Detecting Low-Rank Regions in Omnidirectional Images*

Zoltan Kato[1,2], Gabor Nagy[1], Martin Humenberger[3], Gabriela Csurka[3]
[1]University of Szeged, Institute of Informatics, Hungary
[2]J. Selye University, Komarno, Slovakia
[3]NAVER LABS Europe, France
{kato,nagyg08}@inf.u-szeged.hu {martin.humenberger,gabriela.csurka}@naverlabs.com

Abstract

Planar low-rank regions commonly found in man-made environments, can be used to estimate a rectifying homography that provides valuable information about the camera and the 3D plane they observe. Methods to recover such a homography exist, but detection of low-rank regions is largely unsolved, especially for omnidirectional cameras where significant distortions make the problem even more challenging. In this paper we address this problem as follows. First we propose a method to generate a low-rank probability map on an omnidirectional image and use it to build a training set in a self-supervised manner to train deep models to predict low-rank likelihood maps for omnidirectional images. Second, we propose to adapt regular CNN operators to equirectangular images and to combine them seamlessly into a network where each layer preserves the properties of the equirectangular representation. Finally, on the new KITTI360 dataset, we show that the rectifying homography of detected low-rank regions in such predicted maps allows to factorize out the camera-plane pose up to certain ambiguities that can be easily overcome.

1. Introduction

Omnidirectional or panoramic cameras offer a wide field of view which can be very useful in many applications such as autonomous driving [79, 27, 74, 77], SLAM and indoor robot navigation [59, 52, 7, 51, 34], virtual reality [26, 15, 45], and monitoring systems [9, 25]. This advantage over classical perspective cameras comes with the drawback that the major part of the computer vision literature, addressing the challenges of these applications, was written for perspective cameras. This applies for both, geometric approaches and even more for deep neural networks. Nonetheless, given the advantages of a large field of view of these cameras [81], there is an obvious interest in the computer vision community for models that are designed for this kind of data. In this paper, we are particularly interested in applications such as visual localization [58, 56, 55] or structure-from-motion (SFM) [62, 41] where precise camera pose estimation between two cameras or one camera and physical structure is crucial. Geometry based approaches often critically rely on local keypoint detection and matching [13, 63, 5], which is often challenged by textureless areas, distortions, occlusions, strong illumination changes, and repetitive pattern. Furthermore, if the environment changed between reference image and query image acquisition, even the best feature extractors will have difficulties to find enough relevant correspondences. As an alternative approach, [65, 80, 44] show that by rectifying a so called planar low-rank region, it is possible to estimate the camera pose relative to the corresponding plane in the 3D world. This was further extended in [54, 18, 46] to estimate the relative pose of a complete camera network to a 3D plane. The main idea is to make use of the intrinsic structure of such low-rank textures which can be commonly found in urban environments, *e.g.* on planar building facades (bricks, ornaments, windows, etc.), in order to recover a low-rank matrix and a sparse error matrix where the first one is a canonical view of the region obtained via a *rectifying homography*.

These methods mentioned were designed for perspective cameras and applying them to spherical images is not straightforward. Therefore, as a **first contribution**, we extend these methods, and in particular TILT [80], to omnidirectional cameras.

While TILT provides a solid mathematical optimization framework to estimate the low-rank matrices for given regions, it does not detect low-rank regions. In [14], TILT was used to generate a low-rank likelihood map for an entire image by low-rank decomposition of image patches extracted at multiple scales. Then these maps were used to train a convolutional neural network (CNN) to predict such maps for new images. Their deep network, relying on standard

*This work was partially supported by K120366 of the NKFI-6 fund; EFOP-3.6.3-VEKOP-16-2017-0002; ITMS26210120042 and NFP313010T504 co-funded by the European Regional Development Fund

convolutions, was designed for perspective cameras and as shown in [28, 33, 78, 76, 10, 31, 76, 10] applying standard CNNs or adapting them to omnidirectional images is not trivial. The challenges are caused by distortions on planar representations of a sphere which violate the requirement that the convolution kernel and the signal should be uniformly discretized. In this paper, as a **second contribution**, and taking inspiration from [11, 70, 16], we adapt the low-rank detection network proposed in [14] to equirectangular images. To train our model, called SphSegNet, we rely on a set of low-rank likelihood maps computed for omnidirectional images by applying the adapted TILT algorithm on sliding spherical windows at multiple scales.

Finally, as a **third contribution**, we show on the new KITT360 dataset that the rectifying homography of detected low-rank regions in such predicted likelihood maps allows to factorize out the camera-plane pose up to certain ambiguities that can be easily overcome.

2. Related work

Repetitive and low-rank structures. Popular methods detecting repetitive structures are based on local feature grouping [61, 47, 71] or on the assumption of a single pattern repeated on a 2D (deformed) lattice [22, 43]. Such patterns were used, *e.g.*, for single view facade rectification [8], camera pose estimation [61], or single-view 3D reconstruction [72]. In particular, low-rank patterns were addressed in [44, 80] where an optimal transformations of an image region is iteratively found that can be decomposed into a low-rank matrix and a sparse error matrix. These methods, implicitly assuming perspective images, cannot be directly applied to omnidirectional images as the patterns are not anymore repetitive or low-rank due to the nonlinear distortion of such cameras.

Geometry of omnidirectional images. The geometric formulation of omnidirectional systems was extensively studied [42, 3, 20, 39, 60, 50]. The internal calibration of such cameras depends on these geometric models, which can be solved in a controlled environment, using special calibration pattern [60, 29, 38, 50]. When the camera is calibrated, which is typically the case in practical applications, image points can be lifted to the surface of a unit sphere providing a unified model independent of the inner, non-linear projection of the camera. Unlike the projective case, homography is estimated using these spherical points [37, 6, 19]. A classical solution is to establish a set of point matches and then to estimate the homography based on those. Unfortunately, big variations in shape, resolution, and non-linear distortion, challenges keypoint detectors as well as the extraction of invariant descriptors, which are key components of reliable point matching. For example, proper handling of scale-invariant feature extraction requires special considerations in case of omnidirectional sensors, yielding mathematically elegant but complex algorithms [49]. In [21], a new computation of descriptor patches was introduced for catadioptric omnidirectional cameras which also aims to reach rotation and scale invariance. In [36], a correspondence-less algorithm is proposed to recover relative camera motion.

Deep models on omnidirectional images. Applying CNNs to omnidirectional images is not trivial as any planar representation of a sphere necessarily contains some degree of content deformation which violates the requirement that the convolution kernel and the signal should be uniformly discretized. Therefore, only a few deep neural network architectures were specifically designed to operate on omnidirectional images. These methods in general rely either on icosahedral [28, 33, 78] or equirectangular representations [76, 10, 30, 31] of the omnidirectional images.

In the case of icosahedral representation, the sphere is subdivided into icosahedron at multiple level to mitigate spherical distortion and hence allow standard CNN operations to be applied either on the mesh or on the unfolded icosahedral. Early methods [40, 53] project the sphere onto the six faces of a cube and then apply classical CNN on them. More recent methods consider more complex multi level icosahedron representations and handle the discretization and orientation challenges on the icosahedral manifold [28, 33, 78]. They reparameterize the convolutional kernel as a linear combination of differential operators [28], define orientation-dependent kernels to sample from the triangular faces [33], or use hexagonal filters to address orientation on the unfolded icosahedron mesh [78]. In [17] inverse gnomonic projections are used to render a spherical image to a set of distortion-mitigated, locally-planar image grids that are tangent to a subdivided icosahedron.

Equirectangular projections preserve the spatial relationship of the content, but introduce heavy distortions not suitable for traditional CNNs. [30, 31] proposes to use graph convolutional network where features are inherently invariant to isometric transformations. [68, 69] process equirectangular images with regular convolutions by increasing the kernel size towards the polar regions. Spherical CNNs [10] encode rotation equivariance into the network through spherical convolutions requiring specialized kernels and [76] applied them to detect objects in panoramic images. [11, 70] addresses distortions by warping the planar convolution kernel in a location-dependent manner. [16] proposes mapped convolutions, a more generic solution allowing to perform convolutions on any graph or mesh. The proposed SphSegNet gets inspiration from these methods.

3. Geometry on omnidirectional cameras

A unified model for central omnidirectional cameras was proposed by Geyer and Daniilidis [20]. It represents central

panoramic cameras as a projection onto the surface of a unit sphere $\mathcal{S}$, where the camera coordinate system $\mathcal{C}$ has its origin in the center of $\mathcal{S}$, the z axis is the optical axis pointing towards the viewing direction of the camera and the x and y axes are parallel to that of the omnidirectional image's coordinate system ($\mathcal{I}$) axes. The projection of 3D world points $\mathbf{X} \in \mathcal{W}$ can be performed in two steps: 1) the 3D point $\mathbf{X}$ is projected onto the unit sphere $\mathcal{S}$, obtaining the spherical point $\mathbf{x}_\mathcal{S}$ expressed in $\mathcal{C}$; 2) which is then mapped onto the image plane $\mathcal{I}$ through the camera's internal projection function Φ yielding the image $\mathbf{x} \in \mathcal{I}$ of $\mathbf{X} \in \mathcal{W}$. The relation between $\mathbf{X} \in \mathcal{C}$ and its image $\mathbf{x} \in \mathcal{I}$ in the omnidirectional camera is then given by $\Phi^{-1}(\mathbf{x}) = \mathbf{x}_\mathcal{S} = \frac{\mathbf{X}}{\|\mathbf{X}\|}$.

Although various models for the internal projection function Φ have been proposed [39, 48, 59, 67], knowing Φ (*i.e.* having a calibrated omnidirectional camera) always provides this equivalent *spherical image* by back-projecting the omnidirectional image onto $\mathcal{S}$. Given the $[\mathbf{R}|\mathbf{t}] : \mathcal{W} \to \mathcal{C}$ pose of the camera, the image in the camera of any 3D point $\mathbf{X} \in \mathcal{W}$ can be obtained as follows

$$\mathbf{x}_\mathcal{S} = \frac{\mathbf{R}\mathbf{X} + \mathbf{t}}{\|\mathbf{R}\mathbf{X} + \mathbf{t}\|} \equiv \frac{[\mathbf{R}|\mathbf{t}]\mathbf{X}^\mathcal{W}}{\|[\mathbf{R}|\mathbf{t}]\mathbf{X}^\mathcal{W}\|}. \tag{1}$$

Given a scene plane π, let us denote its normal and distance to the origin by $\mathbf{n}$ and d, respectively. Assuming the 3D coordinate system $\mathcal{P}$ attached to π such that its z axis is defined by the plane normal $\mathbf{n}$, *i.e.* $Z = 0$ for π, a plane point $\mathbf{X} \in \pi$ has homogeneous coordinates in $\mathcal{P} : \mathbf{X}^\mathcal{P} = [X, Y, 0, 1]^\top$. The rotation $\mathbf{S}$ and translation $\mathbf{v}$ denote the *plane pose* [66], which maps $\mathcal{P} \to \mathcal{W}$ such as $\mathbf{X}^\mathcal{W} = [\mathbf{S}|\mathbf{v}]^{\mathcal{P}\to\mathcal{W}}\mathbf{X}^\mathcal{P}$ Thus for $\mathbf{X} \in \pi$, (1) becomes

$$\mathbf{x}_\mathcal{S} \cong [\mathbf{R}|\mathbf{t}]\mathbf{X}^\mathcal{W} = [\mathbf{R}|\mathbf{t}][\mathbf{S}|\mathbf{v}]\mathbf{X}^\mathcal{P} = [\mathbf{R}\mathbf{S}|\mathbf{R}\mathbf{v}+\mathbf{t}]\mathbf{X}^\mathcal{P} \tag{2}$$

Note that $[\mathbf{R}\mathbf{S}|\mathbf{R}\mathbf{v} + \mathbf{t}] \equiv [\mathbf{R}^{\mathcal{P}\to\mathcal{C}}|\mathbf{t}^{\mathcal{P}\to\mathcal{C}}]$ in (2) defines the *plane-camera relative pose* acting from the plane coordinate frame $\mathcal{P}$ to the camera coordinate frame, *i.e.* $[\mathbf{R}^{\mathcal{P}\to\mathcal{C}}|\mathbf{t}^{\mathcal{P}\to\mathcal{C}}] : \mathcal{P} \to \mathcal{C}$.

Planar homographies. The mapping of plane points $\mathbf{X}_\pi \in \pi$ to the camera sphere $\mathcal{S}$ is governed by (2). Hence it is bijective, unless π is going through the camera center, in which case π is invisible. Because of the single viewpoint, planar homographies stay valid for omnidirectional cameras too [37, 19]. The standard planar homography $\mathbf{H}$ is composed up to a scale factor as

$$\mathbf{H} \propto \mathbf{R} + \frac{1}{d}\mathbf{t}\mathbf{n}^\top. \tag{3}$$

The homography transforms the rays as $\mathbf{x}_\mathcal{S} \propto \mathbf{H}\mathbf{X}$, hence the planar homography between the spherical points and plane points is bijective. Using (2), (3), and $Z = 0$, the homography acting between the spherical points $\mathbf{x}_\mathcal{S} \in \mathcal{C}$ and the plane points is obtained as:

$$\mathbf{H}^{\mathcal{P}^*\to\mathcal{C}} \cong \begin{bmatrix}(\mathbf{R}\bar{\mathbf{S}})_1 & (\mathbf{R}\bar{\mathbf{S}})_2 & (\mathbf{R}\mathbf{v}+\mathbf{t})\end{bmatrix} \tag{4}$$

where $\bar{\mathbf{S}}$ denotes the submatrix of $\mathbf{S}$ consisting of its first two columns, and $\mathcal{P}^*$ denotes the 2D $X - Y$ within-plane coordinate system obtained from $\mathcal{P}$.

Rectifying homography. Let us now assume, that our camera sees a 3D plane π with a *low-rank* texture. Considering a 2D texture as a function I_0, it is low-rank if the family of one-dimensional functions $\{I_0(x, y_0) \mid y_0 \in \mathbb{R}\}$ span a finite low-dimensional linear subspace [80]. In practice, only the transformed version I of I_0 is available, which is related by a planar homography $\mathbf{H}$ such that $\mathbf{H}(I_0) = I$. Note however that the image I is not a perfectly transformed versions of the pattern I_0 due to occlusion and noise. Following [80], we model this error with a sparse matrix $\mathbf{E}$, such as $\mathbf{H}^{-1}(I) = \mathbf{I_0} + \mathbf{E}$, where $\mathbf{I_0}$ is the discrete (matrix) representation of I_0. Note that $\mathbf{I_0}$ is the *canonical view* of I, *i.e.* the frontal view of plane π, while $\mathbf{H}^{-1}$ is the *rectifying homography*[1]. which produces this frontal view from the distorted observation I. To estimate $\mathbf{H}$, $\mathbf{I_0}$ and $\mathbf{E}$, Zhang *et al.* [80], propose the **TILT** algorithm that relying on the Augmented Lagrange Multipliers (ALM) solves the following robust rank minimization problem:

$$\begin{aligned} &\min_{\mathbf{I_0},\mathbf{E},\mathbf{H}} \operatorname{rank}(\mathbf{I_0}) + \gamma\|\mathbf{E}\|_0 \\ &\text{s. t.} \quad \mathbf{H}^{-1}(I) = \mathbf{I_0} + \mathbf{E}. \end{aligned} \tag{5}$$

Plane-camera relative pose. Given a low-rank texture I_0, obviously its rank is invariant under any scaling of the function, as well as scaling or translation in the x and y coordinates [80] From a geometric point of view, that means an ambiguity up to a 2D scaling and translation, which can be written as $I_0(x, y) \sim aI_0(s_x x + t_x, s_y y + t_y)$, where $s_x, s_y > 0$. Thus the rectifying homographies of a particular image I form an equivalence class $\mathbf{H}_{\{I\}} = \left\{\forall \mathbf{A}' : \ \mathbf{H}\mathbf{A}' \ \mid \ \mathbf{H}^{-1}(I) = \mathbf{I_0} + \mathbf{E}\right\}$. TILT will find one of these equivalent homographies when run on a particular region, from which one can factorize the camera-plane relative pose using (3). However, this relative pose is acting from a *homography induced* coordinate system $\widetilde{\mathcal{P}}$ on π to the camera coordinate frame $\mathcal{C}$. Indeed, $\mathbf{H}$ is obtained from *intrinsic* visual features of I *without* explicit correspondences between π and I, thus the recovered relative pose is given as $\widetilde{\mathcal{P}} \to \mathcal{C}$ and *not* as $\mathcal{P} \to \mathcal{C}$. To get this, one needs to recover the $\mathcal{P} \to \widetilde{\mathcal{P}}$ map too (*e.g.* via explicit correspondences between the image coordinate frame $\mathcal{I}$ and the plane coordinate frame $\mathcal{P}$). Since both $\widetilde{\mathcal{P}}$ and $\mathcal{P}$ have the z-axis equal to the plane normal $\mathbf{n}$ and a translation along that common z axis is equivalent to a within plane isotropic scaling, only the transformation $\mathcal{P}^* \to \widetilde{\mathcal{P}}^*$ need to be determined. It consists of a 2D rotation θ within π aligning the x and y axes of $\widetilde{\mathcal{P}}$ and $\mathcal{P}$, and the inherent ambiguity in $\mathbf{H}$

[1] Note that for simplicity, we refer to $\mathbf{H}$ as rectifying homography, but strictly speaking the rectification of I is obtained via $\mathbf{H}^{-1}$ and not $\mathbf{H}$!

yielding the special affine transformation $\mathbf{A} : \mathcal{P}^* \rightarrow \widetilde{\mathcal{P}}^*$

$$\mathbf{A} = \begin{bmatrix} s_x \cos\theta & -s_x \sin\theta & t_x \\ s_y \sin\theta & s_y \cos\theta & t_y \\ 0 & 0 & 1 \end{bmatrix} \tag{6}$$

Thus the *plane-camera relative pose* w.r.t. a particular reference plane coordinate system $\mathcal{P}$ can only be obtained by factorizing $(\mathbf{HA})$, for which the knowledge of $\mathbf{A}$ is also needed! However, in practical applications, one often wants to localize a camera w.r.t. a predefined coordinate frame. How to disambiguate $\mathbf{H}$ then? The orientation of the camera $\mathbf{R}$ is well defined only in terms of the z-axis (*i.e.* the angle between the plane normal and the camera optical axis), therefore one can only get a *viewing angle* of the camera w.r.t. the plane. What remains is a rotation around the z-axis in $\mathbf{R}$, which could be fixed by knowing one more direction. Note that the gravity vector provided by an IMU is widely used for such purpose [1, 23, 32]. Assuming the vertical direction in the plane coordinate system $\mathcal{P}$ is known, we can fix the orientation of the axes of $\widetilde{\mathcal{P}}$ and $\mathcal{P}$ (*i.e.* they correspond up to a translation and scaling), and hence recover the full rotation $\mathbf{R} : \mathcal{P} \rightarrow \mathcal{C}$.

To calculate a *relative translation* $\mathbf{t}$, we need one correspondence between $\mathcal{P}^*$ and $\widetilde{\mathcal{P}}^*$, and with a single 2D-3D point correspondence, the *absolute* translation $\mathbf{t}$ can be recovered too, which provides the full plane-camera pose.

Note that in [75] a monocular sparse localization method is proposed, which uses regular 3D points and homographies defined by surfels (small planar patches around such points) to define the camera pose within a world coordinate frame. The proposed low-rank rectifying homography could also be used in such an application, replacing the surfels' generated homography.

4. Spherical low-rank probability map

We have seen that for a region $\mathbf{B}$ we can obtain the rectifying homography $\mathbf{H}$ and its decomposition into the low-rank $\mathbf{B}_0$ (*i.e.* its *canonical view*) and the sparse error $\mathbf{E}$ matrices by running TILT [80] on it. However, TILT works only on *perspective* images so to be able to apply it we need to generate a perspective equivalent of the low-rank spherical region. This can be done by a perspective projection of a spherical region from the centre of the sphere $\mathcal{S}$ onto a tangent plane τ by using *gnomonic projections*.

Gnomonic projections. They allow to define an *implicit* coordinate system on the tangent plane τ as follows: the origin is in the tangent point $\mathbf{x}_{\mathcal{S}}^{\tau} = (\phi_0, \theta_0)$, the x axis is the projection of the great circle corresponding to the latitude coordinate ϕ_0, and the y axis is that of the longitude coordinate θ_0. Obviously, these projections are perpendicular lines on τ. Let $\mathbf{p} = (x_p, y_p)$ denote a point on the tangent plane τ, (ϕ_p, θ_p) are the longitude and latitude coordinate of a spherical point $\mathbf{x}_{\mathcal{S}}$ represented in polar coordinates, and (ϕ_0, θ_0) are the tangent point $\mathbf{x}_{\mathcal{S}}^{\tau}$ coordinates. The gnomonic projection of a spherical point $\mathbf{x}_{\mathcal{S}}$ onto the tangent plane τ is defined by

$$\begin{aligned} x_p &= \frac{\cos\phi_p \sin(\theta_p - \theta_0)}{sin\phi_0 \sin\phi_p - \cos\phi_0 \cos\phi_p \cos(\theta_p - \theta_0)}, \\ y_p &= \frac{\cos\phi_0 \sin\phi_p - \sin\phi_0 \cos\phi_p \cos(\theta_p - \theta_0)}{sin\phi_0 \sin\phi_p - \cos\phi_0 \cos\phi_p \cos(\theta_p - \theta_0)}. \end{aligned} \tag{7}$$

The projection from the tangent plane back to the sphere $\mathcal{S}$ is

$$\begin{aligned} \phi_p &= \sin^{-1}\left(\cos\psi \sin\phi_0 \frac{y_p \sin\psi \cos\phi_0}{\rho}\right), \\ \theta_p &= \theta_0 + \tan^{-1}\left(\frac{x_p \sin\psi}{\rho \cos\phi_0 \cos\psi - \sin\phi_0 \sin\psi}\right) \end{aligned} \tag{8}$$

with $\rho = \sqrt{x_p^2 + y_p^2}$, and $\psi = \tan^{-1}\rho$

The scale of this implicit coordinate system is governed by the Jacobian J of the transformation, hence it is a function of the position $\mathbf{x}_{\mathcal{S}}^{\tau}$. In order to obtain a $m \times m$ sized perspective image on τ, the scaled size of the spherical region will be $J(\mathbf{x}_{\mathcal{S}}^{\tau})m$, and spherical points projected inside will then be used to generate the pixels of the $m \times m$ perspective image via *natural neighbor interpolation*, which uses an area-weighting technique to determine a value for every raster cell from scattered and irregularly spaced data [64]. In order to have a sufficiently detailed $m \times m$ perspective image, we need approximately m^2 pixels of the omnidirectional camera image that maps inside it.

Building the TILT map. Above we have seen how to generate a local perspective equivalent of the neighborhood of a particular pixel of the omnidirectional image. Algorithm 1 summarizes the steps of generation the perspective region[2] for a given tangent point on a sphere.

Running TILT[3] on such a local perspective image provides us the rectifying homography $\mathbf{H}$, a sparse error matrix $\mathbf{E}$, and a low-rank matrix $\mathbf{I}_0$ (see details in Section 3). Therefore we can follow [14] and generate a low-rank likelihood map by considering overlapping sliding spherical windows at multiple scales and predefined steps for an entire omnidirectional image using equirectangular representation. We considered fixed window sizes $l \times l$, with $l \in \{50, 100, 150\}$, and a step size of $l/2$ between neighboring windows. Due to the varying resolution of the omnidirectional image, the actual size of a window at a position $\mathbf{x}_{\mathcal{S}}$ is $J(\mathbf{x}_{\mathcal{S}})l$, and the stepsize changes according to where

[2]Note that, the gnomonic projections yields increased distortion for pixels away from the tangent point $\mathbf{x}_{\mathcal{S}}^{\tau}$, but our purpose is to have a *local* mapping only, providing a perspective image region of size $m \times m$ of a spherical low-rank region, hence the distortion remains negligible.

[3]We used the MATLAB code available from https://people.eecs.berkeley.edu/~yima/matrix-rank/tilt.html.

Algorithm 1 TILT on spherical regions.

Input: The spherical coordinates $\mathbf{c} \in \mathcal{S}$ of a point on the sphere and the desired size $m \times m$ for the perspective region.

Output: The low-rank decomposition of a spherical region centered in $\mathbf{c}$, *i.e.* rectifying homography $\mathbf{H}$, a sparse error matrix $\mathbf{E}$ and a low-rank matrix $\mathbf{I}_0$.

1: Calculate the effective size m_τ of the perspective region such that it contains $\approx m^2$ pixels from the omnidirectional image: $m_\tau = J(\mathbf{c})m$.
2: Calculate the polar coordinates $\mathbf{c}^\tau = (\phi_0, \theta_0)$, and of the points $\mathbf{x}_s = (\phi_s, \theta_s)$ in the spherical region centered in $\mathbf{c}$.
3: Get the *gnomonic projection* $\mathbf{p}_s$ of $\mathbf{x}_s$ on the tangent plane τ using (7).
4: Generate the perspective image $\mathbf{I}$ using the pixels $\mathbf{p_s}$ that are inside the $m_\tau \times m_\tau$ region on τ.
5: Run TILT [80] on $\mathbf{I}$ to get $\mathbf{H}$, $\mathbf{I}_0$, and $\mathbf{E}$.

the window is moving on the surface of $\mathcal{S}$, along longitudes and latitudes (see Figure 1).

To compute the "*low-rankness*" of a particular window w_i^l as a strictly non-negative error (or energy) we follow [14]. Accordingly, $e_i^l = \max(0, \frac{3}{4}(r_i^l + s_i^l + f_i^l - 1))$, where $r_i^l = \text{rank}(\mathbf{I_0}_i^l)/l$, $s_i^l = \|\mathbf{E}_i^l\|_1$, and f_i^l is the residual of the factorization in (5). This score is then used to define a standard exponential distribution $P_i^l = \exp(-e_i^l)$.

Note that a homogeneous region is low-rank (providing a low e_i^l value) but they are useless to estimate a well-defined rectifying homography, therefore, we want to impose $P_i^l = 0$ for homogeneous regions. Homogeneity h_i^l of a window w_i^l is characterized as the percentage of the edge pixels in the window [14]. These discrete probabilities are then propagated over the entire image using wKDE with Gaussian kernels [14]:

$$\mathbf{P}^l = \frac{1}{N^l} \sum_{i^l = N^l} \exp(-e_i^l)\delta(h_i^l > \tau)\mathcal{G}(w_i^l, \sigma^l) \quad (9)$$

where $\delta()$ is the Kronecker delta and the homogeneity threshold τ is set to $\tau = 0.04$. N^l denotes the number of windows in an image at level l, w_i^l the sliding windows and σ^l is function of the window size l. Finally, the probability maps obtained at different levels are averaged to obtain the final low-rank likelihood map $\mathbf{P}$ (see examples in Figure 4).

5. Spherical low-rankness detection network

Obtaining the probability map as described in Section 4, that we will call **TILT map**, is very costly[4] because for every sliding pattern on the sphere at each scale we need to first project the region on the tangent plane and then run a computationally costly optimization (TILT) on the projected region.

[4]Multi-scale, sliding-window TILT on MATLAB takes about 15-20 minutes per image.

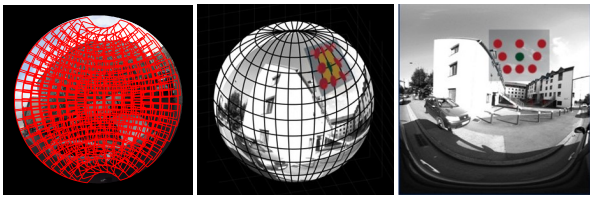

Figure 1. Spherical sliding windows (left); projecting the spherical neighborhood to the tangent plane (middle); and the corresponding locations in the equirectangular image (right).

Therefore, similarly to [14] we propose to use a set of TILT maps to train a network for predicting such low-rankness probability maps. While these maps are only approximated probability distributions of the low-rankness in the spherical image and cannot be considered as perfect ground truth, we rely on the generalization power of the deep network to learn to recognize the implicit structure of low-rank non-homogeneous regions on the spherical image.

One option would be to train a standard CNN network as in [14] with the equirectangular images/TILT maps without modification (see Section 6). However, such model is geometrically incorrect as the regular CNN grid does not take the discontinuities and distortions at the polar regions into account. To address the distortions in the equirectangular images, inspired by [11, 70, 16], we proposes to lift local CNN operations (*e.g.* convolution, pooling) from the equirectangular image to the sphere where the image is represented without distortions (illustrated in Figure 1).

In particular, we extends the pytorch implementation of [24], where the SphereNet convolution is implemented using an intermediate image, that we refer to as *pivot* image on which regular 2D convolutions are applied (see Figure 2). The mapping from the input equirectangular image to the pivot image, we refer to as *grid*, similarly to the mapping function in [11, 70, 16], defines where the $k \times k$ neighborhood on the sphere should be sampled in the equirectangular image on which the convolution is applied. More precisely, each anchor pixel in the equirectangular image (green dot in Figure 1) is lifted first on the sphere with inverse gnomonic projections (8) and its $k \times k$ neighbourhood (red dots) are project to the equirectangular image, by passing through the tangent image, with gnomonic projections (7). The pivot image is the collection of these values stored as image blocks on which we perform the regular CNN operations. Note that by construction, the output of this operation is also an equirectangular image (see Figure 2).

The combined layer of generating the intermediate pivot image and the final equirectangular output we call SphConv2D, SphAvgPool2D or SphMaxPool2D, according to the CNN operation applied on the pivot image. While these operators are sufficient for equirectangular image classification (as in [24]), the extension of the low-rankness detector [14], requires also upscaling operators. These oper-

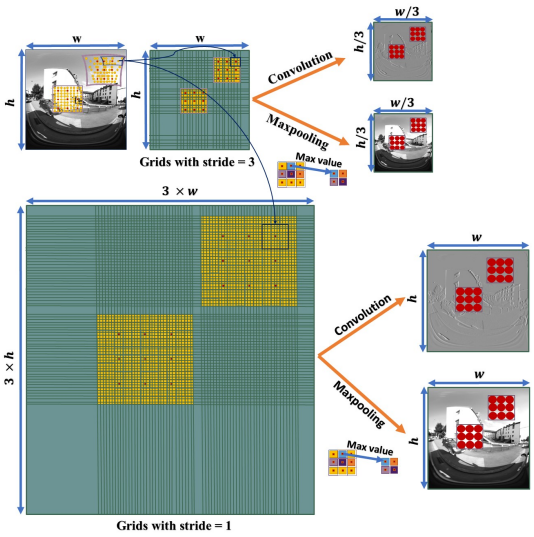

Figure 2. SphConv2D/SphAvgPool2D operators for kernel size $k = 3$ and strides 1 and 3 showing the grids to be filled to generate the pivot image on which the standard CNN filter is applied.

ations cannot be solved in the same manner as convolution or pooling. especially max-unpooling since it requires retaining the positions where the maximum values were collected, which in our case correspond to positions in the pivot image. Therefore, we define SphMaxUnPool2D and SphMaxPool2D layers, where first the conventional CNN unpooling/upscaling is applied to the input layer to obtain a pivot image and then the *inverse* of the grid is used to get an equirectangular output. These operations can be interpreted as inverting the arrows in Figure 2.

With these operators we propose **SphSegNet**, an adaption of the model SegNet [2] used in [14] for predicting low-rankness for an omnidirectional image. We made two major modifications: we used less downscaling/upscaling blocks (3 instead of 5), and more importantly we replaced the regular CNN operators by the ones defined above. Figure 3 illustrates the proposed architecture. Note that the most costly operation in the network is the computation of the sampling locations to build the grid. Fortunately, the grid mapping is uniquely defined by (7) and (8) for a given input size k an s. Therefore, these grids (7 in total) can be pre-computed and stored beforehand. The inversion of the grid is low cost operation performed online.

Training. Our aim is to obtain the output feature map $\mathbf{F}$ of the network to be similar to $\mathbf{P}$ (TILT map) in a probabilistic closeness sense that we measure by the Kullback-Leibler (KL) divergence (as in [14])

$$D(\mathbf{P}||\mathbf{F}) = \sum_{(i,j)\in I} \widehat{\mathbf{P}}(i,j) \log \frac{\widehat{\mathbf{P}}(i,j)}{\widehat{\mathbf{F}}(i,j)}, \qquad (10)$$

where $\widehat{\mathbf{P}}$ and $\widehat{\mathbf{F}}$ are normalized likelihood maps such that

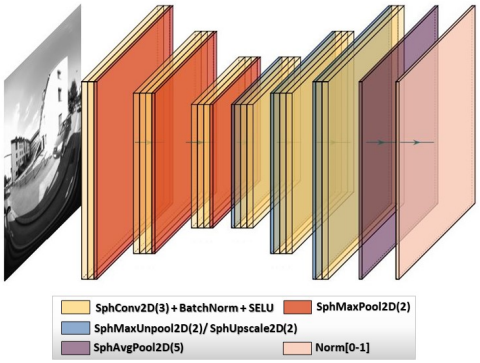

Figure 3. SphSegNet. SphConv2D, SphMaxPool2D, SphMaxUnpool2D, and SphUpscale2D are using the intermediate pivot layer. Here we show them as a single block (that contains both the mapping to the pivot layer and the classical CNN operation on the pivot layer as shown in Figure 2). Numbers in the parenthesis refer to the kernel size. Norm[0-1] refers to normalizing the value of the output layer between 0 and 1 (probability map).

the sum of all values is 1, making the maps equivalent to probability distributions conditioned on the given image. However, we observed that using the mean-square loss between $\mathbf{F}$ and $\mathbf{P}$ helps, in particular at the beginning of the training. Therefore, in contrast to [14], to train the network, we use a weighted combination of the KL-divergence loss (10) and the MSE loss with a dynamic weight pair $(1 - w, w)$, were $w \in [0, 1]$ is higher for MSE at the beginning and progressively decreasing, while the weight for the KL-loss increasing.

6. Experimental results

Datasets. We used the **SILDA** [4] dataset, containing omnidirectional images captured with wide-angle fisheye lenses as training set and **KITTI360** [73], large scale urban dataset containing rich sensory information and full annotations, as test set[5] In particular, KITTI360 contains a Lidar point cloud and omnidirectional images which allows us – together with the ground truth camera poses – to evaluate the camera-plane relative poses computed from detected low-rank regions in an omnidirectional dataset. To generate the TILT maps, as described in Section 4, color images were first transformed into grayscale, second, into equirectangular representation using gnomonic projections (7) (see examples in Figure 4).

Low-rank predicted maps. We used the SILDA images (3500 random samples) and the corresponding TILT maps to train both the proposed SphSegNet denoted as **SphSN-S**, and a corresponding ablative model, SegNet, denoted as **SN-S** that was obtained by replacing the spherical operators by classical ones. Both models were tested on the equirect-

[5]Due to the high cost of generating TILT maps, we selected a random subset of 3500 images from SILDA and 450 images from KITTI360.

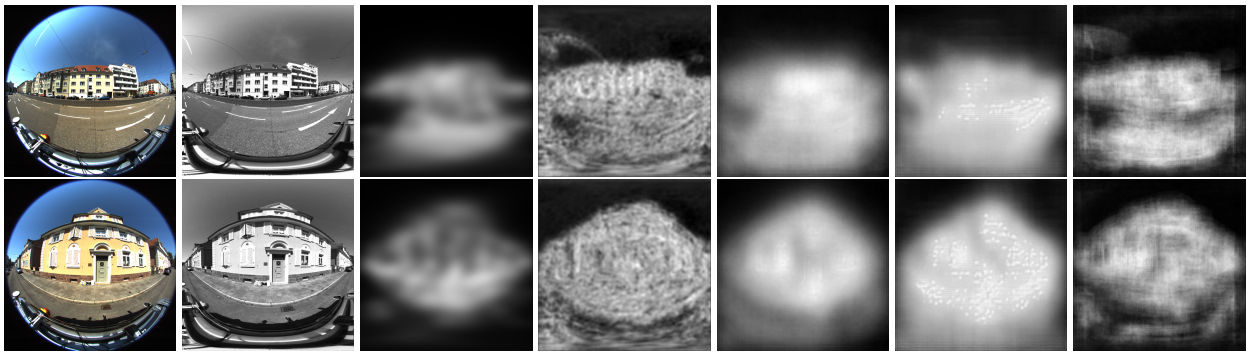

Figure 4. Example result probability maps on the KITTI360 omni dataset. Here we show the original omni image, the equirectangular representation, the TILT probability map, and the predictions obtained with the four deep networks.

angular KITTI360 images (second column in Figure 4). We selected 700 random frames for which we also computed the TILT maps. In addition, for comparison, we also considered two standard CNN models obtained by training SegNet[6] on Aachen dataset [57] used also in [14] and on Cityscapes [12] which is a autonomous driving dataset similar to KITTI360 and SILDA. The two last models referred to as **SN-A** and **SN-CS** respectively, were trained with perspective images and corresponding TILT maps, while **SphSN-S** and **SN-S** were trained with the equirectangular SILDA images and TILT maps obtained with the spherical representation described in Section 4.

In Figure 4 (last 4 rows) we show the results for the above mentioned four models tested on the equirectangular images of KITTI360 (second column). For a first quantitative evaluation, we computed KL divergence between the predicted maps and the corresponding TILT maps (third column). The average KL values for each model are shown in Table 1 (first row). We observe lower KL values for the SegNet models (SN-*) compared to SphSN-S, probably due to the fact that similarly to the TILT map they are smoother than the latter. Note however that this score shows how close the predicted map is to the TILT map, which is not a ground truth. Furthermore, we might obtain a large KL also if we have a good predicted map, but a poor TILT map[7].

Therefore, we propose a better evaluation protocol for these maps, where we factorize relative plane-camera poses from low-rank regions detected at local maxima in the map and compare them with GT poses deduced from the known camera pose and the 3D plane position.

Detecting low-rank region. To obtain low-rank regions from the predicted maps, we search for local maxima at different scales such that the average probability in the corresponding spherical window is higher than 0.5. The second row in Table 1 shows the average number of detected regions per image for each of the methods, including the TILT maps. Note that some of these maxima are on edges between two quasi-homogeneous regions (see *e.g.* in Figure 6), which indeed is a low-rank region but not suitable for pose estimation as the corresponding region is not planar. These maxima can be eliminated by *e.g.* a plane detector method[8] such as PlaneNet [35].

	SphSN-S	**SN-S**	**SN-A**	**SN-CS**	**TILT**
KL div	0.54	0.36	**0.19**	0.31	-
Nb reg.	6.96	5.33	4.84	6.17	7.94
Rot err	**9.85**	10.51	14.87	13.12	11.10
Tr err	**9.29**	9.73	11.69	10.55	10.01

Table 1. **First row**: We show the average KL divergence between the TILT map and the predicted maps obtained for the 700 images of the KITTI360 dataset. **Second row**: The average number of detected low-rank regions with local maxima search in the corresponding prediction map with the average value within the region being above 0.5. **Third and fourth row**: The median over the test set of the rotation (ϵ_R) and translation (ϵ_t) errors obtained with the best regions detected in each image considering the different low-rankness maps.

Homography estimation examples. For each detected spherical regions we run TILT on the corresponding tangent region obtained as described in Section 4 and in particular in Algorithm 1 to obtain the rectifying homography and the canonical view. In Figure 6 we show an example of the detected low-rank spherical region, the corresponding tangent region which is a perspective image, the canonical view obtained with the rectifying homography. We also show in comparison the canonical view obtained via the ground truth homography composed from the GT camera pose and the 3D plane parameters as in (3).

Pose estimation results. Using the estimated rectifying homography, we can factorize the plane-camera pose [66] as described in Section 3. Of course, the ambiguities in the

[6]Where we kept the more complex architecture, *i.e.* 5 downscaling/upscaling blocks as in [14] instead of 3 used in SN-S.

[7]As in [14], we also observed such cases for KITTI360.

[8]Which in our case should be first adapted to omnidirectional images using for example the framework we proposed in this paper.

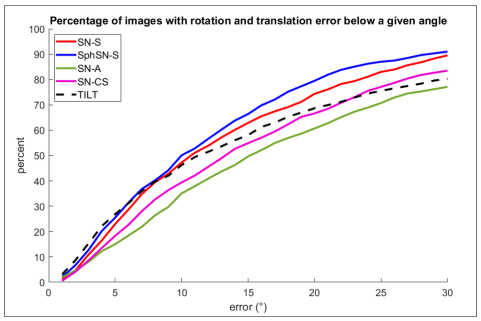

Figure 5. Percentages of images with both the rotation (ϵ_t) and translation (ϵ_t) error below a given threshold, varying the threshold from 1° to 30°.

factorized pose have to be solved in order to be able to directly measure the error in the estimated pose w.r.t. the GT pose. For this purpose, we make use of the 3D point cloud provided in the KITTI360 dataset and determine the 3D plane parameters. This allows us to fix the ambiguity in (6) such that our estimated pose and the GT pose are directly comparable. We then measure the rotation error as $\epsilon_R = \angle\hat{\mathbf{R}}\mathbf{R}^\top$ with $\hat{\mathbf{R}}$ being the rotation factorized from the estimated rectifying homography while $\mathbf{R}$ is the GT rotation. As for the translation, one can only get a unit length translation from the homography factorization as discussed in Section 3 and [66]. Hence the translation error ϵ_t is expressed in terms of the unit translation vector's angle, *i.e.* $\epsilon_t = \arccos \hat{\mathbf{t}}\mathbf{t}^\top$ with $\hat{\mathbf{t}}$ being the factorized unit translation and $\mathbf{t}$ the GT unit translation vector.

For each local maxima in the predicted maps, we consider regions at 5 different scales, run TILT on the corresponding tangent regions, factorize the rectifying homograph to obtain $(\hat{\mathbf{R}}, \hat{\mathbf{t}})$ and compute the rotation (ϵ_R) and translation (ϵ_t) errors. Note that our goal is to compare different low-rankness maps, therefore we only retain the best factorized pose for each images[9]. In the last two rows of Table 1, we show the median of these errors over the KITTI360 test set for each method. For comparison we also considered the TILT maps and proceeded the same. In addition in Figure 5 we plot the percentage of images for which both the rotation (ϵ_R) respectively the translation (ϵ_t) error is below a certain degree up given a threshold from 1° to 30°. We can observe that best results are found with the **SphSN-S** model, which even outperformed the factorized poses accuracy obtained with the regions detected on the TILT map (except very high accuracies). In Figure 6 we show an example pose estimation factorized from such a rectifying homography.

[9]TILT is rather unstable, therefore, considering regions at multiple scales (5) at the selected local maxima positions (see *e.g.* Figure 6), and selecting the best pose gives for each model equal chances to get a good pose factorization. Obviously deployment in real applications would require to select the best region or to robustly combine several detections. This is out of the scope of this paper and is subject of future research.

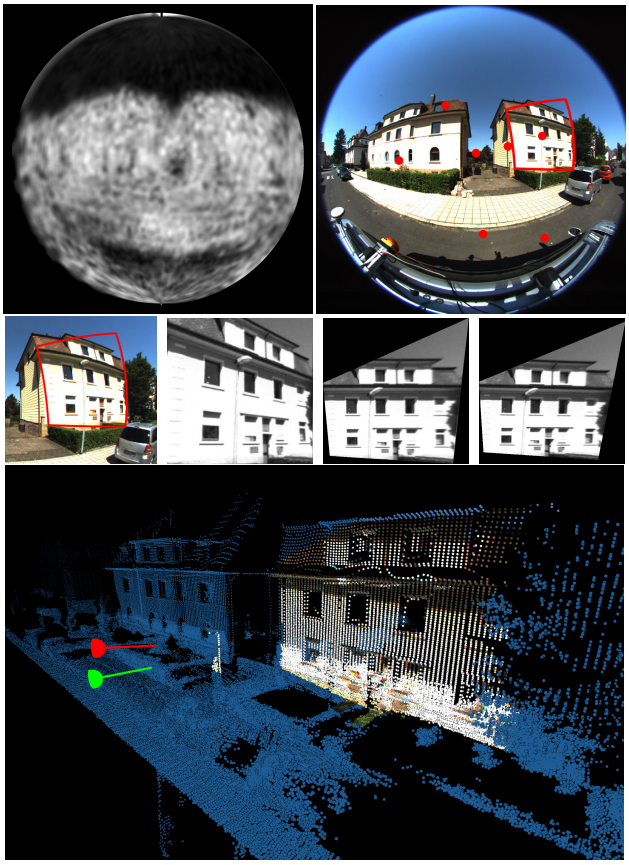

Figure 6. **Top:** 1) The predicted low-rankness probability map, 2) the centers of the detected local maxima regions and the low-rank spherical region yielding the best relative-pose. **Middle:** The detected low-rank spherical region used for TILT: 1) zoom on the region on the spherical image (as red spherical rectangle), 2) the projected square region on the tangent plane, 3) the rectified region with TILT, 4) the ground truth rectification. **Bottom:** Pose factorized from the rectifying homography: The GT camera is shown as a green half-sphere, the estimated camera as a red one from where the color omnidirectional region is backprojected to the 3D point cloud. The rotation error of the estimated pose is $\epsilon_R = 6.8^\circ$, the translation error is $\epsilon_t = 6.7^\circ$.

7. Conclusion

We proposed a method to detect low-rank regions in equirectangular representation of omnidirectional images taking into account explicitly the spherical distortion induced by this representation. To achieve this, we first adapted the perspective-only TILT algorithm and generated low-rankness likelihood maps. Then, we used these maps to train SphSegNet, a deep network with layers adapted to equirectangular format, to predict a low-rankness likelihood map for omnidirectional images. Finally, we show that from the rectifying homography obtained with TILT on low-rank regions detected at local maxima of these maps we can factorize out the camera-plane pose up to certain ambiguities.

References

[1] Hichem Abdellali and Zoltan Kato. Absolute and relative pose estimation of a multi-view camera system using 2d-3d line pairs and vertical direction. In *Proceedings of International Conference on Digital Image Computing: Techniques and Applications*, pages 1–8, Canberra, Australia, Dec. 2018. IEEE. 4

[2] Vijay Badrinarayanan, Alex Kendall, and Roberto Cipolla. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 39(12):2481–2495, 2017. 6

[3] Simon Baker and Shree K. Nayar. A theory of single-viewpoint catadioptric image formation. *International Journal of Computer Vision*, 35(2):175–196, 1999. 2

[4] Vasileios Balntas, Duncan Frost, Rigas Kouskouridas, Axel Barroso-Laguna, Arjang Talattof, Huub Heijnen, and Krystian Mikolajczyk. Silda: Scape imperial localisation dataset, 2019. 6

[5] Vassileios Balntas, Karel Lenc, Andrea Vedaldi, and Krystian Mikolajczyk. Hpatches: A benchmark and evaluation of handcrafted and learned local descriptors. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 1

[6] Guillaume Caron, Eric Marchand, and El Mustapha Mouaddib. Tracking planes in omnidirectional stereovision. In *IEEE International Conference on Robotics and Automation*, pages 6306–6311. IEEE, 2011. 2

[7] D. Caruso, J. Engel, and D. Cremers. Large-scale direct slam for omnidirectional cameras. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots Syst (IROS)*, 2015. 1

[8] David M. Chen, Georges Baatz, Kevin Köser, Sam S. Tsai, Ramakrishna Vedantham, Timo Pylvänäinen, Kimmo Roimela, Xin Chen, Jeff Bach, Marc Pollefeys, Bernd Girod, and Radek Grzeszczuk. City-scale landmark identification on mobile devices. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2011. 2

[9] X. Chen and J. Yang. Towards monitoring human activities using an omnidirectional camera. In *International Conference on Multimodal Interfaces*, 2002. 1

[10] Taco S. Cohen, Mario Geiger, Jonas Köhler, and Max Welling. Spherical CNNs. In *International Conference on Learning Representations (ICLR)*, 2018. 2

[11] Benjamin Coors, Alexandru Paul Condurache, and Andreas Geiger. SphereNet: Learning Spherical Representations for Detection and Classification in Omnidirectional Images. In *European Conference on Computer Vision (ECCV)*, 2018. 2, 5

[12] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The Cityscapes Dataset for Semantic Urban Scene Understanding. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 7

[13] Gabriela Csurka, Christopher R. Dance, and Martin Humenberger. From handcrafted to deep local features. *arXiv preprint arXiv:1807.10254*, 2018. 1

[14] Gabriela Csurka, Zoltan Kato, Andor Juhasz, and Martin Humenberger. Estimating low-rank region likelihood maps. In *Proceedings of International Conference on Computer Vision and Pattern Recognition*, pages 1–10, Seattle, Washington, USA, June 2020. IEEE. 1, 2, 4, 5, 6, 7

[15] Grégoire Dupont de Dinechin and Alexis Paljic. Cinematic virtual reality with motion parallax from a single monoscopic omnidirectional image. In *Digital HERITAGE*, 2018. 1

[16] Marc Eder, True Price, Thanh Vu, Akash Bapat, and Jan-Michael Frahm. Mapped Convolutions. *arXiv preprint arXiv:1906.11096*, 2019. 2, 5

[17] Marc Eder, Mykhailo Shvets, John Lim, and Jan-Michael Frahm. Tangent Images for Mitigating Spherical Distortion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12426–12434, 2020. 2

[18] Robert Frohlich and Zoltan Kato. Simultaneous multi-view relative pose estimation and 3D reconstruction from planar regions. In Gustavo Carneiro and Shaodi You, editors, *Proceedings of ACCV Workshop on Advanced Machine Vision for Real-life and Industrially Relevant Applications*, volume 11367 of *Lecture Notes in Computer Science*, pages 467–483, Perth, Australia, Dec. 2018. Springer. 1

[19] Robert Frohlich, Levente Tamás, and Zoltan Kato. *Handling Uncertainty and Networked Structure in Robot Control*, volume 42 of *Studies in Systems, Decision and Control*, chapter Homography Estimation Between Omnidirectional Cameras Without Point Correspondences, pages 129–151. Springer, Feb. 2016. Chapter 6. 2, 3

[20] C. Geyer and K. Daniilidis. A unifying theory for central panoramic systems. In *European Conference on Computer Vision (ECCV)*, pages 445–462, 2000. 2

[21] D. Gutierrez, A. Rituerto, J.M.M. Montiel, and J.J. Guerrero. Adapting a real-time monocular visual SLAM from conventional to omnidirectional cameras. In *Computer Vision Workshops (ICCV Workshops), 2011 IEEE International Conference on*, pages 343–350, Nov 2011. 2

[22] James Hays, Marius Leordeanu, Alexei A. Efros, and Yanxi Liu. Discovering texture regularity as a higher-order correspondence problem. In *European Conference on Computer Vision (ECCV)*, 2006. 2

[23] Nora Horanyi and Zoltan Kato. Generalized pose estimation from line correspondences with known vertical direction. In *Proceedings of International Conference on 3D Vision*, pages 1–10, Qingdao, China, Oct. 2017. IEEE. 4

[24] Chi Wei Hsiao and Cheng Sun. Spherenet-pytorch. https://github.com/ChiWeiHsiao/SphereNet-pytorch, Oct. 2018. 5

[25] H.N. Hu, Y.C. Lin, M.Y. Liu, H.T. Cheng, Y.J. Chang, and M. Sun. Deep 360 pilot: Learning a deep agent for piloting through 360 sports video. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 1

[26] Sunghoon Im, Hyowon Ha, cois Rameau, Fran and Hae-Gon Jeon. All-Around Depth from Small Motion with a Spherical

Panoramic Camera. In *European Conference on Computer Vision (ECCV)*, 2016. 1

[27] Joel Janai, Fatma Guney, Aseem Behl, and Andreas Geiger. Computer Vision for Autonomous Vehicles: Problems, Datasets and State-of-the-Art. *Foundations and Trends ® Computer Graphics and Vision*, 1–3(12), 2017. 1

[28] Chiyu Max Jiang, Jingwei Huang, Karthik Kashinath, Prabhat, Philip Marcus, and Matthias Niessner. Spherical CNNs on Unstructured Grids. In *International Conference on Learning Representations*, 2019. 2

[29] Juho Kannala and Sami S. Brandt. A generic camera model and calibration method for conventional, wide-angle, and fish-eye lenses. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(8):1335–1340, 2006. 2

[30] Renata Khasanova and Pascal Frossard. Graph-based Classification of Omnidirectional Images. In *ICCV Workshops* , 2017. 2

[31] Renata Khasanova and Pascal Frossard. Graph-based isometry invariant representation learning. In *International Conference on Machine Learning (ICML)*, 2017. 2

[32] Louis Lecrosnier, Remi Boutteau, Pascal Vasseur, Xavier Savatier, and Fridrich Fraundorfer. Camera pose estimation based on PnL with a known vertical direction. *IEEE Robotics and Automation Letters*, 4(4):3852–3859, 2019. 4

[33] Yeonkun Lee, Jaeseok Jeong, Jongseob Yun, Wonjune Cho, and Kuk-Jin Yoon. SpherePHD: Applying CNNs on a Spherical PolyHeDron Representation of 360deg Images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019. 2

[34] Jianfeng Li, Xiaowei Wang, and Shigang Li. Spherical-Model-Based SLAM on Full-View Images for Indoor Environments. *Applied Science*, 8(2268), 2018. 1

[35] Chen Liu, Jimei Yang, Duygu Ceylan, Ersin Yumer, and Yasutaka Furukawa. PlaneNet: Piece-wise Planar Reconstruction from a Single RGB Image. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 7

[36] Ameesh Makadia, Christopher Geyer, and Kostas Daniilidis. Correspondence-free structure from motion. *International Journal of Computer Vision*, 75(3):311–327, Dec. 2007. 2

[37] C. Mei, Selim Benhimane, E. Malis, and Patrick Rives. Efficient homography-based tracking and 3-D reconstruction for single-viewpoint sensors. *Robotics, IEEE Transactions on*, 24(6):1352–1364, Dec. 2008. 2, 3

[38] C. Mei and P. Rives. Single view point omnidirectional camera calibration from planar grids. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 3945–3950, Roma, Italy, April 2007. 2

[39] Branislav Mičušík and Tomáš Pajdla. Para-catadioptric camera auto-calibration from epipolar geometry. In Ki-Sang Hong and Zhengyou Zhang, editors, *Proc. of the Asian Conference on Computer Vision (ACCV)*, volume 2, pages 748–753, Seoul, Korea South, January 2004. Asian Federation of Computer Vision Societies. 2, 3

[40] Rafael Monroy, Sebastian Lutz, Tejo Chalasani, and Aljosa Smolic. SalNet360: Saliency Maps for Omni-directional Images with CNN. *Signal Processing: Image Communication*, 69:26 – 34, 2018. Salient360: Visual attention modeling for 360° Images. 2

[41] Pierre Moulon, Pascal Monasse, and Renaud Marlet. Global fusion of relative motions for robust, accurate and scalable structure from motion. In *IEEE International Conference on Computer Vision (ICCV)*, 2013. 1

[42] Shree K. Nayar. Catadioptric omnidirectional camera. In *Proceedings of the 1997 Conference on Computer Vision and Pattern Recognition (CVPR '97)*, CVPR '97, pages 482–488, Washington, USA, 1997. IEEE Computer Society. 2

[43] Minwoo Park, Kyle Brocklehurst, Robert T. Collins, and Yanxi Liu. Deformed lattice detection in real-world images using mean-shift belief propagation. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 31(10):804—-1816, 2009. 2

[44] Yigang Peng, Arvind Ganesh, John Wright, Wenli Xu, and Yi Ma. RASL: Robust Alignment by Sparse and Low-rank decomposition for linearly correlated images. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 34(11):2233–2246, 2012. 1, 2

[45] V. Popov and A. Gorbenko. Building the panoramic image for mobile robot localization. In *Proceedings of the Applied Mechanics and Materials*, 2013. 1

[46] James Pritts, Zuzana Kukelova, Victor Larsson, and Ondřej Chum. Rectification from radially-distorted scales. In *Asian Conference on Computer Vision (ACCV)*, 2018. 1

[47] James Pritts, Jiří Matas, and Ondřej Chum. Detection, rectification and segmentation of coplanar repeated patterns. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014. 2

[48] Luis Puig, Yalin Bastanlar, Peter Sturm, Josechu Guerrero, and Joao Barreto. Calibration of Central Catadioptric Cameras Using a DLT-Like Approach. *International Journal of Computer Vision*, 93(1):101–114, May 2011. 3

[49] Luis Puig and José Jesús Guerrero. Scale space for central catadioptric systems: Towards a generic camera feature extractor. In *Proceedings of International Conference on Computer Vision*, pages 1599–1606. IEEE, 2011. 2

[50] Luis Puig and José Jesús Guerrero. *Omnidirectional Vision Systems: Calibration, Feature Extraction and 3D Information*. Springer, 2013. 2

[51] L. Ran, Y. Zhang, Q. Zhang, and T.: Yang. Convolutional neural network-based robot navigation using uncalibrated spherical images. *Sensors*, 17(6), 2017. 1

[52] A. Rituerto, L. Puig, and J. Guerrero. Visual slam with an omnidirectional camera. In *International Conference on Pattern Recognition*, 2010. 1

[53] Manuel Ruder, Alexey Dosovitskiy, and Thomas Brox. Artistic Style Transfer for Videos and Spherical Images. *International Journal of Computer Vision*, 126(11):1199–1219, 2018. 2

[54] Zsolt Santa and Zoltan Kato. Pose estimation of ad-hoc mobile camera networks. In *Proceedings of International Conference on Digital Image Computing: Techniques and Applications*, pages 1–8, Hobart, Tasmania, Australia, Nov. 2013. IEEE. 1

[55] Paul-Edouard Sarlin, Cesar Cadena, Roland Siegwart, and Marcin Dymczyk. From coarse to fine: Robust hierarchical localization at large scale. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019. 1

[56] Torsten Sattler, Bastian Leibe, and Leif Kobbelt. Efficient & effective prioritized matching for large-scale image-based localization. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 39(9):1744–1756, 2016. 1

[57] Torsten Sattler, Will Maddern, Carl Toft, Torii Akihiko, Lars Hammarstrand, Erik Stenborg, Daniel Safari, Masatoshi Okutomi, Marc Pollefeys, Josef Sivic, Fredrik Kahl, and Tomáš Pajdla. Benchmarking 6dof outdoor visual localization in changing conditions. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 7

[58] Torsten Sattler, Akihiko Torii, Josef Sivic, Marc Pollefeys, Hajime Taira, Masatoshi Okutomi, and Tomas Pajdla. Are large-scale 3D models really necessary for accurate visual localization? In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 1

[59] Davide Scaramuzza, Agostino Martinelli, and Roland Siegwart. A flexible technique for accurate omnidirectional camera calibration and structure from motion. In *Proceedings of the Fourth IEEE International Conference on Computer Vision Systems*, ICVS-06, pages 45–51, Washington, USA, 2006. IEEE Computer Society. 1, 3

[60] Davide Scaramuzza, Agostino Martinelli, and Roland Siegwart. A toolbox for easily calibrating omnidirectional cameras. In *IEEE/RSJ International Conference on Intelligent Robots*, pages 5695–5701, Bejing, October 9–15 2006. IEEE. 2

[61] Grant Schindler, Panchapagesan Krishnamurthy, Roberto Lublinerman, Yanxi Liu, and Frank Dellaert. Detecting and matching repeated patterns for automatic geo-tagging in urban environments. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2008. 2

[62] Johannes L. Schönberger and Jan-Michael Frahm. Structure-from-motion revisited. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 1

[63] Johannes L. Schonberger, Hans Hardmeier, Torsten Sattler, and Marc Pollefeys. Comparative evaluation of hand-crafted and learned local features. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017. 1

[64] Robin Sibson. *Interpreting Multivariate Data*, chapter A brief description of natural neighbor interpolation, pages 21–36. John Wiley & Sons, 1981. 4

[65] Pablo Sprechmann, Alex M. Bronstein, and Guillermo Sapiro. Learning efficient sparse and low rank models. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 37(9):1821–1833, 2015. 1

[66] Peter Sturm. Algorithms for plane-based pose estimation. In *Proceedings of International Conference on Computer Vision and Pattern Recognition*, volume 1, pages 706–711, June 2000. 3, 7, 8

[67] Peter Sturm, Srikumar Ramalingam, Jean-Philippe Tardif, Simone Gasparini, and Joao Barreto. Camera Models and Fundamental Concepts Used in Geometric Computer Vision. *Foundations and Trends in Computer Graphics and Vision*, 6(1-2):1–183, Jan. 2011. 3

[68] Yu-Chuan Su and Kristen Grauman. Learning Spherical Convolution for Fast Features from 360Imagery. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30, pages 529–539. Curran Associates, Inc., 2017. 2

[69] Yu-Chuan Su and Kristen Grauman. Kernel Transformer Networks for Compact Spherical Convolution. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 9442–9451, 2019. 2

[70] Keisuke Tateno, Nassir Navab, and Federico Tombari. Distortion-Aware Convolutional Filters for Dense Prediction in Panoramic Images. In *Proceedings of the European Conference on Computer Vision (ECCV)*, September 2018. 2, 5

[71] Akihiko Torii, Josef Sivic, Masatoshi Okutomi, and Tomas Pajdla. Visual place recognition with repetitive structures. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 37(11):2346–2359, 2015. 2

[72] Changchang Wu, Jan-Michael Frahm, and Marc Pollefeys. Repetition-based dense single-view reconstruction. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2011. 2

[73] Jun Xie, Martin Kiefel, Ming-Ting Sun, and Andreas Geiger. Semantic instance annotation of street scenes by 3d to 2d label transfer. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 6

[74] Zhi Yan, Li Sun, Tomas Krajnik, and Yassine Ruichek. EU Long-term Dataset with Multiple Sensors for Autonomous Driving. *CoRR*, (arXiv:1909.03330), 2019. 1

[75] Haoyang Ye, Huaiyang Huang, and Ming Liu. Monocular direct sparse localization in a prior 3D surfel map. In *International Conference on Robotics and Automation*, pages 1–7, Paris, France, June 2020. IEEE, IEEE. 4

[76] Dawen Yu and Shunping Ji. Grid Based Spherical CNN for Object Detection from Panoramic Images. *Sensors*, 19(11):2622, 2019. 2

[77] Ekim Yurtsever, Jacob Lambert, Alexander Carballo, and Kazuya Takeda. A Survey of Autonomous Driving: Common Practices and Emerging Technologies. *IEEE Access*, 8, 2020. 1

[78] Chao Zhang, Stephan Liwicki, William Smith, and Roberto Cipolla. Orientation-aware Semantic Segmentation on Icosahedron Spheres. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3533–3541, 2019. 2

[79] T. Zhang, X. Liu, T Mei, G. . Tang, B. Li, and X. Wang. A novel platform for simulation and evaluation of intelligent behavior of driverless vehicle. In *Proceedings of the 2008 IEEE International Conference on Vehicular Electronics and Safety (ICVES)*, 2008. 1

[80] Zhengdong Zhang, Arvind Ganesh, Xiao Liang, and Yi Ma. TILT: Transform invariant low-rank textures. *International Journal of Computer Vision*, 99:1–24, jan 2012. 1, 2, 3, 4, 5

[81] Z. Zhang, H. Rebecq, C. Forster, and D. Scaramuzza. Benefit of large field-of-view cameras for visual odometry. In *International Conference on Robotics and Automation (ICRA)*, pages 801–808, 2016. 1