

Scale Invariant Feature Transform on the Sphere: Theory and Applications

Javier Cruz-Mota · Iva Bogdanova · Benoît Paquier ·
Michel Bierlaire · Jean-Philippe Thiran

Received: 29 April 2009 / Accepted: 21 October 2011 / Published online: 10 November 2011
© Springer Science+Business Media, LLC 2011

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 two 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 mapping is introduced and an algorithm for its estimation is given. This mapping allows to extract objects from an omnidirectional image given their SIFT descriptors in a planar image. Several

experiments, confirming the promising and accurate performance of the system, are conducted.

Keywords Omnidirectional vision · (Spherical) image processing · Feature extraction · Object detection · SIFT · Matching

1 Introduction

Omnidirectional vision has become an important 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 for this kind of cameras yet. Usually, techniques designed for planar images are applied on omnidirectional images. For example on panoramic images, i.e. omnidirectional images mapped on a cylinder (Yuen and MacDonald 2005; Bur et al. 2006), but this is not geometrically 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 hybrid 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

J. Cruz-Mota (✉) · M. Bierlaire
Transport and Mobility Laboratory (TRANSP-OR), École
Polytechnique Fédérale de Lausanne (EPFL), 1015 Lausanne,
Switzerland
e-mail: javier.cruz@epfl.ch

M. Bierlaire
e-mail: michel.bierlaire@epfl.ch

I. Bogdanova
Pattern Recognition Laboratory (PARLAB), École Polytechnique
Fédérale de Lausanne (EPFL), Rue A.L. Breguet 2, 2000
Neuchâtel, Switzerland
e-mail: iva.bogdanova@epfl.ch

Present address:

I. Bogdanova
Embedded Systems Lab (ESL), École Polytechnique Fédérale de
Lausanne (EPFL), 1015 Lausanne, Switzerland

B. Paquier · J.-P. Thiran
Signal Processing Laboratory 5, École Polytechnique Fédérale de
Lausanne (EPFL), 1015 Lausanne, Switzerland

B. Paquier
e-mail: benoit.paquier@epfl.ch

J.-P. Thiran
e-mail: jp.thiran@epfl.ch

robust against changes in the image, such as noise, perspective transformations, illumination changes, etc. The location of these points is often used for extraction of local image descriptors. This is 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 key-points are often used for the computation of local descriptors, which are 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). An exhaustive comparison of local descriptors has been given in 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

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

where $g(x, y, \sigma)$ denotes a two-dimensional Gaussian kernel with standard deviation σ . This is based on the work of Lindeberg (1998), and the convolution of an image with $\psi(x, y, \sigma)$ 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 $L(x, y, t) : \mathbb{R}^2 \times \mathbb{R}^+ \rightarrow \mathbb{R}$ of an image $I(x, y)$ can be equivalently defined in two different ways. The first one is

the evolution over time of the heat distribution $I(x, y)$ in an infinite homogeneous medium:

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

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

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

This scale-space representation of an image is efficiently computed directly 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 three-dimensional 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. 2010).

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 an important 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 their 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 Lilienthal 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

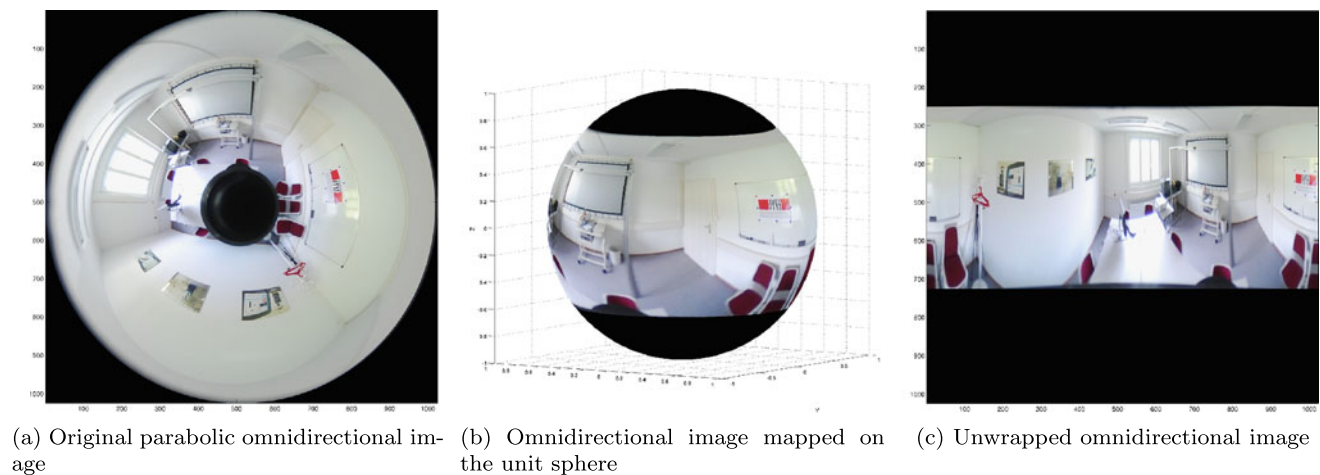


Fig. 1 Example of mapping a parabolic omnidirectional image on the sphere. The unwrapped spherical image (c) is often used for visualisation purposes

Geyer and Daniilidis (2001), where the authors showed that the most common catadioptric omnidirectional images (elliptic, parabolic and hyperbolic) can be bijectively mapped on the surface of a sphere. In particular, for the case of parabolic images, a parabolic projection is equivalent to the composition of normalisation to the unit sphere followed by stereographic projection (see Fig. 1 for an example). 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. (2007a, 2007b) 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. Also, in Mauthner et al. (2006) an interest region matching in omnidirectional images, which uses virtual camera planes, has been developed.

1.2 Motivation: Why SIFT in Spherical Coordinates?

The intuitive approach for defining and performing a SIFT on omnidirectional images is to first map it on a panoramic image (i.e. in cylindrical coordinates) and then to apply the conventional SIFT algorithm. In fact, using the same reasoning, the classical SIFT has been applied to unwrapped omnidirectional images (Goedeme et al. 2005; Tamimi et al. 2006; Valgren and Lilienthal 2007; Scaramuzza and Siegwart 2008). The difficulties in this case come when there is information in the extremities of the omnidirectional image. Such images are obtained by spherical omnidirectional sensors.

On the one hand, the images obtained by omnidirectional sensors suffer under significant deformations. Specific mappings, like panoramic or log-polar, attempt to reduce somehow the distortions but do not succeed completely. A natural choice of a non-deformed domain for the full sphere of view, where there are no limitations on the zenithal range, is the sphere $S^2 \in \mathbb{R}^3$. This is the natural domain of definition of the plenoptic function and the processing of the visual information on the sphere correctly handles the information without introducing any distortion (Tosic and Frossard 2009). On the other hand, the scale-invariant feature transform is based on distinctive invariant features from images for further matching. The features are invariant to image scale and rotation.

Considering the spherical geometry, which is the natural manifold for any omnidirectional image, we need to first recall the basic affine spherical transformations. In general, two types of transformations on the sphere are distinguished: motions (displacements) and dilations (scalings). Concerning the motions, there are three possible rotations. In particular, these are rotations by angles $\varphi \in [0, 2\pi)$, $\psi \in [0, 2\pi)$ and $\theta \in [0, \pi]$. In other words, rotations by φ are those around the x_0 -axis; rotations by θ are those around x_2 -axis, and rotations by ψ , are rotations of the point on the sphere around itself (see Fig. 2). The dilations affect the angle θ . Then, through the plenoptic function, we can relate translations and rotations of an object in the 3D space to rotations on the sphere, when the distance between the object and the origin of the sphere remains constant; or to rotations and dilations on the sphere, when this distance varies. From here, it is obvious that the notions of rotation and dilation on the plane cannot be directly applied to the sphere, since for instance changes on the angle θ introduce a deformation on the observed object that does not correspond to any affine transformation on the 3D space. This implies, in particular,

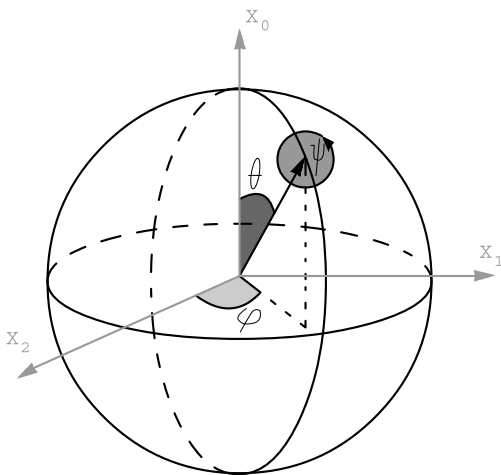


Fig. 2 Rotations on the sphere

that the deformations suffered by an observed object depend on the position of this object on the omnidirectional image. That is why we cannot apply the standard SIFT paradigm to a complex data that is defined in spherical coordinates (as for instance, the omnidirectional image after it has been mapped onto the sphere).

Finally, from here it is clear that applying the standard SIFT on unwrapped omnidirectional image is locally valid, i.e. for big radius of curvature and, consequently, at small scales. Therefore, what concerns the omnidirectional images, we argue that SIFT must take place in spherical geometry. Having a SIFT algorithm that operates directly on the sphere is equivalent to preserving the spherical affine transformations and only in this conditions the transformations are geometrically correct.

1.3 Contributions

The paper has two main contributions, 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, 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 Fig. 1b). The creation procedure of the spherical scale-space is speeded up by successive downsampling of the input image for each octave. This down-sampling generates an aliasing effect when the Spherical Fourier transform is applied at the corresponding level. For this reason, an anti-aliasing criterion is defined to decide whether an image is down-sampled or not.

Our second main contribution concerns the matching setup. In this paper we propose two types of descriptors. The first is used for matching between two spherical images, the second for matching a spherical and a planar image. Both

descriptors can be very useful when working, for example, with hybrid camera networks. In such a case, these descriptors can easily help registering data from all the components of the network.

Finally, we introduce a mapping between planar and spherical images. This mapping sends the contour or regions of an object in a planar image to a spherical one and vice versa. The parameters of this mapping 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 Toolbox.³

This paper is organised as follows. In Sect. 2, the mathematical aspects of the interest point extraction on the sphere are exposed. In Sect. 3, the proposed algorithm is described in detail, as well as the two proposed descriptors. Then, in Sect. 4, a “planar to spherical” mapping is defined and a method for its estimation is given. In Sect. 5, several experimental results are presented. Finally, in Sect. 6, some conclusions and potential lines for future research are given.

2 Spherical Scale-Space

2.1 Spherical Geometry

The 2-sphere ($S^2 \in \mathbb{R}^3$) is a compact manifold of constant positive curvature. In spherical coordinates, each point on the sphere is a three-dimensional vector

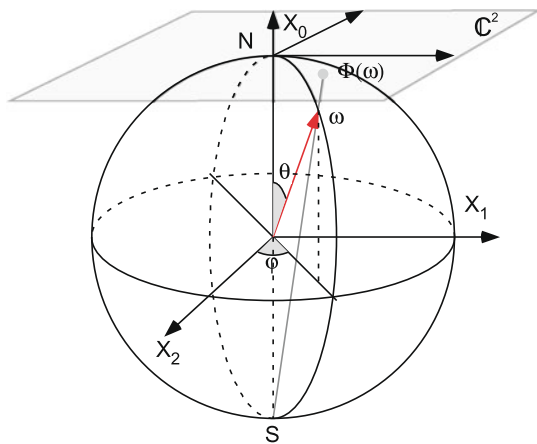
$$\omega = (x_0, x_1, x_2) \equiv (r \cos \theta, r \sin \theta \sin \varphi, r \sin \theta \cos \varphi),$$

with $r \in (0, \infty)$, $\theta \in [0, \pi]$ and $\varphi \in (0, 2\pi]$ as illustrated in Fig. 3a. Figure 3b also illustrates the so called stereographic projection from the South Pole, a projection that maps any point of the sphere onto a point of the tangent plane at the North Pole. If we take the sphere S^2 as the Riemannian

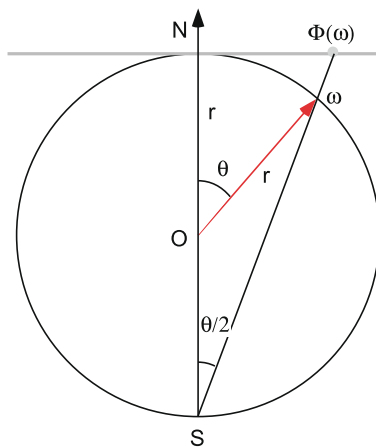
¹<http://transp-or2.epfl.ch/pagesPerso/javierFiles/software.php>.

²<http://rhea.tele.ucl.ac.be/yawtb>.

³<http://vision.ece.ucsb.edu/~zuliani/Code/Packages/RANSAC/>.



(a) Spherical coordinates



(b) Cross-section of a stereographic projection through the South Pole

Fig. 3 Spherical geometry

sphere ($r = 1$) and the tangent plane as the complex plane \mathbb{C}^2 , then the stereographic projection is a bijection given by

$$\Phi(\omega) = 2 \tan \frac{\theta}{2} (\cos \varphi, \sin \varphi), \tag{4}$$

where $\omega \equiv (\theta, \varphi)$, $\theta \in [0, \pi]$, $\varphi \in [0, 2\pi)$.

2.2 Fourier transform on the sphere

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

$$(f * h)(\omega) = \int_{r \in SO(3)} f(r\eta)h(r^{-1}\omega)dr, \tag{5}$$

where $\omega \equiv (\theta, \varphi) \in S^2$, $\theta \in [0, \pi]$, $\varphi \in [0, 2\pi)$ (see Fig. 3a) and η is the north pole. Equation (5) is hard to compute, but as it was demonstrated by Driscoll and Healy (1994), the convolution of two spherical functions $f, h \in L^2(S^2)$ can be

calculated more efficiently as the point-wise product of their spherical Fourier transforms:

$$(\widehat{f * h})(l, m) = 2\pi \sqrt{\frac{4\pi}{2l + 1}} \widehat{f}(l, m)\widehat{h}(l, 0), \tag{6}$$

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 $\widehat{f}(l, m)$ of the expression

$$f(\theta, \varphi) = \sum_{l \geq 0} \sum_{|m| \leq l} \widehat{f}(l, m)Y_l^m(\theta, \varphi), \tag{7}$$

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 Fig. 3a). The spherical harmonics can be factorized as

$$Y_l^m(\theta, \varphi) = k_{l,m} P_l^m(\cos \theta) e^{im\varphi}, \tag{8}$$

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

$$k_{l,m} = \sqrt{\frac{2l + 1}{4\pi} \frac{(l - m)!}{(l + m)!}} \tag{9}$$

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

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

$$\begin{aligned} \widehat{f}(l, m) &= \langle f, Y_l^m \rangle \\ &= k_{l,m} \int_{S^2} d\mu(\omega) \overline{Y_l^m(\omega)} f(\omega), \\ &= k_{l,m} \int_0^{2\pi} \int_0^\pi \overline{Y_l^m(\theta, \varphi)} f(\theta, \varphi) \sin \theta d\theta d\varphi \end{aligned} \tag{10}$$

where $d\mu(\omega) = \sin \theta d\theta d\varphi$ is $SO(3)$ invariant measure on S^2 . Using (8) and (10), 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.

2.3 Spherical DoG as a Scale-Space

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. Let us note that we need to pass through the spherical Fourier domain because convolution on the sphere in

spatial domain (3D) is hard (almost impossible) to compute. See (5) and (6) in Sect. 2.2 for details. This is not the case of the scale-space representation of a planar image, given the separability of the Gaussian filter and the simplicity of the planar (2D) convolution.

In Bulow (2004), the author derives this function as a Green function of the heat equation (2) over S^2 , obtaining

$$g^{S^2}(\theta, \varphi, \sigma) = \sum_{l \in \mathbb{N}} \sqrt{\frac{2l+1}{4\pi}} Y_{l0}(\theta, \varphi) e^{-\frac{l(l+1)\sigma^2}{2}}, \tag{12}$$

$$\widehat{g^{S^2}}(l, m, \sigma) = \sqrt{\frac{2l+1}{4\pi}} e^{-\frac{l(l+1)\sigma^2}{2}}, \tag{13}$$

where g^{S^2} denotes the spherical Gaussian function. Therefore, using (6), the spherical Fourier transform of the scale-space representation of an omnidirectional image mapped on the sphere, $I(\theta, \varphi)$, is

$$\widehat{L^{S^2}}(l, m, \sigma) = \widehat{I}(l, m) e^{-\frac{l(l+1)\sigma^2}{2}} \tag{14}$$

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

$$L^{S^2}(\theta, \varphi, \sigma) = I(\theta, \varphi) * g^{S^2}(\theta, \varphi, \sigma), \tag{15}$$

is the spherical scale-space representation of this image. Finally, the spherical DoG is computed as

$$\psi^{S^2}(\theta, \varphi, \sigma) = L^{S^2}(\theta, \varphi, k\sigma) - L^{S^2}(\theta, \varphi, \sigma). \tag{16}$$

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 LSD of points extracted from different 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 in 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 $d \in [0, 1]$.

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 spherical DoG
 - 4: $E \leftarrow$ Local extrema of spherical DoG
 - 5: **for** each $E_i \in E$ **do**
 - 6: Compute LSD and/or LPD of E_i
 - 7: **end for**
-

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 this 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, $\theta \in [0, \pi]$.

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 (15) iteratively, i.e.

$$L^{S^2}(\theta, \varphi, \sigma_i) = L^{S^2}(\theta, \varphi, \sigma_{i-1}) * g^{S^2}(\theta, \varphi, \tilde{k}_i \sigma_0), \tag{17}$$

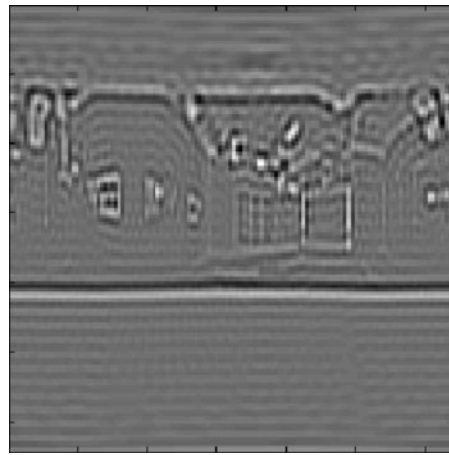
where σ_0 is the initial scale and \tilde{k}_i is chosen in such a way that two neighbouring scales in the spherical scale-space representation are separated by a constant multiplicative factor $k = 2^{1/5}$ (in order to have a constant number S of images per octave). Therefore, $\sigma_i = k\sigma_{i-1} = k^i \sigma_0$ and using the semi-group property of the spherical scale-space representation, we have that $(k^i \sigma_0)^2 + (\tilde{k}_i \sigma_0)^2 = (k^{i+1} \sigma_0)^2$, and so $\tilde{k}_i = k^i \sqrt{k^2 - 1}$. These expressions are also valid in the planar case.

The spherical scale-space representation process is speeded up by downsampling the image by two, instead of increasing the scale σ , each time a complete octave of $\psi^{S^2}(\theta, \varphi, \sigma)$ is 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 taken 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 following condition is tested:

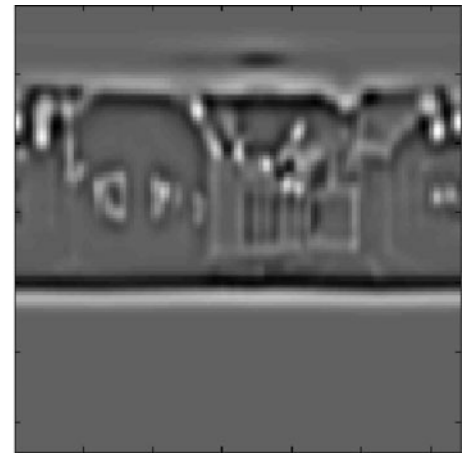
$$e^{-\frac{nH(nH+1)(\sigma_0/k)^2}{8}} \leq e^{-1}, \tag{18}$$

where nH is the new height of the image after reducing its size. This condition assures that the exponential part of (13) remains small for the biggest value of l . If (18) is not fulfilled, instead of reducing the image size for the next octave,

Fig. 4 Example of the effect of the anti-aliasing strategy for the ψ^{S^2} computation of the image in Fig. 1b. 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 $S = 3$



(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

σ is increased and the image size is reduced after the convolution. Aliasing effects can still appear if they are present in the first computed spherical Fourier transform, or if σ increases considerably ((18) 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 Fig. 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 $o = -1$.

3.2 Extrema Extraction

Interest points are local extrema of $\psi^{S^2}(\theta, \varphi, \sigma)$ (16). A local extreme is a point on the spherical grid whose value is bigger (smaller) than 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

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 (18) is satisfied **then**
- 11: Down-sample 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 σ '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 down-sampled by 2^n
- 16: **end if**
- 17: **end for**

expansion of (16):

$$\psi^{S^2}(\theta, \varphi, \sigma) \simeq \psi^{S^2}(\theta_i, \varphi_i, \sigma_i) \frac{\partial \psi^{S^2}}{\partial \Theta} \Big|_{\omega_i} \delta_{\omega_i} + \frac{1}{2} \delta_{\omega_i}^\top \frac{\partial^2 \psi^{S^2}}{\partial \Theta^2} \Big|_{\omega_i} \delta_{\omega_i}, \quad (19)$$

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

$$\begin{aligned} \frac{\partial f}{\partial x^j} &= f_{x^j}(x^1, x^2, \dots, x^j, \dots, x^n) \\ &= \frac{f(\dots, x^j + \Delta_{x^j}, \dots) - f(\dots, x^j - \Delta_{x^j}, \dots)}{2\Delta_{x^j}}. \end{aligned} \quad (20)$$

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 (19) with respect to δ_{ω_i} , the offset $\tilde{\delta}_{\omega_i}$ to the extreme of the fitted function is obtained

$$\tilde{\delta}_{\omega_i} = -\left(\frac{\partial^2 \psi^{S^2}}{\partial \Theta^2}\right)^{-1} \frac{\partial \psi^{S^2}}{\partial \Theta}. \quad (21)$$

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, which is a common strategy in implementations of SIFT. This is due to the fact that a displacement in the σ direction changes completely the conditions where the quadratic function is fitted. 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

$$|\psi^{S^2}(\tilde{\omega}_i)| > \frac{0.02}{k^s 2^\sigma} \quad (22)$$

is not satisfied, then $\tilde{\omega}_i$ is discarded. The threshold value in (22) 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). Principal curvatures of a surface at a given point p are the maximum and minimum curvatures of the resulting curves when intersecting the surface with all the planes containing the normal vector to the surface at p . This test 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

$$\frac{\text{trace}(H^{S^2})^2}{\det(H^{S^2})} < \frac{(r+1)^2}{r}, \quad (23)$$

where

$$H^{S^2} = \begin{pmatrix} \psi_{\theta\theta}^{S^2} & \psi_{\theta\varphi}^{S^2} \\ \psi_{\theta\varphi}^{S^2} & \psi_{\varphi\varphi}^{S^2} \end{pmatrix}, \quad (24)$$

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 (22) and (23) 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

```

then it is discarded. Note also that as we are working on the unit sphere in spherical coordinates, derivatives with respect to φ have a $1/\sin\theta$ coefficient. The full extrema extraction procedure is detailed 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 scale-space representation of the image (see Sects. 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 as

$$\alpha(\theta, \varphi, \sigma) = \arctan\left(\frac{L_{\varphi}^{S^2}(\theta, \varphi, \sigma)}{L_{\theta}^{S^2}(\theta, \varphi, \sigma)}\right). \quad (25)$$

Then, for each considered extreme of the ψ^{S^2} , (25) 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, $p_1 \equiv (\theta_1, \varphi_1)$ and $p_2 \equiv (\theta_2, \varphi_2)$, needs to be calculated. It can be obtained using the Vincenty’s formula (Vincenty 1975):

$$d(p_1, p_2) = \arctan\left(\frac{\sqrt{A^2 + B^2}}{C}\right), \tag{26}$$

where

$$A = \sin \theta_1 \sin \Delta \varphi, \tag{27}$$

$$B = \sin \theta_2 \cos \theta_1 - \cos \theta_2 \sin \theta_1 \cos \Delta \varphi, \tag{28}$$

$$C = \cos \theta_2 \cos \theta_1 + \sin \theta_2 \sin \theta_1 \cos \Delta \varphi, \tag{29}$$

$$\Delta \varphi = \varphi_1 - \varphi_2. \tag{30}$$

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,

$$m(\theta, \varphi, \sigma) = \sqrt{L_\varphi^{S^2}(\theta, \varphi, \sigma)^2 + L_\theta^{S^2}(\theta, \varphi, \sigma)^2}, \tag{31}$$

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 has been 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. This results in multiple principal orientations for the same point.

Then, LSD are computed taking their corresponding principal orientations as reference. This descriptor is a three-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 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 bin value corresponds to the weighted sum of gradient magnitudes of points at the spatial and orientation defined by the bin. The weight value is defined 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

$$v^{\text{Rot}} = v \cos \alpha + u \times v \sin \alpha + u \cdot v(1 - \cos \alpha)u, \tag{32}$$

where the vectors u, v and v^{Rot} are considered in Cartesian coordinates, and the vector v^{Rot} is the result of rotating α degrees the vector v around u .

Algorithm 4 Algorithm for the computation of LSD

- 1: LSD $\leftarrow \emptyset$ the set of Local Spherical Descriptors
- 2: **for** each considered extreme of $\psi^{S^2}, (\theta_i, \varphi_i, \sigma_i)$ **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: MAX \leftarrow maximum histogram value
- 7: **for** each bin value $\geq 0.8\text{MAX}$ **do**
- 8: Fit a parabola around this bin
- 9: $b \leftarrow$ 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 b degrees
- 11: Compute orientations and gradient norms inside this region with respect to b
- 12: LSD _{i} \leftarrow Compute 3-dimensional histogram
- 13: LSD $\leftarrow \{\text{LSD}, \text{LSD}_i\}$
- 14: **end for**
- 15: **end for**

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 computing Local Spherical Descriptors is summarised in Algorithm 4.

3.4 Local Planar Descriptor (LPD)

Local Planar Descriptors (LPD) allow to match points extracted from a spherical image, using Algorithms 2 and 3, and SIFT descriptors of points extracted from planar images. This is of great importance, considering that 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 $p_i \equiv (\theta_i, \varphi_i)$ to be the centre of this planar approximation, which is the stereographic projection on the tangent plane of the sphere at p_i through its antipodal point. This projection of $L^{S^2}(\theta, \varphi, \sigma_i)$ around p_i can be seen as a local approximation of $L(x, y, \sigma)$. In other words, for a point $p_i \equiv (\theta_i, \varphi_i)$, extracted from the spherical image at scale σ_i , a squared window centred at 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 $(\theta_i + \pi/2, \varphi_i + \pi)$ to the plane tangent at p_i . The projected points are linearly interpolated in order to obtain a planar image whose cartesian range is $[-2 \tan \frac{6\sigma_i}{2}, 2 \tan \frac{6\sigma_i}{2}] \times$

Algorithm 5 Algorithm for the computation of LPD

- 1: LPD $\leftarrow \emptyset$ the set of Local Planar Descriptors
- 2: **for** each considered extreme of $\psi^{S^2}, (\theta_i, \varphi_i, \sigma_i)$ **do**
- 3: $L(x, y, \sigma_i^{pl}) \leftarrow$ 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: $LPD_i \leftarrow$ SIFT descriptor of $L(x, y, \sigma_i^{pl})$ at $(x, y) = (0, 0)$
- 5: LPD $\leftarrow \{LPD, LPD_i\}$
- 6: **end for**

$[-2 \tan \frac{6\sigma_i}{2}, 2 \tan \frac{6\sigma_i}{2}]$ and with a pixel spacing of $2 \tan \frac{\pi}{2H}$. H is the height of $L^{S^2}(\theta, \varphi, \sigma_i)$. The equivalent σ_i in the obtained planar image is given below:

$$\sigma_i^{pl} = \frac{\tan \frac{\sigma_i}{2}}{\tan \frac{\pi}{2H}}. \tag{33}$$

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

4 Planar to Spherical Mapping

As mentioned 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 from an object in a planar image to their corresponding points in a spherical image. We suppose that the object is rigid and planar, because only its projection on an image is known. The transfer function, together with the planar to spherical matching, can segment objects in omnidirectional images given their segmentation in a planar image or vice versa.

Let us consider two matched points, the first $p_i^{pl} \equiv (x_{2i}^{pl}, x_{1i}^{pl})$ in a planar image and the second $p_j^{S^2} \equiv (x_{2j}^{S^2}, x_{1j}^{S^2}, x_{0j}^{S^2})$ in a spherical image, both in cartesian coordinates. The idea is to find a linear transformation H that sends the point in the planar image p_i^{pl} , to a point q_{ij} in three-dimensional space, with projection to the unit sphere $p_j^{S^2}$ (see Fig. 5). The linearity of H is given by the rigidity assumption. Let us note that only the transformation from p_i^{pl} to q_{ij} is linear, but not the total mapping from p_i^{pl} to $p_j^{S^2}$. In other words, we look for a 3×3 matrix H that satisfies

$$p_j^{S^2} = \frac{q_{ij}}{\|q_{ij}\|} = \frac{H \tilde{p}_i^{pl}}{\|H \tilde{p}_i^{pl}\|}, \tag{34}$$

where $\|\cdot\|$ denotes the 2-norm and \tilde{p}_i^{pl} is an embedding of p_i^{pl} in \mathbb{R}^3 (more details follow). For estimating H , the planar image is placed tangentially to the sphere where the omnidirectional image is mapped. The central point of the planar

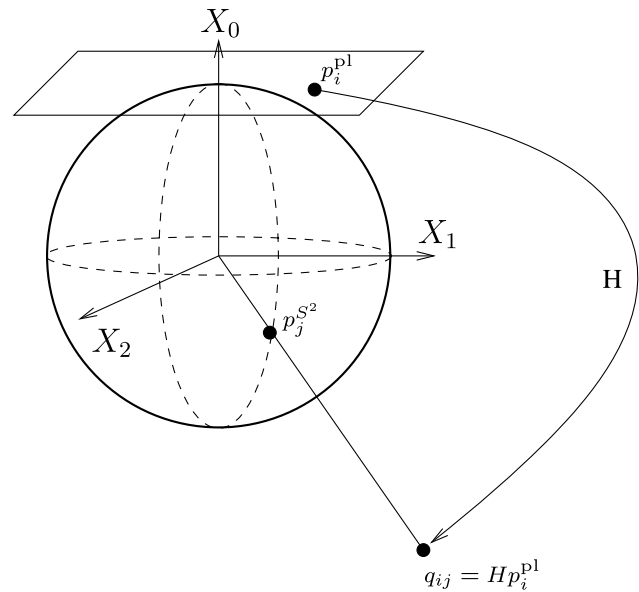


Fig. 5 Graphical sketch of the mapping

image is the contact point with the sphere. In this way, a point $p_i^{pl} \equiv (x_{2i}^{pl}, x_{1i}^{pl})$ of the planar image is embedded in \mathbb{R}^3 as $\tilde{p}_i^{pl} \equiv (x_{2i}^{pl}, x_{1i}^{pl}, 1)$. Then, the fact that $p_j^{S^2}$ and $H \tilde{p}_i^{pl}$ must be collinear is exploited forcing their vectorial product to be zero, i.e. $p_j^{S^2} \times H \tilde{p}_i^{pl} = 0$. The latter condition generates three equations, one for each of the components of the resulting vector of the cross product:

$$-x_{0j}^{S^2} x_{2i}^{pl} h_{21} - x_{0j}^{S^2} x_{1i}^{pl} h_{22} - x_{0j}^{S^2} h_{23} + x_{1j}^{S^2} x_{2i}^{pl} h_{31} + x_{1j}^{S^2} x_{1i}^{pl} h_{32} + x_{1j}^{S^2} h_{33} = 0, \tag{35}$$

$$x_{0j}^{S^2} x_{2i}^{pl} h_{11} + x_{0j}^{S^2} x_{1i}^{pl} h_{12} + x_{0j}^{S^2} h_{13} - x_{2j}^{S^2} x_{2i}^{pl} h_{31} - x_{2j}^{S^2} x_{1i}^{pl} h_{32} - x_{2j}^{S^2} h_{33} = 0, \tag{36}$$

$$-x_{1j}^{S^2} x_{2i}^{pl} h_{11} - x_{1j}^{S^2} x_{1i}^{pl} h_{12} - x_{1j}^{S^2} h_{13} + x_{2j}^{S^2} x_{2i}^{pl} h_{21} + x_{2j}^{S^2} x_{1i}^{pl} h_{22} + x_{2j}^{S^2} h_{23} = 0, \tag{37}$$

where the elements of the matrix H are distributed as

$$H = \begin{pmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{pmatrix} \tag{38}$$

Consequently, if (35), (36) and (37) are expressed in terms of h_{lm} and all the resulting equations for each pair of matched points are put together, a system of equations of the form $A\mathbf{h} = \mathbf{0}$ is obtained, where

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

A is a $3N \times 9$ matrix and N is the number of points matched between the planar and spherical images. If the restriction

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 (35), (36) and (37) (use RANSAC to clean M of false matchings).

$\|\mathbf{h}\| = 1$ is considered, \mathbf{h} can be computed as the eigenvector of matrix A corresponding to the smallest eigenvalue. This eigenvector is the least squares estimator of the solution. Note that although each pair of matched points generates three equations, only two of them are linearly independent. This means that at least four non-collinear pairs of matched points are required for estimating the coefficients of the matrix H .

The estimation of this matrix results in a mapping $h : \mathbb{R}^2 \rightarrow S^2$ that sends points in the planar image to points in the spherical one as follows:

$$h(x, y) = \frac{H \begin{pmatrix} x_2 \\ x_1 \\ 1 \end{pmatrix}}{\left\| H \begin{pmatrix} x_2 \\ x_1 \\ 1 \end{pmatrix} \right\|}. \quad (39)$$

Mapping points of the spherical image onto points of the planar image, can also be done using the transformation H^{-1} and normalising the resulting point by its third component. In this way, a point of the form $(x_2, x_1, 1)$ is obtained.

The estimation of H using all the matched points would give bad results due to false matchings. For avoiding this, the chosen set of matched points for the estimation of H is selected using RANSAC. This procedure also softens the planar assumption. Indeed, RANSAC will treat non-coplanar points as outliers, since they will not fit correctly the model generated by H . The outline of the process for computing H is specified in Algorithm 6.

5 Experimental Results

In our experimental results we use two types of omnidirectional images: parabolic and spherical. Parabolic omnidirectional images are obtained by a catadioptric omnidirectional sensor: a parabolic mirror Kaidan EyeSee 360⁴ 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. After this mapping, the images cover a band of about 100 deg on the sphere. Spherical images are obtained with a Ladybug2 device⁵ and they cover 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 them. Both types of images have the same resolution, 1024×1024 .

In all tests, two interest points p_1 and p_2 that define two interest regions in two different images, I_1 and I_2 respectively, are considered as the same point if after transferring p_2 to I_1 , the overlap error computed using the intersection over union criterion (Mikolajczyk et al. 2005) is smaller than 0.5. The size of the interest regions considered to compute the overlap error is fixed by the scale at which each point is detected (see Sect. 3.3 for details).

This section is organised as follows. In Sect. 5.1, a comparison between standard and spherical SIFT has been done. An example where standard SIFT applied on omnidirectional images fails is given. In Sect. 5.2, the optimal parameters of the algorithm for omnidirectional images are deduced from several performed tests, and then some examples are shown. Finally, in Sect. 5.3, a matching test between an object on a planar image and several omnidirectional images containing the object is performed. In addition, the estimation of the planar to spherical mapping using matchings between planar and omnidirectional images is illustrated.

5.1 “Planar vs. Spherical” Scale Invariant Feature Transform

As commented in Sect. 1.2, at small scales and for points far from the poles, the standard SIFT algorithm can perform acceptably well. But it is important to be aware of this limitation, since as soon as this two hypothesis are not fulfilled, the standard SIFT fails and the extra computation cost of considering the geometry of the sensor needs to be paid. However, this extra cost will provide us more precision and invariance to the deformations that the spherical geometry introduces.

The weaknesses of standard SIFT, when applied to an omnidirectional image, have two origins. On the one hand, planar scale-space computation cannot be directly applied to the sphere. Indeed, the planar scale-space representation does not handle correctly scales on an omnidirectional image. This makes SIFT fail in feature extraction, especially in regions close to the poles, where the incorrectness of the scale-space representation is more obvious. On the other hand, the descriptors of standard SIFT do not take into account the deformations introduced by the sensor on the omnidirectional image. This makes these descriptors not very useful for matching, since they are not robust to changes on the captured objects.

⁴<http://www.kaidan.com>.

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

For showing these two problems, we will compare the results on several example images when standard SIFT is employed (as implemented in Vedaldi and Fulkerson 2008) and when our scale invariant feature transform on the sphere is applied. In all the cases, for obtaining comparable results, both algorithms are run with the same parameters, those proposed by Lowe (2004), $S = 3$ and $\sigma_0 = 1.6$. The octave -1 is not computed in any of the cases.

In the first example we illustrate the loss of extracted features due to the incorrect use of the planar scale-space computation on omnidirectional images. For showing that, we have compared the points extracted on a sequence of omnidirectional images by standard SIFT and by the proposed algorithm. These images were obtained in an office while the sensor points down a table where red objects are placed. One of these objects is moved throughout the sequence and thus it is perceived on the South Pole of the sphere, i.e. just right below the sensor. This object does not have any texture in order to minimise the number of extracted points due to noise in the image and internal texture of the object. The deformation introduced by the fact of moving the object towards a pole is clearly visible in the unwrapped version of the spherical image, in (θ, φ) coordinates (see for instance top image of Fig. 6). After applying the standard SIFT algorithm, in Fig. 6a, we can observe that in the first image (bottom), the red object has a point in the centre and points in the four corners. In the second image, the points in one of the corners and in the centre have disappeared. Finally, in the third image, only one point is extracted on the object. Nevertheless, in Fig. 6b, where the spherical SIFT is applied, in the first image (bottom), the red object has a point in the centre and points in three of its corners. In the second image, the object still has a point in the centre as well as in the three corners. And finally, in the third image, the red object has lost the point in the centre but still has the points in the three corners. In other words, the spherical SIFT extracts the same features on the object independently of its position on the omnidirectional image, while the standard SIFT fails.

In the second example we illustrate the low usefulness of standard SIFT descriptors, when computed on omnidirectional images without considering the spherical geometry. For showing that, we follow the same approach than in the previous example, but this time with an object with rich texture information. This guarantees the extraction by the standard SIFT algorithm of at least some features on the object, even when it is near the poles. In this case, the sequence is composed of two images obtained in an office with the sensor pointing to the ceiling, and the object is a person that moves slightly while standing close to the omnidirectional camera. Features are extracted in both images and afterwards descriptors are computed, using standard SIFT with standard SIFT descriptors and using our proposed scale invariant feature transform on the sphere with LSD. When

a matching is performed between standard SIFT descriptors, only a few are correctly matched (see Fig. 7b). Almost all the descriptors placed on a point that has changed its position between the two images are incorrectly matched. However, when the scale invariant feature transform on the sphere with LSD is employed, only a few descriptors are incorrectly matched (see Fig. 7a). With this example we can see that, independently of the extraction of feature points, ignoring the geometric deformation introduced by the omnidirectional sensor entails a poor robustness of the computed descriptors. Note that in Fig. 7, for ease of visualisation, we only plot matches between descriptors placed on the body of the subject, which is the only region of the image that changes.

5.2 “Omni vs. Omni” Repeatability and Matching

First of all, in order to test LSD matching, some parameters of the algorithm need to be fixed, essentially S and σ_0 . In order to choose the values of S and σ_0 that maximise repeatability, repeatability tests have been performed on 28 real omnidirectional images. These images were taken in three different days and in two different locations, producing images under very different conditions. Some examples are shown in Figs. 8 and 9.

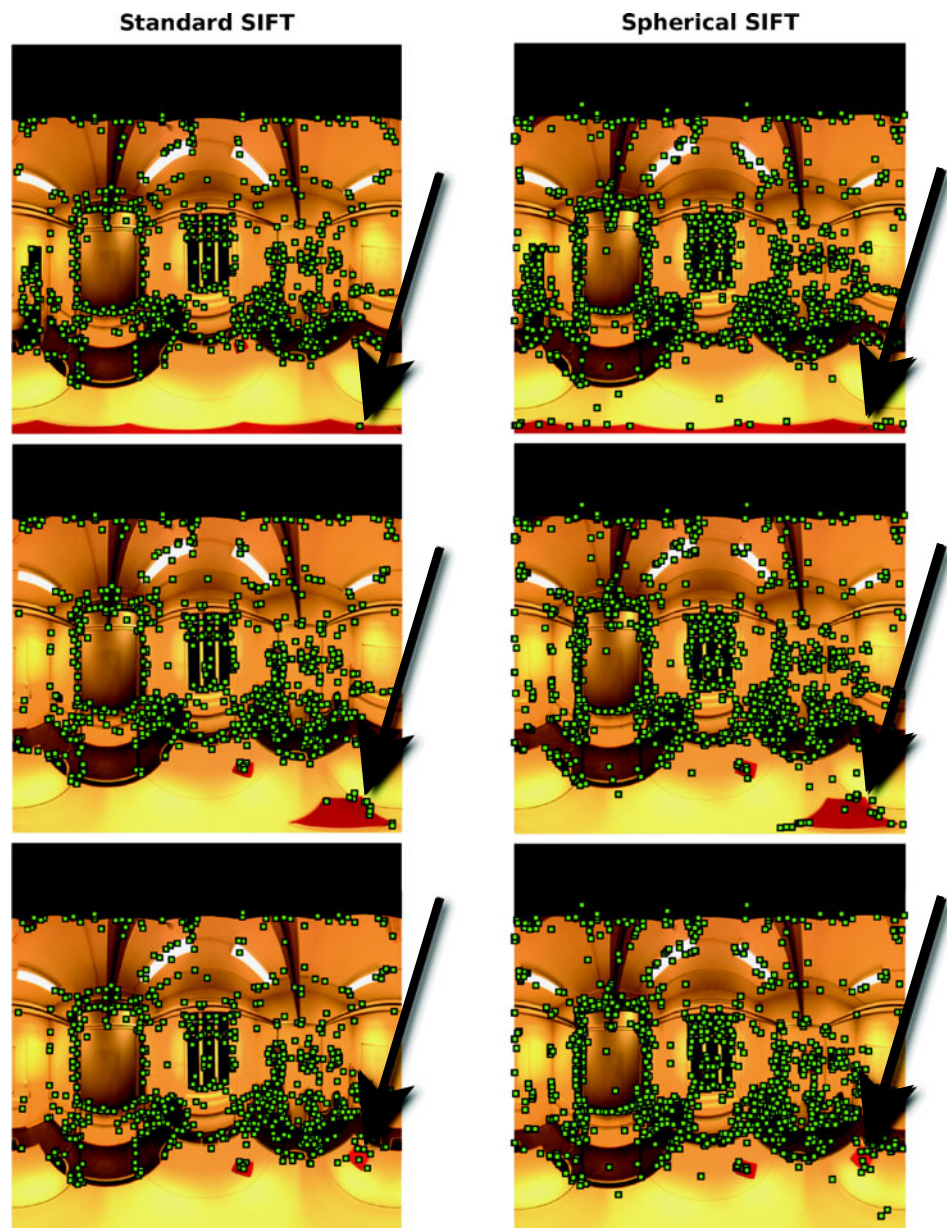
For the purpose of testing, these images have been corrupted with zero mean additive Gaussian noise with standard deviation 0.05 (pixel values are in the range $[0, 1]$) and rotated on the sphere a random angle ψ around X_2 (see Fig. 2). The rotations of the image on the sphere simulate objects captured at different points of the omnidirectional image, and therefore experiencing different deformations, as commented in Sect. 1.2. An example of the hard deformations that these rotations introduce can be observed in Fig. 14(a) (right). Then, the repeatability score for a given pair of images, i and j , is computed as

$$r_{ij} = \frac{nR_{ij}}{\min(n_i, n_j)}, \quad (40)$$

where n_i and n_j are the number of extracted points from images i and j , respectively, and nR_{ij} is the number of repeated points, i.e. points defining regions with an overlap error lower than 0.5 (Mikolajczyk et al. 2005).

The results of repeatability tests are given in Fig. 10. As expected, the repeatability increases for higher values of σ_0 . Although 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

Fig. 6 Comparison between points extracted by a standard implementation of SIFT and those extracted by the proposed scale invariant feature transform on the sphere. The *black arrow* points to the object that moves along the sequence



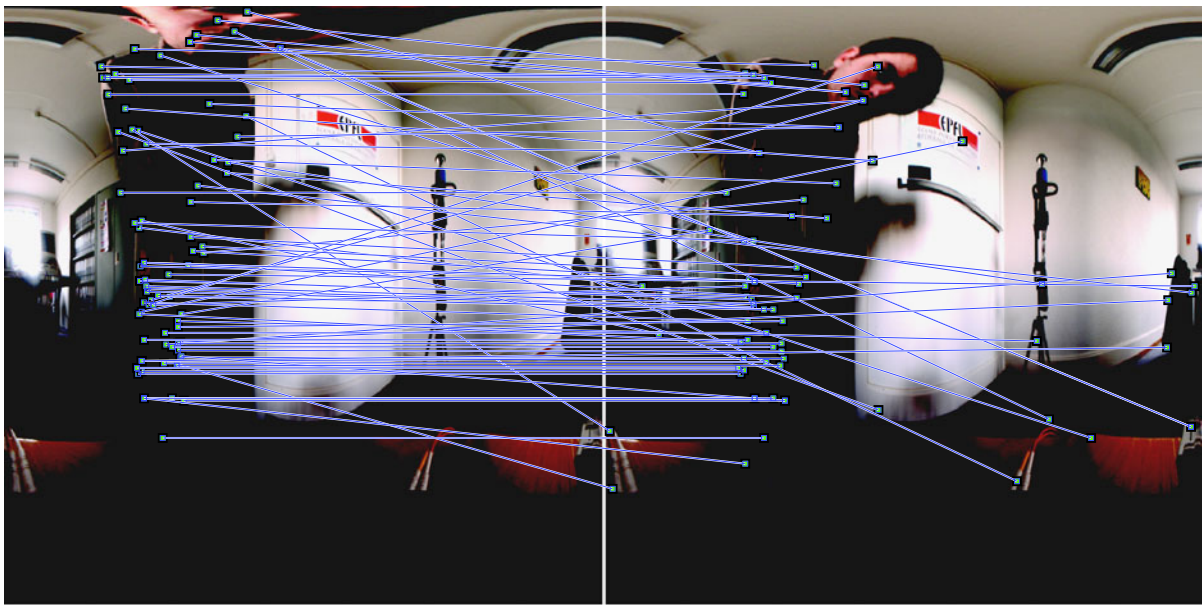
(a) Location of the extracted points by the standard SIFT algorithm applied to a sequence of 3 omnidirectional images. From bottom to up: 808 points, 846 points and 855 points.

(b) Location of the extracted points by the proposed scale invariant feature transform on the sphere applied to a sequence of 3 omnidirectional images. From bottom to up: 1373 points, 1413 points and 1489 points.

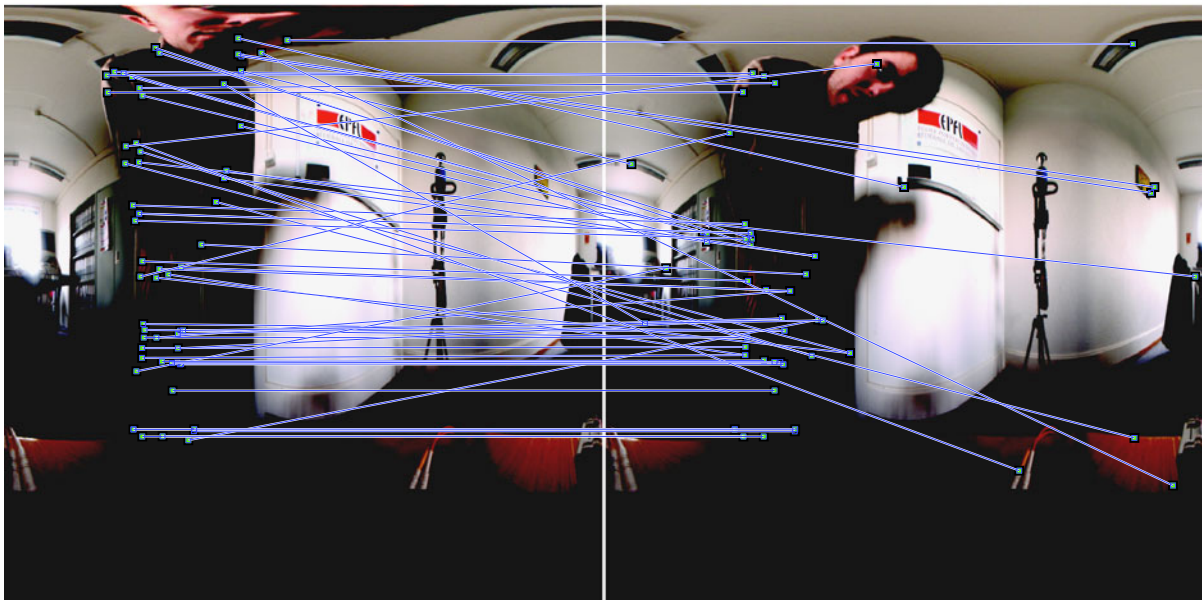
graphics, $\sigma_0 = 3.0\pi/1024$ and $S = 3$ are reasonable values to choose.

For comparing the performance in omnidirectional images of the scale invariant feature transform on the sphere against the standard implementation of SIFT, we have performed a repeatability test comparing both methods. Here we have corrupted omnidirectional images with zero mean additive Gaussian noise with standard deviation 0.05 (pixel values are in the range $[0, 1]$) and then, the obtained images have been rotated 50 times on the sphere a random angle ψ around X_2 , producing a total of 1400 pairs of images. In

Table 1, the mean, maximum and minimum values of repeatability for both methods are shown. As you can see, the results obtained by spherical SIFT are considerably better. Furthermore, in Fig. 11 we have plotted repeatability values as a function of rotation angle (plotted values are obtained by grouping values in bands of 10 degrees). It can be observed that the results obtained with spherical SIFT are independent of the rotation angle and the obtained values are always better than those obtained by standard SIFT. The difference is small for rotation values of 0, 180 and 360 degrees. However, a small rotation of 10 degrees already de-



(a) Matchings obtained using LSD (81 matchings). It can be observed that only a few matchings are incorrect (diagonal lines), independently of the position of the point on the omnidirectional image.



(b) Matchings obtained using standard SIFT (60 matchings). As it can be observed, lots of points are incorrectly matched (diagonal lines) and the few correct matchings are mainly placed close to the centre (equator) of the image, where the body remains almost unchanged.

Fig. 7 Comparison between matching using standard SIFT and matching using the scale invariant feature transform on the sphere with LSD descriptors. Both matching procedures use a matching threshold

of 0.8. Only matchings between points placed on the body of the subject are plotted, since the body is the only thing that changes between the two images

increases the repeatability of standard SIFT considerably. Furthermore, if only spherical images are considered (discarding the parabolic ones) i.e. we consider images that contain information on one of the poles (see Fig. 9). Then, in this case, the repeatability values for the standard SIFT are already 6.7, 7.6 and 4.6 percentage points lower than with the

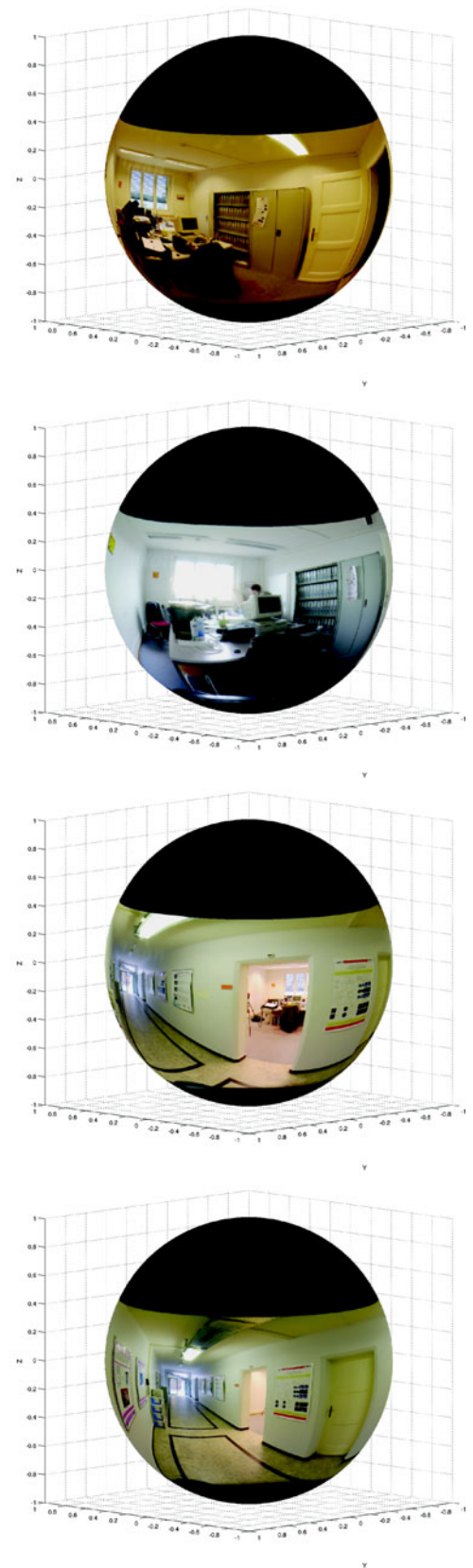
spherical SIFT for 0, 180 and 360 degrees, respectively. This difference also increases considerably increasing the rotated angle.

For these chosen values, $\sigma_0 = 3.0\pi/1024$ and $S = 3$, a matching test has been performed in order to observe the effect of the threshold d (see Sect. 3). Again, real omnidirec-

Fig. 8 Some of the real parabolic omnidirectional images used in our tests of repeatability and matching

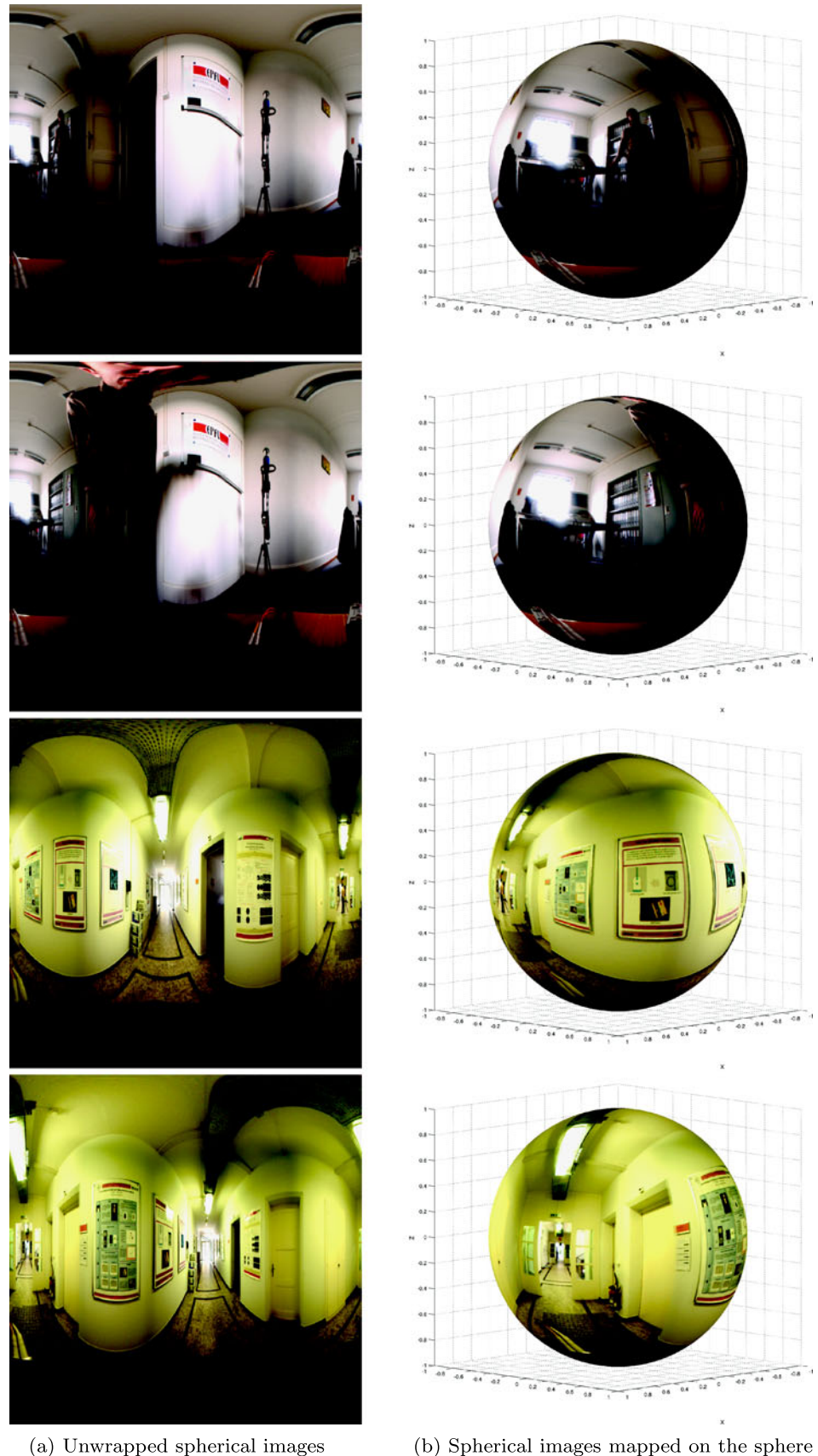


(a) Unwrapped parabolic images



(b) Parabolic images mapped on the sphere

Fig. 9 Some of the real spherical omnidirectional images used in our tests of repeatability and matching



(a) Unwrapped spherical images

(b) Spherical images mapped on the sphere

Table 1 Repeatability values of spherical SIFT and standard SIFT

	Mean repeatability	Maximum repeatability	Minimum repeatability
Spherical SIFT	82.02%	96.39%	69.19%
Standard SIFT (with octave -1)	38.78%	93.16%	9.90%
Standard SIFT (without octave -1)	45.96%	95.44%	11.64%

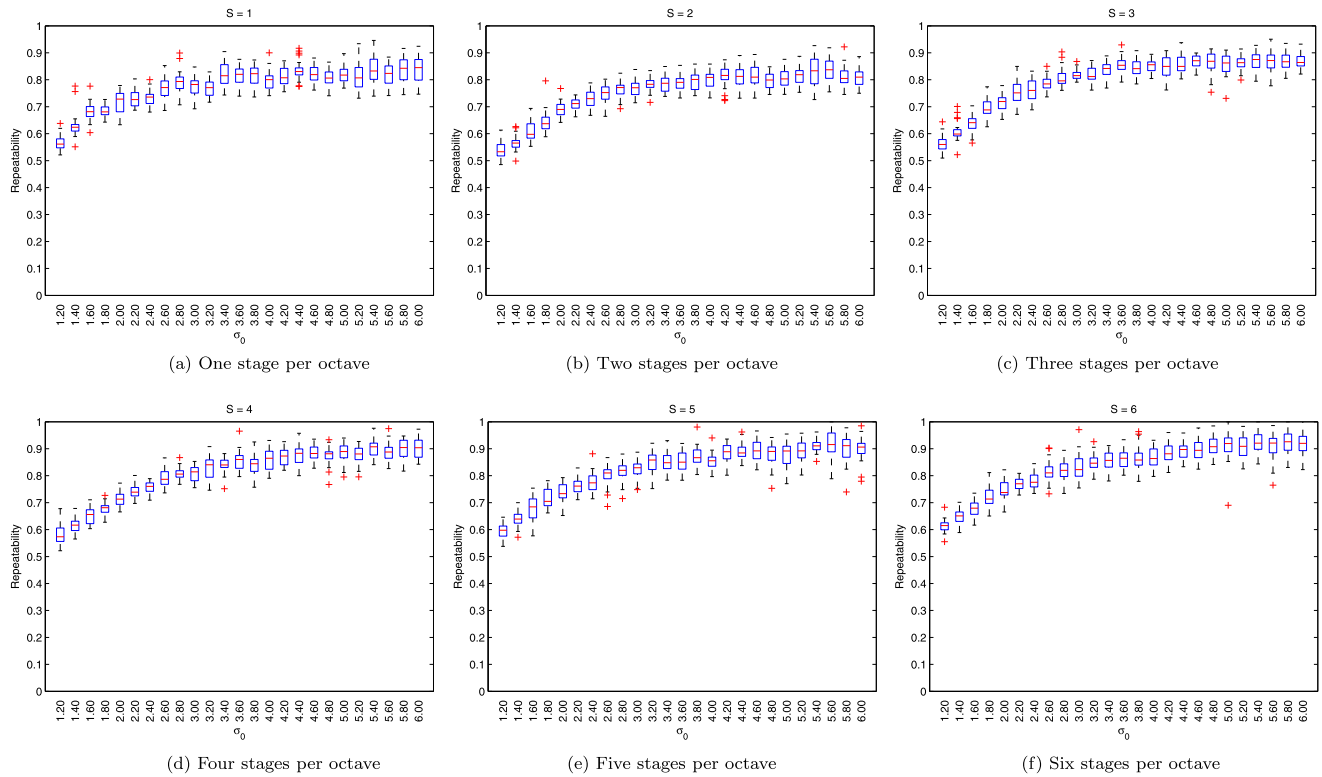


Fig. 10 (Color online) Results of the repeatability tests varying S and σ_0 over a set of 28 omnidirectional images. Note that the values of σ_0 in the graphs are in terms of relative distance between points of the spherical grid, i.e. for an image of 1024×1024 pixels, $\sigma_0 = 2.0$ in the graph means an effective $\sigma_0 = 2.0 \frac{\pi}{1024}$. The boxes in the plot mark the

25th and the 75th percentile, red lines are the median and red stars are 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)

tional images have been artificially rotated on the sphere by a random angle ψ around X_2 and corrupted with zero mean additive Gaussian noise. In Fig. 12, the results of this test can be observed. In Fig. 12a the percentage of correct matchings is computed as the ratio between correct matchings and the total number of matchings. In Fig. 12b, the percentage of correct matchings is computed as the ratio between correct matchings and the total number of repeated points. On the one hand, for a matching threshold higher than 0.3, more than 50% of repeated points are correctly matched. On the other hand, for a matching threshold lower than 0.7 more than 80% of the matched points are correct.

In Fig. 13, results of repeatability and correct matchings as a function of pixel noise are given. As expected, the increase of the amount of noise causes a decrease of repeatability (see Fig. 13a), but observing Figs. 13b and 13c, we

can observe the robustness of the matching, since the decrease on the percentage of correct matchings is considerably lower.

In Figs. 14 and 15, some examples of LSD matching between omnidirectional images are shown. In all the computations, the above mentioned parameters ($S = 3$ and $\sigma_0 = 3.0\pi/1024$) are used, as well as a matching threshold of $d = 0.7$. In Fig. 14, we show the matchings between a parabolic omnidirectional image and the same image rotated on the sphere and corrupted by additive Gaussian noise with zero mean and standard deviation 0.05. 184 LSD are correctly matched between the two images, over a total of 207 matched points (88.9%), with 329 extracted points on the image on the left, and 341 on the image on the right. In Fig. 15, we show the 49 matched LSD obtained between two different parabolic images. Among the incorrect

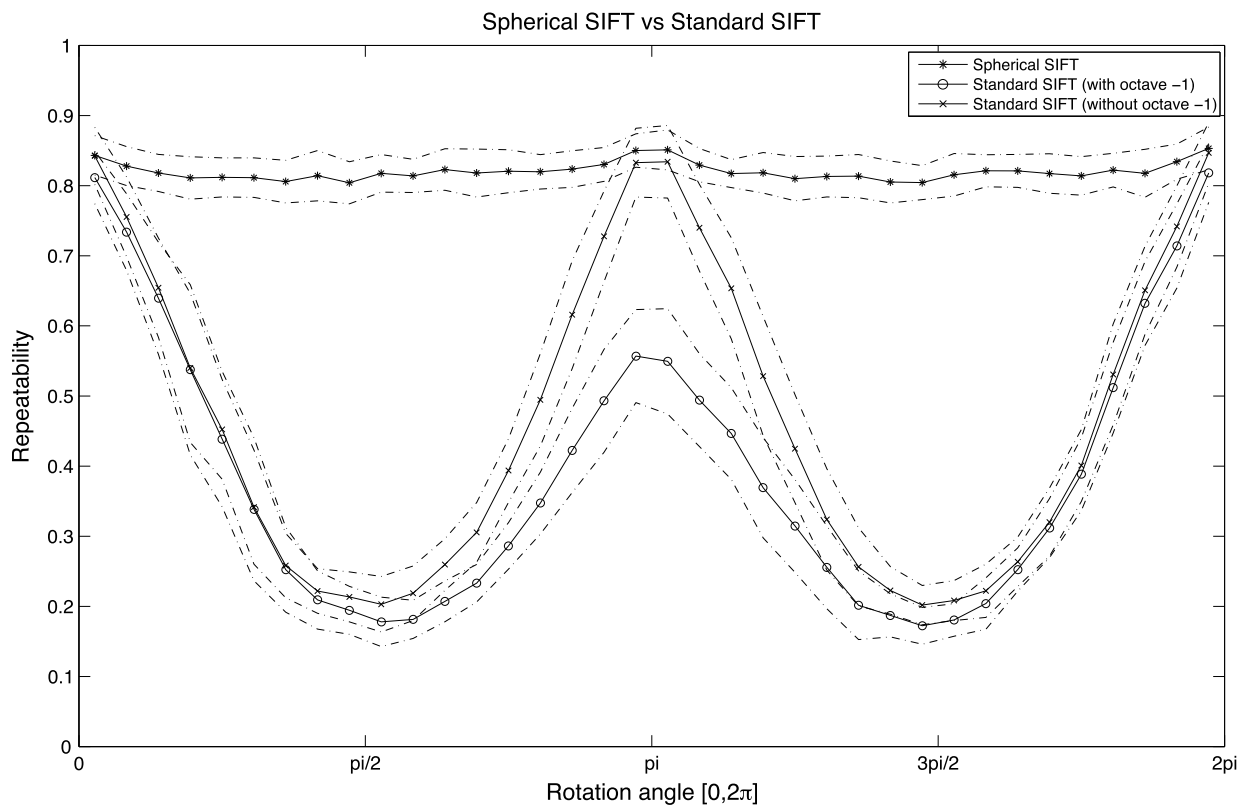
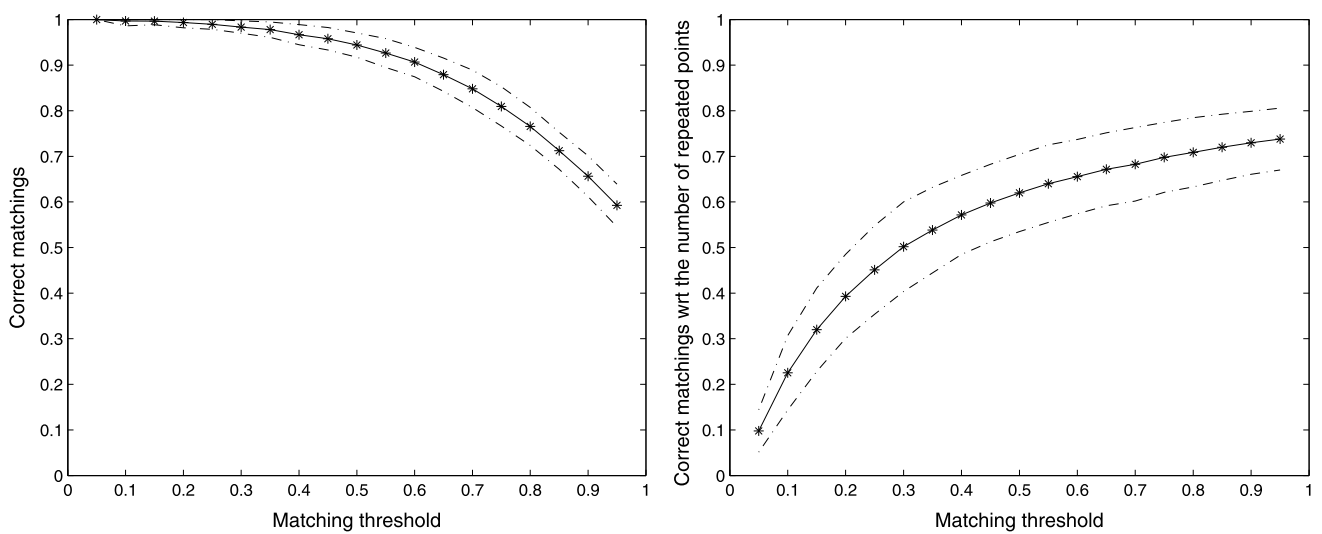


Fig. 11 Plots of repeatability values, as a function of the randomly rotated angle on the sphere, for the spherical SIFT implementation and a standard SIFT implementation (Vedaldi and Fulkerson 2008) with and without the octave -1 . The optimal parameters for both algorithms

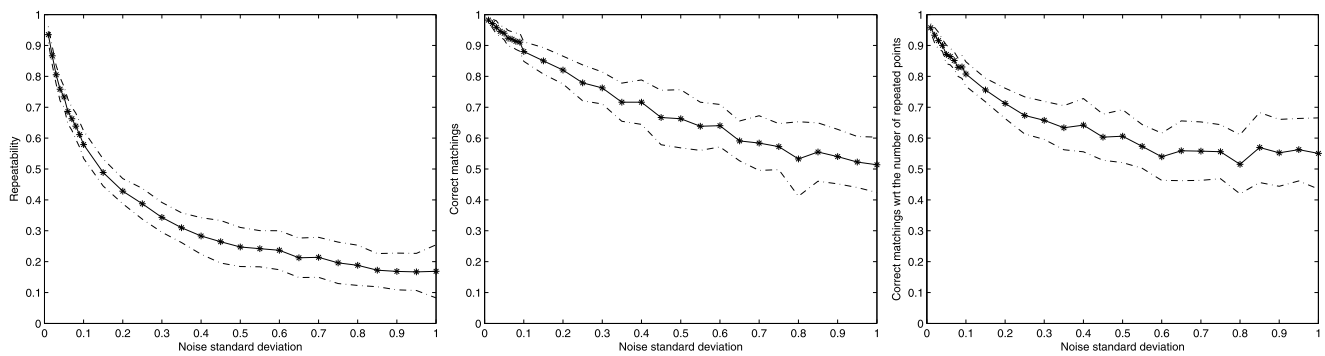
were used. The tests were performed on 28 real omnidirectional images that were corrupted with Gaussian noise and randomly rotated 50 times, producing 1400 pairs of images. *Solid lines* indicate the mean value and *dashed lines* the mean \pm standard deviation



(a) Correct matchings with respect to the total number of matched points, as a function of matching threshold.

(b) Correct matchings with respect to the total number of repeated points, as a function of matching threshold.

Fig. 12 Plots of correct matching tests performed on 28 real omnidirectional images. Correct matchings are plotted as a function of the matching threshold. *Solid lines* indicate the mean value and *dashed lines* the mean \pm standard deviation



(a) Repeatability as a function of the standard deviation of Gaussian noise added to the images. (b) Correct matchings with respect to the total number of matched points, as a function of the standard deviation of Gaussian noise added to the images. (c) Correct matchings with respect to the total number of repeated points, as a function of the standard deviation of Gaussian noise added to the images.

Fig. 13 Plots of repeatability and correct matching tests performed on 28 real omnidirectional images, as a function of the standard deviation of the Gaussian noise added to the images, for $\sigma_0 = 3.0\pi/1024$, $S = 3$

and a matching threshold fixed to 0.6. *Solid lines* indicate the mean value and *dashed lines* the mean \pm standard deviation

matches observed, most of them are actually locally correct, since they are the result of matching the real window with its reflection in the whiteboard, or one of the three identical markers in the whiteboard with another of them, etc.

Let us note that we do not need to use any virtual camera plane framework for performing matching, as in Mauthner et al. (2006). Instead, we perform the matching directly in the spherical coordinates, in which the omnidirectional sensor outputs the images (what concerns the Ladybug2).

5.3 “Planar vs. Omni” Matching

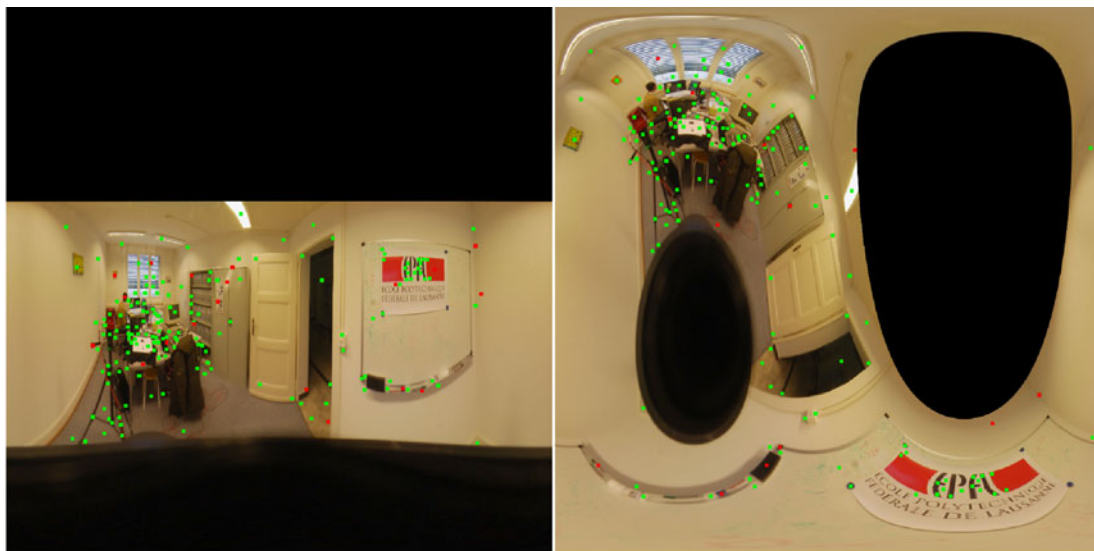
In this section, LPD are tested for matching between points extracted from omnidirectional images (using Algorithm 1) and points extracted from a planar image (using the standard SIFT algorithm). For these experiments, the SIFT parameters proposed in Lowe (2004) ($S = 3$ and $\sigma_0 = 1.6$) are used in both the standard SIFT algorithm and the spherical SIFT algorithm. Note that on the sphere, the equivalent σ parameter is $\sigma_0 = 1.6\pi/1024$. We do not compute octave -1 for speeding up the computation. On the other hand, the images we are working with are already at high resolution.

In a first step, a sequence of spherical images is processed for extracting LPD descriptors. The sequence is shown in Fig. 16, in its unwrapped version and on the sphere. This sequence consists of six images where a spherical camera moves approximately parallel to a poster on a wall. Then, a planar image of the same poster in the same scene is processed in order to extract SIFT descriptors. The result of matching both descriptors are presented in Fig. 16. There,

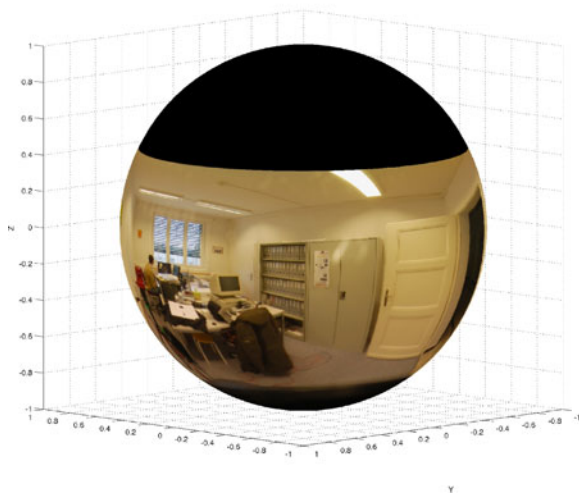
the links between matched points are shown only on the unwrapped version so that the entire sphere is visible. As it can be observed, the proposed algorithm presents a good performance as well as a good stability of the matched points. Some of the incorrect matchings that are present are due to the fact that other posters in the corridor contain some of the images of the original poster. Using the planar to spherical mapping, we can compute the repeatability between the standard SIFT algorithm applied on the planar image and the spherical SIFT algorithm applied on the sequence of spherical images. For an overlapping error lower than 0.5, we obtain an average repeatability of 27.5%, with a maximum repeatability of 34.3% and a minimum of 21.2%. These are good results, since we obtain a good amount of repeated points even considering two completely different images (planar and spherical), with two different resolutions of the poster (in the spherical image, the poster size is 1/4 the size in the planar image) and with some artefacts in the spherical image due to the stitching performed by the Ladybug2 device.

The estimation of the planar to spherical mapping, as introduced in Sect. 4, has also been tested. First, Algorithm 1 is applied to the corresponding omnidirectional images in order to obtain the set of LPD. Then, the standard SIFT descriptors are computed for the planar images, and both descriptors LPD and SIFT are the input of Algorithm 6. Let us recall that this algorithm automatically computes the matching and the mapping.

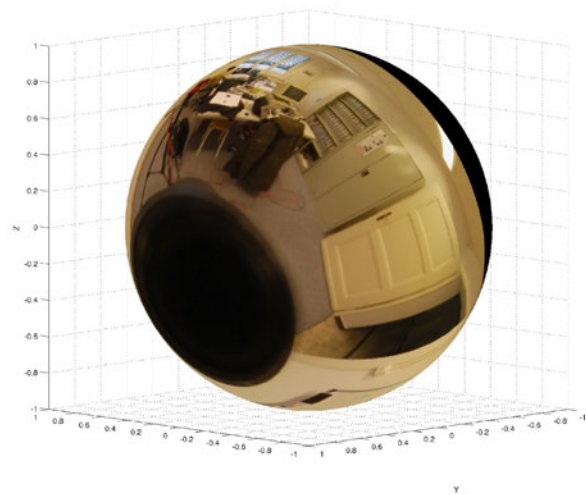
In Fig. 17, results obtained with images containing the EPFL logo (Figs. 17a and 17b) and a poster (Figs. 17c and 17d) are shown. It is interesting to note that the obtained results are satisfactory even with a highly symmetric object, as is the case of the logo, or an object with parts present in several places on the omnidirectional image, as is the case of



(a) Matching between a parabolic omnidirectional image (left) and an artificially rotated and corrupted version of the same image. 184 points are correctly matched (88.9% of the total matched points). Green dots represent correctly matched points and red dots incorrectly matched points.



(b) Parabolic omnidirectional image mapped on the sphere



(c) Parabolic omnidirectional image rotated 72 degrees around X_2 and mapped on the sphere (unwrapped version in Figure(14a)-right)

Fig. 14 (Color online) Example of LSD matching between spherical images. The images are shown unwrapped and on the sphere. The values of the parameters obtained in the previous tests ($S = 3$ and $\sigma_0 =$

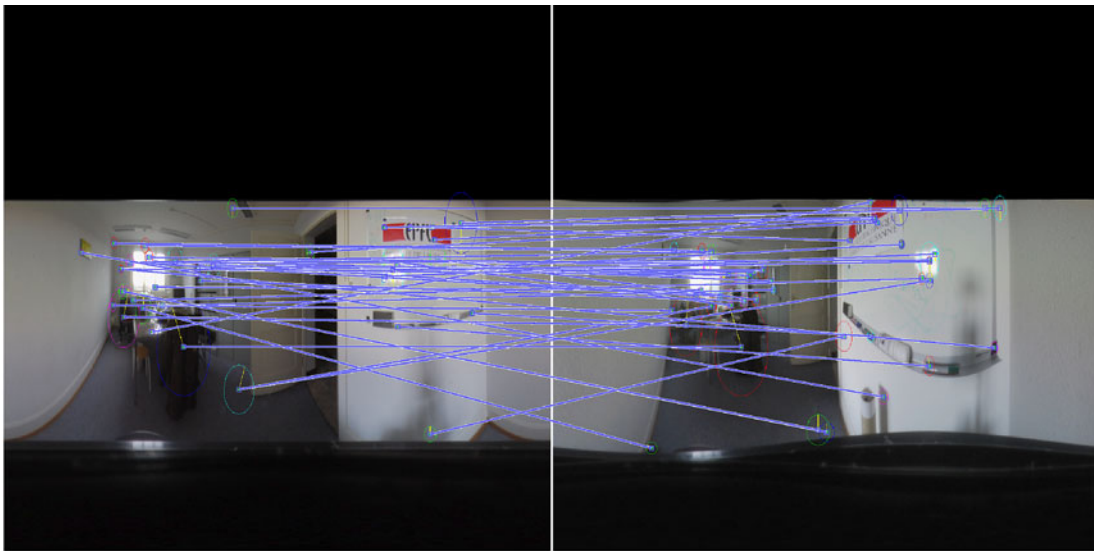
$3.0\pi/1024$) have been used in the computations. The matching threshold has been fixed to 0.7

the poster. Indeed, in these cases a matching can be locally correct while being incorrect considering the whole object or image.

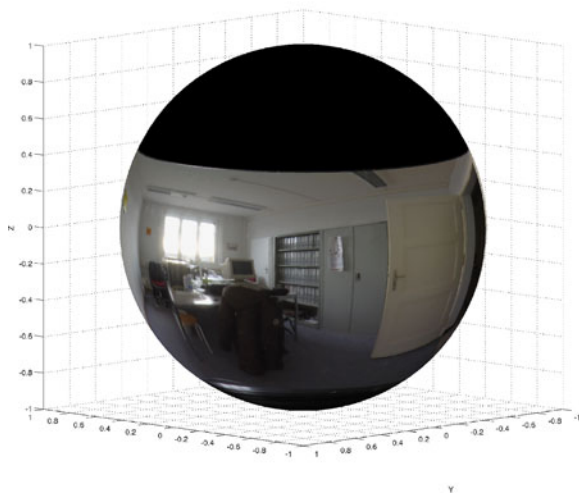
Let us note that the planar to spherical mapping can be computed also for the sequence of images in Fig. 16. Algorithm 6 also succeed on computing the mapping for every omnidirectional image of the sequence. For this computation, the RANSAC algorithm chooses 11, 12, 14, 11, 16 and 15 pairs of matched points (inliers) for computing the mapping for the omnidirectional image in Figs. 16a, 16b, 16c, 16d, 16e and 16f, respectively.

6 Conclusions

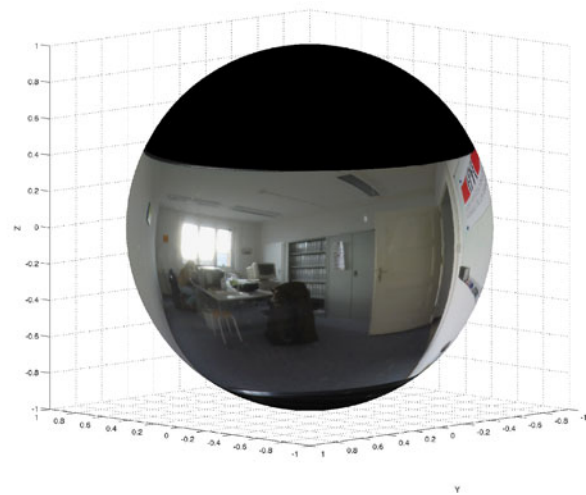
In this paper, we have proposed a SIFT algorithm in spherical coordinates. It is not limited to pure spherical images, since it can also be applied to a wide variety of omnidirectional images that can be mapped on the sphere. Two types of point descriptors have been proposed: Local Spherical Descriptors (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



(a) Matching between two different parabolic omnidirectional images. 49 LSD are matched.



(b) Omnidirectional image in Figure (15a-left) mapped on the sphere



(c) Omnidirectional image in Figure (15a-right) mapped on the sphere

Fig. 15 Example of LSD matching between spherical images. The images are shown unwrapped and on the sphere. The values of the parameters obtained in the previous tests ($S = 3$ and $\sigma_0 = 3.0\pi/1024$) have been used in the computations. The matching threshold has been fixed to 0.7

the matchings between omnidirectional images, the parameters of the 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 mapping.

The tests of repeatability, matching between omnidirectional images and matching between omnidirectional and planar images have shown the feasibility of the approach. Furthermore, this feasibility is sustained by the robust-

ness of the approach with respect to the deformations introduced in the image by the omnidirectional sensor. Indeed, the repeatability of the extracted features on omnidirectional images has been compared with the repeatability obtained using standard SIFT, showing the superiority of the spherical implementation. As it has been explained in Sect. 1.2, the deformations experienced by the objects on a scene depend on their position on the omnidirectional image. This makes that a simple displacement of an object on the 3D space, generating a small rotation on the sphere of the object on the omnidirectional image, introduces a deformation that standard SIFT is not able to han-

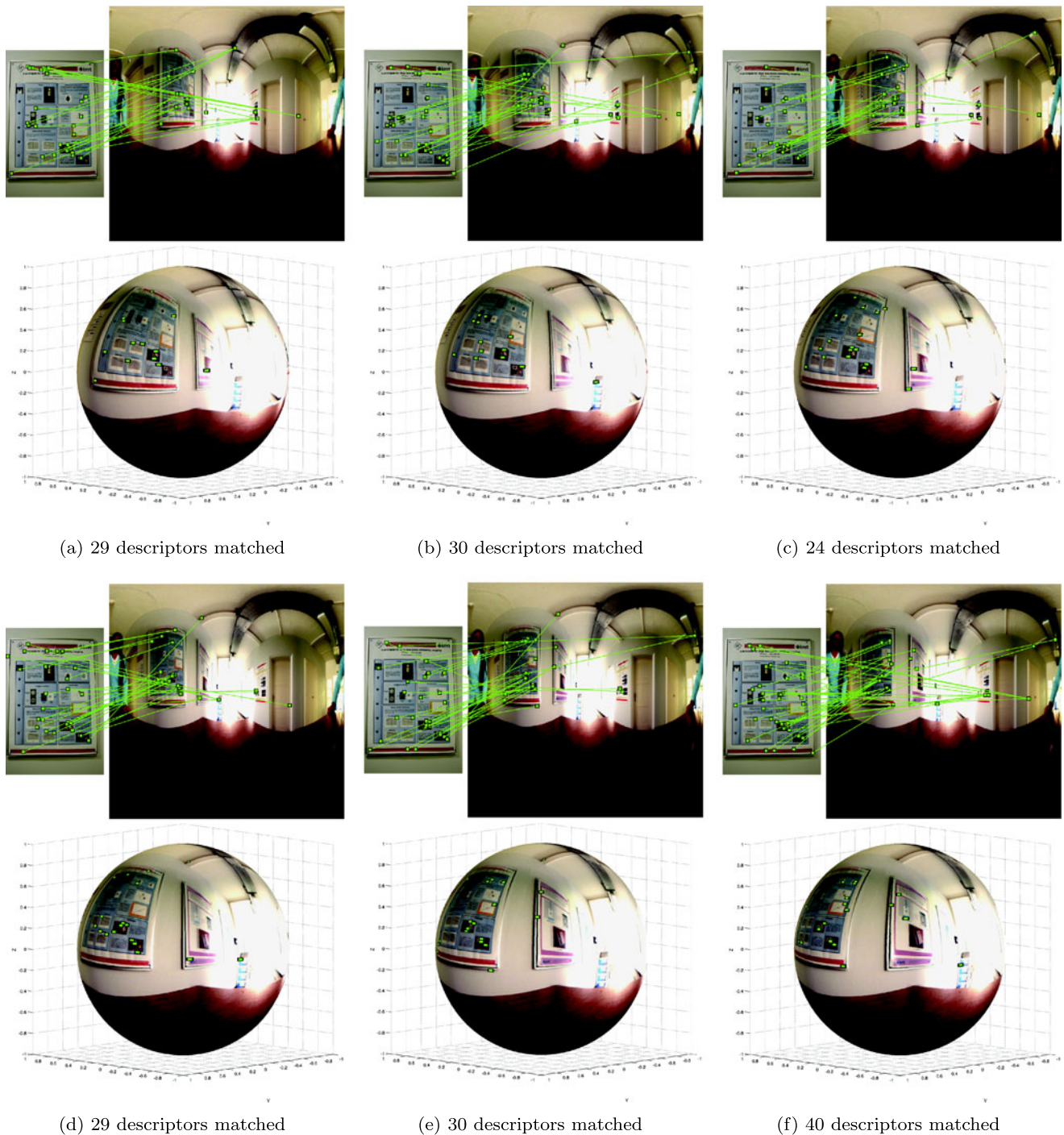


Fig. 16 Matchings obtained between LPD of a sequence of spherical images and SIFT descriptors of a planar image. In the sequence of spherical images (shown unwrapped and on the sphere), the camera

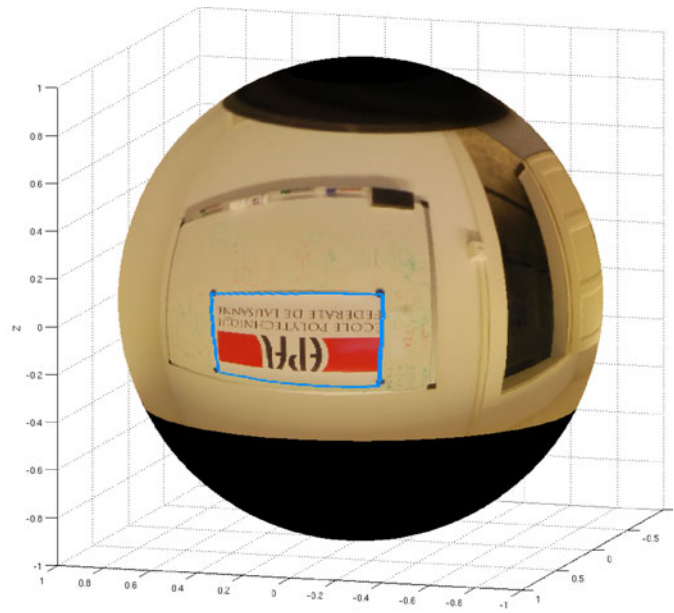
moves approximately parallel to the object present in the planar image. The matching threshold has been set to 0.6

dle correctly. This can be observed in Fig. 11, where it is shown that a small rotation of only 10° reduces the repeatability from approximately 85% to 75% in standard SIFT. This value can drop up to a value of only 20% for positions separated 90° . When using our scale invariant fea-

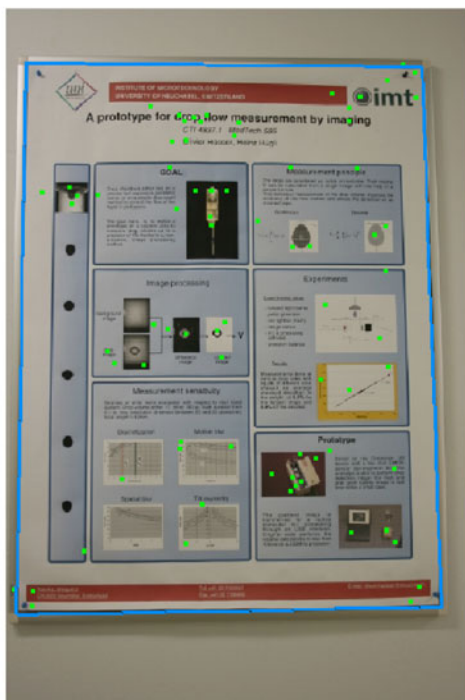
ture transform on the sphere, the values of repeatability are not affected by the deformations introduced by the omnidirectional sensor, remaining around 82% for any value of the difference on the position of objects between two images.



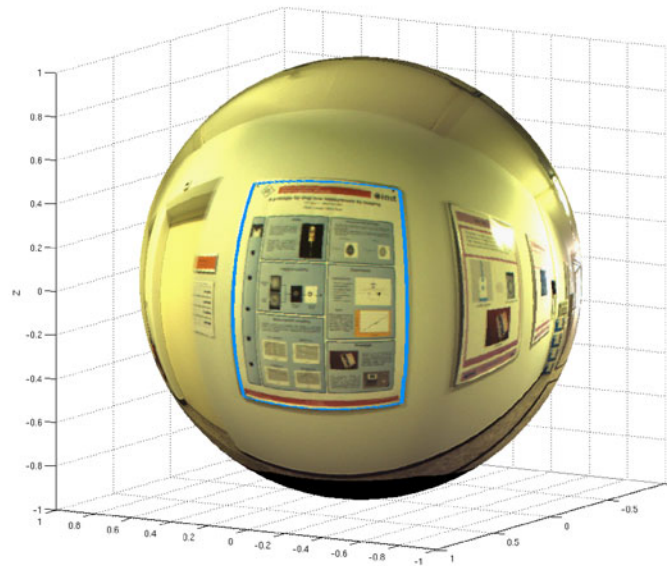
(a) Planar image. The border of the logo (in blue) has been marked by hand.



(b) Parabolic omnidirectional image mapped on the sphere. The blue border is the mapping of the logo border from Figure (17a) using Equation (39).



(c) Planar image. The border of the poster (in blue) has been marked by hand.



(d) Spherical image. The blue border is the mapping of the poster border from Figure (17c) using Equation (39).

Fig. 17 (Color online) Examples of estimation of the mapping between the boundary of an object in a planar image to this object in an omnidirectional image using Algorithm 6. The *green dots* in the planar image are the points whose matching has been used for estimating the mapping

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 33 seconds in a 3.33 GHz processor. If LPD are needed, the stereographic projection for each extracted point requires around 0.1 extra seconds. 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, 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 three-dimensional 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 DoG, 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.

Acknowledgement This work is partly supported by the Swiss National Science Foundation under grant number 200021-117823. The authors would like to acknowledge the three unknown referees for their comments and suggestions, which considerably improved the quality of the paper.

References

- Barut, A. O., & Rączka, R. (1986). *Theory of group representations and applications* (2nd edn.). Singapore: World Scientific. ISBN 9971502178.
- Baumberg, A. (2000). Reliable feature matching across widely separated views. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 774–781).
- Bay, H., Ess, A., Tuytelaars, T., & Van Gool, L. (2008). Speeded-up robust features (SURF). *Computer Vision and Image Understanding*, 110(3), 346–359. <http://www.sciencedirect.com/science/article/B6WCX-4RC2S4T-2/1/cddc9c43940bd8fb62d6ed668de29e20>.
- Bogdanova, I., Bresson, X., Thiran, J. P., & 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., & Eckmann, M. (2001). Into the woods: visual surveillance of noncooperative and camouflaged targets in complex outdoor settings. *Proceedings of the IEEE*, 89(10), 1382–1402. doi:10.1109/5.959337.
- Brox, T., Cremers, D., Gall, J., & Rosenhahn, B. (2010). Combined region- and motion-based 3d tracking of rigid and articulated objects. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. doi:10.1109/TPAMI.2009.32.
- Bulow, T. (2004). Spherical diffusion for 3d surface smoothing. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(12), 1650–1654. doi:10.1109/TPAMI.2004.129.
- Bur, A., Tapus, A., Ouerhani, N., Siegart, R., & Hugli, H. (2006). Robot navigation by panoramic vision and attention guided features. In *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., & 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. doi:10.1109/TCSVT.2008.928223.
- Driscoll, J., & Healy, D. (1994). Computing Fourier transforms and convolutions on the 2-sphere. *Advances in Applied Mathematics*, 15(2), 202–250. doi:10.1006/aama.1994.1008.
- Ehlgen, T., Pajdla, T., & Ammon, D. (2008). Eliminating blind spots for assisted driving. *IEEE Transactions on Intelligent Transportation Systems*, 9(4), 657–665. doi:10.1109/TITS.2008.2006815.
- Fischler, M. A., & 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., & Daniilidis, K. (2001). Catadioptric projective geometry. *International Journal of Computer Vision*, 45(3), 223–243.
- Goedeme, T., Tuytelaars, T., Van Gool, L., Vanacker, G., & Nuttin, M. (2005). Omnidirectional sparse visual path following with occlusion-robust feature tracking. In *Workshop on omnidirectional vision, camera networks and non-classical cameras—OMNIVIS*.
- Hadj-Abdelkader, H., Malis, E., & Rives, P. (2008). Spherical image processing for accurate visual odometry with omnidirectional cameras. In *Workshop on omnidirectional vision, camera networks and non-classical cameras—OMNIVIS*.
- Hansen, P., Corke, P., Boles, W., & Daniilidis, K. (2007a). Scale invariant feature matching with wide angle images. In *Proceedings of the IEEE/RSJ international conference on intelligent robots and systems (IROS)* (pp. 1668–1694). doi:10.1109/IROS.2007.4399266.
- Hansen, P., Corke, P., Boles, W., & Daniilidis, K. (2007b). Scale-invariant features on the sphere. In *International conference on computer vision* (pp. 1–8). doi:10.1109/ICCV.2007.4408893.
- Harris, C., & Stephens, M. J. (1988). A combined corner and edge detector. In *Alvey conference* (pp. 147–152). http://www.csse.uwa.edu.au/~pk/Research/MatlabFns/Spatial/Docs/Harris/A_Combined_Corner_and_Edge_Detector.pdf.
- Kadir, T., & Brady, M. (2001). Saliency, scale and image description. *International Journal of Computer Vision*, 45(2), 83–105. doi:10.1023/A:1012460413855.
- Lindeberg, T. (1998). Feature detection with automatic scale selection. *International Journal of Computer Vision*, 30(2), 79–116. doi:10.1023/A:1008045108935.
- Lowe, D. G. (2004). Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2), 91–110. doi:10.1023/B:VISI.0000029664.99615.94.
- Matas, J., Chum, O., Urba, M., & Pajdla, T. (2002). Robust wide baseline stereo from maximally stable extremal regions. In *Proceedings of the British machine vision conference* (pp. 384–393).
- Mauthner, T., Fraundorfer, F., & Bischof, H. (2006). Region matching for omnidirectional images using virtual camera planes. In *Proceedings of the computer vision winter workshop 2006*.
- Menegatti, E., Pretto, A., Scarpa, A., & Pagello, E. (2006). Omnidirectional vision scan matching for robot localization in dynamic environments. *IEEE Transactions on Robotics*, 22(3), 523–535.
- Mikolajczyk, K., & Schmid, C. (2005). A performance evaluation of local descriptors. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(10), 1615–1630. doi:10.1109/TPAMI.2005.188.

- Mikolajczyk, K., Tuytelaars, T., Schmid, C., Zisserman, A., Matas, J., Schaffalitzky, F., Kadir, T., & van Gool, L. (2005). A comparison of affine region detectors. *International Journal of Computer Vision*, 65(1), 43–72. doi:10.1007/s11263-005-3848-x.
- 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., & Siegwart, R. (2008). Appearance-guided monocular omnidirectional visual odometry for outdoor ground vehicles. *IEEE Transactions on Robotics*, 24(5), 1015–1026.
- Sirmacek, B., & 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. doi:10.1109/TGRS.2008.2008440.
- Smith, S. M., & 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., & Zell, A. (2006). Localization of mobile robots with omnidirectional vision using particle filter and iterative sift. *Robotics and Autonomous Systems*, 54(9), 758–765. doi:10.1016/j.robot.2006.04.018. <http://www.sciencedirect.com/science/article/B6V16-4KJ0SSH-1/2/a0cb5ac27fe99eac909f6767d486baa2>. Selected papers from the 2nd European Conference on Mobile Robots (ECMR '05).
- Tosic, I., & Frossard, P. (2009). Spherical imaging in omnidirectional camera networks. In *Multi-camera networks: concepts and applications*.
- Tuytelaars, T., & Mikolajczyk, K. (2007). Local invariant feature detectors: a survey. *Foundations and Trends in Computer Graphics and Vision*, 3(3), 177–280. doi:10.1561/06000000017.
- Tygart, M. (2008). Fast algorithms for spherical harmonic expansions, II. *Journal of Computational Physics*, 227(8), 4260–4279. doi:10.1016/j.jcp.2007.12.019.
- Valgren, C., & Lilienthal, A. (2007). SIFT, SURF and seasons: long-term outdoor localization using local features. In *Proceedings of the European conference on mobile robots (ECMR)*.
- Van Gool, L. J., Moons, T., & Ungureanu, D. (1996). Affine/photometric invariants for planar intensity patterns. In *Computer vision—ECCV'96* (pp. 642–651). doi:10.1007/BFb0015574.
- Vedaldi, A., & Fulkerson, B. (2008) VLFeat: an open and portable library of computer vision algorithms. <http://www.vlfeat.org/>.
- 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., & MacDonald, B. (2005). Vision-based localization algorithm based on landmark matching, triangulation, reconstruction, and comparison. *IEEE Transactions on Robotics*, 21(2), 217–226. doi:10.1109/TRO.2004.835452.
- Zabih, R., & Woodfill, J. (1994). Non-parametric local transforms for computing visual correspondence. In *Lecture notes in computer science. Computer vision—ECCV'94* (pp. 151–158). Berlin: Springer. doi:10.1007/BFb0028329.