPU. To transform the raw image into the final unwrapped image, the fragment program handles the two mappings described earlier: the equirectangular projection and the catadioptric camera model. For each processed pixel in the final image, $(x,y)$ coordinates are deduced from UV coordinates of the full screen quad. The corresponding 3D vector is computed using the inverse mapping of the equirectangular projection. This vector is then used to address the input image according the catadioptric

camera model [Scaramuzza 06a]. This texture look up is executed using hardware texture linear filtering to maintain smoothness over the final image. Both, the raw panoramic image addressing and the plate equirectangular projection are implemented at a fragment level. Such an implementation benefits from the parallel processing power available in modern GPUs and optimizes computation performances.

## 2.4 System performances

Our prototype is based on an HMD with a FoV of 45° and a 16:9 aspect ratio (ref. SONY HMZ-T1). This aspect ratio fits relatively well with the properties of the equirectangular projection that constrains the displayed image to an aspect ratio of $2:1$. Taking our optical setup into account, the final image is displayed in the HMD with 360° horizontal FoV and 80° vertical FoV (Figure 2.5).

Our image transformation algorithm is implemented in C++ using OpenGL API and the GLSL shading language. We have benchmarked our algorithm on two different platforms with both high (laptop) and low computational performances (netbook). The resulting frame rates obtained in HMD are given in Table 1. With a high-end laptop, the frame rate is far above the refresh rate of the camera (60Hz). But even with a low-cost netbook, the refresh rate meets a real-time constraint (24Hz).

The overall latency of the system (end to end from acquisition to display) has been measured by taking pictures of a precision clock and its image processed by the system and displayed in the HMD. The value seen through the HMD was then subtracted to the directly observed value. With this procedure, we found an average latency of 83 ms using the high performance platform.

**Table 2.1 –** Computation performance of FlyVIZ's image processing (frame rates) benchmarked on two different hardware platforms.

| Hardware | CPU | GPU | Frame Rate (Hz) |
|---|---|---|---|
| 15' laptop | i7-2820QM 2.30GHz | Quadro 2000M | 480 |
| 12' netbook | ATOM D525 1.8GHz | ION2 | 24 |

## 2.5 System in use

The FlyViz system is fully operational and has been tested by multiple users and in different conditions (e.g., indoor or outdoor). We can use different demonstrative scenarios as illustrated in Figures 2.6 and 2.4. During these tests, users get used within the first 15 minutes of practice, letting them to smoothly move in their environment. Users also get used to the new visual feedback loop of their arms and hands, letting them open doors or grasp objects. In these last situations, depth perception is altered since binocular vision is not available, but as suggested by Cutting [Cutting 97], users seem to be able to base depth evaluation on the other depth cues (motion parallaxes, etc.).

(1) (2)

**Figure 2.4 –** Usage scenari. (1) catching a stick out of the natural field-of-view, (2) avoiding a ball thrown from behind.

For the main user, a first scenario consists in grasping an object (a stick) held out by another person. Without moving his/her head, the user instantly perceives the position of the stick and can grab it, even when it is located out of his/her natural field-of-view. In a second scenario, the user is walking and must avoid to be touched by some balls thrown at her/him, even from behind. A third scenario consists in driving a car on a parking lot, being able to see both the external environment and the car interior at the same time (Figure 2.6).

During several tests, the device has been worn for more than an hour, without cybersickness or particular visual fatigue. The main discomfort came from the unbalanced weight of the headset (helmet, camera, optics and HMD: 1650g). Future work is, of course, necessary now to evaluate the learning process, user perception, and the potential exploitation of 360° vision in various tasks.

**Figure 2.5 –** Image displayed in the HMD corresponding to the transformation of the raw image of Figure 2.3

**Figure 2.6 –** Illustrative scenario: driving a car on a parking lot (HMD view).

## 2.6 Conclusion

This chapter has introduced FlyVIZ, a novel display device to augment humans with a 360° vision. FlyVIZ combines an acquisition device and display device to present images with 360° horizontal field-of-view in real-time.

We have designed a proof-of-concept hardware system based on a catadioptric system with a standard camera and a commercial HMD. The prototype consists in a helmet worn by the user with a laptop computer placed in a backpack. The mobility of the system lets the user walk freely in its environment.

We have proposed an image processing algorithm based on an equirectangular projection to transform the acquired images into images compatible with HMD screens and aspect ratio. Our software implementation benefits from parallel processing power provided by modern GPUs. On a standard laptop, the system reaches a frame-rate of 480Hz with a measured latency of 83ms.

Finally, we have shown the operability of the FlyVIZ prototype in different indoor or outdoor scenarios. For example users were able to enjoy grasping an object held out behind their back without turning their head or driving on a (closed) parking lot.

A second version of the prototype, using a smartphone as the processing device, a compact catadioptric lens and the popular Occulus rift as HMD has been developed. It is illustrated in Figure 2.7.

The work discussed in this chapter was presented at ACM Symposium on Virtual Reality Software and Technology (ACM VRST) 2012 in Toronto [PUB2]. The concept and prototype have also been patented [PAT1].


**Figure 2.7 –** FlyVIZ V2. The second iteration of the prototype uses a smartphone, a compact catadioptric lens and an Oculus Rift HMD.

# A Real-time rendering method with wide field-of-view in virtual environments

# 3

Contents



Many virtual reality (VR) applications could require a wide field-of-view (FoV). However, the rendering of virtual environments (VE) with wide FoV raises several critical issues [Petkov 12]. Current approaches are either image-based or geometry-based [Trapp 09b], but often fail at being at the same time: stereo-compatible, GPU-friendly, and able to handle problems related to discontinuities of non-planar projections (see section 1.5).

This chapter proposes a novel geometric method for the stereoscopic rendering of VE with large FoV, i.e. above 120° and up to 360°, as illustrated in Figure 3.1. The method computes the non-planar projection equation for each vertex, and includes an innovative pre-clip stage in order to solve the problems occurring with polygons spanning across the projection discontinuities.

In the remainder of this chapter, we describe step by step the novel approach to render wide FoV stereoscopic images in real-time. The results and overall performance of our method are reported and discussed in section 3.2.2. The Chapter ends with a conclusion.

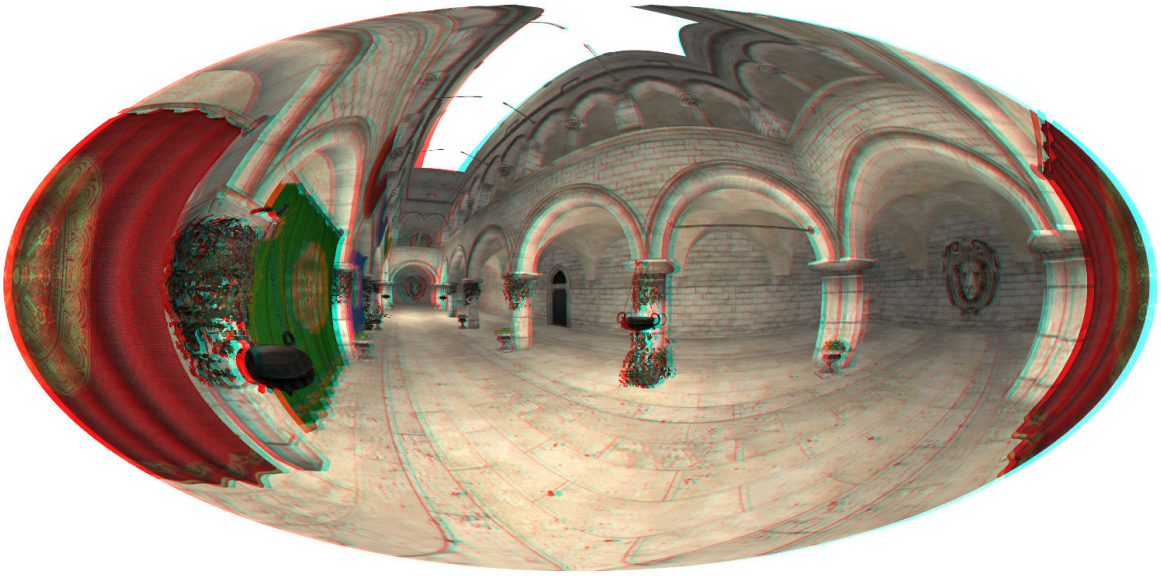


**Figure 3.1 –** Stereoscopic rendering of a virtual environment covering a 360° horizontal field-of-view. The rendering is based on the Hammer projection which is classically used in cartography. Anaglyph image to be viewed with red/cyan glasses.

## 3.1 A novel approach for stereoscopic rendering with non-planar projection and wide FoV

Per vertex projection evaluation (PVPE) of non-planar projection seems to be a compelling technique thanks to its compatibility with the standard rasterization pipeline. From a technical point of view, a geometric PVPE approach for wide FoV rendering involves two main steps. First, the geometry is projected into the unit cube. For large FoV, a non-planar projection is mandatory. Second, the transformed primitives are rasterized through scan conversion, i.e. transformation of vectorial data to bitmaps. For performance reasons, the hardware accelerated rasterization pipeline has to be considered, alongside with its restrictions. More details about the rendering pipeline and the PVPE methods can be found in section 1.5.

### Limitations of current PVPE methods

Therefore, the use of non-planar projection induces two major issues:

- Firstly, the projection does not preserve shapes anymore, leading to significant post-projection errors for geometry with low tessellation level.
- Secondly, the projection includes discontinuities: if a triangle spans across a discontinuity, it is not properly rasterized.

For instance, if we consider a cylindrical equirectangular projection, a triangle spanning across the *Oyz* half-rear plane, represented by red and blue triangles in Figure 3.2, is rasterized improperly as shown in Figure 3.3. The vertices of the polygon are well transformed by the projection equation (their positions after transformation are coherent),

but the clipping stage of the standard pipeline is unable to detect and process such kind of polygon. Instead, the primitive is sent without modification to the next stage and is rasterized across the entire viewport. To make up for this issue Petkov et al. [Petkov 12] propose to simply not render a primitive spanning across a discontinuity of the projection. Alternatively, we propose to include a new clipping stage in the rendering pipeline. This will be presented in section 3.1.3.

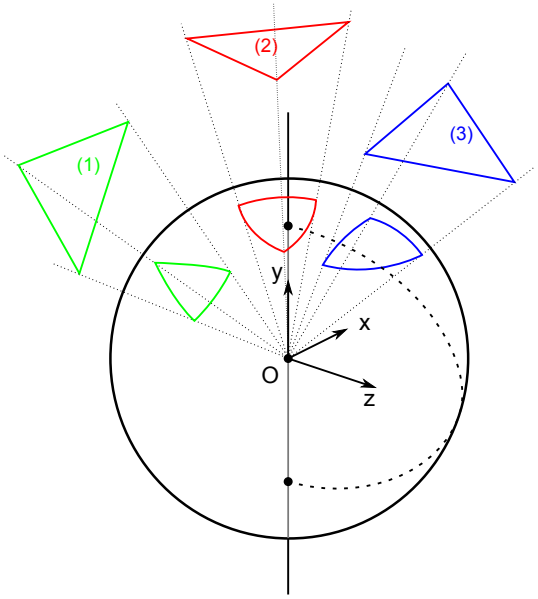



**Figure 3.2 –** Issues when using an equirectangular projection. The projection discontinuity is represented by the dashed line. Triangle (1) is compatible with standard rasterization. Triangles (2) and (3) generate issues at clipping and rasterization stages.

Therefore, to solve the technical challenges induced by wide FoV rendering, we propose an updated rendering pipeline that alleviates limitations of PVPE approaches. The successive steps of this pipeline are illustrated in Figure 3.4.

### 3.1.1 Building a projection suitable for wide FoV real-time rendering

To be suitable for hardware-accelerated rasterization, a non-planar projection needs to remap the desired view volume to the unit cube. For $x$ and $y$, the equirectangular projection (eqation (1.8)) described in chapter 1 is a mapping among others that fits the requirements. In any case, to fit a rasterization pipeline, the equation need to be completed with a $z$ coordinate suitable for depth test. Equation (3.1) matches this requirement by remapping the euclidean distance of the considered point to $[-1,1]$.

$$\begin{cases} x &= \frac{2}{h_{fov}}\lambda \\ y &= \frac{2}{v_{fov}}\phi \\ z &= -1+2\frac{\rho-near}{far-near} \end{cases} \tag{3.1}$$

Where $\lambda = \arctan\frac{X}{-Z}$, $\phi = \arcsin\frac{Y}{\rho}$ and $\rho = \sqrt{X^2+Y^2+Z^2}$.

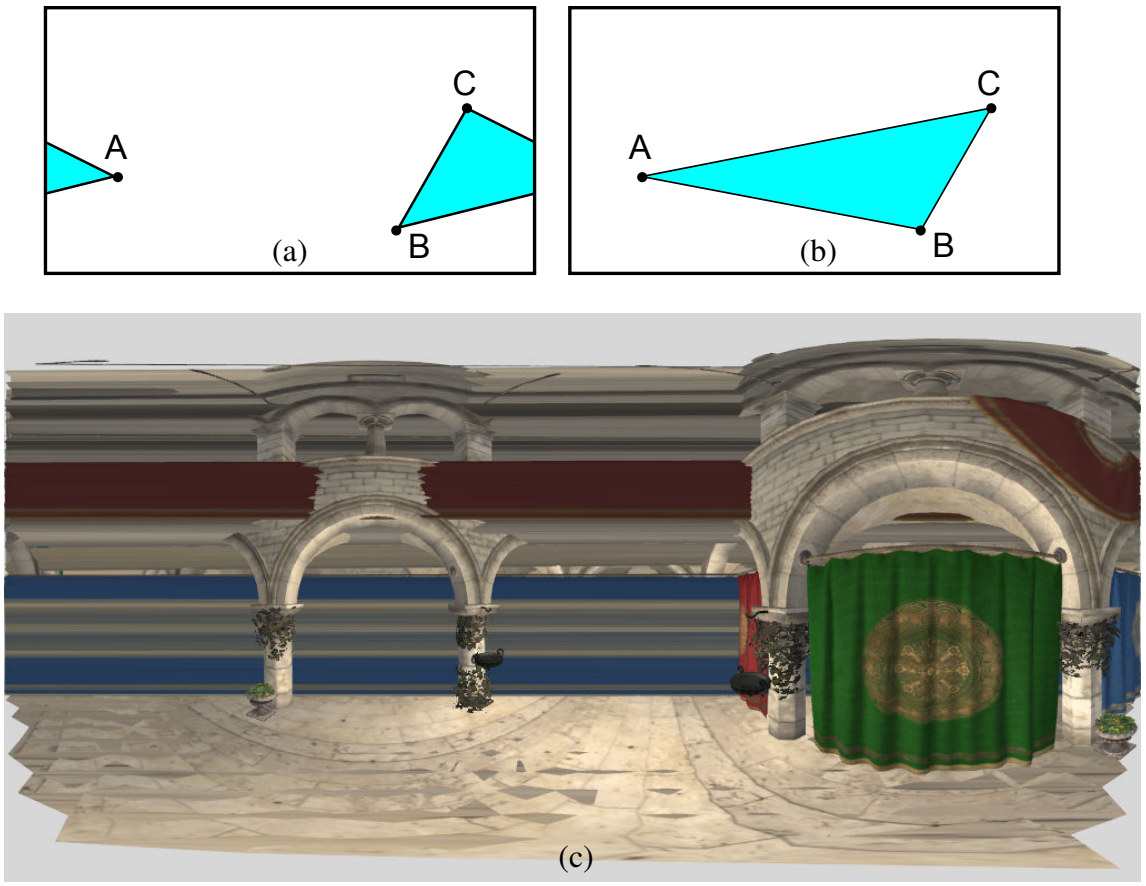



**Figure 3.3 –** Rasterization issue for triangle projected across the discontinuity of an equirectangular projection. (a) desired rendering in the final viewport, after projection. (b) the surface actually generated. (c) the problem on a real scene: here, the triangles composing the red and blue drapes on the very right are rasterized across the whole viewport.

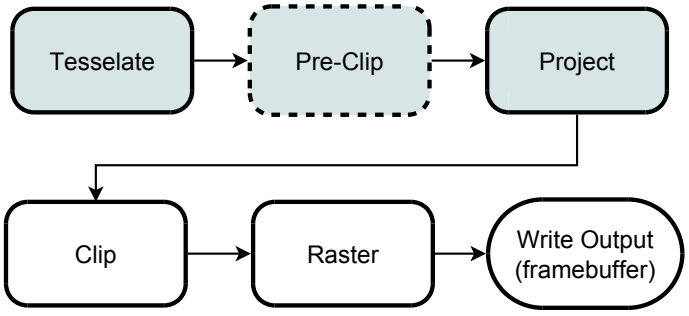



**Figure 3.4 –** Rendering pipeline proposed for wide FoV real-time rendering. Gray stages are implemented in shaders. The new pre-clip stage (dashed) represents our main contribution.

Several other projections can be applied [Snyder 97], selected according to various criteria. The shape of the display surface can be considered [Petkov 12] as well as the properties of the projection used. Conformal projections preserve angles locally. Equidistant and equal-area projections preserve angular distances or surface area (see section 1.2.2). This offers a wide range of possibilities in terms of visual rendering and final perception.

The amount of FoV covered by the final image can be controlled through scalars $v_{fov}$ and $h_{fov}$. The view volume rendered corresponds to $x$ and $y$ coordinates in $[-1,1]$. To cover the full FoV, one can chose to use $h_{fov} = 2\pi$ and $v_{fov} = \pi$. Finally, the $Z$ coordinate is generated as the distance from the projected point to the center of projection, also remapped to $[-1,1]$. To do so, two clip distances are used to define clipping spheres instead of clipping planes used in the perspective case. The whole projection equation is then able to remap the spherical view volume to the unit cube as shown in Figure 3.5.

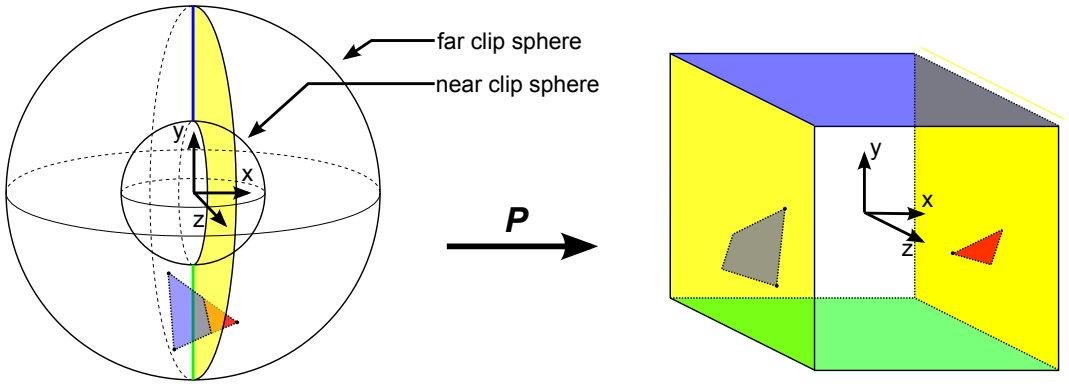



**Figure 3.5 –** Equirectangular projection used for rasterization. The projection maps the volume included in the two spheres to the unit cube. The mapping exhibits discontinuities: correspondence of surfaces is highlighted with colors (blue, yellow and green). The small triangle (see left image) is cut in two parts. This illustrates the need for a new kind of clipping.

### 3.1.2 Tessellation of the virtual environment geometry

PVPE approaches impose a prerequisite on the tessellation of the input geometry. An overview of the requirements can be found in [Petkov 12]. Main guidelines are provided hereafter, in order to clarify its inclusion in our rendering pipeline as displayed in Figure 3.4.

A non-planar projection, such as equirectangular or Hammer projection is built to cover a wide FoV, but introduces distortions in the final view, transforming straight lines into curves, and polygons to curved surfaces. The curvature of the transformed primitive has to be approximated (see section 1.5.3 for an illustration). The granularity of the geometry needs to ensure that the footprint of the primitive after projection is small enough to constrain the projection error to visually acceptable value. To fulfill these needs, the geometry can be processed offline before the rendering or online with the tessellation feature available in modern GPU. Both processes can also be applied. Using a tessellation shader in this context offers the possibility, at each rendered frame, to control the way each polygon is subdivided. Primitives suffering from high distortion can be subdivided to a

finer level whereas primitive in low distortion area of the projection can remain coarser. Such an approach is described in [Petkov 12].

### 3.1.3 Our novel "pre-clip" stage

As illustrated in Figures 3.2, 3.3 and 3.5, when using a non-planar projection, problems arise for triangles spanning across a discontinuity of the projection. The standard clipping algorithm is unable to detect and handle these particular cases. To be correctly rasterized, these triangles need to be specifically processed before applying the projection. For FoV smaller than 180 degrees the problem can be avoided by culling polygons behind the center of projection. For FoV above 180 degrees, current implementations simply discard the rendering of this kind of triangles [Petkov 12]. To maintain the rendering of these triangles, an alternative solution is to clip them. Considering an equirectangular projection, the additional clipping step has to prevent situations in which a primitive is wrongly rasterized due to a span across the projection discontinuity. Focusing on triangle as input primitive, there are two configurations inducing problems (see Figure 3.2):

- *Pole triangle*: the triangle spans across the $y$ axis;
- *Rear triangle*: the triangle spans across the rear half part of the $Oyz$ plane.

Our novel approach consists in splitting the primitive in sub-parts that do not span across a discontinuity. The guideline for the algorithm is first to detect if a triangle needs to be clipped. If so, its edges are processed sequentially to detect if one crosses the discontinuity. New vertices are generated on the discontinuity, and their attributes are computed using barycentric interpolation. If needed, the coordinates of the newly generated vertices are slightly corrected to guarantee that the vertex keeps a coherent position after being projected. In the case of equirectangular projection, the $x$ coordinate is forced to keep the same sign as the other vertices composing the generated triangle, as illustrated in Figure 3.6. The algorithms 3 and 2 describe simplified versions of the algorithms implemented for the rear and pole triangles processing. For readability, it omits barycentric interpolation as well as small correction of values introduced to avoid numerical computation errors. Figure 3.7 illustrates the rendering, after equirectangular projection. The two types of clipping are highlighted: pole triangles are displayed with plain colors and rear triangles with soft gradient colors.

Our approach can comply with any non-planar projection. However the pre-clip stage must be adapted to the discontinuities of the chosen projection. Nevertheless, one strategy for the pre-clip stage can fit for a whole family of projections. For example, the procedure detailed previously fits all cylindrical and pseudo-cylindrical projections [Snyder 97] such as Hammer projection (Figure 3.1).

### 3.1.4 Computation of a coherent stereo pair and adaptation to immersive displays

To generate coherent images for left and right eyes, the warped volume is considered as a regular object. Therefore, the rendering can rely on classical method used for real-time

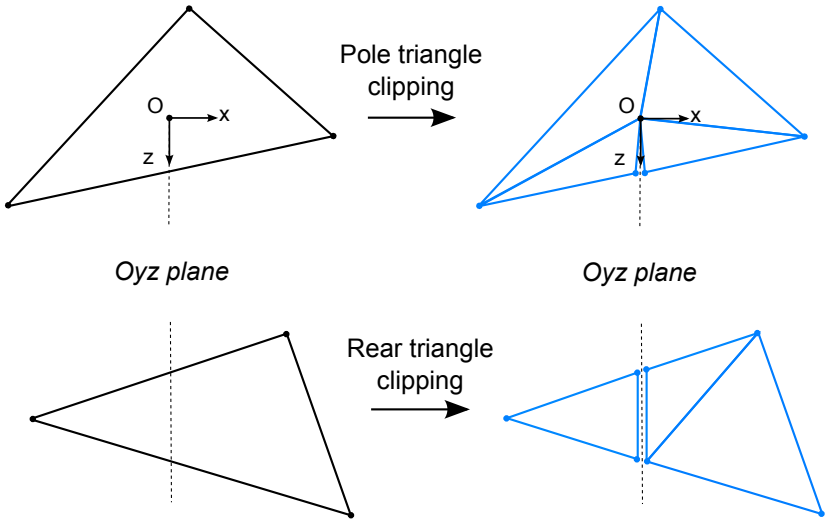



**Figure 3.6 –** Clipping strategies for pole triangles, projected from top or bottom and spanning across $Y$ axis, and rear triangles, projected from rear half-space and spanning across $Oyz$ plane.

**Algorithm 2** Pole Triangle Clipping

**Input:** A triangle $T$, $v_i$ stands for the $i$th vertex
**Output:** A set $OT$ of triangles ready to be projected with an equirectangular projection
**if** $Oy$ intersects $T$ **then**
  $I \leftarrow computeIntersection(T, Oy)$
  $id \leftarrow 0$ ▷ Current ID in vertice list
  **for** $i \leftarrow 0, 2$ **do** ▷ Check each edge for split
    $VL_{id} \leftarrow v_i$
    $id \leftarrow id + 1$
    $temp \leftarrow OyzIntersect(v_i, v_{i+1 \bmod 3})$
    **if** $temp$ **then**
      $VL_{id} \leftarrow temp$
      $id \leftarrow id + 1$
    **end if**
  **end for**
  **for** $i \leftarrow 0, 3$ **do** ▷ Store the 4 triangles
    $OT_i \leftarrow Triangle(I, VL_{i+1 \bmod id}, VL_{i+2 \bmod id})$
  **end for**
**end if**

**Algorithm 3** Rear Triangle Clipping

**Input:** A triangle $T$, $v_i$ stands for the $i$th vertex
**Output:** A set $OT$ of triangles ready to be projected with an equirectangular projection
**if** $Oyz$ intersects $T$ **then**
  $s_0 \leftarrow OyzIntersect(v_0, v_1)$ ▷ Compute $v_0v_1$ edge / $Oyz$ plane intersection
  $s_1 \leftarrow OyzIntersect(v_1, v_2)$
  $s_2 \leftarrow OyzIntersect(v_2, v_0)$
  **if** $s_0$ or $s_1$ or $s_2$ **then** ▷ An edge is crossing the $Oyz$ plane
    $id \leftarrow 0$ ▷ Current ID in vertice list
    $ns \leftarrow -1$ ▷ Handle the non split edge id
    **for** $i \leftarrow 0, 2$ **do** ▷ Check each edge for split
      $VL_{id} \leftarrow v_i$
      $id \leftarrow id + 1$
      **if** $s_i$ **then**
        $VL_{id} = s_i$
        $id \leftarrow id + 1$
      **else**
        $ns \leftarrow id$
      **end if**
    **end for**
    $OT_0 \leftarrow Triangle(VL_{(ns+1) \bmod id}, VL_{ns+2 \bmod id}, VL_{ns+3 \bmod id})$
    $OT_1 \leftarrow Triangle(VL_{ns+3 \bmod id}, VL_{ns+4 \bmod id}, VL_{ns+1 \bmod id})$
    $OT_2 \leftarrow Triangle(VL_{ns+1 \bmod id}, VL_{ns+0 \bmod id}, VL_{ns+4 \bmod id})$
  **end if**
**end if**

**Figure 3.7 –** VE rendered with equirectangular projection. The triangles processed and clipped in the pre-clip stage are highlighted with colors. At the bottom, pole triangles are rendered with solid colors. At the left and right borders, rear triangles that were clipped are rendered with soft colors.

stereoscopic rendering. Two additional perspective projections are used, as illustrates Figure 3.8. First, a zero parallax plane is positioned at the desired distance in the unit cube. It defines the $Z$ where stereoscopic disparity will be null (at this specific distance, left and right images are merged). The two centers of projection (CoP) are positioned according to the desired inter-ocular distance. Near and far clipping planes are also required. They can be chosen to match the unit cube to optimize Z-buffer precision, but this is not mandatory. The final stereoscopic effect can be tuned through three parameters: the inter-ocular distance (*iod*), the user's distance from the projection plane ($d$) and the position of the zero-parallax plane in the unit cube (*offset*). After preliminary testings, we could set the values of these parameters to: $iod = 0.5$, $d = 5$ and $offset = -0.8$. Values are given in the normalized device coordinate frame.

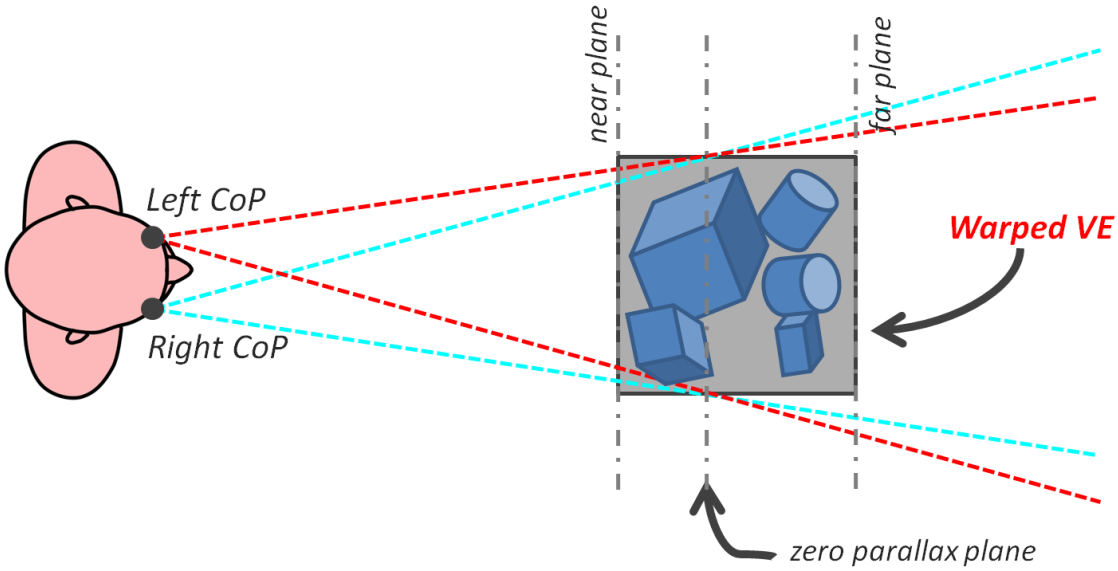



**Figure 3.8 –** Non-planar projection stereoscopic rendering. After being mapped to the unit cube with the non-planar projection, the VE is rendered with two perspective projections as in classical stereoscopic rendering. Two centers of projection, a near clipping plane, a far clipping plane and a zero parallax plane fully determine the two projections.

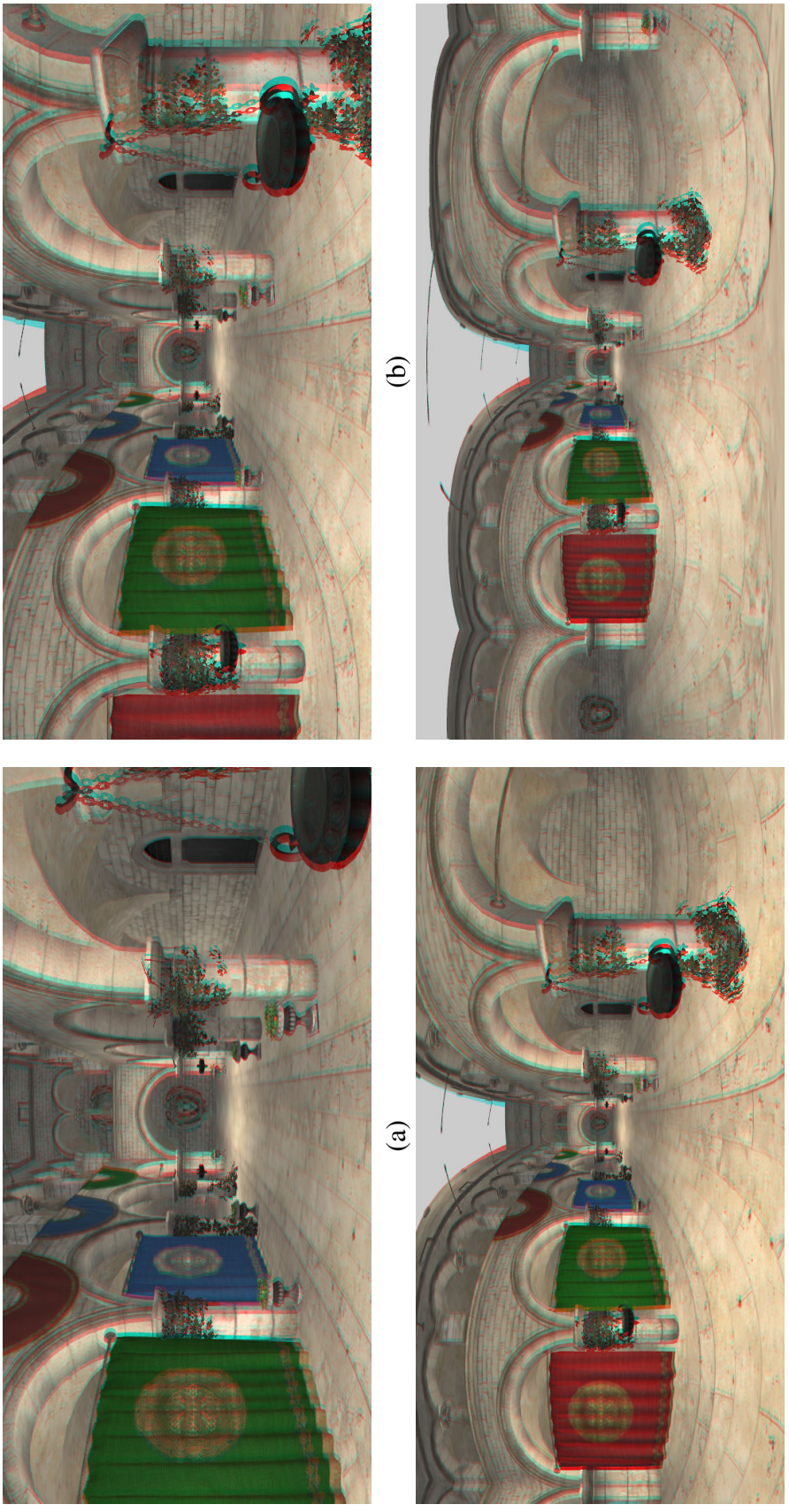



**Figure 3.9 –** Stereoscopic rendering using equirectangular projection and covering different FoV: (a) 120°, (b) 180°, (c) 270° and (d) 360° (anaglyph images).

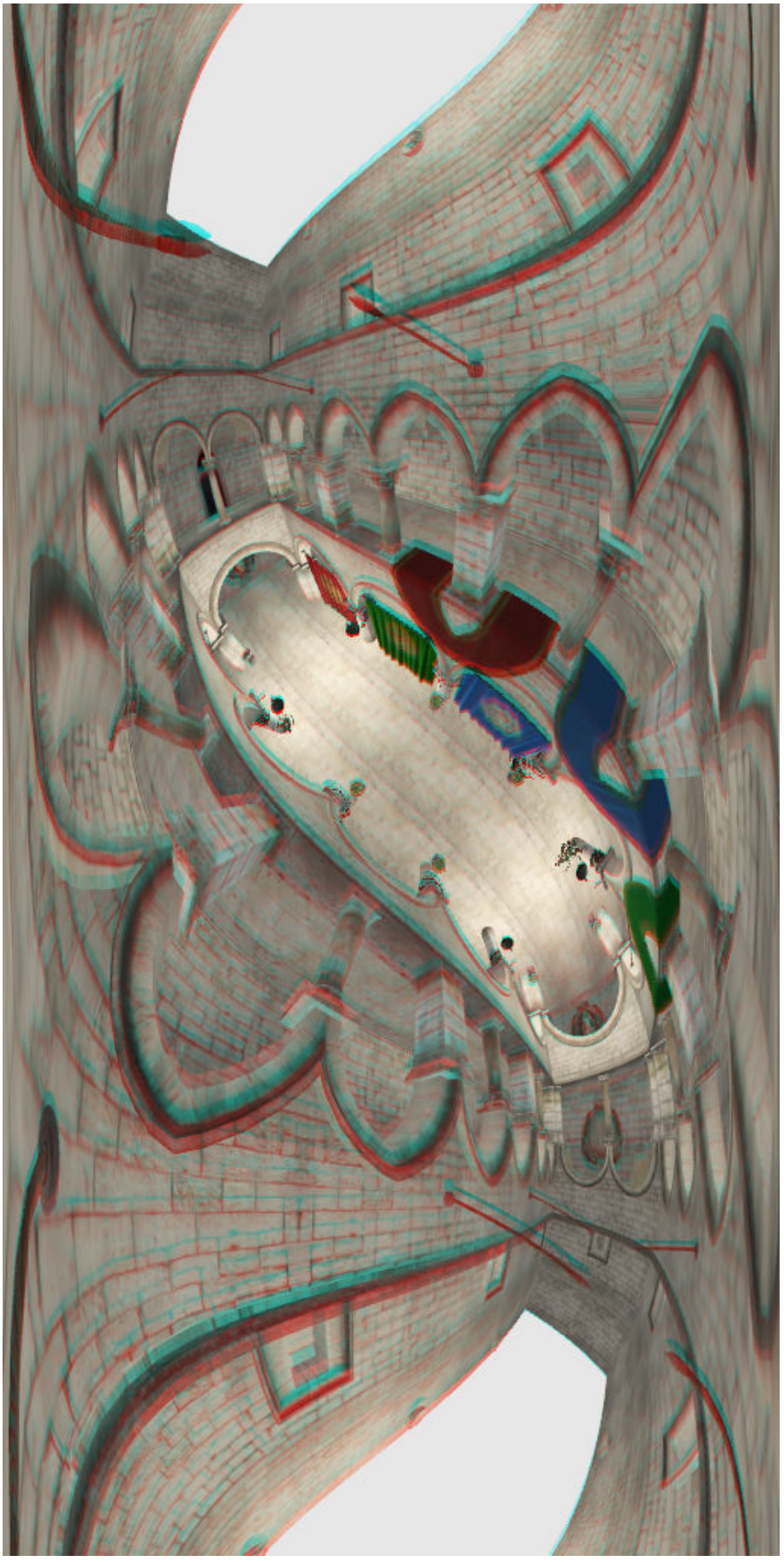


**Figure 3.10 –** Stereoscopic rendering of the VE viewed from the top and using equirectangular projection (anaglyph image).

There are various display systems that can be used to immerse a user in VEs: HMD, immersive rooms (e.g. CAVE), workbenches, or several other configurations of screens surrounding the user. The method we propose can be adapted to most of these setups. Adapting our method to HMD mainly consists in integrating the data from a three degrees-of-freedom tracker (for head rotations). The data is used to modify the model-view matrix and has no influence on the projection used for the rendering. With immersive displays based on planar projection screens (CAVE, Wall, etc.) the data provided by a tracking system with six degrees-of-freedom needs to be integrated. A straightforward method consists in considering the geometry after the projection as a regular object, fitting the unit cube. This object can then be rendered with classical approach [CruzNeira 93], where the tracking data control the center of projection for each perspective projection. Depending on the immersive display targeted, the unit cube may be scaled or translated to match the working area of the display. The method could also be used to target display systems with non-planar screens, such as domes or cylinders [Bourke 06]. In these cases the non-planar projection should be chosen to match the physical frame of the display. The final image could then be generated in one single pass.

## 3.2 Proof of concept and results

The approach presented was implemented to target modern personal computers. Details about the implementation, the results obtained and a discussion are presented hereafter.

### 3.2.1 Implementation

Our approach was implemented in C++ language using OpenGL 4.2 and the GLSL shading language. The selected VE retained is the Sponza Atrium model[1]. The model was modified offline with a digital content creation tool to subdivide its geometry, offloading the tessellation stage during real-time rendering. All the programmable stages of OpenGL 4.2 pipeline (Figure 1.23) were required to implement the proposed pipeline (Figure 3.4).

From a sequential point of view, vertices are first transformed in the vertex shader, handling the model-view transformation. Vectorial vertex attributes such as tangents, normals and bi-tangents are transformed with the inverse transpose model-view matrix as in regular rendering.

The next step consists in refining the geometry (see details in section 3.1.2). With the pre-tessellated scene used, a constant level of additional tessellation in the tessellation control shader was found sufficient to limit the projection error. Alternatively, a more optimal algorithm as proposed in [Petkov 12] could be used.

In the next step, the assembled primitives are pre-clipped and projected in the specially designed geometry shader. The pre-clipping algorithm presented in section 3.1.3 needs to access the list of the vertices composing the primitive. Therefore, the geometry shader is the designed place to implement this stage. The primitives are finally transformed with the desired non-planar projection (section 3.1.1).

[1] Frank Meinl - Sponza model - http://www.crytek.com

Concerning the fragment shader implementation, the Phong lightning equation is evaluated in eye space in order to keep the computation independent from the projection. However, deferred shading approach could be used if required [Deering 88].

Stereoscopy feature was implemented using two different techniques for image separation: anaglyph (color filter separation, ideal for compatibility and diffusion) and frame sequential (also known as active stereoscopy, for a better viewing experience). The two images were rendered sequentially, in two different drawing passes. Alternatively, layered rendering could be implemented to process the scene only once per frame [Trapp 09b].

### 3.2.2 Results

Consequently, our method is able to perform real-time stereoscopic rendering with FoVs up to 360°. Figures 3.1, 3.9 and 3.10 illustrate different examples of stereoscopic images obtained. These images are rendered using anaglyph stereoscopy. Red/cyan glasses are required for viewing. Figure 3.10 represents a top view of the Sponza Atrium VE. In this particular case, the understanding of the resulting view is not trivial. Distortions in the final rendering and the unusual point of view can mislead the viewer. The addition of stereoscopy seems to ease the perception of the structure and the geometry of the VE. But future work would be necessary to formally assess the potential gain in perception.

The performance of our rendering pipeline was assessed using a standard graphics configuration and a single monitor screen. The test platform runs on Intel i3 2120, running at 3.3GHz, with 4Go RAM, and an NVidia GeForce 560 GTX graphic board. Vertical synchronization was disabled. Stereoscopy was disabled (multiplying the final performance by nearly two). Rendered viewport sized $1024 \times 512$ pixels. Two different projections were tested: equirectangular and Hammer. Two VEs with different complexities were used for the benchmark: the Sponza Atrium as a complex pre-tessellated VE (225839 triangles), and a cornell-box scene as a simple VE (34 triangles). Table 3.1 displays the resulting framerates, for both VEs, depending on the different FoVs and projections used.

**Table 3.1 –** Performances of our approach (framerates) as function of FoV, complexity of the VE to render, projection used and usage of anti-aliasing.

| Type of Projection | FoV(°) | Simple VE (fps) | Complex VE (fps) |
|---|---|---|---|
| Perspective | 120 | 948 | 58 |
| Equirectangular | 120 | 488 | 42 |
| Equirectangular | 180 | 449 | 40 |
| Equirectangular | 270 | 446 | 40 |
| Equirectangular | 360 | 459 | 42 |
| Equirectangular with CSAA32x | 360 | 396 | 36 |
| Hammer | 360 | 476 | 38 |
| Hammer with CSAA32x | 360 | 470 | 39 |

### 3.2.3 Discussion

Perspective projection was implemented conserving the pipeline presented in section 3.1, maintaining all the stages but bypassing the pre-clip stage which is no longer necessary. The results for perspective projection and the equirectangular projection give an estimation of the pre-clip stage impact on framerate. Interestingly enough, several cutting-edge hardware improvements such as antialiasing could be incorporated with small processing-time costs. As an example, hardware antialiasing (CSAA 32x) was introduced and tested with no visible impact on framerate. Moreover, the FoV extension seems to not influence the rendering performance. In fact, the FoV tuning is available through the mapping to the unit cube (Equation (1.8)). The main processing duty and the main bottlenecks seem related to the geometry processing steps (clipping and projection) which are implemented in the geometry shader. As our implementation does not feature optimization such as frustum culling, all the geometry is processed whatever the FoV used. Therefore, setting a smaller FoV does not imply better performances. The fragment processing is here a quite simple standard per-pixel Phong shading that does not seem to impair the final performance.

From the user side, a limitation could be observed concerning the perception of the rendered environments. The human FoV is physiologically limited. Providing more FoV in a view can be viewed as the remapping of a wide FoV to a smaller one, fitting the human natural FoV. This induces a down-scaling of the VE observed. Thus, the perception of scales could become biased, even when using an immersive display. Alongside with the scale perception, perception of depth could be also impaired. However as it can be deduced from section 3.1.1, the function used to remap the $z$ coordinates can be tuned. This represents a field of investigation to maximize the quality of the final depth and stereoscopic effects. A limitation could be the accuracy of the resulting image in areas of high distortion. Artifacts can be seen in Figure 3.7 and 3.9.c: the curvature of the pavement is not regular. Rasterization and interpolation of vertex attributes introduce this bias by using linear interpolation. There are two ways to improve this shortcoming. One way is to increase the tessellation of the VE. In this case, a trade-off between performance and image quality has to be done. Another solution consists in performing rasterization of curved primitives. Such methods have already been experimented [Gascuel 08] but are difficult to make efficient (need to enclose the primitive in a bounding shape, discard a lot of fragments...). Future hardware evolution could ease such approach by allowing the programming of the rasterization stage.

## 3.3 Conclusion

Wide FoV rendering relies on non-planar projection to generate a view from a 3D model. Specificities of non-planar projection introduce several rendering problems, assessed differently according the rendering method used: image-based or geometry-based. Considering the geometry-based methods, PVPE approaches usually fail at rendering polygons spanning across the projection discontinuities.

To solve the rendering problems of these polygons, we propose to modify the ren-

dering pipeline to include an innovative pre-clip stage. This stage permits to split the primitive in sub-primitives which are not spanning across the projection discontinuity. Therefore the sub-primitives generated are correctly rasterized.

Our method was demonstrated with two cylindrical projections: the equirectangular projection and the Hammer projection. However, other non-planar projections can easily be supported by adapting the clipping algorithm to the specific discontinuities of the projection chosen. Furthermore, we have extended the method to handle the computation of stereoscopic images and the use of immersive VR systems.

We have implemented the proposed approach using modern hardware accelerated rendering pipeline and we have measured its performances with standard desktop computers and graphics cards. Results successfully meet real-time performances, being capable of displaying a wide range of FoVs, up to 360°.

The work discussed in this chapter was presented at IEEE International Conference on Virtual Reality (IEEE VR) 2014 in Minneapolis [PUB4].

# A method to solve the frame cancellation problem in real-time stereoscopic rendering

# 4

Contents



Frame cancellation is defined as the conflict between two depth cues: stereo disparity and occlusion with the screen border. When this conflict occurs, the user suffers poor depth perception of the scene.

In this chapter a novel method to avoid frame cancellation in real-time stereoscopic rendering is proposed. This method is based on the observation of the stereoscopic viewing volume shape. To solve the D/FO conflict, the proposed approach renders only the part of the viewing volume that is free of conflict by using clipping methods available in standard real-time 3D APIs. This volume is called Stereo Compatible Volume (SCV) and the method is named Stereo Compatible Volume Clipping (SCVC).

The chapter is organized as follows: firstly, section 4.1 details the problem and its consequences, section 4.2 proposes an overview of the existing methods to solve the D/FO conflict and an analysis discussing their limitations and suitability to real-time rendering. Our approach is described in section 4.3, and then it is evaluated in section 4.4.

## 4.1 The frame cancellation problem

Situations where frame cancellation can occur is not limited to wide FoV stereoscopic image, it can occurs in any stereoscopic rendering that display objects with negative parallax (ie. off screen, toward the user). This effect is observed since the early works on stereoscopic movies. Valyus named the phenomenon frame cancellation [Valyus 66].

Stereo disparity tells the user that the perceived object is in front of the screen while the occlusion with the screen border indicates that the screen border is in front of the object. When this conflict occurs, the user suffers poor depth perception of the scene [Wartell 02]. It also leads to uncomfortable viewing and eyestrain due to problems in fusing left and right images [Lipton 07].

Figure 4.1 illustrates the typical configuration of left and right eye viewing volume for a determined display surface. Intersection of the left and right viewing volume is not subject of frame cancellation while the remaining volume represents location of geometry that leads to disparity/frame occlusion conflict (D/FO conflict). Let it be noticed that the positive parallax volume is not pictured since it is not subject to the frame cancellation problem.

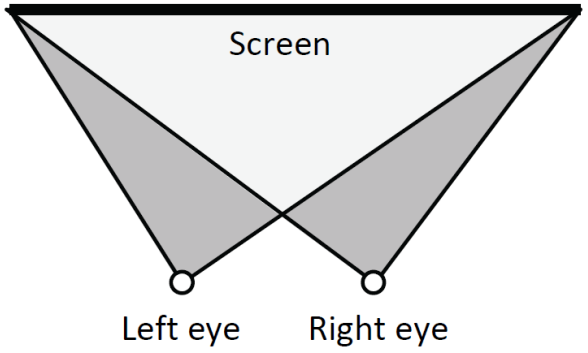



**Figure 4.1 –** Top view of the left and right camera volumes. Light gray: stereo compatible volume. Dark gray: volume subject to disparity / frame occlusion conflict.

## 4.2 Previous work about methods to prevent frame cancellation

Although the problem has been assessed using various technical solutions in stereoscopic movies and other offline computed stereoscopic rendering, fewer methods have been developed with real-time rendering specificities in mind.

### 4.2.1 Methods for offline stereoscopic rendering

Some digital content creation products like Autodesk Maya propose to display the intersection of the 2 view volumes (called safe viewing volume). This technique is meant

to help the creators in placing objects in situation free of D/FO conflict [Autodesk ]. The floating window is a common approach. It consists in inserting black bands on the left and right edges of the screen if the displayed content is subject to D/FO conflict [Autodesk 08]. The result is that the perceived content appears to be in positive parallaxes related to a virtual sub window positioned in user space (Figure 4.2).

Other approaches consist in modifying the disparity of the stereo pair globally or close to the screen edge. This can be seen as a local distortion of the 3D space, ensuring zero or positive parallaxes in areas where frame cancellation occurs. Lipton et al. propose to stretch the stereo pair at the border to insure zero or positive parallaxes while preserving enough negative parallax at the center of the screen so that the viewer experience remains convincing [Lipton 07].

Modifications of the stereo pair disparity are generalized by Lang et al. [Lang 10] who have proposed operators on disparity maps. They are used in different manners to modify disparity of the stereo pair to achieve effects such as emphasizing subject, shifting the scene depth or magnifying the depth range. Given a stereo pair, a disparity map is computed using sparse feature matching. Operators are applied to this map in the way that tonal transformations are applied to bitmap images. Finally the stereo pair is warped to reflect the new disparity map. This is done by employing retargeting techniques [Rubinstein 08], preserving the structure of images in both the spatial and temporal domain.

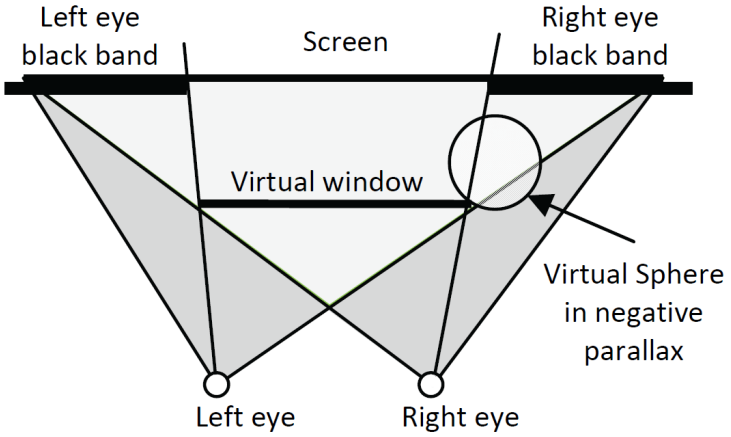



**Figure 4.2 –** Black bands. The part of the virtual sphere subject to frame cancellation is discarded by restricting the viewing volume with black bands.

### 4.2.2 Methods for real-time stereoscopic rendering

Mulder et al. have proposed a real-time implementation of the Black Bands through the Cadre Viewing approach [Mulder 00]. An improvement is also proposed with the Tunnel extension. With an overlay grid, the tunnel can improve depth perception by providing points comfortable to fuse.

A different approach was proposed by Ware et al., who have introduced the concept of cyclopean scale, presented as a rescale of the viewed scene centered on the midpoint between the user's eyes [Ware 97]. Magnitude of the scaling is computed to place the virtual

environment just behind the screen. This clearly avoids frame cancellation as it shifts the scene to the positive parallax half space. Combined with interactions, controls of scene scale and translation have to be performed dynamically in an intelligent manner. This problem is assessed by Wartell et al. through various derived approaches, for example by analyzing the depth buffer to compute valid parameters [Wartell 02].

### 4.2.3 Limits of existing methods

In the category of methods for offline rendering, black band insertion is probably the most adapted to real-time rendering. The main drawback occurs in situations where strong negative parallax objects are subject to D/FO conflict. In this kind of configuration, the virtual window can be very small and dramatically reduce the horizontal field of view. The aspect ratio of the view volume is also heavily altered. Another limitation comes from the dependency of the method on scene analysis: to determine the width of black bands we have to test each object against the stereo compatible volume. Also a part of the SCV is discarded along with the problematic portion. Cadre viewing [Mulder 00] limitations are roughly the same as black bands. The Tunnel extension restricts even more the viewing volume.

The stretch method [Lipton 07] and disparity map operators [Lang 10] are based on the analysis of the disparity of the stereo pair. The disparity map of the stereo pair is usually computed with offline algorithms such as sparse feature matching. A real-time version of these kinds of algorithms could rely on the depth buffer in conjunction with the left to right eye transformation matrix to generate a disparity map. The remaining limitation is the non-constant scale of the perceived scene along screen space that could lead to disturbing effects, especially when used with head tracked display.

Disparity map operators also are difficult to use in real-time as retargeting techniques imply specific computation and human guided tuning in some cases. Its need of time forward information to maintain consistency in the image structure over time is another weakness for real-time rendering.

Cyclopean scale and its variations differ radically from the above methods because they have been designed to target real-time rendering and real-time interactions. The application has to comply with scale modification of the displayed scene (with head tracked displays, maintaining 1:1 scale is often a requirement). The type of geometry to which the method applies is also limited to convex objects so exploration of an open environment is not an option. Cyclopean scale restricts the view volume to positive parallax or very low negative parallax values. Since the negative parallax portion of the viewing volume is an useful area for collocated interactions, this point can be considered as another limitation.

A summary of these limitations can be found in table 4.1. Although frame cancellation is an old and already studied problem, considering these various drawbacks, it seems there is no definitive method to prevent it in the scope of real-time rendering applications.

**Table 4.1 –** Properties and limitations of existing methods for solving disparity/frame occlusion conflict.

| | Black Bands [Autodesk , Mulder 00] | Border stretching [Lipton 07] | Disparity Map [Lang 10] | Cyclopean scale [Ware 97, Wartell 02] |
|---|---|---|---|---|
| Real-time Compatibility | Yes | No | No | Yes |
| Content analysis dependency | Yes | No | No | Yes |
| Disparity analysis dependency | No | Yes | Yes | No |
| Scene scale alteration | No | Yes (at borders) | Yes | Yes |
| View Vol. aspect ratio alteration | Yes | No | No | No |

## 4.3 Our novel approach: stereo compatible volume clipping

Methods presented in the previous sections are dependent on the displayed content due to the analysis of the scene geometry or the analysis of the disparity of the stereo pair. In real-time rendering, this task has to be automated and executed on each frame, leading to potentially complex and time consuming algorithms.

### Definition

In the negative parallax half space, we can define the SCV as the intersection of the left and right view volumes (see Figure 4.1). From a purely geometric point of view, one can solve the D/FO conflict problem optimally by rendering only the SCV.

### Approach

Our method does not try to avoid situations where frame cancellation occurs; it simply rejects parts of the viewing volume that are subject to D/FO conflict. By rejecting these regions, SCVC is independent of displayed assets and does not rely on content or stereo pair disparity analysis: wide environments are treated in the same manner as small convex objects. The method also ensures the widest volume is displayed on both positive and negative parallaxes while aspect ratio is maintained. No special modification of the scene like scaling or translation is done. The view volume aspect ratio is not altered either.

### Implementation

Clipping is a common method used to discard parts of polygons rendered outside of the view volume. Modern 3D APIs let the programmer add arbitrary clipping planes in the rendering pipeline. To insure that only stereo compatible volume is rendered, we add the left clipping plane of the right eye camera to the clipping planes list when rendering the right image and vice versa. We can notice that the view volume in positive parallaxes' half space is not altered.

Implementation using the OpenGL fixed pipeline is straightforward in the case of simple stereoscopic rendering (asymmetric frustums and $x$ axis aligned eyes). The clip plane of interest can be computed using 3 points: center of projection and screen plane top and bottom points (Figure 4.3). For stereoscopic displays without head tracker, this can be computed once and reused. Cost in terms of computation is marginal for plane equation solving while polygon tests against the additional clip planes stay cost effective when handled by the GPU.

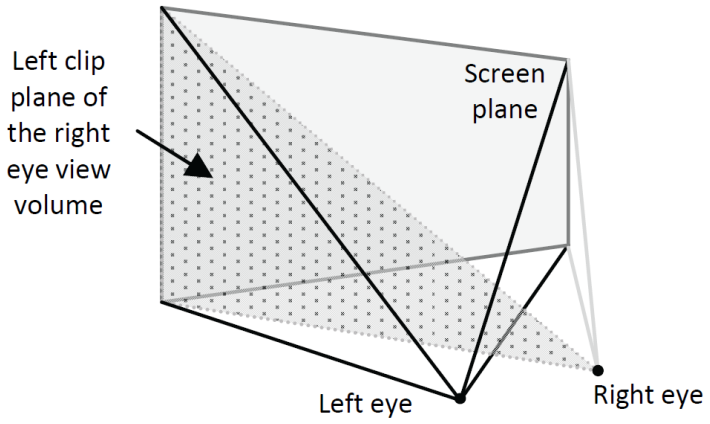



**Figure 4.3 –** SCVC method: Perspective representation of left eye view volume and left clip plane of right eye view volume. This clip plane is used to clip polygons during the rendering of the left view.

With head tracked displays, the eyes are not necessarily aligned horizontally, making the shape of the intersection of the two view volumes more complex. Adding two clipping planes per view is enough to restrict the rendered space to the SCV. In this case, the eyes' relative position determines the clipping planes that have to be used. We can select at each frame the horizontal clipping plane to use with a simple algorithm (we suppose that left and right eyes are not inverted). For the left view volume rendering: if the right eye is higher to the left eye then the bottom clip plane of the right eye view volume has to be added to the clipping planes list. If the right eye is lower to the left eye, add top clip plane of right eye view volume to the clipping planes list.

## 4.4 Evaluation

Our objective was to compare SCVC with Black Bands (which seems to be the best state-of-the-art approach, a method recommended in movie industry and already successfully adapted to interactive real-time rendering). Cyclopean scale approach tends to avoid the

problem by altering the scene representation (scaling and translation) so it have not been selected.

### 4.4.1 Participants

The evaluation was conducted using a panel of 22 participants, 19 males and 3 females aged from 18 to 37, from two groups: naive and initiated users. The 13 naive users had no knowledge of rendering techniques and have very limited experience with stereoscopy. The 9 expert users are university students who have followed real-time rendering courses with explanations of stereoscopic rendering mathematics and algorithms. Subjects suffering from binocular vision pathology were excluded.

### 4.4.2 Experimental apparatus

The experiment apparatus consists of a CRT screen and Stereographics' shutter glasses connected to a computer with an OpenGL stereo enabled graphic board. The user is positioned seated with his head aligned with the screen center at about 80cm.

The test application is written with OpenGL fixed pipeline. It shows a sphere of 6 cm in diameter to the user, moving in the foreground of a plane. The sphere is positioned at 15 cm from the screen plane in the negative parallax half view volume and moves slowly on the x axis with a sine movement from the left to right border. A simple checkerboard texture is used to map the scene objects to let the user fuse left and right images more easily. Care is also taken to set lighting and contrast of the scene at relatively low values to reduce crosstalk (aka ghosting effect) between left and right views. At runtime, the software lets the user toggle the method used by typing a corresponding key. The three methods proposed are Control (no special processing), SCVC and Black Bands (Figure 4.4).

### 4.4.3 Procedure

The experiment starts with a phase where the user can switch between methods as he wishes. Then he is asked open questions about the different methods. Finally he is asked to evaluate different criteria for each method by giving them a score. During the question and evaluation phase, the user remains free to switch from one method to another as he likes.

### 4.4.4 Data collected

A subjective questionnaire was proposed in which participants had to grade from 1 (low appreciation) to 7 (high appreciation) the three techniques (SCVC, Control and Black Bands) according to 5 subjective criteria: (a) Relief Quality, (b) Relief Quality at Borders, (c) Aesthetic, (d) Eye Strain perceived and (e) Global Appreciation. Figure 4.5 shows the results concerning the grades (Likert-scale) obtained by the three different techniques for each of the subjective criteria.

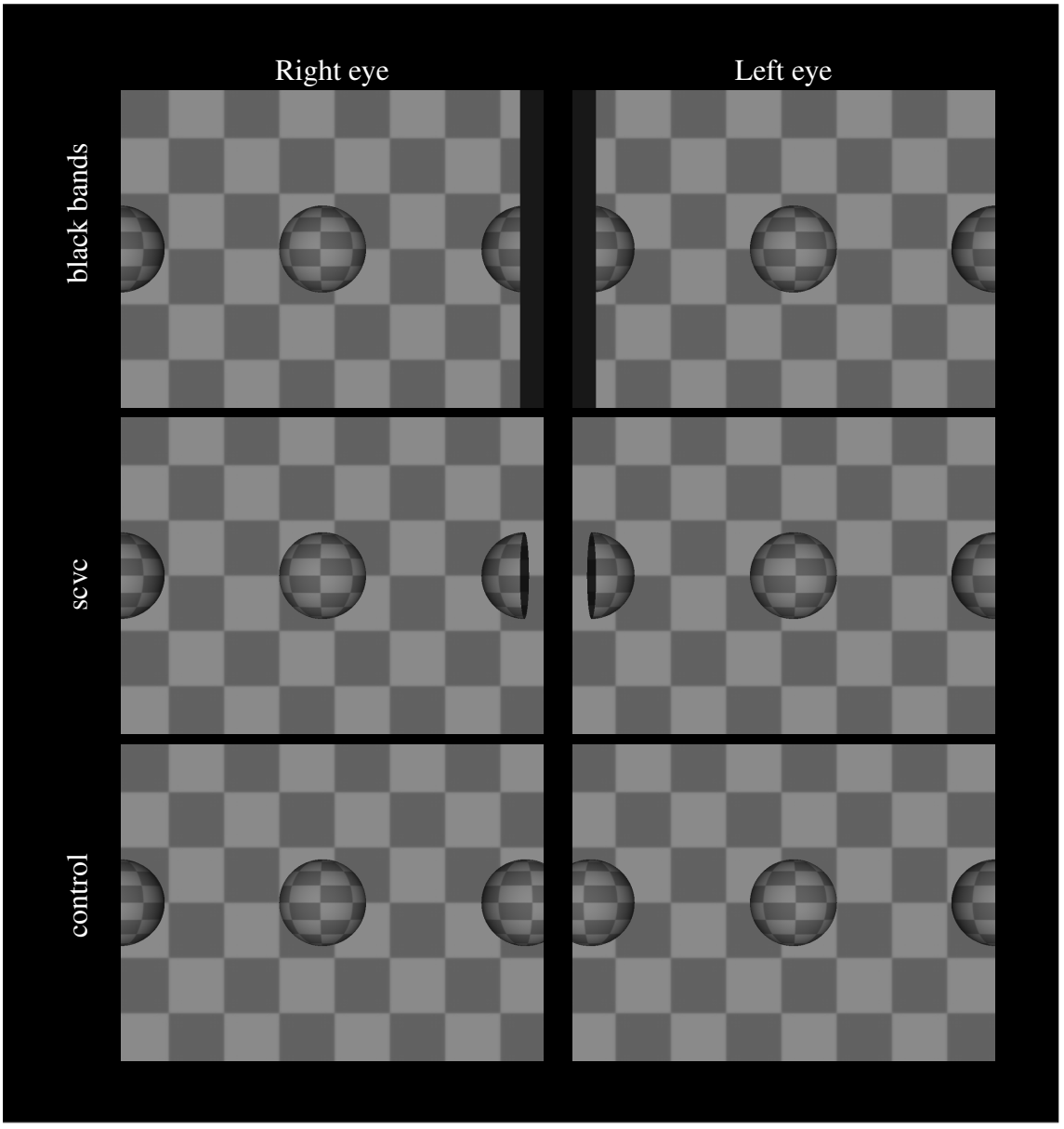



**Figure 4.4 –** Stereoscopic images in the 3 experimental conditions: Black Bands (state of the art method), SCVC, and Control (no special processing). The left and right images are switched ; a reader comfortable with cross eye stereo can fuse left and right images to see the effects.

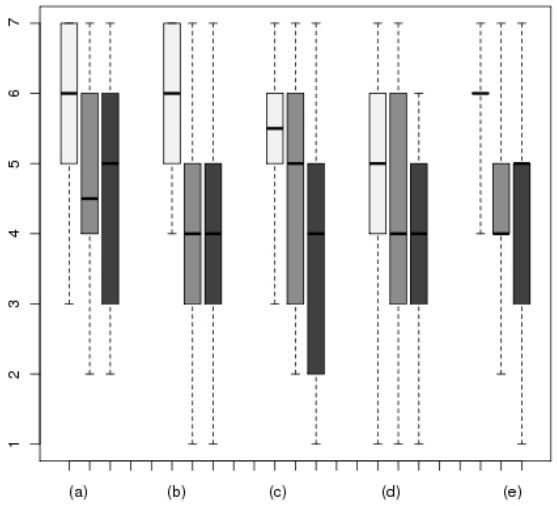



**Figure 4.5 –** Results for the subjective questionnaire for the three different techniques (1) SCVC (white), (2) Control (light grey) and (3) Black Bands (dark grey), with respect to a Likert-scale grading. The subjective criteria are (a) Relief Quality, (b) Relief Quality at Borders, (c) Aesthetic, (d) Eye Strain perceived and (e) Global Appreciation. Each box-plot is delimited by the quartile (25% and 75% without outliers) of the distribution of the condition over the individuals. The median is also represented for each technique.

### 4.4.5 Results

A one-way ANOVA was performed on the three different methods (SCVC, Control and Black Bands). A significant effect was found for all criteria, except Eye Strain criterion ($F(2,21) = 2.31, p-value = 0.109$).

Concerning Relief Quality criterion, a significant effect was found ($F(2,21) = 4.75, p-value = 0.012$). A post-hoc analysis using Tukey's procedure showed that the grading for SCVC method was significantly higher than for Control method ($p-value = 0.02$) and Black Bands method ($p-value = 0.03$). There was no significant adjusted p-value between Control and Black Bands methods.

Concerning Relief Quality at Borders criterion, a significant effect was found ($F(2,21) = 9.15, p-value = 0.0003$). A post-hoc analysis using Tukey's procedure showed that the grading for SCVC method was significantly higher than for Control method ($p-value = 0.001$) and Black Bands method ($p-value = 0.001$). There was no significant adjusted p-value between Control and Black Bands methods.

Concerning Aesthetic criterion, a significant effect was found ($F(2,21) = 7.69, p-value = 0.001$). A post-hoc analysis using Tukey's procedure showed that the grading for SCVC method was significantly higher than for Control method ($p-value = 0.0006$). There were no significant adjusted p-values between Control and Black Bands methods, and SCVC and Control methods.

Concerning Global Appreciation criterion, a significant effect was found ($F(2,21) = 12.018, p-value < 0.0001$). A post-hoc analysis using Tukey's procedure showed that the grading for SCVC method was significantly higher than for Control method ($p-value = 0.0004$) and Black Bands method ($p-value = 0.0001$). There was no significant adjusted $p-value$ between Control and Black Bands methods.

### 4.4.6 Discussion

When analyzing the result, the first remarkable point is the surprisingly low score of the Black Bands method. The marks suggest that the method does not solve the problem and that the perception at the screen border is even worse than when no treatment is applied. We think that this could be explained by the use of a CRT screen with white surrounds for the evaluation. When applied to projected images in dark rooms (theaters), the black bands make the user fuse the black band edges with image edges of the same color and luminosity. Besides, users' comments have also highlighted the modification of the view volume by the Blacks Bands, one declaring "I feel like I'm looking at a reversed 16/9 TV".

The SCVC results confirm that SCVC solves the D/FO conflict problem giving a better depth perception of the sphere in situations when it is partially occluded by the screen borders. Some users have complained of the slicing effect induced by the clipping. This can be disturbing depending on the experience the application designers want to give to the user. For applications centered on photorealism or visual attractiveness, the slicing effect might represent a drawback. Some improvements are possible to minimize this problem. For instance, sliced edges could be filled or progressively dissolved. Clipping along a plane orthogonal to the screen is another option, although it would make the method dependent on scene analysis to define the clipping plane. Future work is now necessary to improve this aspect of the method. Nevertheless, applications that do not focus on aesthetic aspects, such as in CAD software, could already benefit from SCVC directly.

## 4.5 Conclusion

Frame cancellation is a common problem faced in stereoscopic displays. Recent solutions investigated in the movie sector are not easily applicable to real-time rendering and the few methods designed to target VR still suffer from drawbacks. Notably they require scene contents analysis or alter of aspect ratio of the view volume displayed.

We have proposed a novel method that solves the problem in a different way: it only renders the part of the viewing volume that is free of D/FO conflict: we called this volume the Stereo Compatible Volume (SCV) and named the method the Stereocopic Compatible Volume Clipping (SCVC). To discard the problematic part of the viewing volume, the left clipping plane of the right eye projection is added to the clipping planes list when rendering the right image and vice versa. Therefore, this approach does not rely on the analysis of the stereo disparity or the analysis of the scene geometry.

We have conducted an evaluation to compare SCVC with a proven method and a control condition where the disparity/frame-occlusion conflict was happening. Results have shown scores in favor of SCVC for various criteria. SCVC notably improved global appreciation and user's depth perception near the screen edges.

The work discussed in this chapter was presented at IEEE International Symposium on 3D User Interfaces (IEEE 3DUI) 2011 in Singapore [PUB1].

# Experimental evaluation of navigation in virtual environments with a wide field-of-view

5

Contents



Field-of-view (FoV) of immediate visual feedback is generally constrained when navigating in virtual environment (VE). Typical visual rendering in recent video games relies on classical perspective projection, providing a horizontal field-of-view usually around 108°.

Various technical solutions have been proposed to provide an extended field-of-view through non-planar projection. Surprisingly, current techniques do not take advantage of the projection methods developed in the cartography field [Snyder 97]. In this domain, numerous methods are appropriate to display and apprehend a large set of (360°) information onto a 2D plane, such as for sailors or pilots.

Therefore, this chapter intends to investigate users' reaction to an extended field-of-view for the purpose of real-time navigation in VEs. Projection methods inspired by cartography field (see Figure 5.1) are used, and exploited for real-time omnidirectional rendering in VE. The main contribution is an experiment conducted on the use of such omnidirectional visual feedback in a navigation task. During this experimentation, the users were asked to reach targets by navigating in the virtual environment with a first person view. The rendering of the view was computed using the following projections: perspective, equirectangular, Hammer, Albers conic and azimuthal equidistant.

In the remainder of this chapter, the projections used and the rendering technique are described in section 5.1. The user study conducted is then presented in section 5.2. This

section notably describes the population and the experimental apparatus employed, the procedure, the data collected, the results obtained. Finally, these results are discussed.

**Figure 5.1 –** First-person navigation in a VE with a 360° omnidirectional rendering using a Hammer projection method rendered in real-time.

## 5.1 Cartography projection methods for real-time 360° omnidirectional rendering

Previous work experimenting wide FoV are often using intuitive formulation and implementation of mapping equation. But there are actually other projection methods in cartography field which were not applied to real-time rendering. In this experiment, we propose to adapt relevant projection methods from cartography to real-time 3D rendering. In our study, we have intentionally considered only the projection methods able to provide a full lateral 360° FoV. We want here to reach the maximum visual information, at least in the lateral direction. But, of course, the whole approach could also apply to a 180° FoV rendering which would match the physiological characteristics of human vision.

The chosen projection methods were selected to feature interesting properties like equidistance or equality of area and to cover the main families of projections used in cartography field. The three main unfoldable surfaces - plane (e.g. azimuthal projection), cylinder and cone - are considered. The mathematical formulations can be found in section 1.2.2.

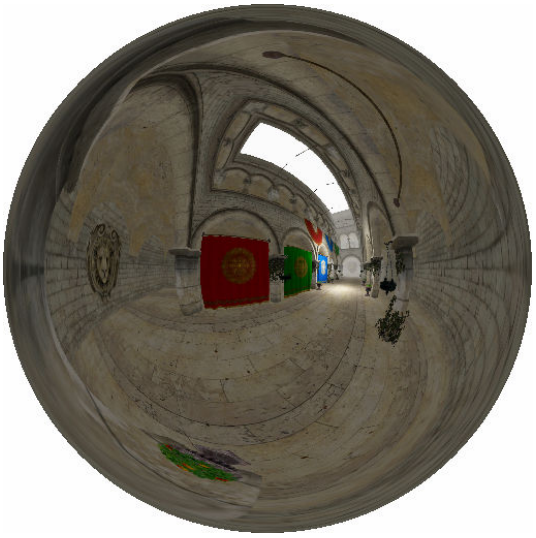
Perspective projection serves as a control condition in the experiment for future comparisons with the other projections (Figure 5.2.a). It is natively supported by graphic hardware, therefore forward rendering is used to render views using this projection. In our experiment, the vertical FoV is set to 75° (the standard upper limit found in video games). The aspect ratio of the rendering viewport determines the horizontal FoV. With our monitor using 16:10 aspect ratio, the computed horizontal FoV is about 102°.

The non-planar projections selected were:

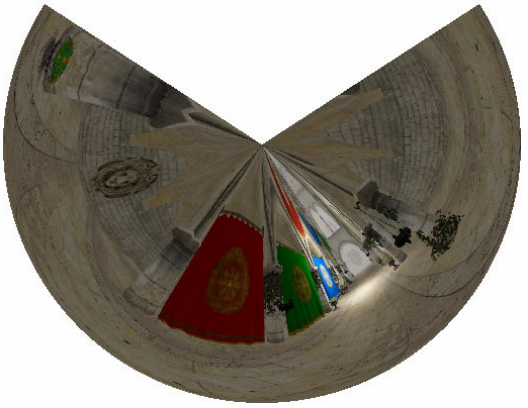
1. Equirectangular projection (Figure 5.2.b)

(a)

(b)



(c)



(d)

**Figure 5.2 –** (a) perspective projection, (b) equirectangular projection ,(c) Albers conic projection, (d) azimuthal equidistant projection.

2. Hammer projection (Figure 5.1)

3. Azimuthal equidistant projection (Figure 5.2.c)

4. Albers equal area conic projection (Figure 5.2.d)

In order to have a flexible framework, an image-based approach is adopted for rendering. Details of this method can be found in section 1.5. Therefore, the inverse functions of the non-planar projections considered are employed (see section 1.2). The whole application prototype was developed in C language and OpenGL API. For performance considerations, the non-planar projections were implemented at fragment shader level, in GLSL.

### 5.1.1 Performances

Although pure performance was not the main objective of our study, our implementation was benchmarked. The target platform is Windows 7, running on Intel i3 2120 (3.3GHz) with 4Go RAM and an NVidia GeForce 560 GTX graphic board. During the benchmark, vertical synchronization was disabled.

**Table 5.1 –** Performances of our algorithm with two different scene complexities and the five selected projections (unit: frames per second).

| Projection method | Simple VE | Rich VE |
|---|---|---|
| Perspective | 1365 | 322 |
| Equirectangular | 898 | 73 |
| Hammer | 812 | 73 |
| Albers Equal Area Conic | 914 | 72 |
| Azimuthal Equidistant | 940 | 73 |

Table 5.1 shows the observed frame rates for the different projections and two different scenes displayed in Figures 5.3 and 5.4. According to these figures, the complexity of the projection equation used has minor impact on the performance. The difference between the two scenes is explained by the overhead in geometry complexity (32 triangles in the simple VE, versus 178324 in the rich VE). The significant difference between the perspective projection and the 360° projections is directly related to the implementation of the cube map generation (the scene is rasterized 6 times in six $1024 \times 1024$ textures). This could be improved by using layered rendering technique and a proper geometry shader[Patidar 06].

In terms of FoV displayed, Figure 5.3 shows the additional visual information available compared to the standard perspective projection and to the real human binocular vision. For comparison purpose, on this figure, perspective FoV and human FoV are projected using an equirectangular projection. This explains the curvature observed on the horizontal borders of the perspective projection.


**Figure 5.3 –** Comparison of the FoV provided by 1) equirectangular projection, 2) human natural binocular vision (light blue), 3) classic perspective projection (light red) with 16:9 aspect ratio (75°×107.5°). To maintain consistency, the perspective projection is remapped into the equirectangular projection, making the horizontal limit curved.

## 5.2 Evaluation

The first objective of our evaluation was to assess if performing a task of object collection in a VE could be enhanced by providing a 360° view of the environment. We have compared the different non-planar projection methods with the traditional perspective projection providing a standard FoV of 75°×102°. The second objective was to compare the non-planar projections between each other. The quantitative evaluation was completed with a subjective questionnaire.

### 5.2.1 Population and experimental apparatus

15 participants, aged from 21 to 44, took part in the experiment. All of them had at least prior experience in video gaming.

A gamepad was used as a navigation interface with a simplified version of the common method found in video games: left analog stick is used to move forward and backward, right analog stick to rotate left and right. Strafe (lateral translation) and vertical aiming were locked. The evaluation software was running with the configuration described in section 5.1.1, completed with a 22 inches flat panel desktop monitor. The monitor provided a resolution of $1680 \times 1050$ with $16:10$ aspect ratio.

### 5.2.2 Procedure

The evaluation was split into two parts. The first part was aiming at collecting quantitative data on user performance in a box collection task. This part of the evaluation took place in a neutral VE, with only a floor textured with a noise pattern (Figure 5.4). Participants had to collect 6 boxes, one after another, with a position randomly chosen among the

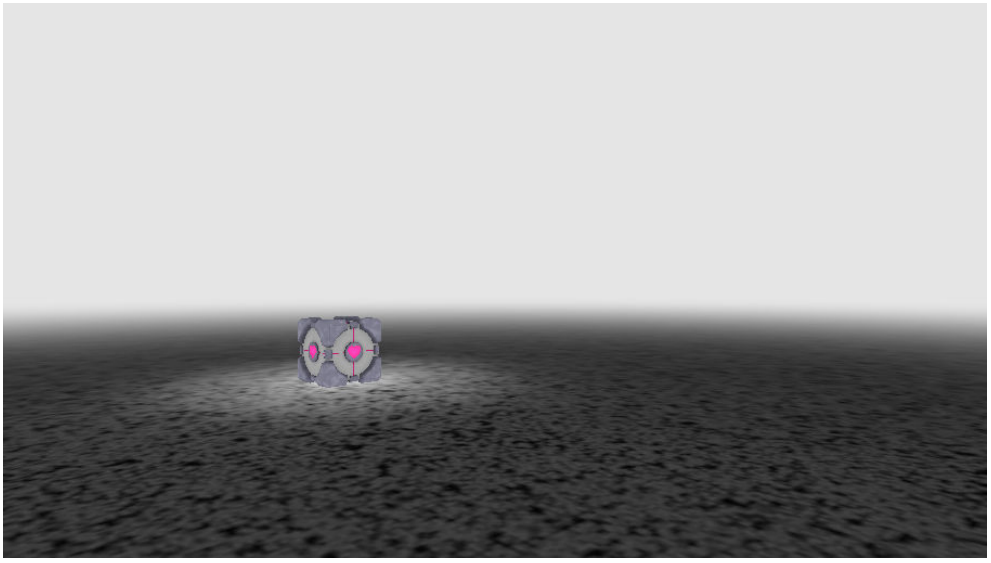


**Figure 5.4 –** Simple VE used for the basic navigation task.

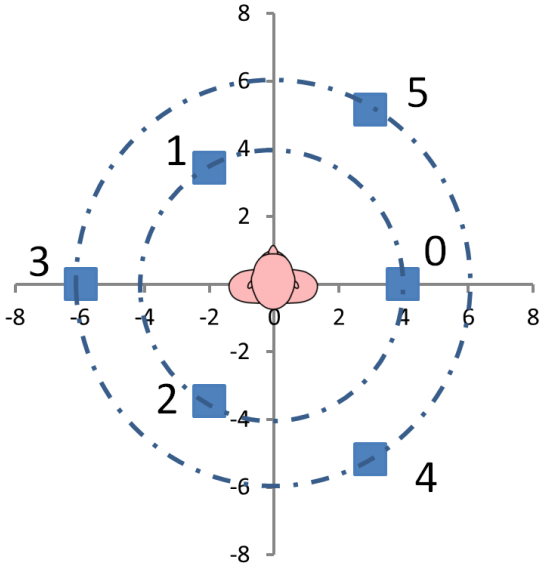



**Figure 5.5 –** Top view of boxes positions used for the object collection task. The user faces the +y axis.

location described by Figure 5.5. The 5 projection methods described in section 5.1 were randomly tested. Each box was collected 3 times leading to a combination of 90 trials for each participant (5 methods$\times$6 positions$\times$3 trials). The position of each user was reinitialized after each trial.

The second part of the evaluation consisted in a subjective evaluation of the proposed method. A richer VE was used for this part (Atrium Sponza Palace). The model was modified by removing few parts to break its symmetry (Figure 5.2.a). Participants were asked to fill in a subjective questionnaire. During the questionnaire, users were free to switch from one projection method to another.

### 5.2.3 Collected data

For the first part, the time to collect each box was recorded. For the subjective questionnaire we used 6 criteria for the 5 projection methods scored with a 7 points scale. These criteria were: (1) Visual fatigue, (2) Ease of displacement in the VE, (3) Aesthetic of the rendering, (4) Strangeness of the rendering, (5) Realism of the rendering, (6) Global appreciation.

**Table 5.2 –** Completion time results (mean completion times to reach the boxes in seconds and standard deviation).

| Projection method | all boxes | box 0 | box 1 | box 2 | box 3 | box 4 | box 5 |
|---|---|---|---|---|---|---|---|
| Perspective | 3.14 (1.39) | 3.64 (1.25) | 1.52 (0.36) | 3.30 (0.92) | 3.99 (1.59) | 4.18 (0.98) | 2.24 (0.45) |
| Equirec-tangular | 2.69 (1.56) | 2.15 (0.69) | 1.69 (1.59) | 3.08 (0.80) | 2.83 (1.14) | 3.64 (0.74) | 2.76 (2.63) |
| Hammer | 2.61 (1.38) | 2.20 (1.11) | 1.54 (0.71) | 3.16 (1.02) | 2.79 (1.90) | 3.53 (0.80) | 2.47 (1.46) |
| Conic | 3.07 (1.38) | 2.87 (2.71) | 2.09 (2.03) | 3.43 (1.12) | 3.10 (1.73) | 4.25 (1.72) | 2.66 (1.22) |
| Azimuthal | 3.04 (2.70) | 2.69 (2.14) | 1.44 (0.61) | 3.90 (1.91) | 3.44 (3.53) | 4.48 (4.17) | 2.12 (0.40) |

### 5.2.4 Results

Concerning the completion times (see table 5.2), we performed a Shapiro test that rejected the normality hypothesis on the data distribution. Thus, we used a non-parametric Friedman test to find differences among the different projections. Post-hoc comparisons were performed using Wilcoxon signed-rank tests with a threshold of 0.05 for significance. Reported p-values were adjusted for multiple comparisons. We found that the time needed to reach a box differed significantly between the 5 projections ($\chi^2 = 5.92$, $p < 0.001$).

Post-hoc analysis revealed that it was significantly faster to perform the task with the equirectangular projection compared to the perspective projection ($p < 0.001$) and the

conic projection ($p = 0.02$). The task was also performed faster with the Hammer projection than with the perspective projection ($p < 0.001$) and the conic projection ($p = 0.001$). Finally the task was performed faster with the azimuthal projection compared to the perspective projection ($p = 0.001$). Taking each box independently, we found further results. Boxes indexes are given in Figure 5.5 for reference. We found particularly significant effect for the projection condition for boxes #3 and #4 ($\chi^2 = 5.70, p < 0.001$ and $\chi^2 = 4.61, p < 0.001$ respectively). The time needed to reach the box was significantly higher for the perspective projection compared to almost all other projections ($p < 0.001$ for box #3 for all the projections, $p < 0.001$ for box #4 for equirectangular and Hammer projections, $p = 0.02$ for box #4 for azimuthal projection).

Concerning the subjective questionnaire, we performed also a Friedman test on the differences between the different projections. We found a significant effect for all the criteria: visual fatigue ($\chi^2 = 3.29, p = 0.009$), ease of displacement ($\chi^2 = 4.12, p < 0.001$), aesthetic rendering ($\chi^2 = 5.62, p < 0.001$), strangeness ($\chi^2 = 6.03, p < 0.001$), realism ($\chi^2 = 5.90, p < 0.001$) and global appreciation ($\chi^2 = 5.90, p < 0.001$).

Post-hoc analysis showed that the perspective projection was rated significantly higher than the conic projection for visual fatigue ($p = 0.008$), aesthetic rendering, strangeness, realism and global appreciation ($p < 0.001$ each time); than azimuthal projection for ease of displacement ($p < 0.001$), aesthetic rendering ($p = 0.01$), strangeness ($p < 0.001$), realism ($p = 0.002$) and global appreciation ($p = 0.002$); than Hammer and equirectangular projections for aesthetic rendering ($p = 0.005$ and $p = 0.004$), strangeness ($p = 0.03$ and $p = 0.04$), realism ($p = 0.03$ each time), global appreciation ($p = 0.03$ each time). The equirectangular projection was significantly higher rather than conic projection for aesthetic rendering ($p < 0.001$), realism and global appreciation ($p = 0.02$); and than azimuthal projection for ease of displacement ($p = 0.007$). The Hammer projection was significantly higher rated than the conic projection for strangeness ($p = 0.01$), realism ($p = 0.02$) and global appreciation ($p = 0.02$). Post-hoc analysis results are summarized in table 5.3.

**Table 5.3 –** Post-hoc analysis for the subjective questionnaire. The criterion name in a cell means that there is a significant effect between the two conditions, the best one is in the raw.

| > | Equirec. | Hammer | Conic | Azimuthal |
|---|---|---|---|---|
| Persp. | Aesthetic<br>Strangeness<br>Realism<br>Global app. | Aesthetic<br>Strangeness<br>Realism<br>Global app. | Visual fatigue<br>Aesthetic<br>Strangeness<br>Realism<br>Global app. | Ease of disp.<br>Aesthetic<br>Strangeness<br>Realism |
| Equirec. | | | Aesthetic<br>Realism<br>Global app. | Ease of disp. |
| Hammer | | | Strangeness<br>Realism<br>Global app. | |

### 5.2.5 Discussion

Overall, the equirectangular, Hammer and azimuthal projections improved the performances of the user by reducing the overall time to collect the boxes. The boxes #3 and #4, which were placed on the side of the user, and out of the field-of-view of the perspective projection method, were particularly faster to reach for the user when an omnidirectional projection was used. The Albers conic projection did not perform as well as the others 360° projection methods. The particular, pie shaped, distortions induced by this projection could explain the results. Investigation on learning time for this projection against the others could be of interest.

The results for the subjective questionnaire show that the perspective projection was globally preferred over the others. User's familiarity with this kind of projection, faced in everyday life (TV, videogames, photography...), could explain this preference as no learning effort is necessary to get comfortable with it. Among the 360° projection methods, the equirectangular and Hammer projection gather the preference of the users. Therefore, in scenarios where 360° omnidirectional vision is recommended, it would be preferable to use them.

## 5.3 Conclusion

In this chapter, we have described a preliminary user study on the influence of omnidirectional vision for a simple navigation task in virtual reality.

Our results show a significant improvement in performance (time needed to collect 3D objects in the virtual scene) when using one of the proposed omnidirectional projections. Participants were able to localize and reach targets more rapidly. Boxes positioned on the side of the user were particularly faster to reach for the user when an omnidirectional projection was used. However, the conical projection did not perform as well as the others.

Among the different omnidirectional projection methods, a subjective preference was found for the equirectangular and Hammer ones. They scored significantly better than the others for realism and global appreciation criteria and the equirectangular projection outperformed the azimuthal on ease of displacement. Such omnidirectional rendering could therefore be used in virtual reality applications in which rapid exploration is desired, or when maximum visibility is required.

The work discussed in this chapter was presented at IEEE International Symposium on 3D User Interfaces (IEEE 3DUI) 2013 conference in Orlando [PUB3].

# Conclusion

In this manuscript, we have studied the extension of the human field-of-view (FoV). The main goal of our contributions was to propose software and hardware tools in order to extend the visual capabilities of human visual system, providing more visual information by integrating a full omnidirectional view of the user's surroundings. The work was focused on two contexts: the extension of the human's FoV for real world visualization and the extension of the human's FoV for virtual environments (VE) visualization.

Three challenges' categories were identified: software, hardware and human factor. Software challenges relate to the process of data provided by acquisition devices (RE) or virtual models (VE) to generate a meaningful representation. This includes models for wide FoV and algorithms for rendering. Hardware challenges encompass the design of devices able to acquire and/or display wide FoV. The specific requirements for wide FoV impose to use non-conventional designs for such devices. Human factor challenges focus on the user perception and interaction with an extended wide FoV vision.

First, we have reviewed the related work in the extension of FoV (Chapter 1). This concerns different aspects: physiology, models, hardware and software. The physiological aspects of the human's FoV were presented. Measurements show a coverage of up to 200° horizontally and 135° vertically. However the acuity is less sharp in the peripheral area. About the mathematical aspects, the domain of cartography proposes interesting solutions for non-planar projections. Hardware devices suitable for usage with wide FoV are of two sorts: acquisition devices and display devices. Several designs were conceived to cover wide FoV, notably fisheye lenses or catadioptric sensors for acquisition devices. For display devices, two radically different approaches are used: making display wearable (HMD) letting the user look around him, or surrounding the user with display surfaces (immersive rooms). Considering the rendering methods able to provide wide FoV images, two approaches are real-time compatibles: image-based method and geometry-based methods, both of them revealing strengths and weaknesses.

Then our first contribution was discussed in Chapter 2. We have introduced a new device to increase the human's FoV, allowing a user to experience 360° vision. We have demonstrated this device by designing a proof-of-concept prototype, the FlyVIZ, based on a catadioptric system with a standard camera and a commercial HMD. The prototype forms a helmet worn by the user with a laptop computer in a backpack. The mobility of the system allows the user to freely walk in its environment. The video stream presented to the user is computed by an image processing algorithm. Is uses an equirectangular projection to transform the acquired images into images compatible with HMD screens and aspect ratio. To provide low latency and high refresh rate, our software implementation benefits from parallel processing power provided by modern GPUs. We have shown the operability of the FlyVIZ prototype in different indoor or outdoor scenarios. For example users were able to enjoy grasping an object held out behind their back without turning their head.

Our second contribution is a novel method for real-time rendering of VEs with a wide FoV (Chapter 3). Specificities of non-planar projection introduce several rendering problems, assessed differently according to the rendering method: image-based or geometry-based. Considering the geometry-based methods, approaches using per vertex projection evaluation (PVPE) fail at rendering polygons spanning across the projection discontinuities. We have proposed to include an innovative pre-clip stage in the rendering pipeline to solve this issue. This stage make possible to split the primitive in sub-primitives which are not spanning across the projection discontinuity. Therefore the sub-primitives generated are correctly rasterized. We have validated the proposed approach using modern rendering pipeline. The implementation could handle non-planar projections such as the projections classically used in cartography. Then we have extended the method to handle the computation of stereoscopic images and the use of immersion VR systems. Our method was benchmarked with standard desktop computers and graphics cards. The implementation has shown that it successfully meets real-time performance, and that it is capable of displaying a wide range of FoVs, up to 360°. Therefore, our novel approach could be used in various VR applications in which the user needs to apprehend more visual information, with an extended FoV.

The frame cancellation phenomenon was then discussed in Chapter 4. Frame cancellation is a problem generated by the conflict between two depth cues: stereo disparity and occlusion with the screen border. When this conflict occurs, the user suffers poor depth perception of the scene. It also leads to uncomfortable viewing and eyestrain due to problems in fusing left and right images. This problem is not specific to wide FoV, but can be encountered in any stereoscopic display. To solve this issue, we have proposed to use Stereoscopy Compatible Volume Clipping (SCVC), a novel method that solves the problem by rendering only the part of the viewing volume that is free of disparity/frame occlusion conflict. To demonstrate the approach, we have described an implementation, which complies even with non-programmable hardware accelerated pipeline. Then we have conducted a user study to evaluate the relevance of our method compared to the state-of-the-art technique. Results have shown that SCVC notably improved users' depth perception near the screen edges and the users expressed preference for SCVC.

In Chapter 5, we have presented a preliminary evaluation of real-time omnidirectional rendering methods for a navigation task in virtual reality. Our results have shown a significant improvement in performance (time needed to collect 3D objects in the virtual scene) when using an omnidirectional projection. The participants were able to localize and reach the targets more rapidly with a 360° lateral FoV. Among the different non-planar projection methods, the subjective preference was given to equirectangular and Hammer projections. Taken together, our results suggest that omnidirectional rendering could be used in virtual reality applications in which fast navigation or full and rapid visual exploration are required.

# Future Work

The work presented in this manuscript leaves some unanswered questions, which could be addressed in future works. We present future research possibilities of the different aspect covered just hereafter.

**FlyVIZ: a display device to increase the human field-of-view in real environments:**

- **Technical improvements.** The FlyVIZ concept and prototype can be improved and we foresee different paths for extensions and enhancements. First, other projections and mapping methods could be tested with other geometric properties, such as conformality, to preserve angles. They could influence the usability of the presented view according to a specific usage scenario. Moreover, instead of directly providing a 360° horizontal FoV, we could use a split-screen approach and different viewports on the image, such as using driving mirrors (rear and/or lateral mirrors). Other hardware components (camera and HMD) could also be tested with different aspect ratio or resolution characteristics. High dynamic range image sensors could also dramatically improve the final image quality. Then, augmented reality applications based on FlyVIZ could be proposed, for instance to improve the perception of a 360° vision with superimposed virtual cues.

- **Evaluations.** Beside these design aspects, user studies could be conducted, to evaluate the user performance for specific tasks. The amount of visual information able in this new vision could be a factor of stress and visual overload. Furthermore, spatial orientation or mental representation of the environment could be unfortunately affected.

- **Potential applications.** There are different application fields that could benefit from an enhanced FoV. In safety and security applications, firemen, policemen or soldiers could benefit from omnidirectional vision to avoid potential dangers or to locate targets more rapidly. In less critical situations, some surveillance applications with a high visual workload, in all directions of space for instance, could also be concerned, such as for aerial traffic regulation. Considering the novel perceptual experience proposed, FlyViz could also be transformed into entertaining applications and devices, as well as experimental materials for new perception and neurosciences studies.

**A real-time rendering method with wide field-of-view in virtual environments:**

- **Technical improvements.** A first field of improvement consists in the tuning of the projection function, with the objective of optimizing the depth perception. The clipping method we have proposed is designed to solve problems at cylindrical projections discontinuities. This clipping is efficient for all cylindrical and pseudo-cylindrical projections (sharing the same discontinuities). Consequently the clipping method presented cannot handle discontinuities for other family of projection (azimuthal or conical for example). However, the general approach still applies,

only the implementation of the clipping along the specific discontinuities of considered projection needs adaptation.

- **Evaluations.** An evaluation of the user performance and perception in the case of a manipulation task in the warped VE could be conducted. The user's perception of the VE and the time needed to adapt to the warped view are critical points to clarify, especially if stereoscopic rendering is used simultaneously.

**A method to solve the frame cancellation problem in real-time stereoscopic rendering:**

- **Technical improvements.** Regarding SCVC, future work concerns integration of our method with realistic rendering. Programmable graphic pipeline could be of help to refine clipped edge rendering though effects like fading or appropriate filling.
- **Evaluations.** An advanced evaluation could be conducted, using a collection of static scenes in conjunction with subjective and quantitative evaluation methods for the eye strain parameter [Stein 12]. The influence of surrounds in the Black Bands method is another point to clarify. An experimental campaign confronting different display setups (projector, monitor) and evaluation conditions (dark, bright rooms) should also be conducted.
- **Potential applications.** Thanks to its simplicity of implementation, SCVC could target a wide range of software, from VR applications to game engines and CAD software.

**Experimental evaluation of navigation in virtual environments with a wide field-of-view:**

- **Evaluations.** Future work could first focus on testing these techniques with other values of FoV (e.g. 180° laterally). Second, more evaluations of the different projection methods with other tasks (object manipulation) or contexts (3D navigation without gravity) would be of interest. Other studies on the influence of the properties of the projection (equidistant, conformal or equal area) on user perception and user's spatial cognition (wayfinding, mental map) should also be carried out.

# Long term perspectives

In Chapter 2 we have described a first prototype of the FlyVIZ. This first prototype, although it demonstrates the concept, has a level of performances that can be drastically improved. Changing parts of the assembly has already been tested in a second version of the prototype, using a smartphone as the processing device, a compact catadioptric lens and the popular Occulus rift as HMD. This second iteration of the concept has significantly improved several aspects such as the FoV of the HMD or the ergonomics and mobility. However, to push the concept further, assemblies of off-the-shelf parts are not able

to provide significant improvement in critical characteristic like latency or image definition. A device useful for outdoors use or critical situations would also require mechanical characteristics (robustness, waterproofing) that only a global and full design can bring. A way to decease the global latency of the system is through the design of specific electronic hardware, using FPGA (Field Programmable Gate Array) in a development phase then ASIC (Application Specific Integrated Circuit). These dedicated circuits would allow suppressing several buffers in the flow (acquisition/process/display) by addressing the image sensor and the display panel directly. Furthermore, the projection equation could be hardwired or stored in a look up table. At the same time the specifically designed electronic could allow a true high definition, in the sense that the final image would be at least at human eye acuity level. This is fundamental as the system is by nature decreasing the acuity, by mapping a large field to a lower one which fits in the human FoV.

Another direction for the long term research concerns human factors. The work presented here gives preliminary results, and our evaluation was focused on a navigation task, in the context of virtual reality. The user interaction with VE through a wide FoV vision raises several other questions. For example, how does the user perform with object manipulation task? As evoked with real environment with FlyVIZ, the user's visual feedback loop is modified. The perception of all the body is altered: for example when raising arms laterally (T-pose), the user see his how arms pointing forward. Clearly, omnidirectionnal vision has strong impact on proprioception; therefore a global study of human factors should integrate proprioceptive aspects. When wearing the device, one can feel a progressive adaption to the new visual feedback. Intuitively, they are cognitive mechanisms to study behind this adaptation. In VE, depending on the display used, this alteration can be present or not: immersive rooms let the user perceive his body as usual, while a HMD could allow to give an image of the user body transformed by the projections used. A study could use this asymmetry to investigate cognitive mechanisms involved in the adaptation to the extended FoV vision.

The extension of human FoV is an ambitious and exciting topic. Future improvement in the involved technologies could bring a lot to the extension of the human field-of-view. The feedbacks we have received during the presentation of FlyVIZ were very encouraging, from academics, professionals and general public. Furthermore, the large set of potential applications let foresee a promising future for the extension of FoV. We hope the work presented in this manuscript will contribute to the maturation of this technology, from the technical point of view as well as human factor considerations.

# Appendix: résumé long en français

# A

Contents



Cette thèse de doctorat propose d'étudier l'extension du champ visuel humain dans un environnement naturel et dans des environnements virtuels.

L'évolution des espèces a donné naissance à un large panel d'animaux avec différents champs de vision. Les proies ont tendance à avoir des champs de vision relativement grand afin de localiser au plus tôt un prédateur potentiel. Au contraire, les prédateurs ont un champ de vision réduit avec une grande acuité, les rendant capables de localiser leurs proies de loin et en scrutant leur environnement. De nos jours, il est possible d'augmenter la vision d'une personne à l'aide de divers instruments. Par exemple la vision de nuit est réalisée grâce à des lunettes qui amplifient la lumière. La vision lointaine peut également être améliorée, augmentant l'acuité, grâce à des instruments optiques comme des jumelles. La technologie de ces instruments a tellement progressé qu'il est même possible pour l'humanité d'observer les confins de l'univers à travers des télescopes. Dans

de telles situations, l'acuité est développée au détriment du champ visuel. Parallèlement, l'extension du champ de vision n'a pas bénéficié d'améliorations comparables et à ce jour il n'existe pas d'instrument optique en mesure d'augmenter le champ visuel de l'homme.

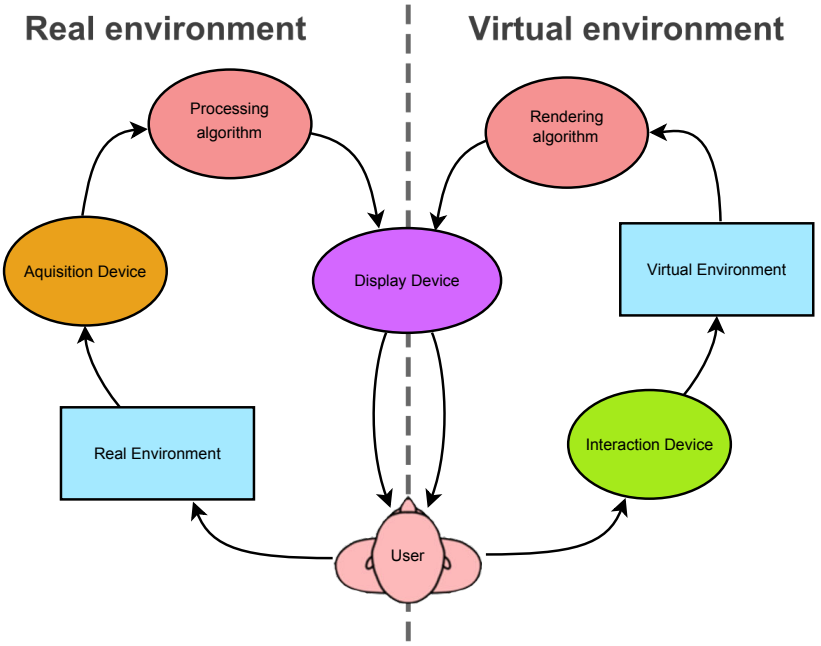



**Figure A.1 –** Structure nécessaire à l'augmentation du champ visuel humain. Dans le contexte de l'environnement réel (à gauche), et dans le contexte d'environnements virtuels (à droite).

La Figure A.1 décrit les composants et l'architecture des systèmes nécessaires pour générer un champ visuel étendu. On y retrouve les boucles fermées en oeuvre lorsqu'un utilisateur interagit avec le monde réel ou dans des environnements virtuels. Dans le cas d'un environnement virtuel (partie droite de la Figure A.1), l'utilisateur interagit avec l'environnement en utilisant des dispositifs d'interaction, puis une nouvelle représentation de l'environnement virtuel est calculée avec un algorithme de rendu, reflétant les changements effectués par l'utilisateur (navigation, l'interaction avec les objets virtuels, etc). Ensuite, cette nouvelle représentation est présentée à l'utilisateur grâce à un dispositif d'affichage. L'ensemble de ce processus doit être effectué suffisamment vite pour se conformer aux contraintes du temps réel.

# Les défis liés à l'extension du champ visuel humain

Les défis pour accroître le champ de vision humain sont de trois types : les défis logiciels, les défis matériels et les défis liés aux facteurs humains. Certains défis sont transverses aux contextes (ils ne dépendent pas des spécificités des environnements réels ou virtuels). Ils sont illustrés dans la Figure A.2.

Les défis logiciels résident dans les algorithmes capables de traiter des données fournies par un dispositif d'acquisition (cas de l'environnement réel) ou des modèles virtuels pour générer une représentation intelligible. Concrètement, ces algorithmes couvrent les outils nécessaires à la modélisation des caméras et autres dispositifs d'acquisition omnidirectionnels ainsi que des algorithmes nécessaires pour rendu d'image d'environnement

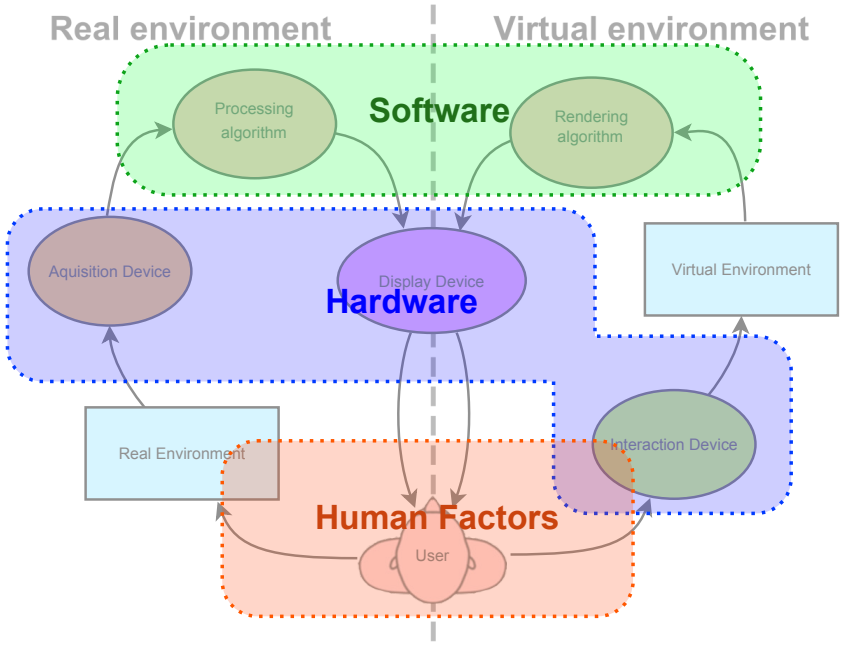



**Figure A.2 –** Les grandes catégories des défis liés à l'extension du champ visuel humain : le matériel, le logiciel et les facteurs humains.

virtuel. Les outils mathématiques fondamentaux liés à ces défis sont les projections non planaires. Ce type de projection est étudié depuis des siècles par les cartographes. Cependant de problèmes se posent quand il s'agit de les utiliser avec des algorithmes de traitement ou de synthèse d'images en temps réel.

Les défis matériels résident dans la conception de dispositifs capables d'étendre le champ visuel d'une personne. Trois types de matériels sont impliqués : les dispositifs d'acquisition, des dispositifs d'affichage et les dispositifs d'interaction. Un dispositif d'acquisition omnidirectionnel doit capturer des informations en provenance d'angles très larges. Dans ces conditions, les approches optiques classiques ne conviennent pas. Les technologies d'affichage nécessitent également une conception particulière tant au niveau des surfaces d'affichage que de l'électronique pour garantir une définition correcte. Pour finir, les dispositifs d'interaction utilisés ont besoin d'un espace de travail compatible avec surface d'affichage utilisée.

Les défis en terme de facteurs humains consistent à comprendre comment les utilisateurs perçoivent et interagissent lorsque leur champ visuel est augmenté. Le système visuel humain est complexe et la façon dont les informations provenant des yeux sont traitées par le cerveau est un sujet de recherche actif en neurosciences. Comme les autres sens, la vision est développée depuis la naissance. Une modification importante du champ visuel soulève donc plusieurs questions : est-ce que l'utilisateur sera en mesure d'utiliser un champ visuel plus grand que son champ visuel naturel ? Peut-il apprendre à utiliser cette vision particulière ? Est-il en mesure de profiter de cette quantité d'informations visuelles supplémentaires ? Y a-t-il des effets secondaires induits par une telle vision ?

En relation avec ces défis, les contributions proposées dans ce manuscrit sont présentées ci-dessous.

# Approche et contributions

La Figure A.3 illustre les différentes contributions de cette thèse.

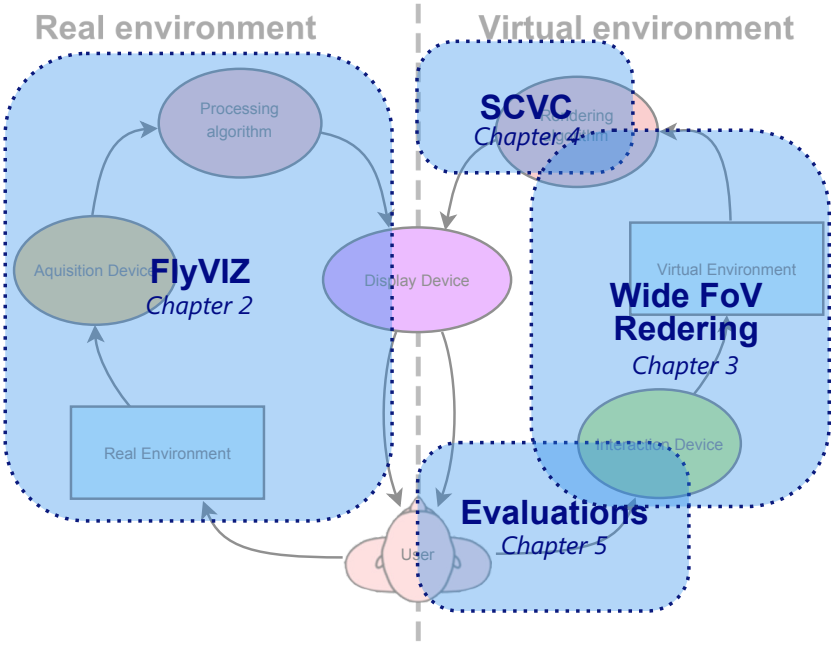



**Figure A.3 –** Les principales contributions de cette thèse de doctorat.

### Première contribution : la conception d'un dispositif capable d'augmenter le champ visuel humain

Un premier objectif consiste en la conception d'un dispositif capable d'augmenter le champ visuel humain. La conception d'un tel dispositif se heurte à différents problèmes : une approche uniquement optique ne semble pas réaliste. Même en considérant l'état de l'art des techniques de conception et fabrication des objectifs photographiques, il paraît compliqué de construire des lunettes capables de collecter, conduire et faire converger la lumière venant de derrière un utilisateur. Le dispositif que nous avons mis au point afin de réaliser cet objectif a été nommé FlyVIZ, en référence à la vision des mouches, capables de percevoir une information visuelle venant de derrière elles. Il est basé sur l'association d'une caméra catadioptrique, d'un logiciel de traitement d'image et d'un visiocasque.

### Deuxième contribution : la conception d'un méthode de rendu d'image pour l'affichage d'environnements virtuels avec un champ visuel étendu

Notre deuxième contribution concerne l'extension du champ visuel lors de la visualisation d'environnements virtuels. Les algorithmes de rendu temps réels modernes utilisent le principe de rasterisation. Ils sont étroitement lié aux processeurs graphiques et conçus pour fonctionner, historiquement, avec des projections perspectives (planes). Cela a permis d'optimiser les temps de calcul, essentiel pour que les algorithmes restent compatibles avec les contraintes du temps réel. Cependant, les équations mathématiques qui décrivent les projections non-planaires sont plus complexes que la projection perspective, générant ainsi des problèmes particuliers. Pour traiter ces problèmes nous proposons

une nouvelle méthode pour le rendu en temps réel des environnements virtuels avec un champ visuel étendu. La méthode proposée est également compatible avec un rendu stéréoscopique. La stéréoscopie est une autre caractéristique importante pour le rendu des environnements virtuels. En fournissant des images différentes à l'oeil gauche et à l'oeil droit, la stéréoscopie permet d'augmenter de manière significative la perception des profondeurs et le sentiment d'immersion. D'une manière générale, le rendu d'images stéréoscopiques soulève de nombreux problèmes de perception. En particulier, on peut se retrouver confronté à des conflits entre différents indices de profondeurs. Plus spécifiquement, un conflit apparaît quand un objet est affiché devant l'écran (parallaxe négative) et qu'il est partiellement occulté par le bord de l'écran. Pour résoudre ce problème, nous proposons une méthode innovante appelé Stereo Compatible Volume Clipping (SCVC), ou le découpage des objets selon le volume qui est compatible avec un affichage stéréoscopique.

### Troisième contribution : les facteurs humains liés à l'extension du champ visuel et évaluations préliminaires dans le contexte des environnements virtuels

L'extension du champ visuel soulève de nombreuses questions vis à vis des facteurs humains en jeu. Comment une vision omnidirectionnelle influence-t-elle la navigation dans un environnement virtuel ? Est-ce que la performance de l'utilisateur pour une tâche spécifique est modifiée ? Une projection non plane est-t-elle mieux adaptée à un certain type de scénario ? Notre troisième contribution s'intéresse à la perception des environnements virtuels avec un champ visuel étendu. Nous avons évalué la performance d' utilisateurs pour une tâche de collecte d'objets. La projection perspective classique a été confrontée à plusieurs méthodes de projection non-planaires.

## A.1 FlyVIZ : un dispositif pour augmenter le champ visuel humain dans l'environnement réel

Ce chapitre décrit la conception et le développement d'un dispositif d'affichage capable d'augmenter le champ visuel humain. Ce nouveau dispositif (voir Figure A.4), baptisé FlyVIZ, est basé sur trois éléments: (1) un système d'acquisition, (2) un visiocasque et (3) un algorithme de traitement d'image.

Le concept général de FlyVIZ, ses composants ainsi que le prototype développé sont présentés ci-après. Les performances ainsi que les usages potentiels d'un tel système sont abordés en suivant.


**Figure A.4 –** Le prototype FlyVIZ. Il combine une caméra omnidirectionnelle et un visiocasque

### A.1.1 Le concept FlyVIZ et le prototype

L'objectif de FlyVIZ est d'augmenter le champ de vision humain, le but étant d'atteindre un champs visuel de 360°. Ce type de vision permet de voir simultanément devant soi et dans son dos, comme certains animaux, les chevaux ou les mouches par exemple. Cette caractéristique est difficile à obtenir à l'aide des dispositifs optiques classiques. Cependant, une approche incluant un système d'acquisition panoramique, un traitement logiciel d'image et un dispositif d'affichage est envisageable.

La réalisation de trois fonctions principales pilote la conception d'un tel système (Figure A.5). D'abord, le système doit capturer l'information visuelle de l'environnement de

l'utilisateur. Le système doit transformer cette vision en une représentation intelligible. Enfin, le système doit mettre cette vue à disposition de l'utilisateur.

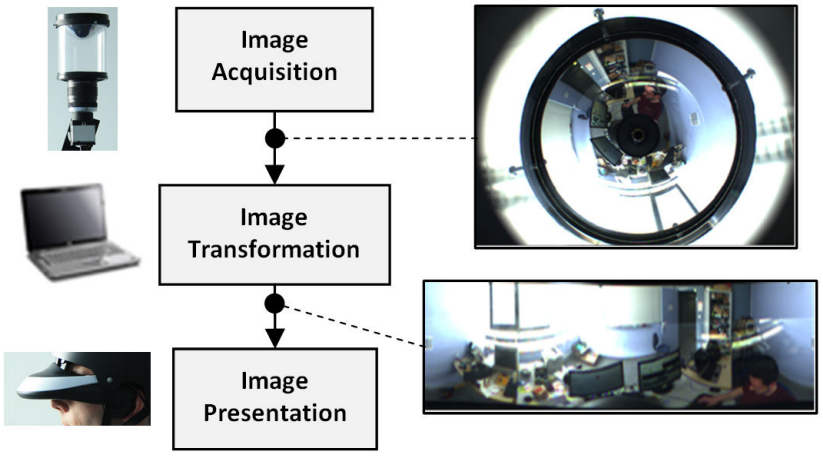



**Figure A.5 –** FlyVIZ components

## A.1.2 Traitement des images

Afin de créer une image cohérente, il faut établir la relation entre un pixel affiché dans le visiocasque et le pixel correspondant dans l'image brute fournie par la caméra catadioptrique. Cette relation s'établit à l'aide de deux projections :

**1** La projection que l'on souhaite utiliser pour la représentation finale, elle établit une correspondance entre une coordonnée dans l'image finale et une direction dans l'espace ;

**2** La projection lié à la caméra catadioptrique utilisée, qui établit la correspondance entre une direction de l'espace et une coordonnée dans l'image brute capturée par la caméra catadioptrique.

On peut exprimer de façon mathématique ces relations :

$$(x,y) \xrightarrow{P} (\lambda,\phi) \xrightarrow{S} (X,Y,Z) \xrightarrow{M} (x',y') \tag{A.1}$$

En pratique, la première étape de l'algorithme consiste à transformer un pixel de coordonnées $(x,y)$ en $(\lambda,\phi)$ grâce à une projection équirectangulaire $P$. Le vecteur $(X,Y,Z)$ ensuite déduit de la forme paramétrée de la sphère unitaire $S$.

$$S\begin{cases} X &= \sin\phi\cos\lambda \\ Y &= \sin\lambda \\ Z &= -\cos\phi\cos\lambda \end{cases} \tag{A.2}$$

Pour finir la coordonnée correspondante dans l'image capturée $(x',y')$ est calculée a partir de $(X,Y,Z)$, en utilisant le modèle mathématique de la caméra $(M)$ et son calibrage [Scaramuzza 06a].

(1) (2)

**Figure A.6 –** Scénarios d'utilisation : (1) saisir un objet hors du champ visuel naturel, (2) éviter une balle lancée de derrière.

### A.1.3 Utilisation du système

Le prototype FlyVIZ est parfaitement opérationnel. Il a été testé par plusieurs utilisateurs dans différentes conditions, en intérieur et en extérieur (Figures A.6 et A.7). Pendant les tests, les utilisateurs s'adaptaient en moyenne en une quinzaine de minutes au système. Ils arrivaient à appréhender la nouvelle vision de leur propre corps, notamment les mains et bras, et pouvaient ainsi saisir des objets ou ouvrir des portes. La perception des distances est altérée en raison de la vision monoculaire, mais les utilisateurs semblent compenser par les autres indices visuels (comme déjà suggéré dans d'autres situations par Cutting [Cutting 97]).

Un premier scénario consiste à saisir un objet présenté dans le dos de l'utilisateur. Sans bouger la tête, ce dernier est capable d'évaluer la position de l'objet et de s'en saisir. Un second scénario consiste à marche et à essayer d'esquiver une balle lancée par l'arrière. Pour finir, un troisième scénario consiste à conduire une voiture sur un parking. Dans ce dernier scénario, il est intéressant de remarquer que l'utilisateur est capable de visualiser simultanément l'extérieur du véhicule et l'intérieur de l'habitacle, banquette arrière comprise (Figure A.8).

Durant les différents tests, le dispositif a pu être porté pendant plus d'une heure, sans que l'utilisateur ne soit sujet au mal de simulation, à des nausées ou à une fatigue visuelle particulière. Le principal défaut rapporté est un inconfort ressentit pendant le port du casque, principalement du à la mauvaise répartition des masses du prototype.

**Figure A.7 –** Image affichée dans le visiocasque, correspondant à l'image transformée.



**Figure A.8 –** Scénario d'utilisation : conduire une voiture sur un parking.

## A.2 Une méthode de rendu des environnements virtuels, en temps réel et avec un champ visuel étendu

Les méthodes évaluant une projection non-planaire par vertex sont des méthodes très intéressantes pour le rendu d'environnements virtuels car elles sont compatibles avec le pipeline graphique standard utilisé par les processeurs graphiques modernes. Cependant elles possèdent deux faiblesses:

- Premièrement, une projection non planaire ne préserve pas les formes (en général, les lignes droites deviennent courbes après projection). Ceci entraine une déformation importante de la géométrie après projection si la scène n'est pas suffisamment subdivisée.
- Deuxièmement, une projection non planaire comporte des discontinuités : si une primitive dessinée chevauche une discontinuité, elle sera mal rastérisé (Figure A.9 et Figure A.10).

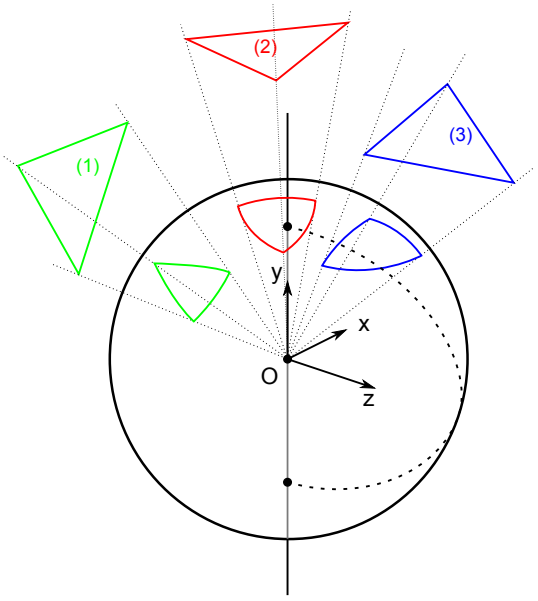



**Figure A.9 –** Problèmes rencontrés avec une projection équirectangulaire. La discontinuité de projection est représentée par la ligne en pointillé. Le triangle (1) est compatible avec le rendu traditionnel alors que les triangles (2) et (3) sont mal restitués.

Pour résoudre ces problèmes, nous proposons de modifier le pipeline graphique, pour inclure une nouvelle étape dans le traitement de primitives dessinées. La Figure A.11 illustre cette approche.

### A.2.1 La nouvelle étape de pré-découpage des primitives

Certaines méthodes proposent de résoudre les problèmes générés par les triangles chevauchant une discontinuité de projection en évitant simplement de les afficher [Petkov 12]. Comme

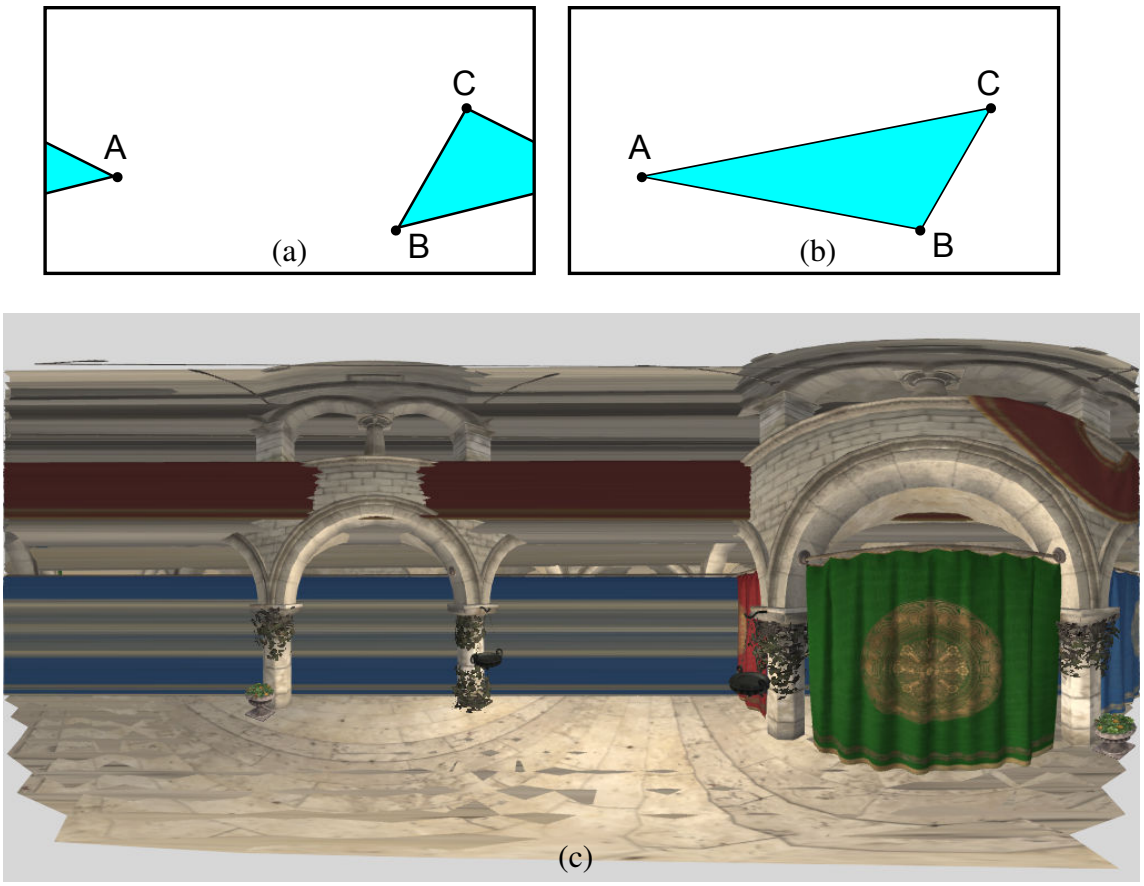



**Figure A.10 –** Problème de rasterisation liés aux discontinuités de projection. (a) le rendu souhaité dans la vue finale. (b) la surface obtenue avec le pipeline standard. (c) le problème illustré dans une scène réaliste : les rideaux rouge et bleu sur la droite sont rasterisés de façon incorrecte et créent des artefacts horizontaux sur l'image finale.

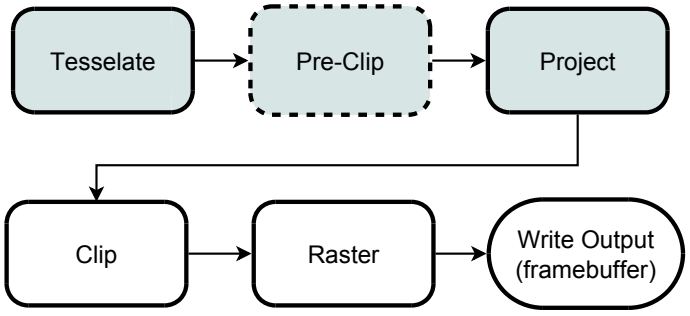



**Figure A.11 –** Le pipeline graphique proposé pour le rendu d'image avec un champ visuel augmenté en temps réel. Les étapes grisées sont implémentées avec des shaders. La nouvelle étape de découpage des polygones est représentée en pointillé.

alternative, nous proposons un traitement spécifique de ces triangles en les découpant afin d'obtenir un ensemble de plusieurs triangles qui ne chevauchent plus les discontinuités et qui peuvent donc être affichés correctement. En pratique, en considérant une projection équirectangulaire, il y a deux situations dans lesquelles un triangle peut chevaucher une discontinuité de projection :

- *Triangle sur un pôle* : le triangle est traversé par l'axe *y*;
- *Triangle arrière* : le triangle est traversé par le demi-plan arrière *Oyz*.

la Figure A.12 décrit comment découper ces deux types de triangle. Un fois découpés, les polygones sont correctement rasterisés, on peut observer le résultat sur la Figure A.14 où les polygones traités sont colorés pour être mis en évidence.

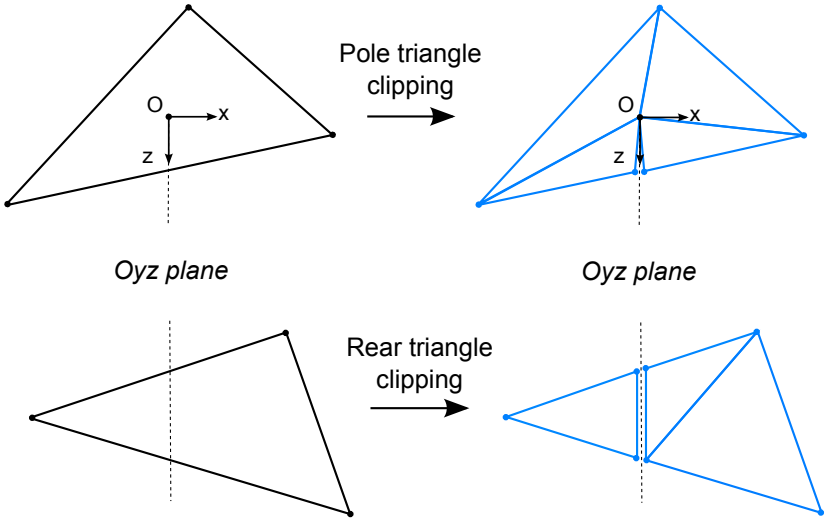



**Figure A.12 –** Méthodes de découpage. Partie supérieure de la Figure : triangle traversé par l'axe *y*. Partie inférieure : triangle traversé par le demi-plan arrière *Oyz*

L'approche globale fonctionne avec différentes projections non planaires sous réserve d'adapter le découpage des polygones pour qu'il corresponde aux discontinuités de la projection non-planaire considérée. Cependant certaines projections partagent les mêmes discontinuités, une implémentation donnée peut donc fonctionner avec une famille complète de projections. Dans notre cas, l'implémentation fonctionne aussi bien avec une projection équirectangulaire qu'avec une projection de Hammer (Figure A.13)

### A.2.2 Calcul d'images stéréoscopiques cohérentes et adaptation au systèmes immersif

Pour calculer la paire d'images nécessaires à une vision stéréoscopique, l'idée est de considérer le volume après projection comme un objet et de calculer la paire d'images en s'appuyant sur les méthodes standards. Deux projections perspectives supplémentaires sont donc utilisées, comme présenté sur la Figure A.15. Pour les caractériser, un premier plan est positionné, définissant la distance à laquelle la parallaxe est nulle (les objets à cette distance ont la même position dans les images gauche et droite). Les deux centres de projections (CoP) sont positionnés pour correspondre à la distance inter oculaire

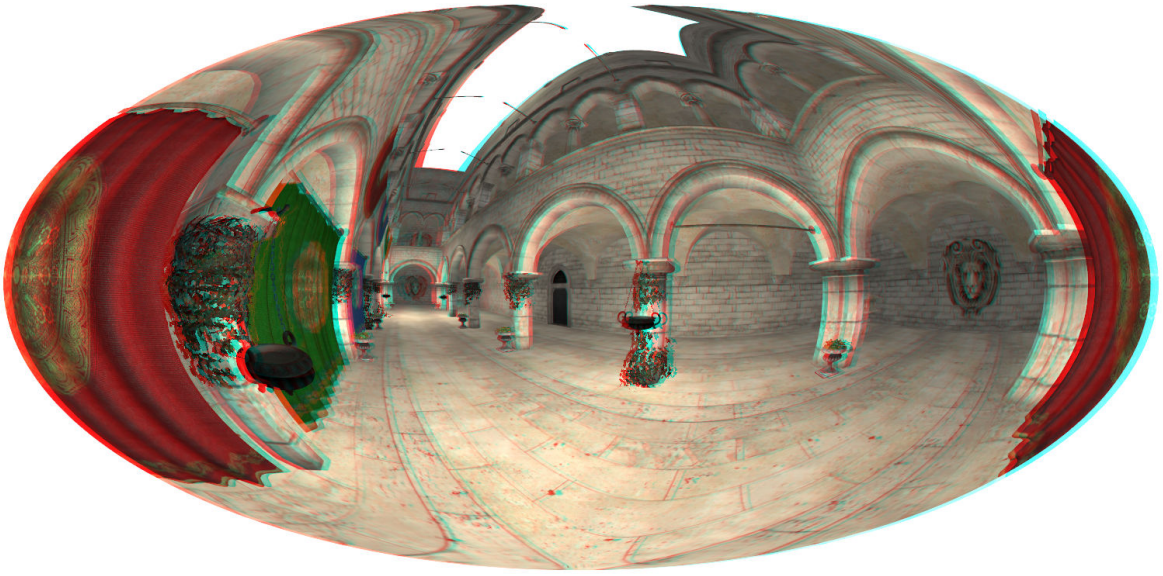


**Figure A.13 –** Rendu stéréoscopique d'un environnement virtuel couvrant 360° horizontalement. Le rendu utilise une projection de Hammer, souvent rencontrée en cartographie. L'illustration utilise une séparation stéréoscopique anaglyphe, et doit être visualisée avec des lunettes à filtres rouge/cyan.

**Figure A.14 –** Environnement virtuel rendu avec une projection équirectangulaire. Les triangles traités et découpés sont mis en évidence avec des couleurs.

souhaitée. Pour finir, comme dans un rendu classique, des plans de clipping proches et lointains sont fixés pour générer des valeurs compatibles avec le Z buffer. Le rendu final peut être ajusté selon 3 paramètres : la distance inter-oculaire (*iod*), la distance au volume projeté ($d$) et la position du plan de parallaxe nulle dans le volume projeté (*offset*). Après quelques tests subjectifs, nous avons choisi les valeurs suivantes: $iod = 0.5$, $d = 5$ et $offset = -0.8$ qui semblent fournir un effet convaincant tout en conservant une visualisation confortable. Ces valeurs sont données sans unité, dans le repère normalisé (normalized device coordinate frame). La Figure A.16 montre les résultats obtenus dans avec une projection équirectangulaire et différentes valeurs de champs visuels.

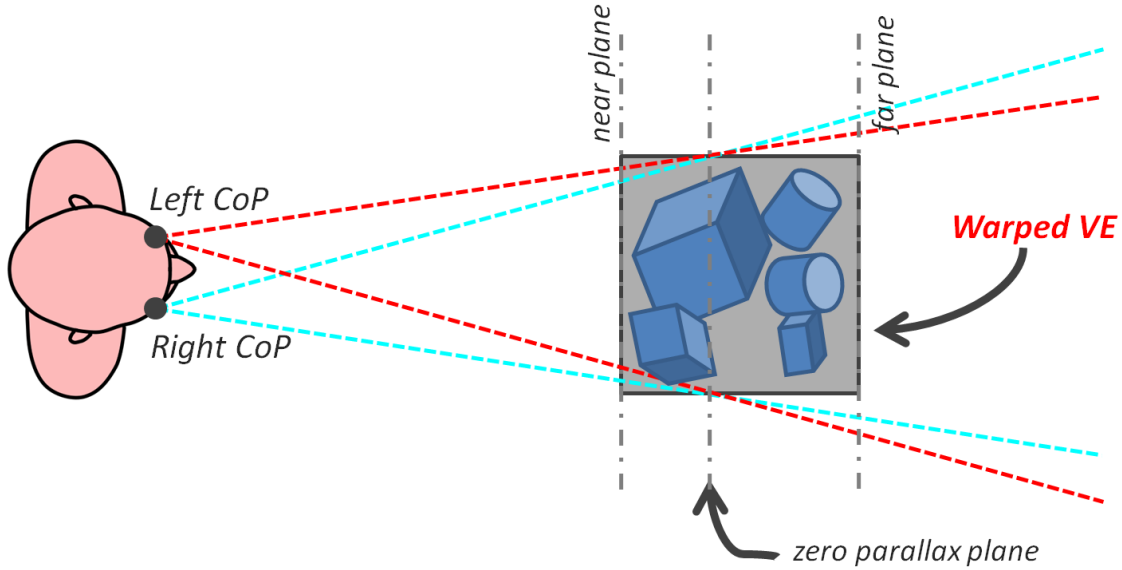
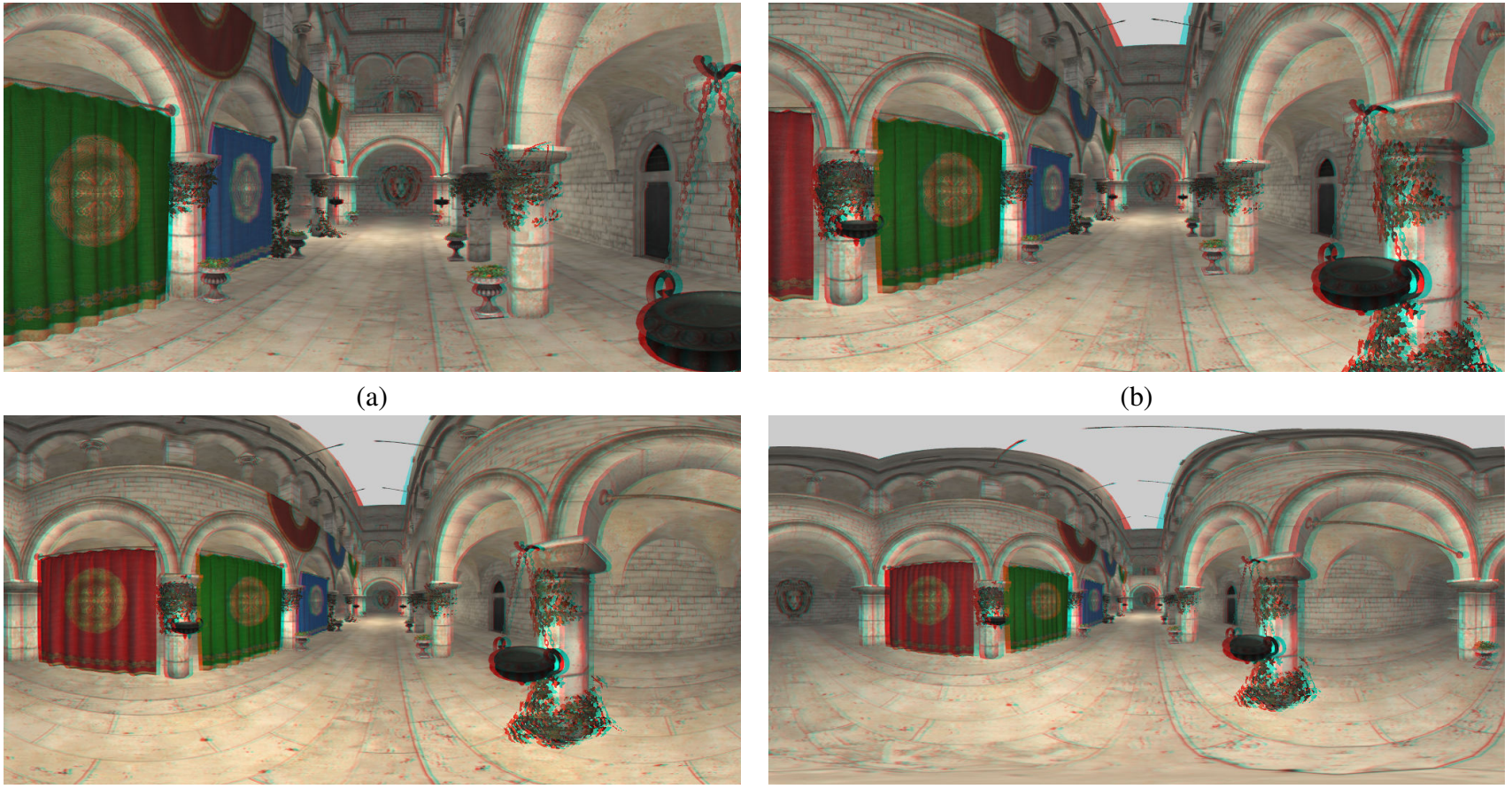



**Figure A.15 –** Rendu stéréoscopique avec des projections non planaires. Après projection, la géométrie de l'environnement virtuel peut être vue comme étant pliée et déformée pour rentrer dans le cube unitaire. L'environnement est ensuite rendu grâce à l'utilisation de deux projections perspectives supplémentaires. Ces projections sont définies par deux centres de projection, un plan de clipping proche, un plan de clipping lointain et un plan déterminant une distance où la parallaxe est nulle.



(a) (b)

(c) (d)

**Figure A.16 –** Rendu stéréoscopique avec une projection équirectangulaire et différentes valeurs de champ visuel : (a) 120°, (b) 180°, (c) 270° et (d) 360° (images anaglyphes).

## A.3 Une méthode pour résoudre le problème de 'frame cancellation' dans le cadre d'un rendu stéréoscopique temps-réel

Le problème de 'frame cancellation' provient du conflit d'indice de profondeur entre la disparité stéréoscopique et l'occultation par les bords de la surface d'affichage (abrégé conflit D/OB ci-après). Ce problème peut se produire dans toute image stéréoscopique qui affiche un objet en parallaxe négative (c'est à dire devant le plan physique de l'image) et partiellement occulté par le bord du support. Ce chapitre propose une nouvelle méthode pour résoudre ce problème. Elle se base sur l'observation de la forme du volume affiché où la stéréoscopie est effective. Pour résoudre le conflit, l'approche proposée n'affiche que la partie du volume qui ne risque pas souffrir du problème. Ce volume est appelé SCV (Stereo Compatible Volum, volume compatible avec la stéréoscopie) et la méthode SCVC (Stereo Compatible Volume Clipping, découpage selon le volume compatible avec la stéréoscopie).

La Figure A.17 illustre la configuration typique pour les volumes gauche et droit affichés en fonction d'une surface d'affichage déterminée. L'intersection de ces deux volume n'est pas sujette au conflit de D/OB, alors qu'un objet affiché dans le reste du volume peut engendrer le problème. Le volume situé en parallaxe positive n'est pas représenté, celui-ci n'étant jamais sujet au conflit D/OB.

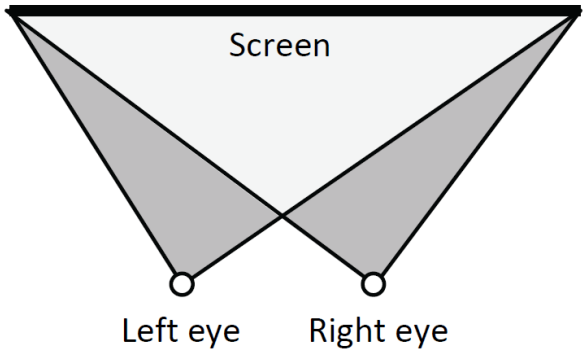



**Figure A.17 –** Volumes visualisés en stéréoscopie. Gris clair: le SCV. Gris foncé : le volume sujet au conflit D/OB.

### Définition et approche

Dans le demi-espace où la parallaxe est négative (c'est à dire devant le support d'affichage), le SCV est défini comme l'intersection entre le volume visible de l'oeil gauche et celui de l'oeil droit (voir Figure 4.1). D'un point de vue géométrique, on peut donc résoudre le problème de conflit V/BS de façon optimale en affichant que le SCV.

Cette approche n'essaye pas d'éviter les situations qui entrainent un conflit. Elle ne dessine simplement que la partie du volume affichable qui n'y est pas sujet. Elle est donc

indépendante de la scène ou des modèles 3D affichés, et elle ne se base pas sur l'analyse de la disparité. Aucun traitement particulier de la scène n'est donc requis.

#### Implementation

Le découpage de polygone (clipping) est un procédé classique pour éviter de dessiner la partie de la géométrie qui est en dehors du volume visualisé. Les API 3D modernes donnent la possibilité d'ajouter des plans de clipping arbitraires lors du rendu 3D. Pour dessiner uniquement le SCV, il suffit d'ajouter le plan de clipping gauche de la projection perspective de l'oeil droit lorsque l'on dessine la vue de l'oeil gauche et vice versa. Il est bon de noter que ce procédé n'altère en rien le demi espace de parallaxe positive. L'implémentation est relativement aisée avec l'API OpenGL. Les équations des plans de clipping sont simplement calculées à partir de 3 points: le centre de projection, et les points correspondants sur le plan de projection (Figure A.18).

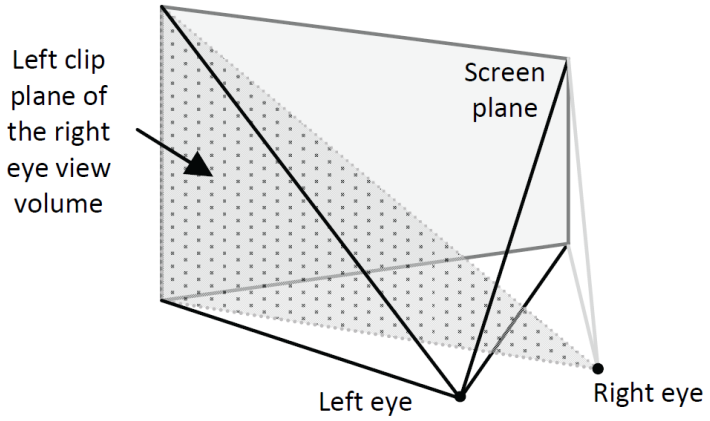



**Figure A.18 –** Plans de découpe de SCVC. Représentation perspective du volume visualisé par l'oeil gauche et du plan de découpe gauche de l'oeil droit. Ce plan de découpe doit être utilisé pendant le rendu de l'oeil gauche.

### A.3.1 Evaluation

L'efficacité de la méthode a été mise à l'épreuve dans une évaluation. Trois conditions ont été comparées : SCVC, les bandes noires (une méthode éprouvée de l'état de l'art) et la condition de contrôle (pour laquelle aucune méthode n'est employée pour résoudre les conflit D/OB). 22 participants ont prit part à l'expérience qui consistait à visualiser une sphère se déplaçant de gauche à droite sur un écran avec une vision stéréoscopique (voir la Figure A.19). Diverses questions étaient ensuite posées aux utilisateurs afin de recueillir leurs appréciations des différentes méthodes.

Après traitement statistique et analyse, les résultats ont permis de confirmer l'amélioration de la perception de la sphere au voisinage des bords de l'écran. Un bémol est à noter cependant en ce qui concerne l'esthétique : certains sujets ont mis en avant l'aspect "coupé" de la sphère. Selon l'utilisation, cela peut représenter un réel problème, si l'on souhaite un rendu réaliste par exemple. Une piste d'amélioration sur ce point pourrait être de faire disparaître progressivement l'objet, ou de combler le plan de découpe avec

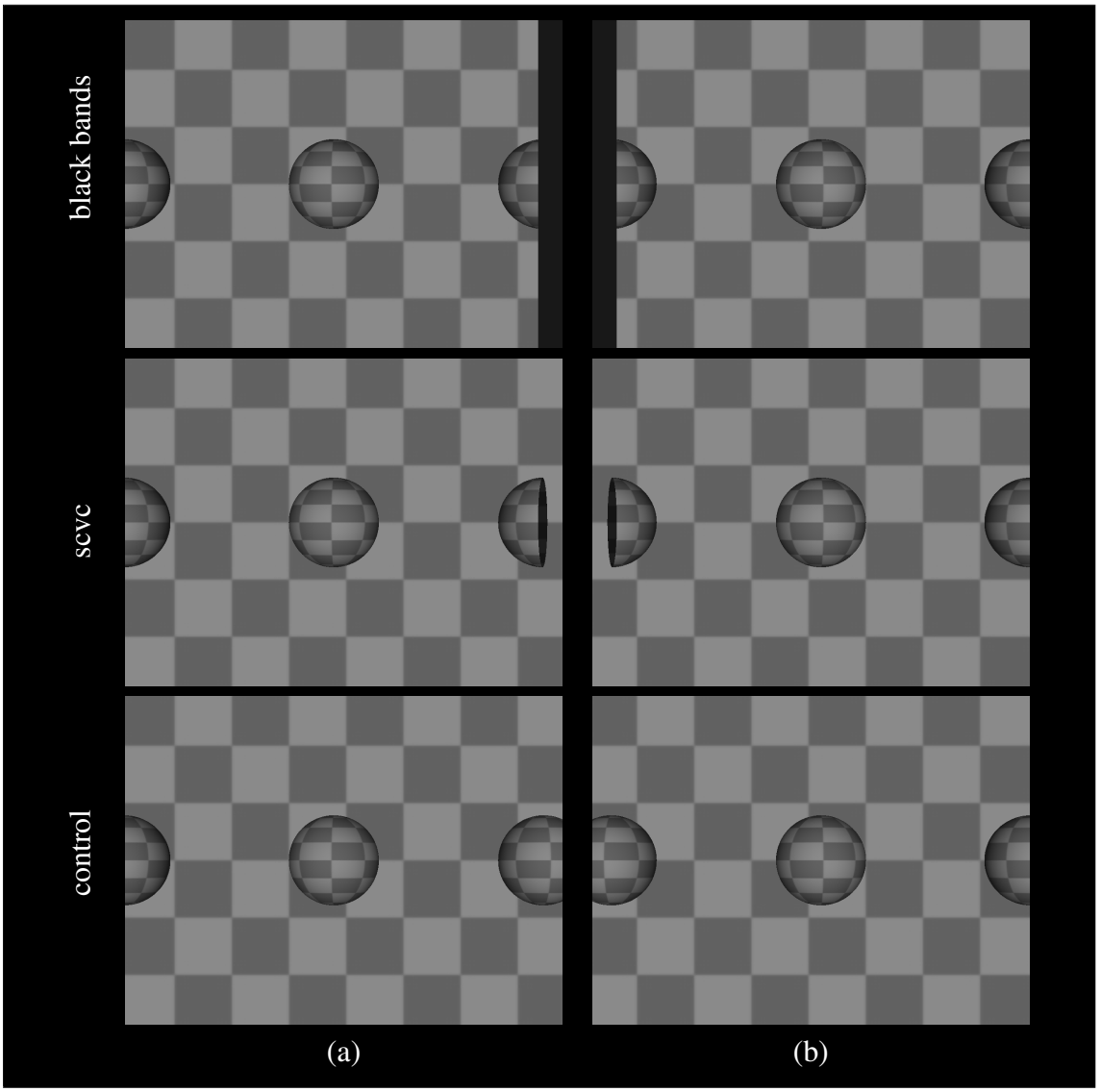



**Figure A.19 –** Images stéréoscopiques dans les trois conditions expérimentales. Les bandes noires (méthode de l'état de l'art), SCVC et la condition de contrôle (sans traitement particulier). Un lecteur confortable avec la stéréoscopie croisée (cross eye stereo) peut fusionner les images par strabisme afin d'observer les différences.

une couleur solide. La méthode tout de même utilisable en l'état pour des applications de conception assistée par ordinateur par exemple où la lisibilité prend le pas sur l'esthétique.

## A.4 Évaluation expérimentale de la navigation dans un environnement virtuel avec un champ visuel étendu

Ce chapitre a pour objectif d'évaluer la navigation dans un environnement virtuel avec un champ visuel étendu. Nous avons confronté une projection perspective offrant un champ visuel standard (102° horizontal) à quatre projections non-planaires qui ont fait leurs preuves dans le domaine de la cartographie. Ces projections permettent de couvrir un champ visuel de 360° et possèdent différentes propriétés leur permettant de préserver les distances ou les aires après projection. les formulations mathématiques sont décrites dans la section 1.2.2 et ne sont donc pas rappelées ici.

**Figure A.20 –** Rendu omnidirectionnel couvrant 360° en utilisant une projection de Hammer.

La projection perspective est la condition de contrôle fournissant un champ visuel standard : 75° verticalement et 102° horizontalement (voir Figure A.21.a).

Les projections non planaires retenues sont les suivantes :

1. la projection équirectangulaire (Figure A.21.b)
2. la projection de Hammer (Figure A.20)
3. la projection azimutale équidistante (Figure A.21.c)
4. la projection conique de Albers (Figure A.21.d)

Pour un maximum de flexibilité une méthode basée image est adoptée pour le rendu. En terme d'amplitude de champ visuel, la Figure A.22 met en évidence la quantité d'information supplémentaire disponible comparé à une projection perspective classique. Pour la comparaison, le champ couvert par la projection perspective et par le champ visuel naturel

(a)

(b)

(c)

(d)

**Figure A.21 –** Les méthodes de projections utilisées pour l'évaluation: (a) projection prospection (ie. la condition de contrôle fournissant un champ visuel standard), (b) projection équirectangulaire,(c) projection conique de Albers, (d) projection azimutale équidistante.


**Figure A.22 –** Comparaison des champs visuels couverts par 1) la projection équirectangulaire, 2) le champ visuel humain binoculaire (bleu clair), 3) une projection perspective classique (rouge clair).

sont projetés en utilisant une projection équirectangulaire. Ceci explique la forme courbe du champ couvert par la projection perspective.

Le premier objectif de notre évaluation est de vérifier si l'exécution d'une tâche effectuée dans un environnent virtuel peut être améliorée grâce à une vision étendue à 360°. Un second objectif est de comparer les projections non planaires entre elles. Une évaluation quantitative a été complétée par un questionnaire recueillant l'avis subjectif des utilisateurs.

### A.4.1 Population et appareil expérimental

15 personnes ont participé, âgées de 21 a 44 ans. Toutes étaient familières avec les jeux vidéo. Une manette de jeu était utilisée pour la navigation.

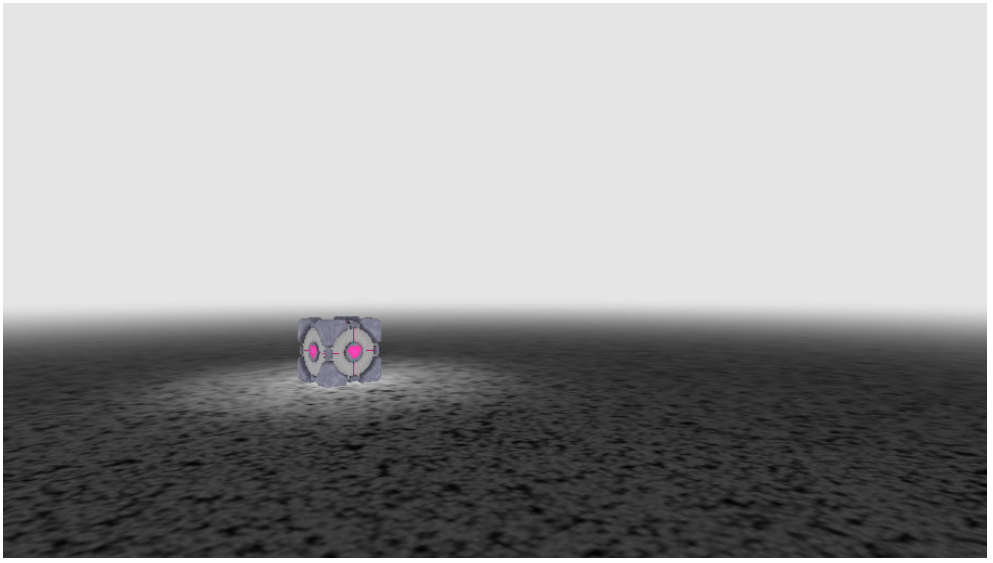


**Figure A.23 –** Environnement virtuel neutre, utilisé dans la première partie de l'évaluation.

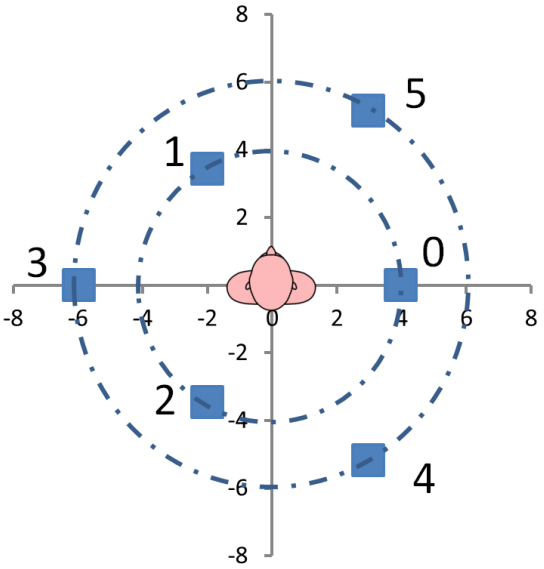



**Figure A.24 –** Position des cibles pour la tâche de collecte d'objets. La position initiale de l'utilisateur est orienté selon l'axe *y*.

### A.4.2 Procédure

L'évaluation se déroulait en deux parties. La première concernait l'évaluation de la performance de l'utilisateur dans le cadre d'une tâche de collecte d'objet (Figure A.23). Chaque participant avait à collecter 6 boites, l'une après l'autre, avec une position tirée aléatoirement parmi les positions décrites dans la Figure A.24. Les 5 projections était testés selon un ordre aléatoire.

La seconde partie consistait en une évaluation subjective des projections proposées. Un environnement virtuel plus détaillé était utilisé pour cette partie (Atrium Sponza Palace, voir Figure A.21.a). Les participant étaient libres de changer de méthode comme ils le souhaitaient pendant qu'ils remplissaient le questionnaire.

### A.4.3 Données collectées et résultats

Pour la première partie, le temps mis pour collecter chaque boite a été enregistré. Pour le questionnaire subjectif, 6 critères ont été utilisés, notés sur une échelle de 7 valeurs, pour les 5 projections. Ces critères étaient : (1) La fatigue visuelle, (2) l'aisance au déplacement dans l'environnent virtuel, (3)l'esthétique du rendu, (4) l'étrangeté du rendu, (5) le réalisme du rendu, (6) une appréciation générale.

Les résultats sont récapitulés dans les tableaux A.1 et A.2.

### A.4.4 Discussion

De façon globale, les projections équirectangulaire et de Hammer ont amélioré la performance en réduisant le temps pris par l'utilisateur pour collecter les boites. Les boites #3 et #4 ont été particulièrement plus rapides à atteindre. Elles étaient placées sur les cotés

**Table A.1 –** Temps pour effectuer la tâche (moyenne du temps pris pour atteindre les boites et variance).

| Projection method | all boxes | box 0 | box 1 | box 2 | box 3 | box 4 | box 5 |
|---|---|---|---|---|---|---|---|
| Perspective | 3.14 (1.39) | 3.64 (1.25) | 1.52 (0.36) | 3.30 (0.92) | 3.99 (1.59) | 4.18 (0.98) | 2.24 (0.45) |
| Equirec-tangular | 2.69 (1.56) | 2.15 (0.69) | 1.69 (1.59) | 3.08 (0.80) | 2.83 (1.14) | 3.64 (0.74) | 2.76 (2.63) |
| Hammer | 2.61 (1.38) | 2.20 (1.11) | 1.54 (0.71) | 3.16 (1.02) | 2.79 (1.90) | 3.53 (0.80) | 2.47 (1.46) |
| Conic | 3.07 (1.38) | 2.87 (2.71) | 2.09 (2.03) | 3.43 (1.12) | 3.10 (1.73) | 4.25 (1.72) | 2.66 (1.22) |
| Azimuthal | 3.04 (2.70) | 2.69 (2.14) | 1.44 (0.61) | 3.90 (1.91) | 3.44 (3.53) | 4.48 (4.17) | 2.12 (0.40) |

**Table A.2 –** Analyse post-hoc du questionnaire. La presence d'un critère dans une cellule du tableau indique un effet significatif entre les deux conditions testées. La meilleure est la condition sur la ligne.

| $>$ | Equirec. | Hammer | Conic | Azimuthal |
|---|---|---|---|---|
| Persp. | Aesthetic<br>Strangeness<br>Realism<br>Global app. | Aesthetic<br>Strangeness<br>Realism<br>Global app. | Visual fatigue<br>Aesthetic<br>Strangeness<br>Realism<br>Global app. | Ease of disp.<br>Aesthetic<br>Strangeness<br>Realism |
| Equirec. | | | Aesthetic<br>Realism<br>Global app. | Ease of disp. |
| Hammer | | | Strangeness<br>Realism<br>Global app. | |

de l'utilisateur, hors du champ visuel accessible avec la projection perspective. La projection de Albers n'a pas donné de résultats comparables aux autres projections. La forme particulière, en camembert peut expliquer cette contre performance. Il serait intéressant d'analyser plus en détail le temps requis pour appréhender cette forme particulière, comparé au temps nécessaire pour autres projections.

Les résultats pour le questionnaire subjectif montre que la projection perspective reste globalement préférée aux autres. Ce type de projection est rencontré régulièrement dans la vie de tous les jours, dans les jeux vidéo, la photographie ou encore le cinéma. Ceci peut expliquer cette préférence, aucun effort n'étant nécessaire pour s'accoutumer à cette projection. Parmi les méthodes de projection à 360°, la projection équirectangulaire et celle de Hammer ont été préférées par les utilisateurs. Leur utilisation est donc à privilégier dans les scénarios où une extension du champ visuel est souhaitée.

# Author's publications

## International conference publications

[PUB4] **J. Ardouin**, A. Lécuyer, M. Marchal, E. Marchand, "Stereoscopic Rendering of Virtual Environments with Wide Field-of-Views up to 360°", Proceedings of IEEE International Conference on Virtual Reality (IEEE VR), Minneapolis, 2014

[PUB3] **J. Ardouin**, A. Lécuyer, M. Marchal, E. Marchand, "Navigating in Virtual Environments with 360° Omnidirectional Rendering", Proceedings of IEEE International Symposium on 3D User Interfaces (IEEE 3DUI), Orlando, 2013

[PUB2] **J. Ardouin**, A. Lécuyer, M. Marchal, C. Riant, E. Marchand, "FlyVIZ: A Novel Display Device to Provide Humans with 360° Vision by Coupling Catadioptric Camera with HMD", Proceedings of ACM Symposium on Virtual Reality Software and Technology (ACM VRST), Toronto, 2012

[PUB1] **J. Ardouin**, A. Lécuyer, M. Marchal, E. Marchand, "Design and Evaluation of Methods to Prevent Frame Cancellation in Real-Time Stereoscopic Rendering", Proceedings of IEEE International Symposium on 3D User Interfaces (IEEE 3DUI), Singapore, 2011

## Patent

[PAT1] Display device suitable for providing an extended field of vision. WO 2013160255 A1.

# Bibliography


[Allison 01] Robert S Allison, Laurence R Harris, Michael Jenkin, Urszula Jasiobedzka, James E Zacher. – Tolerance of temporal delay in virtual environments. – *Virtual Reality, 2001. Proceedings. IEEE*, pp. 247–254. IEEE, 2001. 25

[Autodesk ] Autodesk. – Maya. – http://www.autodesk.com. 65, 67

[Autodesk 08] Autodesk. – Stereoscopic filmmaking whitepaper, the business and technology of stereoscopic filmmaking. – http://www.autodesk.com, 2008. 65

[Baker 98] Simon Baker, Shree K Nayar. – A theory of catadioptric image formation. – *Computer Vision, 1998. Sixth International Conference on*, pp. 35–42. IEEE, 1998. 39, 40

[Barreto 01] J.P. Barreto, H. Araujo. – Issues on the geometry of central catadioptric image formation. – *CVPR*, vol. 2, pp. 422–427, 2001. 23

[Bolte 10] Benjamin Bolte, Gerd Bruder, Frank Steinicke, Klaus Hinrichs, Markus Lappe. – Augmentation techniques for efficient exploration in head-mounted display environments. – *Proceedings of the 17th ACM Symposium on Virtual Reality Software and Technology*, pp. 11–18. ACM, 2010. 24

[Bourke 05] P. Bourke. – Spherical mirror: a new approach to hemispherical dome projection. – *GRAPHITE*, pp. 281–284, 2005. 25

[Bourke 06] Paul Bourke. – Synthetic stereoscopic panoramic images. *Interactive Technologies and Sociotechnical Systems*. – Springer, 2006. 33, 58

[Bowman 04a] Doug A. Bowman, Ernst Kruijff, Joseph J. LaViola, Ivan Poupyrev. – *3D User Interfaces: Theory and Practice*. – Addison Wesley Longman Publishing Co., Inc., 2004. 25

[Bowman 04b] Doug A Bowman, Ernst Kruijff, Joseph J LaViola Jr, Ivan Poupyrev. – *3D user interfaces: theory and practice*. – Addison-Wesley, 2004. 24

[Brosz 07] John Brosz, Faramarz F. Samavati, M. Sheelagh T. Carpendale, Mario Costa Sousa. – Single camera flexible projection. – *Proc. of NPAR*, pp. 33–42, 2007. 33

[Caron 11] G. Caron, D. Eynard. – Multiple camera types simultaneous stereo calibration. – *IEEE Int. Conf. on Robotics and Automation, ICRA'11*, pp. 2933–2938, Shanghai, China, May 2011. 24

[CruzNeira 93] Carolina Cruz-Neira, Daniel J. Sandin, Thomas A. DeFanti. – Surround-screen projection-based virtual reality: the design and implementation of the cave. – *Proc. of SIGGRAPH*, pp. 135–142, 1993. 25, 26, 58

[Cutting 97] James E Cutting. – How the eye measures reality and virtual reality. *Behavior Research Methods, Instruments, & Computers*, 29(1):27–36, 1997. 41, 96

[Deering 88] Michael Deering, Stephanie Winner, Bic Schediwy, Chris Duffy, Neil Hunt. – The triangle processor and normal vector shader: a vlsi system for high performance graphics. – *Proc. of SIGGRAPH*, pp. 21–30, 1988. 59

[Foley 82] James D Foley, Andries Van Dam. – *Fundamentals of interactive computer graphics*. – Addison-Wesley Reading, MA, vol. 2, 1982. 27

[Fuchs 13] E. Fuchs, A. Duane. – *Text-book of Ophthalmology*. – J. B. Lippincott Company, 1913. 8

[Gascuel 08] Jean-Dominique Gascuel, Nicolas Holzschuch, Gabriel Fournier, Bernard Péroche. – Fast non-linear projections using graphics hardware. – *Proc. of I3D*, pp. 107–114, 2008. 60

[Greene 86] N. Greene. – Environment mapping and other applications of world projections. *IEEE Computer Graphics and Applications*, 6(11):21–29, 1986. 31

[Haeberli 90] Paul Haeberli, Kurt Akeley. – The accumulation buffer: hardware support for high-quality rendering. *ACM SIGGRAPH Computer Graphics*, 24(4):309–318, 1990. 35

[Hartley 03] Richard Hartley, Andrew Zisserman. – *Multiple view geometry in computer vision*. – Cambridge university press, 2003. 23

[Kandel 00] Eric R Kandel, James H Schwartz, Thomas M Jessell. – Principles of neural science 4th ed, 2000. 9

[Kingslake 89] Rudolf Kingslake. – *A history of the photographic lens*. – Elsevier, 1989. 15, 18, 19

[Lang 10] Manuel Lang, Alexander Hornung, Oliver Wang, Steven Poulakos, Aljoscha Smolic, Markus Gross. – Nonlinear disparity mapping for stereoscopic 3d. *ACM Transactions on Graphics (TOG)*, 29(4):75, 2010. 65, 66, 67

[Lin 02] JJ-W Lin, Henry Been-Lirn Duh, Donald E Parker, Habib Abi-Rached, Thomas A Furness. – Effects of field of view on presence, enjoyment, memory, and simulator sickness in a virtual environment. – *Virtual Reality, 2002. Proceedings. IEEE*, pp. 164–171. IEEE, 2002. 24

[Lipton 07] Lenny Lipton. – Vertical surround parallax correction. – *Electronic Imaging 2007*, pp. 64900G–64900G. International Society for Optics and Photonics, 2007. 64, 65, 66, 67

[Lorenz 09] Haik Lorenz, Jürgen Döllner. – Real-time piecewise perspective projections. – *Proc. of GRAPP*, pp. 147–155, 2009. 32, 33, 35

[Marchand 07] Eric Marchand, François Chaumette. – Fitting 3d models on central catadioptric images. – *Robotics and Automation, 2007 IEEE International Conference on*, pp. 52–57. IEEE, 2007. 23

[Mulder 00] Jurriaan D Mulder, Robert Van Liere. – Enhancing fish tank vr. – *Virtual Reality, 2000. Proceedings. IEEE*, pp. 91–98. IEEE, 2000. 65, 66, 67

[Nalwa 96] Vic Nalwa. – *A true omnidirectional viewer*. – Research report, technical report, Bell Laboratories, 1996. 19, 22

[Nayar 97] Shree K Nayar. – Catadioptric omnidirectional camera. – *Computer Vision and Pattern Recognition, 1997. Proceedings., 1997 IEEE Computer Society Conference on*, pp. 482–488. IEEE, 1997. 19, 20, 23

[ofDefense 99] Department of Defense. – Design criteria standard - human engineering, 1999. 8, 9

[Oros 02] D. Oros. – A conceptual and practical look into spherical curvilinear projection, 2002. 33

[Osterberg 35] Gustav Osterberg. – *Topography of the layer of rods and cones in the human retina*. – Nyt Nordisk Forlag, 1935. 9, 10

[Palmer 99a] Stephen E Palmer. – *Vision science: Photons to phenomenology*. – MIT press Cambridge, MA, vol. 1, 34–36p., 1999. 9

[Palmer 99b] Stephen E Palmer. – *Vision science: Photons to phenomenology*. – MIT press Cambridge, MA, vol. 1, 80p., 1999. 9

[Patidar 06] S. Patidar, S. Bhattacharjee, J.M. Singh, P.J. Narayanan. – *Exploiting the Shader Model 4.0 Architecture*. – Research report, IIIT Hyderabad, 2006. 76

[Petkov 12] Kaloian Petkov, Charilaos Papadopoulos, Min Zhang, Arie E Kaufman, Xianfeng Gu. – Interactive visibility retargeting in vr using conformal visualization. *IEEE TVCG*, 18(7):1027–1040, 2012. 34, 35, 47, 49, 51, 52, 58, 98

[Rademacher 98] Paul Rademacher, Gary Bishop. – Multiple-center-of-projection images. – *Proceedings of the 25th annual conference on Computer graphics and interactive techniques*, pp. 199–206. ACM, 1998. 13

[Rubinstein 08] Michael Rubinstein, Ariel Shamir, Shai Avidan. – Improved seam carving for video retargeting. – *ACM transactions on graphics (TOG)*, vol. 27, p. 16. ACM, 2008. 65

[Scaramuzza 06a] Davide Scaramuzza, Agostino Martinelli, Roland Siegwart. – A flexible technique for accurate omnidirectional camera calibration and structure from motion. – *Computer Vision Systems, 2006 ICVS'06. IEEE International Conference on*, pp. 45–45. IEEE, 2006. 40, 41, 95

[Scaramuzza 06b] Davide Scaramuzza, Agostino Martinelli, Roland Siegwart. – A toolbox for easily calibrating omnidirectional cameras. – *Intelligent Robots and Systems, 2006 IEEE/RSJ International Conference on*, pp. 5695–5701. IEEE, 2006. 24

[Sherstyuk 12] A. Sherstyuk, A. Treskunov, M. Gavrilova. – Predator-prey vision metaphor for multi-tasking virtual environments. – *3D User Interfaces (3DUI), 2012 IEEE Symposium on*, pp. 81–84, March 2012. 1

[Shirley 09] Peter Shirley, Michael Ashikhmin, Steve Marschner. – *Fundamentals of computer graphics*. – CRC Press, 2009. 27

[Simon 04] A. Simon, R.C. Smith, R.R. Pawlicki. – Omnistereo for panoramic virtual environment display systems. – *Proc. of IEEE VR*, 2004. 25, 33

[Snyder 87] John P Snyder. – Map projections: A working manual (us geological survey professional paper 1395). *United States Government Printing O ce, Washington, DC*, 20402, 1987. 10, 13, 16, 35

[Snyder 97] J.P. Snyder. – *Flattening the Earth: Two Thousand Years of Map Projections*. – University of Chicago Press, 1997. 11, 51, 52, 73

[Spector 90] RH Spector. – *Clinical Methods: The History, Physical, and Laboratory Examinations. 3rd edition.* – Walker HK, Hall WD, Hurst JW, editors., 1990. 8

[Stein 12] Michael Stein, Maxi Robinski. – Simulator sickness in flight simulators of the german armed forces. *Aviation Psychology and Applied Human Factors*, 2(1):11, 2012. 86

[Trapp 08] M. Trapp, J. Döllner. – A generalization approach for 3d viewing deformations of single-center projections. – *Proc. of GRAPP*, pp. 162–170, 2008. 32

[Trapp 09a] M. Trapp, J. Döllner. – Generalization of single-center projections using projection tile screens. *Computer Vision and Computer Graphics. Theory and Applications*. – Springer, 2009. 32

[Trapp 09b] M. Trapp, H. Lorenz, J. Döllner. – Interactive stereo rendering for non-planar projections of 3d virtual environments. – *Proc. of GRAPP*, pp. 199–204, 2009. 35, 47, 59

[Valyus 66] NA Valyus. – Optical systems with multiple exit pupils for viewing stereoscreens without spectacles. *Stereoscopy,(Focal Press, London, 1966) esp*, pp. 161–163, 1966. 64

[Ware 97] Colin Ware, Daniel Fleet. – Integrating flying and fish tank metaphors with cyclopean scale. – *Computer Graphics International, 1997. Proceedings*, pp. 39–46. IEEE, 1997. 65, 67

[Wartell 02] Zachary Justin Wartell. – Stereoscopic head-tracked displays: analysis and development of display algorithms. 2002. 64, 66, 67

[Wright 96] John Westley Wright. – *A Text book of Ophthalmology*. – Columbus : J. L. Trauger, 1896. 8

[Zhang 00] Zhengyou Zhang. – A flexible new technique for camera calibration. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 22(11):1330–1334, 2000. 23

# Abstract

Who have never wanted to have eyes in the back of his head? This doctoral thesis proposes to study the extension of the human field-of-view (FoV) in both real and virtual environments.

First we have designed FlyVIZ, a new device to increase the human FoV. It is composed of a helmet, combining a catadioptric camera, a HMD and an image processing algorithm. Wearing this device allows a user to experience 360° vision of its surroundings. The prototype is demonstrated through scenarii such as grasping an object held out behind their back without turning their head or walking backward through doorways.

Then we have proposed a novel method to render virtual environments with wide FoV in real-time. To solve the rendering issue induced by usage of non-planar projections, we introduce a special stage in real-time rendering pipeline. Our method was then adapted for real-time stereoscopic rendering with 360° FoV. We have conducted a preliminary evaluation of real-time wide FoV rendering for a navigation task in virtual reality. Our results confirm that using a wide FoV rendering method could lead to more efficient navigation in terms of average task completion time. Among the different tested non-planar projection methods, the subjective preference is given to equirectangular and Hammer projections.

We also address the problem of frame cancellation, generated by the conflict between two depth cues: stereo disparity and occlusion with the screen border. We have proposed the Stereoscopy Compatible Volume Clipping (SCVC), solving the problem by rendering only the part of the viewing volume free of disparity - frame occlusion conflict. The method was evaluated and results have shown that SCVC notably improved users? depth perception and that the users expressed preference for SCVC.

Wide FoV opens novel perspectives for environments exploration or monitoring. Therefore, it could benefit to several applications, both in real world context or virtual environments. In safety and security applications, firemen, policemen or soldiers could take advantage of wide FoV. Performance of searching task and fast exploration in virtual environments could also be improved with wide FoV.

# Résumé

Qui n'a jamais souhaité avoir des yeux derrière la tête ? Cette thèse propose d'étudier l'extension du champ visuel humain, que ce soit dans le monde réel ou dans un environnement virtuel.

Nous avons d'abord conçu FlyVIZ, un dispositif qui permet d'augmenter le champ visuel. Il est composé d'une caméra catadioptrique, d'un visiocasque et d'un algorithme de traitement d'image. Lorsqu'un utilisateur porte ce dispositif, il dispose d'une vue à 360 degrés de son environnement. Le prototype a été testé avec succès dans différents scenarios, comme attraper un objet tendu dans le dos sans tourner la tête, ou passer des portes en marchant à reculons.

Ensuite nous avons proposé une nouvelle méthode pour le rendu d'environnements virtuels, avec un champ visuel étendu, et en temps réel. Pour résoudre les problèmes dus à l'utilisation de projections non planaires, nous avons ajouté une nouvelle étape dans le pipeline graphique. Notre méthode a ensuite été adaptée au rendu stéréoscopique avec un champ visuel de 360°. Nous avons mené une évaluation préliminaire sur l'utilisation d'un champ visuel étendu appliqué à une tache de navigation dans des environnements virtuels. Nos résultats semblent confirmer que l'utilisation d'un champ visuel étendu permet une navigation plus efficace, en diminuant le temps moyen pour effectuer une tache. Parmi les différentes projections non planaires testées, une préférence pour les projections équirectangulaire et de Hammer a été exprimée.

Nous avons également traité le problème de conflit d'indice de profondeurs rencontré dans les images stéréoscopiques lorsqu'un objet affiché en parallaxe négative est partiellement occulté par un bord du support d'affichage. Nous avons proposé SCVC (Stereoscopy Compatible Volume Clipping, le découpage de la scène selon le volume compatible avec la stéréoscopie) pour résoudre ce problème en n'affichant que la partie de l'espace qui n'est pas sujette au conflit d'indice de profondeur. La méthode a été évaluée et les résultats ont montrés que SCVC améliore significativement la perception de la profondeur et que les utilisateurs expriment une préférence pour cette méthode.

L'extension du champ visuel humain ouvre de nouvelles perspectives pour l'exploration ou la surveillance de l'environnement d'un utilisateur. Cette extension pourrait bénéficier à diverses applications, que ce soit dans le contexte d'un environnement réel ou virtuel. Pour la sécurité des personnes ou la défense, des pompiers, des policiers ou des soldats pourraient tirer avantage d'un champ visuel étendu. Ce type de visualisation peut également profiter à l'exploration rapide d'environnements virtuels ou à la recherche d'objet dans ces derniers.

INSA de RENNES
Service des Formations

# AVIS DU JURY SUR LA REPRODUCTION DE LA THESE SOUTENUE

**Titre de la thèse:**
Contribution à l'étude de la visualisation d'environnements réels et virtuels avec un champ visuel étendu

**Nom Prénom de l'auteur : ARDOUIN JEROME**

Membres du jury :
- Monsieur DRETTAKIS George
- Monsieur LECUYER Anatole
- Monsieur ARNALDI BRUNO
- Madame MARCHAL Maud
- Monsieur MOREAU Guillaume
- Madame COQUILLART Sabine

Président du jury : Bruno ARNALDI

Date de la soutenance : 17 Décembre 2014

Reproduction de la these soutenue

Thèse pouvant être reproduite en l'état
~~Thèse pouvant être reproduite après corrections suggérées~~

Fait à Rennes, le 17 Décembre 2014

Signature du président de jury

Le Directeur,

M'hamed DRISSI

## Résumé

Qui n'a jamais souhaité avoir des yeux derrière la tête ? Cette thèse propose d'étudier l'extension du champ visuel humain, que ce soit dans le monde réel ou dans un environnement virtuel.

Nous avons d'abord conçu FlyVIZ, un dispositif qui permet d'augmenter le champ visuel. Il est composé d'une caméra catadioptrique, d'un visiocasque et d'un algorithme de traitement d'image. Lorsqu'un utilisateur porte ce dispositif, il dispose d'une vue à 360 degrés de son environnement. Le prototype a été testé avec succès dans différents scenarios, comme attraper un objet tendu dans le dos sans tourner la tête, ou passer des portes en marchant à reculons.

Ensuite nous avons proposé une nouvelle méthode pour le rendu d'environnements virtuels, avec un champ visuel étendu, et en temps réel. Pour résoudre les problèmes dus à l'utilisation de projections non planes, nous avons ajouté une nouvelle étape dans le pipeline graphique. Notre méthode a ensuite été adaptée au rendu stéréoscopique avec un champ visuel de 360°. Nous avons mené une évaluation préliminaire sur l'utilisation d'un champ visuel étendu appliqué à une tache de navigation dans des environnements virtuels. Nos résultats semblent confirmer que l'utilisation d'un champ visuel étendu permet une navigation plus efficace, en diminuant le temps moyen pour effectuer une tache. Parmi les différentes projections non planes testées, une préférence pour les projections équirectangulaire et de Hammer a été exprimée.

Nous avons également traité le problème de conflit d'indice de profondeurs rencontré dans les images stéréoscopiques lorsqu'un objet affiché en parallaxe négative est partiellement occulté par un bord du support d'affichage. Nous avons proposé SCVC (Stereoscopy Compatible Volume Clipping, le découpage de la scène selon le volume compatible avec la stéréoscopie) pour résoudre ce problème en n'affichant que la partie de l'espace qui n'est pas sujette au conflit d'indice de profondeur. La méthode a été évaluée et les résultats ont montrés que SCVC améliore significativement la perception de la profondeur et que les utilisateurs expriment une préférence pour cette méthode.

L'extension du champ visuel humain ouvre de nouvelles perspectives pour l'exploration ou la surveillance de l'environnement d'un utilisateur. Cette extension pourrait bénéficier à diverses applications, que ce soit dans le contexte d'un environnement réel ou virtuel. Pour la sécurité des personnes ou la défense, des pompiers, des policiers ou des soldats pourraient tirer avantage d'un champ visuel étendu. Ce type de visualisation peut également profiter à l'exploration rapide d'environnements virtuels ou à la recherche d'objet dans ces derniers.

## Abstract

Who have never wanted to have eyes in the back of his head? This doctoral thesis proposes to study the extension of the human field-of-view (FoV) in both real and virtual environments.

First we have designed FlyVIZ, a new device to increase the human FoV. It is composed of a helmet, combining a catadioptric camera, a HMD and an image processing algorithm. Wearing this device allows a user to experience 360° vision of its surroundings. The prototype is demonstrated through scenarii such as grasping an object held out behind their back without turning their head or walking backward through doorways.

Then we have proposed a novel method to render virtual environments with wide FoV in real-time. To solve the rendering issue induced by usage of non-planar projections, we introduce a special stage in real-time rendering pipeline. Our method was then adapted for real-time stereoscopic rendering with 360° FoV. We have conducted a preliminary evaluation of real-time wide FoV rendering for a navigation task in virtual reality. Our results confirm that using a wide FoV rendering method could lead to more efficient navigation in terms of average task completion time. Among the different tested non-planar projection methods, the subjective preference is given to equirectangular and Hammer projections.

We also address the problem of frame cancellation, generated by the conflict between two depth cues: stereo disparity and occlusion with the screen border. We have proposed the Stereoscopy Compatible Volume Clipping (SCVC), solving the problem by rendering only the part of the viewing volume free of disparity - frame occlusion conflict. The method was evaluated and results have shown that SCVC notably improved users' depth perception and that the users expressed preference for SCVC.

Wide FoV opens novel perspectives for environments exploration or monitoring. Therefore, it could benefit to several applications, both in real world context or virtual environments. In safety and security applications, firemen, policemen or soldiers could take advantage of wide FoV. Performance of searching task and fast exploration in virtual environments could also be improved with wide FoV.



N° d'ordre : **14ISAR 40** / **D14 - 40**



**Institut National des Sciences Appliquées de Rennes**
20, Avenue des Buttes de Coëmes • CS 70839 • F-35708 Rennes Cedex
Tel : 02 23 23 82 00 - Fax : 02 23 23 83 96