

Tiago José Simões Dias

Augmented Reality using Non-Central Catadioptric Imaging Devices

September 2014



UNIVERSIDADE DE COIMBRA



Department of Electrical and Computer Engineering,
Faculty of Sciences and Technology, University of Coimbra,
3030-290 COIMBRA, PORTUGAL.

A Dissertation for Graduate Study in MSc Program
Master of Science in Electrical and Computer Engineering

Augmented Reality using Non-Central Catadioptric Imaging Devices

Tiago José Simões Dias

Supervisors:

Prof. Dr. Nuno Miguel Mendonça da Silva Gonçalves
Dr. Pedro Daniel dos Santos Miraldo

Jury:

Prof. Dr. Helder de Jesus Araújo
Prof. Dr. Nuno Miguel Mendonça da Silva Gonçalves
Prof. Dr. Rui Paulo Pinto da Rocha

September 2014

Agradecimentos

Quero começar por agradecer ao meu orientador Professor Doutor Nuno Gonçalves e ao meu co-orientador Doutor Pedro Miraldo pelo constante apoio e conhecimento científico disponibilizados durante a realização deste projeto, assim como toda a disponibilidade e, muitas vezes, paciência durante o todo o trajeto desde o início do projeto até à sua conclusão.

Quero também agradecer de forma especial aos meus pais e irmã, José Dias, Maria Isabel Dias e Maria João Dias respetivamente, por todo o incentivo, carinho e acima tudo pela crença que sempre depositaram em mim mesmo nas alturas mais difíceis durante o meu percurso académico.

Quero agradecer a todos os meus colegas do Laboratório de Visão por Computador que sempre foram prestáveis e acima de tudo tornaram a ambientação a este novo meio um pouco mais fácil. Com medo de me esquecer de alguém em especial fica o agradecimento ao Jérôme Mendes, ao Diogo Ferreira, ao Diogo Sereno, ao Miguel Luz, ao Luís Maricato, ao João Costa e ao João Barata.

Quero, por fim, fazer um agradecimento especial a todos meus amigos e à Telma Margalho por todo o carinho, apoio e disponibilidade prestados ao longo do meu percurso académico, sem eles todo o trajeto teria sido bem mais complicado.

Abstract

In this dissertation it is proposed a framework for the use of augmented reality using non-central catadioptric imaging devices. Our system is composed by a non-central catadioptric imaging device formed by a perspective camera and a spherical mirror mounted on a Pioneer 3-DX robot. In this dissertation, our main goal is considering a 3D virtual object in the world, with known 3D coordinates, make the projection of this 3D virtual object into the 2D image plane of a non-central catadioptric imaging device. Our framework presents a solution which allows us to project texturized objects (with detailed images or single color textures) to the image in realtime, up to 20 fps (using a laptop), depending on the 3D object that will be projected. When dealing with an implementation of augmented reality, some important issues must be considered, such as: projection of 3D virtual objects to the 2D image plane, occlusions, illumination and shading. To the best of our knowledge this is the first time that this problem is addressed (all state-of-the-art methods are derived for central camera systems). Thus, since this is an unexplored subject for non-central systems, some reformulations and implementations of algorithms and methodologies must be done to solve our problems. To make clear our approach a pipeline was made, composed by two stages: pre-processing and realtime. Each one of these stages have a sequence of steps, that must be done to preserve correct operation of our framework. The pre-processing stage contains three steps: camera calibration, 3D object triangulation and object texturization. The realtime stage is also composed by three steps: “QI projection”, occlusions and illumination.

To test the robustness of our framework three distinct 3D virtual objects were used. The 3D objects used were: a parallelepiped and Stanford bunny and the happy Buddha from the Stanford repository. For each object several tests were made: using a positional light (at the top of a static robot or in an arbitrary position) pointing to the 3D object; using a moving light source using two different movements (not at the same time); using three light sources, with different colors, with different movements associated at each one pointing to the object, and using a light source, positioned at the top of the robot, in a moving robot where the source is

pointing to the 3D virtual object position.

Keywords: Augmented Reality, Non-central Catadioptric Imaging Devices, Triangulation, Texturization, Image Projection, Occlusions, Illumination.

Resumo

Nesta dissertação é proposta uma ferramenta para uso de realidade aumentada utilizando dispositivos de imagem catadióptricos não centrais. O nosso sistema é composto por um dispositivo catadióptrico não central formado por uma câmara perspectiva e um espelho esférico montados num robot Pioneer 3-DX. Nesta dissertação, o nosso maior objetivo é considerando um objeto virtual 3D, com coordenadas no mundo conhecidas, fazer a projeção deste objeto virtual 3D para o plano imagem 2D de um dispositivo catadióptrico não central. A nossa ferramenta apresenta uma solução que nos permite projetar objetos texturizados (com imagens detalhadas ou texturas de uma cor sólida) para a imagem em tempo real, com um máximo de 20 fps (utilizando um computador portátil), dependendo do objeto 3D a ser projetado. Quando confrontados com uma implementação de realidade aumentada, certos problemas importantes devem ser tidos em conta, tais como: projeção de objetos virtuais 3D para o plano imagem 2D, oclusões, iluminação e sombras. De acordo com o nosso conhecimento esta é a primeira vez que este problema é abordado (todos os métodos que compõem o estado da arte foram apenas abordados para sistema de câmaras centrais). Então, uma vez que é uma ideia inexplorada, para este tipo de sistemas, são feitas reformulações e implementações de algoritmos e metodologias de maneira a resolver os problemas que apresentem. Para tornar clara a nossa abordagem um “pipeline” foi feito, composto por duas etapas: pré-processamento e tempo real. Cada um destes estados é composto por uma sequência de passos que devem ser efetuados de maneira a preservar o funcionamento correto da nossa ferramenta. A etapa de pré-processamento é formada por três passos: calibração da câmara, triangulação do objeto virtual 3D e texturização do objeto. A etapa de tempo real é também formada por três passos: “*QI projection*”, oclusões e iluminação.

Para testar a robustez da nossa ferramenta três objetos virtuais 3D distintos foram utilizados. Os objetos 3D utilizados foram: um paralelepípedo, o “Stanford Bunny” e o “Happy Buddha” disponibilizados pelo repositório da Stanford. Para cada objeto foram feitos vários testes: usando uma fonte de luz posicional (no topo do robô ou numa determinada posição arbitrária)

apontando para o objeto 3D; usando uma fonte de luz com dois movimentos diferentes (não ao mesmo tempo); usando três fontes de luz, de cores distintas, com movimento diferentes apontando para o objeto; usando uma fonte de luz, posicionada no topo do robô, num robô em movimento com a fonte de luz a apontar para a posição do objeto virtual 3D.

Palavras-chave: Realidade Aumentada, Sistemas catadiótricos não centrais, Triangulação, Texturização, Projeção para a Imagem, Oclusões, Iluminação.

Contents

1	Introduction	1
1.1	An approach to our work	3
1.2	Motivation, Goals and Contributions	5
1.3	Outline of the thesis	6
2	Related Work	7
2.1	Camera Model	7
2.2	Camera Pose Estimation	8
2.3	Computation of the Reflection Point on the Mirror	9
2.4	Algorithms to deal with the Occlusions	11
2.5	Illumination and shading models	12
2.6	Augmented reality applications	14
2.6.1	Fournier <i>et al.</i> method	15
2.6.2	Debevec method	16
2.6.3	Sato <i>et al.</i> method	17
3	Proposed Method	23
3.1	The Proposed Pipeline	23
3.1.1	Pre-Processing Stage	23
3.1.2	Realtime Stage	24

4	Detailed Description of the Pipeline	27
4.1	Camera Calibration	27
4.2	3D Object Triangulation	28
4.3	Texture Mapping	29
4.4	QI Projection	30
4.5	Occlusions	32
4.6	Illumination	33
5	Experimental Results	37
6	Conclusions and Future Work	47
6.1	Conclusions	47
6.2	Future Work	48

List of Figures

1.1	In this figure it is presented, on the left, the True Omnidirectional system proposed by Nalwa at [26] formed by four planar mirrors and four cameras. On the right side, it is presented our system built by a non-central catadioptric imaging system and a Pioneer 3-DX robot.	2
1.2	Example of the application of the proposed framework for augmented reality applications, using non-central catadioptric camera models. On the left, the projection of a virtual texturized parallelepiped to the image, on the center, the projection of the Stanford bunny object to the image and, on the right, the projection of the happy Buddha.	3
1.3	Illustration of the Stanford bunny and the happy Buddha virtual objects, in Figs. (a) and (b) respectively, used to test our framework.	4
2.1	Representation of both schemes for the third restriction to find the reflection point on the quadric mirror. In Fig. (a) it is represented the scheme used to solve the problem through the Reflection Law and, in the Fig. (b) the scheme used to solve the problem through the Fermat Principle.	10
2.2	Example of an occlusion problem, presented at [19], between three rectangles. . .	12
2.3	In Fig. (a) it is illustrated the Phong reflection model (the description of the figure is done throughout the section). In Fig. (b) we show an example from [19] for the teapot object when applying different techniques of shading. On the left image a flat shading example and, on the right image, a smooth shading example (Gouraud shading).	13

2.4	In Figs. (a), (b), (c) and (d) are represented the influence of each property individually and all together respectively. The first image shows the influence of the ambient light property, the second image shows the influence of the diffuse light property. the third image shows the influence of the specular light property and the fourth (and final) image shows the combination of all the properties. These images were extracted from [19].	14
2.5	On the left it is represented the real video image with physical object replaced by boxes, on the center figure it is represented only the boxes, instead of the physical objects, and the two augmented objects (highlighted in the figure). On the right it is represented the final result in the merging of the augmented objects (highlighted in the figure) with the real video image. These images were extracted from [14]. .	16
2.6	In Fig. (a) are represented the images acquired by photographing a mirrored ball with three exposure levels: +0, -3.5 and 7.0 stops respectively. In the last image (right side) it can be observed that the full dynamic of the scene was obtained without saturation, which means that intensity, color and direction of all forms of the incident light can be recorded. In Fig.(b) are represented the images used to obtain the full dynamic range environment map of the scene obtained by two light probe images, taken with ninety degrees from distance of each other using three exposure levels are: -4.5, 0 and 4.5 stops. These images were extracted from [10]. .	19
2.7	In Fig.(a) are represented two results from an indoor and an outdoor environment using the radiance map at top corner of both images. In Fig.(b) it is presented the result taking into account the lighting information obtained in Fig.2.6(b). These images were extracted from [10].	20
2.8	In Fig.(a) it is represented the scheme that represents the acquisition of the background photograph. In Fig.(b) it is represented the scheme in using a light probe to obtain the incident radiance near to the location where the virtual objects will be placed. In Fig.(c) it is represented the scheme to construct the light-based model. In Fig.(d) it is represented the scheme for the computation of the global illumination solution. In Fig.(e) it is illustrated the result obtained taking into account the steps mentioned in previous sub-images. These images were extracted from [10].	21

2.9	In Fig.(a) are presented the images of the real environment, the calibration image and the two omnidirectional images (obtained with a camera with a fisheye lens) respectively, for an indoor environment. In Fig.(b) are presented the same type of images for an outdoor environment. These images were extracted from [31].	22
2.10	In Fig.(a) are presented the results obtained with the presented method for an indoor environment. In Fig.(b) are presented the results obtained with the presented method for an outdoor environment. These images were extracted from [31].	22
3.1	Representation of the proposed pipeline for the application of augmented reality in non-central catadioptric camera models.	24
3.2	Representation of the projection of 3D point to the 2D image plane of a general catadioptric camera and the illumination model used.	25
4.1	In this figure it is presented the two images used to apply as textures to our cube faces. On the left side, the image applied to the vertical faces and, on the right side, the image applied to the horizontal faces.	29
4.2	Representation of the problem of the estimation of the 3D virtual object to the 2D image. First we need to know the rigid transformation to do the mapping of each 3D point from the world coordinates to the camera coordinates system, $\mathbf{H}^{(CW)}$, obtained by the camera pose estimation. Then, the transformation from the camera coordinate system to the quadric coordinate system, using the inverse of $\mathbf{H}^{(CO)}$, is given by the calibration step (Sec. 4.1). The next stage is the computation of the reflection point in the mirror through the “QI Projection” method. Then, we just need to apply the transformation of $\mathbf{H}^{(CO)}$ to obtain the point in the mirror coordinates and using the intrinsic parameters matrix obtain the corresponding 2D point in the image.	31
4.3	Example of a spotlight illustrated at [33] showing the cutoff angle that will affect the light intensity in each triangle.	35

5.1	In this figure we show the results of the application of the various steps of the pipeline applied to the virtual parallelepiped. Fig. (a) represent the projection of the 3D triangles to the image, which correspond to the “QI Projection” step of the pipeline. The goal of Fig. (b) is to show the effects of the occlusions between the faces of the virtual parallelepiped. In Figs. (c) and (d) we show the result for the occlusion step using solid color textures, for each face, and using more detailed textures respectively.	38
5.2	In this figure we show the results of the application of the steps of the pipeline to the bunny and the Buddha objects. Fig. (a) represent the projection of the 3D triangles to the image (“QI Projection” step of the pipeline). Figs. (b) and (c) show the influence of the illumination in the objects.	39
5.3	In this figure we show a set of frames in which we apply the proposed framework, considering a moving spotlight affecting the virtual parallelepiped. To obtain this result, a circular movement was applied to the spotlight to show that our solution, for the illumination step, is working correctly.	42
5.4	In this figure we show a set of frames in which we apply the proposed framework, considering a moving spotlight affecting the Stanford bunny. To obtain this result, a circular movement was applied to the spotlight to show that our solution, for the illumination step, is working correctly.	43
5.5	In this figure we show a set of frames in which we apply the proposed framework, considering a moving spotlight affecting the Happy Buddha. To obtain this result, a circular movement was applied to the spotlight to show that our solution, for the illumination step, is working correctly.	44
5.6	In this figure we show a set of frames in which we apply the proposed framework, considering three moving spotlights with different colors (blue, green and red) affecting the Stanford bunny. To obtain this result three different movements were applied to each one of the spotlights to show that our solution, for the illumination step, is working correctly with the use of multiple spotlights in our framework. . .	45

5.7	In this figure we show a set of frames from three different positions of the robot. On the left column, it is shown the image obtained by the auxiliar camera, which is acquiring the realtime events in the real world, on the center column, we show the 3D virtual arena showing the position of the robot in the arena and, on the right column, the result of our framework according to the position of the robot and light focus (which is on the top of the robot).	46
6.1	In this figure it is represented the illumination problem when dealing with multiple objects, Buddha character and the base that it stands on (highlighted).	48

Chapter 1

Introduction

Nowadays, with the constant development and improvement in technology, several ways for people interact with real and virtual environments have been proposed. Two different types of technologies are getting a large focus from the industry: virtual reality and mixed reality. Definitions of each one have been made through the years as stated by Steuer [36] and Milgram and Kishino [21]. Virtual reality can be defined as the immersion of the user in a virtual environment, with which he/she can interact with. The interaction can be made with particular devices, *e.g.* head mounted-devices or motion-sensing gloves. Mixed reality is composed by two different “realities”, one where virtual objects are generated in a real environment (known as augmented reality) and other where real objects are merged with virtual environments (known as augmented virtuality).

Augmented reality has been studied for almost fifty years [3]. As stated by Azuma [4] augmented reality can also be defined as the projection of virtual 3D objects into the image plane. For the conventional perspective camera model, a large number of distinct methods have been presented, *e.g.* [14, 10, 31, 13]. The main reason for the use of these cameras is their simplicity (specially what is related to the projection model) and wide availability. However, in the last two decades, new types of imaging devices have started to be used due to several advantages related to their visual fields. At 1996 Nalwa [26] introduced what he claims to be the first omnidirectional system (built using four cameras pointing to four planar mirrors, Fig. 1.1) which was designed to fulfill the mathematical properties of the perspective cameras. Basically, the goal was to ensure that all the projection rays will intersect at some 3D point (central camera systems). Omnidirectional systems can be very useful for robot navigation, video vigilance systems

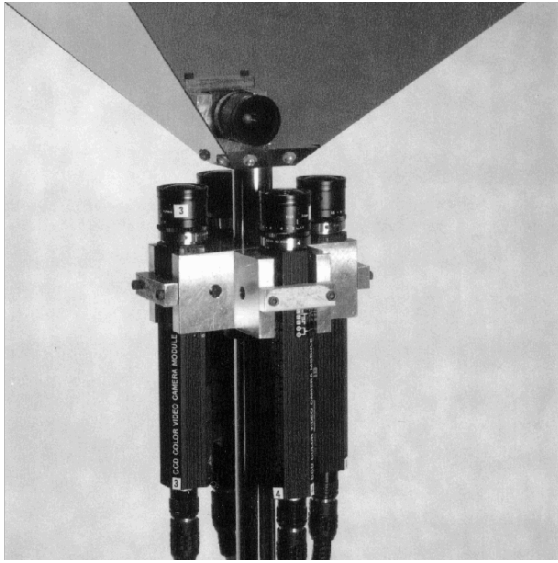


Fig. 1.1: In this figure it is presented, on the left, the True Omnidirectional system proposed by Nalwa at [26] formed by four planar mirrors and four cameras. On the right side, it is presented our system built by a non-central catadioptric imaging system and a Pioneer 3-DX robot.

or medical imaging devices where wide fields of view are fundamental.

In 1997, Nayar and Baker [27] studied the use of a single camera and a single quadric mirror to create omni-directional systems. Later ([5]), they studied the sufficient conditions to ensure that these systems fulfill the central camera properties. Other authors, *e.g.* [15, 6], studied the case where catadioptric cameras fulfill the mathematical properties of the central perspective cameras. The main problem is that, to get central systems, the camera must be perfectly aligned with the mirror's axis of symmetry and one must use a specific type of mirror. For example, one cannot use spherical mirrors. Small misaligned systems or different types of mirrors will not verify the constraint that all the projection lines intersect at a single 3D point, also denoted as viewpoint. Then, most of the times we have a non-central camera system. This problem was analyzed by Swaminathan *et al.* [37]. They found out that the "locus of viewpoints" forms what is called a caustic. They analyzed the properties of this caustics and presented a calibration procedure for non-central conic catadioptric systems. Later, because of the utility of these imaging devices, several authors proposed models and calibration procedures for non-central catadioptric camera system using general quadric mirrors, *e.g.* [20, 29, 30, 1].

In this dissertation, we focus the study of augmented reality for non-central catadioptric

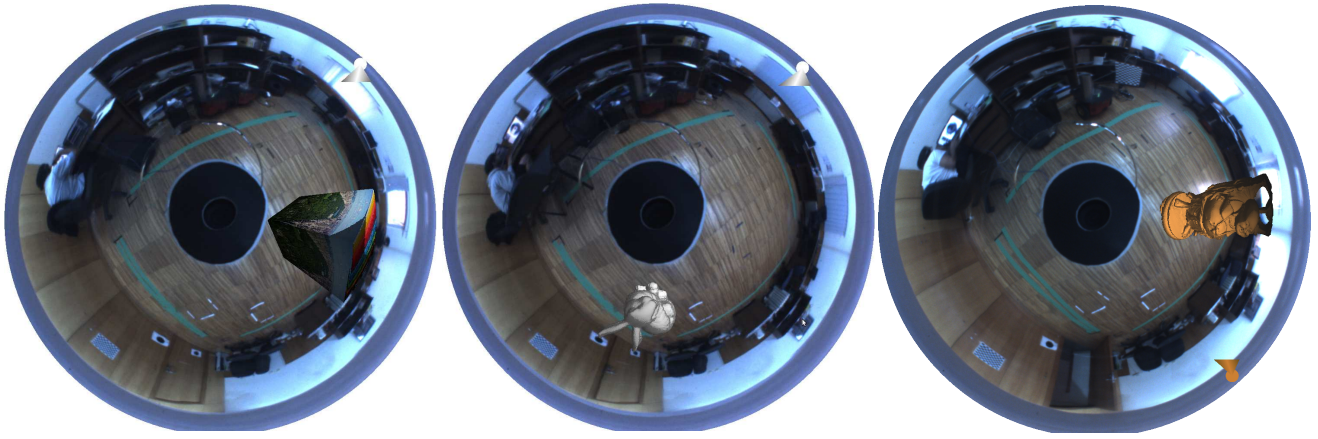


Fig. 1.2: Example of the application of the proposed framework for augmented reality applications, using non-central catadioptric camera models. On the left, the projection of a virtual texturized parallelepiped to the image, on the center, the projection of the Stanford bunny object to the image and, on the right, the projection of the happy Buddha.

imaging devices. In Fig. 1.1 it is represented (on the right side) the system used for testing our framework. Our system is built by a non-central catadioptric imaging device formed with a perspective camera and a spherical mirror mounted on a Pioneer 3-DX robot (from Mobile Robots [25]). In Fig. 1.2 are represented some results with the system used.

1.1 An approach to our work

In this dissertation it is thus proposed a framework for the use of augmented reality using non-central catadioptric imaging devices. An example of the results are shown in Fig. 1.2. To the best of our knowledge, this is the first time that the problem is addressed.

A pipeline based implementation is shown in Fig. 3.1. To get to our goal, new algorithms were created and some well known methods were reformulated, so that they could be applied to non-central catadioptric systems. Assuming that the camera calibration is known and that our 3D object is triangulated and texturized, the most challenge step is to project these 3D triangles (which form the 3D objects) to the image plane. Assuming that the triangles are small enough, the effects of distortion are neglectable [38]. As a result, to project these 3D triangles it is only necessary to take into account the projection of three 3D points (that form the vertices of each triangle) to the 2D image plane. This problem was addressed by Gonçalves [16] (which denoted the problem as “Quadric Intersection (QI) Projection”) and Agrawal *et al.* [2]. Since the geometry

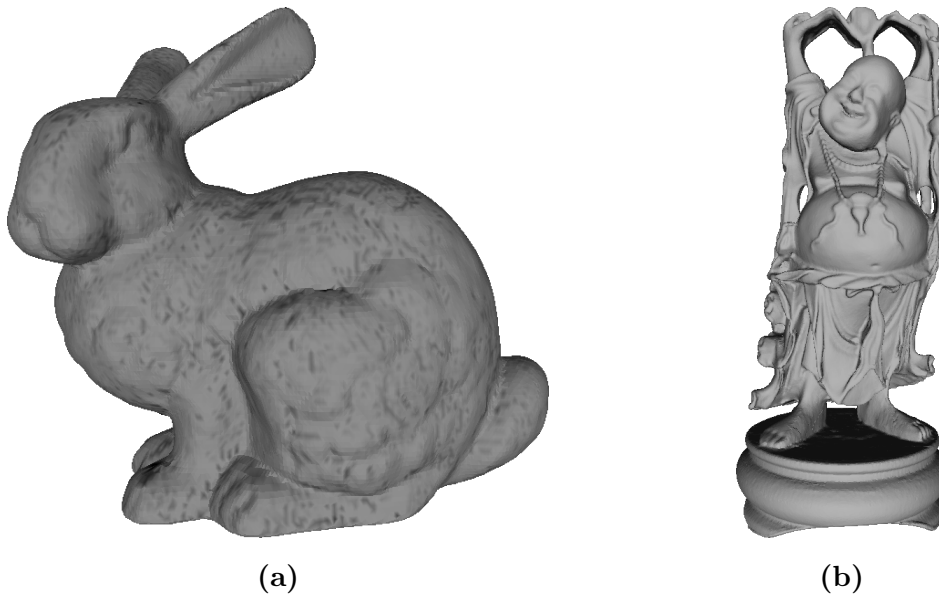


Fig. 1.3: Illustration of the Stanford bunny and the happy Buddha virtual objects, in Figs. (a) and (b) respectively, used to test our framework.

of these imaging systems does not verify most properties of the conventional perspective cameras, conventional approaches to other problems had also to be reformulated, such as occlusions and object illumination and shading.

Occlusions is a very well known problem in 3D computer graphics. Assuming that a 3D virtual object is divided in small 3D pieces (for example 3D triangles), when mapping these small pieces to the image, it must be verified if the pieces are overlapped and, in case they are, which of them are visible or not. To solve this problem, several methodologies were presented, for example: the painter's Algorithm [19, Chapter 36.4], Z-Buffer (also known as Depth Buffer) [19, Chapter 36.3] and A-Buffer [8]. In this dissertation we present a solution for this problem using non-central catadioptric systems.

Another very important issue in computer graphics is the object illumination and shading. Two important models were presented to handle light reflection, Phong reflection model and Torrance-Sparrow reflection model [7]. Thus, for a 3D object with a solid color, without illumination, it is considered that the projection of this 3D object to the image will be represented by a blob (binary large object). The implementation of illumination and shading algorithms will give shape to the projection of the 3D object (this problem is further evaluated in the experimental results in Chapter 5). To solve this problem, several algorithms of flat and smooth shading were proposed such as: flat shading [19, Chapter 6.2], Gouraud shading [19, Chapter 6.3] and Phong

shading [19, Chapter 6.5]. For this problem of illumination/shading a solution is proposed for non-central catadioptric imaging devices.

To conclude, the display of the 2D projected object is done. For simplicity, it was used the OpenGL API (Application Programming Interface), which is frequently used in the development of graphics interfaces.

The proposed method was tested using both a virtual parallelepiped (which had to be triangulated) and two well known objects, in computer graphics field, the Stanford bunny and the Happy Buddha (both already triangulated and presented in Fig. 1.3) from Stanford repository ([35] and [34] respectively).

All the algorithms proposed were implemented in C/C++ language. Because of the complexity of some of the implemented algorithms, specially the “QI Projection” method, the results obtained only got up to 2 frames per second (fps), relatively far from the 25 fps normally associated to a realtime application. Thus, it was decided to use the Graphics Processing Unit (GPU) to release some of the processing from the Central Processing Unit (CPU). So, to solve this problem, it was used the CUDA Toolkit (from NVIDIA Corporation) to improve the results previously obtained only using the CPU. With the use of the GPU, the results were significantly improved, obtaining results near 25 fps.

1.2 Motivation, Goals and Contributions

Augmented reality has been a relevant field of study through the years. The effects of enjoyability and pleasure, even the fascination that the technology provide to the users is being a relevant factor for the development of innumerable applications, *e.g.* Google Glass, mobile applications. It is said that the Google Glass will be the technology with the most impact and acceptance since the internet appearance. Augmented reality can be understood as a multidisciplinary area involving distinct fields of study such as robotics, computer vision and computer graphics. Although the concept of augmented reality is a very well known study for conventional central systems, in this dissertation it is proposed a solution for the implementation of augmented reality using non-central catadioptric devices. Using this type of system our objective is to acquire a larger field of view than a central system could provide. According to all of state of the art studied this is the first time that this problem is addressed using non-central catadioptric systems. Our main

goal is, considering a virtual object in the world with known 3D coordinates, to project the 3D object into a non-central catadioptric camera image. A solution to this problem is proposed in way to perform the projection of texturized objects to the image near to a realtime frame rate.

With the proposed solution in this dissertation several contributions are being made, such as:

1. Implementation of an augmented reality framework using non-central catadioptric imaging devices;
2. Integration of a camera calibration algorithm (Perdigoto and Araujo [30], a camera pose estimation algorithm (Miraldo and Araujo [23]) and an algorithm for the computation of the reflection point in a quadric mirror, also known as the forward projection model, (Gonçalves [16]) in our framework;
3. Reformulation of an algorithm to deal with the occlusions based on the painter's algorithm methodology;
4. Reformulation of a conventional method to deal with the illumination of a virtual object;

A paper was submitted to the 3DV 2014 (International Conference on 3D Vision) conference, with the proposed framework presented in this dissertation (in review).

1.3 Outline of the thesis

In Chapter 2 it is presented the related work for non-central catadioptric system and some applications of augmented reality for conventional perspective systems. In Chapter 3 we present our proposed solution for the application of augmented reality for non-central catadioptric mirrors. For a better understanding a pipeline was made and each stage is described in this section. In Chapter 4 a detailed description of each step of the proposed pipeline stage is made. In Chapter 5 our experimental results and tests are presented for the proposed framework. In Chapter 6 the conclusions and the future work are presented.

Chapter 2

Related Work

In this chapter it will be presented some methodologies and algorithms used for non-central catadioptric imaging devices, that are essential to achieve our goal. For this objective, we have made a study of different fields in the computer vision and computer graphics. As example, it can be mentioned the calibration of a non-central catadioptric camera using a quadric mirror, pose estimation of the camera, computation of the reflection point in the quadric mirror (also known as forward projection model), in the computer vision field, and occlusions, illumination/shading in the computer graphics field.

Also, examples of some augmented reality applications will be presented for the conventional perspective system.

2.1 Camera Model

To solve the problem of the calibration of the camera for a non-central catadioptric system using a quadric mirror several solutions have been presented, as it was mentioned in the introduction chapter (*e.g.* [20, 29, 30, 1]).

The goal of the proposed methods is based in finding certain relevant parameters. First the quadratic equation that defines the quadric mirror must be known or calculated:

$$x^2 + y^2 + Az^2 + Bz - C = 0 \tag{2.1}$$

where the parameters A , B and C define the type of quadric that is taking into account. This

equation can be expressed also in matrix form, that we will denote as $\mathbf{\Omega}$ if and only if:

$$\mathbf{\Omega} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & A & \frac{B}{2} \\ 0 & 0 & \frac{B}{2} & -C \end{bmatrix} \quad (2.2)$$

which is a matrix defined in $\mathbb{R}^{4 \times 4}$.

A 3D point (X , in homogeneous coordinates) belongs to the quadric $\mathbf{\Omega}$ if:

$$X^T \mathbf{\Omega} X = 0 \quad (2.3)$$

After the calculation of the matrix that defines the quadric, it is needed to do the calibration of the matrix of the intrinsic parameters of the camera, that is normally denoted as $\mathbf{K} \in \mathbb{R}^{3 \times 3}$:

$$\mathbf{K} = \begin{bmatrix} f_x & s & u_0 \\ 0 & f_y & v_0 \\ 0 & 0 & 1 \end{bmatrix} \quad (2.4)$$

where f_x and f_y are the focal length parameters in each axis of coordinates (can be understood as the distance between the camera optical center and the 2D image plane), s is the skew parameter which is in the most common cases equal to zero and the u_0 and v_0 , the coordinates of the so-called central pixel or principal point, that correspond to the pixel in the image plane that is intersected by the optical axis. Finally, the matrix of the rigid transformation of the coordinates in the mirror to the camera coordinates, which we will define as $\mathbf{H}^{(CO)} \in \mathbb{R}^{4 \times 4}$, yields:

$$\mathbf{H}^{(CO)} = \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.5)$$

where the parameters r_{ij} are the elements of the rotation matrix and t_x , t_y , and t_z are the translation parameters.

2.2 Camera Pose Estimation

Through the years, several methods for the computation of the camera pose have been presented. When using a mobile camera, we cannot define a global coordinates system to represent the mapping between features in the world (in “world coordinate system”) and the image plane. So, we

need to associate a coordinate system to the camera to solve this problem (we denote this coordinate system as the “camera coordinate system”). As a result, a rigid transformation between the world and camera coordinate systems needs to be calculated. This rigid transformation, that we will define as $\mathbf{H}^{(CW)}$, is a matrix composed by rotation and translation parameters and it is normally described as:

$$\mathbf{H}^{(CW)} = \begin{bmatrix} r'_{11} & r'_{12} & r'_{13} & t'_x \\ r'_{21} & r'_{22} & r'_{23} & t'_y \\ r'_{31} & r'_{32} & r'_{33} & t'_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.6)$$

where the parameters r'_{ij} are the elements of the rotation matrix and t'_x , t'_y , and t'_z are the translation parameters.

General camera models were presented by Grossberg and Nayar [17]. The goal of these models is to represent any imaging device, central or non-central. To compute the camera pose using these models, there are algorithms: for minimal data [9, 28, 22]; for non-minimal data using points [9, 32]; for non-minimal case using 3D lines [23]; and for non-minimal case using planar patterns [24]. The solutions for minimal case are associated with the minimal conditions necessary to calculate this rigid transformation, which consists on the estimation of the pose using the matching between three 3D points and its respective 2D image points. On the other hand, non-minimal case consists in the use of a larger number of features (points, lines or planes). The later case is used to get a robust solution when using data with noise, while minimal solutions are used in hypothesis-and-test estimation methods (*e.g.* RANSAC).

2.3 Computation of the Reflection Point on the Mirror

The computation of the reflection of a 3D point on the mirror is a well known and difficult problem to solve in non-central catadioptric imaging devices. This problem is called as the “Alhazen’s problem” [2]. Solutions were proposed by Gonçalves [16] and Agrawal *et al.* [2] to this problem.

Gonçalves using the “QI Projection” model divided the problem in three constraints. The first one is that the 3D point (in homogeneous coordinates) must belong to the quadric mirror Ω (presented in Sec. 2.1),

$$X^T \Omega X = 0 \quad (2.7)$$

then this point must also belong to another analytical quadric (second constraint), that was presented as \mathbf{S} , that intersect Ω . This reduce our search to this intersection that forms a curve

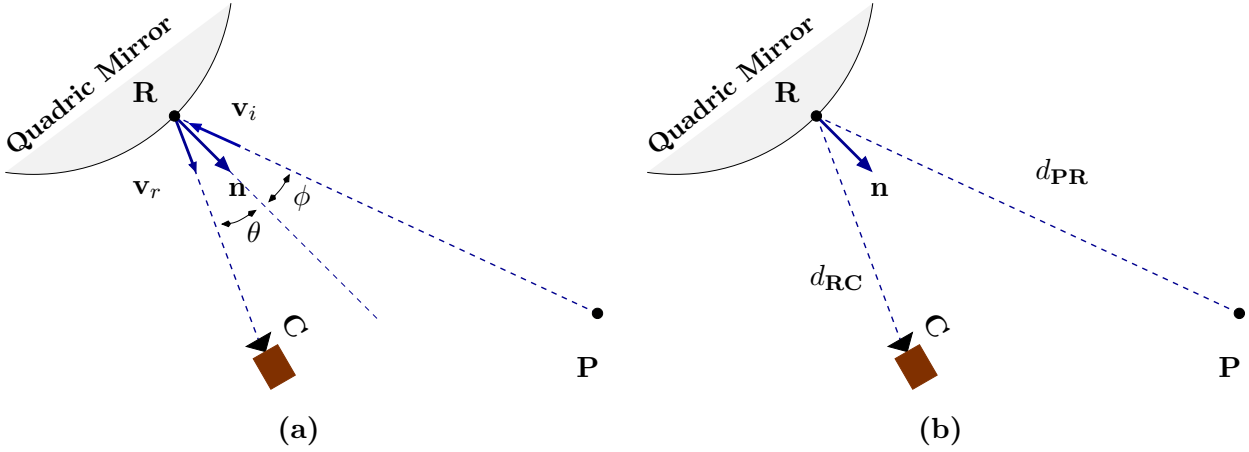


Fig. 2.1: Representation of both schemes for the third restriction to find the reflection point on the quadric mirror. In Fig. (a) it is represented the scheme used to solve the problem through the Reflection Law and, in the Fig. (b) the scheme used to solve the problem through the Fermat Principle.

in space called quartic. The matrix \mathbf{S} is defined by the following equation:

$$\mathbf{S} = \mathbf{M}^T \mathbf{Q}_\infty^* \mathbf{\Omega} + \mathbf{\Omega}^T \mathbf{Q}_\infty^* \mathbf{M} \quad (2.8)$$

where \mathbf{Q}_∞^* is the absolute dual quadric and \mathbf{M} is the matrix composed using the center of projection of the camera and the 3D point. The matrix \mathbf{M} is represented as:

$$\mathbf{M} = \begin{bmatrix} 0 & c_z - Z & -c_y + Y & c_y Z - c_z Y \\ -c_z + Z & 0 & c_x - X & -c_x Z + c_z X \\ c_y - Y & -c_x + X & 0 & c_x Y - c_y X \\ -c_y Z + c_z Y & c_x Z - c_z X & -c_x Y + c_y X & 0 \end{bmatrix} \quad (2.9)$$

where X, Y, Z are the coordinates of a 3D point from the virtual object (that we will define as \mathbf{P} , in homogeneous coordinates) and c_x, c_y, c_z are the coordinates that represent the center of projection of the camera (also in homogeneous coordinates).

Finally, we need to search the optimal point on this curve. For this step, two solutions were presented given rise to the third constraint. One using the Reflection Law (also known as Snell's Law) other using the Fermat Principle. Schemes for both solutions are represented in Fig. 2.1(a) and 2.1(b).

Using the Fermat Principle we can search the reflection point on the mirror (that we will define as \mathbf{R}) where the distance travelled by the light is minimal. This means that the reflection point is the point that minimizes the sum of the distance between the camera center and the reflection point on the mirror (represented by $d_{\mathbf{RC}}$) and the distance between the same reflection

point on the mirror and the 3D point (represented by $d_{\mathbf{PR}}$). The scheme for this solution is represented in Fig. 2.1(b).

On the other hand, using the Snell's Law the optimal reflection point on the mirror (that we will define as \mathbf{R}) will be the point that minimizes the absolute value of the difference between θ (angle between the normal vector - \mathbf{n} and the reflection vector - \mathbf{v}_r) and ϕ (angle between the normal vector - \mathbf{n} and incident vector - \mathbf{v}_i). The scheme for this third constraint is represented in Fig. 2.1(a).

2.4 Algorithms to deal with the Occlusions

The problem of occlusions in computer graphics is a very important problem when projecting 3D objects to a 2D image, an example is presented at Fig. 2.2. This problem is based on the intersection of objects, or parts of the objects, and the definition of each one is closer or farther from the user point of view.

To solve this situation three well known methodologies have been adopted for the past decades: the painter's algorithm, the Z-Buffer (also known as Depth Buffer) and the A-Buffer. In the remaining section it will be briefly described each of these solutions to make clear the key idea of each solution.

The painter's algorithm is based on the drawing of the scene from the back to the front, which means that, all the polygons in the scene will be sorted by descending order of their depth value and then displayed by this order.

The Z-Buffer is based on a buffer strategy which saves the color and the depth of each pixel of the 3D objects that will be displayed in the scene. When rendering to the image if an object is at same pixel as other object the depth values in the buffer (corresponding to both objects) are analyzed and the pixel that is closest from the viewer is chosen.

The A-Buffer follows the same strategy as the Z-Buffer to handle depth, but additionally adds an anti-aliasing strategy. For each pixel a set of sub-pixels are associated. When an occlusion between objects occurs, each color of the overlapped pixels are stored in sub-pixels and the final color will be the sum of all sub-pixel colors. This algorithm is an evolution of the Z-Buffer since transparencies can thus be considered.

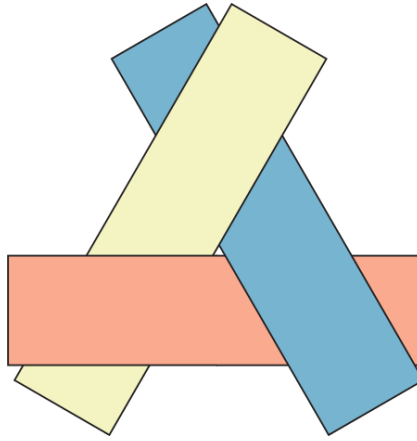


Fig. 2.2: Example of an occlusion problem, presented at [19], between three rectangles.

2.5 Illumination and shading models

When dealing with illumination and shading of objects in computer graphics sometimes these concepts are misunderstood. When assuming the illumination of an object we are taking into account the intensity of light (color) in a specific place of the object. Usually, in computer graphics, to find the intensity of light it is used the Phong reflection model, illustrated in 2.3(a). However, it is important to present the geometric model for a better understanding.

The illumination of a surface depends of four directional vectors (unitary vectors) and two angles:

- The normal vector, \mathbf{N} , which represents the normal to the surface;
- The vector \mathbf{L} that is the vector that points to the light source;
- The vector \mathbf{R} that is the vector that represent the reflected direction of incident ray;
- The vector \mathbf{V} that is the vector that points to the viewer;
- The angle θ is the angle between the vector that points to the light source and the normal vector. The angle between the normal vector and the reflected vector must be the same;
- The angle ϕ is the angle between the reflected ray and the vector that points to the viewer

To compute the illumination in a pixel the Phong Lighting equation is used. This equation is expressed by the equation (4.10) at Sec. 4.6. It is important to mention that when dealing with illumination, the light source is a very important factor for a correct illumination of the scene.



Fig. 2.3: In Fig. (a) it is illustrated the Phong reflection model (the description of the figure is done throughout the section). In Fig. (b) we show an example from [19] for the teapot object when applying different techniques of shading. On the left image a flat shading example and, on the right image, a smooth shading example (Gouraud shading).

We can then define two types of light sources: positional and positional directional. To clarify we can associate a positional light as the sun that illuminate the scene at every directions, when using a positional directional light source we can describe it as a restriction of the rays that are emitted by the sun in a specific direction (normally restricted by a cone of illumination). For a light source three properties can be defined: ambient, diffuse and specular. Each of them has a different contribution to define the color of the vertex. For a better understanding an example is presented in Fig. 2.4. Now, that the properties of the light source were presented, it is relevant to say that the material properties (same properties as the light source, but additionally have an emission and a shininess property) as the global ambient light, have the same importance as the light source. This will be thus described in full detail further in Chap.4 in the illumination section (4.6).

As for the shading, there are two kinds of shading models, in computer graphics, that can be used: flat and smooth. Associated to the flat model is the flat (or constant) shading and associated to the smooth is the Gouraud and Phong shading (which is different from the reflection model). An example is illustrated in Fig. 2.3(b).

When dealing with flat shading, it is used the reflection model to calculate the intensity of light in a single point of each polygon (that forms a 3D object) and the color of the entire polygon will be the same. This is an easy and fast method to use, although, the final result can be, depending on the object, imperfect and innacurate.

For the Gouraud and Phong shading techniques the intensity of color in a polygon is deter-

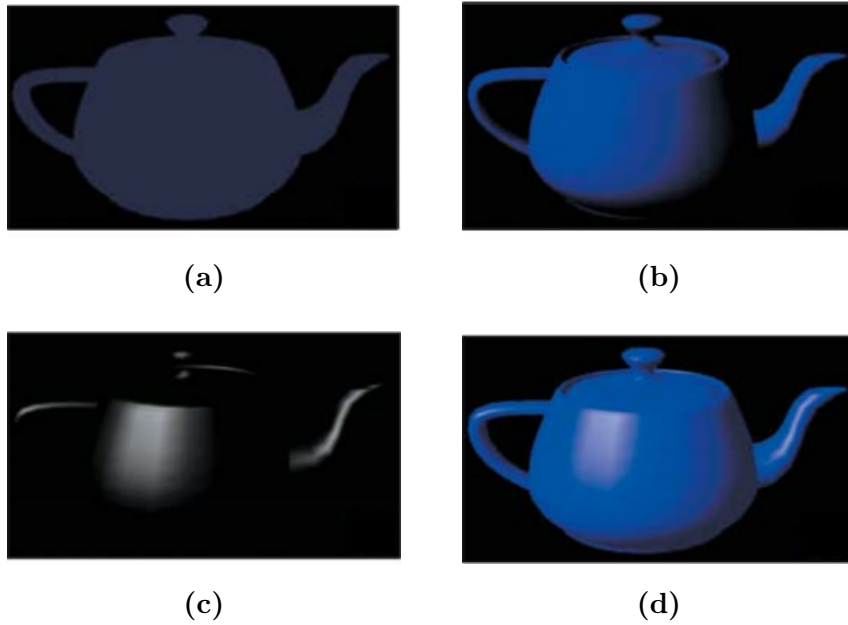


Fig. 2.4: In Figs. (a), (b), (c) and (d) are represented the influence of each property individually and all together respectively. The first image shows the influence of the ambient light property, the second image shows the influence of the diffuse light property. the third image shows the influence of the specular light property and the fourth (and final) image shows the combination of all the properties. These images were extracted from [19].

mined by an interpolation. When using the Gouraud technique, first it is calculated the normal at each vertex of the polygon then, the illumination model is applied to each one of these vertices, and, finally, an interpolation of the color of each vertex is done to give the color to the polygon surface. In the Phong technique a normal to each pixel is calculated and then it is applied the illumination model to give the intensity of color for each pixel. It is easy to understand that the smooth shading models lead to more detailed objects, giving an almost real representation of the 3D virtual object.

2.6 Augmented reality applications

In this section, some well known techniques of augmented reality applications and methods for the conventional central systems are presented. It is important to say that these techniques cannot be applied to our system, since the framework proposed is for non-central catadioptric systems and, according to our knowledge, this is the first time that this problem is addressed.

2.6.1 Fournier *et al.* method

Fournier *et al.* stated [14] that the merging of a real video image with an image generated by a computer emphasize the lack of utility of which one. It was presented that some problems should be solved to make the merging of the images with success, such as, problems of common viewing parameters, common visibility and common illumination. The proposed solution can be called as Computer Augmented Reality. In this work some techniques for approximating the common global illumination for real video images and computer generated images were proposed, assuming that some elements of the scene geometry of the real world and common viewing parameters were already known. Since the real image is a projection of the exact solution for the global illumination in the real world, an approximation for the global illumination of the merged image is made, by making the real video image part of the solution to the global illumination computation. So, a method was built with the following sequence of steps:

- The physical objects that belong to the real scene must be replaced by covering boxes (representation on the left and center images of Fig. 2.5);
- The intensity of the real video image was used as the initial surface radiosity of the visible part of the boxes;
- The surface reflectance of the boxes were approximated by subtracting an estimation of the intensity of illumination based on the concept of ambient light;
- A global illumination computation, using a classic computation of the radiosity, was used to render the computer generated surface, taking into account the real environment where the surface will be represented. The estimation of the global illumination was also used to calculate the amount of correction, in the image intensity, needed for surfaces of the real image;

The viewing parameters were extracted by interactive matching of the synthetic scene with the real video images. The visibility is determined by the relative position of the "blocks" that represent the real objects and the computer generated objects, and by a moving virtual light that was inserted. The results of the merging are presented in Fig 2.5 on the right image.

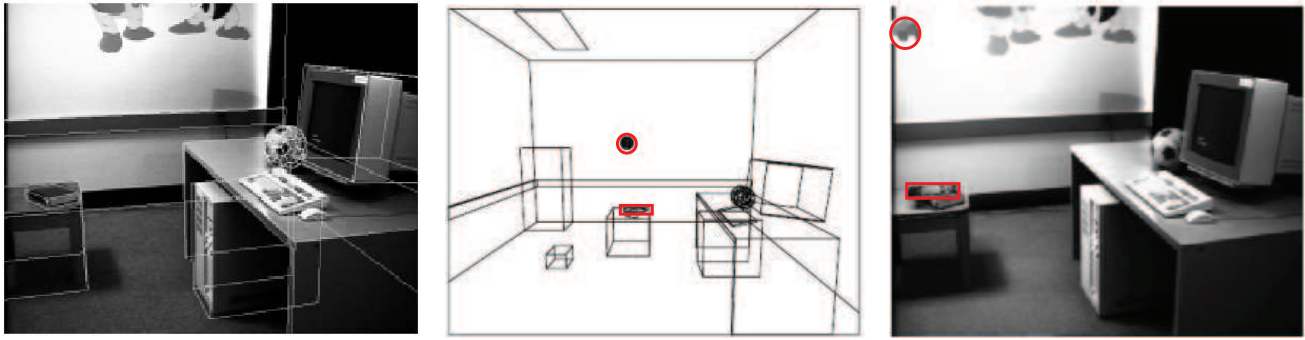


Fig. 2.5: On the left it is represented the real video image with physical object replaced by boxes, on the center figure it is represented only the boxes, instead of the physical objects, and the two augmented objects (highlighted in the figure). On the right it is represented the final result in the merging of the augmented objects (highlighted in the figure) with the real video image. These images were extracted from [14].

2.6.2 Debevec method

Debevec [10] presented a method that calculate the radiance and global illumination applied to the scene in order to display additional objects (virtual objects) with correct lighting in a real scene. Instead of using computer-generated (virtual) light sources, this method is based on an high dynamic range image-based model of the scene to compute the illumination on the new generated objects.

In order to solve the illumination problem, this method divides the scene in three distinct components: the distant scene, the local scene, and the new generated objects (synthetic objects). For a better understanding of this division we will describe each one individually:

- The first component is the distant scene, which is assumed to be the farther part of the scene. This component will not be significantly influenced by the addition of a virtual object to the real scene (taking into account the user perception);
- The second component is the local scene, which is the part of the scene that will be affected by the addition of new objects in the real scene. This component of the scene is composed by information of the estimated reflectance model, in order to receive information about the shadows and the reflected light from the new generated objects;
- The third component of the scene are the virtual generated objects;

In their work it is presented a solution to calculate the correct radiance of the scene, using

a technique previously proposed by Debevec in a previous work ([11]). Using this technique an accurately illumination of the virtual object can be made. To achieve this goal radiance maps must be obtained. These radiance maps were obtained by photographing a mirrored ball with different exposure levels, as illustrated in Fig. 2.6(a). Results using radiation information from indoor and outdoor environments were obtained as illustrated in Fig. 2.7(a). However, another solution in this work is presented using a light probe. A radiance map was obtained using two light probe images, from different positions, with different exposure levels, as illustrated in Fig. 2.6(b). The result obtained using this technique is presented in Fig. 2.7(b). The presented method performs a robust solution for the problem of merging virtual/synthetic objects into real scenes. The main steps of the proposed method are based on:

- A light-based model of the scene is constructed through an approximate geometric model;
- A light probe is used to calculate the incident illumination at the location where the virtual objects will be placed;
- A global illumination solution is then computed taking into account the light coming from the distant scene and interacting with the local scene and the synthetic objects (light that is reflected back to the distant scene is ignored);
- Using a differential rendering technique, the global illumination is then composited into a photograph of the scene;

A final result, taking in consideration all the main steps mentioned above (represented in Figs. 2.8(a),(b),(c),(d), with the application of the presented method is illustrated in Fig. 2.8(e).

2.6.3 Sato *et al.* method

Sato *et al.* [31] presented a solution to merge virtual objects, taking into account their correct shadows and shadings, into an image of a real scene. With the presented method it can be measured automatically the radiance distribution of the real scene and use it for merging the virtual objects into the real scene.

For this method the scene is assumed to be static, which means that movements of objects are not considered and only shadows in the plane, where the object is placed, are taking into account.

This means that shadows between the virtual objects and real objects are not considered using this method.

The proposed method can be described as a sequence of steps:

- The world coordinate system is defined in the real scene and the transformation matrix between the world coordinate system and the image coordinate system is estimated;
- A geometric model of the scene is constructed from two omnidirectional images (obtained using two CCD (charge-coupled device) camera with fisheye lenses, presented in the last two images of the Figs. 2.9(a) and (b)) by using an omnidirectional stereo algorithm;
- Then radiance of the scene is computed from a sequence of omnidirectional images taken with different shutter speeds and mapped in the geometric model, referred in the previous step. The radiance distribution mapped on the geometric model is used for rendering virtual objects that are merged with the image of the real scene;

The main advantage provided by this method is that, for a complex radiance distribution, the virtual objects can be merged into the real scene with shadings and shadows effects almost similar to the reality.

To prove the efficiency of the proposed method, tests for an indoor and outdoor environments, with different virtual objects, were made. The results obtained in these tests are presented in Figs.2.10(a) and (b) respectively.

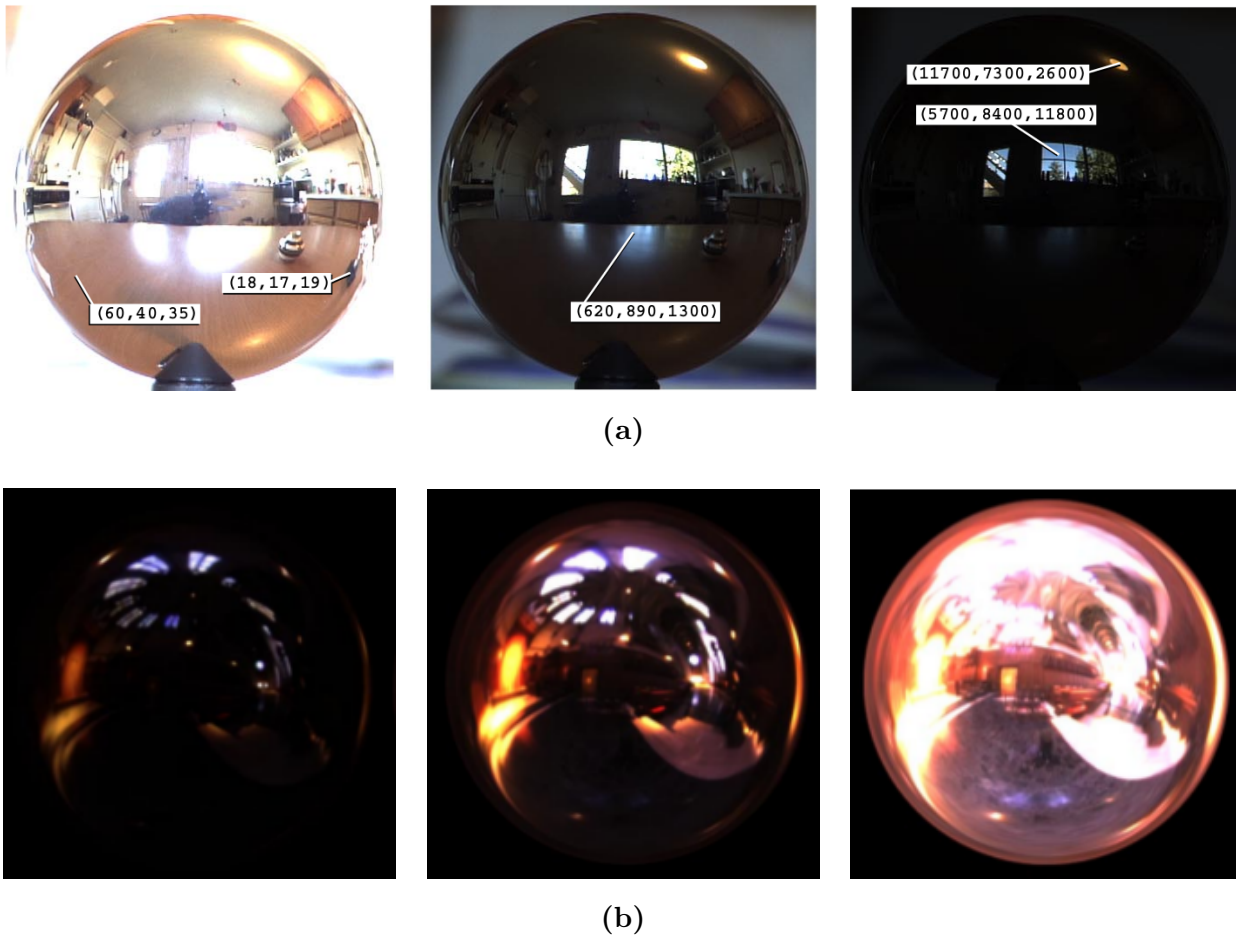
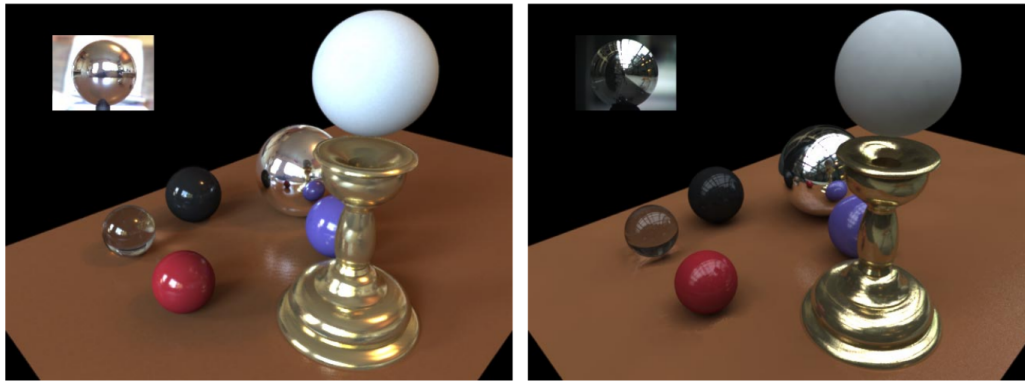


Fig. 2.6: In Fig. (a) are represented the images acquired by photographing a mirrored ball with three exposure levels: +0, -3.5 and 7.0 stops respectively. In the last image (right side) it can be observed that the full dynamic of the scene was obtained without saturation, which means that intensity, color and direction of all forms of the incident light can be recorded. In Fig.(b) are represented the images used to obtain the full dynamic range environment map of the scene obtained by two light probe images, taken with ninety degrees from distance of each other using three exposure levels are: -4.5, 0 and 4.5 stops. These images were extracted from [10].



(a)



(b)

Fig. 2.7: In Fig.(a) are represented two results from an indoor and an outdoor environment using the radiance map at top corner of both images. In Fig.(b) it is presented the result taking into account the lighting information obtained in Fig.2.6(b). These images were extracted from [10].

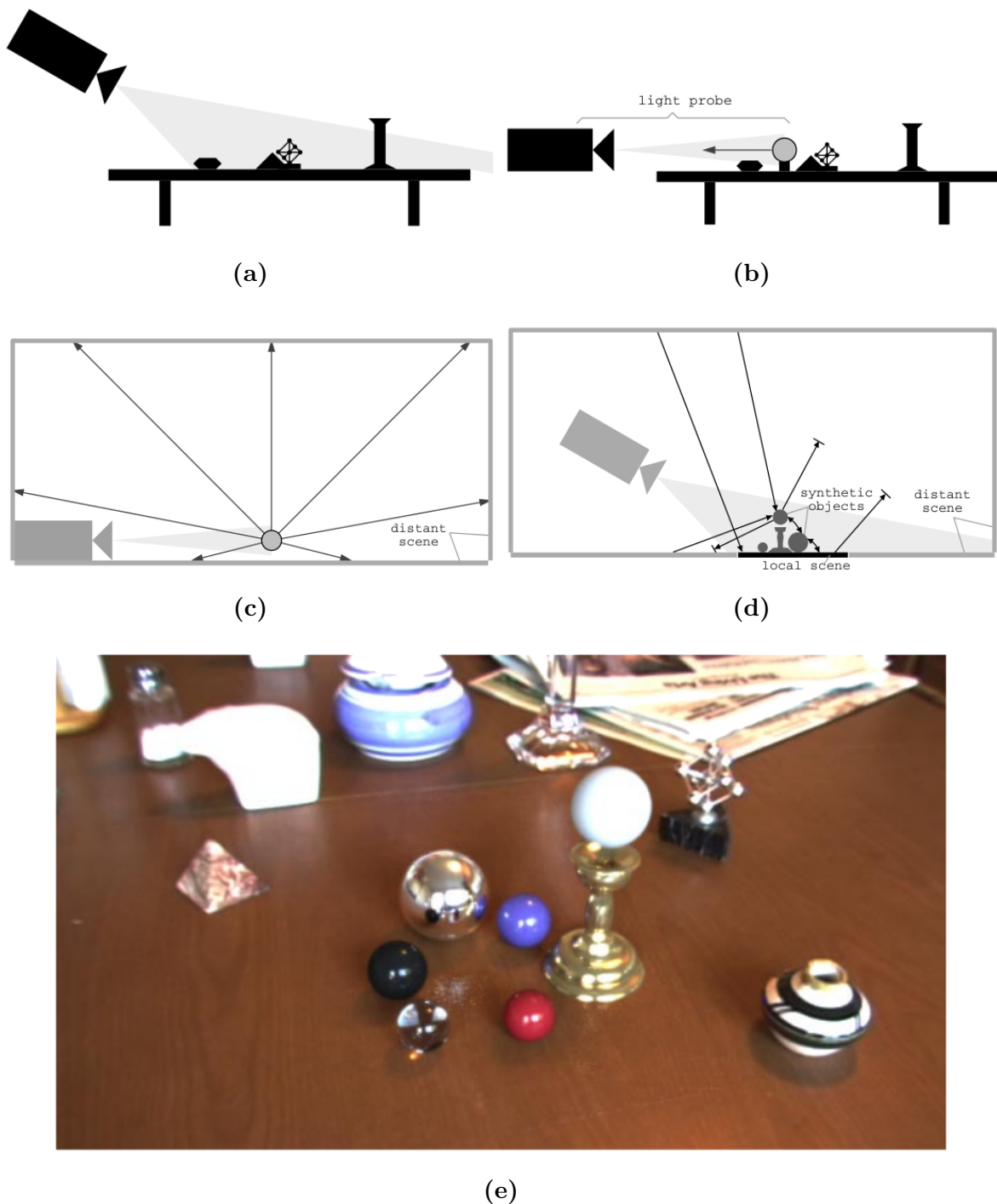


Fig. 2.8: In Fig.(a) it is represented the scheme that represents the acquisition of the background photograph. In Fig.(b) it is represented the scheme in using a light probe to obtain the incident radiance near to the location where the virtual objects will be placed. In Fig.(c) it is represented the scheme to construct the light-based model. In Fig.(d) it is represented the scheme for the computation of the global illumination solution. In Fig.(e) it is illustrated the result obtained taking into account the steps mentioned in previous sub-images. These images were extracted from [10].

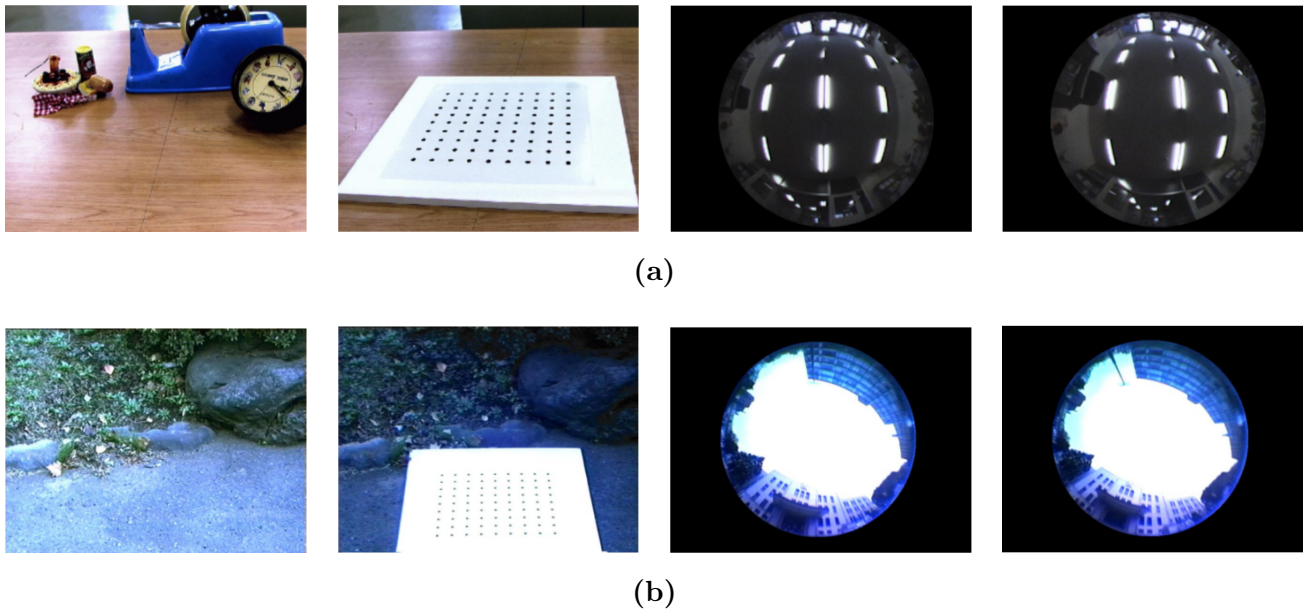


Fig. 2.9: In Fig.(a) are presented the images of the real environment, the calibration image and the two omnidirectional images (obtained with a camera with a fisheye lens) respectively, for an indoor environment. In Fig.(b) are presented the same type of images for an outdoor environment. These images were extracted from [31].

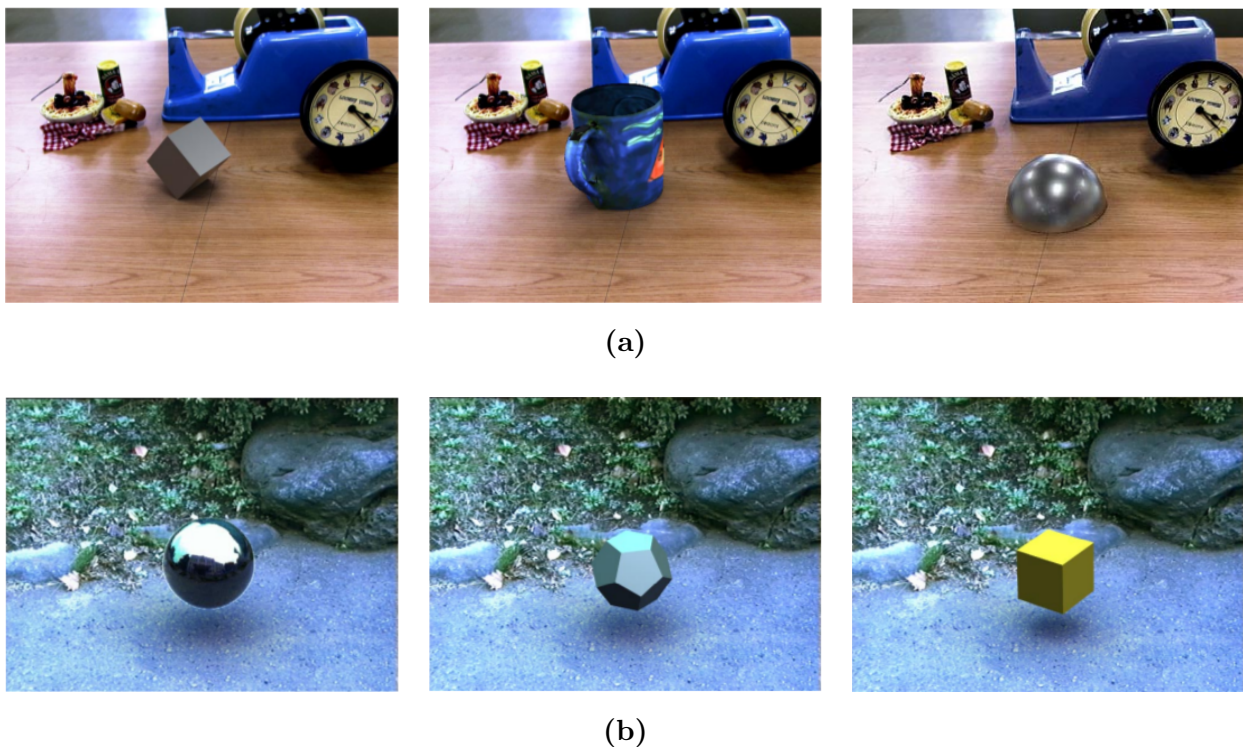


Fig. 2.10: In Fig.(a) are presented the results obtained with the presented method for an indoor environment. In Fig.(b) are presented the results obtained with the presented method for an outdoor environment. These images were extracted from [31].

Chapter 3

Proposed Method

In this chapter it is presented the approach followed to build our framework. For a better understanding of the proposed solution, a representative pipeline was made. The content of this pipeline will be further presented in the remaining sections of this chapter.

3.1 The Proposed Pipeline

The proposed pipeline is divided in two stages: pre-processing and realtime stage, see Fig. 3.1. As described in the introduction chapter 1, for our goal a single number of steps must be considered, such as: camera calibration, 3D Object triangulation, texture mapping, “QI Projection”, occlusions, illumination and the display. In this approach it is assumed that the 3D objects are rigid and static. Thus, to avoid unnecessary computations, the first three steps (that compose the pre-processing stage) can be computed *a priori*. The remaining steps must be computed in a realtime stage, in way to improve the computational time of each step. In the following two subsections the composition of each stage of the pipeline is analyzed.

3.1.1 Pre-Processing Stage

The pre-processing stage is composed by two steps: camera calibration and 3D segmentation of the object. It is well known that all imaging devices are represented by the mapping between pixels and 3D straight lines. The camera calibration consists in the estimation of the parameters that represent this mapping. Since general non-central catadioptric cameras are being considered,

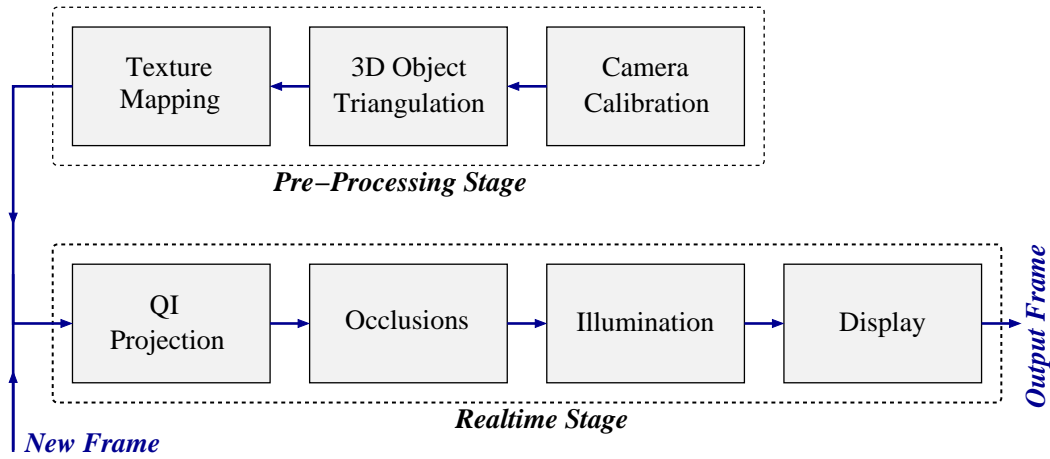


Fig. 3.1: Representation of the proposed pipeline for the application of augmented reality in non-central catadioptric camera models.

the goal is to get the camera intrinsic parameters, the mirror parameters and the transformation matrix between the camera and mirror (in Sec. 4.1 it is presented a detailed description of this step).

The second step of the pre-processing stage is related to the segmentation of the 3D virtual object. As described in the introduction, the virtual object must be decomposed into small 3D features to, later, be projected into the 2D image plane. If the 3D features are small enough, the distortion effects will be neglectable and can be ignored. Similar to most of state-of-the-art approaches, it was used the segmentation of the 3D virtual object in 3D triangles (triangulation). In addition to the 3D segmentation, it was considered the application of textures to the 3D segments according to the 3D virtual object (triangulation). These steps are further analyzed in Sec. 4.2 and 4.3 , respectively.

3.1.2 Realtime Stage

The realtime stage corresponds to the methods that have to be computed each time a new image frame is received. This stage is composed by the following four steps: “QI Projection”, occlusions, illumination and display.

Since very small 3D triangles are used and distortion effects on these triangles are being ignored, their image (texturized) will just depend on the projection of three 3D points to the 2D image plane, that represent the three vertices of the 3D triangles. The “QI Projection” step is about the projection of the 3D points to the 2D image plane (forward projection). Note

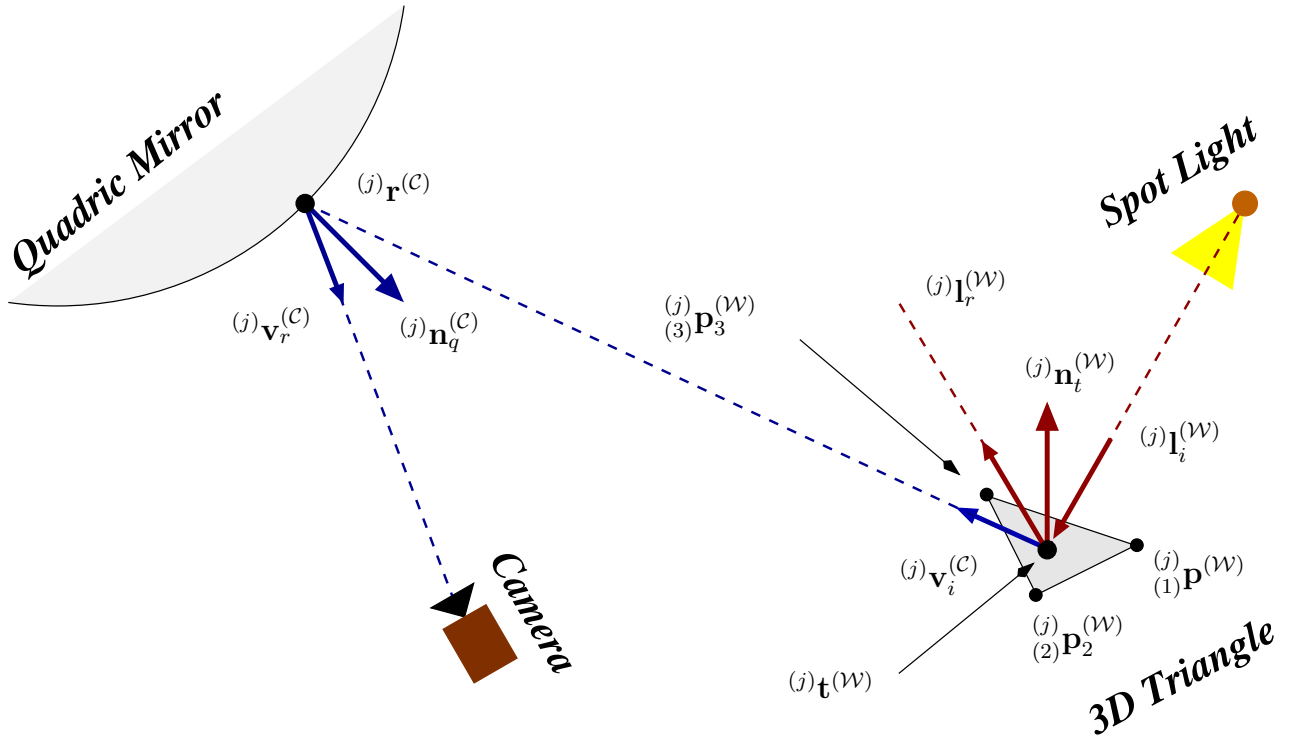


Fig. 3.2: Representation of the projection of 3D point to the 2D image plane of a general catadioptric camera and the illumination model used.

that, since non-central catadioptric systems are being used, this step is not as easy as for the conventional perspective projection. In addition, one has to verify if the coordinate system of the virtual object is aligned with the camera's coordinate system. To deal with this case, before the computation of the projection of 3D points to the 2D image plane, it must be done the computation of the pose (of the camera). This is very important when dealing with a mobile camera. This step is further analyzed in Sec. 4.4.

Since it is considered the projection of small pieces of the 3D objects to the 2D image plane, it is fundamental to understand if these pieces are overlapped and, if they are, which of them are in front. It is proposed a solution based on painter's algorithm. This method was chosen because of its simplicity and efficiency. However, since it is used non-central catadioptric imaging systems, this methodology have to be reformulated, taking into account the geometry of the imaging device. The goal of painter's algorithm is to organize the 3D triangles as a function of the distance of these triangles to the camera system. The problem is then completely solved by displaying the 2D triangles using this order. The main difference between the proposed method and the conventional painter's algorithm is related to the definition of "point of view". If for the conventional perspective camera one can use the camera center (also called the effective view

point) [18] as a “point of view” for all 3D triangles and thus the distance between the triangle and the camera is computed as the distance between each 3D point and the camera center, for our case this cannot be applied. Note that it is being considered non-central imaging devices, which means that doesn’t exist a single point where all the 3D projection lines intersect. As a result, to compute the distance between the 3D triangles and the camera system it is considered the distance between the triangle (the mass center of the triangle is used) and the respective 3D reflection point on the mirror (see Fig. 3.2). The purpose of this adaptation is to understand when a triangle is occluded by another one, when viewed from the reflector surface. Regard that the direction of the viewpoint (defined as the point on the caustic [37]) passes by the reflection point. This problem is fully addressed in Sec. 4.5.

When regarding illumination/shading, there are several proposed approaches [19, Chapter 6]. However, these methods were derived for imaging devices that can be modeled by the central perspective camera and, as a result, cannot be applied to the proposed framework. For this step, it was again derived a very simple methodology. To avoid distortion aberrations (see the previous paragraphs) it was used a large number of very small triangles, so that the complete illumination of 3D triangles can be analyzed as a single point of illumination. For simplicity, the complete illumination of the 3D triangle was considered equal to the illumination of the mass center of the respective 3D triangle. Hence, to compute the illumination parameters, it was used the well known Phong’s reflection model. The equation parameters applied to our case (non-central catadioptric systems) are analyzed in Sec. 4.6. Variations of Phong’s or Gouraud’s methodologies [19, Chapter 6] could also be used, however, since very small 3D triangles are being considered, the variation of the illumination will be neglectable which means that these methods would bring unnecessary computation time.

Now that all the required information (projection of the 3D triangulated virtual object to the 2D image including occlusions and illumination properties), the fourth step is about the display of the object in the current frame. As it was already mentioned in Sec. 1.1, from the introduction chapter, the OpenGL API was used in this step.

Chapter 4

Detailed Description of the Pipeline

In this chapter it will be described, in full detail, the steps that build the proposed pipeline of Fig. 3.1, presented in the previous chapter.

4.1 Camera Calibration

As it was previously described, in this dissertation it is considered the use of non-central catadioptric cameras formed by a central perspective camera and a quadric mirror (see Fig. 3.2). This step is about the calibration of this system. To calibrate the non-central catadioptric camera the method proposed by Perdigoto and Araujo [30] was followed. For that, one has to consider: the calibration of the central perspective camera, which means, estimate the camera calibration $\mathbf{K} \in \mathbb{R}^{3 \times 3}$ such that ${}^{(j)}\mathbf{v}_r^{(c)} \sim \mathbf{K} {}^{(j)}\mathbf{r}^{(c)}$ (where ${}^{(j)}\mathbf{v}_r^{(c)}$ and ${}^{(j)}\mathbf{r}^{(c)}$ are the projection ray of the perspective camera and the respective 3D point on the mirror); and the quadric mirror matrix $\mathbf{\Omega} \in \mathbb{R}^{4 \times 4}$ and $\mathbf{H}^{(CO)} \in \mathbb{R}^{4 \times 4}$ such that

$${}^{(j)}\mathbf{r}^{(c)T} \mathbf{H}^{(CO)T} \mathbf{\Omega} \mathbf{H}^{(CO)} {}^{(j)}\mathbf{r}^{(c)} = 0, \quad (4.1)$$

where $\mathbf{H}^{(CO)}$ is the matrix that transforms a point from the quadric to the camera coordinate system. In our system, we are using a spherical mirror. So, the rotation parameters of the matrix have no influence in the solution. Taking this factor into account our matrix $\mathbf{H}^{(CO)}$ can be described as:

$$\mathbf{H}^{(CO)} = \begin{bmatrix} 1 & 0 & 0 & t_x \\ 0 & 1 & 0 & t_y \\ 0 & 0 & 1 & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Now that we have all the required parameters, we can use the Snell's law to compute the 3D projection direction

$${}^{(j)}\mathbf{v}_i^{(c)} = {}^{(j)}\mathbf{v}_r^{(c)} - 2 \left({}^{(j)}\mathbf{v}_r^{(c)T} {}^{(j)}\mathbf{n}_q^{(c)} \right) {}^{(j)}\mathbf{n}_q^{(c)}, \quad (4.2)$$

where ${}^{(j)}\mathbf{n}_q^{(c)}$ is the normal vector at the 3D quadric mirror point ${}^{(j)}\mathbf{r}^{(c)}$.

4.2 3D Object Triangulation

As mentioned above, it was decided to do the segmentation of the 3D virtual object in triangles. To avoid distortion aberrations, only very small triangles have been considered (the effects of distortion in the image will be neglectable). In this section two different approaches to triangulate the 3D objects were used.

For the parallelepiped, considering that the coordinates of the 3D virtual object are known, as each independent face of the parallelepiped define a plane (where the plane is defined by two axis of coordinates) we can use to solve this problem the Delaunay algorithm [12] for 2D objects. To obtain our goal we used the Delanauy algorithm provided by the OpenCV library. In order to have triangles with similar dimensions, the 2D points (on each face) were uniformly chosen. After the computation of each 2D points that form the vertices of each triangle it is only necessary to add the static coordinate (that belong to the axis that do not define the plane), and we have the 3D points that define each and every triangle.

For the bunny and the Buddha objects the triangulation was already provided by the Stanford repository. So, for these objects, in order to obtain each triangle that form both objects, we only needed to import the 3D coordinates from the PLY (which means Polygon File Format or Stanford Triangle Format) file and form each triangle according to the information provided in it. It is worthy to insist in the fact that it was not the whole object that was imported to the scene, rather only the 3D coordinates that specify each triangle are imported. In addition, in the experiments it were used the Stanford bunny and the happy Buddha with approximately 70k 3D triangles each.

So, let us consider that our object is already triangulated with N 3D triangles. Thus, we know the coordinates of the three 3D points that define the N triangles. Formally

$$\left\{ {}^{(j)}\mathbf{p}_{(1)}^{(W)}, {}^{(j)}\mathbf{p}_{(2)}^{(W)}, {}^{(j)}\mathbf{p}_{(3)}^{(W)} \right\}, \quad \text{for } j = 1, \dots, N \quad (4.3)$$

where ${}^{(j)}\mathbf{p}_{(i)}^{(W)}$ are the coordinates of the i^{th} vertex of the j^{th} triangle.



Fig. 4.1: In this figure it is presented the two images used to apply as textures to our cube faces. On the left side, the image applied to the vertical faces and, on the right side, the image applied to the horizontal faces.

4.3 Texture Mapping

Let be considered, for example, the texturization of the 3D virtual parallelepiped. Using the triangulation defined in Sec. 4.2 , the vertices that form each and all triangles (3D point $\binom{(j)}{(i)}{\mathbf{p}}^{(W)}$) are already known. Since, as already stated at the previous section, we considered each face of the parallelepiped individually, the texture associated to each triangle can be obtained through a simple conversion and association of the 3D world coordinates of each face to the respective texture coordinates of the 2D image that we want to use as texture for the face. This procedure can be done at the pre-processing stage because it is assumed that the coordinates associated to each triangle will not change during the execution of the framework.

To clarify the idea that we are proposing to apply a texture to the parallelepiped faces, a sequence of simple steps are presented:

- Load the 2D image as a texture to apply to the face, after the load of the image the coordinates are automatically normalized (between [0,1] for both 2D axis of coordinates);
- Since we have the triangulation (obtained in the previous section) of each face, we normalize the coordinates of the object (for the axis of coordinates where exists variation) to have the texture image and the face (where the texture will be applied) in the same system of coordinates;
- Finally, we just need to choose a piece (a small triangle) of the loaded texture and paste it in the corresponding position in the face;
- All the previous steps are done for each face that represent the 3D parallelepiped;

For the textures we used the images illustrated at Fig. 4.1. The results of this texturization step will be further presented in the experimental results chapter.

For the Stanford bunny and the happy Buddha, since the goal of the dissertation is not to map a triangulated texture to an irregular surface, a single color (white) texture was applied to all the 3D triangles that define the object.

4.4 QI Projection

In this step, the goal is to compute the projection of 3D triangles (that define the 3D virtual object) to the 2D image plane (see Fig. 4.2). As it was previously described, effects of the triangles distortion are ignored. As a result, this can be computed by projecting the three vertices (that form each triangle) to the image. In order to find a solution for this step, two important computations must be considered: computation of the camera pose and the computation of the reflection point in the mirror. The computation of the camera pose is addressed in this step because it will be a relevant factor for the computation of the reflection point, as it will be further presented in this subsection.

Thus, let us consider the projection of a 3D world point to the 2D image of a non-central catadioptric camera. Since the calibration of the perspective camera is known (see Sec. 4.1), the first thing that must be verified is that the coordinates of the 3D object are represented in the camera coordinates system. In Fig. 3.2, it is intentionally used the superscripts (\mathcal{W}) and (\mathcal{C}) to represent features in the world (in which the 3D object was defined) and the camera coordinate systems, respectively.

As a result, it has to be computed the rigid transformation $\mathbf{H}^{(CW)} \in \mathbb{R}^{4 \times 4}$ between both coordinate systems:

$$\begin{matrix} (j) \\ (i) \end{matrix} \mathbf{p}^{(C)} = \mathbf{H}^{(CW)} \begin{matrix} (j) \\ (i) \end{matrix} \mathbf{p}^{(W)}, \text{ for all } i \text{ and } j. \quad (4.4)$$

To solve the computation of the camera pose matrix ($\mathbf{H}^{(CW)}$) it was used the method proposed by Miraldo and Araujo at [23]. Now, for all the vertices of the triangles $\begin{matrix} (j) \\ (i) \end{matrix} \mathbf{p}^{(C)}$ (in the coordinates of the camera system), the goal is to compute the respective reflection point in the mirror $\begin{matrix} (j) \\ (i) \end{matrix} \mathbf{r}^{(C)}$. The ‘‘QI Projection’’ method proposed by Gonalves [16] was the solution followed. Although, in way to use the ‘‘QI Projection’’ method some computations must be done previously.

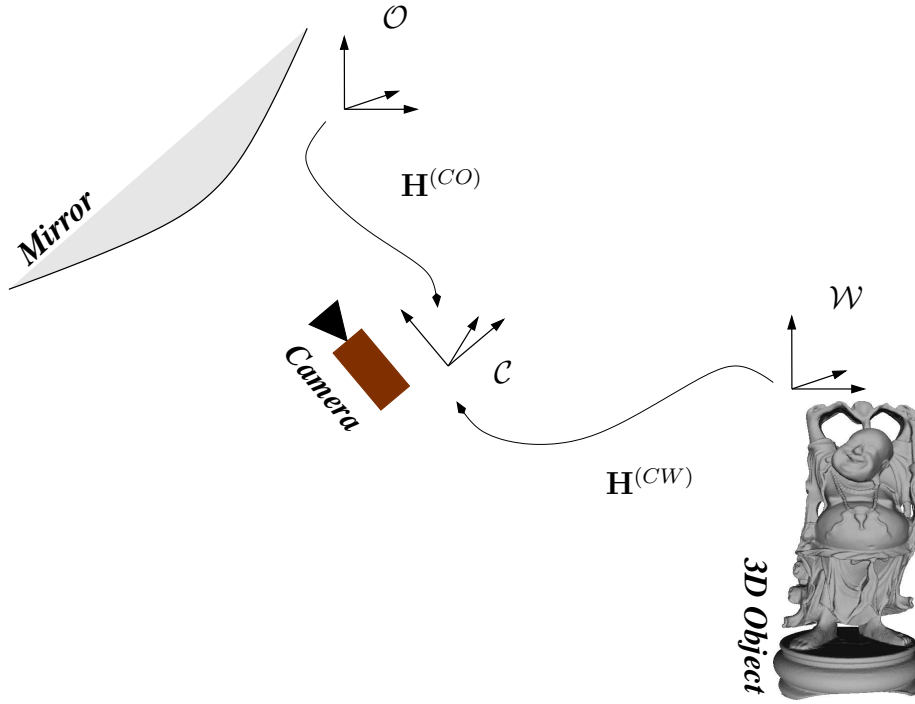


Fig. 4.2: Representation of the problem of the estimation of the 3D virtual object to the 2D image. First we need to know the rigid transformation to do the mapping of each 3D point from the world coordinates to the camera coordinates system, $\mathbf{H}^{(CW)}$, obtained by the camera pose estimation. Then, the transformation from the camera coordinate system to the quadric coordinate system, using the inverse of $\mathbf{H}^{(CO)}$, is given by the calibration step (Sec. 4.1). The next stage is the computation of the reflection point in the mirror through the “QI Projection” method. Then, we just need to apply the transformation of $\mathbf{H}^{(CO)}$ to obtain the point in the mirror coordinates and using the intrinsic parameters matrix obtain the corresponding 2D point in the image.

First, we need to convert the points from the camera coordinate system to the mirror system:

$$\begin{pmatrix} j \\ i \end{pmatrix} \mathbf{p}^{(O)} = \mathbf{H}^{(CO)-1} \begin{pmatrix} j \\ i \end{pmatrix} \mathbf{p}^{(C)}, \text{ for all } i \text{ and } j. \quad (4.5)$$

Then, we use the “QI Projection” method to obtain the reflection point in the mirror:

$$\begin{pmatrix} j \\ i \end{pmatrix} \mathbf{r}^{(O)} = \text{QIProj} \left(\begin{pmatrix} j \\ i \end{pmatrix} \mathbf{p}^{(O)} \right), \text{ for all } i \text{ and } j. \quad (4.6)$$

Now, we just need to convert from the quadric coordinate system to the camera coordinate system:

$$\begin{pmatrix} j \\ i \end{pmatrix} \mathbf{r}^{(C)} = \mathbf{H}^{(CO)} \begin{pmatrix} j \\ i \end{pmatrix} \mathbf{r}^{(O)}, \text{ for all } i \text{ and } j. \quad (4.7)$$

Using these methodology, it can now be assumed that the projection of each and all the 3D triangles that form the 3D virtual object is done. These triangles will be denoted as

$$\left\{ \begin{pmatrix} j \\ (1) \end{pmatrix} \mathbf{u}, \begin{pmatrix} j \\ (2) \end{pmatrix} \mathbf{u}, \begin{pmatrix} j \\ (3) \end{pmatrix} \mathbf{u} \right\}, \text{ where } \begin{pmatrix} j \\ i \end{pmatrix} \mathbf{u} \sim \mathbf{K} \begin{pmatrix} j \\ i \end{pmatrix} \mathbf{r}^{(C)}, \forall j = 1, \dots, N \wedge i = 1, \dots, 3. \quad (4.8)$$

Algorithm 1: Reformulation of painter’s algorithm.

Let ${}^{(j)}\mathbf{p}^{(c)}$ be the 3D coordinates of the i^{th} vertex of the j^{th} triangle and N the number of existing triangles:

for $j = 1$ to N **do**

 Compute ${}^{(j)}\mathbf{t}^{(c)}$ using (4.9);

 Compute ${}^{(j)}\mathbf{r}^{(c)}$ using (4.6) and (4.7);

 Set ${}^{(j)}\xi$ as the distance between ${}^{(j)}\mathbf{r}^{(c)}$ and ${}^{(j)}\mathbf{t}^{(c)}$;

end

Sort all the triangles by descendent order using the computed ${}^{(j)}\xi$, for all $j = 1, \dots, N$;

where ${}^{(j)}\mathbf{u}$ denotes the i^{th} vertex of the j^{th} triangle, in homogeneous coordinates.

4.5 Occlusions

As it was previously described, a solution is proposed based on painter’s Algorithm. However, since we are dealing with non-central imaging devices, conventional solutions cannot be used. From the set of 2D triangles defined in the equation (4.8), one needs to verify the occlusions. The goal of painter’s algorithm is to draw the virtual scene/object (in our case triangles obtained from the equation (4.8)) from the back to the front, or, in other words, from the farthest to the closest triangle. So, in way to solve the occlusion’s problem we used a solution based on the distance from the 3D triangles and their reflection on the quadric mirror. To compute the distance of each 3D triangles j to the catadioptric camera it was considered the depth between the triangle’s mass center

$${}^{(j)}\mathbf{t}^{(c)} = \frac{{}^{(j)}\mathbf{p}^{(c)} + {}^{(j)}\mathbf{p}^{(c)} + {}^{(j)}\mathbf{p}^{(c)}}{3}, \quad (4.9)$$

and its reflection point. Now, the computations presented at the previous section must be done to find the reflection point of the mass center point, ${}^{(j)}\mathbf{r}_t^{(c)}$, in the camera coordinate system. For simplicity and since the triangles are small enough to neglect the distortion of the mirror, only the mass center was used to represent all the triangle. Thus, the depth associated to the mass center is the depth of all the triangle.

This can be easily verified by the analysis of our geometry image of the system, presented in the Fig. 3.2.

The solution for this step is formalized in algorithm 1 which means that, after the application of this algorithm, we have all the triangles ordered by descending order and ready to be displayed

to the scene, in way to avoid successfully the occlusion problem.

4.6 Illumination

The traditional approach to this problem is to express the illumination as a composition of several light sources and their interactions with the physical materials, so as to compose the global illumination effect. Since this is an unstudied problem (illumination in non-central catadioptric systems for augmented reality), our goal is to adopt a simple and efficient approach. Another relevant factor is that we have the projection of a 3D object to 2D image so we cannot use the default functions provided by the OpenGL API (which would be easier to solve), so, we have taken into account the mathematics of lighting to solve this problem. As a result, we decided to use, the well known in computer graphics, Phong's reflection model [33, Chapter 5]. Our illumination equation for the j^{th} triangle is, then, expressed by

$$\begin{aligned}
 {}^{(j)}I^{(\text{ch})} = & \overbrace{K_e^{(\text{ch})} + G_a^{(\text{ch})} K_a^{(\text{ch})}}^{\tilde{I}^{(\text{ch})}} + \\
 & + \underbrace{\sum_{k=1}^M \text{spot}_k \left({}^{(k)}L_a^{(\text{ch})} K_a^{(\text{ch})} + f_k \left({}^{(k)}L_d^{(\text{ch})} K_d^{(\text{ch})} \left(\max \left\{ -\frac{{}^{(j)}\mathbf{l}_i^{(C)T} {}^{(j)}\mathbf{n}_t^{(C)}}{{}^{(k)}\mathbf{l}_i^{(C)}} \right\}, 0 \right) + {}^{(k)}L_s^{(\text{ch})} K_s^{(\text{ch})} \left(\max \left\{ {}^{(j)}\mathbf{v}_i^{(C)T} \frac{{}^{(j)}\mathbf{l}_i^{(C)}}{{}^{(k)}\mathbf{l}_i^{(C)}}, 0 \right\} \right)^{sh} \right) \right)}_{{}^{(j)}\tilde{I}_k^{(\text{ch})}}
 \end{aligned} \tag{4.10}$$

for all color channels, such that: M is the number of spotlights; $K_a^{(\text{ch})}$, $K_d^{(\text{ch})}$, $K_s^{(\text{ch})}$, $K_e^{(\text{ch})}$ and sh are ambient, diffuse, specular, emission, shininess material color intensities; $G_a^{(\text{ch})}$ is the global ambient light property ((ch) denotes the color channel); ${}^{(k)}L_a^{(\text{ch})}$, ${}^{(k)}L_d^{(\text{ch})}$, ${}^{(k)}L_s^{(\text{ch})}$ are the ambient, diffuse and specular intensities of the k^{th} spotlight; boolean parameter f_k is used to control whether a triangle is illuminated or not; and spot_k is the spotlight effect on the j^{th} triangle. Since the distance between the 3D points on the scene (that define the 3D virtual object) are small relatively to the distance between these points and the spotlights, we used one as the attenuation factor, ignoring this effect on the equation.

Regarding the control parameter f_k , when the angle formed from the 3D triangles' normal direction ${}^{(j)}\mathbf{n}_t^{(C)}$ and the incident light ray $-{}^{(j)}\mathbf{l}_i^{(C)}$ is higher than $\pi/2$, the triangle is considered not illuminated, thus setting this control parameter to zero ($f_k = 0$), otherwise, triangles are illuminated and $f_k = 1$.

Although it is relevant to mention that beside both diffuse and specular components are

Algorithm 2: Proposed illumination algorithm.

Let ${}^{(j)}\mathbf{p}^{(C)}$ be the 3D coordinates of the i^{th} vertex of the j^{th} triangle, N the number of existing triangles and ${}^{(k)}\mathbf{d}_{sl}^{(C)}$ the direction of the spotlight:

for $j = 1$ to N **do**

 Compute the normal ${}^{(j)}\mathbf{n}_t^{(C)}$ using (4.11);

 Compute the mass center ${}^{(j)}\mathbf{t}^{(C)}$ using (4.9);

 Compute the reflection point ${}^{(j)}\mathbf{r}_t^{(C)}$ of ${}^{(j)}\mathbf{t}^{(C)}$ using (4.6) and (4.7);

 Compute the visualization vector ${}^{(j)}\mathbf{v}_i^{(C)}$, (4.13);

 Set ${}^{(j)}I^{(\text{ch})} = \tilde{I}^{(\text{ch})}$, see (4.10);

for $k = 1$ to M **do**

 Compute ${}^{(j)}\mathbf{l}_i^{(C)}$ using (4.12);

if $\left(- {}^{(j)}\mathbf{l}_i^{(C)T} {}^{(j)}\mathbf{n}_t^{(C)} \right) \leq 0$ **then**

$f_k = 0$;

else

$f_k = 1$;

end

if $\max \left\{ {}^{(j)}\mathbf{l}_i^{(C)T} {}^{(k)}\mathbf{d}_{sl}^{(C)}, 0 \right\} \geq {}^{(k)}C^{\text{te}}$ **then**

$spot_k = \max \left\{ {}^{(j)}\mathbf{l}_i^{(C)T} {}^{(k)}\mathbf{d}_{sl}^{(C)}, 0 \right\}^{\mathcal{E}}$;

else

$spot_k = 0$;

end

 Add ${}^{(j)}I^{(\text{ch})} = {}^{(j)}I^{(\text{ch})} + {}^{(j)}\check{I}_k^{(\text{ch})}$, see (4.10);

end

end

rejected, the ambient component of the light will still be considered to the illumination of the triangle. The variable $spot_k$ controls the effect of the cutoff angle of the spotlight (controlled by a predefined constant ${}^{(k)}C^{\text{te}}$) considering a certain triangle.

The first step of the proposed method contains some similarities to the proposed occlusions' algorithm. We compute the mass center point of each triangle (4.9) and compute the intensity of the RGB components using the light equation. In addition to these variables, we have to take into account four additional directions (unitary): vector ${}^{(j)}\mathbf{l}_i^{(C)}$ is the direction that points from the object point to the k^{th} light source (assumed to be known); vector ${}^{(j)}\mathbf{n}_t^{(C)}$ is the normal to the j^{th} triangle:

$${}^{(j)}\mathbf{n}_t^{(C)} = \frac{\begin{pmatrix} {}^{(j)}\mathbf{p}^{(C)} - {}^{(j)}\mathbf{p}^{(C)} \\ (1) \end{pmatrix} \times \begin{pmatrix} {}^{(j)}\mathbf{p}^{(C)} - {}^{(j)}\mathbf{p}^{(C)} \\ (2) \end{pmatrix}}{\left| \begin{pmatrix} {}^{(j)}\mathbf{p}^{(C)} - {}^{(j)}\mathbf{p}^{(C)} \\ (1) \end{pmatrix} \times \begin{pmatrix} {}^{(j)}\mathbf{p}^{(C)} - {}^{(j)}\mathbf{p}^{(C)} \\ (2) \end{pmatrix} \right|}; \quad (4.11)$$

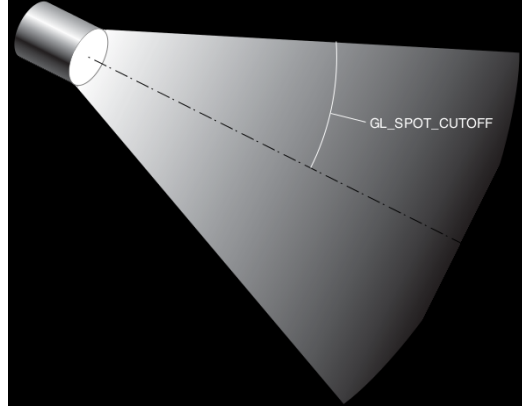


Fig. 4.3: Example of a spotlight illustrated at [33] showing the cutoff angle that will affect the light intensity in each triangle.

vector ${}^{(j)}\mathbf{l}_r^{(C)}$ is the k^{th} reflected direction on the mass center point ${}^{(j)}\mathbf{t}^{(C)}$ that can be computed using the Snell's law:

$${}^{(j)}\mathbf{l}_r^{(C)} = {}^{(j)}\mathbf{l}_i^{(C)} - 2 \left({}^{(j)}\mathbf{l}_i^{(C)T} {}^{(j)}\mathbf{n}_t^{(C)} \right) {}^{(j)}\mathbf{n}_t^{(C)}; \quad (4.12)$$

and vector ${}^{(j)}\mathbf{v}_i^{(C)}$ is the direction that points from ${}^{(j)}\mathbf{t}^{(C)}$ to the viewer's direction:

$${}^{(j)}\mathbf{v}_i^{(C)} = \frac{{}^{(j)}\mathbf{r}_t^{(C)} - {}^{(j)}\mathbf{t}^{(C)}}{\left| {}^{(j)}\mathbf{r}_t^{(C)} - {}^{(j)}\mathbf{t}^{(C)} \right|} \quad (4.13)$$

(note that, since we are using non-central catadioptric cameras, most of the novelty of the proposed approach is in the use of ${}^{(j)}\mathbf{v}_i^{(C)}$). In addition, one has to consider the k^{th} spotlight direction ${}^{(k)}\mathbf{d}_{sl}^{(C)}$, which is also assumed to be known.

As explained in the realtime stage subsection (Chapter 3), the computed components for the mass center of each triangle will be associated to all the area of the respective triangle (the triangles are small enough to neglect distortion). So, in terms of computer graphics analysis, we can state that the implementation of a flat (or constant) shading model was chosen to solve our problem.

Chapter 5

Experimental Results

The goal of this chapter is to evaluate the proposed framework. To ensure that all the steps of the pipeline (Fig. 3.1) work as expected, it were included specific tests for the occlusion, texture mapping and illumination steps. For the experiments, it was used a non-central catadioptric camera formed with a perspective camera and a spherical mirror.

As described in the introduction, our framework was tested using three 3D virtual objects: a virtual parallelepiped, the Stanford bunny [34] and the happy Buddha. While the bunny and the Buddha objects are already triangulated with a solid white texture (wich means that the pre-processing stage is no longer needed), for the virtual parallelepiped it must be taken into account the triangulation and texturization of the object. The methods used were described in Secs. 4.2 and 4.3.

The most important steps of the pipeline are the ones associated with the realtime stage. The first step of this stage correspond to the projection to the 2D image plane of the 3D triangles (that define the 3D virtual objects). Results, for both objects, are shown in Figs. 5.1(a) and 5.2(a). To test the occlusion step, a result using the virtual parallelepiped is presented, as it can be seen from Fig. 5.1(b), the overlap of multiple faces is a relevant and important factor (since the object will clearly lose the original shape, providing bad results). In addition, we also show the results of the solution for the occlusion step applied to the single color texturized faces at Fig. 5.1(c) and for texturized faces with a pattern at Fig. 5.1(d). The results are shown in Fig. 5.1(d). Note that, in this figure, it can be easily seen the effect of the object distortion caused by the geometry of the non-central catadioptric camera model (see for example the image of the top face of the parallelepiped).

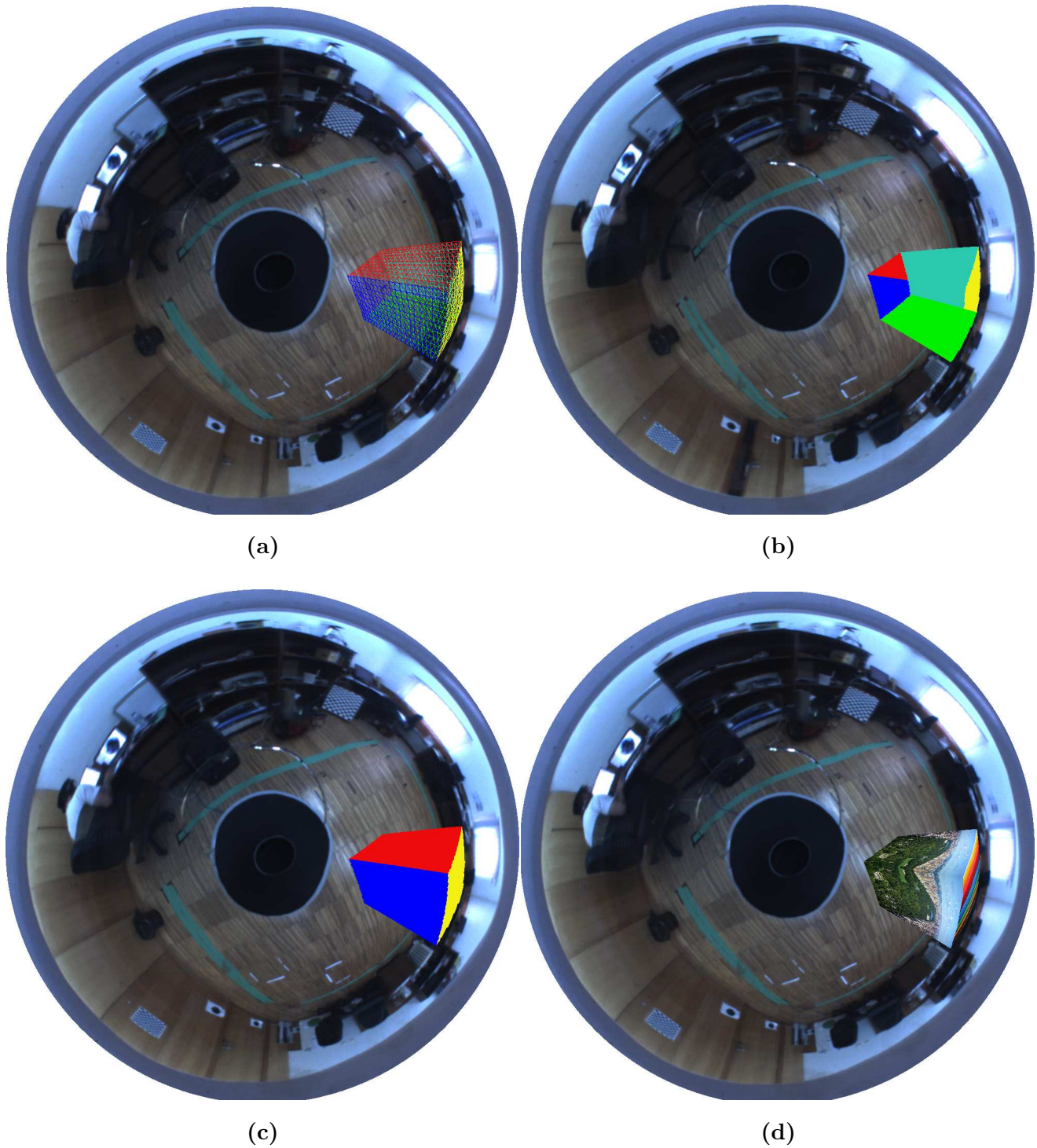


Fig. 5.1: In this figure we show the results of the application of the various steps of the pipeline applied to the virtual parallelepiped. Fig. (a) represent the projection of the 3D triangles to the image, which correspond to the “QI Projection” step of the pipeline. The goal of Fig. (b) is to show the effects of the occlusions between the faces of the virtual parallelepiped. In Figs. (c) and (d) we show the result for the occlusion step using solid color textures, for each face, and using more detailed textures respectively.

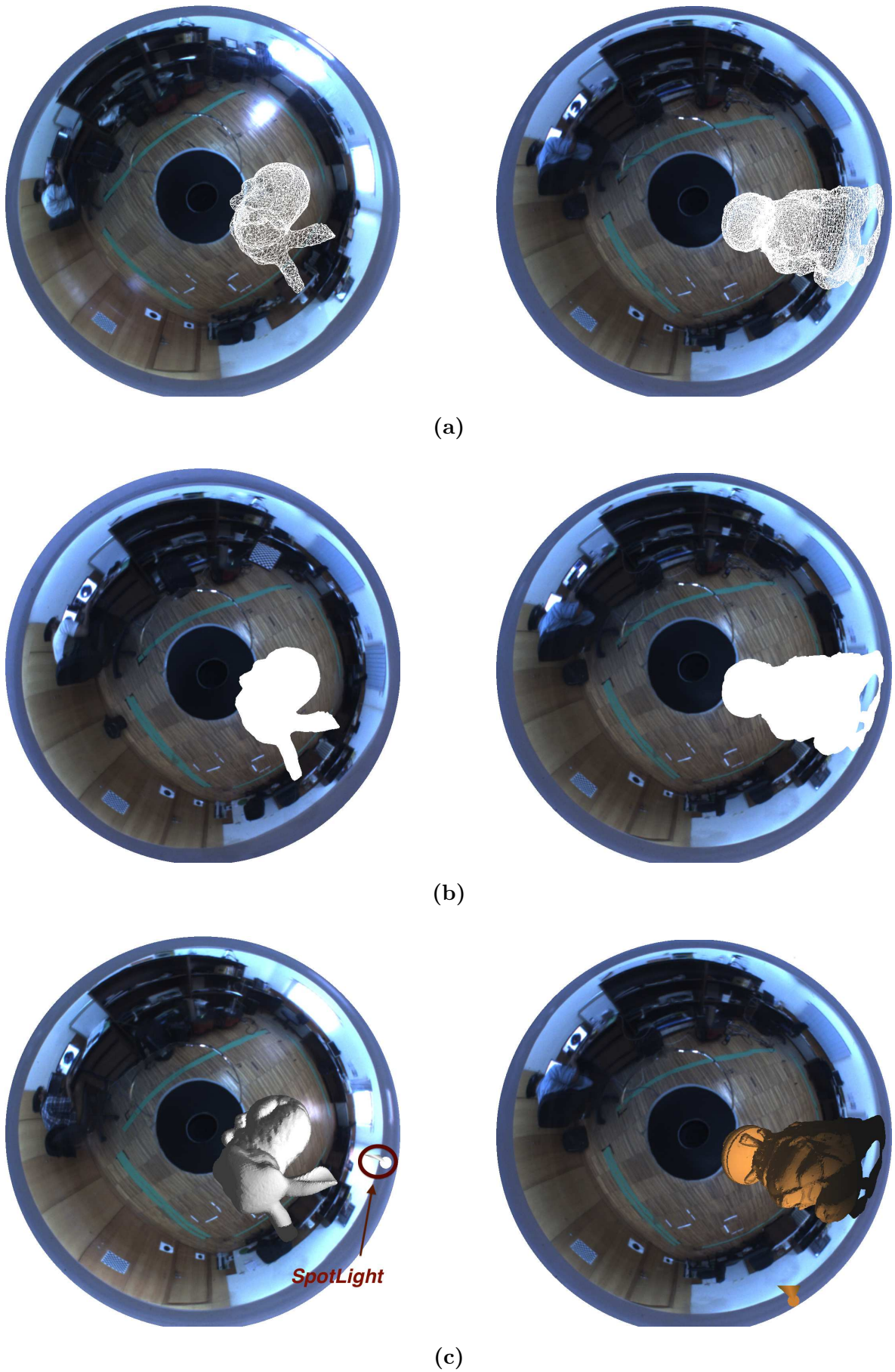


Fig. 5.2: In this figure we show the results of the application of the steps of the pipeline to the bunny and the Buddha objects. Fig. (a) represent the projection of the 3D triangles to the image (“QI Projection” step of the pipeline). Figs. (b) and (c) show the influence of the illumination in the objects.

As it was mentioned in the introduction, without illumination, the projection of a 3D virtual object with a solid color will be represented by a blob. This is shown in Fig. 5.2(b). On the other hand, as we can see from Fig. 5.2(c), illumination will give shape to the projection of the virtual object. For the material intensities/parameters (parameters expressed in the equation (4.10)) a silver cover was used, which is a well-known and a standard material used in computer graphics applications. Additionally, our light source will be treated as a spotlight that moves freely around the scene. We define the light parameters, $L_a^{(ch)}$, $L_d^{(ch)}$ and $L_s^{(ch)}$, to be a white spotlight for the bunny and the parallelepiped objects and a gold spotlight for the buddha object. For the global ambient light property ($G_a^{(ch)}$), since the goal of this dissertation was not to evaluate the intensity of the light rays that have influence in our workspace, different intensities for each of the RGB's components were used to obtain good results for each object. To understand what is happening in the scene and to check if the illumination is working properly (from a certain point of view), a visual representation was made.

In addition to these experiments, a set of images were acquired when a moving spotlight is applied to our scene. The results are shown in Figs. 5.3, 5.4 and 5.5 proving that all steps, from the proposed pipeline, are working well and reproducing good results for each one of the virtual objects used.

Since, the equation of light that was used in our framework can have the influence, in the 3D object projected to the 2D image, in more than one spotlight in the system, an example with multiple moving spotlights (three light sources each one associated with different colors in different positions) was used, giving the results presented in Fig. 5.6. The global light and material properties, for this example, were the same used for the previous experiments, although the light parameters for each spotlight were adjusted in way to give different emission colors from the light source.

For all the experiments above, the spotlight direction points to the center of the arena, that is the place where the virtual objects will be represented. We take this decision in way to prove some consistency between the results obtained. However, it is important to mention that this could be modified since it is a position which can be easily redefined.

A final experiment was done using a moving robot. We made this test for the bunny object, since, as we can see by the previous results obtained it would work for every 3D virtual object. In this case, the spotlight was positioned on the top side of the robot, pointing to the center of the object. An arena was built in 3D to show the movements of the robot in the real world. In

Fig. 5.7 we can see that the light moves and illuminates the object as it was supposed to do. We also can observe that the object is static in the scene, which means, by other words, that the object is always at the same position (center of the arena), only the view acquired by the mirror is different (because the robot is moving). With this experiment we also prove that the camera pose estimation is working well, as it was supposed. We additionally used an auxiliar camera to show what is happening in the real world.

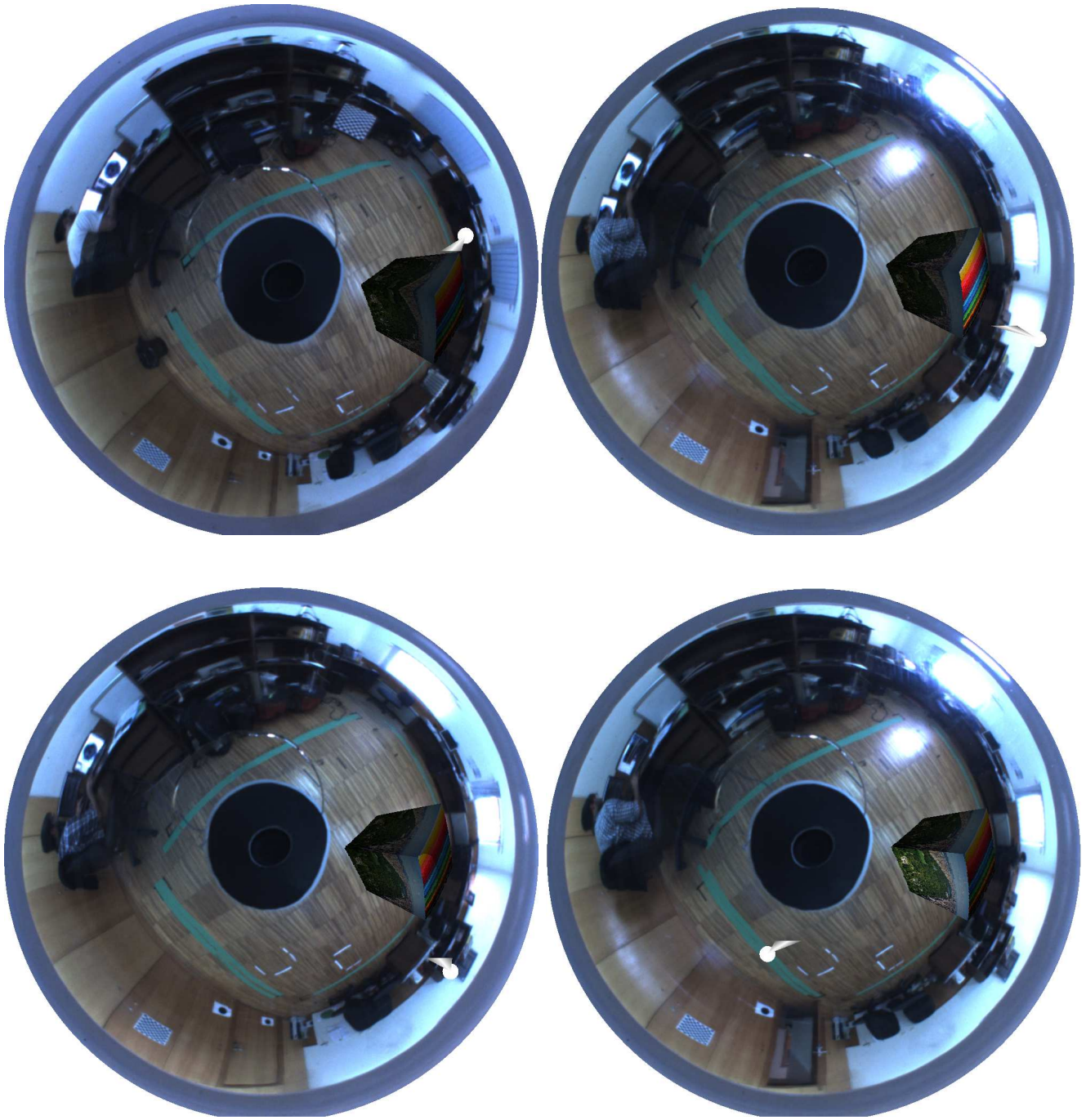


Fig. 5.3: In this figure we show a set of frames in which we apply the proposed framework, considering a moving spotlight affecting the virtual parallelepiped. To obtain this result, a circular movement was applied to the spotlight to show that our solution, for the illumination step, is working correctly.

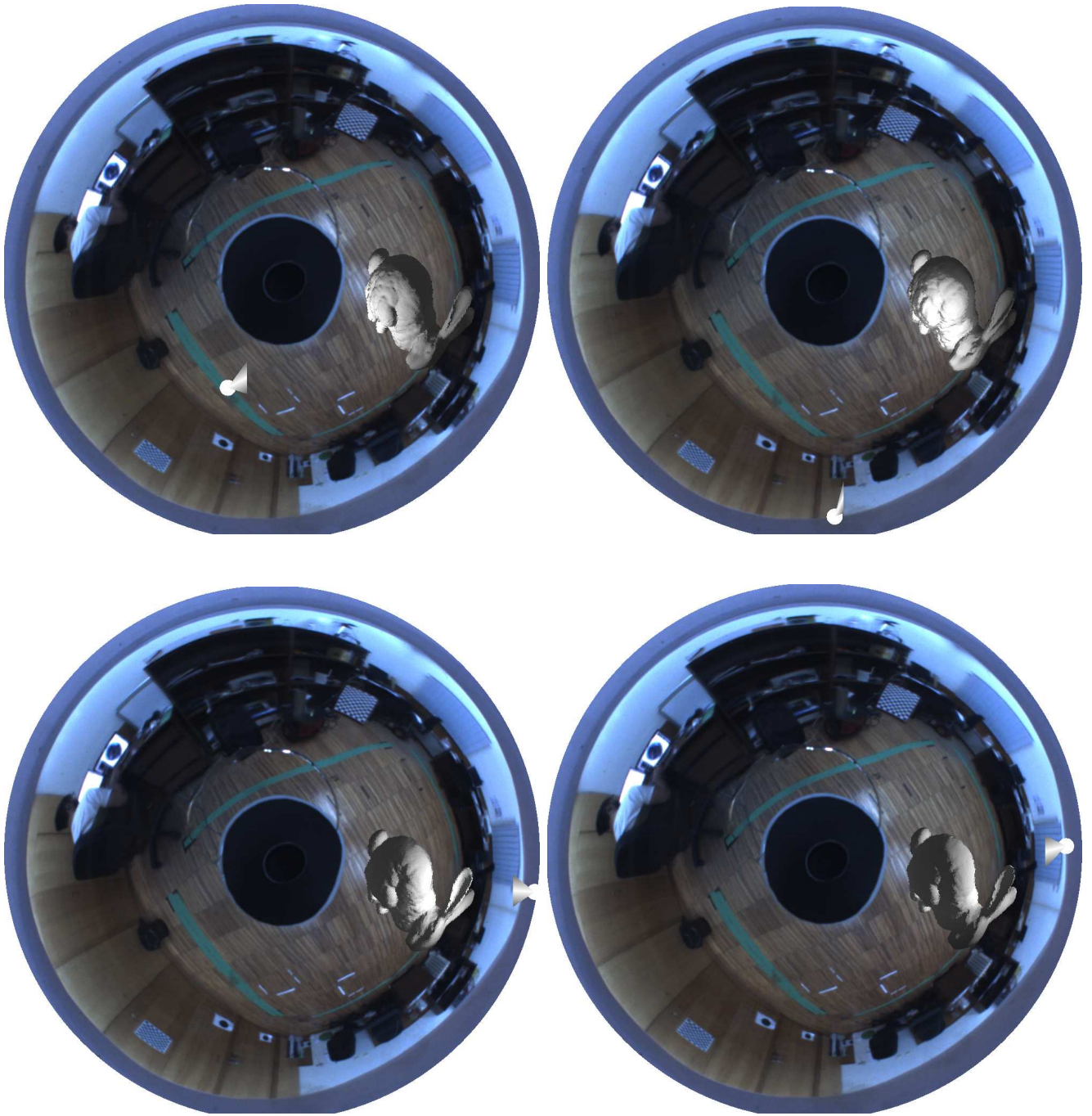


Fig. 5.4: In this figure we show a set of frames in which we apply the proposed framework, considering a moving spotlight affecting the Stanford bunny. To obtain this result, a circular movement was applied to the spotlight to show that our solution, for the illumination step, is working correctly.

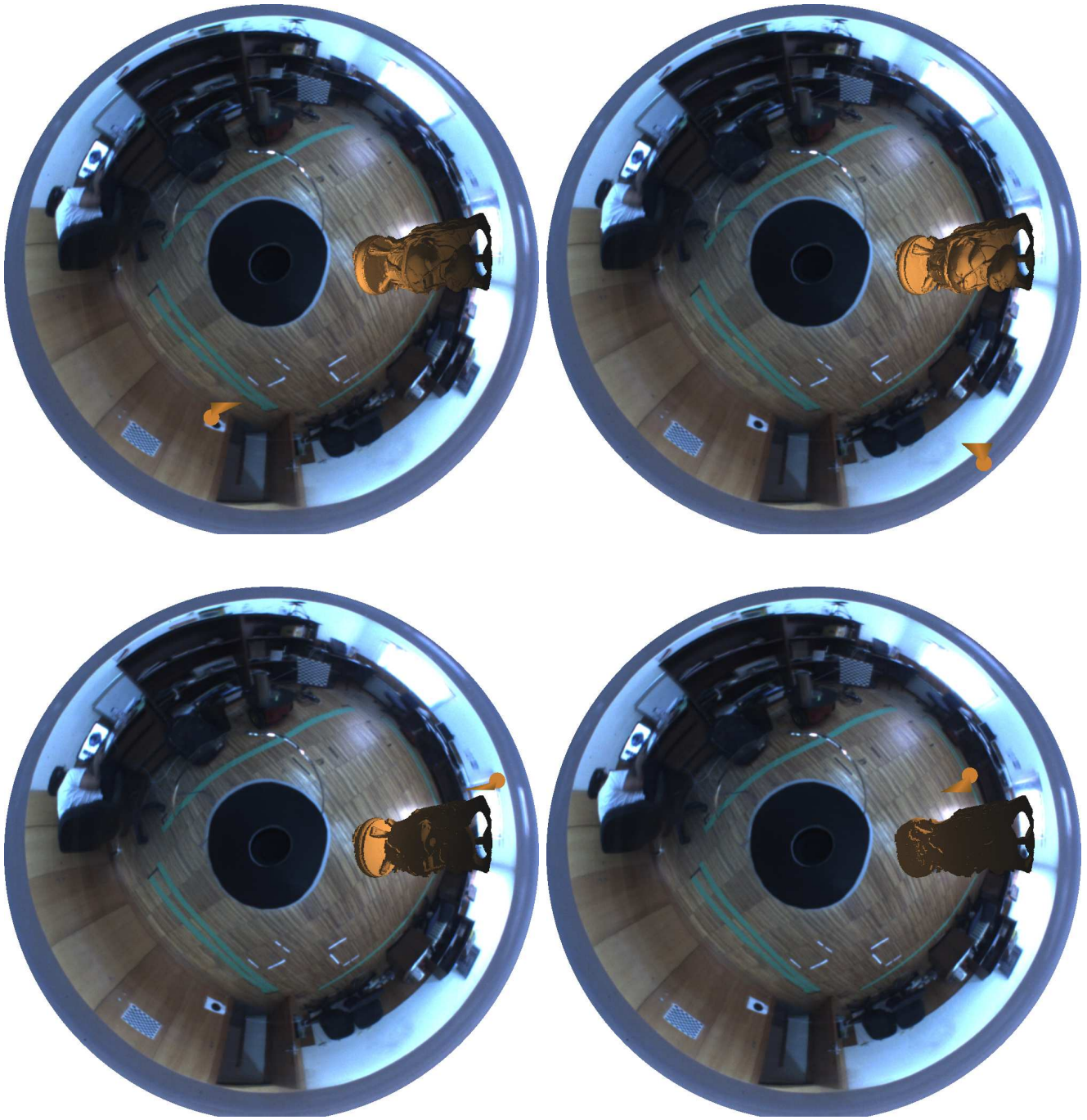


Fig. 5.5: In this figure we show a set of frames in which we apply the proposed framework, considering a moving spotlight affecting the Happy Buddha. To obtain this result, a circular movement was applied to the spotlight to show that our solution, for the illumination step, is working correctly.

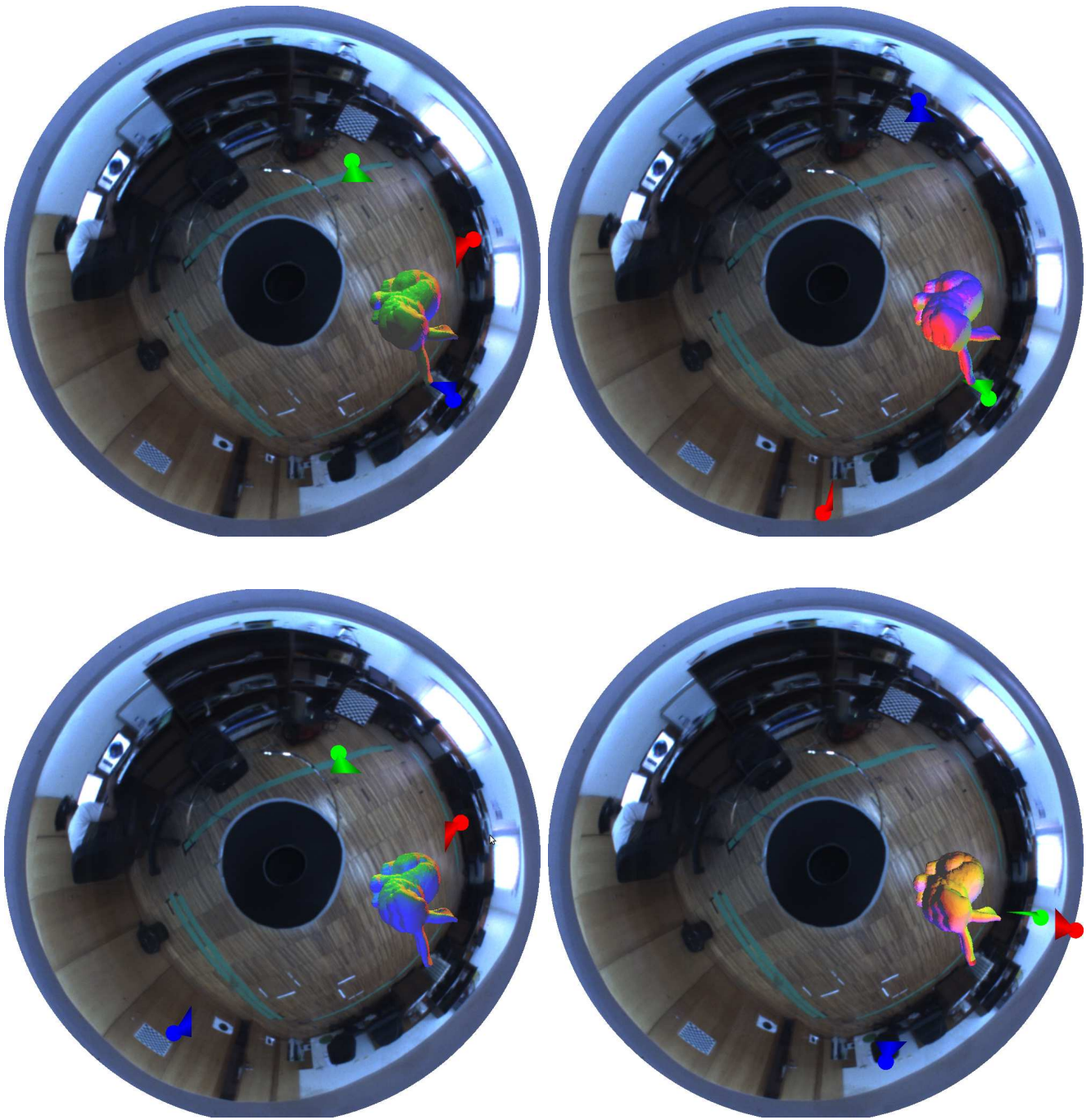


Fig. 5.6: In this figure we show a set of frames in which we apply the proposed framework, considering three moving spotlights with different colors (blue, green and red) affecting the Stanford bunny. To obtain this result three different movements were applied to each one of the spotlights to show that our solution, for the illumination step, is working correctly with the use of multiple spotlights in our framework.

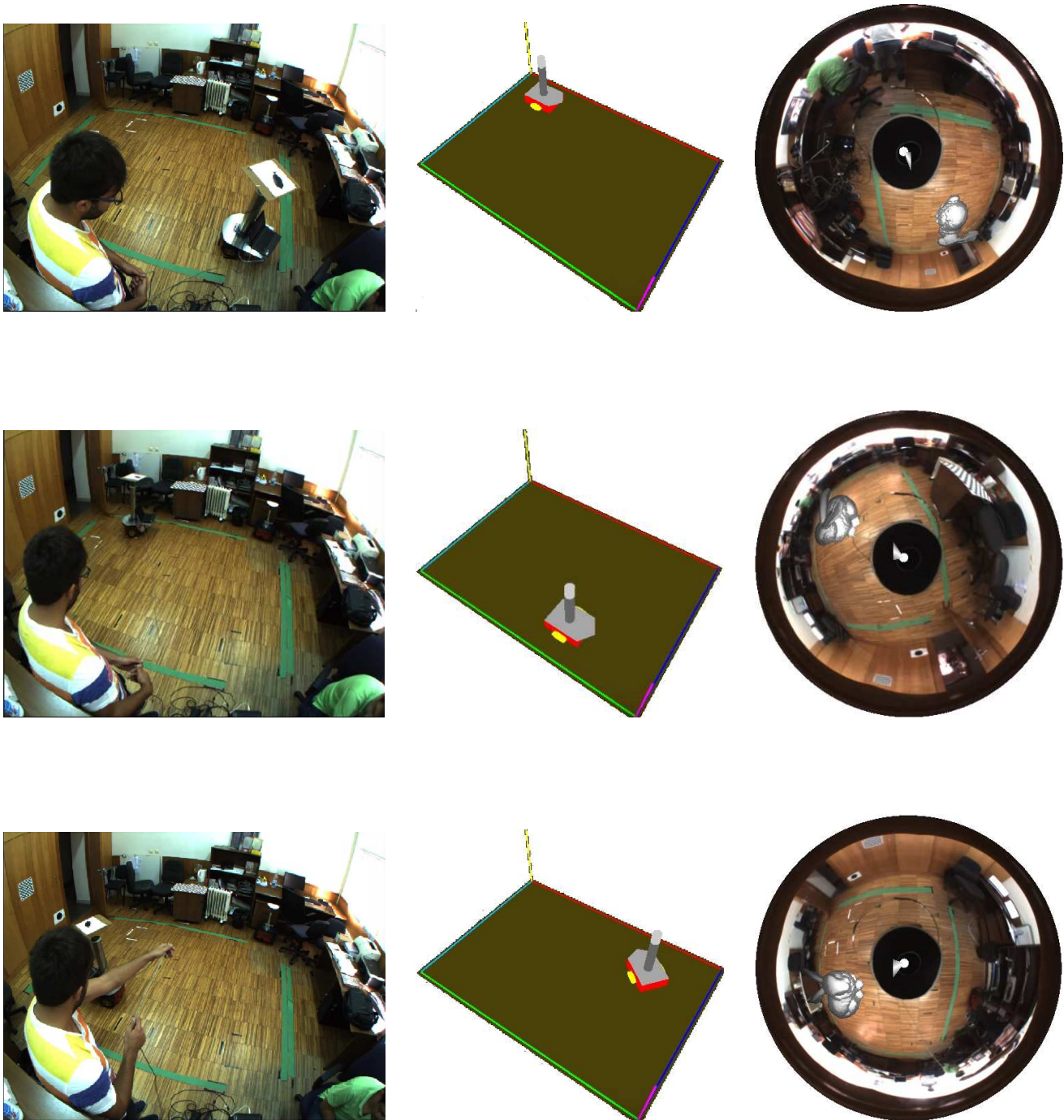


Fig. 5.7: In this figure we show a set of frames from three different positions of the robot. On the left column, it is shown the image obtained by the auxiliary camera, which is acquiring the realtime events in the real world, on the center column, we show the 3D virtual arena showing the position of the robot in the arena and, on the right column, the result of our framework according to the position of the robot and light focus (which is on the top of the robot).

Chapter 6

Conclusions and Future Work

6.1 Conclusions

In this dissertation a framework is proposed for the application of augmented reality using non-central catadioptric imaging devices. It is believed that this is the first time that this problem is addressed. Assuming that the camera is fully calibrated and that the 3D object is segmented (in small 3D triangles) with the respective texture mapping, the proposed framework is completed with the following four simple steps:

- Projection of the 3D triangles to the 2D image plane;
- Check for occlusions on the projected triangles;
- Compute the illumination associated to each projected triangle;
- Display the virtual object in the current frame obtained by the camera;

In our experiments, a laptop was used with a CPU “Intel i7 3630QM” (2.4 *GHz* with 4 cores) and a GPU “NVIDIA GeForce GT 740M” (810 *MHz* with 384 CUDA cores) and the framework was tested using three 3D virtual objects: the parallelepiped (with 45k 3D triangles), the Stanford bunny and the happy Buddha. Using the CUDA Toolkit, it was possible to project the Stanford bunny and the happy Buddha to the image in up to 20 fps (both have approximately 70k 3D triangles). For the parallelepiped, the framework reached 10 fps, which is normal since we had to deal with each face individually and make the association between the triangulated texture and the respective triangles of each face.

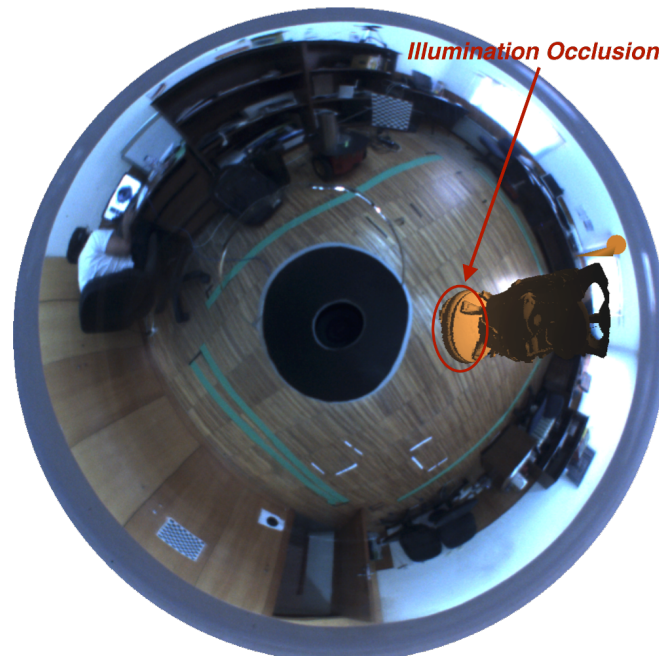


Fig. 6.1: In this figure it is represented the illumination problem when dealing with multiple objects, Buddha character and the base that it stands on (highlighted).

Although the proposed framework has not reached the 25 fps (normally associated to realtime applications), that was not a main objective of this work. So, it can be said that all the objectives initially assumed in this dissertation were completed with success, taking into account that some improvements can be made to build a more robust framework and to improve the computational time.

6.2 Future Work

As future work, it is important to highlight some changes that can be done in way to improve the proposed framework. The first is related to the projection of the triangles. It was intentionally the use of a large number of very small triangles to neglect the distortion associated with the projection of the 3D triangles. However, if we could use the actual projection of 3D triangles (taking into account the distortion), a smaller number of triangles could be used and the computation time could decrease significantly. However, this projection problem has not yet been addressed. Another improvement that would be interesting to consider are the shadows of the virtual objects projected into the real scene, as well as the direct effect of the spotlight on the real scene. In this dissertation, the proposed framework was implemented to deal with single objects. That is a concept that would be interesting to expand. However, this would bring to

the occlusions and illumination some modifications algorithms (in Fig. 6.1, the front part of the base should not be illuminated since the light comes from the back of the Buddha character). Besides, in order to support more than an object in the scene some modifications to our CUDA implementation must be done to get better performance.

References

- [1] A Agrawal and S. Ramalingam. Single image calibration of multi-axial imaging systems. In *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, pages 1399–1406, June 2013.
- [2] A Agrawal, Y. Taguchi, and S. Ramalingam. Beyond alhazen’s problem: Analytical projection model for non-central catadioptric cameras with quadric mirrors. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 2993–3000, June 2011.
- [3] Arthur Appel. Some techniques for shading machine renderings of solids. In *Proceedings of the April 30–May 2, 1968, spring joint computer conference*, pages 37–45. ACM, 1968.
- [4] Ronald T. Azuma et al. A survey of augmented reality. *Presence*, 6(4):355–385, 1997.
- [5] Simon Baker and Shree K. Nayar. A theory of single-viewpoint catadioptric image formation. *International Journal of Computer Vision*, 35(2):175–196, 1999.
- [6] Joao P Barreto and Helder Araujo. Geometric properties of central catadioptric line images. In *Computer Vision—ECCV 2002*, pages 237–251. Springer, 2002.
- [7] James F. Blinn. Models of light reflection for computer synthesized pictures. In *ACM SIGGRAPH Computer Graphics*, volume 11, pages 192–198. ACM, 1977.
- [8] Loren Carpenter. The a-buffer, an antialiased hidden surface method. *ACM SIGGRAPH Computer Graphics*, 18(3):103–108, 1984.
- [9] Chu-Song Chen and Wen-Yan Chang. On pose recovery for generalized visual sensors. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 26(7):848–861, 2004.
- [10] Paul Debevec. Rendering synthetic objects into real scenes: Bridging traditional and image-based graphics with global illumination and high dynamic range photography. In *ACM SIGGRAPH 2008 classes*, page 32. ACM, 2008.

- [11] Paul E. Debevec and Jitendra Malik. Recovering high dynamic range radiance maps from photographs. In *ACM SIGGRAPH 2008 classes*, page 31. ACM, 2008.
- [12] Boris Delaunay. Sur la sphere vide. *Izv. Akad. Nauk SSSR, Otdelenie Matematicheskii i Estestvennyka Nauk*, 7(793-800):1–2, 1934.
- [13] Artur L. dos Santos, Diego Lemos, Jorge Eduardo Falcao Lindoso, and Veronica Teichrieb. Real time ray tracing for augmented reality. In *Virtual and Augmented Reality (SVR), 2012 14th Symposium on*, pages 131–140. IEEE, 2012.
- [14] Alain Fournier, Atjeng S. Gunawan, and Chris Romanzin. Common illumination between real and computer generated scenes. In *Graphics Interface*, pages 254–254. CANADIAN INFORMATION PROCESSING SOCIETY, 1993.
- [15] Christopher Geyer and Kostas Daniilidis. A unifying theory for central panoramic systems and practical implications. In *Computer Vision—ECCV 2000*, pages 445–461. Springer, 2000.
- [16] Nuno Gonçalves. On the reflection point where light reflects to a known destination on quadratic surfaces. *Optics letters*, 35(2):100–102, 2010.
- [17] Michael D. Grossberg and Shree K. Nayar. A general imaging model and a method for finding its parameters. In *Computer Vision, 2001. ICCV 2001. Proceedings. Eighth IEEE International Conference on*, volume 2, pages 108–115. IEEE, 2001.
- [18] Richard Hartley and Andrew Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2000.
- [19] John Hughes, Andries Van Dam, Morgan Mcguire, David F. Skylar, James D. Foley, Steven K. Feiner, and Kurt Akeley. *Computer Graphics: Principles and Practice Third Edition*. Addison-Wesley, United States of America, 2014.
- [20] Branislav Micusik and Tomas Pajdla. Autocalibration & 3d reconstruction with non-central catadioptric cameras. In *Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on*, volume 1, pages I–58. IEEE, 2004.
- [21] Paul Milgram and Fumio Kishino. A taxonomy of mixed reality visual displays. *IEICE TRANSACTIONS on Information and Systems*, 77(12):1321–1329, 1994.

- [22] Pedro Miraldo and Helder Araujo. A Simple and Robust Solution to the Minimal General Pose Estimation. *IEEE Proc. Int'l Conf. Robotics & Automation (ICRA)*, 2014.
- [23] Pedro Miraldo and Helder Araujo. Planar Pose Estimation for General Cameras using Known 3D Lines. *IEEE/RSJ Proc. Int'l Conf. Intelligent Robots & Systems (IROS 2014)*, to be published, 2014.
- [24] Pedro Miraldo and Helder Araujo. Pose Estimation for Non-Central Cameras Using Planes. *IEEE Int'l Conf. Autonomous Robot Systems & Competitions – ROBÓTICA*, 2014.
- [25] MobileRobots, Inc. Pioneer 3-DX. http://www.mobilerobots.com/Mobile_Robots.aspx.
- [26] Vic. Nalwa. A true omnidirectional viewer. Technical report, technical report, Bell Laboratories, 1996.
- [27] Shree K. Nayar and Simon Baker. Catadioptric image formation. In *Proceedings of the 1997 DARPA Image Understanding Workshop*, pages 1431–1437, 1997.
- [28] David Nistér and Henrik Stewénus. A minimal solution to the generalised 3-point pose problem. *Journal of Mathematical Imaging and Vision*, 27(1):67–79, 2007.
- [29] Nuno Gonçalves. *Noncentral Catadioptric Systems with Quadric Mirrors: Geometry and Calibration*. PhD thesis, University of Coimbra, 2008.
- [30] Luis Perdigoto and Helder Araujo. Calibration of mirror position and extrinsic parameters in axial non-central catadioptric systems. *Computer Vision and Image Understanding*, 117(8):909–921, 2013.
- [31] Imari Sato, Yoichi Sato, and Katsushi Ikeuchi. Acquiring a radiance distribution to superimpose virtual objects onto a real scene. *Visualization and Computer Graphics, IEEE Transactions on*, 5(1):1–12, 1999.
- [32] Gerald Schweighofer and Axel Pinz. Globally optimal o (n) solution to the pnp problem for general camera models. In *Proc. BMVC*, pages 1–10, 2008.
- [33] Dave Shreiner. *OpenGL Programming Guide Seventh Edition*. Addison-Wesley, United States of America, 2010.
- [34] Stanford University Computer Graphics Laboratory. Stanford Bunny. <https://graphics.stanford.edu/data/3Dscanrep/>, 1993.

-
- [35] Stanford University Computer Graphics Laboratory. Happy Buddha. <https://graphics.stanford.edu/data/3Dscanrep/>, 1996.
- [36] Jonathan Steuer. Defining virtual reality: Dimensions determining telepresence. *Journal of communication*, 42(4):73–93, 1992.
- [37] Rahul Swaminathan, Michael D. Grossberg, and Shree K. Nayar. Caustics of catadioptric cameras. In *Computer Vision, 2001. ICCV 2001. Proceedings. Eighth IEEE International Conference on*, volume 2, pages 2–9. IEEE, 2001.
- [38] R. Swaminathan, Michael D. Grossberg, and Shree K. Nayar. A perspective on distortions. In *Computer Vision and Pattern Recognition, 2003. Proceedings. 2003 IEEE Computer Society Conference on*, volume 2, pages II–594. IEEE, 2003.