SphereGlue: Learning Keypoint Matching on High Resolution Spherical Images

Christiano Gava[1] Vishal Mukunda[2] Tewodros Habtegebrial[2] Federico Raue[1]
Sebastian Palacio[1] Andreas Dengel[1,2]
{christiano.gava,federico.raue,sebastian.palacio,andreas.dengel}@dfki.de
mukunda@rptu.de t_habtegeb15@cs.uni-kl.de
[1] DFKI [2] University of Kaiserslautern-Landau

Abstract

Traditionally, spherical *keypoint matching has been performed using greedy algorithms, such as Nearest Neighbors (NN) search. NN based algorithms often lead to erroneous or insufficient matches as they fail to leverage global keypoint neighborhood information. Inspired by a recent learned perspective matching approach [53] we introduce SphereGlue: a Graph Neural Network based feature matching for high-resolution spherical images. The proposed model naturally handles the severe distortions resulting from geometric transformations. Rigorous evaluations demonstrate the efficacy of SphereGlue in matching both learned and handcrafted keypoints, on synthetic and real high-resolution spherical images. Moreover, SphereGlue generalizes well to previously unseen real-world and synthetic scenes. Results on camera pose estimation show that SphereGlue can directly replace state-of-the-art matching algorithms, in downstream tasks.*

1. Introduction

Virtual and augmented reality applications, 3D reconstruction, autonomous driving and vision-based robot navigation systems require accurate camera poses. One way to obtain camera poses is through Structure from Motion (SfM), which depends on local feature correspondences. Many models have focused on local feature matching on perspective images [8, 53, 66, 68]. However, camera pose estimation can be more precise using spherical images because of their wide field of view. More formally, this assumption usually holds as there are more constraints determined by local features distributed over the surface of the (spherical) image.

Hence, robust feature detection and matching have to be developed for spherical images. First, traditional (handcraft) techniques exist for spherical keypoint detection [16, 28, 69]. Also, learn-based feature detectors for perspective images [22, 57, 60] can be mapped onto the sphere using local planar approximations [24]. Second, spherical keypoint matching is mainly based on greedy algorithms like Nearest Neighbors (NN) search. NN algorithms exploit neither global nor neighborhood information as only keypoint descriptors are used while ignoring keypoint location. In practice, this usually leads to poor keypoint matching, which in turn impairs camera pose estimation in SfM pipelines.

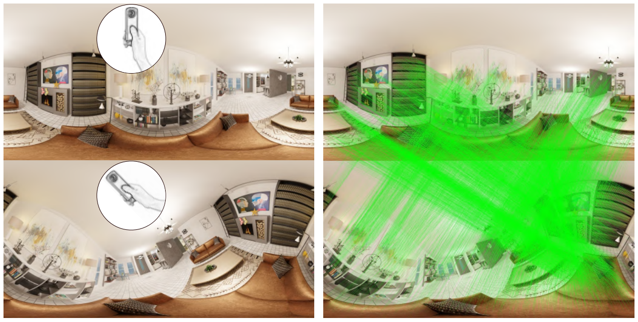

Figure 1. Matching with SphereGlue: keypoints extracted with a front-end handcrafted or learned feature detector are embedded in a *spherical* graph. Combined with a local context aggregation mechanism, our approach naturally models the continuity of the sphere and properly handles distortions caused by camera motion.

Building on recent advances in learned feature matching based on self- and cross-attention for planar perspective images, we propose a novel neural network model for keypoint matching on spherical images by solving a partial soft assignment on the unit sphere. We push the boundaries of learned keypoint matching with two contributions. First, we embed local features as nodes of a *spherical* graph. This allows us to properly model the continuity of the sphere and naturally handle the connectivity of nodes across image boundaries, which is not possible with state-of-the-art learned keypoint matchers. Furthermore, our model is robust to the often severe distortions resulting from spherical geometric transformations caused by camera motion (see Fig. 1). Second, we use graph convolutions based on the Chebyshev polynomial. The use of Chebyshev convolutions overcomes the limitations of a previously introduced message-passing

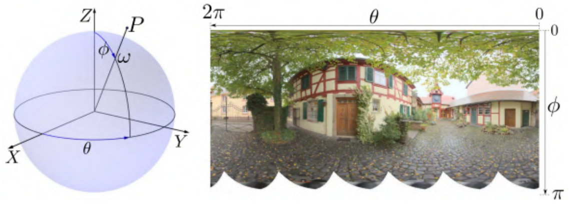

Figure 2. Unit sphere and the relation between spherical coordinates and the equirectangular projection image (ERP).

mechanism for self-attention [53] as it supports localized graph kernels. This formulation provides solid theoretical foundations that allow us to not only define a local neighborhood centered at each node, but also to control it. We show that, even though trained on a small set of $15k$ image pairs, SphereGlue generalizes well to novel synthetic and real-world scenes. Moreover, it outperforms state-of-the-art matching algorithms in spherical SfM by expressive margins.

2. Background and Related Work

Spherical Images

A spherical image is a $180^\circ \times 360^\circ$ environment mapping that captures the entire visible area around the camera. Every visible 3D point P projects onto the surface of the unit sphere at a point ω that is a function of (ϕ, θ). In practice, spherical images are stored as 2D pixel-maps obtained by a latitude-longitude transformation known as an equirectangular projection that produces full panoramic images commonly referred to as ERPs, as shown in Fig. 2.

Different approaches have been proposed to extract keypoints from spherical images. Following the success of SIFT [42], researchers extended it to wide-angle [33, 34] and omnidirectional [16] images by computing the scale space in the spectral domain. In contrast, [28] solves the heat diffusion equation directly on the unit sphere. This allows extraction of keypoints on very high-resolution images at the cost of high computation time. To tackle the efficiency problem while staying in the spatial domain, Zhao *et al*. use local planar approximations [24] to mitigate distortions and present SPHORB [69], a spherical extension to the well-known ORB detector [52]. Although these approaches made significant contributions in detecting keypoints on spherical images, keypoint *matching* is performed using traditional Nearest Neighbors (NN) search.

In the context of spherical SfM, [18, 43] evaluate the performance of different keypoint detectors (planar and spherical for classical [18] and also learned [43] detectors), but are limited to pairs of images. Other methods focus on two- and multi-view spherical pose estimation [29, 31, 46] that can be used, for instance, in dense 3D reconstruction [45]. In all these cases, camera pose estimation can only start once keypoint matches have been established, which is achieved by computing either l_2 or Hamming distance between descriptors, depending on the descriptor type, but the underlying algorithm is nonetheless an instance of NN search.

Deep learning has been applied to spherical images in a variety of research areas, such as object detection [15, 39, 56], 3D model recognition [13, 20], cosmology [20, 40, 47], climate and semantic segmentation [12, 20, 35, 63, 67], shape classification and retrieval [41], single-view depth estimation [58], novel view synthesis [4, 32] and most commonly image or text classification [9, 11, 15, 19, 25, 36, 37, 70]. Similar to keypoint detection, some authors choose to perform convolutions in the spectral domain [9, 11, 13, 25], whereas others prefer the spatial domain. In the latter case, some approaches operate directly on the ERPs [36, 37, 56, 70], use local planar approximations [12, 15, 38, 41, 58, 67] or build on the HEALPix [30] pixelization [20, 21, 47]. However, a learn-based approach for matching keypoints detected on high resolution spherical images is still missing.

Perspective Images

In contrast to spherical images, several methods using deep learning have been developed attempting to improve keypoint matching on perspective images. Recent advances in learn-based local feature detection [6, 10, 22, 23, 44, 50, 57, 60, 65] produced keypoints that are distinctive and robust to rotation, scale, illumination and viewpoint changes. Then, after the computation of putative correspondences obtained with NN search, some approaches aim at improving keypoint matching by classifying matches as inliers or outliers [8, 49, 66, 68], similar in spirit to RANSAC [26]. And like RANSAC, are bound by the aforementioned limitations of NN. To sidestep that limitation, [7] proposes a *handcrafted* statistical framework under the assumption of piece-wise motion smoothness. This assumption is hard to hold in practice, particularly under wide baseline camera motion. Moreover, as pointed out by Yi *et al*. [66], as baseline increases, SAC-based algorithms—which rely on sampling a small subset of the matches to instantiate a hypothesis—tend to suffer because of the increased number of outliers.

Our work is inspired by SuperGlue [53], which replaces the clustering proposed in OA-Net [68] with self- and cross-attention mechanisms and eliminates the need to compute putative keypoint correspondences. This property is specially attractive for high-resolution spherical images, where it is common to detect over $20k$ keypoints per image. However, SuperGlue uses a “planar” graph formulation that reflects the distribution of keypoints on perspective images. Consequently, nodes on opposite sides of the image are weakly or not related. This limits its application to full panoramic images. In contrast, we represent keypoints as nodes on a *spherical* graph, which leads to a natural modeling of the continuity across all borders of the ERPs as context aggregation now depends on the geodesic distance between nodes.

3. Proposed Model

Given a pair of high-resolution spherical images for which keypoints have been detected, we seek a partial assignment that handles the continuity of the sphere and is robust to the non-affine distortions caused by camera motion. Existing works cannot be directly applied to spherical images, as they are designed for perspective images.

Our model builds upon a state-of-the-art perspective keypoint matching technique dubbed SuperGlue [53]. In Section 3.1, we introduce the core aspects of SuperGlue. Section 3.2 presents the proposed approach.

3.1. Overview of SuperGlue

SuperGlue is a neural network designed to predict keypoint correspondences on perspective images by solving a partial soft assignment problem. It combines self- and cross-attention layers for local and global context aggregation as well as to achieve permutation invariance to the input keypoints. A keypoint is denoted by its x and y *pixel* coordinates along with a confidence value c that infers how salient the keypoint is and a descriptor vector $\mathbf{d} \in \mathbb{R}^D$ encoding the visual information in the vicinity of the keypoint location. Together, keypoint coordinates and confidence represent the positional information $\mathbf{p} := (x, y, c)$.

Using Multilayer Perceptron (MLP), each keypoint is embedded into a feature representation $\mathbf{x}_i \in \mathbb{R}^D$ that combines (pixel) position, confidence and visual information:

$$\mathbf{x}_i = \mathbf{d}_i + \text{MLP}_{\text{enc}}\left(\mathbf{p}_i\right). \tag{1}$$

This representation is then fed into a sequence of self- and cross-attention layers that aggregate information using a message-passing strategy. Denoting by $\mathcal{A} := \{1, ..., M\}$ and $\mathcal{B} := \{1, ..., N\}$ the sets of embedded local features of input images A and B, the attention aggregation layers produce features $f^A \in \mathcal{A}$ and $f^B \in \mathcal{B}$ that are subsequently used to express the similarity between keypoints. These similarities are gathered in the so-called score matrix:

$$\mathbf{S}_{i,j} =< f_i^A, f_j^B >, \, \forall (i,j) \in \mathcal{A} \times \mathcal{B}, \tag{2}$$

where $< \cdot, \cdot >$ is the dot product. Matrix $\mathbf{S}$ is then augmented with dustbins to handle occlusion and keypoints that cannot be matched due, for instance, failure of the keypoint detector. The sought partial soft assignment is represented by a matrix $\mathbf{\Gamma}^{M \times N}$ that is obtained after applying the Sinkhorn [17, 54] algorithm and removing the dustbins. Loss is computed as follows. Given ground-truth matches $\mathcal{M} = \{(i,j)\} \subset \mathcal{A} \times \mathcal{B}$ and labeling unmatched keypoints as $\mathcal{I} \subseteq \mathcal{A}$ and $\mathcal{J} \subseteq \mathcal{B}$, the loss $\mathcal{L}$ is given by

$$\mathcal{L} = -\sum_{(i,j)\in\mathcal{M}} \log \bar{\mathbf{\Gamma}}_{i,j} - \sum_{i\in\mathcal{I}} \log \bar{\mathbf{\Gamma}}_{i,N+1} - \sum_{j\in\mathcal{J}} \log \bar{\mathbf{\Gamma}}_{M+1,j}, \tag{3}$$

where $\bar{\mathbf{\Gamma}}^{(M+1)\times(N+1)}$ is the assignment matrix (with dustbins included). Finally, $\mathbf{\Gamma} = \bar{\mathbf{\Gamma}}_{1:M,1:N}$.

3.2. SphereGlue

In this section, our approach and contributions are explained in detail. We adopt a spherical graph model [19–21, 47], where each keypoint location is determined by its unit *Cartesian* coordinates (x, y, z). The choice of Cartesian instead of spherical coordinates (ϕ, θ)—see Fig. 2—is simply because they are more convenient for the computation of geodesic distances. The augmented keypoint position is thus defined as $\mathbf{p}' := (x, y, z, c)$. We also use MLP to combine $\mathbf{p}'$ with the associated visual descriptor $\mathbf{d}$ similar to Eq. 1.

For simplicity, here A and B represent spherical images. Following [47], our spherical graph is modeled as an undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathbf{W})$, where $\mathcal{V}$ is the set of vertices (or nodes) lying on the surface of the unit sphere and $|\mathcal{V}| = M + N$, with M and N the number of local features in A and B, respectively. Edges are represented by $\mathcal{E}$ and $\mathbf{W}$ is the adjacency matrix. The formulations described above are simple but effective. SphereGlue inherits the power and flexibility of SuperGlue with the ability to learn priors over non-affine (often severe) distortions caused by spherical geometric transformations. Moreover, while SuperGlue does not model continuity over image borders, SphereGlue can natively handle partial keypoint assignments in the sphere itself, eliminating the issue of discontinuities altogether.

For our main contribution, we observe that SuperGlue's self-attention mechanism implies aggregation of information from all nodes representing features belonging to the same image. This does not scale well for large graphs, such as those resulting from high-resolution spherical images. Although SuperGlue's self-aggregation mechanism makes sense for perspective images, in the context of spherical images it requires passing information onto nodes that are generally unrelated or irrelevant to a specific node. This is usually the case with antipodal nodes, *i.e.*, those on opposite sides of the sphere. As a result, using SuperGlue's message-passing strategy increases both training and inference time.

One way to address this issue is to replace SuperGlue's message-passing strategy for self-attention with the introduction of a local neighborhood centered at each node. As pointed out in [19, 47], this is indeed possible by using Chebyshev polynomials. They are often used when replacing the spectral formulation of graph convolutions—a process known to be computationally costly—with spatially localized kernels centered at a specific location.

For a given Chebyshev polynomial $\mathbf{T}^{(k)}$ of order k, the Chebyshev convolution produces features $\mathbf{F} \in \mathbb{R}^{Q \times D}$ as

$$\mathbf{F} = \sum_{k=0}^{h} \mathbf{T}^{(k)} \mathbf{X} \mathbf{\Theta}^{(k)}, \tag{4}$$

where $\mathbf{\Theta} \in \mathbb{R}^{D \times D}$ is the matrix of learnable filter parameters, $\mathbf{X} \in \mathbb{R}^{Q \times D}$ contains the keypoint encodings, and $\mathbf{T}^{(k)} \in \mathbb{R}^{Q \times Q}$, with $Q \in \{M, N\}$, *i.e.* the number of key-

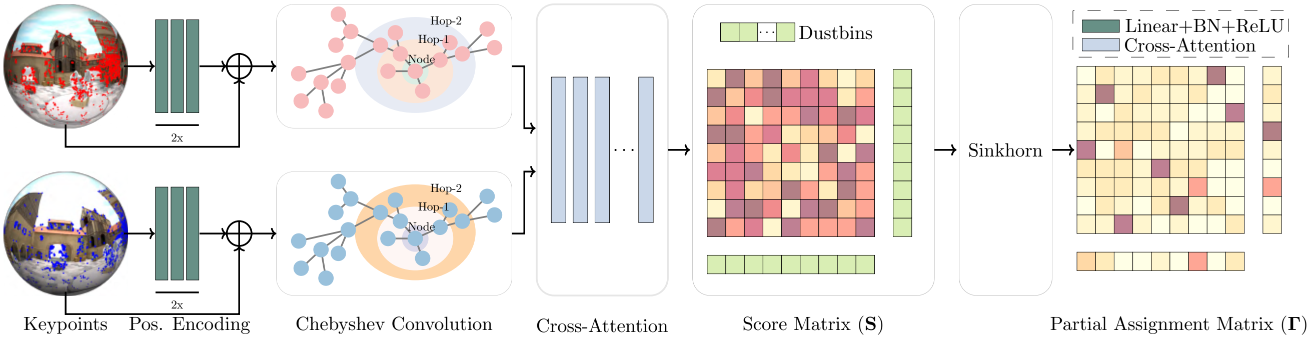

Figure 3. Overview: given two sets of keypoints extracted from high-resolution spherical images, SphereGlue predicts a partial soft assignment matrix $\mathbf{\Gamma}$. Keypoints are encoded combining their location and visual information. The resulting embeddings are fed into the self- and cross-attention blocks (Chebyshev convolutions and message-passing context aggregation layers, respectively). The produced features are gathered in the score matrix $\mathbf{S}$ that expresses the similarity between the keypoints from the input images. The score matrix is augmented with dustbins and Optimal Transport [48, 59], implemented using the Sinkhorn algorithm [17, 54], yields a soft assignment matrix. Finally, after removing the dustbins, the partial soft assignment matrix $\mathbf{\Gamma}$ is obtained.

points in either image A or image B. The parameter h indicates hops, which is the (smallest) number of edges that need to be traversed to connect two nodes. Intuitively, h gives a notion of local neighborhood. See the Chebyshev convolution block in Fig. 3 for a visual representation. Since Chebyshev polynomials may be computed using stable recurrence and denoting $\bar{\mathbf{X}}^{(k)} = \mathbf{T}^{(k)}(\hat{\mathbf{L}})\mathbf{X}$, we can write

$$\begin{aligned} \bar{\mathbf{X}}^{(0)} &= \mathbf{X} \\ \bar{\mathbf{X}}^{(1)} &= \hat{\mathbf{L}}\mathbf{X} \\ \bar{\mathbf{X}}^{(k)} &= 2\hat{\mathbf{L}}\mathbf{X}^{(k-1)} - \mathbf{X}^{(k-2)}. \end{aligned} \tag{5}$$

$\hat{\mathbf{L}} \in \mathbb{R}^{Q\times Q}$ is the rescaled graph Laplacian and given by

$$\hat{\mathbf{L}} = \frac{2\mathbf{L}}{\lambda_{\max}} - \mathbf{I}, \tag{6}$$

where $\mathbf{I}$ is the identity matrix. Finally, $\mathbf{L}$ is the symmetrically normalized Laplacian given by

$$\mathbf{L} = \mathbf{I} - \mathbf{D}^{-1/2}\mathbf{W}\mathbf{D}^{-1/2}, \tag{7}$$

where $\mathbf{W} \in \mathbb{R}^{Q\times Q}$ is the adjacency matrix and $\mathbf{D} \in \mathbb{R}^{Q\times Q}$ is a diagonal matrix so that $\mathbf{D}_{ii} = \sum_j \mathbf{W}_{ij}$.

This way, we can keep the underlying spherical graph structure introduced above and simultaneously influence the convolution by defining the neighborhood while constructing the adjacency matrix. This is achieved by tuning a single scalar parameter: the number of hops h we wish to use, see Eq. 4. This formulation is differentiable, can be easily integrated into the network and provides solid theoretical foundations that allow us to not only define a local neighborhood for each node, but also to *control* it.

We then completely replace the message-passing scheme and implement context aggregation as a Chebyshev convolution block. In contrast to the multiplex architecture proposed in SuperGlue, where self- and cross-attention layers alternate, the Chebyshev convolution block is followed by a sequence of layers implementing only cross-attention. The remaining blocks follow the optimal matching layer proposed in SuperGlue: after augmenting the score matrix $\mathbf{S}$ with dustbins, Sinkhorn [17, 54] is applied and the assignment $\bar{\mathbf{\Gamma}}$ is produced. Finally, the partial assignment $\mathbf{\Gamma}$ is recovered by removing the dustbins. Figure 3 summarizes this idea and provides an overview of SphereGlue. Loss is computed by minimizing the negative log-likelihood as shown in Eq. 3.

4. Experimental Setup

4.1. Dataset

To determine ground-truth keypoint correspondences, it is necessary to have, for each RGB image, an accurate depth map *and* the associated camera pose. Although there is a growing number of datasets in the spherical deep learning community [2, 51, 55, 61, 62, 64, 71–73], none satisfy the conditions above. Even though it is possible to render images according to desired camera poses with Replica [55], the number and size of the scenes are limited and most importantly the depth maps lack the required accuracy. Others, for instance, lack camera pose or the resolution is too small.

We then used Blender [27] to create the required data. A total of 15 artificial scenes were rendered with camera poses randomly generated from a subset of predefined poses. Sample images are depicted in Figure 4(a)-(e). The training set consists of image pairs with maximum baseline of 3 meters and ground-truth keypoint matches, randomly selected from 10 synthetic scenes (7 indoors and 3 outdoors). The test set contains image pairs from 9 novel scenes (5 synthetic and 4 real-world). All 5 synthetic scenes are indoors whereas the real scenes contain both indoor (Meeting Room 1 and 2) and outdoor (Stadium and Town Square) scenarios.

(a) Bank (b) Barbershop (c) Classroom (d) Kartu (e) Warehouse (f) Meeting Room 1 (g) Meeting Room 2 (h) Stadium (i) Town Square

Figure 4. Sample images from our dataset (9 out of 19). Scenes (a)-(e) were used for the experiment presented in Sec. 5.1 and are also part of the training set (except Barbershop, used for testing). Scenes (f)-(i) compose our real-world test set.

All synthetic images as well as Stadium and Town Square are 7070 × 3535 (25 MPixels). Stadium and Town Square were acquired using the Civetta camera [1] whereas Meeting Room 1 and 2 were captured with the Theta-S camera [14] and therefore are 5376 × 2688 (14.5 MPixels).

4.2. Training and Testing

SphereGlue was trained in a supervised manner on a single RTXA6000 over 60 epochs and number of hops $h = 2$. We used 20 nearest neighbors per hop and a fixed learning rate of $1e^{-4}$. Ground-truth keypoint matches are computed for each target feature detector (see Sec. 5 for the chosen detectors) for all image pairs in the training set. We then randomly select a total of $15k$ image pairs ($1.5k$ from each training scene) for which at least 500 ground-truth matches exist. Due to hardware limitations, a maximum of $4k$ keypoints is used for training ($2k$ with ground-truth and $2k$ without correspondences [53]) per image pair. Following [53], we use 20 Sinkhorn iterations. Testing is performed on approx. $170k$ image pairs from the 9 testing scenes. Even though distance between cameras was limited to 3 meters during training, for the real-world scenes all image pairs were used, regardless of the distance between cameras. Also, the maximum number of keypoints per image was raised to $20k$.

5. Results

We start by evaluating SuperGlue on spherical images. The goal is to empirically show the need for an approach that is robust to the spherical geometric transformations under camera motion. We then proceed to the evaluation of SphereGlue against NN search and report results on two classical (SIFT [42] and Akaze [3]) and two learned (SuperPoint [22] and KP2D [57]) keypoint detectors for which code is publicly available. Despite our efforts, the code released by the authors of SPHORB [69] did not yield keypoint matches that were consistent enough for two-view pose estimation. As pointed out in [43], the official implementation of Superpoint fails to retrieve matches under NN search. In this case (NN search), an alternative [1] was used. We refer to the alternative implementation as Superpoint*. Note that SphereGlue results include both implementations. For baseline NN algorithms we use the mutual (or cross-check) and ratio [42] tests. The threshold for the ratio test is 0.75.

Previous work [53, 65] use distance to epipolar line to classify matches as inliers or outliers. We argue that whenever ground-truth matches are available, they should be used instead to measure the matching score (MS). Therefore, we report MS as a ratio between the number of correctly found matches and the number of ground-truth matches.

Evaluation on HPatches [5] carries little significance in the context of spherical images. Furthermore, considering our interest in spherical SfM, in Sec. 5.2 and 5.3 we used [29] to assess the performance of SphereGlue in two-view and multi-view camera pose estimation, respectively.

5.1. SuperGlue on Spherical Images

Sarlin *et al*. [53] state that SuperGlue learns priors over geometric transformations and regularities of the 3D world, but that does not seem to extend to spherical images. To demonstrate that, we conducted the following experiment. We rendered a set of 100 random images from five indoor scenes with different characteristics that aim at capturing the variety and complexity of urban spaces. See Fig. 4 for sample views. For each scene, we apply SuperGlue (with Superpoint as feature detector) using the pre-trained indoor weights on views rendered using two independent criteria: pure rotation and pure translation. We use MS to report the robustness of SuperGlue under each of these conditions.

5.1.1 Pure Rotation

In this experiment, each selected image was rotated around the z-axis (upright) in steps of 10° up to 350°. Figure 5 shows MS as a function of the rotation angle. For rotations close to 0° or 360°, MS is high, as expected, since the rotated images are very similar to the original. However, the score quickly drops, reaching values below 10% for rotations close to 180°. We can conclude that, even without introducing any additional distortion caused by camera displacement (keypoints are only being shifted horizontally), SuperGlue fails to generalize to spherical keypoint matching. Note that this is neither surprising nor a failure of SuperGlue, as it was trained exclusively on perspective images. However, it highlights the need for a method that is robust to spherical transformations that are common under camera motion.

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

Architecture	Detectors	Barbershop			Klaus			Tokyo			Seoul			Shapespark		
		MS	P	R	MS	P	R	MS	P	R	MS	P	R	MS	P	R
SuperGlue	Superpoint	37.24	77.27	37.63	19.77	72.84	19.94	21.81	71.68	22.04	27.59	72.14	27.91	19.29	40.63	22.01
NN-Mutual	Superpoint*	32.63	62.64	34.10	18.03	53.41	18.78	19.10	50.88	20.06	24.00	51.86	25.14	19.98	41.82	22.35
	SIFT	11.71	75.73	11.83	10.07	64.11	10.21	12.18	62.58	12.36	14.83	65.84	15.00	15.57	64.74	15.74
	KP2D	22.43	76.66	22.79	11.93	61.01	12.16	13.65	57.59	14.03	19.25	62.64	19.70	19.44	52.89	20.35
	Akaze	15.89	79.79	16.05	9.25	66.30	9.38	10.10	62.99	10.27	13.00	62.53	13.23	18.50	72.65	18.72
NN-Ratio	Superpoint*	15.63	80.99	15.72	7.90	74.15	7.94	9.46	75.95	9.51	11.09	72.62	11.16	10.44	47.97	10.96
	SIFT	8.71	83.58	8.74	7.46	77.01	7.48	10.41	79.47	10.46	12.36	81.38	12.39	12.26	73.18	12.31
	KP2D	10.60	88.80	10.62	5.28	78.12	5.29	7.29	79.40	7.32	9.81	79.84	9.85	9.47	57.68	9.65
	Akaze	8.74	**89.68**	8.75	4.71	**81.51**	4.72	5.76	**82.59**	5.78	7.81	**83.99**	7.83	8.80	**83.80**	8.81
SphereGlue	Superpoint*	42.77	66.94	45.27	26.24	60.84	27.86	27.74	60.59	29.65	44.88	70.83	48.29	30.30	60.60	32.69
	Superpoint	**53.04**	71.21	**56.20**	33.78	60.93	37.11	**39.79**	66.17	**43.17**	**54.98**	74.83	**59.15**	36.64	62.51	40.25
	SIFT	49.49	58.68	56.14	**38.07**	56.53	**43.50**	32.33	51.89	36.84	49.69	62.37	56.52	**63.97**	64.82	**71.24**
	KP2D	49.81	67.98	53.85	33.86	59.64	37.93	34.36	56.69	38.87	50.33	68.27	56.14	54.01	70.00	58.59
	Akaze	13.14	71.04	13.36	11.77	71.10	12.00	12.21	65.61	12.54	21.17	71.41	22.05	31.90	74.40	33.04

Table 1. Two-view pose estimation: we report matching score (MS), precision (P) and recall (R), in percent. SphereGlue outperforms baseline methods by an expressive margin in MS and R. See Sec. 5.2 and 5.3 for discussions on the results for P.

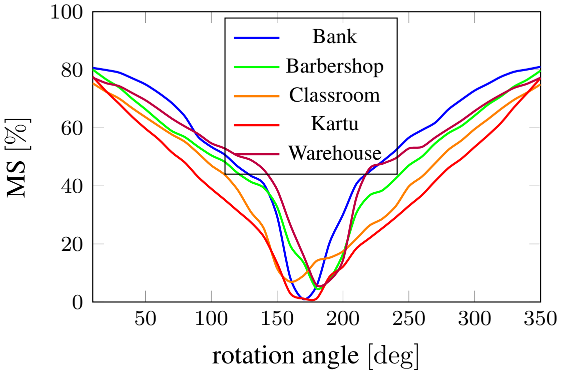

Figure 5. Performance of SuperGlue under pure rotation.

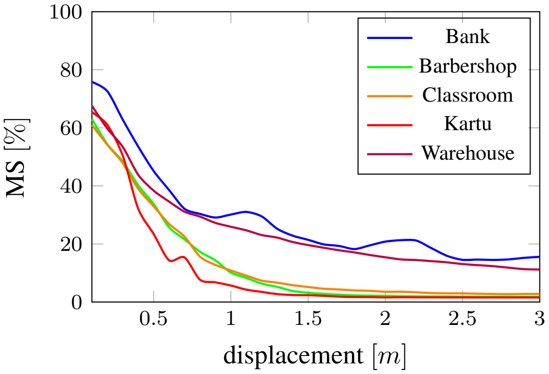

Figure 6. Performance of SuperGlue under pure translation.

5.1.2 Pure Translation

Although SuperGlue can find keypoint matches under challenging perspective viewpoint changes, the same does not hold for spherical images. In this experiment, instead of rotations, we apply pure translation in a random direction with a random displacement in the range $[0, 3]$ meters. We report MS as a function of displacement in Fig. 6. Even for small displacements, the difficulty faced by SuperGlue in finding matches is evident.

5.2. Two-View Pose Estimation

We compare how SphereGlue, NN and SuperGlue [2] matches affect spherical two-view camera pose estimation. Results for SuperGlue were produced with the pre-trained outdoor weights. As in previous work [53, 65], we report matching score (MS) [3], precision (P) and area under the curve (AUC) of the pose error. In addition, we also report recall (R). Since these metrics require ground-truth matches, results presented in this section were obtained from the five synthetic test scenes and exclude the real-world scenes. Images were rendered as described in Sec. 4.1.

Table 1 shows results for MS, P and R. SphereGlue consistently outperforms both NN algorithms and SuperGlue in MS and R by an expressive margin. The exception is precision, which is highest for NN with ratio test for the Akaze detector. Although a more thorough investigation is required to support this, a possible explanation is as follows. Akaze has the shortest descriptor among the detectors used in this work: 61 elements against 128 for SIFT and 256 for Superpoint and KP2D. Under the assumption that the descriptor length is proportional to its capacity to encode information, the Akaze descriptor is less discriminative than its longer counterparts. Given the high number of extracted keypoints (frequently above $10k$ per image) and considering the abundant self-similarities in indoor scenarios, it is expected that a fraction of the Akaze descriptors are, in general, similar to each other. Hence, only a small portion (likely consisting of very good matches) survives the ratio test. This is confirmed by the high P values along with low values of corresponding MS and R. As we show in the Sec. 5.3, this can be harmful for spherical multi-view camera pose estimation. Further analysis is left as future work.

[2] included only in the two-view pose estimation for reference.

[3] computed as the ratio between the number of correctly found matches and ground-truth matches.

Architecture	Detectors	Barbershop			Klaus			Tokyo			Seoul			Shapespark		
		5°	10°	20°	5°	10°	20°	5°	10°	20°	5°	10°	20°	5°	10°	20°
SuperGlue	Superpoint	98.96	99.19	99.54	94.07	95.37	96.61	93.09	95.40	97.13	84.25	88.33	91.81	46.68	49.85	53.91
NN-Mutual	Superpoint*	98.85	99.66	**99.94**	82.94	85.74	88.68	94.03	96.95	98.03	62.61	68.58	75.35	56.94	61.16	65.77
	SIFT	77.21	89.69	97.19	49.41	58.91	67.38	42.45	59.40	74.46	33.83	38.1	42.94	66.66	74.99	82.88
	KP2D	93.87	97.13	99.14	57.07	62.44	66.74	62.22	75.98	85.94	48.10	55.07	63.74	63.18	68.18	73.22
	Akaze	83.56	91.12	96.39	43.62	54.53	65.59	47.62	65.17	82.67	29.76	35.27	42.01	93.01	96.06	98.11
NN-Ratio	Superpoint*	98.33	98.56	99.14	88.27	90.82	93.44	78.45	85.89	91.66	71.65	78.32	85.83	47.13	52.75	59.71
	SIFT	**99.37**	**99.77**	**99.94**	79.14	81.73	84.68	85.42	91.42	95.87	70.91	75.78	81.14	79.01	82.66	87.52
	KP2D	93.20	94.87	95.96	63.11	68.34	73.22	65.28	74.81	83.10	65.23	72.09	79.99	51.14	56.64	63.66
	Akaze	99.02	99.54	99.77	73.18	78.19	83.34	79.93	87.39	92.37	65.12	71.73	79.97	95.97	97.40	98.54
SphereGlue	Superpoint*	99.19	99.48	99.88	**94.46**	**95.94**	**97.30**	**99.85**	**99.93**	99.93	96.03	**97.24**	**98.35**	84.41	87.42	90.65
	Superpoint	99.31	99.59	99.82	91.29	93.44	95.68	99.72	99.90	**99.98**	**96.49**	97.22	98.06	82.81	86.44	91.12
	SIFT	99.02	99.14	99.66	90.63	93.34	95.68	96.94	98.85	99.53	88.18	91.26	94.19	**99.77**	99.86	99.93
	KP2D	98.79	99.19	99.82	89.06	91.97	94.60	96.40	98.35	99.25	92.44	94.79	96.85	99.67	**99.89**	**99.96**
	Akaze	99.02	99.65	**99.94**	87.80	91.01	93.91	93.18	97.16	98.52	70.82	78.66	86.21	99.22	99.49	99.68

Table 2. Two-view pose estimation: we report AUC of the relative pose error, in percent. SphereGlue consistently outperforms both NN algorithms across scenes and keypoint detectors in nearly all scenarios. See Sec. 5.2 for details.

Table 2 shows AUC for relative pose estimation error at 5°, 10° and 20° thresholds. Relative pose error is simply the maximum angular error in rotation and translation. Except for the Barbershop scene, where NN with ratio test for SIFT is marginally superior for 5° and 10°, SphereGlue once again outperforms the baselines. It is worth mentioning that the significantly higher AUC values we obtain when compared to SuperGlue are in part due to the fact that we use spherical instead of perspective images. In other words, it is expected to obtain more accurate camera poses when spherical images are used. The reason is that in this case, constraints on the pose (given by bearing vectors defined by the keypoint locations) are usually distributed all over the unit sphere — in contrast to perspective images, where the bearing vectors can only span the (limited) field of view of the camera — and better drive optimization algorithms towards the true camera pose. Figure 7 shows qualitative keypoint matches for all real-world test scenes. In both indoor and outdoor scenarios, keypoint matches established by SphereGlue are cleaner than those retrieved by NN with mutual test and richer than those produced by NN with ratio test.

Scenes	Detectors	NN-Mutual	NN-Ratio	SphereGlue
Barbershop (80)	Superpoint	-	-	80
	Superpoint*	61	74	80
	SIFT	68	68	80
	KP2D	71	79	80
	Akaze	68	79	80
Klaus (550)	Superpoint	-	-	549
	Superpoint*	550	549	550
	SIFT	532	514	550
	KP2D	547	340	550
	Akaze	550	300	549
Tokyo (90)	Superpoint	-	-	89
	Superpoint*	79	90	90
	SIFT	71	77	88
	KP2D	69	85	90
	Akaze	66	82	90
Seoul (330)	Superpoint	-	-	320
	Superpoint*	329	321	321
	SIFT	249	305	330
	KP2D	313	321	330
	Akaze	318	143	319
Shapepark (860)	Superpoint	-	-	860
	Superpoint*	858	860	856
	SIFT	791	851	860
	KP2D	841	860	859
	Akaze	855	860	856
Town Square (35)	Superpoint	-	-	35
	Superpoint*	35	35	35
	SIFT	35	32	35
	KP2D	35	35	35
	Akaze	35	35	35
Stadium (74)	Superpoint	-	-	74
	Superpoint*	74	73	74
	SIFT	74	71	73
	KP2D	72	73	74
	Akaze	74	71	74
Meeting Room 1 (18)	Superpoint	-	-	18
	Superpoint*	5	3	9
	SIFT	18	18	18
	KP2D	18	18	16
	Akaze	18	18	18
Meeting Room 2 (21)	Superpoint	-	-	13
	Superpoint*	17	21	21
	SIFT	20	21	21
	KP2D	11	21	20
	Akaze	17	20	21

Table 3. Multi-view pose estimation: we report the number of camera poses recovered. SphereGlue can directly replace NN algorithms in spherical SfM pipelines. See Sec. 5.3 for details.

5.3. Multi-View Pose Estimation

In this section, we take a step further and evaluate SphereGlue against NN in the more difficult task of spherical multi-view camera pose estimation. Here we consider all synthetic and real-world test settings. Table 3 summarizes the number of camera poses successfully recovered. The original number of cameras present in each scene is indicated next to the scene name. Synthetic scenes allow us to create challenging settings by rendering a large number of cameras with a variety of poses that result in severe image distortions. For instance, Seoul, Klaus and Shapespark contain 330, 550 and 860 images, respectively, and represent typical urban indoor apartments. Real-world scenarios are also challenging due to repetitive textures and strong illumination changes — in case of outdoor — as well as lack of textures and high self-similarities in man-made structures, such as indoor environments (see Fig. 4).

Thanks to the wide field of view of spherical images and

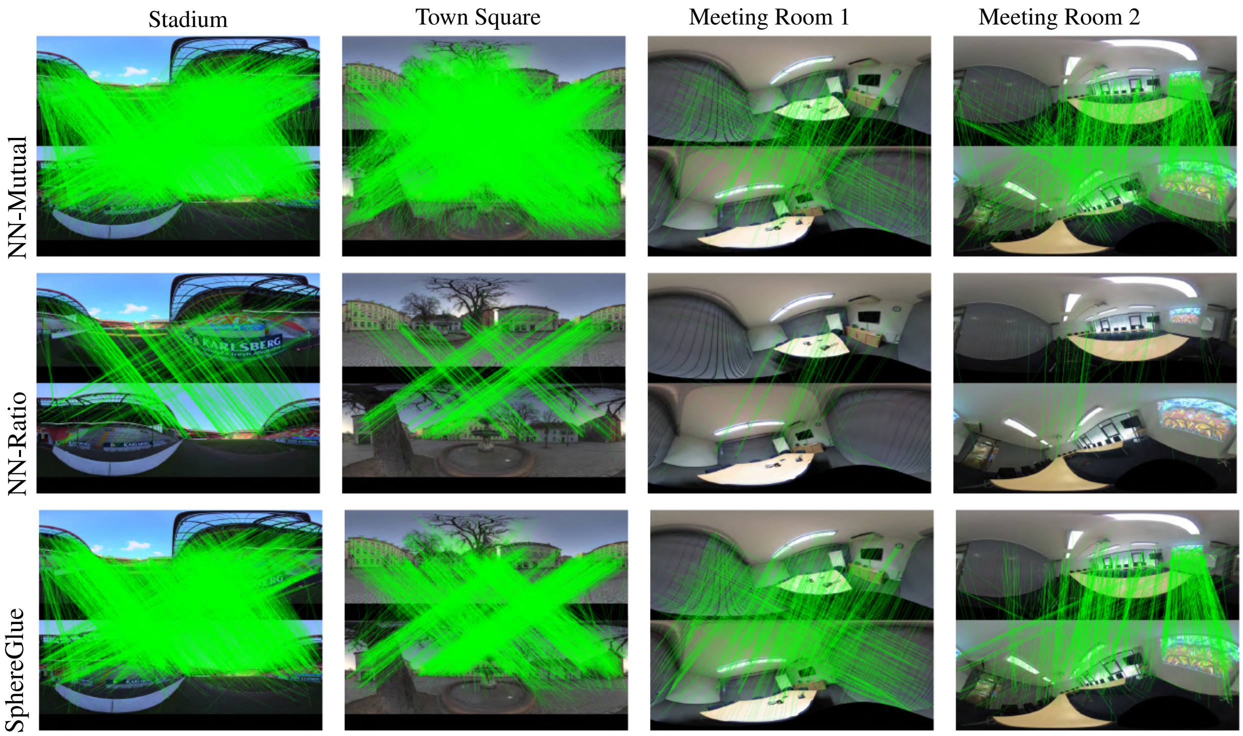

Figure 7. Qualitative keypoint matches: we compare SphereGlue to NN with mutual and ratio tests. NN with mutual test usually retrieves a large number of matches, specially in well-textured environments, but also containing several outliers. SphereGlue produces almost as many matches with significantly less outliers. NN with ratio test tends to produce a clean set of matches at the cost of poor performance in indoor places. In contrast, SphereGlue establishes a richer set of matches even under wide baseline and high distortions. Note how SphereGlue naturally handles the continuity of the sphere and generalizes to very wide, unseen, baselines (Stadium and Town Square).

large number of keypoints extracted per image, in general, enough matches can be found across multiple images. As a result, all detectors performed well under NN and in combination with SphereGlue. Nevertheless, SphereGlue is clearly more reliable than NN across scenes and keypoint detectors. In the following we provide two ways to highlight that. First, when considering only the cases where all camera poses were recovered, SphereGlue succeeds in 30 out of 45 (9 scenes times 5 detectors) cases (66.67%) against 12 and 13 out of 36 cases for NN mutual and ratio, *i.e.* 33.33% and 36.11%, respectively. Second, considering all test scenes and keypoint detectors together (sum of original number of images per scene times the number of detectors), there is a total of 10,290 and 8,232 camera poses to recover for SphereGlue and NN search, respectively. While NN with mutual and ratio tests recover 95.29% and 90.17% of all camera poses, SphereGlue successfully recovers 99.37%.

Finally, as indicated in Sec. 5.2, the fact that NN with ratio test for Akaze has the highest precision does not imply it is more suitable for spherical SfM. This becomes evident from the analysis of results reported in Table 3: it performs worse than NN with mutual test and SphereGlue regarding the total number of recovered camera poses and delivers particularly poor results for Klaus and Seoul scenes.

6. Conclusion

In this paper we proposed the first trainable neural-based keypoint matcher for spherical images. Inspired by recent advances in local feature matching for perspective images, we re-formulate keypoint matching as a partial soft assignment on spherical graphs, which naturally and efficiently model the underlying data. SphereGlue completely replaces the previously introduced message-passing strategy for self-attention with Chebyshev convolution layers, allowing us to fully control the size of the local neighborhood and simultaneously speed up training and inference. Results show that SphereGlue can replace state-of-the-art matching algorithms in spherical camera pose estimation pipelines. Future work includes training on larger sets of image pairs with wider baselines and under more challenging illumination changes. Also, considering that Sinkhorn is an iterative algorithm and thus requires a given number of iterations, investigating alternatives that could speed up training and inference are interesting research directions.

Acknowledgements

This work was supported by projects SustainML (101070408) and Carl Zeiss Stiftung Germany (P2021-02-009).

References

[1] Weiss AG. Civetta. https://weiss-ag.com/civetta360camera/. Retrieved March, 2023. 5

[2] Georgios Albanis, Nikolaos Zioulis, Petros Drakoulis, Vasileios Gkitsas, Vladimiros Sterzentsenko, Federico Alvarez, Dimitrios Zarpalas, and Petros Daras. Pano3d: A holistic benchmark and a solid baseline for 360° depth estimation. *CoRR*, 2021. 4

[3] Pablo Fernández Alcantarilla, Jesús Nuevo, and Adrien Bartoli. Fast Explicit Diffusion for Accelerated Features in Nonlinear Scale Spaces. In *BMVC*, 2013. 5

[4] Benjamin Attal, Selena Ling, Aaron Gokaslan, Christian Richardt, and James Tompkin. Matryodshka: Real-time 6dof video view synthesis using multi-sphere images. In *ECCV*, pages 441–459, 2020. 2

[5] Vassileios Balntas, Karel Lenc, Andrea Vedaldi, and Krystian Mikolajczyk. Hpatches: A benchmark and evaluation of handcrafted and learned local descriptors. In *CVPR*, 2017. 5

[6] Anatoly Belikov and Alexey Potapov. Goodpoint: unsupervised learning of keypoint detection and description. *CoRR*, 2020. 2

[7] Jiawang Bian, Wen-Yan Lin, Yasuyuki Matsushita, Sai-Kit Yeung, Tan-Dat Nguyen, and Ming-Ming Cheng. GMS: Grid-based motion statistics for fast, ultra-robust feature correspondence. In *IEEE/CVPR*, pages 2828–2837, 2017. 2

[8] Eric Brachmann and Carsten Rother. Neural-guided ransac: Learning where to sample model hypotheses. In *ICCV*, page 4321–4330, May 2019. 1, 2

[9] Joan Bruna, Wojciech Zaremba, Arthur Szlam, and Yann Lecun. Spectral networks and locally connected networks on graphs. In *ICLR*, 2014. 2

[10] Peter Hviid Christiansen, Mikkel Fly Kragh, Yury Brodskiy, and Henrik Karstoft. Unsuperpoint: End-to-end unsupervised interest point detector and descriptor. *CoRR*, abs/1907.04011, 2019. 2

[11] Taco Cohen, Mario Geiger, Jonas Köhler, and Max Welling. Convolutional networks for spherical signals. *CoRR*, abs/1709.04893, 2017. 2

[12] Taco Cohen, Maurice Weiler, Berkay Kicanaoglu, and Max Welling. Gauge equivariant convolutional networks and the icosahedral CNN. In *ICML*, volume 97, pages 1321–1330. PMLR, 2019. 2

[13] Taco S. Cohen, Mario Geiger, Jonas Köhler, and Max Welling. Spherical cnns. In *ICLR*, Vancouver, April 30 - May 3 2018. OpenReview.net. 2

[14] Ricoh Company. Ricoh Theta S. https://theta360.com/en/about/theta/s.html. Retrieved March, 2023. 5

[15] Benjamin Coors, Alexandru Paul Condurache, and Andreas Geiger. Spherenet: Learning spherical representations for detection and classification in omnidirectional images. In *ECCV*. Springer, 2018. 2

[16] Javier Cruz-Mota, Iva Bogdanova, Benoît Paquier, Michel Bierlaire, and Jean-Philippe Thiran. Scale Invariant Feature Transform on the Sphere: Theory and Applications. *IJCV*, 98(2):217–241, 2012. 1, 2

[17] Marco Cuturi. Sinkhorn distances: Lightspeed computation of optimal transport. In *NIPS*, volume 26, 2013. 3, 4

[18] Thiago Lopes Trugillo da Silveira and Cláudio Rosito Jung. Evaluation of keypoint extraction and matching for pose estimation using pairs of spherical images. In *SIBGRAPI*, pages 374–381. IEEE Computer Society, 2017. 2

[19] Michaël Defferrard, Xavier Bresson, and Pierre Vandergheynst. Convolutional neural networks on graphs with fast localized spectral filtering. In *NeurIPS*, pages 3837–3845, Barcelona, 2016. 2, 3

[20] Michaël Defferrard, Martino Milani, Frédérick Gusset, and Nathanaël Perraudin. Deepsphere: a graph-based spherical CNN. In *ICLR*, Addis Ababa, Ethiopia, April 2020. OpenReview.net. 2, 3

[21] Michaël Defferrard, Nathanaël Perraudin, Tomasz Kacprzak, and Raphael Sgier. Deepsphere: towards an equivariant graph-based spherical CNN. *CoRR*, abs/1904.05146, 2019. 2, 3

[22] Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. Superpoint: Self-supervised interest point detection and description. In *IEEE/CVPR*, pages 224–236. Computer Vision Foundation / IEEE Computer Society, 2018. 1, 2, 5

[23] Mihai Dusmanu, Ignacio Rocco, Tomas Pajdla, Marc Pollefeys, Josef Sivic, Akihiko Torii, and Torsten Sattler. D2-Net: A Trainable CNN for Joint Detection and Description of Local Features. In *CVPR*, June 2019. 2

[24] Marc Eder, Mykhailo Shvets, John Lim, and Jan-Michael Frahm. Tangent images for mitigating spherical distortion. In *IEEE/CVPR*, June 2020. 1, 2

[25] Carlos Esteves, Christine Allen-Blanchette, Ameesh Makadia, and Kostas Daniilidis. Learning SO(3) equivariant representations with spherical cnns. In *ECCV*, volume 11217 of *Lecture Notes in Computer Science*, pages 54–70, Munich, September 2018. Springer. 2

[26] Martin A. Fischler and Robert 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

[27] Blender Foundation. Blender. http://www.blender.org/. Retrieved March, 2023. 4

[28] Christiano Couto Gava, Jean-Marc Hengen, Bertram Tätz, and Didier Stricker. Keypoint detection and matching on high resolution spherical images. In *ISVC*, pages 363–372, July 2013. 1, 2

[29] Christiano Couto Gava and Didier Stricker. A generalized structure from motion framework for central projection cameras. In *Computer Vision, Imaging and Computer Graphics Theory and Applications*. Springer, 2016. 2, 5

[30] K. M. Górski, E. Hivon, A. J. Banday, B. D. Wandelt, F. K. Hansen, M. Reinecke, and M. Bartelmann. HEALPix: A framework for high-resolution discretization and fast analysis of data distributed on the sphere. *The Astrophysical Journal*, 622(2), 2005. 2

[31] Hao Guan and William A. P. Smith. Structure-from-motion in spherical video using the von mises-fisher distribution. *IEEE Transactions on Image Processing*, 26(2):711–723, 2017. 2

[32] Tewodros Habtegebrial, Christiano Gava, Marcel Rogge, Didier Stricker, and Varun Jampani. Somsi: Spherical novel

view synthesis with soft occlusion multi-sphere images. In *IEEE/CVPR*, pages 15725–15734, 2022. 2

[33] Peter Hansen, Wageeh W. Boles, and Peter Corke. Spherical diffusion for scale-invariant keypoint detection in wide-angle images. In *DICTA*, pages 525–532, 2008. 2

[34] Peter Hansen, Peter Corke, Wageeh Boles, and Kostas Daniilidis. Scale invariant feature matching with wide angle images. In *IROS*, pages 1689–1694. IEEE, 2007. 2

[35] Chiyu Max Jiang, Jingwei Huang, Karthik Kashinath, Prabhat, Philip Marcus, and Matthias Nießner. Spherical cnns on unstructured grids. In *ICLR*, New Orleans, LA, May 6-9 2019. OpenReview.net. 2

[36] R. Khasanova and P. Frossard. Graph-based classification of omnidirectional images. In *ICCVW*, pages 860–869, 2017. 2

[37] Renata Khasanova and Pascal Frossard. Graph-based isometry invariant representation learning. *CoRR*, 2017. 2

[38] Renata Khasanova and Pascal Frossard. Geometry aware convolutional filters for omnidirectional images representation. In *ICML*, volume 97 of *Proceedings of Machine Learning Research*, pages 3351–3359. PMLR, 2019. 2

[39] Risi Kondor, Zhen Lin, and Shubhendu Trivedi. Clebsch-gordan nets: a fully fourier space spherical convolutional neural network. In *NeurIPS*, pages 10138–10147, 2018. 2

[40] N. Krachmalnicoff and M. Tomasi. Convolutional neural networks on the healpix sphere: a pixel-based algorithm and its application to cmb data analysis. *Astronomy & Astrophysics*, 628, Aug 2019. 2

[41] Min Liu, Fupin Yao, Chiho Choi, Ayan Sinha, and Karthik Ramani. Deep learning 3d shapes using alt-az anisotropic 2-sphere convolution. In *ICLR*, New Orleans, May 2019. OpenReview.net. 2

[42] David G. Lowe. Distinctive image features from scale-invariant keypoints. *Int. J. Comput. Vision*, 60:91–110, November 2004. 2, 5

[43] Jeffri Murrugarra-Llerena, Thiago L. T. da Silveira, and Claudio R. Jung. Pose estimation for two-view panoramas based on keypoint matching: A comparative study and critical analysis. In *IEEE/ICCV*, pages 5202–5211, June 2022. 2, 5

[44] Yuki Ono, Eduard Trulls, Pascal Fua, and Kwang Moo Yi. Lf-net: Learning local features from images. *NeurIPS*, 31, 2018. 2

[45] Alain Pagani, Christiano Gava, Yan Cui, Bernd Krolla, Jean-Marc Hengen, and Didier Stricker. Dense 3d point cloud generation from multiple high-resolution spherical images. In *VAST*, Prato, Italy, October 2011. 2

[46] Alain Pagani and Didier Stricker. Structure from motion using full spherical panoramic cameras. In *OMNIVIS*, 2011. 2

[47] Nathanaël Perraudin, Michaël Defferrard, Tomasz Kacprzak, and Raphael Sgier. Deepsphere: Efficient spherical convolutional neural network with healpix sampling for cosmological applications. *CoRR*, abs/1810.12186, 2018. 2, 3

[48] Gabriel Peyré and Marco Cuturi. *Computational Optimal Transport*, volume 11 (5-6). Foundations and Trends in Machine Learning, 2019. 4

[49] Rene Ranftl and Vladlen Koltun. Deep fundamental matrix estimation. In *ECCV*, 2018. 2

[50] Jerome Revaud, Cesar De Souza, Martin Humenberger, and Philippe Weinzaepfel. R2d2: Reliable and repeatable detector and descriptor. In *NeurIPS*. Curran Associates, Inc., 2019. 2

[51] Mike Roberts and Nathan Paczan. Hypersim: A photorealistic synthetic dataset for holistic indoor scene understanding. *CoRR*, 2020. 4

[52] Ethan Rublee, Vincent Rabaud, Kurt Konolige, and Gary Bradski. ORB: an efficient alternative to SIFT or SURF. *IEEE/ICCV*, pages 2564–2571, 2011. 2

[53] Paul-Edouard Sarlin, Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. SuperGlue: Learning feature matching with graph neural networks. In *CVPR*, 2020. 1, 2, 3, 5, 6

[54] Richard Sinkhorn and Paul Knopp. Concerning nonnegative matrices and doubly stochastic matrices. *Pacific Journal of Mathematics*, 21:343–348, 1967. 3, 4

[55] Julian Straub, Thomas Whelan, Lingni Ma, Yufan Chen, Erik Wijmans, Simon Green, Jakob J. Engel, Raul Mur-Artal, Carl Ren, Shobhit Verma, Anton Clarkson, Mingfei Yan, Brian Budge, Yajie Yan, Xiaqing Pan, June Yon, Yuyang Zou, Kimberly Leon, Nigel Carter, Jesus Briales, Tyler Gillingham, Elias Mueggler, Luis Pesqueira, Manolis Savva, Dhruv Batra, Hauke M. Strasdat, Renzo De Nardi, Michael Goesele, Steven Lovegrove, and Richard A. Newcombe. The replica dataset: A digital replica of indoor spaces. *CoRR*, 2019. 4

[56] Yu-Chuan Su and Kristen Grauman. Learning spherical convolution for fast features from 360° imagery. In *NeurIPS*, pages 529–539, 2017. 2

[57] Jiexiong Tang, Hanme Kim, Vitor Guizilini, Sudeep Pillai, and Rares Ambrus. Neural outlier rejection for self-supervised keypoint learning. In *ICLR*, Addis Ababa, Ethiopia, April 2020. OpenReview.net. 1, 2, 5

[58] Keisuke Tateno, Nassir Navab, and Federico Tombari. Distortion-aware convolutional filters for dense prediction in panoramic images. In *ECCV*, September 2018. 2

[59] C. Villani. *Optimal Transport: Old and New*. Springer Berlin Heidelberg, 2016. 4

[60] Bing Wang, Changhao Chen, Zhaopeng Cui, Jie Qin, Chris Xiaoxuan Lu, Zhengdi Yu, Peijun Zhao, Zhen Dong, Fan Zhu, Niki Trigoni, et al. P2-net: Joint description and detection of local features for pixel and point matching. In *IEEE/ICCV*, 2021. 1, 2

[61] Changhee Won, Jongbin Ryu, and Jongwoo Lim. Sweepnet: Wide-baseline omnidirectional depth estimation. In *IEEE/ICRA*, pages 6073–6079, 2019. 4

[62] Changhee Won, Jongbin Ryu, and Jongwoo Lim. End-to-end learning for omnidirectional stereo matching with uncertainty prior. *IEEE/PAMI*, 2020. 4

[63] Yuanyou Xu, Kaiwei Wang, Kailun Yang, Dongming Sun, and Jia Fu. Semantic segmentation of panoramic images using a synthetic dataset. *CoRR*, 2019. 2

[64] Karmesh Yadav, Ram Ramrakhya, Santhosh Kumar Ramakrishnan, Theo Gervet, John Turner, Aaron Gokaslan, Noah Maestre, Angel Xuan Chang, Dhruv Batra, Manolis Savva, Alexander William Clegg, and Devendra Singh Chaplot. Habitat-matterport 3d semantics dataset, 2022. 4

[65] Kwang Moo Yi, Eduard Trulls, Vincent Lepetit, and Pascal Fua. Lift: Learned invariant feature transform. In *ECCV*. Springer International Publishing, 2016. 2, 5, 6

[66] Kwang Moo Yi, Eduard Trulls, Yuki Ono, Vincent Lepetit, Mathieu Salzmann, and Pascal Fua. Learning to find good correspondences. In *IEEE/CVPR*, pages 2666–2674. Computer Vision Foundation / IEEE Computer Society, 2018. 1, 2

[67] Chao Zhang, Stephan Liwicki, William Smith, and Roberto Cipolla. Orientation-aware semantic segmentation on icosahedron spheres. In *ICCV*, October 2019. 2

[68] Jiahui Zhang, Dawei Sun, Zixin Luo, Anbang Yao, Lei Zhou, Tianwei Shen, Yurong Chen, Hongen Liao, and Long Quan. Learning two-view correspondences and geometry using order-aware network. In *IEEE/ICCV*, pages 5844–5853, 2019. 1, 2

[69] Qiang Zhao, Wei Feng, Liang Wan, and Jiawan Zhang. SPHORB: A fast and robust binary feature on the sphere. *Int. J. Comput. Vis.*, 113(2):143–159, 2015. 1, 2, 5

[70] Qiang Zhao, Chen Zhu, Feng Dai, Yike Ma, Guoqing Jin, and Yongdong Zhang. Distortion-aware cnns for spherical images. In *IJCAI*, pages 1198–1204, 2018. 2

[71] Jia Zheng, Junfei Zhang, Jing Li, Rui Tang, Shenghua Gao, and Zihan Zhou. Structured3d: A large photo-realistic dataset for structured 3d modeling. *CoRR*, 2019. 4

[72] Nikolaos Zioulis, Antonis Karakottas, Dimitris Zarpalas, Federic Alvarez, and Petros Daras. Spherical view synthesis for self-supervised 360^o depth estimation. In *International Conference on 3D Vision (3DV)*, September 2019. 4

[73] Nikolaos Zioulis, Antonis Karakottas, Dimitrios Zarpalas, and Petros Daras. Omnidepth: Dense depth estimation for indoors spherical panoramas. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 448–465, 2018. 4