Pose Estimation for Two-View Panoramas based on Keypoint Matching: a Comparative Study and Critical Analysis

Jeffri Murrugarra-Llerena Thiago L. T. da Silveira Claudio R. Jung
Institute of Informatics
Federal University of Rio Grande do Sul
{jeffri.mllerena, tltsilveira, crjung}@inf.ufrgs.br

Abstract

Pose estimation is a crucial problem in several computer vision and robotics applications. For the two-view scenario, the typical pipeline consists of finding point correspondences between the two views and using them to estimate the pose. However, most available keypoint extraction and matching methods were designed to work with perspective images and may fail under not-affine distortions present in wide-angle or omnidirectional media, which are becoming increasingly popular in recent years. This paper presents a comprehensive comparative analysis of different keypoint matching algorithms for panoramas coupled to different linear and non-linear approaches for pose estimation. As an additional contribution, we explore a recent approach for mitigating spherical distortions using tangent plane projections, which can be coupled with any planar descriptor, and allows the adaptation of recent learning-based methods. We evaluate the combination of keypoint matching and pose estimation methods using the rotation and translation error of the estimated pose in different scenarios (indoor and outdoor), and our results indicate that SPHORB and "tangent SIFT" are competitive algorithms. We also show that tangent plane adaptations frequently present competitive results, and some optimization steps consistently improve the performance in all methods. We provide code at https://github.com/Artcs1/Keypoints

1. Introduction

We are currently facing a growing popularization of devices for capturing and visualizing 360° (also called spherical, panoramic, or omnidirectional) media [21, 36]. The increase of available panoramic content – in particular related to immersive navigation in virtual, augmented, and mixed reality (VR/AR/MR) [9, 21] – motivates big companies to design 360°-based applications such as Facebook (360 photos), Google (Street View), and Snapchat (Local Lenses).

Omnidirectional images provide a full field of view (FoV) [10] and are defined on the surface of a unit sphere [7]. Most of the literature uses the standard spherical representation on the plane, called the equirectangular projection (ERP) [9, 13, 17]. However, any mapping to the plane will introduce distortions [36], so that techniques designed for perspective image might not perform as expected when applied to the ERP and other representations.

Pose estimation using panoramas is a crucial step in several applications such as 3D reconstruction [9], spherical video edition or stabilization [18, 37], and robotic navigation [25], to name a few. For the generic two-view problem using perspective images, the typical pipeline involves keypoint detection/matching and then estimating the pose based on the putative correspondences. The latter stage might involve the computation of epipolar matrices (fundamental and/or essential matrix), direct pose estimation using non-linear approaches such as bundle adjustment, or more often a combination of both [19]. The pipeline for panoramas is similar, but the main steps (keypoint detection/matching and pose estimation) must be adapted to account for the distortions in spherical images. On the other hand, degeneracy caused by the selection of co-planar points is unlikely to arise in full-FoV imagery [42].

The main goal of this paper is to perform a thorough comparison of existing keypoint detection and matching (designed for perspective images or tailored to panoramas) when combined with linear and non-linear pose estimation methods. In particular, we evaluate the quality of estimated poses regarding the different motion components individually (rotation and translation) and for scenarios with different characteristics (e.g., indoor vs. outdoor). We also explore the recent tangent-plane mapping approach [13] for adapting planar descriptors to the spherical domain, increasing the range of tested methods. Unlike other works that only evaluate a subset of approaches or use limited datasets [8, 13, 17, 43], we perform extensive experiments using several methods with a trade-off fair-optimal setup for each technique in each tested scenario.

2. Related Work

2.1. Keypoint Detection and Matching in Panoramas

Keypoint detection and matching for perspective images present a consolidated literature [29]. Until the last decade, hand-crafted methods such as SIFT [28], ORB [34], and BRISK [24] were widely used in several applications. However, applying them directly to panoramas tends to produce poor results due to strong distortions caused by planar mappings such as the ERP [41]. Cubemap projection involves six nearly planar projections with smaller FoVs, producing less distortions. However, the FoV of each face is larger than typical pinhole cameras, and the 90° connections of the faces can generate artifacts at the boundaries, prone to outliers [43]. As such, adaptations to work with spherical distortions have been proposed. As examples, we can cite approaches like SIFT on the sphere (SSIFT) [7], spherical ORB (SPHORB) [43], and spherical BRISK (BRISKS) [17], inspired on their planar counterparts but adjusting to the local geometry of the sphere.

It is worth mentioning the recent effort devoted to learning-based approaches using deep neural networks for sparse matching. Earlier methods such as LIFT [40] presented good results but with increased computational burden, and more recent methods such as Superpoint [12] present state-of-the-art accuracy with reduced cost. However, they might suffer from the generalization problem (i.e., dataset dependency) as generic deep-learning methods, which is particularly increased in the cross-domain experiments: they are trained on perspective images, and some lack of accuracy is expected when directly applied on panoramas.

As 360° media become popular, several authors present deep networks tailored to the spherical geometry. In particular, Eder *et al*. [13] use tangent planes concerning the barycenter points of a geodesic grid to mitigate distortions in other sphere-to-plane mappings, which can be explored to adapt planar keypoint descriptors to panoramas. Although an application with SIFT descriptor was shown in [13], a minimal evaluation was performed. The current paper tailors different traditional planar keypoint description and matching approaches [12, 28, 34] to work with panoramas using tangent plane projections [13]. Fig. 1 illustrates keypoints extracted using the Superpoint algorithm when applied to the ERP and tangent projections of the sphere. We perform a thorough comparison of these methods with and without the adaptation, including also approaches explicitly designed for panoramas [7, 43].

Note that a comparative study of keypoint descriptors for pose estimation in panoramas was presented in [8]. However, their dataset was limited (a single scenario was used, which might bias the analysis and conclusions), and they did not explore the tangent-plane projection approach nor consider learning-based keypoint matching methods.

2.2. Two-View Panorama Pose Estimation

The two-view pose estimation problem for perspective images consists of estimating the extrinsic parameters[1] (rotation matrix $R \in SO(3)$ and translation vector $\boldsymbol{t} \in \mathbb{R}^3$) of one camera relative to the other. A key concept in two-view geometry is the *epipolar* matrix, which relates corresponding points in two images [20]. In the uncalibrated camera setup, we seek for the *fundamental* matrix F, whereas the *essential* matrix E is estimated when the camera intrinsics are known. Estimating epipolar matrices requires the computation of correspondence points between the two views. The classical linear 8-Point Algorithm (8-PA) [27] is known for solving for the fundamental matrix. Some studies [20, 32, 35] found that data normalization, data centering, and non-linear optimization improve the robustness of the "vanilla" 8-PA. Essential matrices have only 5 degrees-of-freedom, allowing algorithms with fewer correspondence points to be developed, such as the classical non-linear 5-PA [31]. These "baseline" methods can also be coupled with outlier removal methods such as Random Sample Consensus (RANSAC) [15] or one of its many variants [2, 3, 33].

For two-view panoramas, the main results concerning epipolar matrices hold [32], recalling that the spherical model typically does not involve intrinsic parameters (so that the essential and fundamental matrices coincide) [1, 18]. However, specific schemes have been devised to cope with the non-uniform sampling of the ERP [18, 32]. Based on the theoretical results from [10], Solarte and colleagues [35] recently introduced the SK non-linear optimization to deal with outliers and to improve the stability of 8-PA (RobustPA) obtaining state-of-the-art results.

In the current study, we explore different strategies for pose estimation. We first use the classical methods, i.e., the 8-PA [20] and 5-PA [31], from which we estimate the (initial) pose parameters. We then explore the non-linear refinements from [32, 35]. All these pose estimation methods are coupled with different keypoint matching approaches and hence generate a wide range of combinations.

3. The Proposed Methodology

Our main goal is to evaluate different feature matching algorithms coupled with pose estimation algorithms in the context of panoramas. As noticed in [7, 17, 43], spherically-adapted methods present better results than their planar counterparts. However, such conclusion was based on limited datasets and using generic keypoint detection/matching

[1] The translation vector can be estimated up to a scalar factor when only two views are present. The 5 degrees-of-freedom translation vector is also referred as the translation direction.

metrics, which might not directly correlate with pose estimation. Hence, we also consider analyzing planar methods either by applying them directly to ERP images or by locally adapting them with tangent plane projections.

3.1. Spherical Keypoint Detection and Matching

Comparing different keypoint detection and matching approaches involves designing a fair setup, since each method provides descriptors with different characteristics that might require specific matching criteria. In this work, we use the recommended parameters and keypoint matching distance functions for each method. Unfortunately, we were not able to evaluate BRISKS [17] due to the absence of publicly available source code. Also, we do not evaluate SSIFT [7] because the official code provided by the authors is limited to square ERPs (which implies that the latitude angular sampling is not the same as the longitude), and their inferior results compared to BRISKS [17] and SPHORB [43]. We next briefly describe each approach used in our analysis.

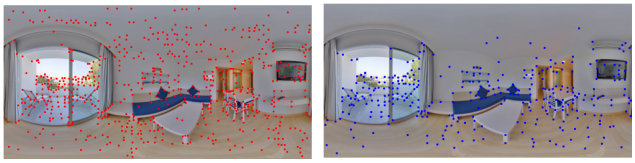
SIFT [28] is a hand-crafted planar algorithm. It consists in: (i) defining a multi-scale space to detect possible keypoints; (ii) identifying keypoints; (iii) assigning the orientation of the keypoints; (iv) computing the descriptor for each keypoint; and (iv) finding matches with the ℓ_2 norm.

ORB [34] is hand-crafted planar algorithm that combines modifications of the FAST detector and BRIEF [6] descriptor, achieving a fast and a multi-scale keypoint detector. ORB's pipeline consists in: (i) building a pyramid to produce multi-scale features; (ii) applying the oriented FAST detector with Harris score to sort them; (iii) computing the rBRIEF descriptor with the top keypoints; and (iv) matching features using the hamming distance.

Superpoint [12] is an end-to-end learning-based planar algorithm. It can be summarized as: (i) building a base detector with a synthetic dataset consisting of 2D triangles, rectangles, ellipses and lines; (ii) applying the base detector plus homography functions in the MS-COCO dataset [26] to get ground truth keypoint for complex images; (iii) training the Superpoint architecture with the ground truth; and (iv) matching features with ℓ_2 norm. In the original paper [12], Superpoint outperforms ORB [34], SIFT [28] and LIFT [40] in the repeatability, which is a general-purpose evaluation metric for keypoints [30].

SPHORB [43] is a hand-crafted spherical algorithm. It adapts planar ORB to the spherical domain by discretizing the unit sphere using a geodesic grid. SPHORB authors' claims more effectiveness than SSIFT [7] and results were comparable to BRISKS [17].

Tangent Plane Projection [13] is a "holistic" approach to map functions defined on the unit sphere to several tangent planes, aiming to mitigate local distortions. In particular, it can be used to adapt any planar descriptor to the spherical domain in a seamless manner by exploring the tangent planes. It was shown in [13] that "tangent SIFT" is a competitive descriptor for panoramas, and in this work, we extend the analysis for two other planar descriptors: the handcrafted-based approach ORB [34] and the learning-based approach Superpoint [12]. This strategy is better suited than cubemap representations because the number n of tangent planes can be controlled (opposed to the six planes in cubemaps), and hence the distortion tends to be smaller at each local projection as n increases.



(a) (b)

Figure 1. Superpoint keypoints extracted from images represented by (a) ERP and (b) tangent narrow-FoV projections.

As stated by Eder et al. [13], the tangent plane projection consists in projecting a spherical image to n bounded planes using multiple gnomonic projections. The number of projections is given by $n = 20 \times 4^b$, where b is the subdivision level of an icosphere. The tangent image dimension d is given by $d = 2^{s-b}$, where s relates to the sphere resolution, i.e., the number of subdivisions of the icosphere that results in more vertices than pixels in the ERP. Finally, the keypoint extraction step occurs on the narrow-FoV projections, with the keypoint positions mapped back to the sphere using the inverse-gnomonic projection and the corresponding descriptors kept unchanged. Here, we set $s = 8$ to deal with 512×1024 ERP images, and set $b = 1$ to have an acceptable trade-off between distortion-handling and computation time, which leads to $n = 80$, and $d = 128$.

3.2. Two-View Pose Estimation

The complete pose recovery problem involves the estimation of six parameters: three of them related to the rotation matrix R, and the 3D translation vector $\boldsymbol{t}$. For the particular case of two-view imaging, the full pose estimation problem is ambiguous, and we can only estimate the translation vector up to a scaling factor, leading to an unit vector characterized by its direction. Next, we briefly describe the two-view pose estimation methods used in our analysis.

Eight Point-Algorithm (8-PA) [20] is a linear algorithm that requires at least eight correspondence points to retrieve epipolar matrices. Its pipeline can be summarized as: (i) building a linear system $A_8\boldsymbol{e} = \mathbf{0}$, where A_8 is composed

by the correspondences points; (ii) obtaining $\boldsymbol{e}$ by singular value decomposition (SVD) and unwarping it to a 3×3 matrix E; (iii) enforcing rank-2 constraint to E with equal singular values; and (iv) extracting R and $\boldsymbol{t}$ from the epipolar matrix using Procrustes projection, as suggested in [18].

Five Point-Algorithm (5-PA) [31] is a non-linear algorithm that requires at least five correspondences points to estimate the essential matrix. It consists in: (i) building a system $A_5\boldsymbol{e} = 0$, where A_5 is composed by the correspondences points; (ii) computing four matrices X, Y, Z and W that span the null space of A_5, and then writing the essential matrix E as a linear combination of its basis elements; and (iii) calculate the scalars by finding the root of cubic constraints of e. The pose parameters R and $\boldsymbol{t}$ can be extracted from E as done in the 8-PA algorithm.

Non-linear least-squares optimization (NLR) [32] is refinement step. It receives the inliers produced by a pose recovery algorithm, as the 8- or 5-PA, and obtains the constant-weight least-squares solution. NLR refines the inliers in each iteration to pick the more "robust" points.

SK Non-linear optimization (SK) [35] is a precondition step recently proposed to improve the robustness of the 8-PA or 5-PA algorithms tailored to spherical images. The method consist in: (i) finding a pre-conditional diagonal matrix with entries $N = (S, S, K)$ that transforms the features from the unit sphere to an ovoid structure such that the epipolar error is minimized; (ii) transforming original keypoints according to N; (iii) applying the 8-PA to the transformed data and denormalizing based on N; and (iv) extracting the pose parameters as in the 8-PA algorithm.

In this work, we initially compute the epipolar matrix using either the 8-PA or the 5-PA as described above, supplied by outlier detection and removal via RANSAC [15] with proper spherical decision functions [32]. In some tests we couple this "baseline" method with SK non-linear optimization step [35] tailored to spherical images or a non-linear least squares optimization [32] or both (called NLRSK), resulting in a total of four combinations for each baseline.

3.3. Evaluation Metrics

Given the rotation matrix R and unit translation vector $\boldsymbol{t}$ obtained by keypoint detection/matching and pose estimation, we can define separate error measures for the individual pose components (rotation and translation). We compute the angular rotation matrix error [38] defined as

$$R_{error} = \cos^{-1}\left(\frac{\text{trace}(R^T R') - 1}{2}\right), \tag{1}$$

where R and R' are the ground truth and estimated rotations, respectively. Since the translation estimates are obtained up to a scale parameter, we compute the translation angular error [39] given by

$$t_{error} = \cos^{-1}\left(\boldsymbol{t} \cdot \boldsymbol{t}'\right), \tag{2}$$

where $\boldsymbol{t}$ and $\boldsymbol{t}'$ are the ground truth and estimated translation unit vectors.

We propose to compute the precision of the error estimates individually by defining an angular threshold T_θ. A given component of the pose (rotation or translation) is considered correct if its error is smaller than T_θ. It may be useful in applications where one can select a computationally lighter approach that might not be the most accurate, but matches at the desired threshold.

3.4. Evaluation Datasets

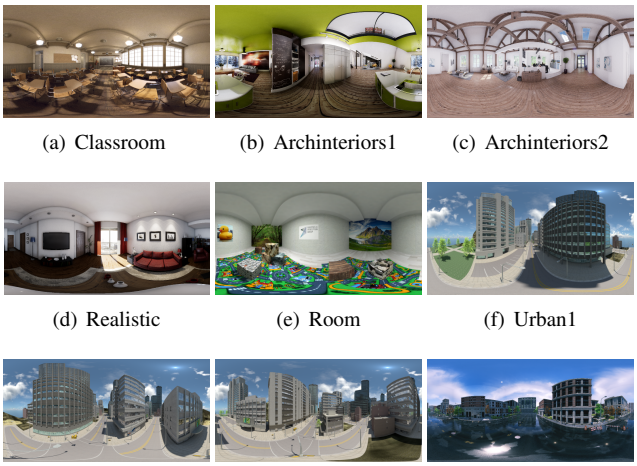
Comparing two-view pose estimation algorithms depends on choosing suitable datasets with two views of the same scene along with ground-truth poses. Despite the existence of publicly available datasets containing annotated poses and pairs of *perspective* images such as KITTI [16], we are not aware of annotated datasets with panoramas.

Due to data scarcity, we use open-source 3D modeling softwares such as Blender[2] and UnrealCV[3] to generate realistic synthetic panoramas from public 3D models (to manipulate UnrealCV scenes, we used the OmniSCV [4] package). Using synthetic data to train or evaluate models related to panoramas is a common practice [8, 22, 23, 32, 44], and extrapolating the conclusions drawn based on synthetic data to real-world images is highly dependent on the variability of the scenes. In our context, the variability strongly relates to textural information (for keypoint matching) and the the range of distances (pose estimation). Although we do not have a large set of 3D models to explicitly evaluate the impact of textural information, the use of indoor vs. outdoor datasets capture the expected difference of depth values.

For indoor scenes we used the Classroom[4], Room[5], RealisticRendering[6], Archinteriors1 and Archinteriors2[6] datasets. These indoors scenes (see Fig. 2, top row) present finite distances except in windows areas. We choose multiple canonical view for each scene, then perform random translations (within the scene 3D dimensions) and random rotation (real or synthetic rotation depending of the experiment) to generate pairs of panoramas with known relative poses. Finally, we manually delete images where an object overlaps most of the rendered scene (for instance, when the virtual camera is very close to an object).

For outdoor scenes, we use the Urban[7] (we selected three

[2] https://www.blender.org/
[3] https://unrealcv.org/
[4] https://download.blender.org/demo/test/classroom.zip
[5] http://rpg.ifi.uzh.ch/fov.html
[6] http://docs.unrealcv.org/en/latest/reference/model_zoo.html
[7] http://rpg.ifi.uzh.ch/fov.html



(a) Classroom (b) Archinteriors1 (c) Archinteriors2

(d) Realistic (e) Room (f) Urban1

(g) Urban2 (h) Urban3 (i) UrbanCity

Figure 2. Examples of 3D models in our dataset. In the top row, we present canonical images for indoors scenes. In the bottom row, we present canonical images for outdoor scenes.

subsets, named 1, 2, and 3) and UrbanCity[6] models. Unlike indoor scenes, they present objects farther away from the camera and regions with unbounded distances (e.g., the sky), as illustrated in the bottom row of Fig. 2. We select several canonical images and follow the methodology used for the indoor dataset for generating pairs of panoramas. The presence of large (or unbounded) depth values might be a challenge, since they might generate disparities smaller than a pixel and hence small matching errors can have a strong effect. As such, differences when analyzing indoor and outdoor scenes are expected.

4. Experimental Results

As described before, we assess the results produced by three planar (with and without adaptation with tangent planes) and one spherical keypoint algorithm. More precisely, we use OpenCV's implementation of SIFT [7] and ORB [34], a third-party implementation[8] of Superpoint [12] (the official implementation failed to retrieve matched features), named SPOINT in the experimental evaluation; and the official source code from SPHORB [43]. Additionally, we adapt SIFT, ORB and Superpoint for spherical images using the tangent plane procedure as described in Section 3.1, calling them TSIFT, TORB and TSPOINT hereafter. In all our tests, we keep the default parameters suggested in the papers, employing Lowe's ratio [28] and an adequate norm with a threshold $\gamma = 0.75$ for matching the descriptors and finally selecting the maximum keypoints produced for a given method (an average of obtained keypoints is presented in Table 1). Also, the next experiments introduce random translation (within a fixed range according to each scene) and a random rotation (where the three Euler angles are randomly chosen).

[8]https://github.com/rpautrat/SuperPoint

	ORB	TORB	SIFT	TSIFT	SPOINT	TSPOINT	SPHORB
Number of Keypoints	3893	1848	939	481	809	399	5034
Standard Deviation	94	60	34	36	16	12	120

Table 1. Average keypoints of each descriptors acroos our tested images.

4.1. Interpolate rotation vs. Realistic rotation

In theory, a spherical image allows arbitrary 3D rotations, meaning that a single panorama can be used to generate several rotated versions without the knowledge of the 3D scene (see [11] for more foundations). However, the ERP representation provides a non-uniform sampling of the sphere, so that performing rotations requires interpolation and re-sampling. Hence, this "interpolate rotation" (IR) process might generate artifacts that can affect keypoint detection and matching. On the other hand, since we have access to the 3D scene, we can perform the rotation on the virtual camera, generating a synthetic view that emulates the distortions of a real camera, and call this process "rendering rotation" (RR).

(a) Interpolate Rotation (b) Realistic Rotation

Figure 3. Interpolate Rotation vs Realistic Rotation

We performed a preliminary analysis using a single dataset (Room). For each canonical view, we generated two rotated versions using IR and RR with the same rotation matrix (see Figure 3), and estimate the pose rotation component of both panorama pairs. Table 2 shows the rotation accuracy for different angular acceptance thresholds ($T_\theta \in \{0.1°, 0.2°, 0.5°, 1°\}$) using the 8-PA RANSAC strategy. For low acceptance thresholds, IR presents better results in ORB, TORB, and SIFT, while RR outperforms in the remaining techniques. However, when the threshold is relaxed to $1°$, the accuracy gets close to 100% in both IR and RR for all tested keypoint algorithms. This preliminary experiment indicates that either IR or RR can be used, except if we use very restrictive angular thresholds ($< 1°$). It also shows that all tested approaches perform well for pure rotations. Hence, we incorporate translation operations to understand the limitations of the techniques as shown in [8]. Based on these results, we use IR in subsequent experiments, since it is much faster than RR.

threshold		ORB	TORB	SIFT	TSIFT	SPOINT	TSPOINT	SPHORB
0.1 °	RR	0.8	3.2	10.8	**40.8**	**14.8**	**14.8**	**100**
	IR	**10.8**	**100**	**13.6**	6.0	8.8	4.4	3.2
0.2 °	RR	6.4	22	44	**96.4**	**52**	**57.2**	**100**
	IR	**43.6**	**100**	**50.8**	28.4	46.0	16.0	20.4
0.5 °	RR	49.2	96.4	94.4	100	94.0	**97.6**	**100**
	IR	**97.2**	**100**	**97.2**	100	**97.2**	92.4	95.2
1 °	RR	98.8	100	99.2	100	100	100	100
	IR	**100**	100	**99.6**	100	100	100	100

Table 2. Accuracy (%) of rotation estimation using rendering rotation (RR) vs. interpolate rotation (IR).

4.2. Pure rotation vs. pure translation

In this experiment, we compare the pose accuracy produced by the keypoint methods under only rotational[9] or only translational movement. As noted by other authors [31, 38], the translation errors tend to be larger than rotation errors when perspective cameras are used, and this behavior was corroborated for spherical images in [10]. Hence, the acceptance thresholds for translation were more relaxed than those used in the rotation experiments.

Tables 3 and 4 show the individual accuracy for rotations and translations, respectively, using a set of 1,000 images across all datasets. The poses were estimated with the 8-PA RANSAC pipeline – when rotation estimation is required the translation component is ignored, and vice-versa. Overall, our results corroborate the findings of [10]: estimating translation is (much) harder than estimating the rotation. For the rotation-only experiment, we can see that the baseline 8-PA can achieve an accuracy close (or equal to) 100% regardless of the keypoint algorithm at an acceptance threshold $T_\theta = 1°$, which is already quite restrictive. TORB shows a consistent performance even for very low thresholds. As for the translation-only experiment, even for a relatively loose threshold $T_\theta = 20°$ the top accuracy did not reach 80%. We could not identify a clear pattern on which keypoint method was better. For the translation-only experiment, we identify TSIFT as the best method at $T_\theta \geq 5°$. However, for lower T_θ values, SPOINT and TSPOINT are better.

4.3. Combined Rotation and Translation

In this section, we evaluate the generic pose estimation problem that might contain both rotation and translation components. Moreover, we provide a separate analysis for indoor and outdoor datasets, since they present different overall characteristics (mainly related to the range of depth values present in the scene). We generated a total of 1,000 image pairs in each scenario, and evaluated the full-range of pose estimation methods: 8-PA and 5-PA, each of them either applied individually or combined with a refinement step (least squares optimization, SK non-linear optimization, and a combination of both).

[9]The validity of using the essential matrix for pure rotation estimation was recently justified in [5].

	Rotation threshold			
	0.1 °	0.2 °	0.5 °	1 °
ORB	15.2	55.7	95.8	99.4
TORB	**94.2**	**100**	**100**	100
SIFT	9.8	45.1	90.2	98.9
TSIFT	6.0	31.2	99.3	100
SPOINT	7.5	47.3	99.0	100
TSPOINT	3.5	17.6	94.1	100
SPHORB	3.8	20.6	99.7	100

Table 3. Accuracy (%) for different acceptance thresholds in a rotation-only experiment.

	Translation threshold				
	1 °	2 °	5 °	10 °	20 °
ORB	8.86	21.2	46.56	67.29	69.26
TORB	6.11	14.68	37.04	67.29	69.26
SIFT	5.71	16.59	41.14	61.54	70.62
TSIFT	7.51	22.10	**52.18**	**69.36**	**76.99**
SPOINT	**14.62**	25.55	47.87	60.04	68.19
TSPOINT	14.28	**27.7**	50.9	60.9	68.25
SPHORB	9.67	24.28	51.75	64.17	71.64

Table 4. Accuracy (%) for different acceptance thresholds in a translation-only experiment.

4.3.1 Indoor Scenes

Table 5 shows the accuracy achieved by all descriptors using all pose estimation methods for both translation and rotation errors. Although the accuracy values for the rotation component are still higher than the translation for similar thresholds (corroborating the results in [10]), they are considerable lower than those obtained when pure rotations were applied. Regardless of the chosen pose estimation method, the most competitive algorithms are SPHORB and TSIFT for both rotation and translation estimates (see the best and second best results highlighted in red and blue, respectively, in Table 5). We also note a considerable accuracy boost when comparing TSIFT with SIFT, meaning that tangent plane adaptation plays an important role for adjusting the local geometry of the sphere. This also happens when comparing TORB with ORB, but with a less noticeable improvement. We can also see that SPOINT did not produce good results, which could be expected since it was trained with perspective images and tested with ERP images. To our surprise, its tangent adaptation (TSPOINT) did not improve the results (on the contrary, they were actually worse). A possible explanation is that tangent images for indoor scenes are different enough from those used to train SPOINT, producing a domain shift.

		Rotation										Translation									
		1 °		2 °		5 °		10 °		20 °		1 °		2 °		5 °		10 °		20 °	
		5PA	8PA	5PA	8PA	5PA	8PA	5PA	8PA	5PA	8PA	5PA	8PA	5PA	8PA	5PA	8PA	5PA	8PA	5PA	8PA
ORB		20.6	20.6	43.2	38.28	67.7	56.51	81.7	73.35	92.5	89.38	13.4	10.42	25.8	23.25	51.3	46.29	70.3	61.92	83.9	76.95
	SK	21.0	21.8	43.4	38.48	68.1	57.31	82.3	73.95	93.3	88.78	13.0	10.62	26.2	23.05	52.5	46.6	70.1	67.9	84.1	79.3
	NLR	32.8	32.6	51.9	49.7	70.7	63.7	82.5	76.7	92.5	87.9	16.8	12.0	31.8	28.2	53.7	50.5	70.5	65.5	84.9	75.7
	NLRSK	50.1	46.2	63.1	54.1	74.7	66.5	84.9	77.3	92.9	87.5	31.4	27.6	45.2	40.8	61.7	57.1	74.9	67.9	86.1	79.3
TORB		31.2	24.9	49.1	38.5	70.5	59.8	79.7	71.4	89.7	86.9	22.2	16.8	36.8	28.9	58.3	52.2	71.9	67.8	81.9	79.7
	SK	32.0	25.3	49.3	38.5	71.5	59.8	80.1	71.4	89.5	87.3	22.6	16.8	37.8	28.9	58.7	52.0	71.5	68.1	81.5	79.9
	NLR	44.6	37.8	60.5	50.5	74.7	65.3	81.7	77.5	90.1	87.1	24.4	21.8	42.6	33.6	61.5	52.0	72.3	68.1	83.7	78.3
	NLRSK	56.9	47.7	65.7	55.9	75.5	67.9	82.5	78.5	89.9	87.7	40.4	33.2	52.7	47.0	68.1	61.7	76.1	70.5	84.1	78.9
SIFT		21.8	19.4	45.0	37.4	67.3	58.7	81.1	71.9	89.3	86.3	13.6	13.6	29.2	26.8	57.5	47.4	71.3	61.9	84.3	73.5
	SK	23.6	20.0	46.4	38.2	68.5	59.3	82.3	72.1	89.3	86.5	14.2	13.4	29.8	26.6	57.5	48.1	71.7	61.8	84.5	73.9
	NLR	34.8	29.6	54.7	46.8	71.5	61.7	81.3	74.5	89.9	86.9	15.4	14.4	35.4	28.0	58.3	47.7	71.9	61.9	83.3	72.7
	NLRSK	50.1	43.49	60.1	52.1	72.9	65.1	81.5	75.9	89.7	86.3	32.8	28.2	45.2	40.2	62.3	54.9	73.7	62.7	84.5	74.5
TSIFT		33.8	31.6	56.9	51.9	78.1	74.3	85.7	82.5	94.3	90.7	22.2	19.4	42.0	41.8	69.1	65.7	82.1	79.9	89.9	86.5
	SK	35.0	31.86	57.1	51.9	78.3	74.7	86.3	82.9	94.5	90.7	22.0	19.2	43.0	42.2	69.1	65.9	81.7	80.3	89.7	86.5
	NLR	46.4	41.8	65.5	60.1	81.9	77.3	89.5	84.3	95.1	91.9	28.2	24.8	51.1	48.1	72.9	70.9	84.1	80.9	**91.1**	87.1
	NLRSK	63.7	58.9	73.1	68.7	83.1	78.7	89.7	85.9	95.1	91.7	47.9	44.6	65.3	61.7	78.3	75.5	85.9	82.3	90.9	87.5
SPOINT		16.4	17.3	29.6	27.7	45.0	40.4	55.6	51.1	66.2	63.3	9.4	9.66	20.4	22.7	36.8	36.6	50.0	45.6	58.2	54.5
	SK	17.4	17.1	30.4	29.5	46.2	41.6	56.0	54.5	66.4	65.6	10.8	9.66	21.2	22.7	39.2	37.4	51.8	46.0	59.0	55.3
	NLR	27.2	27.1	41.4	37.1	54.0	47.6	59.2	54.5	66.6	65.5	15.0	12.9	29.2	29.2	47.4	43.2	56.8	50.9	61.4	58.8
	NLRSK	33.0	30.4	44.0	38.7	54.6	46.2	60.2	54.9	67.2	65.3	22.0	19.2	35.4	32.2	49.4	44.8	57.0	50.3	62.0	58.8
TSPOINT		15.8	12.8	28.0	21.0	38.2	38.0	45.6	48.1	52.8	56.1	6.20	7.4	18.0	15.3	34.6	32.2	42.2	40.0	48.2	50.3
	SK	16.6	12.8	28.6	22.4	39.0	38.4	47.2	48.1	54.0	56.9	6.8	7.6	19.2	15.2	34.8	32.2	42.8	40.2	48.8	50.5
	NLR	25.6	20.0	37.4	32.3	46.8	45.0	50.4	52.3	56.0	58.9	13.2	9.9	28.0	23.8	44.0	41.6	49.2	49.2	54.8	56.5
	NLRSK	29.4	26.0	38.6	34.9	47.0	45.66	50.4	52.5	56.2	58.3	18.0	15.9	29.6	29.0	45.0	41.8	49.6	49.4	54.6	55.5
SPHORB		37.2	33.2	57.7	47.2	76.7	68.4	88.7	80.6	93.1	90.6	22.0	20.8	46.4	36.6	68.7	65.0	80.1	76.4	89.7	86.0
	SK	38.0	33.8	58.3	48.8	76.7	68.6	88.9	81.0	94.7	91.2	21.8	21.0	47.0	36.8	68.7	65.6	80.7	76.4	89.7	86.2
	NLR	47.2	42.2	64.7	59.6	78.3	74.0	87.9	85.0	94.9	92.2	28.0	22.8	50.1	44.8	71.3	68.4	81.9	78.6	90.1	87.4
	NLRSK	62.9	58.6	71.7	67.2	80.3	77.2	89.1	85.6	**95.5**	92.6	41.4	35.0	64.3	57.0	75.7	71.6	3.5	79.4	90.7	88.0

Table 5. Pose estimation accuracy (in %) on *indoor* scenes with mixed translation and rotation based on the 5PA and 8PA algorithms

When comparing different pose estimation methods, we noted that the 5-PA method yields a slight improvement over the 8-PA in lower thresholds ($T_\theta < 5.0$). The SK optimization did not show considerable accuracy gain over the baseline methods, but NLR optimization and NLRSK optimization improve the performance of all descriptors, particularly when lower thresholds are selected.

4.3.2 Outdoor Scenes

Table 6 is similar to Table 5, but relating to the results in the outdoor scenes. Again, rotation results are better than translations ones for all angular thresholds, which also confirms that translation is more challenging even in a mixed setup for outdoor images. Both SPHORB and TSIFT are competitive algorithms for rotation accuracy (as in the experiment with indoor scenarios), but TSIFT outperforms SPHORB in translation by a reasonable margin (in fact, TSIFT seems to be in general better than SPHORB for translation).

As in the experiment with indoor scenes, different pose recovery approaches present similar behavior, with a consistent improvement when NLRSK is employed. Opposed to indoor results, however, the use of tangent plane projections boost all methods (including SPOINT). Finally, our results indicate that recovering pose in outdoor scenes seems to be more challenging than indoors, particularly for translation estimation using tighter thresholds. For example, the best translation accuracy using $T_t = 1°$ in indoor images is almost 48%, compared to less than 23% for outdoor scenes. We hypothesize that this behavior might be due to very large or infinite depth values in outdoor scenarios.

Overall, the difference of performance between indoor and outdoor datasets (see Tables 5 and 6) suggests that the performance of a descriptor can be biased to a scenario or dataset. On the other hand, improvements by refinement steps (NLR and NLRSK) are consistent.

4.4. Computational Cost

In our experiments, we use a laptop with an Intel i7 processor, 6 physical cores, 12 threads, 16GB DDR4 RAM, and an RTX3060 GPU card that boosts the performance of some methods, particularly SPOINT and the adaptations using tangent planes. An analytical comparison of the computational complexity is difficult, since some methods (e.g. SPOINT) can strongly benefit from specific hardware components such as GPUs. In this work, we present an analysis of the running times based on the original codes provided by the authors. Most descriptors are implemented in Python programming language; SPHORB's source code is provided in C++, but we add a Python wrapper to it. Table 7 shows the average time in seconds to process a single image using different descriptors (including keypoint extraction and matching), as well the running times for pose estimation using different strategies (ran on CPU only). We can observe that using GPUs did not impact planar handcrafted features (SIFT, ORB and SPHORB) in our tests, but GPU-tailored implementations can be used (such as [14]). We note that the tangent plane procedure incurs an overhead that depends on the base descriptor, and choosing weather

		Rotation										Translation									
		1 °		2 °		5 °		10 °		20 °		1 °		2 °		5 °		10 °		20 °	
		5PA	8PA	5PA	8PA	5PA	8PA	5PA	8PA	5PA	8PA	5PA	8PA	5PA	8PA	5PA	8PA	5PA	8PA	5PA	8PA
ORB		6.25	6.25	24.0	17.2	54.7	42.7	72.5	66.5	85.0	85.5	3.00	1.00	7.75	6.25	25.2	20.5	42.7	33.2	63.2	54.2
	SK	11.0	5.80	24.5	16.8	55.7	40.0	72.7	65.2	86.0	84.8	2.75	1.60	7.75	6.20	25.7	18.0	42.2	30.0	63.5	52.0
	NLR	10.7	8.50	27.5	20.7	55.2	43.0	71.7	63.2	85.0	83.0	2.50	2.25	8.00	7.00	26.5	23.2	42.5	33.5	62.5	56.5
	NLRSK	19.2	16.2	34.0	23.5	58.5	44.2	72.7	63.0	85.2	82.7	6.75	4.50	16.5	11.5	31.7	24.2	47.7	36.5	64.0	58.7
TORB		22.0	18.2	50.3	37.2	78.9	69.5	88.9	86.7	94.4	95.2	5.01	4.50	15.2	13.0	36.8	33.2	51.3	50.2	65.1	63.2
	SK	24.3	20.0	51.6	37.2	80.2	69.6	89.2	86.4	94.7	94.4	4.51	4.20	14.5	12.4	36.5	33.0	51.6	48.8	65.1	60.6
	NLR	31.0	24.5	55.1	48.2	81.7	75.7	90.4	88.2	94.4	95.5	5.51	5.75	15.0	13.7	37.5	36.0	51.6	50.7	65.6	64.2
	NLRSK	44.6	39.2	63.9	61.5	82.9	80.5	90.4	90.0	94.4	95.2	16.5	14.0	29.8	28.0	44.3	43.2	56.6	55.0	67.9	66.2
SIFT		13.4	7.00	35.4	23.7	67.0	55.0	81.2	77.7	92.1	92.2	4.30	2.00	16.9	9.50	35.7	27.7	54.6	48.5	72.1	69.0
	SK	14.4	6.20	36.9	23.2	68.6	53.0	82.2	77.4	92.9	91.8	5.32	2.80	18.9	9.20	37.4	26.8	55.9	46.2	72.1	66.0
	NLR	18.9	12.5	40.2	30.0	68.1	56.7	81.5	78.2	91.9	89.7	5.3	3.00	16.7	8.75	37.9	29.7	55.1	48.5	72.1	68.2
	NLRSK	30.8	24.0	50.8	38.7	70.6	56.7	82.7	77.0	91.9	90.2	12.9	9.2	25.3	17.7	43.5	36.0	58.9	53.5	73.1	68.0
TSIFT		29.9	20.4	63.4	50.7	89.2	81.3	94.8	93.4	96.6	97.4	9.46	8.59	21.2	25.0	56.5	53.2	76.7	72.4	84.4	84.0
	SK	30.6	20.1	64.1	48.7	89.5	80.2	94.6	93.2	96.4	97.3	10.2	8.23	21.9	23.6	56.0	51.8	76.7	70.5	84.6	82.7
	NLR	34.7	26.7	66.5	57.5	90.0	84.0	94.6	94.7	97.1	97.4	11.2	7.83	25.3	28.2	56.2	55.8	76.9	73.9	84.4	85.3
	NLRSK	56.5	46.2	75.4	69.1	91.5	86.3	94.3	95.2	97.9	97.7	22.7	25.5	46.8	43.9	69.3	64.9	78.2	78.2	85.1	**85.8**
SPOINT		27.5	22.5	43.5	37.2	61.0	53.7	68.0	66.0	74.2	73.5	6.25	5.00	17.0	11.5	27.2	22.0	36.2	29.0	42.5	36.7
	SK	28.0	24.0	43.7	38.7	60.7	55.5	68.7	65.5	75.2	73.5	6.25	5.25	16.2	12.2	27.5	23.5	36.7	30.0	42.7	36.7
	NLR	44.0	36.5	57.5	50.7	69.2	62.2	73.2	69.0	77.7	75.5	10.7	8.25	21.7	16.0	33.7	28.0	40.5	34.2	47.0	40.0
	NLRSK	47.7	40.2	60.0	53.5	69.5	63.0	74.2	69.2	78.2	75.0	15.2	12.7	26.5	22.0	36.5	30.7	43.0	36.2	49.2	41.2
TSPOINT		26.5	24.3	46.7	45.3	72.0	68.1	80.0	76.6	84.5	81.9	3.50	4.01	11.0	10.2	26.0	26.0	35.5	35.3	44.2	42.1
	SK	27.7	25.8	48.5	48.6	72.0	68.6	81.0	78.2	85.0	82.2	4.7	3.76	12.2	10.2	27.5	25.3	36.5	35.8	44.5	43.3
	NLR	46.0	42.8	67.2	64.6	78.7	77.1	85.2	82.2	86.7	84.4	6.50	6.77	16.7	16.7	33.5	33.0	42.5	40.6	48.7	48.1
	NLRSK	53.2	50.8	70.7	67.4	80.2	79.2	85.7	82.7	87.5	83.9	10.0	11.7	22.7	22.5	36.7	35.3	44.2	43.1	51.2	51.6
SPHORB		28.3	23.2	61.1	47.7	87.9	80.0	94.7	94.7	96.7	97.7	6.77	5.50	18.0	17.5	37.3	40.5	49.1	53.5	64.4	66.5
	SK	30.3	23.7	62.1	48.7	87.4	80.5	94.9	95.7	96.9	97.7	7.52	5.50	18.3	18.0	38.3	40.2	50.6	52.5	64.1	65.7
	NLR	36.3	31.5	64.9	59.5	88.9	86.0	94.4	95.5	96.7	98.0	7.27	8.00	19.5	19.7	37.5	42.7	50.6	56.2	64.1	68.0
	NLRSK	56.6	54.5	76.1	73.2	91.9	89.2	95.2	95.5	96.7	**98.2**	16.7	17.0	30.8	36.7	49.8	50.5	58.1	59.5	66.9	70.5

Table 6. Pose estimation accuracy (in %) on *outdoor* scenes with mixed translation and rotation based on the 5PA and 8PA algorithms

to use it or not must be based on an application-dependent cost-benefit analysis. For the particular case of SIFT, the overhead is around 30% in GPU but it yields considerable accuracy boosts particularly for low acceptance thresholds. Pose recovery running times are dominated by the 8-PA (OpenCV implementation), which depends on the number of matched points and outliers. Pre- and post-processing strategies add a very small overhead.

	ORB	TORB	SIFT	TSIFT	SPOINT	TSPOINT	SPHORB
Time (s) with GPU	0.54	1.00	1.03	1.35	1.21	5.23	1.29
Time (s) with CPU	0.54	1.17	1.03	1.46	2.45	10.33	1.29
Time (s) only 8PA	0.11	0.08	0.08	0.07	0.02	0.02	0.12
Time (s) only SK	0.12	0.08	0.08	0.07	0.02	0.02	0.13
Time (s) only NLR	0.12	0.08	0.08	0.07	0.02	0.02	0.13
Time (s) only NLRSK	0.13	0.09	0.09	0.08	0.02	0.02	0.14

Table 7. Average ruinng time (s) of keypoint extraction/matching and pose estimation (with maximum number of keypoints returned by each method, see Table 1.

5. Conclusions

In this paper, we presented a comparative analysis of seven keypoint matching algorithms applied to 360° image pairs using several pose estimation approaches. Three of them are popular planar methods, which are also adapted to the spherical domain using tangent projections. The other was specifically designed to work on the spherical domain.

We generated thousands of pairs of synthetic panoramas and assessed the methods under translation and rotation angular errors. We experimentally showed that estimating the translation is more challenging than rotation. We also noted that TSIFT and SPHORB attained the most competitive results, but their performance can be affected by the dataset. The running times for TSIFT are a little higher than those for SPHORB, but this overhead might be compensated by the accuracy increase for some scenarios (see Table 6). We also noted that the tangent planarization procedure often improves the accuracy over the baseline keypoint algorithm, but at a computation overhead.

Our tests involved two baseline pose estimation methods (8-PA and 5-PA) with or without coupling with pre- and post-processing steps. Our results indicated that the tested NLR optimization consistently improves the results in both indoor and outdoor scenarios, but SK optimization only produces better results when jointly with NLR.

6. Acknowledgments

We thank the financial support from Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq), and Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES) – Finance Code 001, Brazil.

References

[1] Torii Akihiko, Imiya Atsushi, and Naoya Ohnishi. Two-and three-view geometry for spherical cameras. *Proc. of the Sixth*

Workshop on Omnidirectional Vision, Camera Networks and Non- classical Cameras, 105:29–34, 2005. 2

[2] Daniel Barath and Jiří Matas. Graph-cut ransac. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018. 2

[3] Daniel Barath, Jiri Matas, and Jana Noskova. Magsac: Marginalizing sample consensus. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019. 2

[4] Bruno Berenguel-Baeta, Jesus Bermudez-Cameo, and Jose J. Guerrero. Omniscv: An omnidirectional synthetic image generator for computer vision. *Sensors*, 20(7), 2020. 4

[5] Qi Cai, Yuanxin Wu, Lilian Zhang, and Peike Zhang. Equivalent constraints for two-view geometry: Pose solution/pure rotation identification and 3d reconstruction. *International Journal of Computer Vision*, 127(2):163–180, 2019. 6

[6] Michael Calonder, Vincent Lepetit, Mustafa Ozuysal, Tomasz Trzcinski, Christoph Strecha, and Pascal Fua. Brief: Computing a local binary descriptor very fast. *IEEE transactions on pattern analysis and machine intelligence*, 34(7):1281–1298, 2011. 3

[7] J. Cruz-Mota, I. Bogdanova, B Paquier, M. Bierlaire, and J. P. Thiran. Scale invariant feature transform on the sphere: Theory and applications. *Intl. Jour. Comp. Vis.*, 98(2):217–241, 2012. 1, 2, 3, 5

[8] T. L. T. da Silveira and C. R. Jung. Evaluation of Keypoint Extraction and Matching for Pose Estimation Using Pairs of Spherical Images. In *Proceedings SIBGRAPI*, pages 374–381, 2017. 1, 2, 4, 5

[9] T. L. T. da Silveira and C. R. Jung. Dense 3D Scene Reconstruction from Multiple Spherical Images for 3-DoF+ VR Applications. In *Proceedings IEEE VR*, pages 9–18, 2019. 1

[10] T. L. T. da Silveira and C. R. Jung. Perturbation analysis of the 8-point algorithm: A case study for wide fov cameras. In *Proceedings CVPR*, pages 11757–11766, 2019. 1, 2, 6

[11] Thiago L. T. da Silveira, Paulo G. L. Pinto, Jeffri Murrugarra-Llerena, and Cláudio R. Jung. 3d scene geometry estimation from 360° imagery: A survey. New York, NY, USA, feb 2022. Association for Computing Machinery. Just Accepted. 5

[12] D. DeTone, T. Malisiewicz, and A. Rabinovich. Superpoint: Self-supervised interest point detection and description. In *Proceedings CVPR Workshop*, pages 337–33712, 2018. 2, 3, 5

[13] M. Eder, M. Shvets, J. Lim, and J. M. Frahm. Tangent images for mitigating spherical distortion. In *Proceedings CVPR*, pages 12423–12431, 2020. 1, 2, 3

[14] Hannes Fassold and Jakub Rosner. A real-time gpu implementation of the sift algorithm for large-scale video analysis tasks. In *Real-Time Image and Video Processing 2015*, volume 9400, page 940007. International Society for Optics and Photonics, 2015. 7

[15] M. A. Fischler and R. C. Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, 1981. 2, 4

[16] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 3354–3361, 2012. 4

[17] H. Guan and W. A. P. Smith. BRISKS: Binary Features for Spherical Images on a Geodesic Grid. In *Proceedings CVPR*, pages 4886–4894, 2017. 1, 2, 3

[18] H. Guan and W. A. P. Smith. Structure-From-Motion in Spherical Video Using the von Mises-Fisher Distribution. *IEEE Trans. on Image Processing*, 26(2):711–723, 2017. 1, 2, 4

[19] Richard Hartley and Andrew Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge, 2003. 1

[20] R.I. I Hartley. In defense of the eight-point algorithm. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(6):580–593, jun 1997. 2, 3

[21] S. Im, H. Ha, F. Rameau, H. Jeon, G. Choe, and I. S. Kweon. All-around depth from small motion with a spherical panoramic camera. In *Proceedings ECCV*, pages 156–172, 2016. 1

[22] Raehyuk Jung, Aiden Seung Joon Lee, Amirsaman Ashtari, and Jean-Charles Bazin. Deep360Up: A Deep Learning-Based Approach for Automatic VR Image Upright Adjustment. In *2019 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*, pages 1–8. IEEE, mar 2019. 4

[23] Po Kong Lai, Shuang Xie, Jochen Lang, and Robert Laqaruere. Real-Time Panoramic Depth Maps from Omnidirectional Stereo Images for 6 DoF Videos in Virtual Reality. *2019 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*, pages 405–412, 2019. 4

[24] S. Leutenegger, M. Chli, and R. Y. Siegwart. Brisk: Binary robust invariant scalable keypoints. In *Proceedings ICCV*, pages 2548–2555, 2011. 2

[25] J. Li, X. Wang, and S. Li. Spherical-model-based SLAM on full-view images for indoor environments. *Applied Sciences*, 8(11):2268, 2018. 1

[26] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft coco: Common objects in context. In David Fleet, Tomas Pajdla, Bernt Schiele, and Tinne Tuytelaars, editors, *Computer Vision – ECCV 2014*, pages 740–755, Cham, 2014. Springer International Publishing. 3

[27] H.C. Longuet-Higgins. A computer algorithm for reconstructing a scene from two projections. In *Proceedings Readings in Computer Vision*, pages 61 – 62. 1987. 2

[28] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *Intl. Jour. Comp. Vis.*, 60(2):91–110, 2004. 2, 3, 5

[29] Jiayi Ma, Xingyu Jiang, Aoxiang Fan, Junjun Jiang, and Junchi Yan. Image matching from handcrafted to deep features: A survey. *International Journal of Computer Vision*, 129(1):23–79, 2021. 2

[30] K. Mikolajczyk and C. Schmid. A performance evaluation of local descriptors. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(10):1615–1630, 2005. 3

[31] David Nistér. An efficient solution to the five-point relative pose problem. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(6):756–770, 2004. 2, 4, 6

[32] Alain Pagani and Didier Stricker. Structure from Motion using full spherical panoramic cameras. In *2011 IEEE International Conference on Computer Vision Workshops (ICCV Workshops)*, pages 375–382. IEEE, nov 2011. 2, 4

[33] Tianzhu Qiao and Huaping Liu. Improved least median of squares localization for non-line-of-sight mitigation. *IEEE Communications Letters*, 18(8):1451–1454, 2014. 2

[34] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski. Orb: An efficient alternative to sift or surf. In *Proceedings ICCV*, page 2564–2571, 2011. 2, 3, 5

[35] Bolivar Solarte, Chin-Hsuan Wu, Kuan-Wei Lu, Min Sun, Wei-Chen Chiu, and Yi-Hsuan Tsai. Robust 360-8pa: Redesigning the normalized 8-point algorithm for 360-fov images, 2021. 2, 4

[36] Y. Su and K. Grauman. Learning Spherical Convolution for Fast Features from 360° Imagery. In *Proceedings NeurIPS*, pages 529–539, 2017. 1

[37] Joanna Tarko, James Tompkin, and Christian Richardt. Real-time virtual object insertion for moving 360° videos. In *The 17th International Conference on Virtual-Reality Continuum and its Applications in Industry*. ACM, Nov. 2019. 1

[38] T.Y. Tian, C. Tomasi, and D.J. Heeger. Comparison of approaches to egomotion computation. In *Proceedings CVPR IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 315–320, 1996. 4, 6

[39] Jiaolong Yang, Hongdong li, and Yunde Jia. Optimal essential matrix estimation via inlier-set maximization. 09 2014. 4

[40] K. Yi, E. Trulls, V. Lepetit, and P. Fua. Lift: Learned invariant feature transform. In *Proceedings ECCV*, volume 9910, pages 467–483, 2016. 2, 3

[41] L. Zelnik-Manor, G. Peters, and P. Perona. Squaring the circle in panoramas. In *Proceedings ICCV*, volume 2, pages 1292–1299, 2005. 2

[42] Zichao Zhang, Henri Rebecq, Christian Forster, and Davide Scaramuzza. Benefit of large field-of-view cameras for visual odometry. *Proceedings - IEEE International Conference on Robotics and Automation*, 2016-June:801–808, 2016. 1

[43] Q. Zhao, W. Feng, L. Wan, and J. Zhang. SPHORB: A Fast and Robust Binary Feature on the Sphere. *Intl. Jour. Comp. Vis.*, 113(2):143–159, 2014. 1, 2, 3, 5

[44] Nikolaos Zioulis, Antonis Karakottas, Dimitrios Zarpalas, and Petros Daras. OmniDepth: Dense Depth Estimation for Indoors Spherical Panoramas. pages 453–471. 2018. 4