ArUcOmni: detection of highly reliable fiducial markers in panoramic images

Jaouad Hajjami
Forssea Robotics, Paris, France
L@bISEN, Vision-AD team, Brest, France
jaouad.hajjami@isen-ouest.yncrea.fr

Jordan Caracotte
UPJV, MIS lab,
Amiens, France
jordan.caracotte@u-picardie.fr

Guillaume Caron
CNRS-AIST JRL, UMI3218/IRL, Tsukuba, Japan
UPJV, MIS lab, Amiens, France
guillaume.caron@u-picardie.fr

Thibault Napoléon
L@bISEN, Vision-AD team
Brest, France
thibault.napoleon@isen-ouest.yncrea.fr

Abstract

In this paper, we propose an adaptation of marker detection algorithm for panoramic cameras such as catadioptric and fisheye sensors. Due to distortions and non-uniform resolution of such sensors, the methods that are commonly used in perspective images cannot be applied directly. This work is in contrast with the existing marker detection framework: Automatic reliable fiducial markers Under occlusion (ArUco) for a conventional camera. To keep the same performance for panoramic cameras, our method is based on a spherical representation of the image that allows the marker to be detected and to estimate its 3D pose. We evaluate our approach on a new shared dataset that consists of a 3D rig of markers taken with two different sensors: a catadioptric camera and a fisheye camera. The evaluation has been performed against ArUco algorithm without rectification and with one of the rectified approaches based on the fisheye model.

1. Introduction

Panoramic cameras provide a $360° \times 180°$ field of view, much larger than the usual $60°$ of conventional cameras. A very wide field of view leads to more robust and reliable vision-based estimation such as visual odometry [23], robot navigation [14, 6] or visual tracking and servoing [20, 13].

In fact, panoramic sensors are already widely used in various fields, such as video surveillance [19, 12] or 3D reconstruction [18]. However, very few works tackled marker detection in panoramic images, despite basic markers [17] that are not reliable as markers for conventional vision [7, 11, 5, 21] commonly used for augmented reality. The latter mostly fails with panoramic images, as our results show, and one way to solve the issue would be to rectify the panoramic images. Berveglieri *et al*. [4] rectified the fisheye image into four horizontal lateral views, then they were used for camera calibration using ArUco [7] fiducial markers. The main drawback of the rectified approach most often yields to lose some field of view by cropping the boundaries of the image.

In this paper, we start in Sec. 2 with a review of the pipeline of marker detection algorithm integrated in ArUco. We then briefly present an approach for image rectification that we call ArUco-rectified, followed by a detailed presentation of our approach called ArUcOmni for marker detection using a spherical model and pose estimation. In Sec. 3, we evaluate our approach (ArUcOmni) against ArUco-rectified and to the conventional method (ArUco). Finally, in Sec. 4, we conclude this paper with perspectives for future work.

2. ArUco pipeline

In this section, we will start with a brief recall of the main steps of the marker detection algorithm for ArUco. An ArUco fiducial marker can be detected with the following steps:

- Image binarization with adaptive thresholding.
- Contours extraction in the binary image.
- Contours approximation to polygons and selection of those with four vertices only.
- Homography estimation to get the canonical form of the potential marker (a square).
- Homography application to remove the perspective (warping).
- Binary code extraction from the canonical image of the marker.
- Subpixel Corners refinement.

- Pose estimation.

All these steps should be adapted to the spherical form of the image due to very strong distortions [15, 16]. In this article, we will demonstrate how we adapt the key steps only of the algorithm to improve the performance in terms of rate of marker detection and also for pose estimation. Therefore, in the next section we will be focusing on the following three steps:

- Homography estimation to get the canonical form of the potential marker.
- Homography application to remove the perspective transformation (warping).
- Pose estimation.

Homographies are projective transformations and therefore remain valid for fisheye and catadioptric sensors of single viewpoint. In the first step, we will consider the homography as an inhomogeneous set of linear equations and solve them with a linear solver. After that, we apply the solution to warp the image into a canonical form. The last step of the algorithm is the pose estimation using 2D/3D combination of the corners of the marker. For that, we first estimate the pose linearly and then we use that as initialization to a non-linear optimization which consists of minimizing the global reprojection error using Levenberg-Marquardt algorithm.

2.1. ArUco-rectified

One of the ways to indirectly process an image taken with a wide angle camera is by rectification. In this article and for a comparison purpose, we will use a method from Kannala *et al.* [10] for central wide-angle cameras like the ones we're using, catadioptric and fisheye sensors. The full model to get the distorted coordinates $\mathbf{x}_d = (x_d, y_d)$ is as follows:

$$\mathbf{x}_d = r(\theta)i(\phi) + \Delta_r(\theta,\phi)i(\phi) + \Delta_t(\theta,\phi)j(\phi) \quad (1)$$

Where, $i(\phi)$ and $j(\phi)$ are the unit vectors in the radial and tangential directions respectively, $\Delta_r(\theta,\phi)$ and $\Delta_t(\theta,\phi)$ are the two distorsion terms that acts in the radial and tangential direction.

This approach rectifies the image in such a way to get the same properties as the pinhole model like straight lines to remain straight under perspective projection, which is a very useful property in the case of marker detection when they feature straight lines ArUco ones.

2.2. ArUcOmni

In this section, we start with a brief description of the spherical perspective projection that will be used in this article. Geyer [8] and Barreto [3] had been working on the unified spherical model for central catadioptric including some fisheye cameras. From a theoretical point of view, fisheye lenses do not respect the single point of view because they consist of a set of lenses that do not respect a single projection center but it was proven by various works, Ying *et al.* [22] for example, that the unified spherical projection model is a very good approximation. This is the basic step to estimating the homography through the sphere of that model. We will also see how to apply the homography to warp the image.

2.2.1 The unified spherical model

A projection of a 3D point into a pixel through a sphere can be done using the following projections:

Perspective projection: The perspective projection maps 3D points onto a normalised plane followed by a mapping to the image plane. The complete projection is obtained by:

- Projection onto the normalised plane:

$$\mathbf{x} = (x, y, 1) = pr(\boldsymbol{X}) = (\frac{X}{Z}, \frac{Y}{Z}, 1) \quad (2)$$

- Projection onto the image plane:

$$\mathbf{u} = (u, v, 1) = K\boldsymbol{x}^t \quad (3)$$

Where, $\mathbf{X} = (X, Y, Z)$ is a 3D point in the camera frame, x is a point in the normalised plane, K is the camera projection matrix:

$$K = \begin{pmatrix} f_x & 0 & u_0 \\ 0 & f_y & v_0 \\ 0 & 0 & 1 \end{pmatrix} \quad (4)$$

Where, f_x and f_y are the horizontal and the vertical focal lengths respectively.

We summarize the complete perspective projection with:

$$pr_\gamma(\mathbf{X}) = Kpr(\mathbf{X}) \quad (5)$$

Where, $\gamma = \{f_x,\ f_y,\ u_0,\ v_0\}$ represents the intrinsic camera parameters estimated by the calibration of the camera.

Stereographic projection: The spherical model that will be using in this paper has been proposed by Barreto [2]. Barreto's model is based on a double projection via a unit sphere $\mathbb{S}^2 = \{\mathbf{Xs} = (X_s, Y_s, Z_s) \in \mathbb{R}^3 / \|\mathbf{Xs}\| = 1\}$ centered on the mirror focal point and without considering the optical distortion. The convention adopted in this paper is slightly different in terms of the axis-convention in which we consider the z-axis of the camera pointing backwards.

The following steps describe the different projections involved:

- Projection onto a unit sphere:

$$\mathbf{Xs} = (\frac{X}{\rho}, \frac{Y}{\rho}, \frac{Z}{\rho}) \quad (6)$$

Where, $\rho = \sqrt{X^2 + Y^2 + Z^2}$.

- Projection onto the normalised image plane with a different projection center in which the unit sphere has a coordinate $(0\,0\,\xi)^t$:

$$\mathbf{x} = (x, y, 1) = (\frac{Xs}{Zs+\xi}, \frac{Ys}{Zs+\xi}, 1) \quad (7)$$

Where, ξ depends on the geometry of the mirror.
- Projection onto the image plane using formula-3.

We can also get the spherical coordinates from the image plane through the inverse of the projection presented above, formulas-6-7:

$$\mathbf{Xs} = pr_{\xi}^{-1}(\mathbf{x}) = \begin{pmatrix} \frac{\xi+\sqrt{1+(1-\xi^2)(x^2+y^2)}}{x^2+y^2+1}x \\ \frac{\xi+\sqrt{1+(1-\xi^2)(x^2+y^2)}}{x^2+y^2+1}y \\ \frac{\xi+\sqrt{1+(1-\xi^2)(x^2+y^2)}}{x^2+y^2+1} - \xi \end{pmatrix} \quad (8)$$

The formula below is the complete projection to project a 3D point onto the image plane via the sphere:

$$pr_{\gamma_2}(\boldsymbol{X}) = Kpr_{\xi}(\boldsymbol{X}) \quad (9)$$

Where, $\gamma_2 = \{f_x,\ f_y,\ u_0,\ v_0,\ \xi\}$ are the intrinsic camera parameters in the case of spherical model estimated by the calibration of the camera and we set $\gamma_2 = \{f_x,\ f_y,\ u_0,\ v_0,\ 0\}$ in the case of the virtual camera.

2.2.2 Homography on the sphere

In this section, we will develop the linear system of equations to estimate the homography which is a 3×3 projective transformation matrix defined up to scale factor to map one image to another.

Given a set of four 3D points on the sphere $X_s^i \in \mathbb{S}^2$ and their corresponding set $X_s^{i'} \in \mathbb{S}^2$, we will estimate the homography that maps $\mathbf{X}_s'$ to $\mathbf{X}_s$:

$$\mathbf{X}_s^{i'} \propto H\mathbf{X}_s^i \quad (10)$$

Where, $\mathbf{X}_s^{i'} = (x_i'\ y_i'\ z_i')^t$ and $\mathbf{X}_s^i = (x_i\ y_i\ z_i)^t$.

We can then write this equation as the cross product of both terms:

$$\mathbf{X}_s'^i \times H.\mathbf{X}_s^i = \begin{bmatrix} x_i' \\ y_i' \\ z_i' \end{bmatrix} \times \begin{bmatrix} h_1^t\ \mathbf{X}_s^i \\ h_2^t\ \mathbf{X}_s^i \\ h_3^t\ \mathbf{X}_s^i \end{bmatrix} = 0 \quad (11)$$

Where h_j is the j-th row of H.

Developing $\mathbf{X}_s'^i$ with its coordinates, we get:

$$\mathbf{X}_s'^i \times H.\mathbf{X}_s^i = \begin{bmatrix} 0 & -z_j'\mathbf{X}_s^i & y_j'\mathbf{X}_s^i \\ z_j'\mathbf{X}_s^i & 0 & -x_j'\mathbf{X}_s^i \\ -y_j'\mathbf{X}_s^i & x_j'\mathbf{X}_s^i & 0 \end{bmatrix} \begin{bmatrix} h_1 \\ h_2 \\ h_3 \end{bmatrix} \quad (12)$$

If we omit the third equation in H and imposing the condition $h_{33} = 1$, where $\mathbf{h} = [\mathbf{h}_1\ \mathbf{h}_2\ \mathbf{h}_3]^t$ is a vector made up of the entries of H, which can be justified by the fact that the solution is determined up to scale, the form of this equation system can be written as:

$$A_i\ \mathbf{h} = \mathbf{b} \quad (13)$$

Where,
$$A_i = \begin{bmatrix} 0 & 0 & 0 & -x_i z_i' & -y_i z_i' & -z_i z_i' & x_i y_i' & x_i y_i' \\ x_i z_i' & y_i z_i' & z_i z_i' & 0 & 0 & 0 & x_i x_i' & y_i x_i' \end{bmatrix}$$
and
$$\mathbf{b} = \begin{bmatrix} -z_i y_i' \\ z_i x_i' \end{bmatrix}$$

Solving for h: Let H be the homography for points belonging to the sphere, K_r and K_v are the intrinsic parameters of the real and the virtual cameras respectively. For the virtual camera K_v, we choose u_0 and v_0 in such a way to get the desired dimension of the source image which is the canonical form of the marker.

The following steps summarize the procedure of homography estimation:
- Using the projection formula-8, we project the four corners of the marker into the sphere:

$$\boldsymbol{X'}_s = pr_{\xi}^{-1}(K_r^{-1}\mathbf{u}^t) \quad (14)$$

- The same projection, using the virtual camera ($K_v,\ \xi = 0$) this time, for the four corners of the canonical form (the source image):

$$\boldsymbol{X}_s = pr_{\xi=0}^{-1}(K_v^{-1}\mathbf{u}^t) \quad (15)$$

- Homography estimation by solving the system-13 using $\boldsymbol{X'}_s$ and $\boldsymbol{X}_s$.

This system of equations can be solved for h linearly using, for example, Gaussian elimination with optimal pivot element chosen.

2.2.3 Warping

In this section, we will apply the homography to the canonical form that is remapped to the destination image and extract the intensities (warping). Figure-1 shows an example of the source image of three different markers in their canonical form on the right and on the left the corresponding polygons on the real image (the destination image).

As explained previously, the homography is applied to the spherical coordinates. In our practical situation, those coordinates are the points inside the polygon formed by the corners of the marker.

The following steps summarize the procedure of warping:
- We project every point $\mathbf{u}$ of the source image into the sphere using the virtual camera, formula-15.

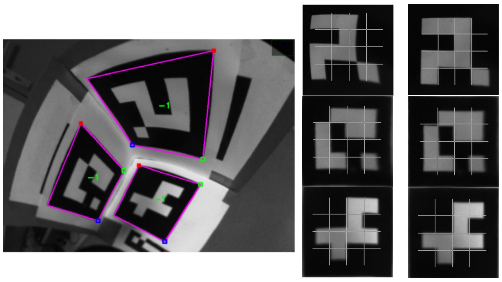


Figure 1. Warping markers taken with a catadioptric camera (on the left) and their canonical version on the right. The markers in the middle were warped by ArUco and the ones on the right were warped by our method. The grid is then applied to extract the binary code.

- Apply the homography, estimated previously by solving the system-13, to **Xs**:

$$X'_s = H.X_s \tag{16}$$

- Using the formula-9, we project $\mathbf{X}'_s$ into the image plane to get the intensity of the point $\mathbf{u}'$ in the source image which corresponds to $\mathbf{u}$.

Figure-1 illustrates a comparison between our method and ArUco. Solving the homography based on the sphere model allows to remove the distortions of the catadioptric camera, hence detect all the markers that ArUco based on conventional homography, failed to detect.

In the process of warping, many of the pixels are not integers, so we have to estimate those pixels in the destination image based on their neighborhood pixels. In this paper, we used the bilinear interpolation on the image plane.

2.2.4 Pose estimation

In this article, we use the pose estimation method from Ameller [1] which gives a direct solution by linearizing a system of six polynomial equations for four points (24 polynomials in 24 monomials):

$$\rho_i P_{jk}(\rho_j, \rho_k) = 0 \;/\; i, j < k = [1, 2, 3, 4] \tag{17}$$

Where,

- $P_{jk}(\rho_j, \rho_k) = \rho_j^2 + \rho_k^2 - 2cos(\theta_{jk})\rho_j\rho_k - d_{jk}^2$
- ρ_i: is the distance between the 3D points and the camera.
- d_{jk}: is the inter-point distance between the j-th and k-th object points.
- θ_{jk}: is the 3D angle formed by the j-th and k-th object points.

We then form a matrix of 24×24 from the system-17 which is then decomposed by SVD to get the null space monomial vector $\mathbf{N}_v$. The vector $\mathbf{N}_v$ allows to get the depth ρ_i, the rotation R and then the translation t, [9], between the camera and the object ${}^cM_o = \{R, t\}$. After that the pose ${}^cM_o$ is optimized using a first order optimization such as Levenberg-Marquardt (LM) to get the smallest reprojection error:

$$\min_{R,t} \sum_{i=1}^{4} \|pr_{\gamma_2}(R\mathbf{X}_i + t) - \mathbf{x}_i\|^2, \tag{18}$$

Where, $\mathbf{x}_i$ and $\mathbf{X}_i$ are 2D-3D correspondences.

3. Experiments

3.1. Protocol

In this section, we report experiments to quantify the effectiveness of our approach of adapting ArUco marker detection to panoramic sensors such as the catadioptric camera or fisheye cameras.

In view of the lack of a database of fiducial markers taken with those sensors, we recorded the *ArUcOmni dataset* of panoramic images (Fig. 2) of a 3D markers rig, put at various poses, with both catadioptric and fisheye sensors. The rig is made of three orthogonal faces of a cube, inside which ArUco markers are glued. By doing so, one can easily measure manually the transformation matrices between each pair of markers, mainly made of rotations of $90°$ in space and ruler-measured translations. Such transformations are considered as ground truth in the 3D space, free of any camera frame.

The ArUcOmni high-resolution dataset of 225 images is open-source[1] that was taken with a 2056×1542 pixels IDS UI-5280CP camera equipped with a VStone hypercatadioptric optics and a 1280×1024 pixels IDS UI-124xLE-C camera equipped with a Fujinon fisheye lens. This dataset is used in this evaluation to compare detection rates (Sec. 3.2) and pose estimation precision (Sec. 3.3), between our ArUcOmni approach and ArUco as well as its straightforward adaptation ArUco-Rectified.

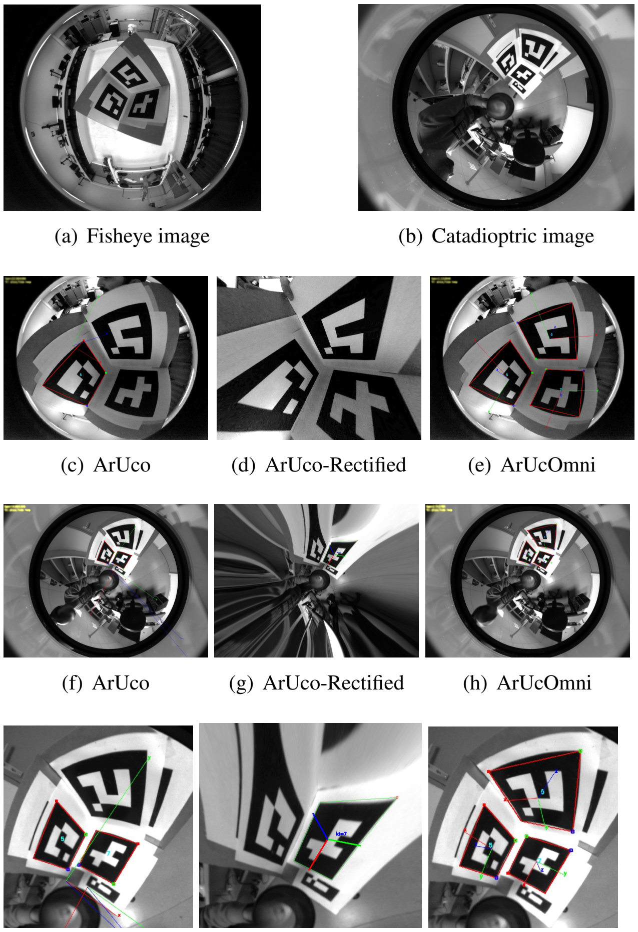
3.2. Comparison of detection rates

The first comparison is based on the rate of detection of the markers. Table-1 summarizes the results of the three methods on both catadioptric images (Omni) and fisheye images (Fisheye).

As it can be observed, our method (ArUcOmni) outperforms the comparison for catadioptric images with a score of nearly 100%. However, ArUco-Rectified performs better for fisheye images in some situations. Figure-3 shows one of those situations where marker#5 is detected.

ArUcOmni failed to detect the marker#5 when the marker plane is nearly 90 degrees angle with the camera

[1] Available for download at `http://mis.u-picardie.fr/~g-caron/pub/data/ArUcOmni_dataset.zip`.



(a) Fisheye image (b) Catadioptric image

(c) ArUco (d) ArUco-Rectified (e) ArUcOmni

(f) ArUco (g) ArUco-Rectified (h) ArUcOmni

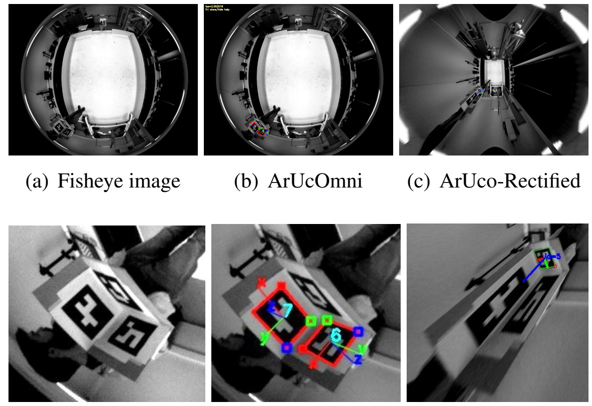
(i) ArUco (Zoom) (j) ArUco-Rectified (Zoom) (k) ArUcOmni (Zoom)

Figure 2. Examples from the dataset: Raw images (first row), marker detection using three different methods on both fisheye images (second row), catadioptric images (third row) and zoom into the markers in the catadioptric images (fourth row).

line of sight. That is because polygon extraction fails in the polygon approximation of the contours since based on straight lines which are not straight in the case of panoramic images. However, ArUco-Rectified rectified the marker in a way that it becomes nearly parallel to the image plane, thus the marker detection succeeded. The last observation concerns ArUco when it outperforms ArUco-Rectified in the

	ArUco		**ArUco-Rectified**		**ArUcOmni**	
Marker	Omni	Fisheye	Omni	Fisheye	Omni	Fisheye
5	90.48	10.89	21.69	**74.26**	**97.88**	61.39
6	53.44	40.59	08.99	73.27	**96.83**	**80.20**
7	72.49	40.59	54.50	77.23	**100.00**	**89.11**

Table 1. The rate of detection of markers (%). We have highlighted in bold the highest rates of detection.



(a) Fisheye image (b) ArUcOmni (c) ArUco-Rectified

(d) Fisheye image (Zoom) (e) ArUcOmni (Zoom) (f) ArUco-Rectified (Zoom)

Figure 3. An example where ArUcOmni failed to detect the marker#5. From left to right: the fisheye image, ArUco-Rectified and ArUcOmni (the first row) and the zoom into the markers (the second row).

case of catadioptric images. That could be explained by the fact that most of the rectified markers are either very distorted as in figure-2 (j) and/or broken because some points are rectified outside the image as in figure-2 (d).

3.3. Comparison of estimated poses precision

Once a marker is detected, its 4 corners allow its full 3D pose estimation in the camera frame. In order to ease the comparison of estimations between the two cameras, both calibrated with both Barreto's [3] projection model and Kannala's *et al.* [10] one, estimated 3D poses of markers in the camera frame are composed in order to get the estimated 3D rigid transformation between each pair of detected markers, for each image. Then, these marker-to-marker transformations are compared to the manually measured ground truth of the 3-face rig of markers. The Matlab script for computing the marker-to-marker estimated transformations and comparing them to the ground truth is publicly available, as well as the estimated poses of every detected marker of the ArUcOmni dataset. In Table-2 we give a summary of the estimation errors and the standard deviations from the dataset. We denote $\{R, t\}$ to be the orientation and the position of the absolute errors respectively.

Table-2 shows the estimation errors of ArUcOmni are much lower than ArUcoRectified both in terms of position and orientation. Estimations errors of ArUco are included in the table for illustration purposes only because they are obviously poor since ArUco does not take into account the geometry of panoramic images.

	Pose estimation error		
Methods	$\{R, t\}$	Omni	Fisheye
ArUco	R	19.2 (12.8)	27.8 (15.6)
	t	353 (189)	192 (181)
ArUco-Rectified	R	3.93 (1.35)	3.27 (2.98)
	t	58.6 (11.8)	29.5 (50.3)
ArUcOmni	R	**1.39 (0.72)**	**1.48 (1.9)**
	t	**7.57 (3.15)**	**9.62 (18.2)**

Table 2. Estimation of the absolute errors of marker-to-marker transformations in the 3-face rig. Units: position in mm, orientation in degrees. Values in parentheses are standard deviations. We have highlighted in bold the lowest errors.

4. Conclusion

We have proposed an adaptation of the algorithm of marker detection from ArUco for panoramic images that are based on the unified spherical model. First, the markers are detected by a remap to their canonical form using a spherical projection which removes distortions. After that, the pose estimation is also adapted and refined using a Levenberg-Marquardt optimization to minimize the reprojection error. The evaluations show that our method (ArUcOmni) outperforms the original algorithm (ArUco) and the fisheye rectification (ArUco-Rectified) both in terms of the rate of detection of the markers and pose precision. As a perspective, remains to be adapted the rest of ArUco's pipeline algorithm to fully benefit from the panoramic view, like the image processing part, polygon detection and approximation.

References

[1] Marc-André Ameller, Bill Triggs, and Lon Quan. Camera pose revisited–new linear algorithms. Inria research report.

[2] Joao Pedro Barreto and Helder Araujo. Issues on the geometry of central catadioptric image formation. In *IEEE Conference on Computer Vision and Pattern Recognition. CVPR 2001*, volume 2, 2001.

[3] João Pedro de Almeida Barreto. *General central projection systems: Modeling, calibration and visual servoing*. PhD thesis, 2004.

[4] A Berveglieri and AMG Tommaselli. Tree stem reconstruction using vertical fisheye images: A preliminary study. *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences-ISPRS Archives*, pages 627–632, 2016.

[5] Mark Fiala. Designing highly reliable fiducial markers. *IEEE Transactions on Pattern analysis and machine intelligence*, 32(7):1317–1324, 2009.

[6] Romeo Tatsambon Fomena, Han Ul Yoon, Andrea Cherubini, François Chaumette, and Seth Hutchinson. Coarsely calibrated visual servoing of a mobile robot using a catadioptric vision system. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 5432–5437.

[7] Sergio Garrido-Jurado, Rafael Muñoz-Salinas, Francisco José Madrid-Cuevas, and Manuel Jesús Marín-Jiménez. Automatic generation and detection of highly reliable fiducial markers under occlusion. *Pattern Recognition*, 47(6):2280–2292, 2014.

[8] Christopher Michael Geyer. *Catadioptric Projective Geometry: theory and applications*. University of Pennsylvania, 2002.

[9] Berthold KP Horn. Closed-form solution of absolute orientation using unit quaternions. *Josa a*, 4(4):629–642, 1987.

[10] Juho Kannala and Sami S Brandt. A generic camera model and calibration method for conventional, wide-angle, and fish-eye lenses. *IEEE transactions on pattern analysis and machine intelligence*, 28(8):1335–1340, 2006.

[11] Hirokazu Kato and Mark Billinghurst. Marker tracking and hmd calibration for a video-based augmented reality conferencing system. In *IEEE and ACM International Workshop on Augmented Reality (IWAR'99)*, pages 85–94, 1999.

[12] Hyungtae Kim, Eunjung Chae, Gwanghyun Jo, and Joonki Paik. Fisheye lens-based surveillance camera for wide field-of-view monitoring. In *IEEE International Conference on Consumer Electronics (ICCE)*, pages 505–506, 2015.

[13] Romain Marie, Hela Ben Said, Joanny Stéphant, and Ouiddad Labbani-Igbida. Visual servoing on the generalized voronoi diagram using an omnidirectional camera. *Journal of Intelligent & Robotic Systems*, 94(3-4):793–804, 2019.

[14] Gian Luca Mariottini and Domenico Prattichizzo. Image-based visual servoing with central catadioptric cameras. *The International Journal of Robotics Research*, 27(1):41–56, 2008.

[15] Christopher Mei, Selim Benhimane, Ezio Malis, and Patrick Rives. Homography-based tracking for central catadioptric cameras. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 669–674, 2006.

[16] Christopher Mei, Selim Benhimane, Ezio Malis, and Patrick Rives. Efficient homography-based tracking and 3-d reconstruction for single-viewpoint sensors. *IEEE Transactions on Robotics*, 24(6):1352–1364, 2008.

[17] Yoshihiko Mochizuki, Atsushi Imiya, and Akihiko Torii. Circle-marker detection method for omnidirectional images and its application to robot positioning. In *IEEE 11th International Conference on Computer Vision*, pages 1–8, 2007.

[18] Julien Moreau, Sébastien Ambellouis, and Yassine Ruichek. 3d reconstruction of urban environments based on fisheye stereovision. In *International Conference on Signal Image Technology and Internet Based Systems*, pages 36–41, 2012.

[19] Yoshio Onoe, Naokazu Yokoya, Kazumasa Yamazawa, and Haruo Takemura. Visual surveillance and monitoring system using an omnidirectional video camera. In *IEEE International Conference on Pattern Recognition*, volume 1, pages 588–592, 1998.

[20] Omar Tahri, Youcef Mezouar, François Chaumette, and Peter Corke. Decoupled image-based visual servoing for cameras obeying the unified projection model. *IEEE Transactions on Robotics*, 26(4):684–697, 2010.

[21] John Wang and Edwin Olson. Apriltag 2: Efficient and robust fiducial detection. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4193–4198, 2016.

[22] Xianghua Ying and Zhanyi Hu. Can we consider central catadioptric cameras and fisheye cameras within a unified imaging model. In *Computer Vision - ECCV*, pages 442–455, Berlin, Heidelberg, 2004.

[23] Zichao Zhang, Henri Rebecq, Christian Forster, and Davide Scaramuzza. Benefit of large field-of-view cameras for visual odometry. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 801–808, 2016.