Visual Computing in 360°: Foundations, Challenges, and Applications

Thiago L. T. da Silveira *and* Cláudio R. Jung
Institute of Informatics
Federal University of Rio Grande do Sul
Porto Alegre, Brazil
{tltsilveira, crjung}@inf.ufrgs.br

***Abstract*—Omnidirectional media are becoming widespread with the increasing popularization of devices for capture and visualization. Unlike traditional pinhole-based images, omnidirectional images are defined on the surface of a sphere, present a full field of view, and store light intensities from a whole scene. In particular, applications exploring immersive augmented, mixed, and virtual reality experiences can strongly benefit from omnidirectional vision. Though omnidirectional images are defined on the spherical domain, they are commonly mapped to one or multiple planes. Those sphere-to-plane mappings generate distorted images, and, if directly applied, most traditional visual computing algorithms tend to present some quality degradation. This tutorial paper revises the spherical imaging model, common capture device types, and prominent representation formats. It also discusses the significant challenges of spherical visual computing and showcases the advances in three selected applications.**

***Index Terms*—360° images, spherical images, omnidirectional images, panoramas**

I. Introduction

Omnidirectional images (a.k.a. spherical, 360-degree, or panoramic images) are gaining popularity mainly because the devices involved in capture are becoming cheaper [1]. When visualized in head-mounted displays (HMDs), 360° media help provide immersive user experiences in augmented, mixed, and virtual reality (AR/MR/VR) applications [2].

Unlike regular pinhole-based images that are defined on the plane, omnidirectional imagery lie on the sphere surface [3], [4]. 360° images present a full field of view (FoV) and capture the whole scene information ($360° \times 180°$). Fig. 1 depicts two images taken by regular and spherical cameras placed at the same pose within a realistic 3D model[1]. Despite the benefits, traditional visual computing algorithms designed to work on the planar domain are not directly applicable to panoramas because of the topology discrepancy.

Panoramas are indeed defined on the spherical domain, but they are commonly represented in a (multi-)planar form [5]. Many sphere-to-plane functions can be used to generate the planar representation, but none is free of distortions [6], [7]. Even though a panorama is represented in the plane (like in "world map" format), the algorithm still needs to take into account the introduced deformations to be accurate in its task [8], [9].

[1]The *Classroom* model is available under CC0 license in https://www.blender.org.

(a) (b)

Fig. 1: Two captures of the same 3D model at the same pose. The first view is taken by a (a) narrow-FoV pinhole-based camera and the second one comes from a (b) 360° camera.

Compared to the traditional visual computing field, *spherical* visual computing is still embryonic so that few problems are addressed under this renewed optics. This tutorial paper sheds some light on how one may expect spherical visual computing to differ from traditional and what efforts can be employed to account for these discrepancies.

The rest of this tutorial paper is organized as follows. Section II revises the fundamentals of the spherical imaging model. Common image acquisition systems and widely adopted representation formats are detailed in Section III. Challenges of processing spherical images and possible solutions are discussed in Section IV. Selected applications that benefit from the full FoV of spherical images are considered throughout Section V. Finally, some final remarks are drawn in Section VI.

II. Fundamentals

A pinhole-based camera is modeled by central projection where a ray comes from a three-dimensional (3D) world point, passes through its center of projection, and reaches the image plane [10]. The particularities of the 3D-2D mapping, such as the scene coverage in the image, depend on the camera matrix which combines intrinsic and extrinsic parameters [10].

The spherical imaging model derives from central and spherical projections [11] and abstracts the camera itself as a space-localized and oriented *unit sphere* [12]. As this type of camera covers the full FoV, each non-occluded surrounding

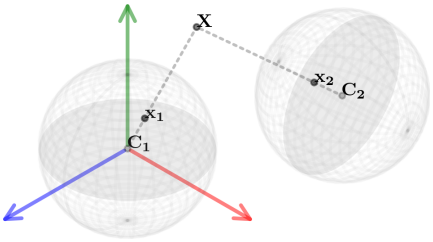


Fig. 2: Projection of a 3D world point $\mathbf{X}$ onto two spherical cameras with different extrinsic parameters.

3D world point is imaged using spherical projection [13]. This imaging model considers no intrinsic parameters and assumes that the spherical camera is fully represented by its six-degrees of freedom (6-DoF) extrinsics [14], [15].

More precisely, once a world coordinate system is preset, one can place the camera at position $\mathbf{C} \in \mathbb{R}^3$ and orientate it using a rotation matrix $\mathbf{R} \in SO(3)$. Thus, we explain the camera by its extrinsics $[\mathbf{R}|\mathbf{t}]$, where $\mathbf{t} = -\mathbf{RC} \in \mathbb{R}^3$ is called a "translation vector".

A 3D world point $\mathbf{X} \in \mathbb{R}^3$, parameterized by the same coordinate system, is then projected to that camera by

$$\mathbf{x} = \frac{\mathbf{RX} + \mathbf{t}}{\|\mathbf{RX} + \mathbf{t}\|_2}, \tag{1}$$

where the imaged point $\mathbf{x}$ lies on the surface of a unit sphere, i.e., $\mathbf{x} \in S^2 \subset \mathbb{R}^3$ [13].

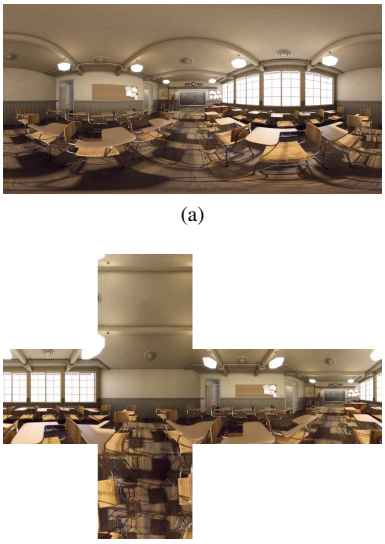
Fig. 2 depicts a 3D world point $\mathbf{X}$ projected onto two 360° cameras. One of the cameras is placed at the origin of and aligned to the world coordinate system, having extrinsics $[\mathbf{R}_1 = \mathbf{I}|\mathbf{t}_1 = -\mathbf{R}_1\mathbf{C}_1 = \mathbf{0}]$. The other camera is not at the origin and has a different orientation, presenting extrinsics $[\mathbf{R}_2|\mathbf{t}_2 = -\mathbf{R}_2\mathbf{C}_2 \neq \mathbf{0}]$. Note that the imaged points $\mathbf{x}_1$ and $\mathbf{x}_2$ are described in local image coordinates (w.r.t. each camera), having no explicit information about the original camera poses in the preset coordinate system.

III. Acquisition and Representation

The most common pipelines for acquiring omnidirectional images involve using one or more regular, planar silicon sensors [16]. In fact, differently from what the spherical imaging model suggests, there is no single-sensor device for capturing all the scene information at once [17].

Catadioptric imaging devices combine a regular camera with a convex-shaped (conic, spherical, parabolic, or hyperbolic) mirror, and allow capturing the whole horizontal FoV [18]. This approach, however, suffers from sensor/mirror self-occlusion and commonly outputs images represented in cylindrical form [8]. Since they have restricted vertical FoV and fragile mirror components, catadioptric devices are rare in recent research and industrial applications.

A polydioptric imaging system, on the other hand, organizes a variable number of regular cameras pointing outwards in a rig. Each camera captures a narrow portion of the scenario, and all the views are combined in a software-based procedure called image stitching (mosaicking) [19]. Polydioptric rigs are often bulky and expensive, but they can produce high-resolution panoramas with customizable FoV [20].



(a)

(b)

Fig. 3: Two planar representations of the same 360° image. The first view is in (a) equirectangular format and the second one is in (b) cube map format.

A more recent approach employed by many manufacturers combines two opposite located sensors equipped with fish-eye lenses [21]. Each sensor captures a hemispherical image suitable for two-view full-FoV stitching [22]. These portable and cheap devices simplified and democratized the acquisition of real-world 360° content and boosted the AR/MR/VR industry and research on related areas [23].

Regardless of the acquisition pipeline, if the camera matrices are known, one can warp the (single or multiple) image(s) onto the (potentially incompletely covered) unit sphere [13]. Still, various sphere-to-plane functions can be used for storing and processing 360° images that might be more convenient.

The equirectangular projection is considered the *de facto* planar representation of the sphere [12], [24]. It is also known as a latitude-longitude map [25] and allows easy pixel mapping from plane to sphere and vice-versa. As a given imaged point $\mathbf{x}$ lies on the surface of a unit sphere, it can be rewritten in spherical coordinates $(r = 1, \theta, \phi)$ as [13]

$$\mathbf{x} = [\cos\theta \sin\phi \;\; \sin\theta \sin\phi \;\; \cos\phi]^\top, \tag{2}$$

where $\theta \in [0, 2\pi)$ and $\phi \in [0, \pi)$.

Since an omnidirectional camera covers the full FoV, there is information associated to every position (θ, ϕ) on the sphere, and the image can be represented in a $[0, 2\pi) \times [0, \pi)$ plane. In fact, the light intensity associated to an imaged point $\mathbf{x}$ maps to the pixel position (x, y) of a $w \times h$ equirectangular image, where $x = \lfloor \frac{\theta w}{2\pi} \rfloor$ and $y = \lfloor \frac{\phi h}{\pi} \rfloor$.

The equirectangular projection has a non-uniform sampling that distorts the scene objects depending on their location in the image [8], being particularly heavy near the poles [26].

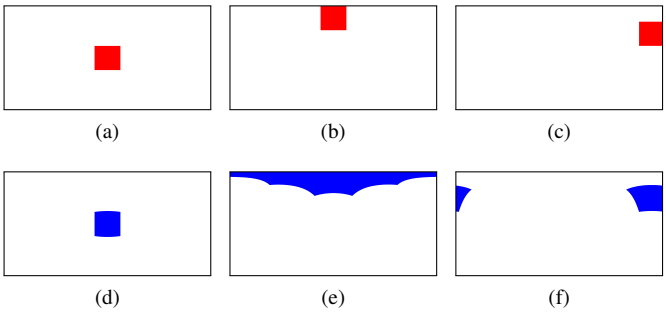


Fig. 4: Support of an ideal filter kernel in different positions of (a)–(c) pinhole-based and (d)–(f) equirectangular images.

Many other sphere-to-plane mappings can be applied, but they all introduce some distortion [6]. Since the distortions induced by spherical projection depend on the FoV amplitude [27], some authors propose to map the sphere onto a circumscribed cube. This process results in six narrower-FoV images and it is known as cube-map projection [27], [28] or sky-box [29]. Fig. 3 illustrates the mapping of the 360° image in Fig. 1b to the equirectangular and cube-map representations.

Other emerging representations are based on successive divisions of a 3D geometric form and try to mitigate the distortions from the spherical imaging model even more. Prominent approaches are the icosphere/tangent planes projection [30] that derive from an icosahedron and those based on an octahedron [31]. It is worth mentioning that exchanging formats may lead to loss of information and introduce artifacts [7] due to the required sub-pixel transformations [32].

IV. Challenges

Most challenges faced in spherical visual computing relate to the adopted sphere-to-plane mapping. Processing panoramas on the spherical domain avoids tackling topological issues but requires adaptations of algorithms originally devised for perspective images.

Let us recall that the equirectangular format is the standard planar representation of 360° images, widely employed in industry and research [6]. As briefly discussed in Section III, an equirectangular image is non-uniform sampled on the sphere [8], which translates to a "stretching effect" particularly prominent near the poles [26]. In fact, the poles are highly oversampled, i.e., the first and last image rows collapse to the north and south poles of the sphere, respectively. Hence, they replicate information in all columns. In general, the spacing between adjacent points along a row in the equirectangular format is proportional to $\sin\phi$ [33], leading to a strong imbalance between the equator line and the poles. Moreover, equirectangular images have a cyclical property [16], [31], i.e., its left and right boundaries should connect. We refer the reader to Fig. 3a for an illustration of the abovementioned issues.

The use of equirectangular images comes from their simplicity and because they contain all the scene information. Exploring the whole scene context through a single image with a rectangular domain is very appealing, especially in the deep learning era. We conduct the following discussion considering the deep learning context, but understanding the common issues faced in spherical visual computing can be applied to other computational solutions that extrapolate learning-based approaches.

The core idea behind a convolutional layer is that it contains filters with spatially-invariant support (receptive field size) and weights [34]. Standard convolutional kernels are rectangular or, more commonly, squared and are applied over the image in a sliding window fashion. Because of the distortions of the equirectangular projection, applying these regular filters to a panorama causes uneven (spherical) regions to be covered depending on the filter position. In fact, the support of the kernel filter should ideally be adjusted depending on the latitude ϕ of the image [6].

Figs. 4a and 4d show the ideal support of a hypothetical filter (larger than common for better visualization) on the middle of pinhole-based and equirectangular images, respectively. When the filter center is at the equator ($\phi = \frac{\pi}{2}$) of the spherical image, its support is barely distorted. Fig. 4e illustrates the adjustment required for the filter to cover the same area on the sphere surface as it approaches the north pole ($\phi \to 0$) of an equirectangular image. As we can observe, the shape of the filter is no longer rectangular, with wider support closer to the pole (top row). Fig. 4b, on the other hand, shows the effect of a standard convolutional kernel, which is fixed and covers a smaller spatial portion of the sphere compared to the equator line. Finally, Fig. 4c depicts what happens when a traditional filter touches a lateral border of a regular image. Often, the filter is simply applied after zero-padding or data extrapolation in a regular convolution. In the spherical case, an ideal filter should perform a circular convolution instead, as shown in Fig. 4f.

The circularity issue can be solved in a simple manner by applying a circular padding treatment, as done in [35]–[37]. The deformation of the kernel, on the other hand, is harder to handle. Some approaches propose to adjust the convolutions (and sometimes pooling operations) to tackle the geometry-induced distortions [6], [38]–[40]. Dilated convolutions are used in [38] so that their horizontal receptive field increases when approaching the image poles. Alternatively, the authors in [6] propose to learn weights that adjust the responses of a flat filter to accommodate to the equirectangular distortions. Distortion-aware convolutions, proposed in [39], adjust the receptive field to sample points within the ideal support as discussed early. A similar idea is explored in [40], where deformable convolutions are adjusted to the sampling induced by the equirectangular mapping.

It is worth mentioning that other sphere-to-plane mappings, like the cube-map planar representation, alleviate the distortions since they involve tangent plane projections with smaller FoVs. In fact, there is a context-deformation trade-off linked to the image FoV [27]. Dealing with any multi-plane representation may alleviate the problem with distortions but introduce

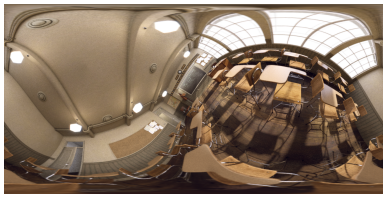


Fig. 5: Omnidirectional capture (in equirectangular format) that is not gravity aligned.

a less controllable discontinuity issue along the boundaries of each adjacent projection. The reader can look back to Fig. 3b and perceive how intricate the cube faces connections are. Dealing with cube-maps requires adequate treatment, such as face padding [41] or post-processing stitching [27].

Other issues that may impact applications related to the content rather than the sphere-to-plane mapping include imperfect image stitching after acquisition [7], highly variable object appearance [8], and bad horizon alignment [31], [40].

Finally, the availability of standardized benchmarks for comparing 360° visual computing techniques is still an open issue. For the first problem discussed in the next section, the authors often adopt general-purpose datasets (such as [42]) and assume the images are rectified. The nature of the problem allows generating an arbitrary number of annotations by synthesizing rotations, and quantitative results are often obtained using angular distances [23], [43], [44]. The recent survey from [16] compiles tens of datasets and figures of merit (omitted here due to space restrictions) for the other two applications discussed in the next section. Annotations and metrics are much more intricate for those two problems.

V. Selected Applications

Some works have started addressing visual computing problems using spherical images in the last few years. Omnidirectional vision boosts many applications, and we selected three of them that greatly explore the full FoV of panoramas: gravity alignment, layout estimation, and depth estimation.

A. Gravity Alignment

Gravity alignment (a.k.a. horizon alignment or upright adjustment) aims to realign the image content upright [43] so that the equator line of the panorama is parallel to the ground plane. Fig. 5 illustrates a capture of the same scene as in Fig. 1, but with a different camera orientation. Note that the ground plane presents a sinusoidal form. Recall that panoramas are defined on the sphere surface, and thus they naturally present three DoFs. This means that, aside from inaccuracies caused by sphere-to-plane reprojection [45], 360° images can be rotated to any orientation without loss of information.

Thus, the goal of an upright adjustment method is to estimate a rotation matrix $\mathbf{R}^\uparrow \in SO(3)$ that aligns the ground plane with the equator and stands up the scene objects. Note that using the equirectangular format allows exploring full contextual information. Adjusting the orientation of a panorama is accomplished by projecting the image to the unit sphere using the relation in Eq. (2), rotating the sphere by $\mathbf{R}^{\uparrow\top}$, and backprojecting the light intensities to the plane. For example, correcting the upright vector of the image in Fig. 5, which was captured with a tilted 360° camera, would ideally result in the image shown in Fig 3a.

Some works try to estimate $\mathbf{R}^\uparrow$ looking for geometric cues in the images, such as vanishing points and straight lines [46], [47]. These approaches tend to fail in nature scenes where those primitives are not clearly visible. More recently, some authors started tackling this problem from a learning-based perspective. Some of them propose to regress the Euler angles [23] or the upwards unit vector [43] that aligns the image, whereas others estimate rotation parameters from a discrete set of possible values [44], [48].

Although gravity alignment is essential for helping AR/MR/VR users get comfortable immersive experiences [23], it might also be helpful as an intermediate step for other applications. For example, several existing single-image layout or depth estimation approaches require upright rectified images as input to be accurate [43].

B. Layout Estimation

Layout (room) estimation aims to recover a scale-arbitrary, sparse 3D representation of the corners/joints between the walls/ceiling/floor of an indoor capture [16]. Early methods were semi-automatic or explicitly used geometric primitives to estimate the layout [49], [50]. Recent methods address layout estimation from a learning perspective and use a single panorama as input.

The choice over the representation of the input image varies, but the equirectangular format is by far the most used [16]. If an equirectangular image is gravity-aligned, the layout joints are parallel, and efficient 1D features in the latent space can be explored [35], [51]. Some methods opt for regressing the walls/ceiling/floor joints and others their edges [52].

Layout estimation methods often add geometric constraints to guide the optimization process [16]. The simplest yet realistic room layout is the cuboid/box-shaped layout [53]. More generic approaches follow the Manhattan assumption, which comprises more intricate room layouts such as "L-shaped" and others where the walls are perpendicular to each other [40], [54]. Finally, the Atlanta assumption allows even curved walls and approaches an unrestricted layout scenario [55].

Fig. 6 depicts the 3D representation of the layout inferred from the image in Fig. 3a using the approach from [54]. Note that, for this application, only the rough (box) layout is expected to be adequately projected to the 3D space.

C. Depth Estimation

Estimating depth from perspective images is a well-known problem in visual computing. Classical approaches use two or more overlapping views from the same scene but might require thousands of them to cover the full FoV [16]. Depth estimates can be either sparse by matching keypoints or dense by adopting approaches like optical flow to find correspondences. In the dense case, every pixel in a given reference image has

Fig. 6: Layout estimate from a single 360° image using [54].

Fig. 7: Depth estimate from a single 360° image using [57].

an associated depth value, and, except by occlusions, the scene can be fully 3D reconstructed.

From a geometrical point of view, only two 360° captures of the same scene are required for full-FoV depth estimation [36]. Stereo-based 3D reconstruction methods match N correspondences $(\mathbf{x}_1^i, \mathbf{x}_2^i), i = 1, 2, \ldots, N$, in both images to infer the relative pose by exploring the epipolar geometry [13] via

$$\mathbf{x}_2^{i\top} \mathbf{E} \mathbf{x}_1^i = 0, \tag{3}$$

where $\mathbf{E} = [\mathbf{t}]_\times \mathbf{R}$ is the Essential matrix and $[\cdot]_\times$ is the skew-symmetric matrix corresponding to the cross-product with $\mathbf{t}$ [12]. Then, the depth $\alpha^i \in \mathbb{R}$ associated to the points in a canonical view, e.g., $\alpha^i \mathbf{x_1}^i = \tilde{\mathbf{X}}^i \approx \mathbf{X}^i$, can be estimated via direct linear transformation (DLT) or similar approaches [14].

Multi-view-based 3D reconstruction adds robustness to outlier correspondences [36]. In this case, after the two-view reconstruction, the "Spherical n-Point" [14] problem is solved by matching estimated world points $\tilde{\mathbf{X}}^i$ to image points $\mathbf{x}_k^i$ of the k-th capture, $k \in \mathbb{N}, k > 2$, estimating the pose and the depth of novel points via a DLT-based calibrated reconstruction procedure [14], [56]. In stereo- and multi-view-based reconstructions, it is common to add non-linear refinement procedures for the pose, depth, and the pose and depth together [14], [36], [56].

More recently, many works started inferring dense depth from a single panorama, configuring it as an ill-posed but very appealing problem. Most methods either use equirectangular images [39], [57], which contain heavy distortions, or cube-map projections [27], which have face discontinuities. Some techniques adopt both representations and encode their mapping in the learning process to mitigate these problems [41], [58]. Other approaches consider more refined multi-plane representations and specialized network architectures [31].

Fig. 7 shows the point cloud associated with the depth map estimated from the image in Fig. 3a by the U-Net-based learning model in [57]. Note that each pixel has a depth value associated, unlike in the layout estimation problem.

It is worth mentioning that monocular depth estimation should work outdoors, unlike in the layout estimation problem. However, most methods may not generalize to these environments because of the lack of annotated datasets and might not handle infinity depth values, such as in the sky.

VI. Final Remarks

This tutorial paper aims to pave the way for the first contact of researchers in spherical visual computing. We review the spherical imaging model, the most common acquisition pipelines, and prominent representation formats. Then we discuss the significant challenges faced in the area and highlight recent advances in three applications that fully explore full-FoV images. We expect this paper to be a brief yet solid source addressing this renewed subject.

Acknowledgments

We thank the financial support from Fundação de Amparo à Pesquisa do Estado do Rio Grande do Sul (FAPERGS), Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq), and Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES) – Finance Code 001, Brazil.

References

[1] J. Huang, Z. Chen, D. Ceylan, and H. Jin, "6-DoF VR videos with a single 360-camera," in *IEEE Virtual Reality*, 2017, pp. 37–44.

[2] A. Serrano, I. Kim, Z. Chen, S. DIVerdi, D. Gutierrez, A. Hertzmann, and B. Masia, "Motion parallax for 360° RGBD video," *IEEE Transactions on Visualization and Computer Graphics*, vol. 25, no. 5, pp. 1817–1827, 2019.

[3] S. Li, "Binocular spherical stereo," *IEEE Transactions on Intelligent Transportation Systems*, vol. 9, no. 4, pp. 589–600, 2008.

[4] J. Fujiki, A. Torii, and S. Akaho, "Epipolar Geometry Via Rectification of Spherical Images," in *Computer Vision/Computer Graphics Collaboration Techniques*. Springer Berlin Heidelberg, 2007, vol. 4418, pp. 461–471.

[5] W. Yang, Y. Qian, J. K. Kamarainen, F. Cricri, and L. Fan, "Object Detection in Equirectangular Panorama," *Proceedings - International Conference on Pattern Recognition*, vol. 2018-August, pp. 2190–2195, 2018.

[6] Y.-C. Su and K. Grauman, "Learning Spherical Convolution for Fast Features from 360\textdegree Imagery," in *Conference on Neural Information Processing Systems*, 2017, pp. 529–539.

[7] R. G. d. 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*, vol. 30, no. 8, pp. 2524–2537, 2020.

[8] 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, no. 2, pp. 217–241, 2012.

[9] T. L. T. da Silveira, A. Q. de Oliveira, M. Walter, and C. R. Jung, "Fast and accurate superpixel algorithms for 360° images," *Signal Processing*, vol. 189, p. 108277, 2021.

[10] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*. Cambridge, 2003.

[11] S. Li and K. Fukumori, "Spherical stereo for the construction of immersive vr environment," in *IEEE Virtual Reality*, 2005, pp. 217–222.

[12] T. L. T. da Silveira and C. R. Jung, "Perturbation Analysis of the 8-Point Algorithm: A Case Study for Wide FoV Cameras," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 11 757–11 766.

[13] T. Akihiko, I. Atsushi, and N. Ohnishi, "Two-and three-view geometry for spherical cameras," *Workshop on Omnidirectional Vision, Camera Networks and Non-classical Cameras*, vol. 105, pp. 29–34, 2005.

[14] H. Guan and W. A. P. Smith, "Structure-From-Motion in Spherical Video Using the von Mises-Fisher Distribution," *IEEE Transactions on Image Processing*, vol. 26, no. 2, pp. 711–723, 2017.

[15] B. Krolla, M. Diebold, B. Goldlücke, and D. Stricker, "Spherical light fields," *British Machine Vision Conference*, no. 67.1-67.12, 2014.

[16] T. L. T. da Silveira, P. G. L. Pinto, J. Murrugarra-Llerena, and C. R. Jung, "3d scene geometry estimation from 360° imagery: A survey," *ACM Computing Surveys*, 2022, just Accepted.

[17] J. D. Adarve and R. Mahony, "Spherepix: A data structure for spherical image processing," *IEEE Robotics and Automation Letters*, vol. 2, no. 2, pp. 483–490, 2017.

[18] S. K. Nayar, "Catadioptric Omnidirectional Camera*," in *Conference on Computer Vision and Pattern Recognition*, 1997, pp. 482–488.

[19] S. Im, H. Ha, F. Rameau, H.-G. Jeon, G. Choe, and I. S. Kweon, "All-around depth from small motion with a spherical panoramic camera," in *European Conference on Computer Vision*, 2016, pp. 156–172.

[20] G. Fangi, R. Pierdicca, M. Sturari, and E. S. Malinverni, "Improving spherical photogrammetry using 360° OMNI-Cameras: Use cases and new applications," *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. 42, no. 2, pp. 331–337, 2018.

[21] Y. Shan and S. Li, "Descriptor Matching for a Discrete Spherical Image With a Convolutional Neural Network," *IEEE Access*, vol. 6, pp. 20 748–20 755, 2018.

[22] I. Lo, K. Shih, and H. H. Chen, "Image stitching for dual fisheye cameras," in *IEEE International Conference on Image Processing*, 2018, pp. 3164–3168.

[23] R. Jung, A. S. J. Lee, A. Ashtari, and J.-C. Bazin, "Deep360Up: A Deep Learning-Based Approach for Automatic VR Image Upright Adjustment," in *IEEE Conference on Virtual Reality and 3D User Interfaces*, 2019, pp. 1–8.

[24] M. Eder, P. Moulon, and L. Guan, "Pano Popups: Indoor 3D Reconstruction with a Plane-Aware Network," in *2019 International Conference on 3D Vision (3DV)*. IEEE, 2019, pp. 76–84.

[25] C. C. Gava, D. Stricker, and S. Yokota, "Dense Scene Reconstruction from Spherical Light Fields," in *IEEE International Conference on Image Processing*, 2018, pp. 4178–4182.

[26] L. S. Ferreira, L. Sacht, and L. Velho, "Local Moebius transformations applied to omnidirectional images," *Computers & Graphics*, vol. 68, pp. 77–83, 2017.

[27] T. L. T. da Silveira, L. P. Dalaqua, and C. R. Jung, "Indoor Depth Estimation from Single Spherical Images," in *IEEE International Conference on Image Processing*, 2018, pp. 2935–2939.

[28] F. Dai, C. Zhu, Y. Ma, J. Cao, Q. Zhao, and Y. Zhang, "Freely Explore the Scene with 360° Field of View," in *IEEE Conference on Virtual Reality and 3D User Interfaces*, 2019, pp. 888–889.

[29] S. Song, A. Zeng, A. X. Chang, M. Savva, S. Savarese, and T. Funkhouser, "Im2Pano3D: Extrapolating 360° Structure and Semantics Beyond the Field of View," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, vol. 1, 2018, pp. 3847–3856.

[30] M. Eder, M. Shvets, J. Lim, and J.-M. Frahm, "Tangent images for mitigating spherical distortion," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2020.

[31] Y. Lee, J. Jeong, J. Yun, W. Cho, and K.-J. Yoon, "Spherephd: Applying cnns on 360° images with non-euclidean spherical polyhedron representation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 1–1, 2020.

[32] B. Coors, A. P. Condurache, and A. Geiger, "SphereNet: Learning spherical representations for detection and classification in omnidirectional images," *European Conference on Computer Vision*, pp. 525–541, 2018.

[33] F. De Simone, P. Frossard, P. Wilkins, N. Birkbeck, and A. Kokaram, "Geometry-driven quantization for omnidirectional image coding," *2016 Picture Coding Symposium, PCS 2016*, 2017.

[34] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. The MIT Press, 2016.

[35] C. Sun, C.-W. Hsiao, M. Sun, and H.-T. Chen, "HorizonNet: Learning Room Layout with 1D Representation and Pano Stretch Data Augmentation," pp. 1047–1056, 2019.

[36] T. L. T. da Silveira and C. R. Jung, "Dense 3D Scene Reconstruction from Multiple Spherical Images for 3-DoF+ VR Applications," in *IEEE Conference on Virtual Reality and 3D User Interfaces*, 2019, pp. 9–18.

[37] N. Zioulis, F. Alvarez, D. Zarpalas, and P. Daras, "Single-shot cuboids: Geodesics-based end-to-end manhattan aligned layout estimation from spherical panoramas," p. 104160, 2021.

[38] N. Zioulis, A. Karakottas, D. Zarpalas, and P. Daras, "OmniDepth: Dense Depth Estimation for Indoors Spherical Panoramas," in *European Conference on Computer Vision*, 2018, pp. 453–471.

[39] K. Tateno, N. Navab, and F. Tombari, "Distortion-Aware Convolutional Filters for Dense Prediction in Panoramic Images," *European Conference on Computer Vision*, pp. 732–750, 2018.

[40] C. Fernandez-Labrador, J. M. Facil, A. Perez-Yus, C. Demonceaux, J. Civera, and J. Guerrero, "Corners for layout: End-to-end layout recovery from 360 images," *IEEE Robotics and Automation Letters*, pp. 1–1, 2020.

[41] F.-E. Wang, Y.-H. Yeh, M. Sun, W.-C. Chiu, and Y.-H. Tsai, "Bifuse: Monocular 360 depth estimation via bi-projection fusion," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[42] J. Xiao, K. A. E., A. Oliva, and A. Torralba, "Recognizing scene viewpoint using panoramic place representation," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2012, pp. 2695–2702.

[43] M. A. Bergmann, P. G. L. Pinto, T. L. T. da Silveira, and C. R. Jung, "Gravity alignment for single panorama depth inference," in *Conference on Graphics, Patterns and Images (SIBGRAPI)*. IEEE, 2021, pp. 1–8.

[44] R. Jung, S. Cho, and J. Kwon, "Upright adjustment with graph convolutional networks," in *IEEE ICIP*. IEEE, 2020, pp. 1058–1062.

[45] J. Murrugarra-Llerena, T. L. T. da Silveira, and C. R. Jung, "Pose estimation for two-view panoramas based on keypoint matching: A comparative study and critical analysis," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2022, pp. 5202–5211.

[46] J. Jung, B. Kim, J.-Y. Lee, B. Kim, and S. Lee, "Robust upright adjustment of 360 spherical panoramas," *The Visual Computer*, vol. 33, no. 6, pp. 737–747, 2017.

[47] L. Zhang, H. Lu, X. Hu, and R. Koch, "Vanishing point estimation and line classification in a manhattan world with a unifying camera model," *International Journal of Computer Vision*, vol. 117, no. 2, pp. 111–130, 2016.

[48] Y. Shan and S. Li, "Discrete spherical image representation for cnn-based inclination estimation," *IEEE Access*, vol. 8, pp. 2008–2022, 2019.

[49] H. Yang and H. Zhang, "Modeling room structure from indoor panorama," in *ACM SIGGRAPH International Conference on Virtual-Reality Continuum and Its Applications in Industry*, 2014, pp. 47–55.

[50] H. Jia and S. Li, "Estimating structure of indoor scene from a single full-view image," in *IEEE International Conference on Robotics and Automation*, 2015, pp. 4851–4858.

[51] C. Sun, M. Sun, and H.-T. Chen, "Hohonet: 360 indoor holistic understanding with latent horizontal features," pp. 2573–2582, 2021.

[52] G. Pintore, C. Mura, F. Ganovelli, L. Fuentes-Perez, R. Pajarola, and E. Gobbetti, "State-of-the-art in automatic 3d reconstruction of structured indoor environments," *Computer Graphics Forum*, vol. 39, no. 2, 2020.

[53] Y. Zhang, S. Song, P. Tan, and J. Xiao, "PanoContext: A whole-room 3D context model for panoramic scene understanding," in *European Conference on Computer Vision*, 2014.

[54] F.-E. Wang, Y.-H. Yeh, M. Sun, W.-C. Chiu, and Y.-H. Tsai, "LED2-Net: Monocular 360° layout estimation via differentiable depth rendering," pp. 12 956–12 965, 2021.

[55] G. Pintore, M. Agus, and E. Gobbetti, "AtlantaNet: Inferring the 3D indoor layout from a single 360 image beyond the Manhattan world assumption," in *European Conference on Computer Vision*, 2020.

[56] A. Pagani and D. Stricker, "Structure from Motion using full spherical panoramic cameras," in *IEEE International Conference on Computer Vision Workshops*, 2011, pp. 375–382.

[57] G. Albanis, N. Zioulis, P. Drakoulis, V. Gkitsas, V. Sterzentsenko, F. Alvarez, D. Zarpalas, and P. Daras, "Pano3d: A holistic benchmark and a solid baseline for 360° depth estimation," in *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2021, pp. 3722–3732.

[58] H. Jiang, Z. Sheng, S. Zhu, Z. Dong, and R. Huang, "Unifuse: Unidirectional fusion for 360° panorama depth estimation," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 1519–1526, 2021.