

Scale Invariant Feature Transform on the Sphere: Theory and Applications

Javier Cruz-Mota^{*} Iva Bogdanova[†] Benoît Paquier[‡]
Michel Bierlaire^{*} Jean-Philippe Thiran[‡]

April 26, 2009

Report TRANSP-OR 090426
Transport and Mobility Laboratory
Ecole Polytechnique Fédérale de Lausanne
`transp-or.epfl.ch`

^{*}Transp-OR, École Polytechnique Fédérale de Lausanne, CH-1015 Lausanne, Switzerland, javier.cruz@epfl.ch, michel.bierlaire@epfl.ch

[†]PaRLab, École Polytechnique Fédérale de Lausanne, Rue A. L. Breguet 2, CH-2000 Neuchâtel, Switzerland, iva.bogdanova@epfl.ch

[‡]LTS5, École Polytechnique Fédérale de Lausanne, CH-1015 Lausanne, Switzerland, benoit.paquier@epfl.ch, jp.thiran@epfl.ch

Abstract

A SIFT algorithm in spherical coordinates for omnidirectional images is proposed. This algorithm can generate two types of local descriptors, Local Spherical Descriptors and Local Planar Descriptors. With the first ones, point matching between omnidirectional images can be performed, and with the second ones, the same matching process can be done but between omnidirectional and planar images. Furthermore, a planar to spherical transformation is introduced and an algorithm for its estimation is given. This transformation allows to extract objects from an omnidirectional image given their SIFT descriptors in a planar image. Several experiments, confirming the good performance of the system, are conducted.

1 Introduction

Omnidirectional vision has become a popular topic in computer vision. One of its main benefits is that one omnidirectional camera can cover 360° around it. As with conventional (planar) images, image matching is a main aspect of many computer vision problems involving omnidirectional images, although it has not been widely studied yet for this kind of cameras. Usually, techniques designed for planar images are applied on omnidirectional images, as for example on panoramic images, i.e. omnidirectional images mapped on a cylinder (Yuen and MacDonald, 2005; Bur et al., 2006). This is not mathematically correct. Even if locally those algorithms are still valid, as soon as bigger regions of the image are considered, it is not the case anymore due to the deformation that the omnidirectional sensor introduces. Moreover, not only do omnidirectional and planar images coexist, but they are often used jointly, for instance in camera networks. This is a source of new problems, since not only matching between omnidirectional images is needed, but also between omnidirectional and planar ones.

A common way of tackling the matching problem between two given images is by using interest points. These are points in an image that fulfil some “interest” criterion. This criterion is usually defined in such a way that the obtained points have a well-defined position, contain as much local information on the surroundings as possible and are robust against changes in the image, such as noise, perspective transformations, illumination changes, etc. The location of these points is often used for the extraction of local image descriptors. These are a transformation of the local image data into an element of the transformed space, usually a vector, where some characteristics are coded, as for example the shape, the orientation, the colour, the texture, etc. They can be used afterwards for instance in matching or image registration.

1.1 State-of-the-art

Interest points are widely used nowadays by computer vision algorithms. As commented before, two main aspects make these points useful: robustness against image changes and richness of local information in terms of

local image structure. A wide variety of interest points has been defined to best combine these two aspects, like for example Harris-Stephens corners (Harris and Stephens, 1988), SUSAN corners (Smith and Brady, 1997), salient regions (Kadir and Brady, 2001), Maximally Stable Extremal Regions (MSER) (Matas et al., 2002) or extrema of the Difference of Gaussians (DoG) (Lowe, 2004). An excellent survey on this kind of points can be found in (Tuytelaars and Mikolajczyk, 2007). Due to their stability, these keypoints are often used for the computation of local descriptors, which can be used afterwards for several tasks, such as tracking, object detection or region matching. A wide variety of local descriptors has also been proposed in the literature (Zabih and Woodfill, 1994; Van Gool et al., 1996; Baumberg, 2000; Lowe, 2004; Mikolajczyk and Schmid, 2005). For an exhaustive comparison of local descriptors see (Mikolajczyk and Schmid, 2005).

Scale Invariant Feature Transform (SIFT), introduced in (Lowe, 2004), is a well-known algorithm that successfully combines both notions. For interest points, it considers extrema of the Difference of Gaussians, and for local descriptors, a histogram of orientations. The SIFT algorithm detects points in a scale-invariant way, as extrema in the response of the convolution of the image with a DoG function

$$\psi(\mathbf{x}, \mathbf{y}, \sigma) = g(\mathbf{x}, \mathbf{y}, k\sigma) - g(\mathbf{x}, \mathbf{y}, \sigma), \quad (1)$$

where $g(\mathbf{x}, \mathbf{y}, \sigma)$ denotes a 2-dimensional Gaussian kernel with standard deviation σ . This is based on the work of T. Lindeberg in (Lindeberg, 1998), and the convolution of an image with $\psi(\mathbf{x}, \mathbf{y}, \sigma)$ can be computed as the difference of consecutive images in the scale-space representation of the image, choosing properly the value of k . The scale-space representation $L(\mathbf{x}, \mathbf{y}, t) : \mathbb{R}^2 \times \mathbb{R}^+ \rightarrow \mathbb{R}$ of an image $I(\mathbf{x}, \mathbf{y})$ can be equivalently defined in two different ways. The first one is the evolution over time of the heat distribution $I(\mathbf{x}, \mathbf{y})$ in an infinite homogeneous medium:

$$\partial_t L(\mathbf{x}, \mathbf{y}, t) = \frac{1}{2} \nabla^2 L(\mathbf{x}, \mathbf{y}, t), \quad (2)$$

where the initial condition is $L(\mathbf{x}, \mathbf{y}, 0) = I(\mathbf{x}, \mathbf{y})$. The second one is the successive convolution of the image with a Gaussian kernel, $g(\mathbf{x}, \mathbf{y}, \sigma)$, of

standard deviation $\sigma = \sqrt{t}$:

$$L(\mathbf{x}, \mathbf{y}, \sigma) = g(\mathbf{x}, \mathbf{y}, \sigma) * I(\mathbf{x}, \mathbf{y}). \quad (3)$$

This scale-space representation of an image can be computed efficiently using the definition of the convolution, thanks to the separability of the Gaussian filter. The local data around each interest point is then used to compute SIFT descriptors. These local descriptors are invariant to rotation and scale changes. They consist of a 3D histogram: two spatial dimensions and one dimension for orientations. The size of this region depends on the scale at which the point has been detected. Thanks to its simplicity, good results in terms of repeatability and accuracy on matching, it has been used to treat applications requiring tracking or matching of regions (Sirmacek and Unsalan, 2009; Brox et al., Accepted for future publication).

Several variants of the SIFT algorithm have appeared, trying to improve the interest point extraction or the local descriptor. Among those trying to improve the interest point extraction, the most remarkable representative is probably the Speed-Up Robust Features (SURF) algorithm (Bay et al., 2008). For those trying to improve the local descriptor, a good representative is the Gradient Location and Orientation Histogram (GLOH) introduced in (Mikolajczyk and Schmid, 2005).

All these algorithms and techniques have been developed to work with regular (planar) images or videos. Over the last years, though, omnidirectional imaging has become a popular topic, due to both, the availability of simple sensors (e.g. parabolic mirrors mounted on regular cameras) and the great advantages it provides (e.g. a 360 degrees view in one single image). This kind of sensors has a lot of applications, such as video surveillance (Boult et al., 2001) or object tracking (Chen et al., 2008), and its use has become very common in robot navigation (Menegatti et al., 2006) and in autonomous vehicles (Ehlgen et al., 2008; Scaramuzza and Siegwart, 2008). Interest points and local descriptors-based techniques, such as SIFT, have been applied to omnidirectional images due to their good performance in planar images (Goedeme et al., 2005; Tamimi et al., 2006; Valgren and Lilien-

thal, 2007; Scaramuzza and Siegwart, 2008). Recently, several efforts have been made to develop algorithms specifically designed to treat these omnidirectional images (Bogdanova et al., 2007; Hadj-Abdelkader et al., 2008). An important aid in this sense were the results of Geyer and Daniilidis in (Geyer and Daniilidis, 2001), where they show that the most common catadioptric omnidirectional images can be bijectively mapped on the surface of a sphere. In Figure 1, the equivalence between a parabolic projection and a central projection followed by a stereographic projection is schematically shown; and in Figure 2, a particular example of mapping a real parabolic omnidirectional image on the sphere through inverse stereographic projection can be seen. Consequently, a whole family of omnidirectional images can be processed by algorithms treating spherical images. The mapping from the captured image to the sphere is the only adaptation needed for each element of the family. Based on this result, Hansen et al. (Hansen et al., 2007b; Hansen et al., 2007a) developed a SIFT-like algorithm on the sphere to match points between wide-angle images. In this algorithm, the point extraction is computed on the back-projection of the spherical scale-space to the wide-angle image plane, and the descriptor is computed using a fixed size patch of 41×41 pixels around each extracted point at the corresponding scale.

1.2 Contributions

This paper is two fold, it proposes both a SIFT algorithm in spherical coordinates and a new approach to match points between two spherical images or between spherical and planar images.

First of all, we propose an interest point extractor on the sphere based on the spherical scale-space representation and the SIFT algorithm. This algorithm processes omnidirectional images mapped on the sphere (see Figure 2b) in order to take into account the deformation introduced by the omnidirectional sensor (see Figure 2a). The creation procedure of the spherical scale-space is speeded up by successive downsampling of the input image for each octave. However, considering that spherical Fourier transforms are com-

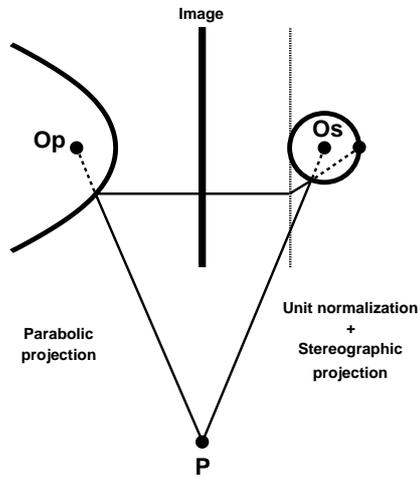


Figure 1: Equivalence between the image of a point P through a parabolic projection of centre O_P and through a normalisation to the unit sphere with centre O_S followed by stereographic projection.

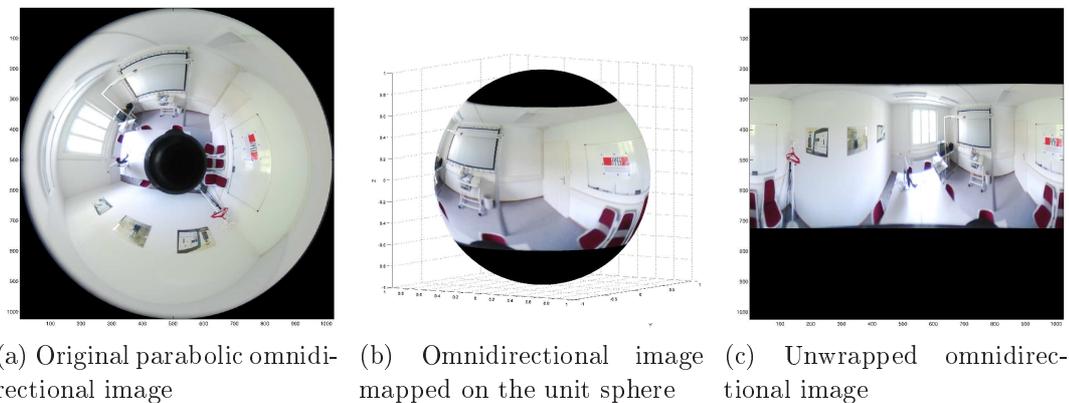


Figure 2: Example of mapping a parabolic omnidirectional image on the sphere. The unwrapped spherical image (Figure 2c) is often used for visualisation purposes.

puted, this division is subject to aliasing. For this reason, an anti-aliasing criterion is defined to decide whether an image is downsampled or not.

Our second main contribution concerns the matching part. We here propose two types of descriptors. The first one is meant to carry out matching between two spherical images. And the second one allows to perform matching between spherical and planar images. Both descriptors can be very useful when working, for example, with camera networks containing at the same time omnidirectional and regular cameras, what is known as hybrid camera networks. In such case, the descriptors proposed here can easily help registering data from all the components of the network.

Finally, we also introduce a transformation between planar and spherical images. This transformation can be used to send the contour or regions of a planar object in a planar image to a spherical one and vice versa. The parameters of this transformation are estimated by means of the obtained matched points, cleaning false detections with the Random Sample Consensus (RANSAC) algorithm (Fischler and Bolles, 1981). The inputs of the estimation process are respectively the matched points from the omnidirectional and the planar images.

Several experiments are performed on real omnidirectional images to test the proposed algorithms. The code developed for these tests has been implemented in Matlab[®] and source code and images are freely available¹ under the GPL license. The source code requires the installation of the “Yet Another Wavelet Toolbox” (YAWTb)² for MatLab. This library provides an efficient way of computing the spherical harmonic transformations as well as a nice visualisation interface. Finally, for the RANSAC routines, we used the RANSAC Toolbox³.

This paper is organised as follows. In Section 2, the mathematical as-

¹<http://transp-or2.epfl.ch/pagesPerso/javierFiles/software.php>

²<http://rhea.tele.ucl.ac.be/yawtb>

³<http://www.mathworks.com/matlabcentral/fileexchange/18555>

pects of the interest point extraction on the sphere are exposed. In Section 3, the proposed algorithm is described in detail, as well as the two proposed descriptors. Then, in Section 4, a “plane to sphere” transformation (similar to a homography between planar images) is defined and a method for its estimation is given. In Section 5, several experiments are presented, between omnidirectional images in Section 5.1 and between omnidirectional and planar images in Section 5.2. Finally, in Section 6, some conclusions and potential lines for future research are given.

Now, we can start with the basic notions of harmonic analysis on the sphere.

2 Spherical scale-space

Let us consider two functions $f, h \in L^2(S^2)$ defined on the two-sphere $S^2 \in \mathbb{R}^3$. Then, the convolution on the sphere reads

$$(f * h)(\omega) = \int_{r \in \text{SO}(3)} f(r\eta)h(r^{-1}\omega)dr, \quad (4)$$

where $\omega \equiv (\theta, \varphi) \in S^2$, $\theta \in [0, \pi]$, $\varphi \in [0, 2\pi)$ (see Figure 3) and η is the north pole. Equation 4 is hard to compute, but as it was demonstrated by Driscoll and Healy in (Driscoll and Dennis M. Healy, 1994), the convolution of two spherical functions $f, h \in L^2(S^2)$ can be calculated more efficiently as the pointwise product of their spherical Fourier transforms:

$$\widehat{(f * h)}(\mathbf{l}, \mathbf{m}) = 2\pi\sqrt{\frac{4\pi}{2\mathbf{l} + 1}}\widehat{f}(\mathbf{l}, \mathbf{m})\widehat{h}(\mathbf{l}, 0), \quad (5)$$

where $\widehat{(\cdot)}$ is the spherical Fourier transform of the function.

The spherical Fourier transform of a function $f \in L^2(S^2)$ is the set of coefficients of the expansion of this function in terms of spherical harmonics Y_l^m , i.e. the coefficients $\widehat{f}(\mathbf{l}, \mathbf{m})$ of the expression

$$f(\theta, \varphi) = \sum_{\mathbf{l} \geq 0} \sum_{|\mathbf{m}| \leq \mathbf{l}} \widehat{f}(\mathbf{l}, \mathbf{m})Y_l^m(\theta, \varphi), \quad (6)$$

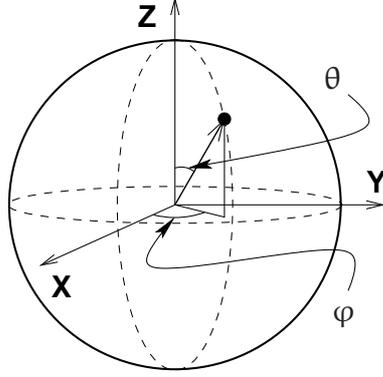


Figure 3: Spherical coordinates of a point on S^2 ($\rho = 1$)

where the function $f(\theta, \varphi)$ and the spherical harmonics $Y_l^m(\theta, \varphi)$ are expressed in spherical coordinates for the unit sphere ($0 \leq \theta \leq \pi$, $0 < \varphi \leq 2\pi$, $\rho = 1$, see Figure 3). The spherical harmonics can be factorized as

$$Y_l^m(\theta, \varphi) = k_{l,m} P_l^m(\cos \theta) e^{im\varphi}, \quad (7)$$

where P_l^m is an associated Legendre polynomial and $k_{l,m}$ is a normalisation constant that is

$$k_{l,m} = \sqrt{\frac{2l+1}{4\pi} \frac{(l-m)!}{(l+m)!}} \quad (8)$$

in case of orthonormal spherical harmonics (see (Barut and Rączka, 1986) for further details).

Then, the spherical Fourier transform of a function $f \in L^2(S^2)$ can be calculated as the projection of this function on the orthonormal basis of the spherical harmonics

$$\begin{aligned} \hat{f}(l, m) &= \langle f, Y_l^m \rangle = \\ &= k_{l,m} \int_{S^2} d\mu(\omega) \overline{Y_l^m(\omega)} f(\omega), \end{aligned} \quad (9)$$

where $d\mu = \sin \theta d\theta d\omega$ is $SO(3)$ invariant measure on S^2 . Using Equation 7 and Equation 9, it is easy to see that the spherical Fourier transform is

a regular Fourier transform in φ followed by a projection on the associated Legendre polynomial.

At this point, the only missing element to build the spherical scale-space representation of a spherical image is the function that plays the role of the Gaussian kernel in the planar case. In (Bulow, 2004), the author derives this function as a Green function of the heat equation (Equation 2) over S^2 , obtaining

$$g^{S^2}(\theta, \varphi, \sigma) = \sum_{l \in \mathbb{N}} \sqrt{\frac{2l+1}{4\pi}} Y_{l0}(\theta, \varphi) e^{-\frac{l(l+1)\sigma^2}{2}}, \quad (10)$$

$$\widehat{g}^{S^2}(l, m, \sigma) = \sqrt{\frac{2l+1}{4\pi}} e^{-\frac{l(l+1)\sigma^2}{2}}, \quad (11)$$

where g^{S^2} denotes the spherical Gaussian function. Therefore, using Equation 5, the spherical Fourier transform of the scale-space representation of an omnidirectional image mapped on the sphere, $I(\theta, \varphi)$, is

$$\widehat{L}^{S^2}(l, m, \sigma) = \widehat{I}(l, m) e^{-\frac{l(l+1)\sigma^2}{2}} \quad (12)$$

for the set of considered scales (different values of σ), and its inverse spherical Fourier transform,

$$L^{S^2}(\theta, \varphi, \sigma) = I(\theta, \varphi) * g^{S^2}(\theta, \varphi, \sigma), \quad (13)$$

is the spherical scale-space representation of this image. Finally, the spherical difference of Gaussians can be computed as

$$\psi^{S^2}(\theta, \varphi, \sigma) = L^{S^2}(\theta, \varphi, k\sigma) - L^{S^2}(\theta, \varphi, \sigma). \quad (14)$$

Using these expressions, the algorithm for the extraction of interest points will be presented in the next section.

3 SIFT on the sphere

Let us define the SIFT algorithm in spherical coordinates. In this algorithm, the extraction of interest points and the local descriptor calculations are

performed on the surface of the unit sphere. Here, we propose two types of descriptors: Local Spherical Descriptors (LSD) and Local Planar Descriptors (LPD). The first one is computed directly on the sphere and is intended to be matched with other LSD of points extracted from other omnidirectional images. The second one is generated using a local planar approximation of the region around the extracted interest point, and can be matched with regular SIFT descriptors of points extracted from planar images. For the matching procedure we follow the method proposed by (Lowe, 2004). It consists of pairing the nearest points in terms of the distance between their descriptors, if and only if the ratio between this distance and the second smallest distance is lower than a fixed threshold d . Along all the paper, a matching between two points, p_1 and p_2 , in two different images is considered as correct if the distance between p_1 and p_2 in the first image is lower than σ_1 , where σ_1 is the scale at which the point p_1 was detected. This can be equivalently defined in the second image using σ_2 , σ_2 being the scale of detection of p_2 in this image.

The workflow of the spherical SIFT algorithm is summarised in Algorithm 1. Each one of the steps is described in details in the following sections. Throughout the paper, a spherical image will be considered defined in a (θ, φ) -grid where columns are points of constant longitude, $\varphi \in [0, 2\pi)$, and rows are points of constant latitude, $\theta \in [0, \pi]$.

Algorithm 1 Spherical SIFT algorithm

- 1: $I(\theta, \varphi) \leftarrow$ omnidirectional input image mapped on S^2
 - 2: Compute spherical scale-space representation of $I(\theta, \varphi)$
 - 3: Compute difference of Gaussians
 - 4: $E \leftarrow$ Local extrema of difference of Gaussians
 - 5: for each $E_i \in E$ do
 - 6: Compute LSD and/or LPD of E_i
 - 7: end for
-

3.1 Spherical scale-space and difference of Gaussians

The spherical scale-space representation of a spherical image $I(\theta, \varphi)$ (ρ is fixed to 1) is computed using iteratively Equation 12, i.e.

$$L^{S^2}(\theta, \varphi, \sigma_i) = L^{S^2}(\theta, \varphi, \sigma_{i-1}) * g^{S^2}(\theta, \varphi, \tilde{k}_i \sigma_0), \quad (15)$$

where σ_0 is the initial scale and \tilde{k}_i is chosen in such a way that two neighbouring scales in the spherical scale-space representation are separated by a constant multiplicative factor $k = 2^{1/S}$ (in order to have a constant number S of images per octave). Therefore, $\sigma_i = k\sigma_{i-1} = k^i\sigma_0$ and using the semi-group property of the spherical scale-space representation, we have that $(k^i\sigma_0)^2 + (\tilde{k}_i\sigma_0)^2 = (k^{i+1}\sigma)^2$, and so $\tilde{k}_i = k^i\sqrt{k^2 - 1}$. These expressions are also valid in the planar case.

The generation process of the spherical scale-space representation is speeded up by downsampling the image by two, instead of increasing the σ , each time a complete octave of $\psi^{S^2}(\theta, \varphi, \sigma)$ can be obtained. This is the common practice in the planar case too, but in the spherical case there is a peculiarity. In order to obtain $L^{S^2}(\theta, \varphi, \sigma)$, a spherical Fourier transform is computed and, therefore, aliasing has to be taking into account. This process of downsampling by 2 the images is especially sensitive to aliasing, since the bandwidth of the spherical Fourier transform is also divided by 2. For this reason, after the computation of each octave, the next condition is tested:

$$e^{\frac{-nH(nH+1)(\sigma_0/k)^2}{8}} \leq e^{-1}, \quad (16)$$

where nH is the new height of the image after reducing its size. If it is not fulfilled, instead of reducing the image size for the next octave, σ are increased and the image size is reduced after the convolution. Aliasing effects can still appear if they are already present in the first computed spherical Fourier transform, or if σ increases considerably (Equation 16 not fulfilled even for the current H before downsampling). An example of the effect of applying this anti-aliasing criterion before downsampling an intermediate image in the computation of ψ^{S^2} , is shown in Figure 4.

The input images are supposed to have a nominal standard deviation σ_N of half pixel, which in our case means $\sigma_N = 0.5\pi/H$, where H is the height of the spherical image. To obtain the first image of the spherical scale space, $L^{S^2}(\theta, \varphi, \sigma_0/k)$, the input image is convolved with a spherical Gaussian filter with standard deviation $\sigma = \sqrt{(\sigma_0/k)^2 - \sigma_N^2}$. The computation of ψ^{S^2} is shown in Algorithm 2. Note that the size of the input image can be doubled before starting the process. Then, $\sigma_N = \pi/H$ and the first loop starts at $o = -1$.

Algorithm 2 Spherical scale-space and difference of Gaussians computation

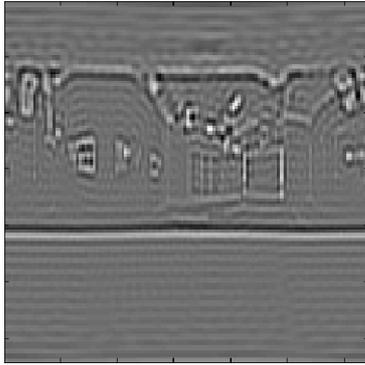
```

1:  $S \leftarrow$  number of stages per octave
2:  $O \leftarrow$  number of octaves
3:  $n \leftarrow 0$ 
4: for  $o = 0$  to  $O$  do
5:   Compute  $L^{S^2}(\theta, \varphi, 2^o\sigma_0/k)$ 
6:   for  $s = 0$  to  $S + 1$  do
7:     Compute  $L^{S^2}(\theta, \varphi, 2^o k^s \sigma_0)$ 
8:     Compute  $\psi^{S^2}(\theta, \varphi, 2^o k^{s-1} \sigma_0)$ 
9:   end for
10:  if Equation 16 is satisfied then
11:    Downsample by 2 the starting image of the current loop and use it
        for the next one
12:  else
13:     $n \leftarrow n + 1$ 
14:    Double the  $\sigma$ 's of the current loop and use them in the next loop
15:    Each  $L^{S^2}(\theta, \varphi, \sigma)$  in the next loop has to be downsampled by  $2^n$ 
16:  end if
17: end for

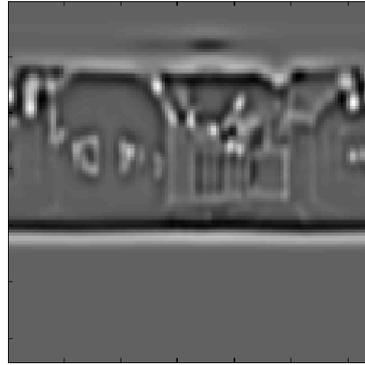
```

3.2 Extrema extraction

Interest points are local extrema of $\psi^{S^2}(\theta, \varphi, \sigma)$ (Equation 14). A local extreme is a point on the spherical grid whose value is bigger (smaller) than



(a) $\psi^{S^2}(\theta, \varphi, 2^3 k \sigma_0)$ (third stage of the fourth octave) downsampling the image without applying the anti-aliasing criterion



(b) $\psi^{S^2}(\theta, \varphi, 2^3 k \sigma_0)$ (third stage of the fourth octave) downsampling the image if the anti-aliasing criterion is fulfilled

Figure 4: Example of the effect of the anti-aliasing strategy for the ψ^{S^2} computation of the image in Figure 2b. The image size is 1024×1024 and the spherical scale-space was generated using $\sigma_0 = 1.6\pi/1024$, $\sigma_N = 0.5\pi/1024$ and $S = 3$.

its 8 neighbours, bigger (smaller) than its 9 neighbours in the scale above and bigger (smaller) than its 9 neighbours in the scale below. Note that, contrary to a planar image, an image on the sphere has no borders and then, points located at the last column (highest values of φ) are neighbours with points located at the first column (lowest values of φ) and vice versa. These simple comparisons give the extrema candidates, but principal curvature and contrast conditions are imposed on these points afterwards in order to keep only the most stable ones.

For each detected local extreme of $\psi^{S^2}(\theta, \varphi, \sigma)$, $\omega_i \equiv (\theta_i, \varphi_i, \sigma_i)$, a quadratic function is fitted by using a Taylor expansion of Equation 14:

$$\begin{aligned} \psi^{S^2}(\theta, \varphi, \sigma) &\simeq \psi^{S^2}(\theta_i, \varphi_i, \sigma_i) + \\ &+ \left. \frac{\partial \psi^{S^2}}{\partial \Theta} \right|_{\omega_i}^\top \delta_{\omega_i} + \frac{1}{2} \delta_{\omega_i}^\top \left. \frac{\partial^2 \psi^{S^2}}{\partial \Theta^2} \right|_{\omega_i} \delta_{\omega_i}, \end{aligned} \quad (17)$$

where $\Theta \equiv (\theta, \varphi, \sigma)$ and $\delta_{\omega_i} = (\theta - \theta_i, \varphi - \varphi_i, \sigma - \sigma_i)^\top$. The derivatives are calculated as the central finite differences approximation of the derivatives of the image in that point, i.e. for a function $f: \mathbb{R}^n \rightarrow \mathbb{R}^m$ the central finite difference approximation of the derivative with respect to the j th variable, x^j is

$$\begin{aligned} \frac{\partial f}{\partial x^j} &= f_{x^j}(x^1, x^2, \dots, x^j, \dots, x^n) = \\ &= \frac{f(\dots, x^j + \Delta_{x^j}, \dots) - f(\dots, x^j - \Delta_{x^j}, \dots)}{2\Delta_{x^j}}. \end{aligned} \quad (18)$$

From now on, the notation f_x will be used to express the derivative (or the finite differences approximation) of f with respect to x .

Taking the derivative of Equation 17 with respect to δ_{ω_i} , the offset $\tilde{\delta}_{\omega_i}$ to the extreme of the fitted function is obtained

$$\tilde{\delta}_{\omega_i} = - \left(\frac{\partial^2 \psi^{S^2}}{\partial \Theta^2} \right)^{-1} \frac{\partial \psi^{S^2}}{\partial \Theta}. \quad (19)$$

If any of the components of vector $\tilde{\delta}_{\omega_i}$ is bigger than half the separation between two points in this dimension, the point ω_i is moved to its neighbour in this dimension and the process repeated. To avoid loops, the maximum number of iterations in the implemented version has been fixed to 5. For the tests presented in this paper, the movement in the σ dimension has not been taken into account. At the end of this iterative process, a point $\tilde{\omega}_i$ is obtained.

Once $\tilde{\omega}_i$ has been obtained, the contrast at this point is computed, and if the condition

$$|\psi^{S^2}(\tilde{\delta}_{\tilde{\omega}_i})| > \frac{0.02}{k^s 2^o} \quad (20)$$

is not satisfied, $\tilde{\omega}_i$ is discarded. The threshold value in Equation 20 has been defined empirically.

Finally, the ratio r of principal curvatures is obtained at $\tilde{\omega}_i$ and the point is kept if and only if $r < 10$ (same value than in (Lowe, 2004)). This eliminates points situated along edges, where one principal curvature is high but the other is low, which produces unstable points. In other words, if a point does not satisfy the following condition

$$\frac{\text{trace}(\mathbf{H}^{S^2})^2}{\det(\mathbf{H}^{S^2})} < \frac{(r+1)^2}{r}, \quad (21)$$

where

$$\mathbf{H}^{S^2} = \begin{pmatrix} \psi_{\theta\theta}^{S^2} & \psi_{\theta\varphi}^{S^2} \\ \psi_{\theta\varphi}^{S^2} & \psi_{\varphi\varphi}^{S^2} \end{pmatrix}, \quad (22)$$

it is discarded. The full extrema extraction procedure can be found in Algorithm 3.

3.3 Local Spherical Descriptor (LSD)

In order to match points extracted from different omnidirectional images and obtained with the proposed algorithm, a Local Spherical Descriptor (LSD) is computed at each point. This descriptor is obtained using the spherical

Algorithm 3 Algorithm for the extraction of “good” local extrema

```

1:  $E \leftarrow \emptyset$  the set of local extrema
2: for  $o = 0$  (or  $o = -1$ ) to  $O$  do
3:   for  $s = 0$  to  $S - 1$  do
4:     for each point  $\omega_i \equiv (\theta_i, \varphi_i, \sigma_i)$  of  $\psi^{S^2}(\theta, \varphi, 2^o k^s \sigma_0)$  do
5:       if  $\omega_i$  is a local extreme then
6:         Compute  $\tilde{\omega}_i$ 
7:         if Equations 20 and 21 are satisfied at  $\tilde{\omega}_i$  then
8:            $E \leftarrow \{E, \tilde{\omega}_i\}$ 
9:         end if
10:      end if
11:    end for
12:  end for
13: end for

```

scale-space representation of the image (see Sections 2 and 3.1) and consists of a set of histograms of orientations in a region around the given point. The size of this region depends on the scale (σ) at which the point has been detected. Orientations are computed with respect to a principal orientation of the point, which makes the descriptor invariant to rotations around the axis that links the point with the centre of the sphere. The complete procedure is detailed below.

First, the orientation of a point in the spherical scale space representation has to be defined. Let us have a point $(\theta, \varphi) \in S^2$ at scale σ . Its orientation is defined as the angle of the gradient of L^{S^2} in that point, with the 0 degrees pointing to the south pole and the 90 degrees to bigger values of φ . These gradients are obtained using the central finite differences approximation of the derivatives (Equation 18) as

$$\alpha(\theta, \varphi, \sigma) = \arctan \left(\frac{L_{\varphi}^{S^2}(\theta, \varphi, \sigma)}{L_{\theta}^{S^2}(\theta, \varphi, \sigma)} \right). \quad (23)$$

Then, for each considered extreme of the ψ^{S^2} , Equation 23 is used to compute the orientations of surrounding points on the spherical grid in a

$3\sigma \times 3\sigma$ squared window centred at the extreme (where σ is the scale at which each extreme was located). To define this window, the distance between two points on the unit sphere, $\mathbf{p}_1 \equiv (\theta_1, \varphi_1)$ and $\mathbf{p}_2 \equiv (\theta_2, \varphi_2)$, needs to be calculated. It can be obtained using the Vincenty's formula (Vincenty, 1975):

$$d(\mathbf{p}_1, \mathbf{p}_2) = \arctan \left(\frac{\sqrt{A^2 + B^2}}{C} \right), \quad (24)$$

where

$$A = \sin \theta_1 \sin \Delta\varphi, \quad (25)$$

$$B = \sin \theta_2 \cos \theta_1 - \cos \theta_2 \sin \theta_1 \cos \Delta\varphi, \quad (26)$$

$$C = \cos \theta_2 \cos \theta_1 + \sin \theta_2 \sin \theta_1 \cos \Delta\varphi, \quad (27)$$

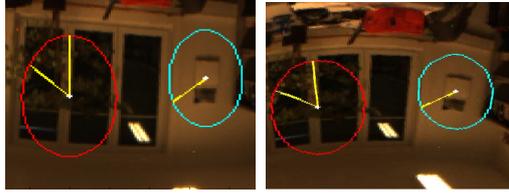
$$\Delta\varphi = \varphi_1 - \varphi_2. \quad (28)$$

For each window, a histogram of orientations is computed using the orientations of points of the spherical grid that are inside. The orientation value at each point defines the bin, and the value added to this corresponding bin is the norm of the gradient at that point,

$$m(\theta, \varphi, \sigma) = \sqrt{L_\varphi^{S^2}(\theta, \varphi, \sigma)^2 + L_\theta^{S^2}(\theta, \varphi, \sigma)^2}, \quad (29)$$

weighted by a Gaussian centred on the extreme and of standard deviation 1.5σ . For this histogram, 36 orientations are considered. Finally, once the histogram is computed, the principal orientation is calculated as the axis of a parabola fitted around its maximum. If there are bins greater than 0.8 times the biggest one, they are also considered, which results in multiple principal orientations for the same point.

For each principal orientation of each extreme point, the orientations of the points around can be recomputed with respect to this orientation, and the LSD can be computed. This descriptor is a 3-dimensional histogram of orientations (two spatial dimensions and one dimension for orientations) where all the orientations are considered with respect to the principal one. The produced histogram has $4^2 \times 8$ bins (4^2 bins for the spatial dimension



(a) Portion of the un- (b) Portion of the
wrapped image sphere

Figure 5: Example of two points detected using the proposed algorithm. The yellow lines show the principal orientation. As it can be seen, one of the points have two principal orientations.

and 8 bins for the orientations) and is computed considering the points of the spherical grid contained in a $6\sigma \times 6\sigma$ squared window centred at the extreme and rotated according to the principal orientation. Each entry for each bin corresponds to the sum of the gradient magnitudes of points corresponding to this spatial and orientation bin, weighted by a Gaussian centred on the extreme and of standard deviation 1.5σ . The rotation of the window on the surface of the sphere can be computed using the Rodrigues' rotation formula (Rodrigues, 1840) for the rotation of vectors, given by

$$\mathbf{v}^{\text{Rot}} = \mathbf{v} \cos \alpha + \mathbf{u} \times \mathbf{v} \sin \alpha + \mathbf{u} \cdot \mathbf{v} (1 - \cos \alpha) \mathbf{u}, \quad (30)$$

where the vectors are considered in Cartesian coordinates, and the vector \mathbf{v}^{Rot} is the rotated version of \mathbf{v} around \mathbf{u} , α degrees.

In order to avoid boundary effects, the values of each gradient sample are distributed by trilinear interpolation into adjacent histogram bins. The resulting histogram is normalised, each bin thresholded to 0.2 and normalised again, in order to make it robust to contrast changes. The algorithm for the computation of Local Spherical Descriptors is summarised in Algorithm 4.

Algorithm 4 Algorithm for the computation of LSD

- 1: $\text{LSD} \leftarrow \emptyset$ the set of local spherical descriptors
 - 2: for each considered extreme of ψ^{S^2} , $(\theta_i, \varphi_i, \sigma_i)$ do
 - 3: Select a squared region of size $3\sigma_i \times 3\sigma_i$ centred at (θ_i, φ_i)
 - 4: Compute orientations and gradient norms inside this region
 - 5: Compute histogram of orientations
 - 6: $\text{MAX} \leftarrow$ maximum histogram value
 - 7: for each bin value $\geq 0.8\text{MAX}$ do
 - 8: Fit a parabola around this bin
 - 9: $\mathbf{b} \leftarrow$ axis of the parabola
 - 10: Select a squared region of size $6\sigma_i \times 6\sigma_i$ centred at (θ_i, φ_i) and rotated \mathbf{b} degrees
 - 11: Compute orientations and gradient norms inside this region with respect to \mathbf{b}
 - 12: $\text{LSD}_i \leftarrow$ Compute 3-dimensional histogram
 - 13: $\text{LSD} \leftarrow \{\text{LSD}, \text{LSD}_i\}$
 - 14: end for
 - 15: end for
-

3.4 Local Planar Descriptor (LPD)

Local Planar Descriptors (LPD) allow to carry out matching between points extracted from an spherical image, with the algorithm described above, and SIFT descriptors of points extracted from planar images. This is of great importance since, for instance, a preexisting database of SIFT descriptors computed on planar images could be used to detect objects on the omnidirectional image.

The LPD is a regular SIFT descriptor computed on a planar approximation of the region around each interest point $\omega_i \equiv (\theta_i, \varphi_i, \sigma_i)$. We consider $\mathbf{p}_i \equiv (\theta_i, \varphi_i)$ to be the centre of this planar approximation, which is the stereographic projection from the antipodal point of \mathbf{p}_i on the tangent plane on the sphere at \mathbf{p}_i . This stereographic projection of $L^{S^2}(\theta, \varphi, \sigma_i)$ around \mathbf{p}_i can be seen as a local approximation of $L(\mathbf{x}, \mathbf{y}, \sigma)$. In other words, for a point $\mathbf{p}_i \equiv (\theta_i, \varphi_i)$, extracted from the spherical image at the scale σ_i , a squared window centred on ω_i on $L^{S^2}(\theta, \varphi, \sigma_i)$ and of size equal to the minimum between $12\sigma_i$ and π , is stereographically projected from $(\theta_i + \pi/2, \varphi + \pi)$ to the plane tangent at \mathbf{p}_i . The projected points are linearly interpolated in order to obtain a planar image whose cartesian range is $[-2 \tan \frac{6\sigma_i}{2}, 2 \tan \frac{6\sigma_i}{2}] \times [-2 \tan \frac{6\sigma_i}{2}, 2 \tan \frac{6\sigma_i}{2}]$ and with a pixel spacing of $2 \tan \frac{\pi}{2H}$, where H is the height of $L^{S^2}(\theta, \varphi, \sigma_i)$. The equivalent σ_i in the obtained planar image is

$$\sigma_i^{\text{pl}} = \frac{\tan \frac{\sigma_i}{2}}{\tan \frac{\pi}{2H}}. \quad (31)$$

The outline of the computation of Local Planar Descriptors is given in Algorithm 5.

4 Planar to spherical transformation

As said before, LPD can be matched with regular planar SIFT descriptors extracted from planar images. In addition to this new kind of matching, we propose a method to estimate the function that transfers points in a planar

Algorithm 5 Algorithm for the computation of LPD

- 1: $\text{LPD} \leftarrow \emptyset$ the set of local planar descriptors
 - 2: for each considered extreme of ψ^{S^2} , $(\theta_i, \varphi_i, \sigma_i)$ do
 - 3: $L(x, y, \sigma_i^{\text{pl}}) \leftarrow$ stereographic projection of $L(\theta, \varphi, \sigma_i)$ from $(\theta_i + \frac{\pi}{2}, \varphi_i + \pi)$ to the tangent plane at (θ_i, φ_i)
 - 4: $\text{LSD}_i \leftarrow$ SIFT descriptor of $L(x, y, \sigma_i^{\text{pl}})$ at $(x, y) = (0, 0)$
 - 5: $\text{LSD} \leftarrow \{\text{LSD}, \text{LSD}_i\}$
 - 6: end for
-

image to their corresponding points in a spherical image, like an homography does between two planar images. Furthermore, given a segmentation of a planar image, this transformation could be used for instance to extract the segmented objects or regions from an omnidirectional image.

Let us consider a match between a point $\mathbf{p}_i^{\text{pl}} \equiv (x_i^{\text{pl}}, y_i^{\text{pl}})$ in a planar image and a point $\mathbf{p}_j^{S^2} \equiv (x_j^{S^2}, y_j^{S^2}, z_j^{S^2})$ in a spherical image, both in cartesian coordinates. The idea is to find a linear transformation \mathbf{H} which sends \mathbf{p}_i^{pl} to a point \mathbf{q}_{ij} in the space whose projection on the unit sphere is $\mathbf{p}_j^{S^2}$ (see Figure 6). In other words, we look for a 3×3 matrix \mathbf{H} that satisfies

$$\mathbf{p}_j^{S^2} = \frac{\mathbf{q}_{ij}}{\|\mathbf{q}_{ij}\|} = \frac{\mathbf{H}\mathbf{p}_i^{\text{pl}}}{\|\mathbf{H}\mathbf{p}_i^{\text{pl}}\|}, \quad (32)$$

where $\|\cdot\|$ denotes the 2-norm. To achieve that, the central point of the planar image is tangentially placed at the north pole of the sphere that contains the omnidirectional image. In this way, a point $\mathbf{p}_i^{\text{pl}} \equiv (x_i^{\text{pl}}, y_i^{\text{pl}})$ of the planar image has coordinates $(x_i^{\text{pl}}, y_i^{\text{pl}}, 1) \in \mathbb{R}^3$. Then, in order to estimate \mathbf{H} , the fact that $\mathbf{p}_j^{S^2}$ and $\mathbf{H}\mathbf{p}_i^{\text{pl}}$ must be collinear is exploited forcing their vectorial product to be zero, i.e. $\mathbf{p}_j^{S^2} \times \mathbf{H}\mathbf{p}_i^{\text{pl}} = \mathbf{0}$. As in a typical homography estimation problem between two planar images, this condition can be expressed in terms of the elements of \mathbf{H} , giving us 3 equations for

each point

$$\begin{aligned}
& - z_j^{S^2} x_i^{pl} h_{21} - z_j^{S^2} y_i^{pl} h_{22} - z_j^{S^2} z_i^{pl} h_{23} \\
& + y_j^{S^2} x_i^{pl} h_{31} + y_j^{S^2} y_i^{pl} h_{32} + y_j^{S^2} z_i^{pl} h_{33} = 0,
\end{aligned} \tag{33}$$

$$\begin{aligned}
& z_j^{S^2} x_i^{pl} h_{11} + z_j^{S^2} y_i^{pl} h_{12} + z_j^{S^2} z_i^{pl} h_{13} \\
& - x_j^{S^2} x_i^{pl} h_{31} - x_j^{S^2} y_i^{pl} h_{32} - x_j^{S^2} z_i^{pl} h_{33} = 0,
\end{aligned} \tag{34}$$

$$\begin{aligned}
& - y_j^{S^2} x_i^{pl} h_{11} + y_j^{S^2} y_i^{pl} h_{12} + y_j^{S^2} z_i^{pl} h_{13} \\
& + x_j^{S^2} x_i^{pl} h_{21} + x_j^{S^2} y_i^{pl} h_{22} + x_j^{S^2} z_i^{pl} h_{23} = 0,
\end{aligned} \tag{35}$$

where the elements of the matrix H are distributed as

$$H = \begin{pmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{pmatrix} \tag{36}$$

Consequently, if Equations (33, 34, 35) are expressed in terms of h_{lm} and all the resulting equations for each pair of matched points are put together, we obtain a system of equations of the form $Ah = 0$, where

$$h = (h_{11}, h_{12}, h_{13}, h_{21}, h_{22}, h_{23}, h_{31}, h_{32}, h_{33})^\top$$

and A is a $3N \times 9$ matrix, N being the number of points matched between the planar and the spherical image. If the restriction $\|h\| = 1$ is considered, h can be computed as the eigenvector of A corresponding to the smallest eigenvalue, which is the least squares estimator of the solution.

The estimation of this matrix, gives a transformation $h : \mathbb{R}^2 \longrightarrow S^2$ which sends points in the planar image to points in the spherical one

$$h(x, y) = \frac{H \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}}{\left\| H \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} \right\|}. \tag{37}$$

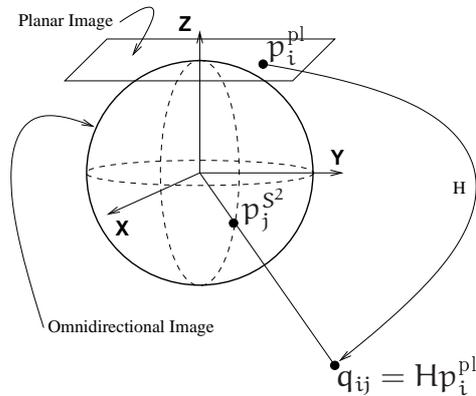


Figure 6: Graphical sketch of the transformation

Transforming points of the spherical image into points of the planar image, can also be done using H^{-1} and normalising the resulting point by its third component, to obtain a point of the form $(x, y, 1)$.

The estimation of H using all the matched points can give bad results due to false matchings. For this reason, in the tests presented in Section 5.1 the set of matched points chosen for the estimation of H is selected using RANSAC. The outline of the process for computing H can be found in Algorithm 6.

Algorithm 6 Algorithm for the estimation of H

- 1: LPD \leftarrow set of local planar descriptors of the spherical image
 - 2: SIFTDesc \leftarrow set of SIFT descriptors of the planar image
 - 3: M \leftarrow matching points between SIFTDesc and LPD
 - 4: H \leftarrow eigenvector with minimum eigenvalue of the matrix defined using Equations 33, 34 and 35 (use RANSAC to clean M of false matchings).
-

5 Experimental results

For our experimental results we use two types of omnidirectional images: parabolic and spherical. The parabolic omnidirectional image is obtained by a catadioptric omnidirectional sensor. We use a parabolic mirror Kaidan EyeSee 360 deg⁴ in combination with a Nikon D40X camera. In order to apply our algorithm on this kind of images, we first need to map them on the sphere according to the projection shown on Figure 1. After this mapping, these images cover a band of about 100 deg on the sphere. The spherical images are obtained with a Ladybug2 device⁵. These images cover around 75% of the sphere. It is important to note that the Ladybug2 outputs the images directly in spherical coordinates and thus no mapping on the sphere is needed for these images. The resolution of both types of images is 1024×1024 . It is important to note that for this image size, the considered values of σ and the use of the anti-aliasing criterion, the aliasing can be neglected. For example, for $\sigma = 4.6\pi/1024$ and a bandwidth of $1024/2 = 512$, $\widehat{g^{S^2}}(512, m, \sigma) \approx 3.95 \cdot 10^{-11}$.

5.1 “Omni vs Omni” matching

First of all, in order to test LSD matching, some parameters of the algorithm need to be fixed, essentially S and σ_0 . To choose them, some test images have been randomly rotated around the x-axis and corrupted with zero mean additive Gaussian noise with standard deviation 0.05 (pixel values are supposed to be in the range $[0, 1]$). Since the transformation between each test image and its rotated and corrupted version is known, for a given point in one image, its corresponding point in the other image can be computed. Therefore, the repeatability of the extracted points can be measured as

$$r_{ij} = \frac{nR}{\min(n_i, n_j)}, \quad (38)$$

⁴<http://www.kaidan.com>

⁵<http://www.ptgrey.com/products/ladybug2/>

where n_i and n_j are the number of extracted points from images i and j , respectively, and nR is the number of repeated points.

On Figure 7 we show the results of the repeatability tests obtained with 20 parabolic omnidirectional images. The same tests for 8 spherical omnidirectional images are shown on Figure 8. These results are similar for both types of omnidirectional images. As expected, the repeatability increases for higher values of σ_0 . But a higher σ_0 also means that the extrema of the DoG at lower scales are lost. Consequently, a compromise has to be found between the smallest scale of the extrema detected and the tolerated amount of “noise” (not repeatable points) between all the extracted points. On the other hand, higher values of S imply a greater number of stages per octave, which requires more computation time. Looking at the graphics, $\sigma_0 = 4.6\pi/1024$ and $S = 3$ are reasonable values to choose. In Table 1, matching results using these values and varying the distance threshold are shown. We can see there how by decreasing the threshold, the number of total matchings is reduced but the percentage of correct matchings is increased. In Figure 9, results of repeatability and correct matchings as a function of pixel noise are given. These graphics show the robustness of the algorithm against image noise.

Looking at the repeatability results, it can also be observed that results similar to these of the original SIFT are obtained with a higher σ_0 value. This is probably due to the fact that the transformation from the original omnidirectional image to the image on S^2 , requires some interpolation. This interpolation can add some artifacts to the image that can produce unstable points that are therefore not good to consider. A higher value of σ_0 solves this problem, as these artifacts disappear with the first convolution with the Gaussian.

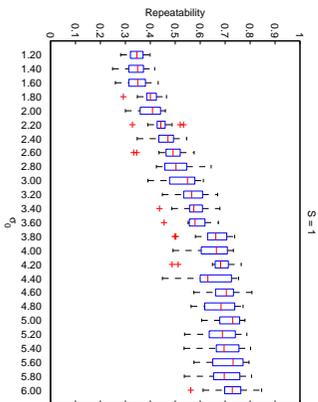
In Figures 10, 11 and 12, some matching results are illustrated. The values of the working parameters are $S = 3$ and $\sigma_0 = 4.6\pi/1024$. First, the matching between two versions of the same parabolic omnidirectional image mapped on the sphere is performed and shown on Figure 10. On the left is the original image and on the right the same image but rotated and cor-

rupted by additive zero mean Gaussian noise with standard deviation 0.05. In total, 136 points were extracted from the original image and 170 from its rotated and corrupted version. From these points, 126 matchings have been obtained. Among them, 115 are correct and 11 incorrect. All the correct matchings are shown in green, but for a purpose of visualisation, only 5 of them are linked. The incorrect matches are shown in red.

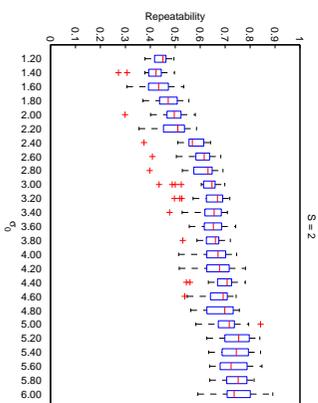
In Figure 11, a plot of the matchings between two parabolic omnidirectional images is illustrated. Both images are of the same scene and were taken from two slightly different places and on two different days. Lighting conditions are very different and there are also some small changes of objects in the scene. The maximum distance ratio was fixed to $d = 0.58$ in order to have a reduced number of matchings. Indeed, as we are matching two different images without any information about the change in the point of view, correct matchings can only be controlled visually. Finally, in Figure 12 is illustrated a plot of the obtained matchings between parabolic and spherical images. As in the other case, the images were taken on two different days and from slightly different places. The maximum distance ratio in this case was fixed to $d = 0.64$.

5.2 “Planar vs Omni” matching

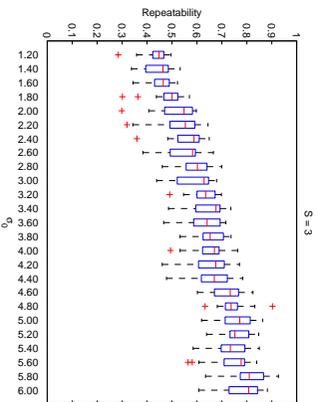
In this section, LPD are tested for matching between points extracted from omnidirectional images (using Algorithm 1) and points extracted from a planar image (using the original SIFT algorithm). For these tests, the SIFT parameters proposed in (Lowe, 2004) ($S = 3$ and $\sigma_0 = 1.6$) are used in both the SIFT algorithm and the spherical SIFT algorithm. Note that on the sphere, the equivalent σ parameter is $\sigma_0 = 1.6\pi/1024$. The omnidirectional image is not doubled to compute octave -1 because the processing of a 2048×2048 image, with all the spherical Fourier transforms involved, takes too much time. In addition, some tests have been done computing octave -1 and it did not increase the number of correct matchings significantly. This could be another effect of the interpolation while mapping the omnidirec-



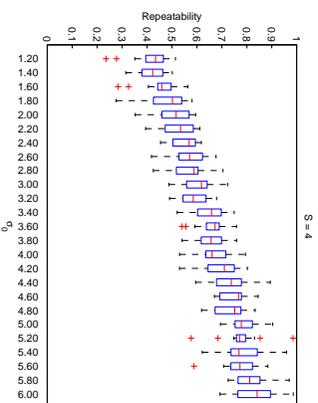
(a) One stage per octave



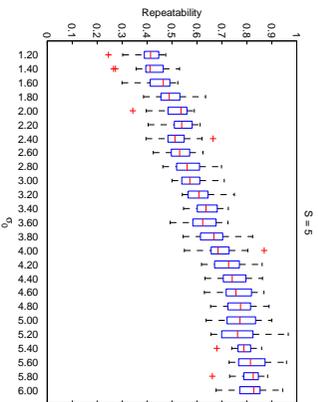
(b) Two stages per octave



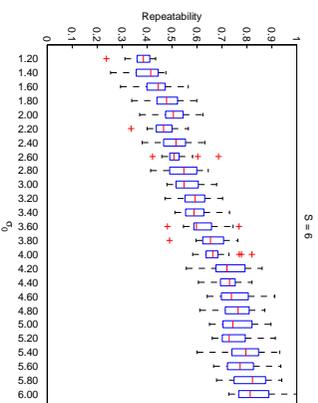
(c) Three stages per octave



(d) Four stages per octave

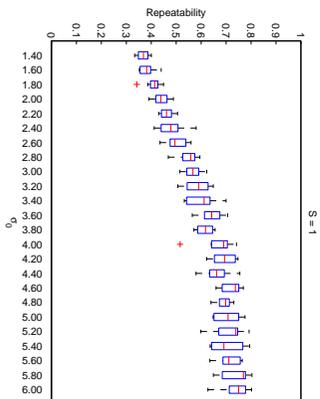


(e) Five stages per octave

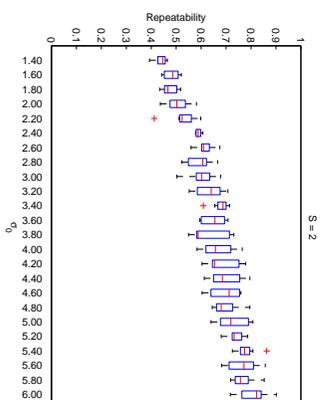


(f) Six stages per octave

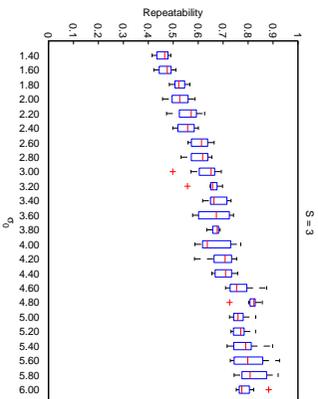
Figure 7: Results of the repeatability tests varying S and σ_0 over a set of 20 parabolic omnidirectional images. Note that the values of σ_0 in the graphs are in terms of relative distance between points of the spherical grid, i.e. for an image of 1024×1024 pixels, $\sigma_0 = 2.0$ in the graph means an effective $\sigma_0 = 2.0 \frac{\pi}{1024}$. The graphs are the output of the boxplot function of Matlab, that plots a box between the 25th and the 75th percentile, a red line for the median and red stars for values considered as outliers (values larger than $q_{75} + 1.5(q_{75} - q_{25})$ or smaller than $q_{25} - 1.5(q_{75} - q_{25})$, where q_{25} and q_{75} are the 25th and 75th percentiles, respectively).



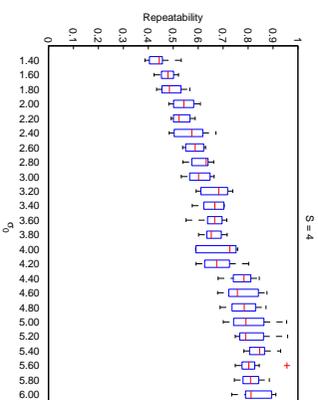
(a) One stage per octave



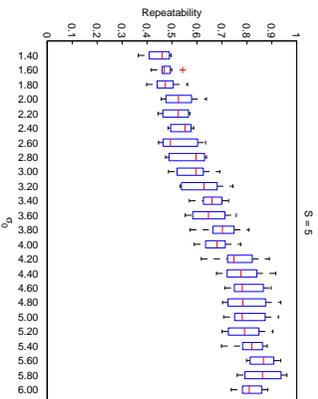
(b) Two stages per octave



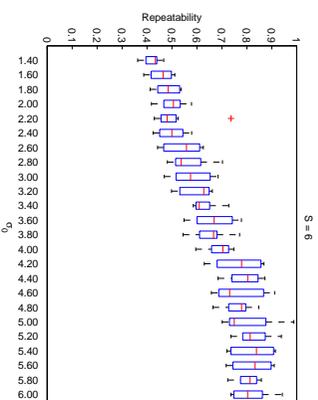
(c) Three stages per octave



(d) Four stages per octave

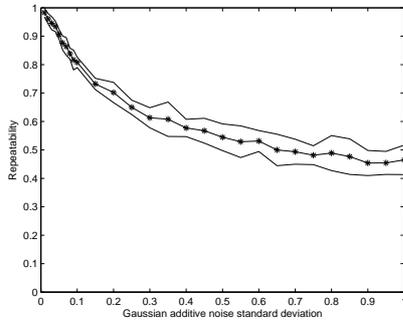


(e) Five stages per octave

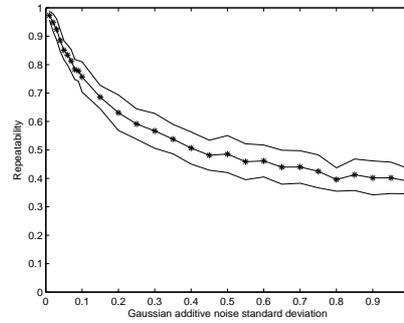


(f) Six stages per octave

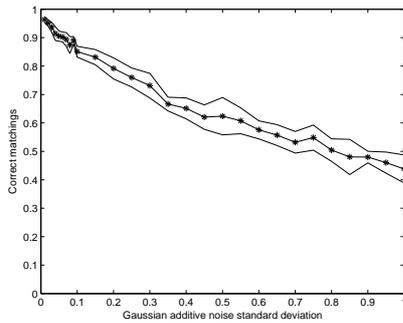
Figure 8: Results of the repeatability tests varying S and σ_0 over a set of 8 images captured using a Ladybug2 device. As in Figure 7, the graphs are the output of the boxplot function of Matlab and the values of σ_0 in the graphs are in terms of relative distance between points of the spherical grid.



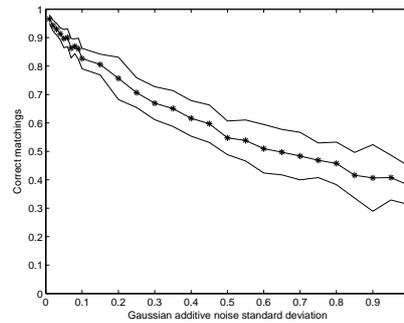
(a) Repeatability varying the amount of noise (spherical images)



(b) Repeatability varying the amount of noise (parabolic omnidirectional images)



(c) Matching performance with $d = 0.6$ varying the amount of noise (spherical images)



(d) Matching performance with $d = 0.6$ varying the amount of noise (parabolic omnidirectional images)

Figure 9: Plots of repeatability and correct matching between omnidirectional images and their corrupted version, varying the standard deviation of Gaussian noise added. The central line represents the mean value and the other two correspond to the mean $\pm \sigma$. The percentage of correct matchings has been computed as the number of correct matchings divided by the number of repeated points. The statistics have been calculated on 23 parabolic omnidirectional images and 8 spherical images.

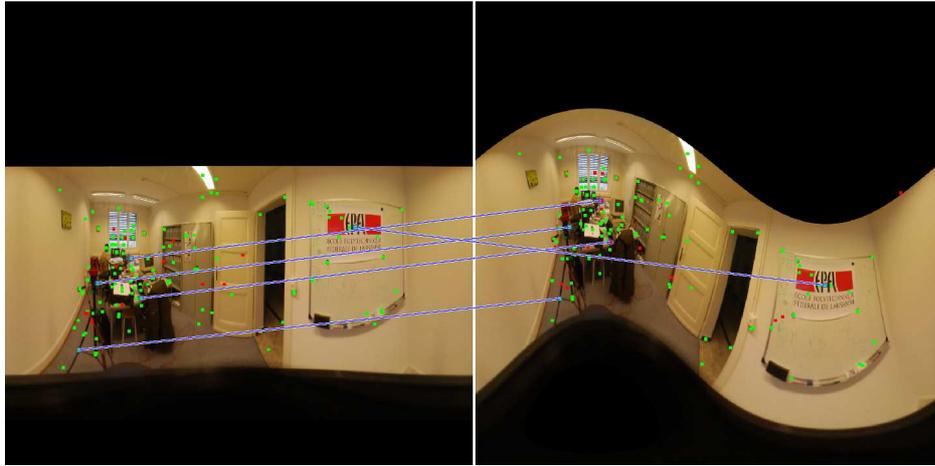


Figure 10: $\varphi - \theta$ -plot of matchings between a parabolic omnidirectional image (left) and the same omnidirectional image rotated $\pi/8$ radians around the x -axis and corrupted by additive zero mean Gaussian noise with standard deviation 0.05 (right). The values $S = 3$, $\sigma_0 = 4.6\pi/1024$ and a maximum distance ratio of 0.8 have been used in the computations. Green dots represent correct matchings, red dots are incorrect matchings and for a purpose of visualisation, 5 correct matchings have been linked. In total, 229 points were extracted from the image on the left and 225 from the image on the right, producing 208 matchings (182 correct and 26 incorrect).

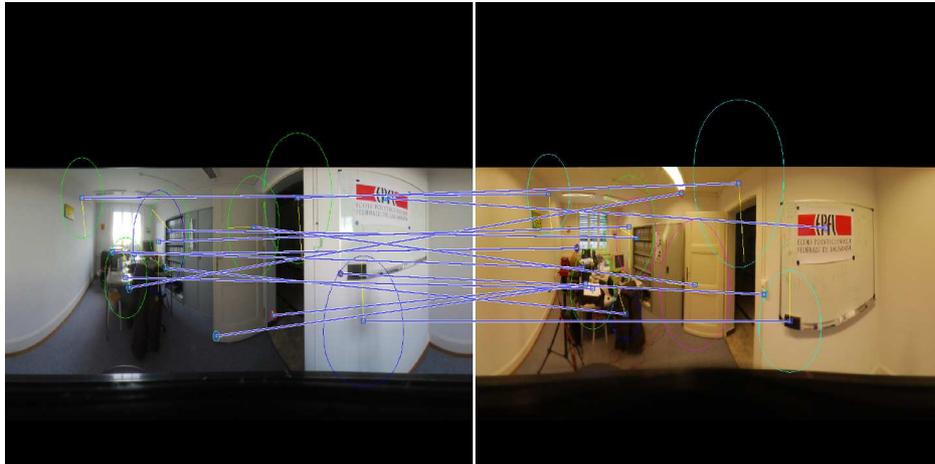


Figure 11: $\varphi - \theta$ plot of matchings between two parabolic omnidirectional images. The values $S = 3$, $\sigma_0 = 4.6\pi/1024$ and a maximum distance ratio of 0.58 have been used in the computations, obtaining 15 matchings (9 correct and 6 incorrect, counted visually). The distance ratio has been chosen to have few point matchings, for visualisation purposes.

tional image on the sphere, which can cause that doubling the size of the image afterwards adds too much noise, and then, too many unstable points.

On Figure 13, two examples of matching between omnidirectional and planar images are shown. An example with a parabolic omnidirectional image is shown on Figure 13a, while on Figure 13b a spherical image is used. For both omnidirectional images, LPDs are extracted as explained in Section 3.4. The planar images represent an object from the omnidirectional scene. For instance, in Figure 13a the object in the planar image is the École Polytechnique Fédérale de Lausanne (EPFL) logo, while in Figure 13b it is a poster. On each of the planar images, the usual SIFT algorithm (Lowe, 2004) has been applied. Comparing Figures 13a and 13b, we note that more points are matched with the spherical image than with the parabolic one (181 and 105 respectively). This higher number of matchings with spherical images is the common situation among all the performed tests. Although, if compared with respect to the total number of extracted points, the results are very

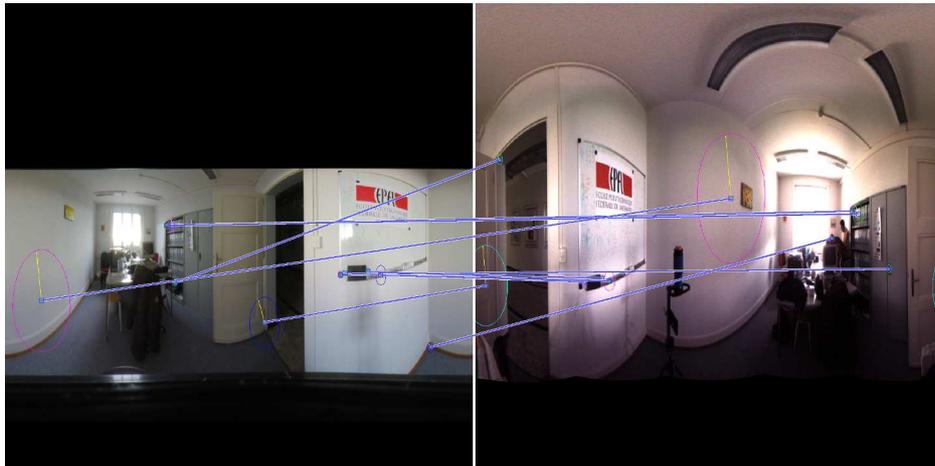


Figure 12: $\varphi - \theta$ plot of matchings between a parabolic omnidirectional image (left) and a spherical image (right). The values $S = 3$, $\sigma_0 = 4.6\pi/1024$ and a maximum distance ratio of 0.62 have been used in the computations, obtaining 10 matchings (6 correct and 4 incorrect, counted visually). The distance ratio has been chosen to have few point matchings, for visualisation purposes.

similar in both cases. Due to the higher level of detail of the spherical image, more points at the lower scales are extracted. This results in a bigger number of matchings between spherical and planar images, while this is not the case between parabolic and planar ones. In fact, the parabolic image somehow loses information because of the interpolation needed for mapping it on the sphere.

The estimation of the planar to spherical transformation, as introduced in Section 4, has been tested using Algorithm 6. First, Algorithm 1 was applied to the corresponding omnidirectional image in order to obtain the set of LPDs. These then serve as inputs of Algorithm 6. Then, the regular SIFT descriptors are computed for the planar images. On Figure 14 the result obtained with images containing the EPFL logo is shown. It is interesting to note that the obtained results are satisfactory even with a highly symmetric object as is the case of the chosen one. In fact, in this case a matching could be locally correct while being incorrect considering the whole object.

The same test for a spherical omnidirectional image is illustrated in Figure 15. The spherical image is captured in a corridor and the planar image contains a poster as an object of the omnidirectional scene. As expected, the poster was correctly extracted after applying Algorithm 6.

6 Conclusions

In this paper, we have proposed a SIFT algorithm in spherical coordinates. It is not limited to pure spherical images, since it can also be applied to a wide variety of omnidirectional images that can be bijectively mapped on the surface of the unit sphere. Two types of descriptors have been proposed to be computed on the points given by the algorithm, Local Spherical Descriptors (LSD) and Local Planar Descriptors (LPD). Using these descriptors, we have successfully performed point matchings between omnidirectional images, with LSD, and between omnidirectional and planar images, with LPD. For the matchings between omnidirectional images, the parameters of the



(a) Unwrapped parabolic omnidirectional image mapped on the sphere (left) vs planar image (right)



(b) Unwrapped spherical image (left) vs planar image (right)

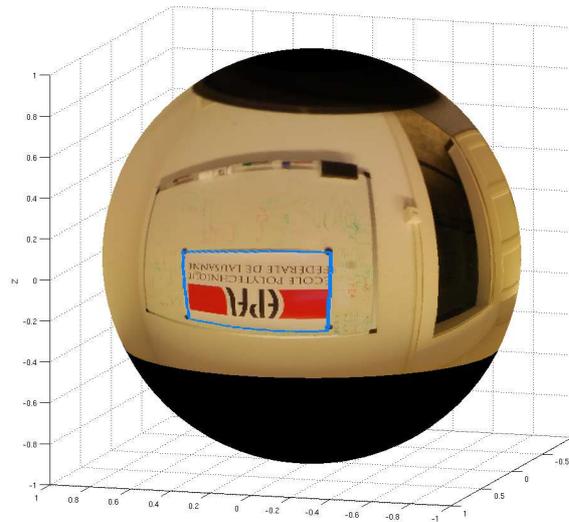
Figure 13: Examples of matchings (green dots) between omnidirectional and planar images for a distance threshold of 0.8. These points have been used as input for Algorithm 6 to generate the images in Figures 14 and 15



(a) Planar image. The small green dots are the matched points between the planar and the spherical image that have been used to compute the transformation. The border of the logo (blue square) has been marked by hand.



(b) Original parabolic image

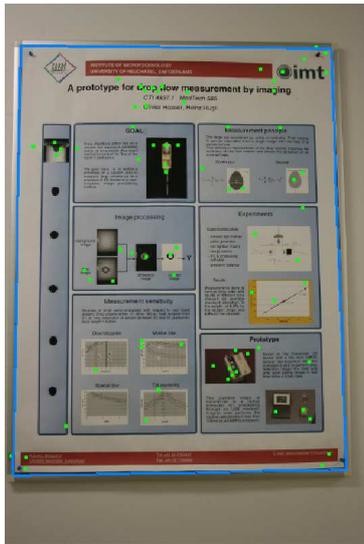


(c) Spherical image. The blue figure is the transformation of the blue square in Figure 14a by means of Equation 37, with H computed using Algorithm 6.

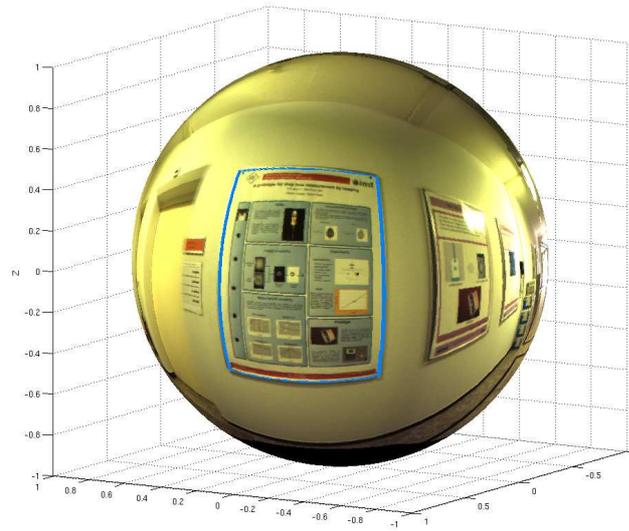
Figure 14: Example of the transformation of the boundary of an object in a planar image to this object in a parabolic omnidirectional image. The transformation function has been obtained using Algorithm 6.



(a) Original Ladybug2 image



(b) Planar image. The small green dots are the matched points between the planar and the spherical image that have been used to compute the transformation. The border of the poster (blue square) has been marked by hand.



(c) Spherical image. The blue figure is the transformation of the blue square in Figure 15b by means of Equation 37, with H computed using Algorithm 6.

Figure 15: Example of the transformation of the boundary of an object in a planar image to this object in a spherical omnidirectional image. The transformation function has been obtained using the Algorithm 6.

algorithm have been chosen according to the results obtained on test images varying S and σ_0 . For the planar vs omnidirectional case, the same parameter values as those proposed in (Lowe, 2004) have been kept. Finally, point matchings obtained in this last case have been successfully used to estimate a planar to spherical transformation. Potential applications of the proposed algorithm are global tracking in hybrid camera networks (together with the SIFT algorithm for planar images), motion estimation in omnidirectional images, object detection and extraction from omnidirectional images and, in general, any problem requiring a matching between points in omnidirectional images or between points in omnidirectional and planar images.

The main drawback of the proposed algorithm is the computation time. Indeed, for a 1024×1024 image, the complete point extraction and LSD computation takes around 1 minute. Computation time depends, however, on the number of stages per octave, the σ_0 value and the number of points extracted. This time could be reduced by optimising the code, but the bottleneck of the spherical Fourier transforms will always be present. This bottleneck, however, could be minimised by implementing the efficient algorithm for the spherical Fourier transform presented recently in (Tygert, 2008).

Many directions for further research can be considered starting from this work. For example, the use of other local descriptors could be tested, in particular the Gradient Location and Orientation Histogram (GLOH) seems to be very appropriate for spherical images. Experiments with real images whose full 3D viewpoint change is known would also be very useful in order to better estimate the optimum parameters of the algorithm and test its performance. And last but not least, the study of approximations of the spherical difference of Gaussians, as it is done in SURF (Bay et al., 2008) for the planar case, would be very interesting, since it could lead to a real-time approximation of this algorithm.

References

- Barut, A. O. and Rączka, R. (1986). Theory of group representations and applications, 2 edn, World Scientific. ISBN 9971502178.
- Baumberg, A. (2000). Reliable feature matching across widely separated views, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 774–781.
- Bay, H., Ess, A., Tuytelaars, T. and Van Gool, L. J. (2008). Speeded-up robust features (SURF), Computer Vision and Image Understanding 110(3): 346–359.
URL: <http://www.sciencedirect.com/science/article/B6WCX-4RC2S4T-2/1/cddc9c43940bd8fb62d6ed668de29e20>
- Bogdanova, I., Bresson, X., Thiran, J.-P. and Vandergheynst, P. (2007). Scale-space analysis and active contours for omnidirectional images, IEEE Transactions on Image Processing 16(7): 1888–1901.
- Boult, T. E., Micheals, R. J., Gao, X. and Eckmann, M. (2001). Into the woods: Visual surveillance of noncooperative and camouflaged targets in complex outdoor settings, Proceedings of IEEE 89(10): 1382–1402.
- Brox, T., Cremers, D., Gall, J. and Rosenhahn, B. (Accepted for future publication). Combined region- and motion-based 3d tracking of rigid and articulated objects, IEEE Transactions on Pattern Analysis and Machine Intelligence .
- Bulow, T. (2004). Spherical diffusion for 3d surface smoothing, IEEE Transactions on Pattern Analysis and Machine Intelligence 26(12): 1650–1654.
- Bur, A., Tapus, A., Ouerhani, N., Siegart, R. and Hugli, H. (2006). Robot navigation by panoramic vision and attention guided features, Proceedings of the 18th International Conference on Pattern Recognition, Washington DC, pp. 695–698.
- Chen, C.-H., Yao, Y., Page, D., Abidi, B., Koschan, A. and Abidi, M. (2008). Heterogeneous fusion of omnidirectional and PTZ cameras for multiple

- object tracking, *IEEE Transactions on Circuits and Systems for Video Technology* 18(8): 1052–1063.
- Driscoll, J. R. and Dennis M. Healy, J. (1994). Computing fourier transforms and convolutions on the 2-sphere, *Advances in Applied Mathematics* 15(2): 202–250.
- Ehlgen, T., Pajdla, T. and Ammon, D. (2008). Eliminating blind spots for assisted driving, *IEEE Transactions on Intelligent Transportation Systems* 9(4): 657–665.
- Fischler, M. A. and Bolles, R. C. (1981). Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography, *Communications of the ACM* 24: 381–395.
- Geyer, C. and Daniilidis, K. (2001). Catadioptric projective geometry, *International Journal of Computer Vision* 45(3): 223–243.
- Goedeme, T., Tuytelaars, T., Van Gool, L., Vanacker, G. and Nuttin, M. (2005). Omnidirectional sparse visual path following with occlusion-robust feature tracking, *Workshop on Omnidirectional Vision, Camera Networks and Non-classical Cameras - OMNIVIS*.
- Hadj-Abdelkader, H., Malis, E. and Rives, P. (2008). Spherical image processing for accurate visual odometry with omnidirectional cameras, *Workshop on Omnidirectional Vision, Camera Networks and Non-classical Cameras - OMNIVIS*.
- Hansen, P., Corke, P., Boles, W. and Daniilidis, K. (2007a). Scale invariant feature matching with wide angle images, *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 1968–1994.
- Hansen, P., Corke, P., Boles, W. and Daniilidis, K. (2007b). Scale-invariant features on the sphere, *International Conference on Computer Vision*, pp. 1–8.

- Harris, C. and Stephens, M. J. (1988). A combined corner and edge detector, Alvey Conference, pp. 147–152.
URL: http://www.csse.uwa.edu.au/~pk/Research/MatlabFns/Spatial/Docs/Harris/A_Corner.html
- Kadir, T. and Brady, M. (2001). Saliency, scale and image description, International Journal of Computer Vision 45(2): 83–105.
- Lindeberg, T. (1998). Feature detection with automatic scale selection, International Journal of Computer Vision 30(2): 79–116.
- Lowe, D. G. (2004). Distinctive image features from scale-invariant keypoints, International Journal of Computer Vision 60(2): 91–110.
- Matas, J., Chum, O., Urba, M. and Pajdla, T. (2002). Robust wide baseline stereo from maximally stable extremal regions, Proceedings of the British Machine Vision Conference, pp. 384–393.
- Menegatti, E., Pretto, A., Scarpa, A. and Pagello, E. (2006). Omnidirectional vision scan matching for robot localization in dynamic environments, IEEE Transactions on Robotics 22(3): 523–535.
- Mikolajczyk, K. and Schmid, C. (2005). A performance evaluation of local descriptors, IEEE Transactions on Pattern Analysis and Machine Intelligence 27(10): 1615–1630.
- Rodrigues, O. (1840). Des lois géométriques qui régissent les déplacements d’un système solide dans l’espace, et de la variation des coordonnées provenant de ces déplacements considérés indépendamment des causes qui peuvent les produire, Journal de Mathématiques Pures et Appliquées 5: 380–440.
- Scaramuzza, D. and Siegwart, R. (2008). Appearance-guided monocular omnidirectional visual odometry for outdoor ground vehicles, IEEE Transactions on Robotics 24(5): 1015–1026.
- Sirmacek, B. and Unsalan, C. (2009). Urban-area and building detection using sift keypoints and graph theory, IEEE Transactions on Geoscience and Remote Sensing 47(4): 1156–1167.

- Smith, S. M. and Brady, J. M. (1997). Susan - a new approach to low level image processing, *International Journal of Computer Vision* 23: 45–78.
- Tamimi, H., Andreasson, H., Treptow, A., Duckett, T. and Zell, A. (2006). Localization of mobile robots with omnidirectional vision using particle filter and iterative sift, *Robotics and Autonomous Systems* 54(9): 758 – 765. Selected papers from the 2nd European Conference on Mobile Robots (ECMR '05).
URL: <http://www.sciencedirect.com/science/article/B6V16-4KJ0SSH-1/2/a0cb5ac27fe99eec909f6767d486baa2>
- Tuytelaars, T. and Mikolajczyk, K. (2007). Local invariant feature detectors: A survey, *Foundations and Trends in Computer Graphics and Vision* 3(3): 177–280.
URL: <http://dx.doi.org/10.1561/06000000017>
- Tygart, M. (2008). Fast algorithms for spherical harmonic expansions, II, *Journal of Computational Physics* 227(8): 4260–4279.
- Valgren, C. and Lilienthal, A. (2007). SIFT, SURF and seasons: Long-term outdoor localization using local features, *Proceedings of the European Conference on Mobile Robots (ECMR)*.
- Van Gool, L. J., Moons, T. and Ungureanu, D. (1996). Affine/photometric invariants for planar intensity patterns, *Computer Vision - ECCV'96*, pp. 642–651.
URL: <http://dx.doi.org/10.1007/BFb0015574>
- Vincenty, T. (1975). Direct and inverse solutions of geodesics on the ellipsoid with application of nested equations, *Survey Review* XXIII(176): 88–93.
- Yuen, D. and MacDonald, B. (2005). Vision-based localization algorithm based on landmark matching, triangulation, reconstruction, and comparison, *IEEE Transactions on Robotics* 21(2): 217–226.
- Zabih, R. and Woodfill, J. (1994). Non-parametric local transforms for computing visual correspondence, *Computer Vision - ECCV'94, Lecture Notes in Computer Science*, Springer Berlin/Heidelberg, pp. 151–158.

Table 1: Statistics for correct matchings between 31 omnidirectional images (23 parabolic and 8 spherical) and their rotated noisy version. The rotation is done around the x axis and its value is randomly chosen between 0 and 2π . The added noise is zero mean Gaussian noise with standard deviation 0.05. For the original parabolic omnidirectional images, the mean number of extracted points is 217 (std. deviation = 82), and for their noisy rotated versions is 223 (std. deviation = 99), with a mean repeatability of 0.743 (std. deviation = 0.081). For the original spherical images, the mean number of extracted points is 400 (std. deviation = 132), and for their noisy rotated versions is 394 (std. deviation = 120), with a mean repeatability of 0.79 (std. deviation = 0.033). The values in the table without parenthesis are the mean values, and the ones with parenthesis are the standard deviations.

Distance Threshold	Correct Matchings			
	Parabolic Omnidirectional Images		Spherical Images	
	Percentage	Absolute Value	Percentage	Absolute Value
0.05	1.00 (0.01)	11 (10)	1.00 (0.00)	27 (44)
0.10	1.00 (0.01)	24 (16)	1.00 (0.01)	52 (62)
0.15	1.00 (0.01)	32 (19)	1.00 (0.01)	70 (74)
0.20	1.00 (0.01)	39 (23)	1.00 (0.01)	82 (77)
0.25	0.99 (0.03)	44 (25)	1.00 (0.01)	90 (81)
0.30	0.98 (0.04)	48 (27)	1.00 (0.00)	100 (84)
0.35	0.97 (0.05)	52 (29)	0.99 (0.01)	107 (87)
0.40	0.96 (0.06)	56 (31)	0.99 (0.02)	115 (89)
0.45	0.96 (0.05)	60 (32)	0.97 (0.02)	120 (92)
0.50	0.94 (0.06)	64 (34)	0.97 (0.03)	126 (92)
0.55	0.93 (0.06)	67 (35)	0.95 (0.03)	133 (93)
0.60	0.90 (0.07)	71 (37)	0.93 (0.03)	138 (95)
0.65	0.87 (0.08)	75 (38)	0.91 (0.04)	144 (98)
0.70	0.84 (0.08)	79 (39)	0.89 (0.04)	150 (100)
0.75	0.80 (0.09)	82 (40)	0.84 (0.06)	156 (101)
0.80	0.75 (0.11)	85 (41)	0.79 (0.07)	164 (105)
0.85	0.70 (0.13)	89 (42)	0.71 (0.10)	169 (107)
0.90	0.63 (0.13)	94 (44)	0.64 (0.11)	178 (109)
0.95	0.56 (0.14)	99 (45)	0.56 (0.12)	185 (111)