Noname manuscript No.
(will be inserted by the editor)

A Framework for Augmented Reality using Non-Central Catadioptric Cameras

Tiago Dias · Pedro Miraldo · Nuno Gonçalves

Received: date / Accepted: date

Abstract This paper addresses the problem of augmented reality on images acquired from non-central catadioptric systems. We propose a solution which allows the projection of textured objects to images of these type of systems and, depending on the complexity of the objects, can run up to 20 fps, using a 1328x1048 image resolution. The main contributions are related with the image formation of the non-central catadioptric cameras: projection of the 3D segments onto the image of non-central catadioptric cameras; occlusions; and illumination/shading. To validate the proposed solution, we used a non-central catadioptric camera formed with a perspective camera and a spherical mirror. Also, to test the robustness of the proposed method, we used a regular object (a parallelepiped) and three well known irregular objects in computer graphics: "bunny", "happy buddha" and "dragon", from Stanford database.

Keywords Augmented Reality · Non-Central Catadioptric Cameras · Forward-Projection

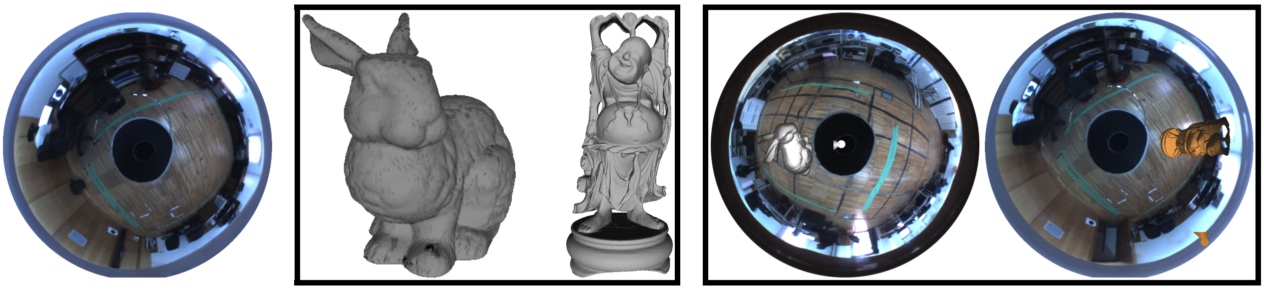
1 Introduction

Augmented reality has been studied for almost fifty years [1]. As stated by Azuma [2], augmented reality can be defined as the projection of 3D virtual objects to the 2D image plane. For the conventional perspective camera, a large number of distinct methods have been presented, *e.g.* [3,4, 5,6]. The main reasons for the use of these cameras are their simplicity (specially what is related to the projection model) and wide availability.

Geometrically, any imaging device can be modeled by the association between image pixels and unconstrained 3D straight projection lines [7]. When all these lines intersect at a single 3D point (also called effective viewpoint), they are called central. Otherwise, they are called non-central. Most state-of-the-art on computer vision and computer graphics methods/algorithms were developed under the assumption that images are acquired by sensors verifying the pinhole camera model (central perspective cameras [8]), thus free from distortions. However, with appropriate undistortion methods, any central camera system can be modeled by a central perspective camera [9]. As a result, the same methods/algorithms can be easily applied to all central camera systems. For these reasons, when possible, researchers tried to design new camera systems that verify the "single viewpoint" constraint (central cameras). The first central omnidirectional camera system was proposed by Nalwa in 1996 [10], which consists in aligning four perspective cameras with four mirrors. Later (following Nayar's work [11]), several authors started to build omnidirectional cameras, combining perspective cameras with quadric mirrors (catadioptric camera systems). In theory, as shown in [12], it is possible to define a set of conditions (using specific types of mirrors and a perfectly alignment between the camera and mirror) which ensures that such systems are central. However, small misalignments (for example between the camera and mirror(s)) or using other types of mirrors (for example spherical mirrors) will imply that these systems will not verify the single viewpoint constraint. This means that, in practice, omnidirectional catadioptric systems are non-central cameras [13]. As a result, distortion cannot be

Tiago Dias · Nuno Gonçalves
Institute of Systems and Robotics, Department of Electrical and Computer Engineering, University of Coimbra, Portugal
Tel.: +231-239-796-201
Fax: +231-239-406-672
E-mail: {tdias, nunogon}@isr.uc.pt

Pedro Miraldo
Institute for Systems and Robotics (LARSyS), Instituto Superior Técnico,Universidade de Lisboa, Lisboa, Portugal
Tel.: 351-21-8418289
Fax: 351-21-8418291
E-mail: pmiraldo@isr.tecnico.ulisboa.pt



(a) Original Image. (b) 3D objects example. (c) Final results.

Fig. 1: This paper addresses the projection of a virtual object (e.g. Fig. (b)) to an image, acquired from a non-central catadioptric camera, Fig. (a). Due to the geometry of the imaging device (specially what is related with forward projection techniques), conventional techniques cannot be used. The main contributions of the paper are: projection of the objects' skeleton, occlusions, and illumination (which all depend on the geometry of the imaging device). Results of the proposed framework are shown at Fig. (c).

modeled without prior knowledge of the 3D world from the scene (unwrapped images cannot be recovered), which means that augmented reality methods, used on perspective cameras, cannot be applied. Several authors proposed models and calibration procedures for non-central catadioptric camera systems using general quadric mirrors, *e.g.* [14, 15, 16, 17].

In this paper we propose a framework for the use of augmented reality using non-central catadioptric imaging devices. To the best of our knowledge, this is the first time that the problem is addressed. An example of the obtained results are shown in Fig. 1. Augmented reality for omnidirectional catadioptric cameras can be extremely useful for human-computer interaction [18], with several important applications in robotics. Two examples of these applications are: teleoperation [19] (creation and projection of 3D virtual landmarks to assist the human on robot navigation) and the creation of augmented reality environment simulations [20] (creation and projection of 3D objects to simulate real scenarios). Another example of an environment simulation (using augmented reality) is its application on the medical surgery (see e.g. [21]). During medical surgeries, the frontal view of a camera is very important. Although, to prevent damage on organs that are not visible from this perspective, non-central omnidirectional cameras can be used to provide a larger field of view. With a larger field of view, we have more information about the surrounding environment, which can help us making better decisions. The same justification can be applied to teleoperation on robotics. Since we can acquire 360 degrees of the scene, we can make a faster detection of objects, placed in the environment, and decide faster on the best trajectory. The use of augmented reality, in both cases, can be very useful for the creation of simulated environments, which can provide to the user a good experience for a specific task.

To better understand the proposed solution, we built a pipeline aiming at representing the tasks required to get the goal (shown in Fig. 2). To reach this goal, new algorithms and some well known methods had to be created/reformulated, such that they can be applied to non-central catadioptric systems. Assuming that the camera is calibrated and that our 3D object is divided in segments (triangles), one of the most challenging steps is the projection of these triangles (which form the 3D objects) onto the image plane. Considering that the triangles are small enough, we can neglect the effects of distortion [9]. Thus, to project these 3D triangles, one just needs to take into account the projection of three 3D points (that form the vertices of the triangles). The forward projection of 3D points for images of non-central catadioptric cameras was addressed by Gonçalves [22] and Agrawal [23].

As it was previously said, the geometry of these imaging systems does not verify most properties of the conventional perspective cameras. Thus, we also had to reformulate conventional computer graphics techniques: such as occlusions and illumination. Occlusions are a very well known problem in Computer Graphics. When a 3D virtual object is divided in small 3D pieces (for example 3D triangles), when mapping these pieces to the image one have to verify if the pieces are overlapped and, if they are, which of them are visible and which of them are not. To solve this problem, several methodologies were proposed: the Painter's Algorithm [24, Chapter 36.4], Z-Buffer (also known as Depth Buffer) [24, Chapter 36.3] and A-Buffer [25]. Another very important step is the object illumination. If we consider a 3D object with a solid colour, without illumination the projection of this 3D object to the image will be represented by a BLOB (Bynary Large OBject). The illumination, combined with a shading technique, will give the illusion of shape to the projection of the 3D object (this problem is better identified at the illumination section). To solve this problem, several algorithms were proposed, such as: Flat shading [24, Chapter 6.2], Gouraud shading [26] and Phong shading [27]. To conclude, we have to display the projection of the virtual object onto the image.

We have implemented the proposed framework in C/C++. Because of its complexity, specially in the projection's step, we only got up to 2 frames per second (fps), for an image size of 1328x1048. Then, to improve the computational time of our framework, we used the CUDA toolkit (from NVIDIA), and we get up to 20 fps. In this paper, we assume as realtime the perception of movement associated to the human eye, which is near to 25 frames per second.

This work is an extension of the paper "A Framework for Augmented Reality using Non-Central Catadioptric Cameras" presented in IEEE Intl Conf. on Autonomous Robot Systems and Competitions. We introduce the following changes:

– A larger and more detailed introduction and description of the proposed pipeline (Secs. 1 and 2);
– Regarding the illumination, in addition to Flat shading, we also adjust the Gouraud shading technique and took into account the illuminations occlusions problem to work with non-central catadioptric cameras (Sec. 3.6);
– New experimental results have been added to evaluate the proposed framework (Sec. 4).

This article is organized as follows: in Sec. 2, we describe the pipeline of the proposed framework and, in Sec. 3, each step of the framework is described in more detail. In Sec. 4 we show the experiments with the results of the application of the proposed framework and in Sec. 5 we give the conclusions of the paper.

2 Our Approach

To ensure that our framework runs in realtime, we divided the pipeline in two stages: pre-processing and realtime stages, see Fig. 2. As described in the introduction section, to achieve our goal, one has to take into account the following steps: camera calibration, 3D object segmentation, texture mapping, skeleton projection, occlusions, illumination and display. In this paper, we are assuming that our 3D object is rigid and static. As a result, to avoid unnecessary computational effort, the first three steps can be computed *a priori*. The remaining steps have to be computed in realtime. In the following two subsections we analyze the two stages of our pipeline.

2.1 Pre-Processing Stage

The pre-processing stage is built by three steps: camera calibration and 3D segmentation and texturization of the virtual 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 we are considering general non-central catadioptric cameras, the goal is to get the camera intrinsic parameters, the mirror parameters, and the transformation between the camera and mirror (in Sec. 3.1 we present a detailed description of this step).

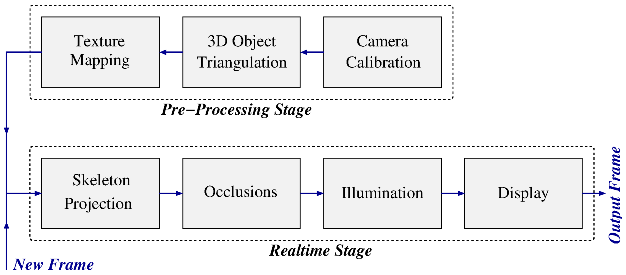


Fig. 2: Representation of the proposed pipeline for the use of augmented reality on non-central catadioptric cameras. We divided the problem in two stages: pre-processing stage, where camera parameters and 3D object information is computed; and the realtime stage where the pre-processed object is mapped onto the image plane.

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 segments which, later on, will be projected onto the 2D image plane. If these segments are small enough, the distortion effects will be neglectable and can be ignored. Similar to most state-of-the-art approaches, we used the segmentation of the 3D virtual object in 3D triangles. We test our method using a virtual paralellepiped (which we had to triangulate) and three objects from Stanford database [28] ("bunny", "happy buddha" and "dragon" already triangulated).

In addition to the 3D segmentation, the third step is related with the texturization of the 3D segments according to the 3D virtual object. These steps are further analyzed in Sec. 3.2 and 3.3, respectively.

2.2 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 formed by the following four steps: "skeleton projection", occlusions, illumination and display.

Since we are using very small 3D triangles, and we are ignoring the distortion effects on these triangles, their image (textured) will just depend on the projection of three 3D points to the 2D image plane that represent the three vertices of each 3D triangle. The "skeleton projection" step is related

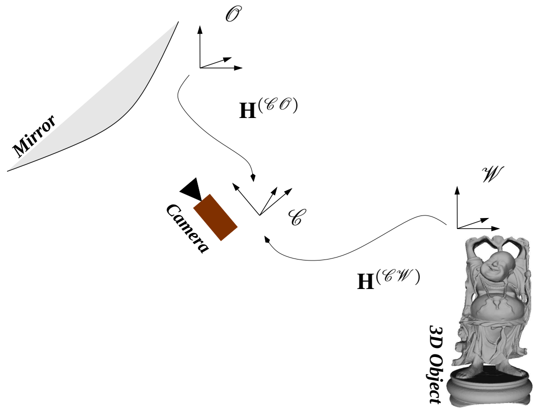


Fig. 3: Depiction of the problem related to the projection of a 3D object onto the image plane, using a non-central imaging device. This figure shows the three coordinates systems that must be considered: world ($\mathscr{W}$), camera ($\mathscr{C}$) and mirror ($\mathscr{O}$) respectively). Also represented are the transformations between the coordinates systems: between mirror and camera coordinates ($\mathbf{H}^{(\mathscr{C}\mathscr{O})}$) and between world and camera coordinates ($\mathbf{H}^{(\mathscr{C}\mathscr{W})}$).

with the projection of the triangles' vertices onto the image plane. Note that, since we are using non-central catadioptric cameras, this step is not as easy as 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 situation, before computing the projection of 3D points to the image plane, we have to estimate the pose of the camera. This is a very important issue when we have a mobile camera. This step is further analyzed in Sec. 3.4.

Since we are considering the projection of small segments onto the image plane, it is very important to understand if these segments are overlapped and, if they are, which of them are in front. The main difference between the proposed method and the conventional algorithms is related to the definition of "point of view". For the conventional perspective camera, one can use the camera center (also called the effective view point [8]) as a "point of view" for all 3D triangles, and the distance between the triangle and the camera is computed as a distance between the 3D segment and the camera center. For our case, this cannot be applied. Note that we are considering non-central imaging devices, which means that there isn't a single point where all the 3D projection lines intersect. To solve this problem, we propose a solution based on the Painter's Algorithm methodology, which consists in drawing the scene (small segments) from the farthest to the nearest. This problem is fully addressed in Sec. 3.5.

Note that, Z-Buffer is probably the simplest and most widely used technique to solve this problem. However, this method requires the association between pixels and coordinates of 3D points, for all pixels that define the object. We want to avoid this because of the complexity associated with the projection of points to images of non-central catadioptric systems (state-of-the-art solution cannot be applied directly). Moreover, we are ignoring the distortion effects on the projection of the triangles (by considering a large number of small 3D triangles), which means that there is no easy way to compute the matching between all pixels and respective 3D points that belong to the triangles.

When regarding illumination and shading, there are several proposed approaches [24, 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 in our framework. For simplicity, we used Flat shading technique, which considers the complete illumination of the 3D triangle equal to the illumination of the mass center of the respective 3D triangle. In addition, since we are dealing with irrelugar surfaces (Stanford objects), we also reformulate the Gouraud shading technique (which is usually used for smooth objects) to work with general non-central catadioptric cameras. This technique uses the colour of each of the triangle's vertices and, knowing this information, defines the colour of all the triangle's pixels using a linear interpolation process.

As for the illumination parameters, we reformulate the well known Phong's reflection model. The equation parameters applied to our case (non-central catadioptric systems) are analyzed in Sec. 3.6.

Now that we have 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. For simplicity, we used the OpenGL to render/display the virtual object on the current frame obtained from the camera.

3 Detailed Steps of the Pipeline

In this section, we will describe in detail the steps in which the proposed pipeline of Fig. 2 are decomposed. For now on, we will use the superscripts $(\mathscr{W})$, $(\mathscr{C})$ and $(\mathscr{O})$ to represent features in the world (in which the 3D object was defined), camera and mirror coordinates, respectively.

3.1 Camera Calibration

As we previously described, in this paper we are considering the use of non-central catadioptric cameras formed by a central perspective camera and a quadric mirror (see Fig. 4). This step is about the calibration of this system. For that,

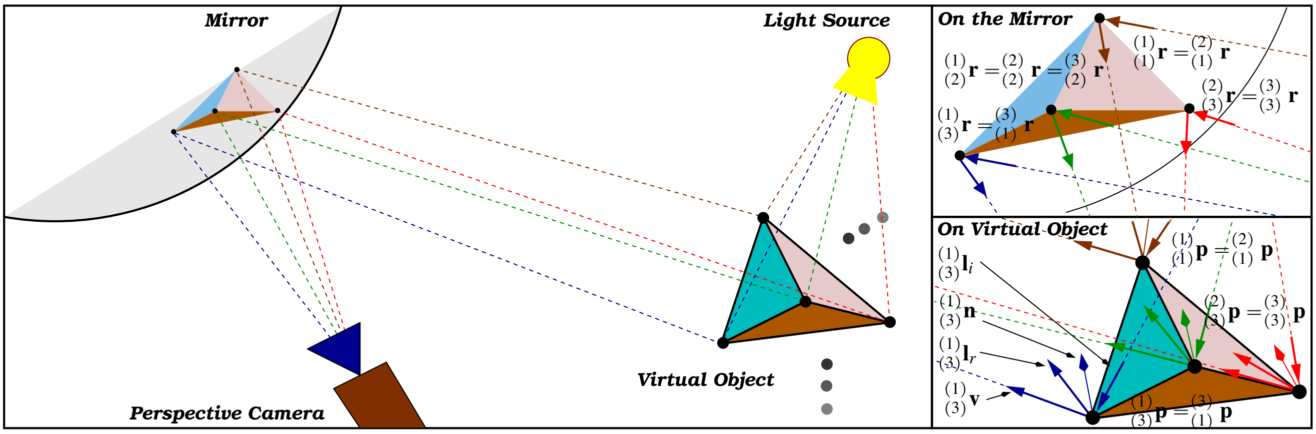


Fig. 4: Representation of the projection (proposed at Sec. 3.4) and illumination steps (described at Sec. 3.6). Points ${}^{(j)}_{(i)}\mathbf{p}$ represent vertices of the triangle and the main goal is to project these vertices to the image, which consists on the estimation of the respective points on the mirror ${}^{(j)}_{(i)}\mathbf{r}$. The directions ${}^{(j)}_{(i)}\mathbf{l}_i$, ${}^{(j)}_{(i)}\mathbf{n}$, ${}^{(j)}_{(i)}\mathbf{l}_r$, and ${}^{(j)}_{(i)}\mathbf{v}$ denote the incident ray (that came from the light source), the normal at the respective point, the reflected ray and the viewer's direction respectively.

one has to consider: the calibration of the central perspective camera, which means, estimate the camera parameters $\mathbf{K} \in \mathbb{R}^{3\times 3}$ such that ${}^{(j)}\mathbf{v}_r^{(\mathscr{C})} \sim \mathbf{K}\,{}^{(j)}\mathbf{r}^{(\mathscr{C})}$ (where ${}^{(j)}\mathbf{v}_r^{(\mathscr{C})}$ and ${}^{(j)}\mathbf{r}^{(\mathscr{C})}$ are the projection ray of the perspective camera and the respective 3D point on the mirror); and the mirror parameters matrix $\Omega \in \mathbb{R}^{4\times 4}$ and $\mathbf{H}^{(\mathscr{OC})} \in \mathbb{R}^{4\times 4}$ such that

$$ {}^{(j)}\mathbf{r}^{(\mathscr{C})\,T}\,\mathbf{H}^{(\mathscr{OC})\,T}\,\Omega\,\mathbf{H}^{(\mathscr{OC})}\,{}^{(j)}\mathbf{r}^{(\mathscr{C})} = 0, \tag{1} $$

where $\mathbf{H}^{(\mathscr{OC})}$ is the matrix that transforms a point from the quadric to the camera coordinate systems, see Fig. 3. Now that we have all the required parameters, we can use the Snell's law to compute the 3D projection direction

$$ {}^{(j)}\mathbf{v}_i^{(\mathscr{C})} = {}^{(j)}\mathbf{v}_r^{(\mathscr{C})} - 2\left({}^{(j)}\mathbf{v}_r^{(\mathscr{C})\,T}\,{}^{(j)}\mathbf{n}_q^{(\mathscr{C})}\right){}^{(j)}\mathbf{n}_q^{(\mathscr{C})}, \tag{2} $$

where ${}^{(j)}\mathbf{n}_q^{(\mathscr{C})}$ is the normal vector at the 3D quadric mirror point ${}^{(j)}\mathbf{r}^{(\mathscr{C})}$. To calibrate the non-central catadioptric camera, we follow the method proposed by Perdigoto and Araujo [16].

3.2 3D Object Triangulation

As mentioned above, we decided to segment the virtual object in 3D triangles. To avoid distortion aberrations, we just considered very small triangles (the distortion in the image will be very small). Let us consider that we know the coordinates of the 3D virtual object (which we know from definition). As a result, points that belong to that 3D object can be referenced. Using these points we can use Delaunay algorithm [29] to compute the 3D triangles that define the virtual object. In addition, in our experiments, we also used three 3D objects that were already triangulated: the Stanford “bunny“, the “happy buddha“ and the ”dragon“.

Let us consider that an object is, already, triangulated with N 3D triangles. Thus, we know the coordinates of the three vertices that define the N triangles. Formally

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

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

Fig. 5(a) presents the result of the segmentation for the 3D parallelepiped object.

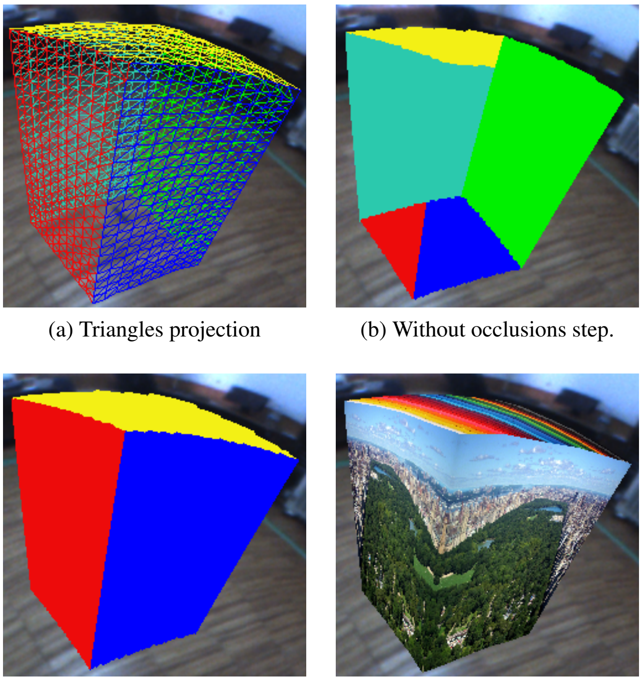
3.3 Texture Mapping

Let us consider, for example, the texturization of a 3D virtual parallelepiped. Using the triangulation defined in Sec. 3.2, we know the vertices that form all triangles (3D point ${}^{(j)}_{(i)}\mathbf{p}^{(\mathscr{W})}$). Since we consider the 2D faces individually, one can obtain the texture associated to each triangle through a conversion of the 3D world coordinates of each face to the respective texture coordinates (a 2D image). This procedure can be done at the pre-processing stage because we are considering that the coordinates associated to each triangle will not change (static objects). For the Stanford objects, since the goal of our work is not to map a texture to an irregular surface, we used a single colour texture to all the 3D triangles that define the object.

Fig. 5(d) presents the result of the texturization for the 3D parallelepiped object.

3.4 Skeleton Projection

In this step, the goal is to compute the projection of 3D triangles (that define the 3D virtual object) onto the image plane.



(a) Triangles projection (b) Without occlusions step.

(c) With occlusions step. (d) Texture mapping.

Fig. 5: Results of the triangles' projection and occlusion steps applied to a 3D parallelepiped cube. Fig. (a) shows the projection of the 3D triangles onto the image. Figs. (b) and (c) show the effects of the occlusion step (before and after respectively) and Fig. (d) shows the result of the occlusion step with textured faces. The effects of distortion can be easily seen from any of these images.

Since we are ignoring the effects of triangle's distortion, this can be computed simply by projecting the three vertices (that define each triangle) to the image. Let us consider the projection of 3D world points to the image plane of a non-central catadioptric camera. Since we know the parameters of the calibration of the perspective camera (see Sec. 3.1), this problem can be seen as the estimation of the 3D reflection point on the mirror (see Fig. 4). The first thing one needs to verify is if the coordinate system of the 3D object is the same as the coordinates of the camera system. As a result, we have to compute the rigid transformation $\mathbf{H}^{(CW)} \in \mathbb{R}^{4\times4}$ between both coordinate systems (see Fig. 3)

$$ {}^{(j)}_{(i)}\tilde{\mathbf{p}}^{(\mathscr{C})} \sim \mathbf{H}^{(\mathscr{C}\mathscr{W})}\, {}^{(j)}_{(i)}\tilde{\mathbf{p}}^{(\mathscr{W})}, \quad (4) $$

where $\tilde{\mathbf{p}}$ denotes the homogeneous representation of $\mathbf{p}$. This problem is known as the absolute camera pose estimation. Several authors addressed this problem, *e.g.* [30,31,32]. In the experiments, we used the method proposed by Miraldo and Araujo at [33]. This is very important since the goal is to use a mobile camera. Each time a new image is received, the pose must be recomputed. From now on, we will assume

Algorithm 1: Reformulation of Painter's algorithm for images of non-central catadioptric cameras.

Let ${}^{(j)}_{(i)}\mathbf{p}$ 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 mass center ${}^{(j)}\mathbf{t}$ for each triangle (7);
  Compute reflection point ${}^{(j)}\mathbf{r_t}$, using [22] ;
  Set ${}^{(j)}\xi$ as the distance between ${}^{(j)}\mathbf{r}$ and ${}^{(j)}\mathbf{t}$;
end
Sort all the triangles by descending order using ${}^{(j)}\xi$, for all $j = 1, \dots, N$;

that 3D points are already known in the camera coordinate system.

Now, for all the vertices of the triangles ${}^{(j)}_{(i)}\mathbf{p}^{(\mathscr{C})}$ (in the coordinates of the camera system), the goal is to compute the reflection point in the mirror ${}^{(j)}_{(i)}\mathbf{r}^{(\mathscr{C})}$. We used the solution method proposed by Gonçalves [22]. Note that other solutions could be used, for instance the method proposed by Agrawal *et al.* [23]. These methods are quite complex and the goal in this paper is not to address this problem. Therefore, we will consider a black box such that

$$ {}^{(j)}_{(i)}\mathbf{r}^{(\mathscr{C})} = \text{fProj}\left({}^{(j)}_{(i)}\mathbf{p}^{(\mathscr{C})}\right), \text{ for all } i \text{ and } j. \quad (5) $$

Using this methodology, we can now assume that we have the projection of all the 3D triangles that form the 3D virtual object. We will denote these triangles (on the image plane) as

$$ \left\{{}^{(j)}_{(1)}\mathbf{u}, {}^{(j)}_{(2)}\mathbf{u}, {}^{(j)}_{(3)}\mathbf{u}\right\}, \text{where } {}^{(j)}_{(i)}\mathbf{u} \sim \mathbf{K}\,{}^{(j)}_{(i)}\mathbf{r} \text{ and } {}^{(j)}_{(i)}\mathbf{p} \mapsto {}^{(j)}_{(i)}\mathbf{r}, \;\; \forall j = 1, \dots, N, \quad (6) $$

where ${}^{(j)}_{(i)}\mathbf{u}$ are the coordinates of the vertices on the image plane and $\mathbf{K} \in \mathbb{R}^{3\times3}$ are the camera intrinsic parameters [8].

3.5 Occlusions

As it was previously said, to solve the occlusions' problem, we propose a solution based on Painter's Algorithm. This method was chosen because of its simplicity and efficiency. However, since we are using 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 between these triangles and the camera system. As a result, to compute the distance between the 3D triangles and the camera system, we consider the distance between the triangle (we use the mass center of the triangle for simplicity) and the respective 3D reflection point on the mirror (see Fig. 4). In more detail, to compute the distance of each triangles j to the catadioptric camera, we consider the depth between the triangle's mass center

$$ {}^{(j)}\mathbf{t}^{(\mathscr{C})} = \frac{{}^{(j)}_{(1)}\mathbf{p}^{(\mathscr{C})} + {}^{(j)}_{(2)}\mathbf{p}^{(\mathscr{C})} + {}^{(j)}_{(3)}\mathbf{p}^{(\mathscr{C})}}{3}, \quad (7) $$

and its reflection point

$$ {}^{(j)}\mathbf{r}_t^{(\mathscr{C})} = \mathrm{fProj}\left({}^{(j)}\mathbf{t}^{(\mathscr{C})}\right), \text{ for all } j. \quad (8) $$

This step is formalized in algorithm 1. After its application, we have the 2D triangles in descending order and ready to display.

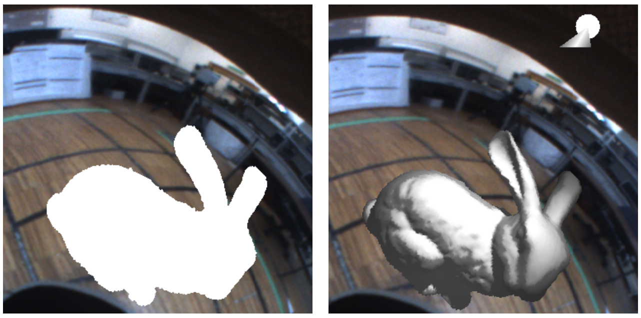
Results for this step are presented in Figs. 5(b) and (c). The first figure shows the projection of the 3D parallelepiped, without using the proposed occlusions' solution. The second figure shows the results after the application of the proposed algorithm. One can see that the proposed algorithm works well as expected and the problem is completely solved.

3.6 Illumination

In augmented reality, an object without illumination will be represented as a BLOB. When regarding irregular objects, we will not have the perception of the object's shape. To better understand the problem and its consequences, we show two images of the projected object without and with the application of the illumination step. The results are shown in Fig. 6.

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) and the scene's global illumination. We start from the Phong's reflection equation and derive a solution to work with non-central catadioptric cameras, (12). The three color channels are computed separately. For each channel and for a single point (on the image), we then defined two illumination components: $\tilde{I}^{(\mathrm{ch})}$, which represents the influence of both global and light source ambient properties on the object's material; and $\breve{I}^{(\mathrm{ch})}$, which represents the influence of the diffuse and specular light source properties on the object's material. The first one does not depend on the geometry of the camera systems and does not require further analysis. On the other hand, the latter depends on the object's projection to the image. Next, we analyse in more detail each components:

- **Diffuse reflection**: related with the object shape. It depends on the direction of the incident ray (that comes from the light source) and the surface normal at the respective 3D point (vertex position);
- **Specular reflection**: associated with the shininess reflected by the object. It depends on the reflection ray (that can be obtained using (2), assuming that the incident ray of the light source and the surface normal are known) and direction to the viewer's position. The incident ray is known (which is given by the position of the light source) and we can obtain its reflection ray using the Snell's law. Since we are using non-central systems, the direction to viewer's position can not be computed such as conventional techniques. For central cameras, this direction is computed by considering viewer's position at the "single view point". To solve this problem, we define the viewer's position at the respective reflection point on the mirror, which can be computed using [22, 23].



(a) (b)

Fig. 6: Results of the application of the illumination step to the "bunny" object. In Fig. (a) we show the results without the illumination step. As it can be seen, without illumination the object will be represented as a BLOB. In Fig. (b) we show the same results with the illumination step.

The ambient, diffuse and specular components are computed for all the vertices of the triangles, considering individually each light source influence.

In addition to these components, we need to take into account four additional directions (unitary): vector ${}^{(j)}_{(k)}\mathbf{l}_i^{(\mathscr{C})}$ represents the direction that points from the object point to the k^{th} light source (assumed to be known); vector ${}^{(j)}\mathbf{n}_t^{(\mathscr{C})}$ denotes the normal to the j^{th} triangle

$$ {}^{(j)}\mathbf{n}_t^{(\mathscr{C})} = \frac{\left({}^{(j)}_{(1)}\mathbf{p}^{(\mathscr{C})} - {}^{(j)}_{(3)}\mathbf{p}^{(\mathscr{C})}\right) \times \left({}^{(j)}_{(2)}\mathbf{p}^{(\mathscr{C})} - {}^{(j)}_{(3)}\mathbf{p}^{(\mathscr{C})}\right)}{\left|\left({}^{(j)}_{(1)}\mathbf{p}^{(\mathscr{C})} - {}^{(j)}_{(3)}\mathbf{p}^{(\mathscr{C})}\right) \times \left({}^{(j)}_{(2)}\mathbf{p}^{(\mathscr{C})} - {}^{(j)}_{(3)}\mathbf{p}^{(\mathscr{C})}\right)\right|}; \quad (9) $$

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

$$ {}^{(j)}_{(k)}\mathbf{l}_r^{(\mathscr{C})} = {}^{(j)}_{(k)}\mathbf{l}_i^{(\mathscr{C})} - 2\left({}^{(j)}_{(k)}\mathbf{l}_i^{(\mathscr{C})\,T}\ {}^{(j)}\mathbf{n}_t^{(\mathscr{C})}\right) {}^{(j)}\mathbf{n}_t^{(\mathscr{C})}; \quad (10) $$

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

$$ {}^{(j)}\mathbf{v}_i^{(\mathscr{C})} = \frac{{}^{(j)}\mathbf{r}_t^{(\mathscr{C})} - {}^{(j)}\mathbf{t}^{(\mathscr{C})}}{\left|{}^{(j)}\mathbf{r}_t^{(\mathscr{C})} - {}^{(j)}\mathbf{t}^{(\mathscr{C})}\right|} \quad (11) $$

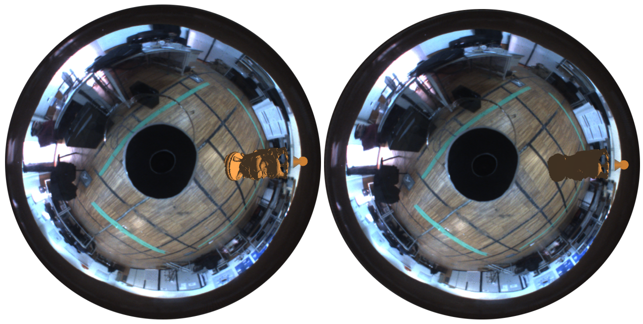
Algorithm 2: Proposed illumination algorithm.

Let ${}^{(j)}_{(i)}\mathbf{p}$ be the 3D coordinates of the i^{th} vertex of the j^{th} triangle, N the number of existing triangles. M the number of light sources, $\mathbf{d}_{sl\,(k)}$ the direction of the spotlight and Ω the union between the spotlight and j^{th} triangle's edges:
for $j = 1$ *to* N **do**
  Compute vertices' normal ${}^{(j)}_{(i)}\mathbf{n}_t$;
  Compute the reflection points ${}^{(j)}_{(i)}\mathbf{t} \mapsto {}^{(j)}_{(i)}\mathbf{r}$;
  Compute the visualization vectors ${}^{(j)}_{(i)}\mathbf{v}$;
  Set ${}^{(j)}_{(i)}I^{(\text{ch})} = {}^{(j)}_{(i)}\tilde{I}^{(\text{ch})}$ for each vertex;
  for $k = 1$ *to* M **do**
    Compute the reflection rays ${}_{(i)}\mathbf{l}_{r\,(k)}$;
    Set ${}_{(i)}f_k = 1$ and ${}_{(i)}spot_k = 0$;
    if *angle between* ${}_{(i)}\mathbf{l}_{i\,(k)}$ *and* ${}^{(j)}_{(i)}\mathbf{n}$ *bigger than zero* **then**
      ${}_{(i)}f_k = 0$;
    end
    if *maximum of* $\left\langle {}_{(i)}\mathbf{l}_{i\,(k)}, \mathbf{d}_{sl\,(k)} \right\rangle$ *and* 0 *bigger than* $C^{te}{}_{(k)}$ **then**
      ${}_{(i)}spot_k = \max\left\{ {}_{(i)}\mathbf{l}_{i\,(k)}{}^{T}\,\mathbf{d}_{sl\,(k)}, 0 \right\}^{\mathscr{E}}$;
    end
    Add ${}^{(j)}_{(i)}I^{(\text{ch})} = {}^{(j)}_{(i)}I^{(\text{ch})} + {}^{(j)}_{(i)}\breve{I}^{(\text{ch})}_{k}$ for each vertex, see (12);
  end
  Calculate ${}^{(j)}I^{(\text{ch})}$ using a linear interpolation of ${}^{(j)}_{(i)}I^{(\text{ch})}$;
end

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

Regarding the shading, we could use variations of Flat, Phong, or Gouraud's techniques (note that all of them need changes in what is related with the image formation). In our experiments we used both Flat and Gouraud's methodologies. As it was previously said, Gouraud's technique allows a smoother transition between the triangles. As it was mentioned, this methodology calculates the colour of the triangle using a linear interpolation process between the colour of the three vertices, that forms the respective triangle. The proposed solution using the Gouraud technique is formalized in Algorithm 2. For the flat shading technique, the main difference (considering the Gouraud's method) is that the three vertices of each triangle will have the same colour.

To conclude this step, we also had to take into account another illumination problem, which we denote as the occlusions' illumination problem. Let us consider the case where a triangle is behind another triangle, regarding the spotlight position. Note that, for the occlusions illumination problem and since we assumed each piece (triangle) as an independent part, the main issue is that the triangles which should not have a colour associated (because of nearests triangles are in front) will have. This occurs because the Phongs Reflection Model will calculate the colour of triangle, only considering its normal. Thus, triangles that are occluded (considering the light source) by others triangles will be illuminated. In this case, the first triangle should not be illuminated. However, the proposed Algorithm 2 does not solve this problem. This problem does not depend on the geometry of the imaging device and there are several solutions in the literature that could be used to solve this problem. In this paper we implemented a simple method, which basically searches if a triangle k is occluded by any other triangle and, if it is occluded, sets $occ_k = 0$ (otherwise it will be $occ_k = 1$). Later, this parameter will be used on (12) (which already takes into account this parameter). Results before and after the application of this step are shown in Figs.7 (a) and (b), respectively.



(a) (b)

Fig. 7: Depiction of the illumination's occlusion problem. Fig. (a) shows the results of the illumination without taking into account the illumination's occlusions (a triangle's illumination is occluded by another triangle). As it can be seen, in a realistic scenario and taking into account the light position, the triangles on the base and on the front of the object should not be illuminated (there are triangles in front of them when regarding the light source). Fig. (b) shows the results considering the illumination's occlusions.

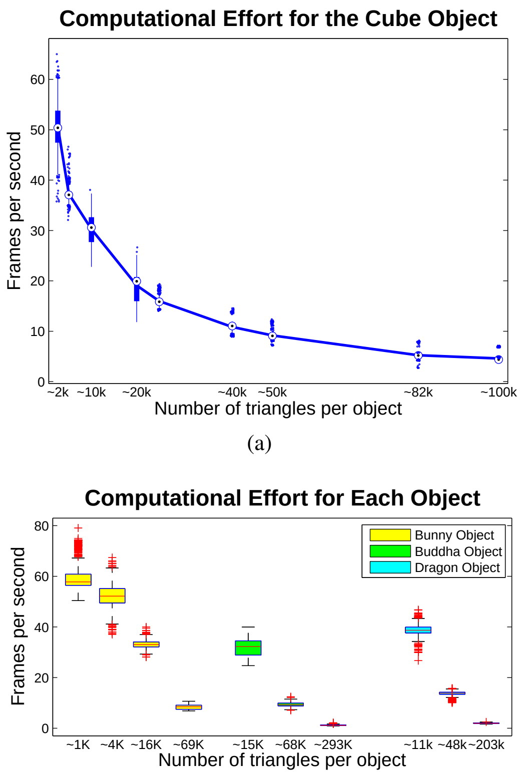
4 Experiments

The goal of this section is to evaluate the proposed framework. On the experiments, we used a non-central catadioptric camera formed with a perspective camera and a spherical mirror. As described in the introduction section, we test our framework using four 3D virtual objects: parallelepiped, "bunny", "happy buddha" and "dragon". Note that, on the detailed description subsections of the proposed solution, we already presented some experiments to evaluate the respective steps.

For the illumination parameters, (12), we chose to wrap our virtual objects with silver, which is a standard material in computer graphics. Additionally, our light source will be

$$
{}^{(j)}_{(i)}I^{(\mathrm{ch})} = \overbrace{K_e^{(\mathrm{ch})} + G_a^{(\mathrm{ch})} K_a^{(\mathrm{ch})} + {}_{(i)}\,spot_k \sum_{k=1}^{M} \underbrace{{}_{(k)}L_a^{(\mathrm{ch})} K_a^{(\mathrm{ch})}}_{\textbf{ambient component}}}^{\tilde{I}^{(\mathrm{ch})}} +
$$
$$
+ {}_{(i)}\,spot_k\; {}_{(i)}f_k\; {}_{(i)}occ_k \sum_{k=1}^{M} \underbrace{\left(\overbrace{{}_{(k)}L_d^{(\mathrm{ch})} K_d^{(\mathrm{ch})} \max\left\{ -{}^{(j)}_{(i)}\mathbf{l}_{i(k)}{}^T\, {}^{(j)}_{(i)}\mathbf{n}, 0 \right\}}^{\textbf{diffuse component}} + \overbrace{{}_{(k)}L_s^{(\mathrm{ch})} K_s^{(\mathrm{ch})} \max\left\{ {}^{(j)}_{(i)}\mathbf{v}^T\, {}^{(j)}_{(i)}\mathbf{l}_{r(k)}, 0 \right\}^{sh}}^{\textbf{specular component}} \right)}_{{}^{(j)}\tilde{I}_k^{(\mathrm{ch})}} \quad (12)
$$

Illumination equation for a single 3D point using non-central catadioptric cameras: M is the number of light sources; $K_a^{(\mathrm{ch})}$, $K_d^{(\mathrm{ch})}$, $K_s^{(\mathrm{ch})}$ $K_e^{(\mathrm{ch})}$ and sh are ambient, diffuse, specular, emission, shininess material color properties; $G_a^{(ch)}$ is the global ambient light property ((ch) denotes the color channel); ${}_{(k)}L_a^{(\mathrm{ch})}$, ${}_{(k)}L_d^{(\mathrm{ch})}$, ${}_{(k)}L_s^{(\mathrm{ch})}$ are the ambient, diffuse and specular properties of the k^{th} light source; boolean parameters ${}_{(i)}f_k$ and ${}_{(i)}occ_k$ are used to control whether the point is illuminated or not; and ${}_{(i)}spot_k$ controls the cutoff angle of the light source (definition of spotlight). A graphical representation of the directions ${}^{(j)}_{(i)}\mathbf{l}_{i(k)}$, ${}^{(j)}_{(i)}\mathbf{l}_{r(k)}$, ${}^{(j)}_{(i)}\mathbf{n}$, and ${}^{(j)}_{(i)}\mathbf{v}$ is shown in Fig. 4.



(b)

Fig. 8: Results of the computational effort for all the 3D objects. In Fig.(a) we present the number of frames per second obtained using different number of triangles for the cube object. In Fig.(b), we show the relation between the number of frames per second and the number of triangles for the "bunny", "buddha" and "dragon" objects.

treated as a spotlight, which is a positional and directional light source. In these experiments, we are always pointing the spotlight direction to the center of the 3D object. We also defined $L_a^{(\mathrm{ch})}$, $L_d^{(\mathrm{ch})}$ and $L_s^{(\mathrm{ch})}$ to be white for the parallelepiped and the "bunny" objects. For the "buddha" and the "dragon" objects, we used a gold and red spotlight, respectively. For the global ambient light property ($G_a^{(ch)}$) we used an arbitrary constant for each of the channel components.

On the first experiment, we captured a set of images when considering a moving spotlight. The results are shown in Fig. 9. For this experiment we used the parallelepiped, "bunny", "buddha" and "dragon" objects. For the first two objects (parallelepiped and "bunny", first and second row, respectively), we used our framework without taking into account the parameter occ_k, which means that illumination's occlusions between triangles are not taked into account. Regarding the shading we used a Flat shading technique. For the "buddha" and the "dragon" (third and fourth rows) illumination's occlusions and Gouraud shading technique were used. As it can be easily seen, comparing these images with the results of the first and second rows, these results are more realistic, when using irregular surfaces (as it would be expected). In addition, to evaluate the computational effort, we repeated these tests using different number of triangles that define each objects. Taking into account our results, a good dimension of the triangles is around 0.1188cm^2, for distances greater than 10cm. For each object, approximately 300 frames were captured (with different spotlight positions), saving the comutational time required to compute each frame. The estatistical distribution of each sequence are shown in Fig. 8(b). As expected, the execution time is higher (inverse of the frames per second) when the number of triangles (that form the 3D object) increases. From our point of view, the computational complexity of the pre-processing stage is not critical, and that is why we did not include any reference to the required computation time. Note that the pre-processing stage only needs to be ran one time. Videos with the complete sequences (recorded in realtime) are sent in the supplementary material.

In addition, we propose an experiment using multiple lights sources in the scene. For this test, we used three moving spotlights with different colours and movements, all pointing to the "bunny" object. The results of this experiment can be seen in Fig. 10. A video with these results are also sent in the supplementary material.

To conclude the experiments, we considered the same camera system but, in this case, mounted on a mobile robot (Pioneer 3D-X robot [34]). For this case, to ensure that the object's position is independent on the position of the robot/camera, the pose of the robot is computed (in this paper we used the method proposed by Miraldo and Helder at [33]) before the application of the augmented reality. The results, for this experiment, are shown in Fig. 11.

5 Conclusions

In this paper we address the use of Augmented Reality on images of a non-central catadioptric system. We believe that this is the first time that this problem is addressed. The goal of this paper is to identify differences between Augmented Reality using conventional perspective cameras versus non-central catadioptric cameras. We saw that, in theory, to be able to use augmented reality on non-central catadioptric cameras, one needs to take into account changes on the following steps: projection of the 3D triangles to the 2D image plane; check for occlusions on the projected triangles; and compute the illumination associated to each triangle. After identifying and understanding these problems, we proposed changes to each of these steps. From the experimental results, we conclude that the proposed solutions work well and in realtime.

Now, since we fully understand the differences between Augmented Reality using conventional perspective cameras and non-central catadioptric cameras, we can highlight some future work. The first is related to the projection of the triangles. We intentionally chose to use a large number of very small triangles to neglect the distortion effects associated with the projection of the 3D triangles. However, if this distortion can be considered, a smaller number of triangles could be used and the computation time would decrease significantly. Another improvement that we intend to consider is the shadows' effects of the virtual object, projected onto the real scene, as well as the direct effect of the light source on the real scene. All the steps/algorithms presented in this paper were implemented on ROS and will be available when the paper is accepted.

References

1. A. Appel, "Some techniques for shading machine renderings of solids," *Proceeding of the AFIPS*, 1968.
2. R. T. Azuma, "A Survey of Augmented Reality," *Presence:Teleoperators and Virtual Environments: MIT Press Journal*, 1997.
3. A. Fournier, A. S. Gunawan, and C. Romanzin, "Common Illumination between Real and Computer Generated Scenes," *Proceeding of Graphics Interface (GI'93)*, 1993.
4. I. Sato, Y. Sato, and K. Ikeuchi, "Acquiring a Radiance Distribution to Superimpose Virtual Objects onto a Real Scene," *IEEE Trans. on Visualization and Computer Graphics*, 1999.
5. P. Debevec, "Rendering Synthetic Objects into Real Scenes: Bridging Traditional and Image-based Graphics with Global Illumination and High Dynamic Range Photography," *ACM SIGGRAPH*, 2008.
6. A. L. Santos, D. Lemos, J. E. F. Lindoso, and V. Teichrieb, "Real Time Ray Tracing for Augmented Reality," *IEEE Symposium on Virtual and Augmented Reality (SVR)*, 2012.
7. M. D. Grossberg and S. K. Nayar, "A General Imaging Model and a Method for Finding its Parameters," *IEEE Proc. Int'l Conf. Computer Vision (ICCV)*, 2003.
8. R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2000.
9. R. Swaminathan, M. D. Grossberg, and S. K. Nayar, "A Perspective on Distortions," *IEEE Proc. Computer Vision and Pattern Recognition (CVPR)*, 2003.
10. V. S. Nalwa, "A True Omni-Directional Viewer," *Technichal report, Bell Laboratories*, 1996.
11. S. K. Nayar and S. Baker, "Catadioptric Image Formation," *DARPA Proc. Image Understanding Workshop*, 1997.
12. S. Baker and S. K. Nayar, "A Theory of Single-Viewpoint Catadioptric Image Formation," *Int'l J. Computer Vision*, 1999.
13. R. Swaminathan, M. D. Grossberg, and S. K. Nayar, "Caustics of Catadioptric Cameras," *IEEE Proc. International Conference on Computer Vision (ICCV)*, 2001.
14. B. Micusik and T. Pajdla, "Autocalibration & 3D Reconstruction with Non-central Catadioptric Cameras," *IEEE Proc. Computer Vision and Pattern Recognition (CVPR)*, 2004.
15. Nuno Gonçalves, "Noncentral Catadioptric Systems with Quadric Mirrors: Geometry and Calibration," Ph.D. dissertation, University of Coimbra, 2008.
16. L. Perdigoto and H. Araujo, "Calibration of mirror position and extrinsic parameters in axial non-central catadioptric systems," *Computer Vision and Image Understanding*, 2013.
17. A. Agrawal and S. Ramalingam, "Single Image Calibration of Multi-Axial Imaging Systems," *IEEE Proc. Computer Vision and Pattern Recognition (CVPR)*, 2013.
18. M. A. Goodrich and A. C. Schultz, "Human-Robot Interaction: A Survey," *Foundations and Trends in Human-Computer Interaction*, 2007.
19. K. Chintamani, A. Cao, R. D. Ellis, and A. K. Pandya, "Improved Telemanipulator Navigation During Display-Control Misalignments Using Augmented Reality Cues," *IEEE Trans. Systems, Man and Cybernetics, Part A: Systems and Humans*, 2010.
20. I. Y.-H. Chen, B. MacDonald, and B. Wunsche, "Mixed Reality Simulation for Mobile Robots," *IEEE Proc. Int'l Conf. Robotics and Automation (ICRA)*, 2009.
21. H. Fuchs, M. A. Livingston, R. Raskar, D. Colucci, K. Keller, A. State, J. R. Crawford, P. Rademacher, S. H. Drake, and A. A. Meyer, "Augmented Reality Visualization for Laparoscopic Surgery," *Intl Conf. on Medical Image Computing and Computer Assisted Intervention (MICCAI)*, 2009.
22. N. Gonçalves, "On the reflection point where light reflects to a known destination in quadric surfaces," *Optics Letters*, 2010.
23. A. Agrawal, Y. Taguchi, and S. Ramalingam, "Beyond alhazen'Problem:Analytical Projection Model for Non-Central Catadioptric Cameras with Quadric Mirrors," *IEEE Proc. Computer Vision and Pattern Recognition (CVPR)*, 2011.
24. J. Hughes, A. V. Dam, M. Mcguire, D. F. Skylar, J. D. Foley, S. K. Feiner, and K. Akeley, *Computer Graphics: Principles and Practice (3rd Edition)*. Addison-Wesley, 2014.
25. L. Carpenter, "The A-buffer, an Antialiased Hidden Surface Method," *ACM SIGGRAPH*, 1984.
26. H. Gouraud, "Continuous Shading of Curved Surfaces," *IEEE Trans. Comput.*, 1971.

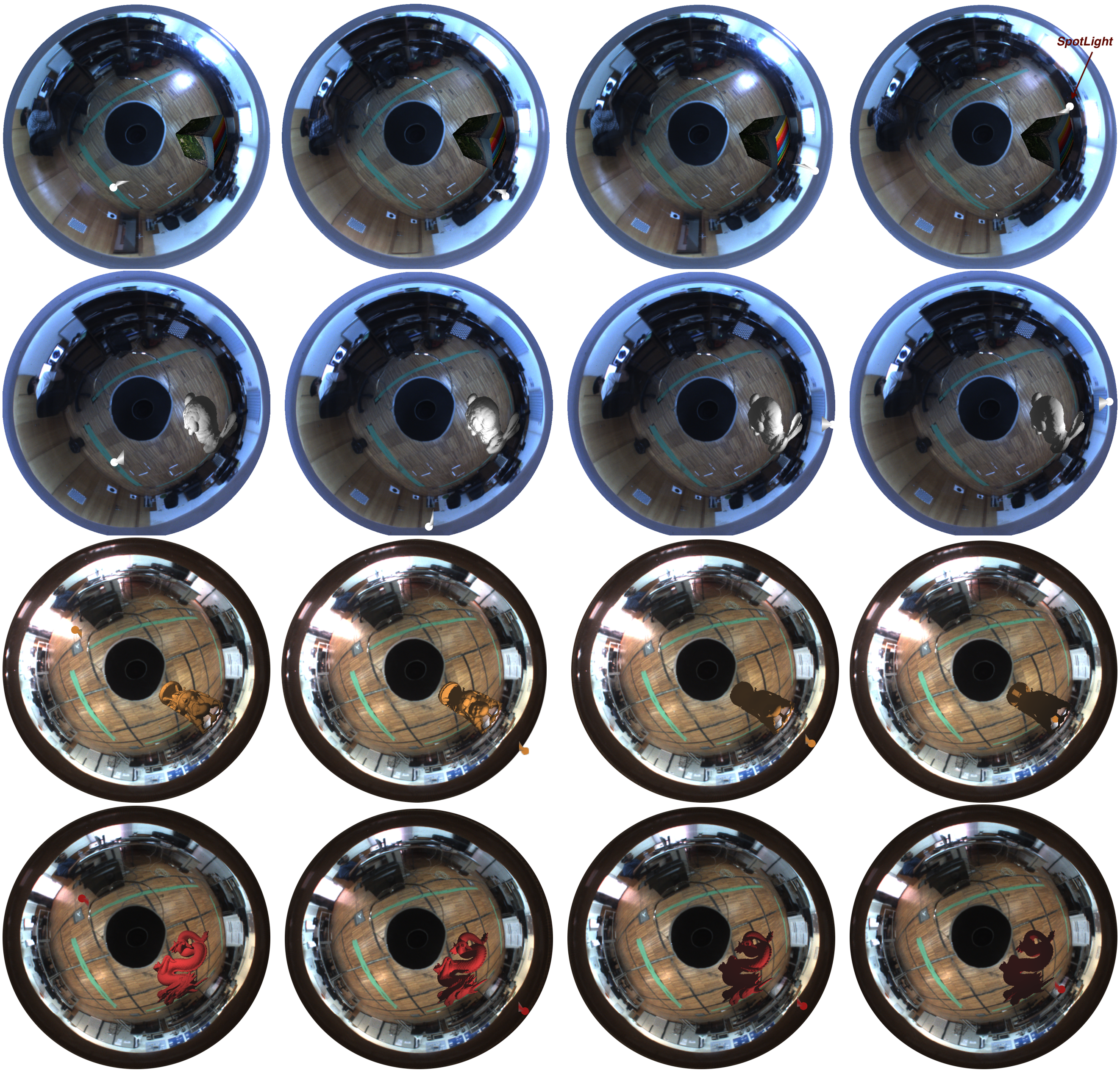


Fig. 9: Results of the application of the proposed framework, considering a moving spotlight. We used all the 3D objects: parallelepiped (first row), "bunny" (second row), "buddha" (third row) and "dragon" (fourth row). For the parallelepiped and the "bunny" we tested our framework without taking into account the illumination's occlusions between triangles using a Flat shading technique. For the "buddha" and "dragon", the illumination's occlusions were taken into account and a Gouraud shading technique was used (for more detail see Sec. 3.6). Videos (recorded in real time) with the complete sequences are sent in supplementary material.

27. B. T. Phong, "Illumination for Computer Generated Pictures," *Commun. ACM*, 1975.
28. Stanford University Computer Graphics Laboratory, "Stanford Bunny," https://graphics.stanford.edu/data/3Dscanrep/, 1993.
29. B. Delaunay, "Sur la sphere vide," *Izv. Akad. Nauk SSSR*, 1934.
30. C.-S. Chen and W.-Y. Chang, "On Pose Recovery for Generalized Visual Sensors," *IEEE Trans. Pattern Analysis and Machine Intelligence*, 2004.
31. G. Schweighofer and A. Pinz, "Globally Optimal O(n) Solution to the PnP Problem for General Camera Models," *Proc. British Machine Vision Conference (BMVC)*, 2008.
32. P. Miraldo and H. Araujo, "Pose Estimation for Non-Central Cameras Using Planes," *IEEE Int'l Conf. Autonomous Robot Systems & Competitions – ROBÓTICA*, 2014.
33. ——, "Planar Pose Estimation for General Cameras using Known 3D Lines," *IEEE/RSJ Proc. Int'l Conf. Intelligent Robots & Sys-*

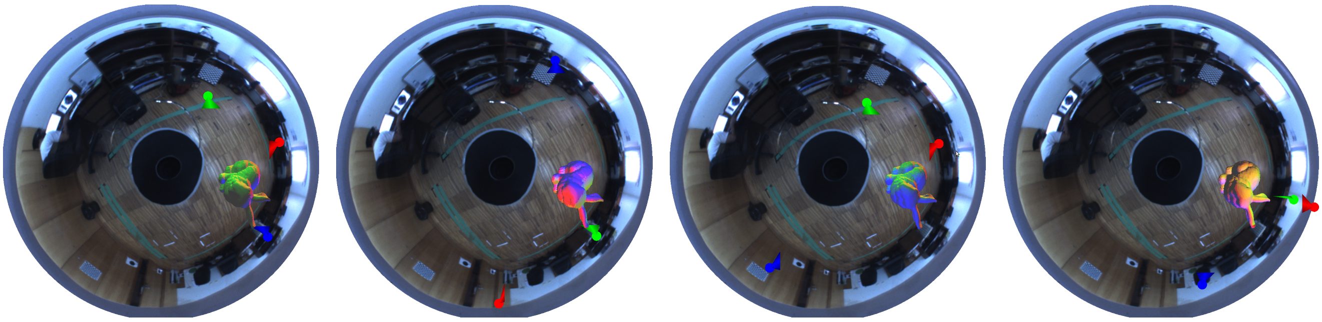


Fig. 10: In this figure we show a set of frames in which we apply the proposed framework, considering three moving spotlights with different colors (blue, green and red) affecting the Stanford bunny. To obtain this result three different movements were applied to each one of the spotlights to show that our solution, for the illumination step, is working correctly with the use of multiple spotlights in our framework.

tems (IROS), 2014.
34. MobileRobots, Inc., "Pioneer 3-DX," http://www.mobilerobots.com/Mobile_Robots.aspx.

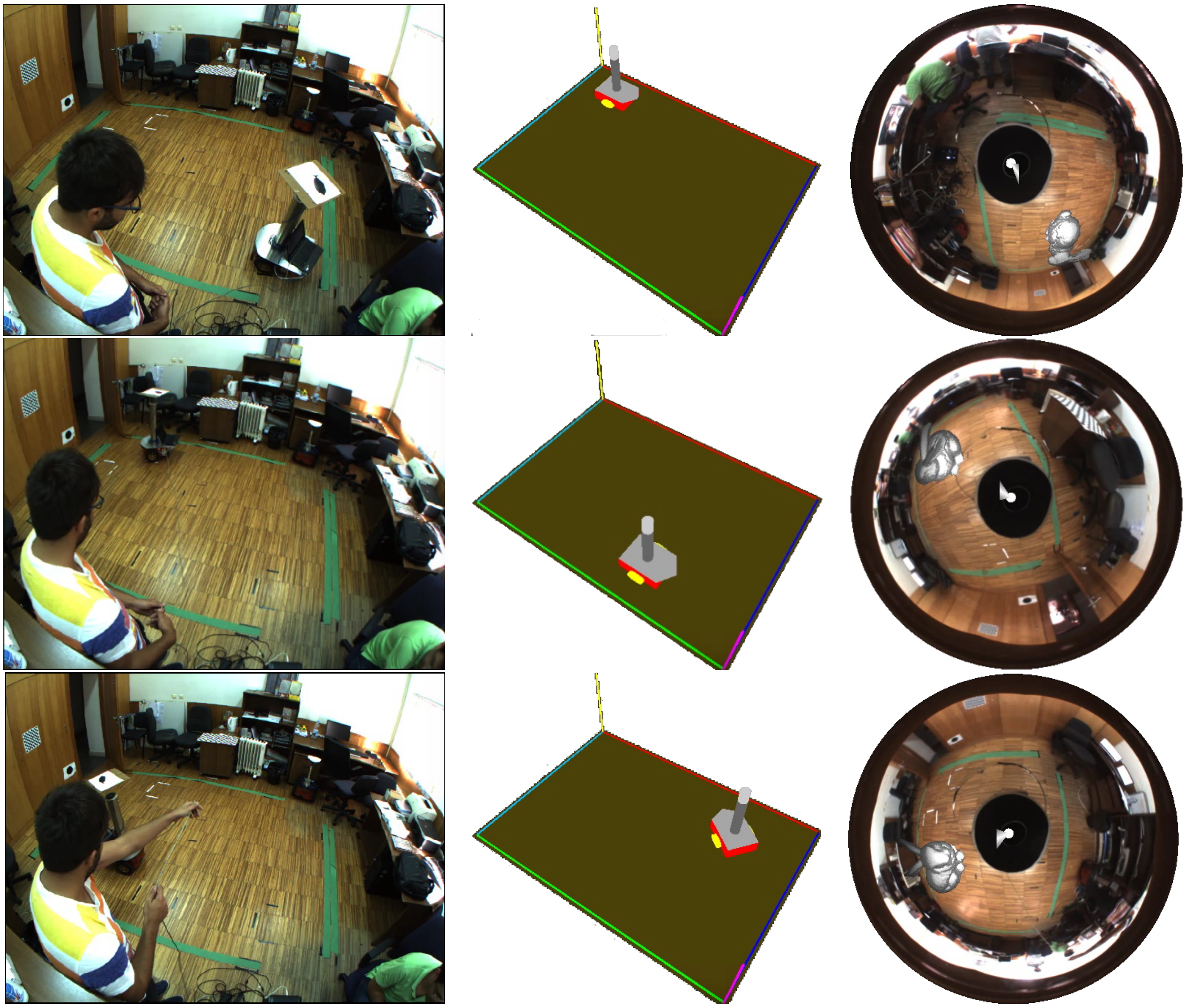


Fig. 11: Results of our framework for three different positions of the robot. On the left column, we present the image obtained by the auxiliar camera, which is acquiring the realtime events in the real world, on the center column, we show the 3D virtual arena showing the position of the robot in the arena and, on the right column, it is presented the result of our framework according to the position of the robot and light focus (which is on the top of the robot).