2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
Congress Center Hamburg
Sept 28 - Oct 2, 2015. Hamburg, Germany

Augmented Reality on Robot Navigation using Non-Central Catadioptric Cameras

Tiago Dias, Pedro Miraldo, Nuno Gonçalves, and Pedro U. Lima

Abstract— **In this paper we present a framework for the application of augmented reality to a mobile robot, using non-central camera systems. Considering a virtual object in the world with known local 3D coordinates, the goal is to project this object into the image of a non-central catadioptric imaging device. We propose a solution to this problem which allows us to project textured objects to the image in real-time (up to 20 fps): projection of 3D segments to the image; occlusions; and illumination. In addition, since we are considering that the imaging device is on a mobile robot, one needs to take into account the real-time localization of the robot. 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). To evaluate the proposed framework we test the solution using a mobile robot and a non-central catadioptric camera (using a spherical mirror).**

I. INTRODUCTION

Augmented reality has been studied for almost fifty years [4], [21]. As stated by Azuma [5], augmented reality can be defined as the projection of virtual 3D objects to the image plane. For the conventional perspective camera model, several methods have been presented, *e.g.* [13], [30], [12], [29]. 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. In 1996, Nalwa [24] introduced what he claims to be the first omni-directional system, 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). Omni-directional systems can be very useful for robot navigation, video surveillance systems or medical imaging devices where wide fields of view are fundamental.

With appropriate undistortion procedures, any central camera system can be modeled by a perspective camera [34], and, thus, 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 view point constraint (central cameras). Baker and Nayar [25], [6] studied the use of a single camera and a single quadric mirror to create omni-directional systems. The main problem is that, to get central systems, the camera must be perfectly aligned with the mirror's axis of symmetry and we must use a specific type mirror (for example, spherical mirrors cannot be used). Small misaligned systems or specific types of mirrors will not verify the constraint that all the projection lines intersect at a single 3D point, also denoted as viewpoint. Then, in practice, we will have a non-central camera system [33]. Later, because of the utility of these imaging devices, several authors proposed models and calibration methods for non-central catadioptric camera systems using general quadric mirrors and general position of the camera, relatively to the mirror *e.g.* [20], [26], [27], [2]. Contrarily to central camera systrems, generally, it is not possible to get undistorted images from non-central catadioptric cameras. Thus and in general, conventional techniques cannot be applied to these cases. In this paper it is proposed a framework for the use of augmented reality using these imaging devices. An example of the results are shown in Fig. 1.

Augmented reality can be extremely useful for human-robot interation [16], with several important applications in robotics. Two examples of these applications are: teleoperation [11] (creation and projection of 3D virtual landmarks to assist the human on robot navigation) and simulations on augmented reality environments [10] (creation and projection of 3D objects to simulate real scenarios). An example of an environment simulation (using augmented reality) is its application on medical surgeries (see *e.g.* [14]). Note that, medical doctors are used to work directly on distorted images.

The proposed framework is shown at Fig. 2. To get to our goal, we had to create new algorithms and reformulate some methods, so that they could be applied to our system. Assuming that we know the camera calibration and that our 3D object is triangulated and textured, the most challenge task is to project these 3D triangles (which form the 3D objects) to the image plane. Moreover, since we are dealing with a moving robot, we have to take into account the real-time localization of the robot's position, which can be represented as the estimation of the camera pose.

Our framework can be divided into two stages, which we will denote as pre-processing and real-time stages. Pre-processing stage will include all steps that can be computed *a priori* (avoiding unnecessary steps that could increase the computation time), while the real-time stage include the steps that depend on a certain parameters (that can vary) such as camera and light source positions. The main contributions of this paper are (further details, including state-of-the-art approaches, will be given in the next section):

T. Dias and N. Gonçalves are with the Institute for Systems and Robotics, Department of Electrical and Computer Engineering, University of Coimbra, Portugal, E-Mail: {tdias,nunogon}@isr.uc.pt. P. Miraldo and P. U. Lima are with the Institute for Systems and Robotics (LARSyS), Instituto Superior Técnico, University of Lisbon, Portugal. E-Mail: {pmiraldo,pal}@isr.tecnico.ulisboa.pt.

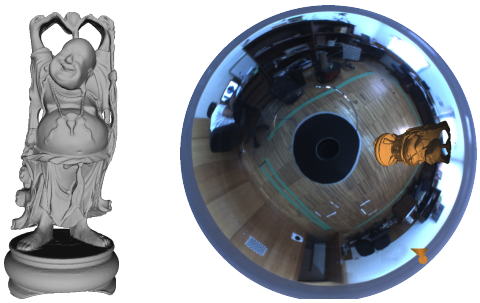


Fig. 1. Augmented reality on images of non-central catadioptric cameras. On the left image we present the original object and, on the right image, we present a result of the proposed framework.

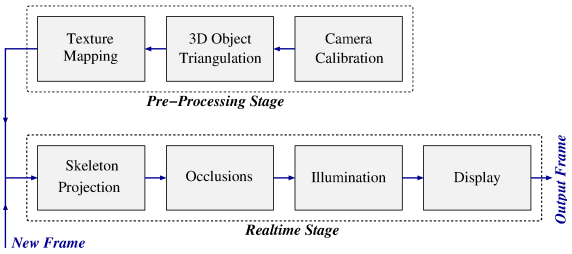


Fig. 2. Representation of the proposed framework for the application of augmented reality, using non-central catadioptric cameras. We divided the problem in two stages: pre-processing stage, where camera parameters and 3D object information is computed; and the realtime stage where the pre-processed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

- Use of augmented reality on an exact model using non-central catadioptric devices – which consists on the creation of a framework that works for non-central catadioptric taking into account the model's distortion;
- Projection of the object's skeleton – which consists in the projection of the object's segments to the non-central catadioptric image;
- Occlusions – one needs to verify if the pieces (already projected to the image) are overlapped and, if they are, verify which of them are visible or not;
- Illumination – illumination and shading will give shape to the projection of the 3D object.

We have implemented the proposed framework in C/C++ language. To obtain a better performance the CUDA toolkit (from NVIDIA) was used.

In Sec. II, we describe the proposed solution. In Sec. III, we show the results of the proposed framework using a non-central catadioptric camera (with a spherical mirror) on a mobile robot and, in Sec. IV, we give the conclusions of the paper.

II. Augmented Reality using Non-Central Catadioptric Cameras

As it was previously explained, we divided the pipeline in two stages: pre-processing and realtime stages, see Fig. 2. To get the final results, one have to take into account the following steps: camera calibration, 3D object triangulation, skeleton projection, occlusions, and illumination. In this paper we assume that our 3D object is rigid and static. The two stages are described in the following subsections.

A. Pre-Processing Stage

Pre-processing stage is composed by two steps (see Fig. 2): 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. Camera calibration consists in the estimation of the parameters that represent this mapping. For a non-central catadioptric system, this is achieved by computing the camera intrinsic parameters, the mirror parameters, and the transformation between the camera and mirror [27].

The second step of this 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. Similar to most of state-of-the-art approaches, we used the segmentation of the 3D virtual object in 3D triangles. We test our method using a virtual cube (which we had to triangulate) and two well known objects in computer graphics, the Stanford "bunny" and the "happy Buddha" (already triangulated) [32].

B. Real-time Stage

Real-time stage corresponds to the methods that have to be computed, each time a new image frame is received. As a result, we include the following four steps: skeleton projection, occlusions, illumination, and display. All these steps depend on the geometry of the imaging device and, as we describe in the introduction, since for images of non-central camera models we cannot get unwrapped images, new algorithms have to be defined [34].

1) Projection: Assuming that we know the camera calibration and that our 3D object is triangulated and textured, one of the most challenge task is the projection of these 3D triangles (which form the 3D objects) to the image plane. Assuming that these triangles are small enough, the effects of distortion are neglectable [34]. To avoid complex parameterizations that certainly require more computation effort, in this paper we will consider a large number of very small triangles, thus ignoring the distortion on the projection of 3D triangles. As a result, we just need to consider the projection of three 3D points (that form the vertices of the triangles) to the image plane. Contrarily to the projection of 3D points to the image of a perspective camera, the projection for non-central catadioptric systems is quite complex (*e.g.* [15], [3]. In addition, one has to verify if the coordinate system of the virtual object is aligned with the camera's coordinate system. This problem is known as the absolute pose problem. Let us consider 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. Originally, we know the 3D coordinates of points in the world frame (vertices of the 3D triangles that define the object). Let us denote these points as $\mathbf{p}^{(\mathcal{W})} \in \mathbb{R}^3$. The goal is to compute the rigid transformation $\mathbf{H}^{(\mathcal{CW})} \in \mathbb{R}^{4\times4}$ that transform points from the world to the camera coordinate systems such that

$$\widetilde{\mathbf{p}}^{(\mathcal{C})} \sim \mathbf{H}^{(CW)}\,\widetilde{\mathbf{p}}^{(\mathcal{W})}, \tag{1}$$

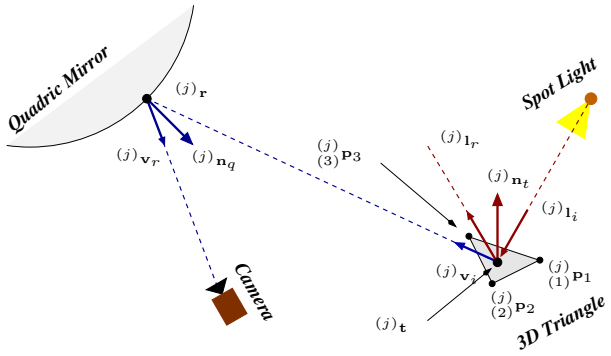


Fig. 3. Representation of both projection of 3D point to the image of a general catadioptric camera and its respective illumination.

where $\widetilde{\mathbf{p}}$ denotes the homogeneous representation of $\mathbf{p}$. Several authors addressed this problem, *e.g.* [9], [31], [23]. In this paper we used [22]. 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, we will assume that 3D points are already known in the camera coordinate system.

Let us denote the vertices of the triangles as ${}^{(j)}_{(i)}\mathbf{p}$ (i^{th} vertex of the j^{th} triangle). The goal of this step is to compute the respective reflection point in the mirror ${}^{(j)}_{(i)}\mathbf{r}$ (see Fig. 3). To compute this point, one can use for example [15], [3]. These methods are quite complex and, since the goal in this paper is not to address this problem, we will consider this as a black box. However, one have to take into account the computation effort required for this projection. Unlike the perspective case, where the projection of 3D points only requires a simple and fast matrix multiplication (matrix of the camera's internal parameters times the 3D point), the computation of the exact reflection point ${}^{(j)}_{(i)}\mathbf{r}$ requires much more computation effort (this is very important for the next steps). Using this approach, we can now assume that we have the projection of all the 3D triangles that form the object. We will denote these triangles as

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

${}^{(j)}_{(i)}\mathbf{u}$ are the coordinates of the vertices on the image plane and $\mathbf{K} \in \mathbb{R}^{3\times 3}$ are the camera intrinsic parameters [18]. A graphical representation of the proposed solution is shown in Fig. 3. On Fig. 4(a) we show a skeleton projection example of the cube object.

Since we already have the image coordinates vertices of each triangle, the matching of each texture is given by a simple affine transformation between the texture on the 3D triangle and the triangle on the image. Fig. 4(d) shows these results.

2) Occlusions: Occlusions is a very well known problem in 3D computer graphics. For perspective cameras, several solutions were proposed (*e.g.* the Painter's algorithm [19], Z-Buffer (also known as Depth Buffer) [19], and A-Buffer [8]).

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 $\left\{ {}^{(j)}_{(1)}\mathbf{p}, {}^{(j)}_{(2)}\mathbf{p}, {}^{(j)}_{(3)}\mathbf{p} \right\}$;
  Compute ${}^{(j)}\mathbf{r_t}$ using [15], [3] ;
  Set ${}^{(j)}\xi$ as the distance between ${}^{(j)}\mathbf{r}$ and ${}^{(j)}\mathbf{t}$;
end
Sort all the triangles by descendant order using the computed ${}^{(j)}\xi$, for all $j = 1, \ldots, N$;

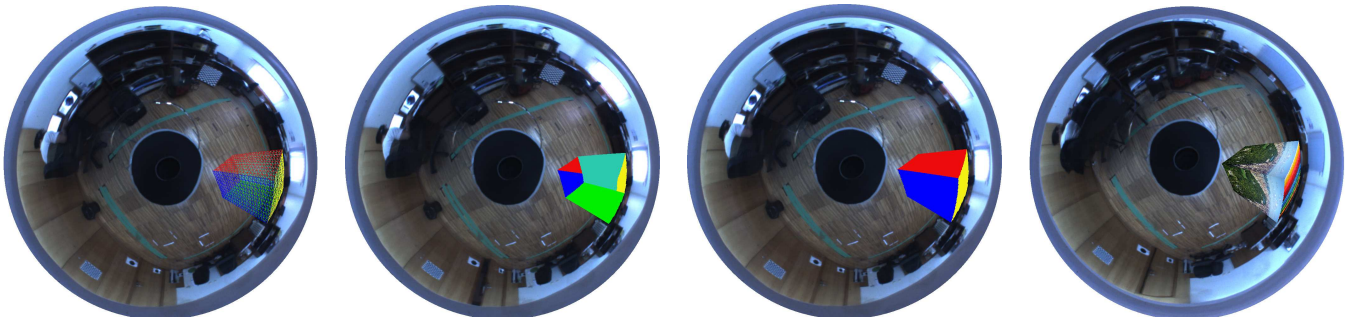
Z-Buffer is the simplest and most used technique. However, this method requires the association between pixels and coordinates of 3D points on the object, for all pixels that define the object. We want to avoid this because of the complexity associated with the projection of points on non-central catadioptric systems (described in the previous section). Moreover, as described in the previous section, we are ignoring the distortion effects on the projection of the triangles (by considering a large number of 3D small triangles) which means that, using this formulation, there is no easy way to precisely associate pixels with 3D points that belong to the objects.

Since we already have the projection of the triangles (with an associated texture), the goal is just to check which triangles are in front and make sure that they are visible. Then, we propose a simple solution based on painter's algorithm methodology. Since we are using non-central catadioptric imaging systems, conventional algorithms cannot be used. These methods need to be reformulated, taking into account the geometry of these imaging devices. The goal of painter's methodology is to organize all 3D triangles as a function of the distance between each triangle and the camera system. Then, the problem is solved by displaying the 2D triangles using this order. If for central cameras one can use the camera center (also called the effective view point [18]) as the referencial for the distance, in our problem this cannot be applied (non-central catadioptric system). To compute the distance between the 3D triangles and the camera system we thus consider the distance between the triangle (we use the mass center of the triangle) and the respective 3D reflection point on the mirror (see Fig. 3). This step is formalized in Algorithm 1. After the application of this algorithm, we have the 2D triangles in descending order and ready to be displayed. The effect of this step can be seen by Fig. 4(b) (without applying the proposed algorithm) and Fig. 4(c) (after the application of Algorithm 1).

3) Illumination: When considering a 3D object with a solid color without illumination, the projection of this object to the image will be a BLOB (Binary Large OBject), see Fig. 6(a). The use of an illumination model and a shading technique will create the illusion of shape on a projected object. For perspective cameras, to compute the intensity of light associated with a single pixel (illumination), two models were proposed: Phong reflection model and Torrance-

$$ {}^{(j)}I^{(\mathrm{ch})} = \overbrace{K_e^{(\mathrm{ch})} + G_a^{(\mathrm{ch})} K_a^{(\mathrm{ch})}}^{\tilde{I}^{(\mathrm{ch})}} + \sum_{k=1}^{M} \underbrace{spot_k \left({}_{(k)}L_a^{(\mathrm{ch})} K_a^{(\mathrm{ch})} + f_k \left({}_{(k)}L_d^{(\mathrm{ch})} K_d^{(\mathrm{ch})} \left(\max \left\{ -{}^{(j)}_{(k)}\mathbf{l}_i{}^T \, {}^{(j)}\mathbf{n}_t, 0 \right\} \right) + {}_{(k)}L_s^{(\mathrm{ch})} K_s^{(\mathrm{ch})} \left(\max \left\{ {}^{(j)}\mathbf{v}_i{}^T \, {}^{(j)}_{(k)}\mathbf{l}_r, 0 \right\} \right)^{sh} \right) \right)}_{{}^{(j)}\tilde{I}_k^{(\mathrm{ch})}} \quad (3) $$

M is the number of spotlights; $K_a^{(\mathrm{ch})}$, $K_d^{(\mathrm{ch})}$, $K_s^{(\mathrm{ch})}$ $K_e^{(\mathrm{ch})}$ and sh are ambient, diffuse, specular, emission, shininess material color intensities; $G_a^{(ch)}$ 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 intensities of the k^{th} spotlight; boolean parameters f_k are used to control whether a triangle is illuminated or not; and $spot_k$ controls the cutoff angle of the spotlight. A graphical representation of directions ${}^{(j)}_{(k)}\mathbf{l}_i$, ${}^{(j)}_{(k)}\mathbf{l}_r$, ${}^{(j)}_{(j)}\mathbf{n_t}$, and ${}^{(j)}\mathbf{v}_i$ is shown in Fig. 3.



(a) (b) (c) (d)

Fig. 4. Results of the application of first two steps of the pipelines realtime stage, applied to the 3D virtual cube. Fig. (a) represents the projection of the 3D triangles (that define the 3D object) to the image, which correspond to the skeleton projection step of the pipeline. The goal of Fig. (b) and (c) is to show the effects of the occlusion step and in Fig. (d) we show the result of the occlusion step with textured faces.

Sparrow reflection model (both described at [7]). Also, for the shading problem, several techniques were proposed, such as: Flat shading [19], Gouraud shading [17], and Phong shading [28]. We want to stress out that this step depends on the geometry of the imaging device, which means that these conventional techniques cannot be used directly. Then, new solutions have to be derived.

Again, we decided to derive a very simple method using a large number of small triangles. Thus, we can analyze the complete illumination of each 3D triangle as a single point of illumination (flat shading technique). Then, we consider that the complete illumination of each 3D triangle is equal to the illumination of its mass center. To compute the illumination parameters, we rewrite the well known Phong's reflection model [7], taking into account the image formation of a non-central catadioptric system. Note that, we could use the variations of other methodologies ([28], [17]) instead of considering only the illumination of mass center. However this would require computations for more points of the triangle, which means that these methods would bring unnecessary computational time. The proposed illumination equation (including several light sources and their interactions with the physical materials) for the j^{th} triangle is, then, expressed by (3) (on the top of page 4) for all color channels. The proposed solution is formalized in Algorithm 2. Results after using the proposed illumination algorithm can be seen in Fig. 6(b) for the "buddha" object.

III. Experiments

To test our framework, we used a non-central catadioptric camera formed with a perspective camera and a spherical mirror, mounted on a mobile robot (Pioneer 3D-X [1]). To calibrate the non-central catadioptric camera, we used the method proposed by Perdigoto and Araujo [27] and the pose (which have to be computed each time a new frame is received) was computed using [22]. A virtual light source was included at the top of the mobile robot. For the illumination parameters (parameters of (3)), we chose to cover our virtual objects with silver, which is a well-known and standard material in computer graphics. Additionally, our light source will be treated as a spotlight (positional and directional light source), that moves with the robot. We defined $L_a^{(ch)}$, $L_d^{(ch)}$ and $L_s^{(ch)}$ (light source parameters) to be white for the cube and the "bunny" objects and gold for the "buddha" object. For the global ambient light property ($G_a^{(ch)}$) we used standard values for each of the RGBs components. We predefined a path through the arena and set the position of the virtual object in the middle. We used a laptop with CPU "Intel i7 3630QM" (2.4 GHz with 4 cores) and GPU "NVIDIA GeForce GT 740M" (810 MHz with 384 CUDA cores) to run the complete framework and got up to 20fps (for objects with approximately 40K triangles). The results for three frames are shown in Fig. 5. A video with the complete sequence is sent in the supplementary material.

Other results, using a moving spotlight (with different movements) for the cube and the "buddha" objects are sent in

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 and ${}_{(k)}\mathbf{d}_{sl}$ the direction of the spotlight:

for $j = 1$ *to* N **do**
- Compute the normal of the j^{th} triangle ${}^{(j)}\mathbf{n}_t$;
- Compute the mass center ${}^{(j)}\mathbf{t}$;
- Compute the reflection point ${}^{(j)}\mathbf{t} \mapsto {}^{(j)}\ \mathbf{r}_t$;
- Compute the visualization vector ${}^{(j)}\mathbf{v}_i$;
- Set ${}^{(j)}I^{(\text{ch})} = \tilde{I}^{(\text{ch})}$ (see (3));
- **for** $k = 1$ *to* M **do**
  - Compute ${}^{(j)}_{(k)}\mathbf{l}_r$;
  - Set $f_k = 1$ and $spot_k = 0$;
  - **if** *angle between* ${}^{(j)}_{(k)}\mathbf{l}_i$ *and* ${}^{(j)}\mathbf{n}_t$ *bigger than zero* **then**
    - $f_k = 0$;
  - **end**
  - **if** *maximum of* $\left\langle {}^{(j)}_{(k)}\mathbf{l}_{i,(k)}\ \mathbf{d}_{sl} \right\rangle$ *and* 0 *bigger than* ${}_{(k)}C^{te}$ **then**
    - $spot_k = \max\left\{ {}^{(j)}_{(k)}\mathbf{l}_i{}^{T}\ {}_{(k)}\mathbf{d}_{sl}, 0 \right\}^{\mathcal{E}}$;
  - **end**
  - Add ${}^{(j)}I^{(\text{ch})} = {}^{(j)}\ I^{(\text{ch})} + {}^{(j)}\ \breve{I}^{(\text{ch})}_k$, see (3) – top of page 4;
- **end**

end

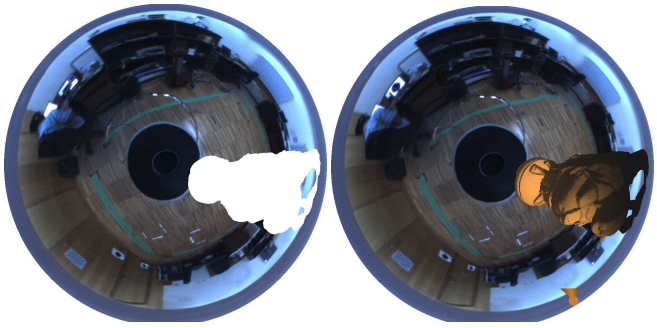
supplementary material (results related to Figs. 4(d) and 6(b) presented in the previous section).

IV. Conclusions

In this paper we address Augmented Reality for images of non-central catadioptric. We believe that this is the first time that this problem is addressed. Theoretically, the goal is to identify differences between Augmented Reality on conventional perspective cameras vs on non-central catadioptric cameras. We saw that, 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 projected triangles. After identifying and understanding these problems, we proposed changes to conventional techniques to solve the problem. From the experimental result, we conclude that the proposed solutions work very well, with acceptable computation effort. As future work, we would like to highlight some changes that could improve the proposed framework. 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 associated with the projection of the 3D triangles. However, if this distortion can be accounted for the projection of 3D triangles, a smaller number of triangles could be used, which could decrease the computation time. Another improvement that we intend to consider are shadows of the virtual objects projected to the real scene, as well as the direct effect of the spotlight on the real scene.

Acknowledgments

P. Miraldo and P. U. Lima were supported by the European Commission Project RoCKIn [FP7-ICT-601012] and partially supported by FCT through the project [UID/EEA/50009/2013]. T. Dias and N. Gonçalves are also supported by FCT through the project [UID/EEA/00048/2013].



(a) (b)

Fig. 6. Results of the application of the illumination step to the "happy buddha" object. In Fig. (a) we present the result of the framework without using illumination. In Fig. (b) we present the same result using the illumination step. Note that, for the "happy buddha" object, we used a gold color for the spotlight.

References

[1] Adept MobileRobots, Inc., "Pioneer 3-DX," http://www.mobilerobots.com/Mobile_Robots.aspx.

[2] A. Agrawal and S. Ramalingam, "Single Image Calibration of Multi-Axial Imaging Systems," *IEEE Proc. Computer Vision and Pattern Recognition (CVPR)*, 2013.

[3] 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.

[4] A. Appel, "Some techniques for shading machine renderings of solids," *Proc. of American Federation of Information Processing Societies (AFIPS)*, 1968.

[5] R. T. Azuma, "A Survey of Augmented Reality," *Presence:Teleoperators and Virtual Environments: MIT Press Journal*, 1997.

[6] S. Baker and S. K. Nayar, "A Theory of Single-Viewpoint Catadioptric Image Formation," *Int'l J. Computer Vision*, 1999.

[7] J. F. Blinn, "Models of Light Reflection for Computer Synthesized Pictures," *ACM SIGGRAPH*, 1977.

[8] L. Carpenter, "The A-buffer, an antialiased hidden surface method," *ACM Proc. of SIGGRAPH*, 1984.

[9] C.-S. Chen and W.-Y. Chang, "On Pose Recovery for Generalized Visual Sensors," *IEEE Trans. Pattern Analysis and Machine Intelligence*, 2004.

[10] I. Y.-H. Chen, B. MacDonald, and B. Wünsche, "Mixed Reality Simulation for Mobile Robots," *IEEE Proc. Int'l Conf. Robotics and Automation (ICRA)*, 2009.

[11] 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.

[12] P. Debevec, "Rendering Synthetic Objects into Real Scenes: Bridging Traditional and Image-based Graphics with Global Illumination and High Dynamic Range Photography," *ACM Proc. SIGGRAPH*, 2008.

[13] A. Fournier, A. S. Gunawan, and C. Romanzin, "Common Illumination between Real and Computer Generated Scenes," *Proceeding of Graphics Interface (GI'93)*, 1993.

[14] H. Fuchs, M. A. Livingston, R. Raskar, D. Colucci, K. Keller, J. R. Crawford, P. Rademacher, S. H. Drake, and A. A. Meyer, "Augmented Reality Visualization for Laparoscopic Surgery," *Int'l Conf. Medical Image Computing and Computer Assisted Intervention (MICCAI)*, 1998.

[15] N. Gonçalves, "On the reflection point where light reflects to a known destination in quadric surfaces," *Optics Letters*, 2010.

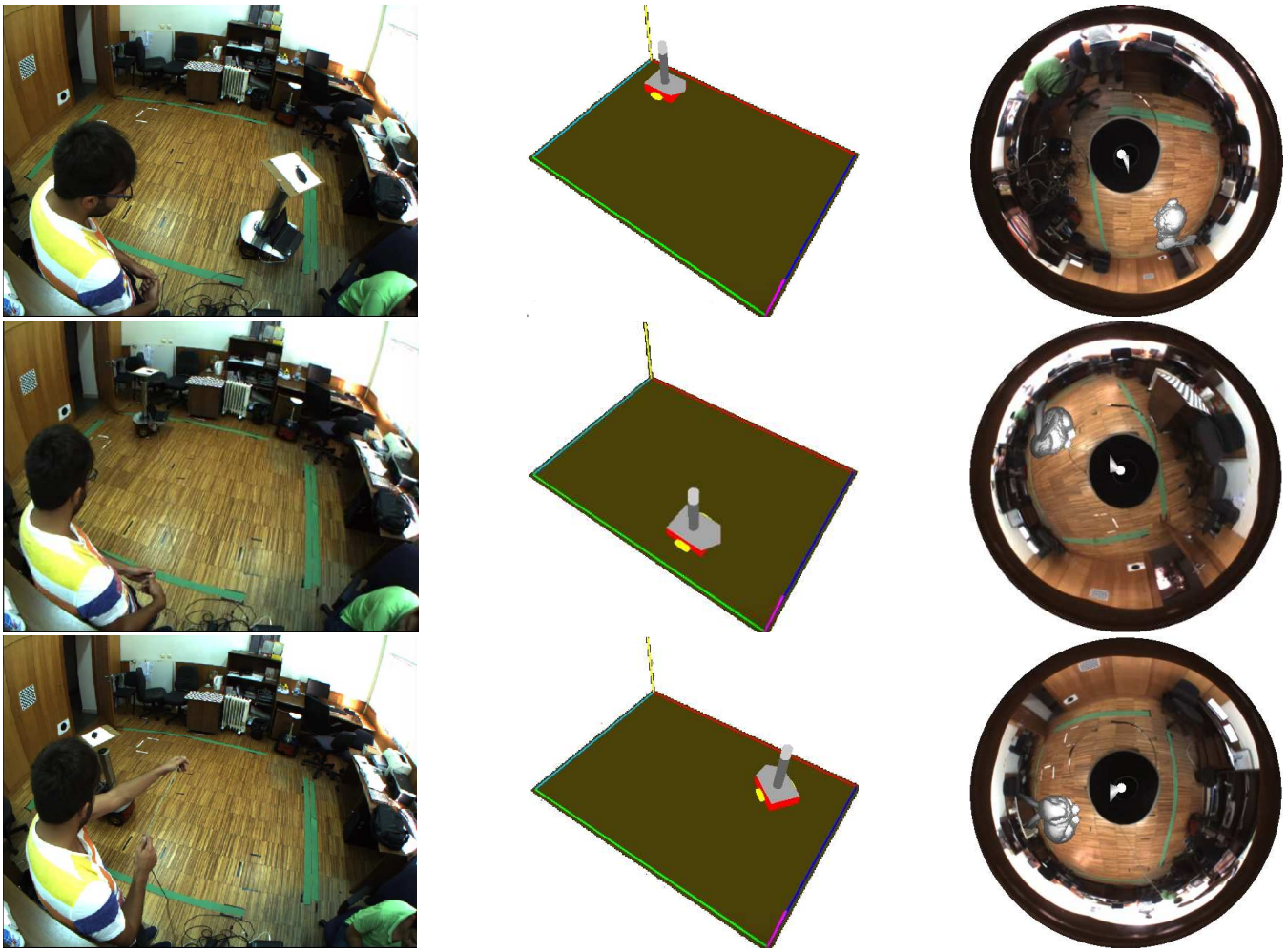


Fig. 5. Results of our framework for three different positions of the robot. On the left column, we present the image obtained by the auxiliar camera, which is acquiring the realtime events in the real world, on the center column, we show the 3D virtual arena showing the position of the robot in the arena and, on the right column, 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).

[16] M. A. Goodrich and A. C. Schultz, "Human-Robot Interaction: A Survey," *Foundations and Trends in Human-Computer Interaction*, 2007.

[17] H. Gouraud, "Continuous Shading of Curved Surfaces," *IEEE Trans. Comput.*, 1971.

[18] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2000.

[19] J. Hughes, A. V. Dam, M. Mcguire, D. F. Skylar, J. D. Foley, S. K. Feiner, and K. Akeley, *Computer Graphics:Principles and Practice Third Edition*. United States of America: Addison-Wesley, 2014.

[20] B. Micusik and T. Pajdla, "Autocalibration & 3D Reconstruction with Non-central Catadioptric Cameras," *IEEE Proc. Computer Vision and Pattern Recognition (CVPR)*, 2004.

[21] P. Milgram and F. Kishino, "A Taxonomy of Mixed Reality Visual Displays," *IEICE Trans. Information and Systems*, 1994.

[22] P. Miraldo and H. Araujo, "Planar Pose Estimation for General Cameras using Known 3D Lines," *IEEE/RSJ Proc. Int'l Conf. Intelligent Robots & Systems (IROS)*, 2014.

[23] ——, "Pose Estimation for Non-Central Cameras Using Planes," *IEEE Int'l Conf. Autonomous Robot Systems & Competitions – ROBÓTICA*, 2014.

[24] V. S. Nalwa, "A True Omni-Directional Viewer," *Technichal report, Bell Laboratories*, 1996.

[25] S. K. Nayar and S. Baker, "Catadioptric Image Formation," *Proceedings of the 1997 DARPA Image Understanding Workshop*, 1997.

[26] Nuno Gonçalves, "Noncentral Catadioptric Systems with Quadric Mirrors: Geometry and Calibration," Ph.D. dissertation, University of Coimbra, 2008.

[27] L. Perdigoto and H. Araujo, "Calibration of mirror position and extrinsic parameters in axial non-central catadioptric systems," *Computer Vision and Image Understanding*, 2013.

[28] B. T. Phong, "Illumination for Computer Generated Pictures," *Commun. ACM*, 1975.

[29] A. L. Santos, D. Lemos, J. E. F. Lindoso, and V. Teichrieb, "Real Time Ray Tracing for Augmented Reality," *IEEE Symposium Virtual and Augmented Reality (SVR)*, 2012.

[30] I. Sato, Y. Sato, and K. Ikeuchi, "Acquiring a Radiance Distribution to Superimpose Virtual Objects onto a Real Scene," *IEEE Trans. Visualization and Computer Graphics*, 1999.

[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] Stanford University Computer Graphics Laboratory, "Stanford Bunny," https://graphics.stanford.edu/data/3Dscanrep/, 1993.

[33] R. Swaminathan, M. D. Grossberg, and S. K. Nayar, "Caustics of Catadioptric Cameras," *IEEE Proc. Int'l Conf. Computer Vision (ICCV)*, 2001.

[34] ——, "A Perspective on Distortions," *IEEE Proc. Computer Vision and Pattern Recognition (CVPR)*, 2003.