Sphere mapping for feature extraction from 360° fish-eye captures

Fatma Hawary, Thomas Maugey, Christine Guillemot

SPHERE MAPPING FOR FEATURE EXTRACTION FROM 360° FISH-EYE CAPTURES

Fatma Hawary, Thomas Maugey and Christine Guillemot

Inria, Rennes Bretagne-Atlantique

ABSTRACT

Equirectangular projection is commonly used to map 360° captures into planar representation, so that existent processing methods can be directly applied to such content. Such format introduces stitching distortions that could impact the efficiency of further processing such as camera pose estimation, 3D point localization and depth estimation. Indeed, even if some algorithms, mainly feature descriptors, tend to remap the projected images into a sphere, important radial distortions remain existent in the processed data. In this paper, we propose to adapt the spherical model to the geometry of the 360° fish-eye camera, and avoid the stitching process. We consider the angular coordinates of feature points on the sphere for evaluation. We assess the precision of different operations such as camera rotation angle estimation and 3D point depth calculation on spherical camera images. Experimental results show that the proposed fish-eye adapted sphere mapping allows more stability in angle estimation, as well as in 3D point localization, compared to the one on projected and stitched contents.

***Index Terms*—** Spherical image, fish-eye camera, 360° , omnidirectional capture, feature extraction.

1. INTRODUCTION

Omnidirectional cameras have gained popularity as part of the rising interest for Virtual Reality (VR) and Augmented Reality (AR) technologies. They capture content that can also be useful for a variety of other applications, ranging from autonomous mobile robots to video-based security applications. On the consumer level, 360° contents are made widely available today on media sharing platforms such as Youtube. However, the use of 360° content, *e.g.* in computer vision applications, poses new challenging problems. Indeed, techniques designed for planar images have already been applied to omnidirectional images, either by mapping them to panoramic images (via Equirectangular Projection (ERP) [1]), or on a cylinder [2], or on cube surfaces [3]. However, these methods are not geometrically accurate since deformations are introduced by the omnidirectional sensor, and are not handled through these projections. Even if the algorithms can be locally valid, larger regions of the images may exhibit large distortions that could affect their performance. Two main reasons may cause such distortions: the geometry characteristics of the capture system, and the stitching introduced when producing the planar representation image. In practice, most rendered omnidirectional images result from a stitching process used to merge the overlapping field of view from the different sensors and hence produce a wide-view image. Indeed, when combining and warping the semi-spherical images of semi-spherical sensors, stitching introduces artifacts such as blurring, misaligned edges and object deformation [4].

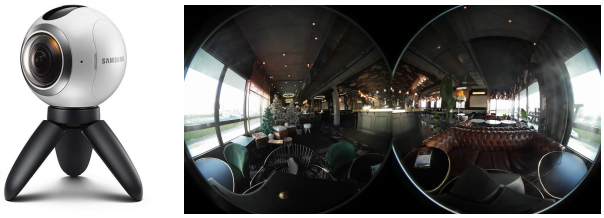

Fig. 1: (Left) Samsung Gear 360 Camera. (Right) Gear 360 dual-fisheye output image (3840×1920 pixels).

Besides, an important operation, when searching for matches between images in computer vision, consists in computing interest points also known as keypoints. Keypoints are intended to capture discriminative image features in a way that should be robust to changes, such as noise, perspective transformations, lighting changes, etc. A large variety of keypoint detection strategies exist in the literature, such as Affine normalization-based detectors [5, 6], edge-based region detectors [7], intensity extrema-based region detectors [7] and salient region detectors [8]. All these different techniques have been developed to work with planar images, but with the increasing interest in 360° content, several efforts have been made to design such techniques especially to deal with omnidirectional images [9, 10]. An important contribution in this context was the work of Geyer and Daniilidis [11] where the authors show that most catadioptric sensor images can be bijectively mapped onto the surface of a unit sphere. This also holds for cameras with fish-eye lenses in practice [12]. Thus, algorithms that treat spherical images can be applied to process a large variety of 360° images.

Spherical feature extractors have been proposed, either

This work has been supported by the EU H2020 Research and Innovation Programme under grant agreement No 694122 (ERC advanced grant CLIM).

using Harris corner detector [10], or by detecting and describing keypoints based on the spherical harmonics [13] referred to as Spherical SIFT. Qin and Li [14] use a combination of the Harris corner and planar SIFT descriptors on a bisection-based geodesic division of the sphere. A more recent work of Zhao *et al.* [15] proposes a spherical binary feature detector and descriptor, named SPHORB, where the mapping of the image on the sphere relies on a geodesic grid (an equal-area hexagonal grid [16]). The FAST detector [17] and an ORB-like descriptor [18] are used in the spherical 6-neighborhood system. This SPHORB algorithm is robust against camera rotation and noise, and has shown better performance compared to other existing spherical descriptors mentioned previously. Guan and Smith [19] use the same geodesic grid to map a panoramic image onto the sphere and extend the BRISK feature [20] to operate on spherical images.

Projecting equirectangular images onto the sphere helps to handle mapping distortions. However, stitching errors remain. These errors are likely to reduce the precision of feature detection methods, and further point localization and depth estimation algorithms. This triggers one question: can we trust feature extraction from a distorted image? Said differently, if a keypoint has been detected in a given direction, will it correspond to the exact angle, or will there be an angular distortion due to stitching?

In this paper, we propose to overcome stitching-related distortions, by avoiding equirectangular projection and directly building the spherical signal from the 360° images captured with multi-fisheye cameras. This is based on the Unified Spherical Model [21] and uses the intrinsic parameters of the cameras. A projection based on the geometry model of the camera is applied in order to fit the sensor image (Fig. 1 right) on the sphere, on which a feature extraction can be efficiently conducted using a spherical detector and descriptor. We have implemented here our method in the context of the SPHORB [15] algorithm, but it can be compatible with any descriptor that can be applied on the sphere. We demonstrate that taking into account the fish-eye geometry in the SPHORB algorithm leads to a much better precision of the 3D point localization, compared to the case when the SPHORB uses the stitched equirectangular image. For that purpose, we propose two experiments, in which the feature points are used for camera pose estimation and triangulation-based depth estimation. Experimental results show that our approach both leads to more accurate pose and depth estimation. Throughout this paper, we refer to the raw fish-eye camera images as FEC (Fig. 1, right), and to the stitched images resulting from equirectangular projections as ERP.

2. FEATURE DETECTOR AND DESCRIPTOR ON THE SPHERE

In this section, we describe an algorithm for spherical feature extraction, namely SPHORB, that allows the detection of interest points on the sphere. The main steps of the algorithm can be summarized as follows. It first builds the spherical image from the input image, and then detects keypoints using the FAST algorithm adapted on spherical coordinates, and constructs ORB descriptors for the selected keypoints.

2.1. Sphere Mapping

The SPHORB algorithm relies on a uniform sampling of the sphere, following the structure of a hexagonal geodesic grid [15, 16]. Among the geometric properties of this grid, we can mention the similarity in grid angles of all the cells at the same level of grid resolution, as well as the regular geodesic distance between cell centers, which allows considering local neighborhood as planar (and thus using directly planar distances between cells). This regularity makes it possible to apply the same binary test for detection on the whole sphere. Once the geodesic grid is constructed, the algorithm fills each sample point of the sphere with the color of the corresponding pixel in the equirectangular image by mapping its resolution to the resolution of the spherical grid.

2.2. Spherical FAST

The method applies the FAST detector that compares intensities between a central pixel p and its surrounding pixels (a neighborhood of 18 pixels in total). The pixel p is classified as a corner if there exist n_f pixels x that satisfy the following equation:

$$|I(x) - I(p)| > t, \tag{1}$$

where $I(x)$ corresponds to the intensity of the neighboring pixel x and $I(p)$ of the pixel p, t is a threshold.

To ensure scale invariance, the image is sampled, stored and processed on the sphere in a multi-scale manner, following a pyramidal structure of spheres at different sampling resolutions. The scale pyramid of the spherical image is built, by subdividing the icosahedron at different levels. The spherical FAST features are then produced at each scale level, and their union provides the detected features of the spherical image.

2.3. Spherical rBRIEF

For each detected corner pixel p, a descriptor is constructed from the intensity comparisons with pixels contained in its neighborhood patch of a predefined radius:

$$\tau(p; x, y) = \begin{cases} 1, & if I(x) < I(y) \\ 0, & otherwise, \end{cases} \tag{2}$$

where x, y are the pixel neighbors of p, and τ the intensity comparison.

3. FISH-EYE ADAPTED SPHERICAL MAPPING

The SPHORB algorithm, like most state-of-the-art feature extraction algorithms, takes as input equirectangular (ERP) im-

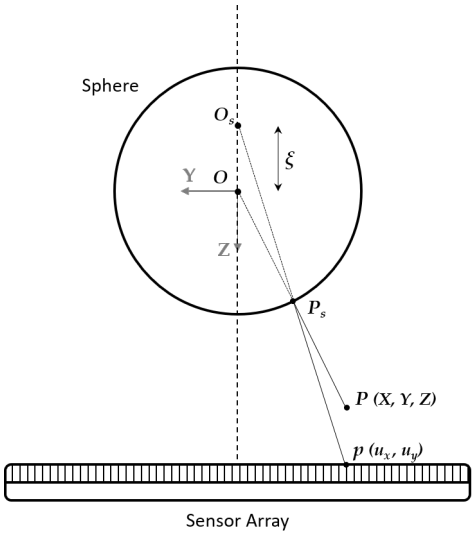

Fig. 2: The Unified Spherical Model.

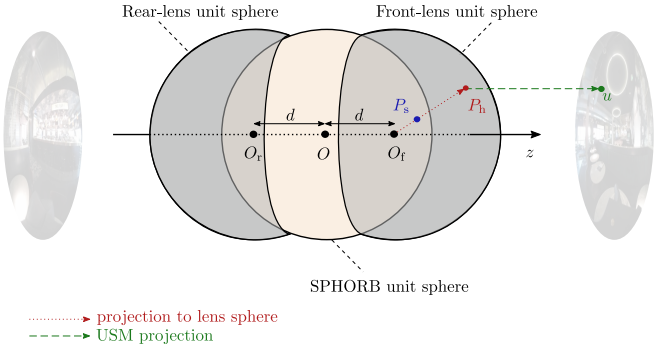

Fig. 3: Illustration of the fish-eye geometry-based sphere projection: first, each point P_s of the principal unit sphere is projected onto the corresponding lens sphere in point P_h. Second, P_h is mapped to the corresponding pixel u in the semi-spherical sensor image.

ages that suffer from stitching artifacts. The ERP format consist of projecting the spherical signal onto a plane by sampling it on an equirectangular grid, where the longitude and latitude of each sample are used as coordinates projected on the plane. In the case of spherical multi-sensor cameras, stitching errors are introduced when merging the different sensor images and mapping them to one equirectangular image. We consider here dual-fisheye lens cameras, and propose to adapt spherical descriptors to be defined directly on the recorded 360° data, hence not impacted by stitching. For this purpose, we map the sensor image (Fig. 1 right) directly onto a unit sphere, by taking into account the intrinsic parameters of the camera. Our method is detailed in the following.

3.1. Dual-Fisheye Camera Geometry

Dual-fisheye lens cameras have been increasingly used for 360° content acquisition. Unlike traditional perspective cameras, a 360° camera captures the entire viewing sphere surrounding its optical center, providing an omnidirectional field of view. However, a dual-fisheye camera uses two fish-eye lenses (front and rear) with a field of view close to 195° each (see Fig. 1). The limited overlapping field of views and misalignment between the two lenses give rise to visible discontinuities in the stitched image boundaries (see Fig. 4). One direct consequence of such distortions would be errors in the orientation estimation of the detected keypoints in images of the captured scene.

3.2. Unified Spherical Model

The extraction of information from any captured image always relies on its geometrical representation. While perspective representations are simple, the case of fish-eye camera images is more complex, due to non-linear distortions. In this context, Geyer and Daniilidis [21] demonstrated that every catadioptric projection is equivalent to a projective mapping from a sphere, centered in the single viewpoint, to a plane with the projection center placed on the perpendicular to the plane and distant of ξ from the center of the sphere (see Fig. 2). This is known as the Unified Spherical Model (USM). It consists of two main steps. It first projects a real-world 3D point P of coordinates (X, Y, Z) onto a point P_s on a unitary sphere centered in the optical center O (Eq.(3)). Secondly, it projects back this point P_s onto the camera sensor image plane via a perspective projection centered at a center O_s with a disparity ξ from O.

$$P_s = \frac{P}{||P||} \tag{3}$$

$$u = \begin{pmatrix} u_x \\ u_y \\ 1 \end{pmatrix} \equiv \mathbf{K} \cdot \left(P_s + \begin{pmatrix} 0 \\ 0 \\ \xi \end{pmatrix} \right) = \mathbf{K} \cdot \begin{pmatrix} \frac{X}{\sqrt{X+Y+Z}} \\ \frac{Y}{\sqrt{X+Y+Z}} \\ \frac{Z}{\sqrt{X+Y+Z}} + \xi \end{pmatrix} \tag{4}$$

The USM presents the advantages of being reversible, and uses only the intrinsic parameters of the camera to accurately and concisely represent it. The USM will be used in the following to map the information in 360° images onto a unit sphere on which the feature extraction is directly conducted.

3.3. Fish-eye Adapted Spherical Feature Extraction

We describe here the proposed feature extraction method operating directly on the sphere, adapted to fish-eye geometry. The main idea is to build the sphere by finding the corresponding pixel of each point in the image captured by the fish-eye camera and then to apply feature extraction algorithm directly on this sphere. In order to ensure uniformly distributed points on the sphere (with a similar neighborhood structure), the geodesic grid is used in [15] to choose the sampled sphere point positions. This set of points is selected by subdividing icosahedrons inscribed within the unit sphere into finer resolutions [16]. Once the geodesic grid is constructed (following

the same method as in [15]), we fill each spherical grid point with the color information of the corresponding pixel on the sensor image. Since the fish-eye camera sensor consists of two semi-spherical fish-eye lenses (front and rear) with distant optical centers, we consider these two hemispheres as two independent cameras, with a different reference system (see Fig. 3). Thus, we first project each point P_s of the principal unit sphere onto the corresponding lens (front or rear) unit sphere. Here, for simplicity, we differentiate points corresponding to the two hemispherical lenses with the sign of its coordinate Z_s in direction z. Equation 5 illustrates this projection:

$$P_h = \frac{P_s}{||P_s||} = \begin{pmatrix} \frac{X_s}{\sqrt{X_s+Y_s+(Z_s \pm d)}} \\ \frac{Y_s}{\sqrt{X_s+Y_s+(Z_s \pm d)}} \\ \frac{Z_s \pm d}{\sqrt{X_s+Y_s+(Z_s \pm d)}} \end{pmatrix}, \quad (5)$$

with d being the distance between the optical center of each lens to the center of the camera. Secondly, we map the hemispherical lens point P_h into the sensor image to find the corresponding pixel position u (in green in Fig. 3), following the Unified Spherical Model (USM). Equation (4) of the USM becomes then:

$$u = \begin{pmatrix} u_x \\ u_y \\ 1 \end{pmatrix} \equiv \mathbf{K} \cdot \left(P_h + \begin{pmatrix} 0 \\ 0 \\ \xi \end{pmatrix} \right) = \mathbf{K} \cdot \begin{pmatrix} \frac{X_h}{\sqrt{X_h+Y_h+Z_h}} \\ \frac{Y_h}{\sqrt{X_h+Y_h+Z_h}} \\ \frac{Z_h}{\sqrt{X_h+Y_h+Z_h}} + \xi \end{pmatrix} \quad (6)$$

Once all the sphere samples are mapped with the captured image, we apply the feature extraction method SPHORB to find the best keypoints.

Fig. 4: Example of a synthetic scene captured with a 360° fish-eye camera model: (top) original spherical image, (bottom) stitched equirectangular image.

4. EXPERIMENTS

In this section, we present the experiments that we conducted to evaluate the advantage of adapting the sphere construction to the fish-eye camera when extracting spherical features. We evaluate the scene points localization accuracy through two applications: a camera pose estimation task and triangulation-based depth estimation.

4.1. Camera pose estimation: rotation angle

Considering the fact that all points on a rotating sphere should have the same rotation speed (or the same rotation angle), we built an experimental protocol in which a fish-eye camera is fixed on top of a rotating device. With this set-up, we estimated the rotation angle of different scene points from the captured videos. We consider the angular coordinates of feature points on the sphere. We assess the stability of the corresponding angles when capturing the scene by a rotating fish-eye camera. For this purpose, the keypoints that have been detected and matched between equally-spaced frames of the captured video are used to track the angle variation. We apply the SPHORB algorithm on both fish-eye images (FEC) and their corresponding equirectangular projection images (ERP) obtained using the Samsung Gear 360 application. We estimated the angular rotation by following each detected point through different frames of the captured video and by calculating the displacement between 15 equally-distant frames. In order to assess the variation of the angles through the video, Fig. 5 illustrates the standard deviation (in degrees) of the estimated rotation angles calculated based on angular coordinates of all the detected keypoints at each frame, both for ERP and FEC images. Extracted frames from the captured real scene videos are also shown in Fig. 5. One can see that using the equirectangular projection produces less stable estimations of the camera rotation angle. An average inconsistency error of $1.24°$ results from angle estimation in ERP images, against a much lower error of about $0.74°$ in the case of FEC-adapted sphere mapping. Potential errors in the estimation of point direction in the scene can be introduced due to such high variations. As an example, in the case of 4K-resolution images, a direction estimation error of $1.24°$ corresponds to a displacement error of around 14 pixels. In the following, we propose a further study of the impact of such errors on 3D point localization from two 360° cameras.

4.2. Triangulation-based depth estimation

The algorithms used for vision-based localization usually rely on detecting and matching interest points between different views of the scene, or different frames of a video captured with a moving camera. They then estimate the depth of each point by triangulation. Taking the example of two 360° fish-eye cameras capturing the same scene at different positions, a localization algorithm would estimate the position of a 3D point by matching points detected on the two camera images, that describe the same point. Fig. 6 illustrates the case we consider in this study. Two fish-eye cameras placed at C_1 and C_2 and at a distance d from each other, capture a scene and a

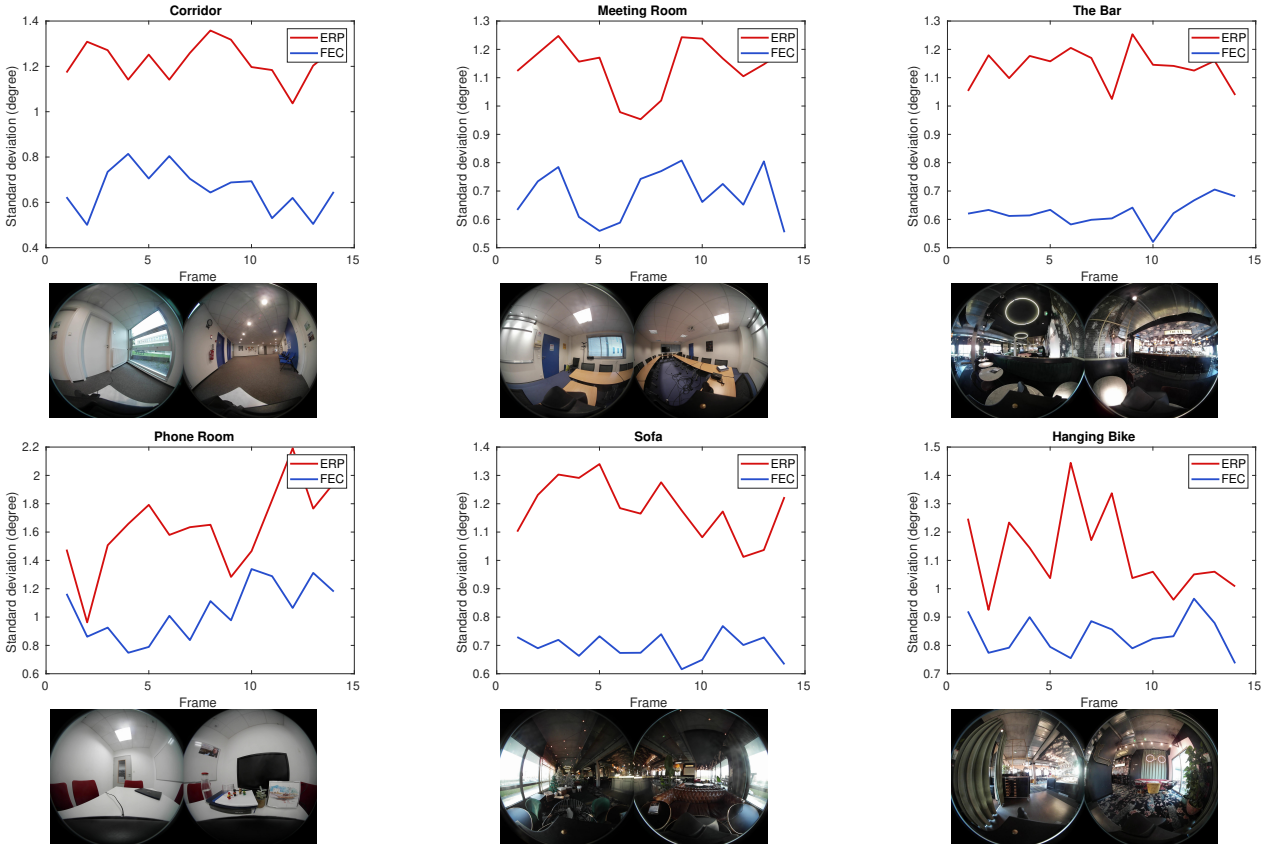

Fig. 5: Standard deviation of the estimated rotation angle for several scenes.

3D point P in the scene. Here, a planar representation is used for simplicity. The point P is detected through two points S_1 and S_2 on the spheres centered at C_1 and C_2 respectively. In a perfect scenario, the 3D point corresponding to two matching pixels is the intersection of the two light rays associated with these pixels. As errors may occur due to the acquisition, the calibration or the matching, we use the midpoint method [22] between the two rays defined by $\overrightarrow{C_1S_1}$ and $\overrightarrow{C_2S_2}$ to find the position of the 3D point. The objective is to find the 3D points W_1 and W_2 such as the distance $\overrightarrow{W_1W_2}$ is minimal, which corresponds to the line segment commonly perpendicular to the rays with a director vector defined as:

$$\overrightarrow{W_1W_2} = \alpha\,\overrightarrow{n}\ ,\ \overrightarrow{n} = \frac{\overrightarrow{C_1S_1} \wedge \overrightarrow{C_2S_2}}{\|\overrightarrow{C_1S_1} \wedge \overrightarrow{C_2S_2}\|} \tag{7}$$

We put two 360° fish-eye cameras in a synthetic scene in Blender to generate spherical images (Fig. 4, top). These images are then stitched into ERP images (Fig. 4, bottom). After applying SPHORB both on the raw FEC images and on the ERP images, we triangulate each pair of matched points to locate their corresponding 3D point. We evaluate the triangulation performance on the keypoints that are common in the two matching cases (ERP and FEC). We compare these results to the triangulation b resulting from an ideal pixel mapping (at the accurate directions) of the sphere obtained from Blender. The percentage of depth error is averaged on all the detected points, and calculated for different camera distances going from 1 to 20 cm. Table 1 illustrates the impact of the point direction error (*i.e.* pixel mapping error on the sphere) on the estimated position b' of the corresponding point P:

$$e = 100 \cdot \frac{\Delta b}{b} = 100 \cdot \frac{|b - b'|}{b} \tag{8}$$

One can see that the percentage of error in depth estimation is always lower in the case of fish-eye adapted sphere mapping of the images compared to stitching-based mapping. The results still present errors which are inherent to the fact that the two semi-spherical sensor images are mapped to one single unit sphere, thus including projection errors due to the distance between their two centers. This may be improved by directly using the unit sphere corresponding to each hemispherical lens.

5. CONCLUSION

We present in this paper a solution for sphere mapping that avoids applying stitching on 360° images captured with fish-eye cameras. In fact, projecting spherical images to planar representations such as equirectangular projection introduces

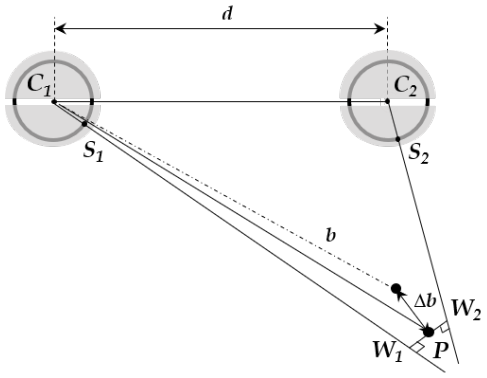

Fig. 6: Illustration of triangulation with two fish-eye cameras.

Table 1: Impact of spherical point projection error on 3D point localization (ERP *vs.* FEC)

$d\ (cm)$	1	5	10	15	20
$e_{ERP}(\%)$	21.90	13.26	14.12	9.58	3.20
$e_{FEC}(\%)$	17.30	7.18	10.37	6.00	2.16

geometrical distortions with implications on the accuracy of applications such as angle estimation, depth calculation and 3D scene reconstruction. We propose to adapt spherical models used by feature extraction methods to the geometry of the fish-eye camera. The captured 360° image is directly mapped onto a sphere using a uniformly-structured sampling grid. We evaluate the interest of the proposed fish-eye adapted sphere construction by assessing the stability of angle estimation within the detected feature points through videos captured by a rotating fish-eye camera. Additional experiments are conducted to assess the accuracy of depth estimation based on the triangulation of matched points on the sphere. Experimental results show the advantage of the proposed fish-eye adapted sphere mapping of the 360° images compared to mapping from stitched images.

6. REFERENCES

[1] J. Xiao, K. A. Ehinger, A. Oliva, and A. Torralba, "Recognizing Scene Viewpoint using Panoramic Place Representation," *CVPR*, 2012.

[2] G. Kweon and Y.-H. Choi, "Image-processing Based Panoramic Camera Employing Single Fisheye Lens," *Journal of the Optical Society of Korea*, vol. 14, no. 3, pp. 245–259, 2010.

[3] J. Huang, Z. Chen, D. Ceylan, and H. Jin, "6-DOF VR Videos with a Single 360-Camera," *IEEE Virtual Reality (VR)*, 2017.

[4] R. G. de A. Azevedo, N. Birkbeck, F. De Simone, I. Janatra, B. Adsumilli, and P. Frossard, "Visual Distortions in 360-degree Videos," *IEEE Transactions on Circuits and Systems for Video Technology*, 2019.

[5] K. Mikolajczyk and C. Schmid, "An affine invariant interest point detector," *ECCV*, pp. 128–142, 2002.

[6] K. Mikolajczyk and C. Schmid, "Scale and Affine Invariant Interest Point Detectors," *International Journal of Computer Vision*, pp. 63–86, 2004.

[7] T. Tuytelaars and L. Van Gool, "Matching Widely Separated Views Based on Affine Invariant Regions," *International Journal of Computer Vision*, vol. 60, no. 1, pp. 61–85, 2004.

[8] T. Kadir, A. Zisserman, and M. Brady, "An affine invariant salient region detector," *ECCV*, 2004.

[9] I. Bogdanova, X. Bresson, J. P. Thiran, and P. Vandergheynst, "Scale-space analysis and active contours for omnidirectional images," *IEEE TIP*, vol. 16, no. 7, pp. 1888–1901, 2007.

[10] H. Hadj-Abdelkader, E. Malis, and P. Rives, "Spherical image processing for accurate visual odometry with omnidirectional cameras," *OMNIVIS*, 2008.

[11] C. Geyer and K. Daniilidis, "Catadioptric projective geometry," *International Journal of Computer Vision*, vol. 45, no. 3, pp. 223–243, 2001.

[12] J. Courbon, Y. Mezouar, and P. Martinet, "Evaluation of the Unified Model of the Sphere for Fisheye Cameras in Robotic Applications," *Advanced Robotics*, vol. 26, no. 8, pp. 947–967, 2012.

[13] J. Cruz-Mota, I. Bogdanova, B. Paquier, M. Bierlaire, and J.-P. Thiran, "Scale Invariant Feature Transform on the Sphere: Theory and Applications," *International Journal of Computer Vision*, vol. 98, pp. 217–241, 2012.

[14] X. Qin and S. Li, "Finding scale-invariant corner feature from full-view image based on discrete spherical model," *ICSAI*, p. 1914–1918, 2012.

[15] Q. Zhao, W. Feng, L. Wan, and J. Zhang, "SPHORB: A Fast and Robust Binary Feature on the Sphere," *International Journal of Computer Vision*, vol. 113, pp. 143–159, 2015.

[16] D. A. Randall, T. D. Ringler, and R. P. Heikes, "Climate modeling with spherical geodesic grids," *Computing in Science Engineering*, vol. 4, no. 5, pp. 32–41, 2002.

[17] E. Rosten, R. Porter, and T. Drummond, "Faster and better: A machine learning approach to corner detection," *IEEE TPAMI*, vol. 32, no. 1, pp. 105–119, 2010.

[18] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, "ORB: An efficient alternative to sift or surf," *ICCV*, pp. 2564–2571, 2011.

[19] H. Guan and W. A. P. Smith, "BRISKS: Binary Features for Spherical Images on a Geodesic Grid," *CVPR*, 2017.

[20] S. Leutenegger, M. Chli, and R. Y. Siegwart, "BRISK: Binary Robust Invariant Scalable Keypoints," *ICCV*, pp. 2548–2555, 2011.

[21] C. Geyer and K. Daniilidis, "A Unifying Theory for Central Panoramic Systems and Practical Implications," *ECCV*, pp. 445–461, 2000.

[22] R. Boutteau, "Reconstruction tridimensionnelle de l´environnement d´un robot mobile à partir d´informations de vision omnidirectionnelle pour la préparation d'intervention," *PhD Thesis.*, 2010.