



Visual odometry methods (Comport, Malis, & Rives, 2007; Konolige & Agrawal, 2008; Nistér, Naroditsky, & Bergen, 2004) are, however, incremental and prone to small drifts, which, when integrated over time, become increasingly significant over large distances. In a similar way, simultaneous localization and mapping (SLAM) approaches (Davison & Murray, 2002; Durrant-Whyte & Bailey, 2006; Klein & Murray, 2007; Montemerlo, Thrun, Koller, & Wegbreit, 2002; Thrun, 2002) allow both localization and cartography, giving alternative and promising solutions. Classically based on the extended Kalman filter, these techniques have proven to work over very long trajectories, however their robustness and accuracy are highly dependent on tuning low-level feature extraction-matching, and the filtering approach often smooths out important non-linear information.

A solution to minimize drift is to use a precomputed 3D model of the environment obtained during a learning phase. This kind of approach has several advantages compared to pure SLAM approaches:

- Map building is performed via a learning phase that can be computed offline using powerful yet computationally expensive techniques.
- More computational effort can be dedicated to robust real-time pose estimation online.
- Localization with respect to the 3D model remains drift-free.
- If the 3D model is at scale, monocular localization is at scale, too.

In visual navigation, two main classes of approach exist, namely global parametric 3D models and image-based key-frame models. In the first case, 3D parametric models have been used extensively for object tracking by minimizing the reprojection error between the model and salient features in the images (Comport, Marchand, Pressigout, & Chaumette, 2006; Drummond, Society, & Cipolla, 2002; Lothe, Bourgeois, Dekeyser, Royer, & Dhome, 2010; Marchand, Bouthemy, & Chaumette, 2001; Vacchetti, Lepetit, & Fua, 2004). This kind of approach represents the 3D data in a single global reference frame and has difficulty in representing local data accurately. Even if automatic techniques that build 3D parametric models of large-scale environments are becoming more and more accurate (Craciun, Papanoditis, & Schmitt, 2010; Hammoudi, Dornaika, Soheilian, & Papanoditis, 2010; Lafarge & Mallet, 2011), they rely heavily on the structure of urban environments and are therefore not well adapted to unstructured environments.

On the other hand, key-frame techniques aim at representing the world with a set of images positioned in two or three dimensions, without performing an explicit 3D reconstruction of the environment. This allows us to store raw (unmodified) local sensor data in the representation for high accuracy, and it maintains a topological

framework at large scale that is accurate enough to ensure the connectivity between locally precise key-frames. In the literature, several approaches have successively performed localization using an image-based memory. In Royer, Lhuillier, Dhome, and Chateau (2005) and Konolige and Agrawal (2008), a database of key-frames, containing geometric point features (Harris), is used for online camera localization. A similar approach is proposed in Courbon, Mezouar, and Martinet (2009) with an omnidirectional camera. In Jogan and Leonardis (2000), a cylindrical panorama memory is proposed, and in Cobzas, Zhang, and Jagersand (2003) a panoramic image memory combined with depth information extracted from a laser scan is used for localization.

Those approaches, however, all rely on feature extraction and matching, and therefore they do not take full advantage of the dense information contained in the images. Our main contribution is to propose an accurate and robust image-based representation that allows us to map large-scale unstructured environments as densely as possible. The aim is to provide a world model that provides maximal robustness and maximal flexibility for online localization and navigation purposes. Online real-time localization efficiency is achieved not by extracting and matching features, but by proposing a novel nonlinear saliency measure for individual pixels. In particular, the proposed representation is composed of a graph of augmented visual spheres containing omnidirectional RGB-D information (color and depth) for many spatial locations. This representation allows us to efficiently perform online localization and navigation using state-of-the-art dense registration techniques (Comport et al., 2007; Meilland, Comport, & Rives, 2010; Meilland, Rives, & Comport, 2012).

This present publication is the culmination of many previously published articles that were carried out in the context of the French national CityVIP project aimed at autonomous vehicle navigation. The individual contributions have never been published together, and the aim of this article is to provide the global scope of this complex system while demonstrating new experimental results for several large-scale field experiments:

- In Comport et al. (2010), a dense direct approach was proposed for visual odometry.
- In Meilland et al. (2010), a spherical representation was proposed along with a saliency selection approach.
- In Meilland, Comport, and Rives (2011a), a spherical sensor and real-time localization with respect to a spherical database were proposed.
- In Meilland, Comport, and Rives (2011b), a robust approach to illumination variations was proposed.
- In Meilland et al. (2012), autonomous navigation in challenging environments was demonstrated.

This paper has been structured in such a way that the relevant literature review is included at the beginning of each section. As such, in Section 2 the global nonlinear optimization criterion that is minimized will be introduced and formalized. In Section 3, a dense spherical key-frame model will be introduced to represent the 3D world. Section 4 will detail a novel spherical acquisition system that was built for the purpose of this project to acquire streams of photometric color and depth panoramas at 45 Hz. In Section 5, we will provide computational tools to allow real-time localization of different types of cameras with respect to this world model. Finally, Section 6 will detail several field experiments. Initial results for the acquisition and mapping of large-scale environments will be given. Results from several sites, including the XIIth district of Paris, INRIA Sophia-Antipolis, and Place Jaude in Clermont-Ferrand, will be given. In the final section, a fully autonomous, vision-only navigation system will be demonstrated to run successfully in a highly dynamic unstructured environment involving pedestrians, trams, cars, and varying illumination conditions.

## 2. VISUAL LOCALIZATION

Feature-based methods (e.g., Davison & Murray, 2002; Howard, 2008; Kitt, Geiger, & Lategahn, 2010; Mei, Sibley, Cummins, Newman, & Reid, 2010; Mouragnon, Lhuillier, Dhome, Dekeyser, & Sayd, 2006; Nistér et al. 2004; Tardif, George, Laverne, Kelly, & Stentz, 2010) rely on an intermediary estimation process based on thresholding (Harris & Stephens, 1988; Lowe, 2004) before requiring matching between frames to recover camera motion. This feature extraction and matching process is often badly conditioned, noisy, and not robust, and therefore it must rely on higher-level robust estimation techniques and on filtering.

Direct approaches (*image-based*), however, do not rely on this feature extraction and matching process. The camera motion is directly estimated by minimizing a nonlinear intensity error between images via a parametric warping function. In this way, the matching and the motion estimation are performed simultaneously at each step of the optimization. Classically direct approaches have focused on region-of-interest tracking, whether they be modeled by affine (Hager & Belhumeur, 1998), planar (Baker & Matthews, 2001; Dame & Marchand, 2010; Irani & Anandan, 1998, 2000; Lucas & Kanade, 1981; Malis, 2004), or multiple-plane tracking (Caron, Marchand, & Mouaddib, 2011; Mei, Benhimane, Malis, & Rives, 2006; Shi & Tomasi, 1994; Silveira, Malis, & Rives, 2008). In Comport et al. (2007), direct approaches were generalized to use the full image densely and track 6 DOF pose using stereo cameras while mapping the environment through dense stereo matching. This approach allowed direct methods to perform accurate and robust visual odometry over large scales.

### 2.1. Direct 3D Registration

To begin, consider an image  $\mathbf{I}$  with a color brightness function  $\mathbf{I} : \Omega \rightarrow \mathbb{R}^+; (\mathbf{p}) \mapsto \mathbf{I}(\mathbf{p})$ , where  $\Omega = [1, n] \times [1, m] \subset \mathbb{R}^2$ ,  $\mathbf{p} = (\mathbf{p}_1, \mathbf{p}_2, \dots, \mathbf{p}_{nm})^\top \in \mathbb{R}^{nm \times 2} \subset \Omega$  are pixel locations within the image, and  $n \times m$  is the dimension of the image. Now consider that for each pixel of the image, depth information is also available such that  $\mathbf{Z} : \Omega \rightarrow \mathbb{R}^+; (\mathbf{p}) \mapsto \mathbf{Z}(\mathbf{p})$ . It is convenient to consider the set of measurements in vector form such that  $\mathbf{I}(\mathbf{p}) \in \mathbb{R}^{nm \times 1}$  and  $\mathbf{Z}(\mathbf{p}) \in \mathbb{R}^{nm \times 1}$ . The set  $\mathcal{I} = \{\mathbf{I}, \mathbf{Z}\}$  will be denoted as an augmented image.

The objective here is to register a *current* image  $\mathbf{I}$  with a *reference* augmented image  $\mathcal{I}^*$ , where  $\mathbf{I}$  is undergoing a full 3D transformation  $\tilde{\mathbf{T}}(\mathbf{x}) = (\mathbf{R}, \mathbf{t}) \in \mathbb{SE}(3)$  defined between  $\mathbf{I}$  and  $\mathcal{I}^*$ . A superscript  $*$  will be used throughout to designate the reference view variables.

Throughout,  $\mathbf{R} \in \mathbb{SO}(3)$  is a rotation matrix and  $\mathbf{t} \in \mathbb{R}(3)$  is a translation vector. The vector  $\mathbf{x} = (\boldsymbol{\omega}, \mathbf{v}) \in \mathbb{R}^6$  is the 6 DOF twist, which is related to a pose  $\mathbf{T}$  via the matrix exponential

$$\mathbf{T}(\mathbf{x}) = e^{[\mathbf{x}]_\wedge} = \int_0^1 \mathbf{x} dt \in \mathbb{SE}(3), \quad (1)$$

with the operator  $[\cdot]_\wedge$  as

$$[\mathbf{x}]_\wedge = \begin{bmatrix} [\boldsymbol{\omega}]_\times & \mathbf{v} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \in \mathfrak{se}(3),$$

where  $[\cdot]_\times$  represents the skew symmetric matrix operator.

Under a full brightness consistency assumption, the current image intensities can be warped via novel view synthesis (Avidan & Shashua, 1997) onto the reference view such that

$$\mathbf{I}^*(\mathbf{P}^*) = \mathbf{I}(w(\mathbf{T}; \mathbf{Z}^*, \mathbf{p}^*)), \quad (2)$$

where the warping function  $w(\cdot)$  transfers the current image intensities onto the reference pixels  $\mathbf{p}^*$  via the depth-map  $\mathbf{Z}^*$ . This warping function depends on the acquisition sensor, and it can be, for example, a perspective projection, an omnidirectional projection, or a spherical projection (see Appendix A 1).

Now suppose that only a close approximation  $\hat{\mathbf{T}}$  of the true pose  $\tilde{\mathbf{T}}$  is available. The aim is to estimate the incremental pose transformation  $\mathbf{x}$  of the true value  $\tilde{\mathbf{x}}$ , which satisfies

$$\tilde{\mathbf{T}} = \hat{\mathbf{T}}\mathbf{T}(\mathbf{x}). \quad (3)$$

The unknown  $\mathbf{x}$  can be estimated by minimizing the following intensities error:

$$\mathbf{e}(\mathbf{x}) = \rho(\mathbf{I}(w(\hat{\mathbf{T}}\mathbf{T}(\mathbf{x}); \mathbf{Z}^*, \mathbf{p}^*)) - \mathbf{I}^*(\mathbf{P}^*)), \quad (4)$$

where  $\rho(\cdot)$  is a robust outlier rejection computed by M-estimation using the Huber influence function (Huber, 1981). This function allows us to reject outliers such as moving objects and local illumination changes between the images.

## 2.2. Nonlinear Minimization

The error function of Eq. (4) can be minimized using an iteratively reweighted least-squared optimization:

$$\mathbf{x} = \arg \min_{\mathbf{x}} \mathbf{e}(\mathbf{x})^T \mathbf{D} \mathbf{e}(\mathbf{x}), \quad (5)$$

by  $\nabla \mathbf{e}(\mathbf{x})|_{\mathbf{x}=\bar{\mathbf{x}}} = \mathbf{0}$ , where  $\nabla$  is the gradient operator with respect to the unknown  $\mathbf{x}$ , assuming a global minimum is reached at  $\mathbf{x} = \bar{\mathbf{x}}$ .

An efficient second-order minimization (ESM) is employed (Malis, 2004), which allows us to precompute most of the minimization parts directly on the reference image. In this case, the unknown  $\mathbf{x}$  is iteratively updated using a Gauss-Newton-like procedure:

$$\mathbf{x} = -(\mathbf{J}^T \mathbf{D} \mathbf{J})^{-1} \mathbf{J}^T \mathbf{D} \mathbf{e}, \quad (6)$$

where  $\mathbf{Q} = \mathbf{J}^T \mathbf{D} \mathbf{J}$  is the robust Gauss-Newton Hessian approximation,  $\mathbf{J}$  is the warping Jacobian matrix of dimension  $nm \times 6$ , and  $\mathbf{D}$  is a diagonal weighting matrix of dimensions  $nm \times nm$  related to the robust function  $\rho(\cdot)$  (see details in Appendix A 3).

The pose estimate is updated at each step by a homogeneous transformation:

$$\hat{\mathbf{T}} \leftarrow \hat{\mathbf{T}} \mathbf{T}(\mathbf{x}), \quad (7)$$

and the minimization is iterated until the increments on  $\mathbf{x}$  are sufficiently small.

## 2.3. Multiple Key-frame Registration

In Section 2.1, only one reference image was considered. This approach is highly incremental and prone to drift and linearization approximation. If more than one reference image is available, then it is possible to consider global bundle adjustment approaches (Triggs, Mclauchlan, Hartley, & Fitzgibbon, 1999), however these approaches are not computationally efficient for real-time applications. Furthermore, in large scale mapping approaches, the connectivity between the various camera poses is relatively sparse, meaning that performing global bundle adjustment is not efficient. Alternatively, it is possible to consider sliding window bundle adjustment (Mouragnon et al., 2006; Sibley, Matthies, & Sukhatme, 2008), which improves performance at little cost.

In this paper, the localization problem will be formulated as a simultaneous minimization of several intensities from  $N$  locally close reference images from a key-frame graph such that

$$\mathbf{e}(\mathbf{x}) = \begin{bmatrix} \mathbf{I}(w(\hat{\mathbf{T}}\mathbf{T}(\mathbf{x})\mathbf{I}; \mathbf{Z}^1, \mathbf{p}^1)) - \mathbf{I}^1(\mathbf{p}^1) \\ \vdots \\ \mathbf{I}(w(\hat{\mathbf{T}}\mathbf{T}(\mathbf{x})\mathbf{T}^N; \mathbf{Z}^N, \mathbf{p}^N)) - \mathbf{I}^N(\mathbf{p}^N) \end{bmatrix}, \quad (8)$$

where  $\mathbf{T}^N$  is the transformation between each key-frame.

## 2.4. Multiresolution Approach

One of the major drawbacks of direct approaches, and more generally of iterative approaches, is that the initial pose  $\hat{\mathbf{T}}$  must be close to the solution  $\hat{\mathbf{T}}$  to converge. The convergence domain and the algorithm performance can be improved using a coarse-to-fine optimization strategy. This is achieved using multiresolution image pyramids [e.g., constructed by Gaussian filtering and subsampling (Burt & Adelson, 1983)]. The minimization begins at the lowest resolution, and the result is used to initialize the next level repeatedly until the highest resolution is reached. This greatly improves the convergence domain/speed, and some local minima can be avoided.

## 3. 3D WORLD REPRESENTATION

### 3.1. Parametric Models

Many approaches in the literature use parametric models to efficiently represent the environment and store *prior* information about the scene. Computer-aided design (CAD) models can be used to directly inject prior information about the 3D model into a localization system, and some examples include Brown (1971), Comport et al. (2006), Drummond et al. (2002), Lowe (1991), Marchand et al. (2001), and Vacchetti et al. (2004). In this case, camera pose (orientation and translation) is estimated with respect to the model reference frame by minimizing the error between the reprojected 3D model and corresponding features extracted from the images. The features may include points, lines (Lowe, 1991; Marchand et al., 2001), planes (Benhimane & Malis, 2004), contours (Drummond et al., 2002), cylinders, or even combinations of them (Comport et al., 2006). While these algorithms work well, they are based on highly complicated modeling procedures as well as structured visual features in the images, meaning that they are rarely used for environments larger than the scale of a room, and they do not easily scale to unstructured environments.

Several studies have been conducted to improve localization techniques for larger-scale environments using parametric models. Lothe et al. (2010) used an approximate global 3D model to align a local map, reconstructed using a visual SLAM approach, with the geometric part of the model in order to correct for drift. In Cappelle, El Najjar, Charpillet, and Pomorski (2011), a 3D model is only used to detect obstacles observed between the images perceived by a camera and virtual images, while the localization of the camera is obtained by a global-positioning-system-real-time kinematic (GPS-RTK) technique. In Irschara, Zach, Frahm, and Bischof (2009), a sparse 3D point model is reconstructed using a structure-from-motion (SfM) algorithm, and it is used to locate a camera through matched scale-invariant feature transform (SIFT) points.

In general, these parametric models are capable of representing environments or target objects in an efficient

yet approximate way. Even if methods for automatically reconstructing large-scale environments using parametric models are becoming more and more precise (Craciun et al., 2010; Furukawa & Ponce, 2010; Hammoudi et al., 2010; Lafarge & Mallet, 2011; Shan, Adams, Curless, Furukawa, & Seitz, 2013), the resulting reconstructed models remain not completely photorealistic, and they cannot easily be compared directly with live sensor data. This type of representation is heavily influenced by 3D modeling tools provided by the computer graphics community, who are only interested in rendering visually pleasing scenes but not in representing the sensor data in the best possible way. Realistic computer graphics models are obtained by texture mapping onto 3D planar façades, which have been extracted from the environment. Unfortunately, this type of approximation introduces many artefacts and visual inconsistencies that cannot easily be overcome by robust estimation methods. To be robust to these errors, Caron, Dame, and Marchand (2014) proposed using a mutual information approach (Viola & Wells, 1995) in order to register a synthetic image generated from an approximate textured 3D model with a real image. This metric allows us to compare highly different images (between the sensor and model), however the calculations required for the alignment are computationally expensive.

### 3.2. Image-based Models

An alternative approach to using parametric 3D models based on features is to represent the environment using an image-based approach, also known as an egocentric model. In this case, key-frame images, acquired during a mapping phase, are stored in a database, and the sensor data remain in their raw unmodified form. The learning phase also allows us to estimate the transformations between the key-frames and store their position in global world coordinates. The local information contained in the key-frames is then used to estimate the position of a camera navigating within the premapped environment. Unlike the parametric methods, these maps are much simpler to acquire and they provide greater accuracy and robustness since the raw sensor data have not been approximated and a direct image error can be minimized. Furthermore, the topological connections between the key-frames remove the necessity to represent all the local map data in a global reference frame, and they prevent the introduction of bias due to the choice of the world coordinate system. In this case, the local sensor data remain accurate and they are not contaminated by the global drift in the sensor pose.

In the literature, the proposed quantitative localization methods using key-frames all rely on feature-based techniques (Courbon, Mezouar, & Martinet 2008; Royer et al., 2005). To provide a maximum field of view, which better constrains 6 DOF pose estimation (Baker, Fermuller, Aloimonos, & Pless, 2001), a spherical representation is

proposed. This kind of representation, already found in commercial applications such as *Google Earth* [cf. Anguelov et al. (2010)], allows an immersive visualization of the scene using panoramic images, but it is not adapted for accurate localization or navigation since the spherical images are very sparsely positioned in the world, and accurate dense 3D maps are not available, even though some recent work (Klingner, Martin, & Roseborough, 2013) shows that SFM can be used to reconstruct dense point clouds and improve the image locations.

### 3.3. Local Dense Representation: Augmented Visual Sphere

In the representation proposed in this paper, a local area in space is defined by an augmented spherical key-frame that captures the light field of all viewing directions from a particular point in 3D space along with its depth map (see Figure 1). This allows us to extend robust and accurate state-of-the-art dense full-image approaches (Comport et al., 2007) to their most general form (i.e., all viewing directions are densely modeled). To formalize this, each sphere will be defined by the set

$$\mathcal{S} = \{\mathbf{I}_S, \mathbf{p}_S, \mathbf{Z}_S, \mathbf{W}_S\}, \quad (9)$$

where  $\mathbf{I}_S$  is the photometric spherical image (this image is obtained from the custom camera system presented in Section 4.3 by warping multiple images onto the sphere);  $\mathbf{p}_S = \{\mathbf{q}_1, \dots, \mathbf{q}_n\}$  is a set of evenly spaced points on the unit sphere where  $\mathbf{q} \in S^2$  [these points have been sampled uniformly on the sphere as in Meilland et al. (2010)];  $\mathbf{Z}_S$  are the depths associated with each pixel, which have been obtained from dense stereo matching [the 3D point is subsequently defined in the sphere as  $\mathbf{V}_S = (\mathbf{p}_S, \mathbf{Z}_S)$ ]; and  $\mathbf{W}_S$  is a saliency image that contains knowledge of good pixels to use for tracking applications. It is obtained using a non-linear pixel selection, computed by analyzing the Jacobian of the warping function as detailed in Section 3.6.

### 3.4. Global Topologic Representation: Omnidirectional Key-frame Graph

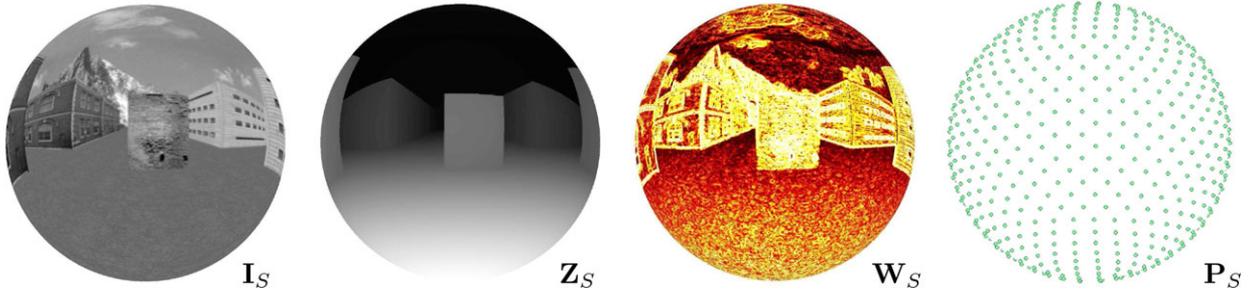
On a global scale, each local spherical key-frame will be connected by a topological graph of poses as shown on Figure 2. This egocentric 3D model of the environment is defined by the following graph:

$$\mathcal{G} = \{\mathcal{S}_1, \dots, \mathcal{S}_n; \mathbf{x}_1, \dots, \mathbf{x}_m\}, \quad (10)$$

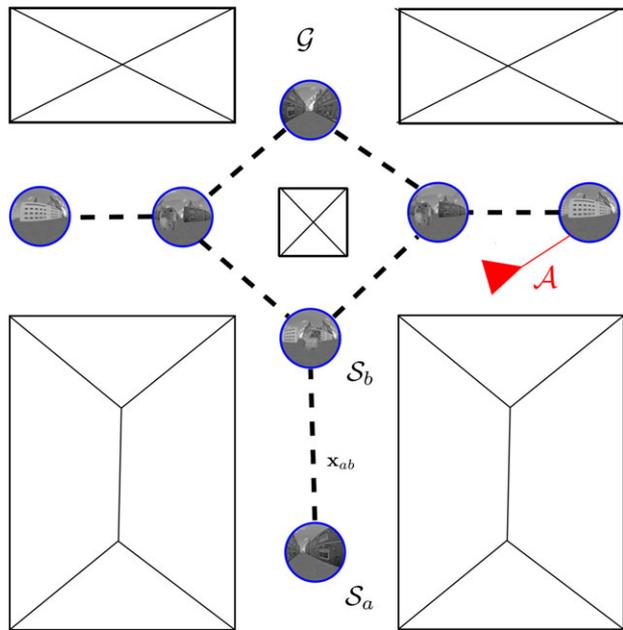
where  $\mathcal{S}_i$  are *augmented spheres* that are connected by a minimal parametrization  $\mathbf{x}$  of each pose.

The main advantages of this spherical representation are as follows:

- An omnidirectional key-frame representation allows local sensor data to be conserved (without transformation) and subsequently ensure photometric consistency. This



**Figure 1.** Local representation: augmented sphere  $\mathcal{S}$  containing intensities  $\mathbf{I}_S$ , depth-map  $\mathbf{Z}_S$ , saliency  $\mathbf{W}_S$ , and a sampling  $\mathbf{p}_S$ .



**Figure 2.** Egocentric representation: graph of spheres  $\mathcal{G}$  allowing the localization of a vehicle  $\mathcal{V}$  navigating locally within the graph.

enhances the performance of direct registration techniques (accuracy, robustness, convergence speed, and domain of convergence).

- A topological graph representation encodes the viewing trajectory, which allows the uncertainty associated with the sensor measurements to be retained, yet a 3D model can be generated from this representation if needed.
- The map can be accessed in constant time independently of the graph size.
- Augmenting photometric spherical panoramas with dense depth enables the performance of local novel view synthesis necessary for direct registration techniques.
- A spherical representation provides all local view directions and therefore provides a general representation that can encode data from different kinds of sensors, such as

perspective cameras, multiview cameras, or omnidirectional cameras and laser range finders.

- Full-view sensors condition the observability of 3D motion well (Baker et al., 2001), which greatly improves robustness.
- This representation can be made invariant to illumination variation as in Meilland et al. (2011b).

### 3.5. Global Spherical Key-frame Positioning

To accurately recover the position of the spheres of the graph with respect to one-another, a 6 DOF spherical localisation model is used based on an accurate dense localisation (Comport et al., 2010; Meilland et al., 2010). Considering  $\mathcal{S}^*$ , an augmented sphere defined in Section 3.3, the objective is to compute the pose between a reference sphere  $\mathcal{S}^*$  and the next one,  $\mathcal{S}$ . The localisation problem is solved using a dense visual odometry approach that uses a direct 3D image registration technique that provides accurate pose estimation. Since this is a local optimization approach, it is assumed that the camera frame-rate is high (30 Hz) and that interframe displacements are small.

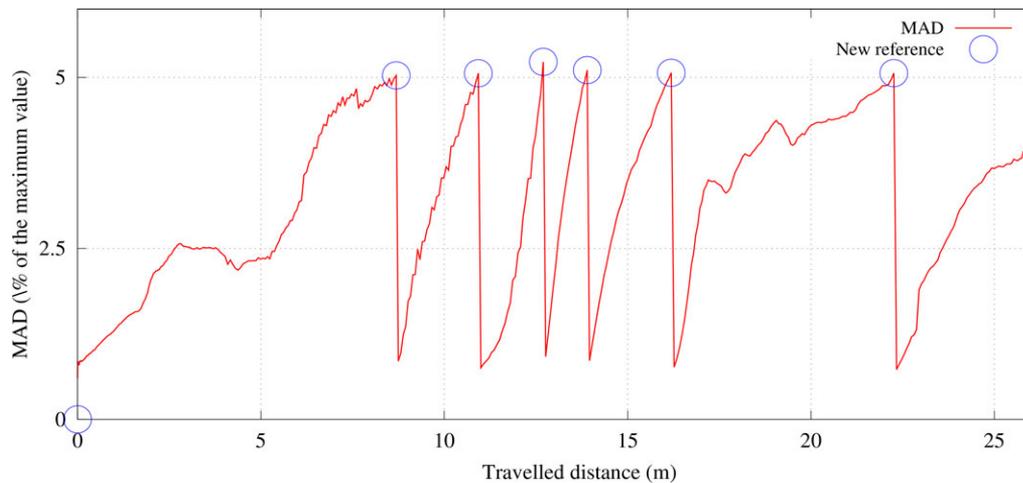
The intensity error measure between a reference sphere and a current sphere is then defined as follows:

$$\mathbf{e}(\mathbf{x}) = \rho(\mathbf{I}_s[w(\widehat{\mathbf{T}}\mathbf{T}(\mathbf{x}); \mathbf{Z}_S^*, \mathbf{p}_S^*)] - \mathbf{I}_s^*(\mathbf{p}_S^*)), \quad (11)$$

where  $w(\cdot)$  is the spherical warping function, which transfers 3D points onto the current sphere. The direct minimization procedure was given in Section 2. The main difference with respect to Eq. (4) is that the sphere is sampled, uniformly if possible, within polar spherical coordinates. The interested reader can refer to Meilland et al. (2010), where the HealPix algorithm (Gorski et al., 2005) is used for uniform sampling.

#### 3.5.1. Spherical Node Selection

Indeed, the vertices of the graph should be carefully placed in the world so as to represent the environment with little redundancy. One preliminary technique to achieve this goal locally is to observe criteria between an initially selected reference sphere and surrounding spheres. In practice,



**Figure 3.** Evolution of the median absolute deviation (MAD) with respect to the traveled distance. The MAD value increases differently with respect to the scene configuration: between  $0 < \Delta < 8$  and  $16 < \Delta < 25$ , the scene geometry is far from the observation viewpoint. Between  $8 < \Delta < 16$ , the scene geometry is close to the sensor. A new reference is reconstructed when the MAD exceeds 5% of the maximum intensity error (e.g., 255 gray levels).

the trajectory of the acquisition system along a sequence is computed by integrating elementary displacements estimated from successive spherical registration. The strategy used here is to maintain as long as possible the reference sphere to minimize the drift introduced when a new reference sphere is taken. Therefore, a new reference sphere is placed according to the median absolute deviation (MAD):

$$\lambda < \text{median}[|\mathbf{e} - \text{median}(\mathbf{e})|], \quad (12)$$

where  $\mathbf{e}$  is the vector of intensities error between the current sphere and the previous set of reference spheres defined in Eq. (11) at the end of the minimization.

This criterion is directly related to the geometry of the environment via the warping function of Eq. (11), providing a robust measure of the photometric changes between the images, allowing us to quantify:

- the amount of occlusions related to the scene geometry and the sensor viewpoints,
- the effective resolution changes related to the distance between the images.

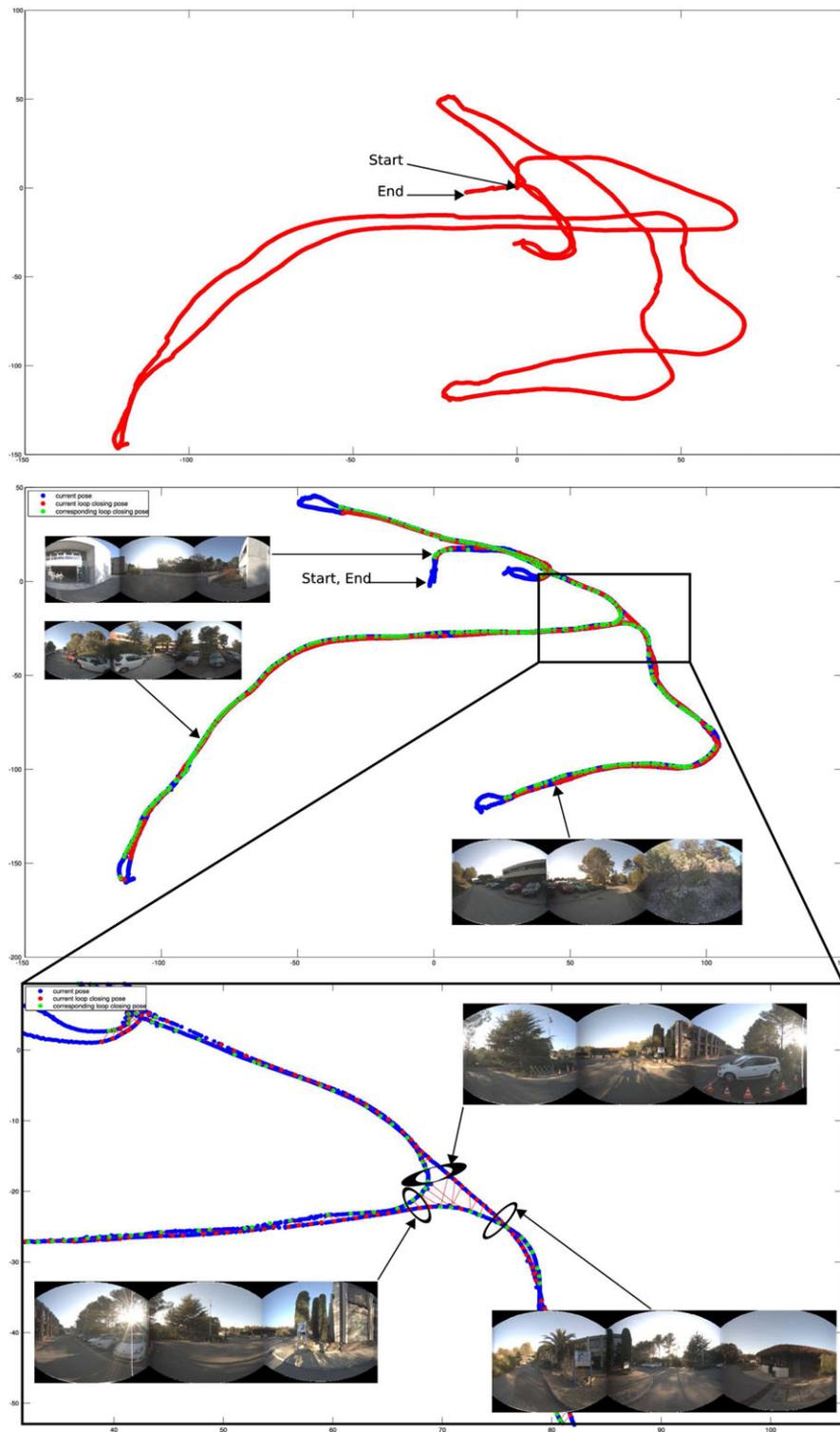
Figure 3 shows an example of the evolution of the median absolute deviation for a 500-image trajectory with respect to the distance  $\Delta$  traveled by the vehicle. When the environment is far from the camera viewpoint ( $0 < \Delta < 8$  and  $16 < \Delta < 25$ ), the MAD increases slowly, due to very small changes between the images. When the environment is closer to the camera ( $8 < \Delta < 16$ ), the MAD value increases rapidly until the selection threshold  $\lambda$  is reached, which allows us to sample more spheres only when necessary.

### 3.5.2. Loop Closure Detection and Drift Compensation

Since a dense spherical visual odometry technique was used, small amounts of drift are integrated along the sequence (typically 1%, which could cause the final graph to become inconsistent or redundant (i.e., a route mapped twice in opposite directions)). To correct the drift and remove redundant spheres, the spherical loop closing technique proposed by Chapoulie, Rives, & Filliat (2011) was used. This method is based on SIFT descriptors augmented by their distribution across the sphere, which is represented by histograms. A dictionary is incrementally built online, which allows us to detect images of similar appearance, regardless of the image orientation. This approach was specifically designed for the spherical key-frame representation proposed in this paper.

When a loop closure is detected, a dense registration is performed between the candidates and the detection is validated if the residual error after the minimization is below a predefined threshold. Using the new pose provided by the loop closure, the graph is augmented with the new constraints and then optimized using the pose-graph library TORO (Grisetti, Stachniss, Grzonka, & Burgard, 2009). This open-source library implements the method given in Grisetti, Grzonka, Stachniss, Pfaff, and Burgard (2007), which is an extension of Olson, Leonard, and Teller (2006). It allows the errors introduced by the new constraints in the graph to be minimized by a stochastic gradient descent method.

Figure 4 shows the benefits of using the loop closures. The top image corresponds to the estimated trajectory done with only visual odometry. The start point and the end point should be at the same position. The middle image



**Figure 4.** Drift correction using loop closures: Top image corresponds to the estimated trajectory done with only visual odometry. The start point and the end point should be at the same position. Middle image corresponds to the corrected trajectory taking into account loop closures. Bottom image details a place where loop closures are detected from perpendicular points of view.

corresponds to the corrected trajectory, taking into account loop closures. The bottom image details a place where loop closures are detected from perpendicular points of view.

In addition to compensating for the drift, the loop closure detection method allows us to drastically reduce the size of the graph by merging the redundant spherical views.

### 3.6. Information Selection

The essence of direct approaches is to minimize pixel intensities directly between images. Images are, however, clearly redundant, i.e., a lot of information is not overly important for navigation, such as completely untextured parts. This kind of information should not be used for several reasons. One of them is that if spheres are constructed from a panoramic multicamera system at high resolution, the number of pixels to warp at each iteration of the real-time minimization process could be extremely large. Therefore, reducing the dimension is essential for real-time computing.

A classic approach (Baker & Matthews, 2001; Comport et al., 2007) often employed with direct techniques is to sort and select only the best corners or edges (intensity gradients) in gray-level images:

$$i = \arg \max_i \|\nabla I(i)\|. \tag{13}$$

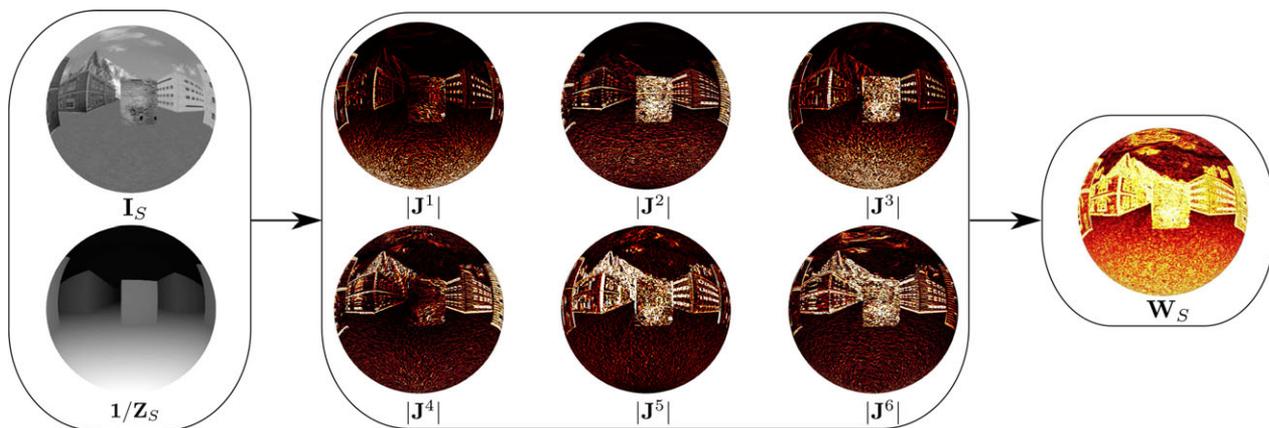
This naive approach does not consider the importance of the structure of the scene and can lead to the selection of nonobservable measurements. For example, selecting high-intensity gradient points at infinity, rather than more informative but low gradient close points in a scene, will yield imprecise results. In the worst-case scenario, it will not even be possible to estimate translation because of the invariance of infinite points in pure translation. At the same time, infinite points are also useful because they improve the stability

and robustness of the tracking algorithm due to their small displacement in the images for large translations. The difficulty is therefore to analyze the accuracy and robustness of selected points. In Dellaert & Collins (1999), a saliency map is built by sorting the pixels with respect to the variance of their Jacobian. A subset is then randomly extracted from the best pixels. This approach is finally applied to estimate only rotations. In Benhimane, Ladikos, Lepetit, and Navab (2007), only linear or quadratic subsets are extracted for template-based tracking. Such subsets are good for linear convergence, allowing it to converge quickly. Unfortunately, this method suggests rejecting strong edges that are essential for obtaining good precision.

The novelty of the approach proposed here is to quantify the effect of the geometric structure and the image intensity measurements on navigation by analyzing directly the entire analytical Jacobian, which relates scene movement to sensor movement. The aim is to select points that best condition the six degrees of freedom of the pose. Indeed, the reference Jacobian matrix  $J(\tilde{x})$  directly combines the gray-level gradient (image derivatives) and the geometric gradient such that  $J(\tilde{x}) = J_I \cdot J_G$ . This matrix can be decomposed into six parts corresponding to each degree of freedom of the transformation [called *steepest descent images* by Baker & Matthews (2001)]:

$$J(\tilde{x}) = [J^1 \ J^2 \ J^3 \ J^4 \ J^5 \ J^6] \in \mathbb{R}^{nm \times 6}. \tag{14}$$

Each column vector of  $J$  contains the gradient associated to one degree of freedom, and it can be interpreted as an  $m \times n$  saliency map after reordering its elements into a matrix. Figure 5 shows an example of six saliency maps, where a bright value indicates a strong gradient. The first three images ( $|J^1|$ ,  $|J^2|$ , and  $|J^3|$ ) show the translational part: close 3D measurements in the scene have a strong gradient.



**Figure 5.**  $I_S$ : Reference image.  $Z_S$ : Depth-map.  $|J^i|$ : absolute value of the reordered Jacobian matrix for each degree of freedom.  $W_S$ : final saliency map after selection (first selected pixels are bright, last pixels are dark). From the six saliency maps, corresponding to the six columns of the Jacobian matrix, the proposed algorithm iteratively selects the best pixels to produce a final saliency map containing the sorted values, which maximizes the observability of each degree of freedom.

The last three images ( $|J^4|$ ,  $|J^5|$ , and  $|J^6|$ ) show the rotational components, which are invariant to the scene geometry.

The objective here is to extract a subset  $\bar{J} = \{\bar{J}^1, \bar{J}^2, \bar{J}^3, \bar{J}^4, \bar{J}^5, \bar{J}^6\} \subset J$ , of dimensions  $p \times 6$ , with  $p \ll nm$ , containing the pixels that best condition each degree of freedom of  $J$ .

The proposed approach is a sorting algorithm, where each line  $k$  of the new matrix  $\bar{J}$  is obtained by

$$\bar{J}_k = \arg \max_j (|J_i^j| \setminus \bar{J}), \quad (15)$$

which corresponds to selecting an entire line of the original matrix  $J$ , corresponding to the best gradient of the  $j$ th column.  $\bar{J} \subset J$  is an intermediate subset containing the lines of  $J$  that are already selected, and  $\setminus \bar{J}$  prevents selecting the same line twice. The term (15) is iteratively applied for each degree of freedom, allowing the same number of pixels to be selected for each degree of freedom, until all pixels are sorted.

In practice, the matrix  $\bar{J}$  is not explicitly reconstructed (cf. algorithm 1); the indices of each line of  $J$  are stored in saliency order in the saliency map  $W_s$ . During the learning phase,  $J(\bar{x})$  is precomputed for each reference image and all the pixels are sorted according to Eq. (15). During the on-line localization phase, a dynamic selection of the pixel is performed with respect to the saliency map and the current camera viewpoint, allowing us to use only the best  $p$  pixels that are seen by the camera, while maximizing the observability of the 6 DOFs. In the next warping functions, this dynamic selection will be denoted by the function  $s(\cdot)$ :

$$(Z^s, p^s) = s(Z, p), \quad (16)$$

which returns the best subset of pixels from the saliency map  $W_s$ .

Since a multiresolution approach is employed for the registration, the selection algorithm is performed on each resolution of the reference image and stored in the database.

---

#### Algorithm 1 Selection algorithm

---

**Require:**  $J \in \mathbb{R}^{nm \times 6}$   
**Ensure:**  $W_s \in \mathbb{N}^{nm}$   
 $k \leftarrow 1$   
**for**  $i = 1 \rightarrow nm/6$  **do**  
    **for**  $j = 1 \rightarrow 6$  **do**  
         $index \leftarrow \arg \max_i (|J_i^j| \setminus \bar{J})$   
        **for**  $j = 1 \rightarrow 6$  **do**  
             $\bar{J}_k^j \leftarrow J_{index}^j$   
        **end for**  
         $W_s(k) \leftarrow index$   
         $k \leftarrow k + 1$   
    **end for**  
**end for**  
 Return  $W_s$ .

---

## 4. SPHERICAL SENSORS

The spherical key-frame model presented in the previous section requires an adequate system for acquiring these types of measurements. Due to the novelty of this representation, however, no classical sensors were available, and it was necessary to develop a customized system. As such, several classic omnidirectional and panoramic camera systems are analyzed for this purpose, and two new multiview RGB-D camera systems are presented.

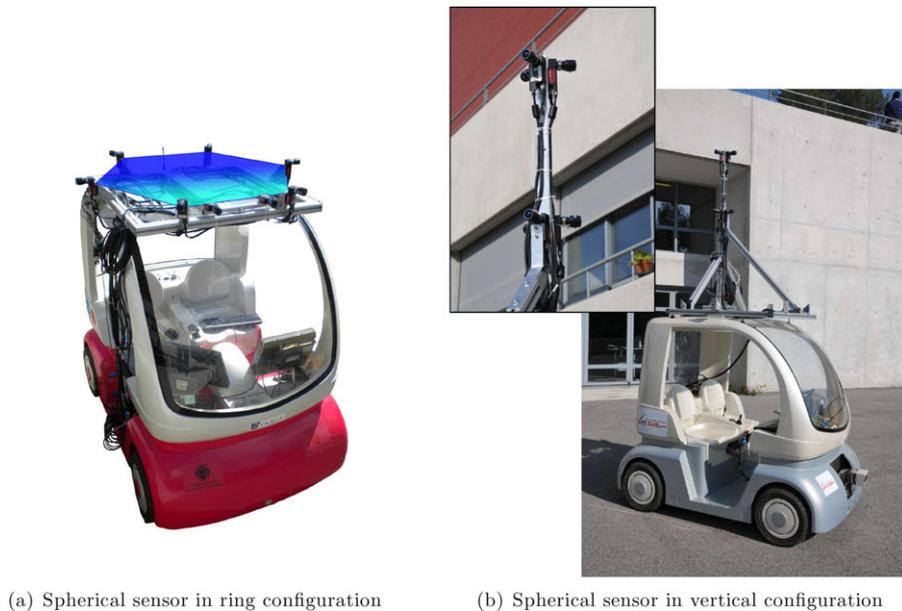
### 4.1. Photometric Sphere

Capturing omnidirectional views of the environment can be achieved using either omnidirectional catadioptric cameras (Nayar, 1997) or with multiple perspective cameras. An omnidirectional camera allows us to capture panoramic images with a  $360^\circ$  horizontal field of view. However, it suffers from a low and nonuniform spatial resolution and a limited vertical field of view (half-sphere,  $<90^\circ$ ). This kind of sensor is therefore not efficient for outdoor mapping. Multiview stitching techniques allow us to build high-resolution panoramic images by stitching and blending perspective images together (Szeliski, 2006). Those images can be acquired using several cameras (Baker et al., 2001), or from a sequence of images (Lovegrove & Davison, 2010). This kind of approach always assumes a unique center of projection and may suffer from parallax artefacts.

### 4.2. Depth Sphere Acquisition

**Passive systems** are capable of simultaneously capturing a dense depth-map from cameras alone without any additional system. To the best of our knowledge, only omnidirectional stereo viewing systems have been developed and studied (Arıcan & Frossard, 2007; Caron et al., 2011; Goncalves & Araujo, 2004; He, Luo, Geng, Zhu, & Hao, 2007; Lui & Jarvis, 2010; Micusik & Pajdla, 2004; Ragot, Rossi, Savatier, Ertaud, & Mazari, 2008). Multiview spherical systems have also been widely studied in the literature, however it is generally assumed that all cameras view the same object and that the cameras are all converging to that point (Kutulakos & Seitz, 2000). On the other hand, diverging multicamera systems (Baker et al., 2001; Szeliski, 2006) all aim to have a common center of projection and are unable to compute a depth image.

**Active systems** require projecting laser or light patterns onto the environment to obtain dense depth measurements. In Gallegos, Meilland, Rives, and Comport (2010), a spherical image is obtained using a classic omnidirectional camera. The depth information is obtained using a plan laser range finder (LRF) aligned with the camera optical center. This approach assumes a planar and structured environment to propagate the depth onto the sphere. In Cobzas et al. (2003), a similar idea is used but a pan/tilt perspective camera is used with a Lidar to



**Figure 6.** Spherical RGB-D sensors.

reconstruct cylindrical augmented panoramas. Active RGB-D sensors such as Microsoft Kinect and Asus Xtion are actually popular in the robotics community. The projection of an infrared pattern allows us to recover dense depth-maps in real-time and can be used in SLAM systems such as Audras, Comport, Meilland, and Rives (2011), Henry, Krainin, Herbst, Ren, and Fox (2010), and Newcombe et al. (2011a). In Spinello and Arras (2011), three sensors are used to reconstruct a larger field of view. This kind of sensor is, however, sensitive to sunlight and is therefore limited to indoor mapping.

### 4.3. Spherical Acquisition System

To acquire augmented visual spheres, for large-scale environment mapping, a first multicamera system was proposed initially in Meilland et al. (2011a). This system will not be detailed here, but the interested reader can refer to that article. This sensor, shown in Figure 6(a), was composed of six wide-angle stereo cameras, placed in a ring configuration so that dense stereo matching can be performed between each camera pair. This compact configuration allowed to build spherical photometric panoramas with a depth information associated with each pixel of the sphere. One of the major drawbacks of this system is that the angles between the optical axes are clearly diverging ( $60^\circ$ ), and wide-baseline stereo matching is necessary to handle the large differences in image resolution. Several dense matching algorithms have been tested, from standard block matching to more advanced approaches (Geiger, Roser, & Urtasun, 2010; Hirschmuller, 2008; Kolmogorov & Zabih, 2001; Ogale &

Aloimonos, 2005; Tola, Lepetit, & Fua, 2010). Hirschmuller (2008) and Geiger et al. (2010) performed fairly, but the large differences in resolution lead to several mismatches, which result in sparse depth-maps. Since the depth information is necessary to correctly warp and blend the intensities onto the final sphere, the resulting photometric spheres were not fully dense.

To overcome the limitations of the first sensor, a second configuration has been studied. This sensor system, similar to the MARS sensor (Earthmine, 2009), is composed of three stereo pairs back-to-back in a vertical configuration, as can be seen in Figure 6(b). Each triplet of cameras is considered to have a unique center of projection, allowing us to use standard stitching algorithms to build the photometric spheres. This configuration is similar to those using omnidirectional stereo cameras, such as Lui & Jarvis (2010), except that using a multicamera system allows us to reconstruct high-resolution spheres, which are necessary for outdoor mapping. The procedure to build an augmented sphere with this system can be summarized as follows:

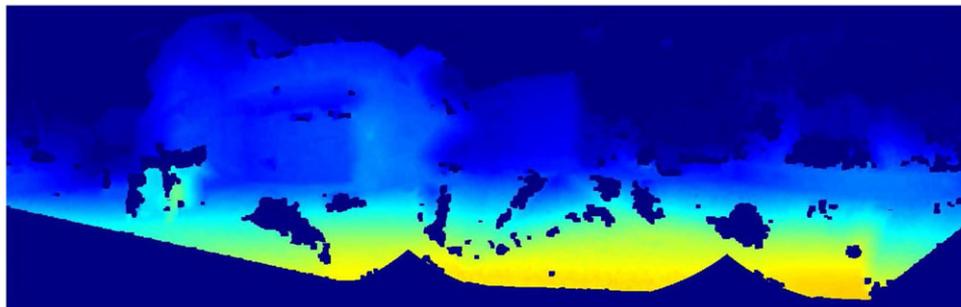
- (1) Calibration of the extrinsic and intrinsic parameters of each camera with a checker-board, via bundle adjustment: this calibration step is only performed once since the system is rigid.
- (2) For each triplet of cameras, perform image stitching with Laplacian blending (Burt & Adelson, 1983), which results in two spherical panoramas with a vertical baseline (top and bottom; see Figure 7).
- (3) Rectification of the spherical panoramas to ensure the epipolar constraints on the spheres.



(a) Top spherical image



(b) Bottom spherical image



(c) Corresponding dense spherical depth map obtained using (Geiger et al., 2010)

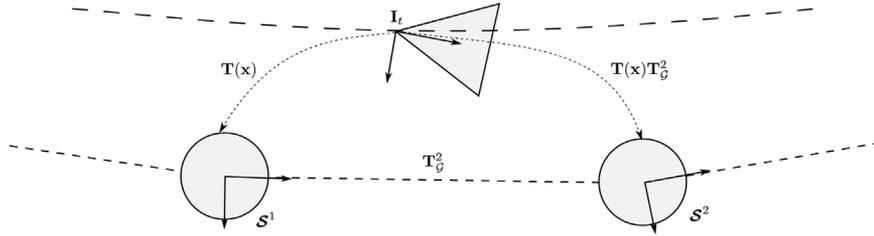
**Figure 7.** Example of an RGB-D panorama obtained using the new vertical system.

- (4) Dense matching between the spheres.
- (5) Triangulation between the spheres to recover depth.

In this configuration, dense matching is much more robust since the sensor is composed of three classic fronto-parallel stereo pairs back to back. Since the images can be directly warped onto the spheres by stitching, the resulting photometric spheres are also dense. One potential issue is that due to the vertical alignment, vertical lines in the scene are collinear with the epipolar lines: this may create ambiguities during dense matching, especially in structured environments (in practice, this degenerated case was not ob-

served in outdoor environments). Another issue with this system is the approximation of a unique center of projection for each camera triplet, which can lead to parallax artefacts when stitching the images. In practice, those effects are only visible for very close objects (e.g.,  $< 1$  m), which is not an issue when mapping outdoor environments.

As can be seen in Figure 7, this new system provides high-quality photometric spheres and depth-maps, since dense matching can be performed directly between the spherical panoramas, leading to consistent results when using a semiglobal approach such as that of Hirschmuller (2008) or Geiger et al. (2010).



**Figure 8.** The image  $I_t$  is localized using simultaneously the nodes  $S^1$  and  $S^2$ , which ensure a smooth localization.

## 5. REAL-TIME LOCALIZATION

This section will present several techniques that have been used to improve the real-time performance of 6 DOF pose estimation with respect to the previously described dense key-frame graph of augmented spherical images. First of all, it will be shown how the graph is used for real-time monocular pose estimation in an asymmetric manner (i.e., a single perspective image is compared to the closest augmented spherical image). Following that, a multiple reference sphere localization that provides a smooth localization will be proposed along with a technique that improves robustness in dynamic environments. Finally, an approach to first estimating only rotation before refining the full pose will be given that greatly improves convergence speed.

### 5.1. Real-time Monocular 6-DOF Localization

#### 5.1.1. Online Sphere Selection

It is assumed that during online navigation, a current image  $I$ , captured by a generic camera (e.g., monocular, stereo or omnidirectional), and an initial guess  $\hat{T}_G$  of the current camera position within the graph are available. To choose the closest sphere for tracking within the graph, it is necessary to define a metric. Contrary to nonspherical approaches, a sphere provides all viewing directions, and therefore it is not necessary to consider the rotational distance (to ensure image overlap). The closest sphere is subsequently determined uniquely by translational distance between the initial estimate  $\hat{T}_G$  and the sphere's poses:

$$T_s^* = \arg \min_i \left[ \left\| \mathbf{e}_4^T (T_G^i)^{-1} \hat{T}_G \right\| \right] \quad \forall i \in \mathcal{G}, \quad (17)$$

where  $\mathbf{e}_4 = [0 \ 0 \ 0 \ 1]^T$  is a vector that extracts the translational part of a pose  $T$ .

In particular, this prevents choosing a reference sphere that has similar rotation but large translational difference, which induces self-occlusions of buildings and also differences in image resolution caused by distance (which affects direct registration methods).

#### 5.1.2. Efficient Minimization

Since a sphere provides all local information necessary for 6 DOF localization (geometric and photometric information),

an accurate estimate of the pose is obtained by an efficient direct minimization:

$$\mathbf{e}(\mathbf{x}) = \mathbf{I} \left( w(\hat{T}T(\mathbf{x}); s(Z_s^*, \mathbf{p}_s^*)) \right) - \mathbf{I}_s^* \left( s(Z_s^*, \mathbf{p}_s^*) \right), \quad (18)$$

where  $\mathbf{x}$  is the unknown 6 DOF pose increment obtained as detailed in Section 2. The warping function  $w(\cdot)$  transfers the current image intensities onto the reference sphere via both the perspective and spherical warping functions given in Eqs. (A.1) and (A.4), using the depth information  $Z_s$  of the reference sphere. The function  $s(\cdot)$ , introduced in Section 3.6, selects only informative pixels with respect to a saliency map  $\mathbf{W}_s$ , which is already precomputed on the reference sphere and stored in the graph. This selection speeds up the tracking algorithm without degrading either the observability of 3D motion or the accuracy.

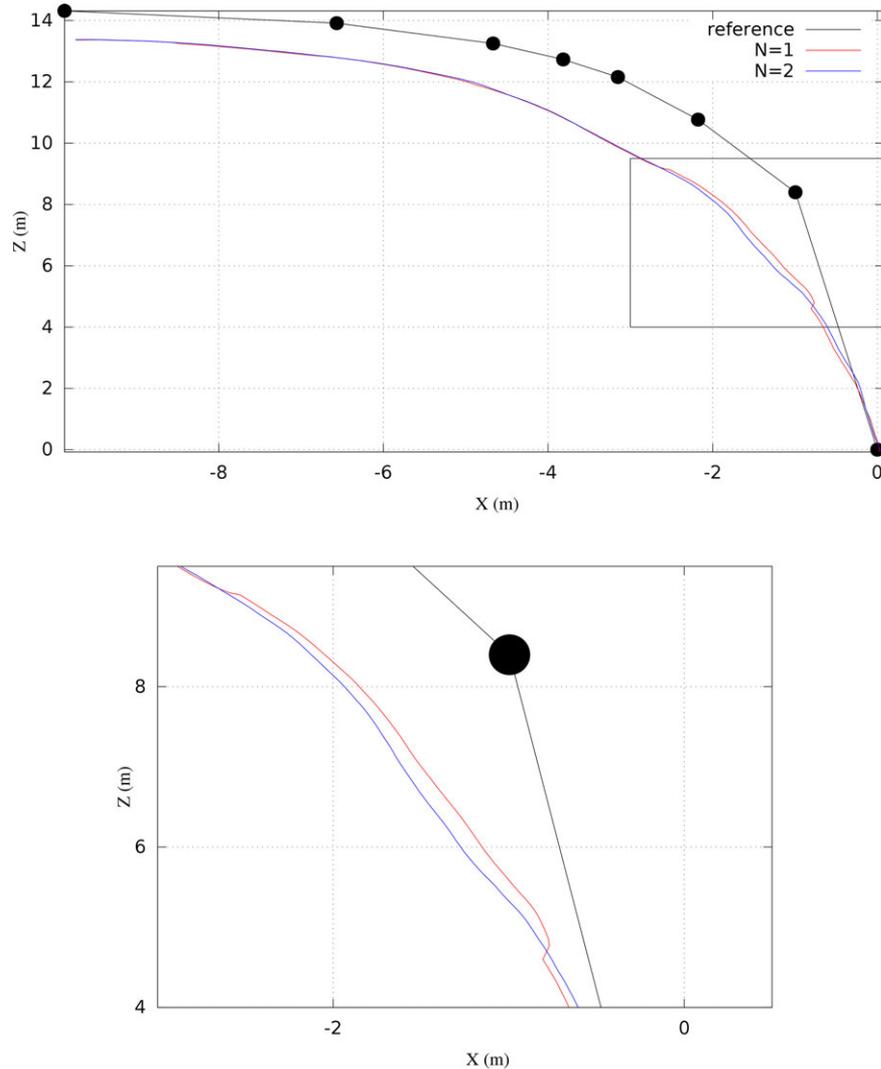
#### 5.1.3. Smooth Localization

During this localization phase, the live current image is permanently registered onto the closest reference sphere, which is retained for localization until a new sphere is selected according to criterion (17). As the current image gets further from the reference image, the amount of occlusions and outliers increases due to viewpoint and resolution changes, leading to localization inaccuracies. This can create discontinuities in the trajectory when a new reference sphere is chosen (see Figure 9). Even if these discontinuities are small (e.g., a few centimeters), they can impact the visual servoing used for autonomous driving.

To obtain smoother trajectories and a more robust localization, it is proposed here to use simultaneously two reference images in the minimization, successively selected by the criteria (17). This usually corresponds to selecting the previous and next spheres in the graph, as is shown in Figure 8.

The localization problem can be formulated as a simultaneous minimization of two intensity errors:

$$\mathbf{e}_G(\mathbf{x}) = \begin{bmatrix} \mathbf{I} \left( w(\hat{T}_s T(\mathbf{x}) \mathbf{I}; s(Z_s^1, \mathbf{p}_s^1)) \right) - \mathbf{I}_s^1 \left( s(Z_s^1, \mathbf{p}_s^1) \right) \\ \mathbf{I} \left( w(\hat{T}_s T(\mathbf{x}) T_G^2; s(Z_s^2, \mathbf{p}_s^2)) \right) - \mathbf{I}_s^2 \left( s(Z_s^2, \mathbf{p}_s^2) \right) \end{bmatrix}, \quad (19)$$



**Figure 9.** Multinode localization: The black disk represents the position of the reference spheres. Red denotes the trajectory estimated using only the closest reference node. The trajectory contains discontinuities when switching reference. Blue denotes the same trajectory, estimated using the two closest nodes. The obtained trajectory is continuous, which better represents the expected trajectory of the vehicle.

where  $T_G^2$  is the pose of the second reference sphere  $S^2$  with respect to the first reference sphere  $S^1$ .

Using two reference images in the minimization introduces more pixels in the cost function. To maintain high-frequency tracking, the number of salient pixels used in the minimization is simply shared between the spheres. Note that Eq. (8) can easily be expanded to more reference spheres, but it appears experimentally that two spheres are sufficient for smooth and robust localization. Indeed, using reference images that are too distant from the current view introduces additional occlusions that reduce the efficiency of the minimization.

Figure 9 compares a single-node localization (red) with a multinode localization (blue). When using a single node, it can be seen that the trajectory is not continuous when the reference image is switched (around  $X = -1$ ,  $Z = 5$ ). When using simultaneously two nodes, the estimated trajectory remains smooth.

## 5.2. Efficient Estimation of Local 3D Rotations

Typically, the apparent motion of pixels in an image is dominated by rotation motions. Indeed, for a pure rotation, the pixel's motion is independent of the scene geometry. For a

**Table I.** Computing time per step of the real-time localization algorithm for 60k pixels.

Step	Time
Reference images selection	<0.01 ms
Gaussian pyramid	0.70 ms
Interframe rotation estimation	0.48 ms/iteration
6 DOF localization	1.13 ms/iteration

vision-based localization algorithm, large rotations between successive frames are often prone to failure. To ensure a better initialization, a direct registration technique is employed to estimate the local interframe 3D rotation of the camera as in Lovegrove and Davison (2010), Mei et al. (2010), and Newcombe, Lovegrove, and Davison (2011b).

The smallest image in the multiresolution pyramid at time  $t$  is registered with the corresponding image at time  $t - 1$ . Since those images are generated from successive Gaussian filters and subsampling, rotations are clearly dominant and translations can be neglected. The corresponding error is then obtained as follows:

$$\mathbf{e}(\mathbf{x}_\omega) = \mathbf{I}_t(\mathbf{w}(\mathbf{K}\hat{\mathbf{R}}\mathbf{R}(\omega)\mathbf{K}^{-1}; \mathbf{p})) - \mathbf{I}_{t-1}(\mathbf{p}), \quad (20)$$

where  $\mathbf{K}$  are the intrinsics parameters of the camera,  $\hat{\mathbf{R}}$  is the initial rotation guess,  $\omega \in \mathbb{R}^3$  contains the unknown angular velocities, and  $\mathbf{R}(\omega) \in \mathbb{SO}(3)$  is obtained by the exponential matrix of  $[\omega]_\times$ . This error can be efficiently minimized as described in Section 2.2, and it allows us to quickly recover large angular motions. Since a lower resolution image is used, it is possible to perform dense minimization (see computing times in Table I).

### 5.3. Dynamic Environment Mapping

In practice, outdoor environments have very challenging dynamic effects that up until now have been handled as outliers, however when both extreme illumination variations and dynamic moving objects are evolving within the scene, the robust estimator breaks down. As such, we have also created an additional model for the illumination variations, whereby a global parameter is estimated for changes such as the sun while local variations are still handled as outliers. To remain robust to dynamic changes, a temporal visual odometry error is added to the model-based error introduced in Section 5.1.2. This consists in simultaneously registering the current image  $\mathbf{I}_t$  with the model  $\mathbf{I}^*$  and with the last registered image  $\mathbf{I}_{t-1}^w$ . In this case, the model-based error ( $\mathbf{e}_{\text{MB}}$ ) criterion ensures that the pose estimation does not drift but is not able to handle illumination change, while the visual odometry error ( $\mathbf{e}_{\text{VO}}$ ) is invariant to the large illumination change (variation interframe at 45 Hz is considered negligible).

Figure 10 shows an example of hybrid minimization, after registration of the current image for the experimental stereo dataset presented in Section 6.3.4. The scene contains both global illumination changes and local illumination changes, as well as dynamic occlusions (e.g., moving cars, bicycle, pedestrians). After registration, the model-based residual error  $\mathbf{e}_{\text{MB}}$  is large, due to the saturation of the building façade on the left, which is underexposed in the reference image  $\mathbf{I}^*$  and saturated in the current image  $\mathbf{I}_t$  at time  $t$ . Large errors are also generated by moving vehicles and by non-Lambertian reflections on the leaves of the trees. The associated M-estimator weights  $\mathbf{D}_{\text{MB}}$  are rejecting most of the information. On the other hand, the residual visual odometry error  $\mathbf{e}_{\text{VO}}$ , which is defined between the last warped image  $\mathbf{I}_{t-1}^w$  at time  $t - 1$  and the current image  $\mathbf{I}_t$ , is relatively low. Only moving objects are rejected by the M-estimator weights, which ensures a robust and fast convergence of the minimization. More details can be found in Meilland et al. (2011b).

### 5.4. Initialization and Tracking Failure

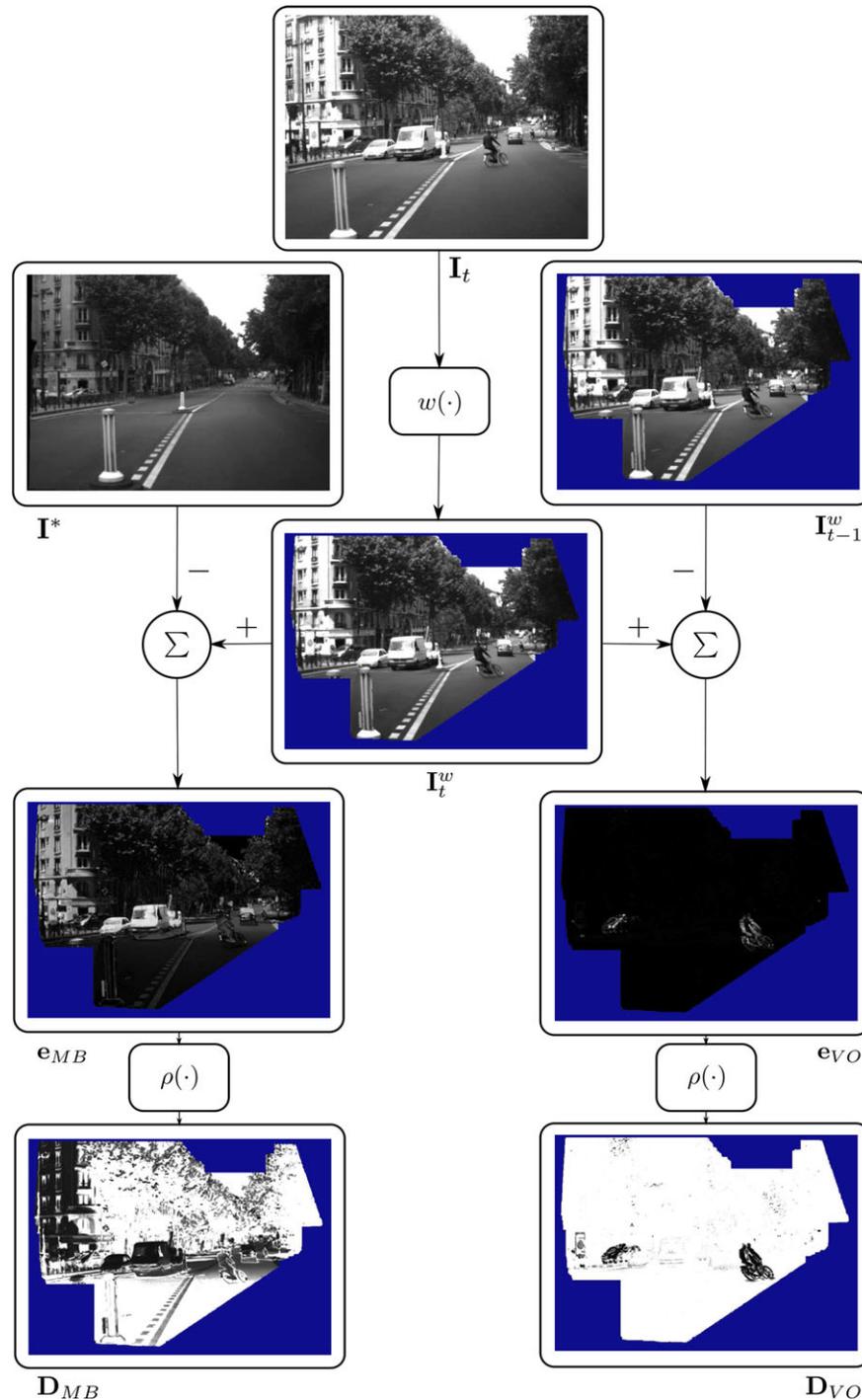
To handle initialization and tracking failure, a simple technique is employed. The current camera pose with respect to the reference sphere is supposed to be close to the identity. Dense registration is then performed with all the spheres of the graph, using the smallest image size in the pyramid. After alignment, the normalized cross correlation (NCC) is computed for each sphere of the graph, and the best score is used to select the reference sphere. Even if this initialization step does not fit well with large-scale databases, it needs less than 30 ms per reference sphere, which is 9 s for a 300-sphere graph. A dedicated technique such as that in Chapoulie et al. (2011) and Cummins & Newman (2008) will be investigated in future work.

## 6. FIELD EXPERIMENTS

In this section, several large-scale field experiments involving each step of the proposed framework are presented. First, dense spherical environment mapping experiments are presented. Following this, real-time localization and autonomous navigation experiments are detailed in the context of a national urban autonomous navigation challenge. Due to the overall complexity of the system and the breakup into a map learning phase and an online real-time navigation phase, it is suggested that the reader view the video associated with this paper before proceeding. The video illustrating these experiments can be found with the paper submission or alternatively at the following address: <http://youtu.be/wozzYVDQg2g>.

### 6.1. Implementation

A real-time implementation of the online localization algorithm has been made in C++. Most parts of the code are SSE



**Figure 10.** Hybrid visual tracking. Blue denotes depth information that is not available in the reference image. On the left-hand side, model-based registration (MB), the current image is registered with respect to the model, which contains large intensity errors ( $e_{MB}$ ). The associated M-estimator weights  $D_{MB}$  are rejecting most of the information. On the right and side, visual odometry registration (VO), a temporally close error is minimized ( $e_{VO}$ ). The M-estimator weights  $D_{VO}$  are only rejecting moving objects.

optimized (steaming SIMD extensions) in order to take advantage of native vector instructions of modern processors. The iterative minimization is performed in a multithreaded environment. It involves computing a Gaussian pyramid of the current image, warping the current image, computing robust statistics (median, MAD) using histograms, and computing the weighted Gauss-Newton Hessian and gradient approximations (see Section 2.2). The pose increment is then obtained using Cholesky factorization of the  $6 \times 6$  matrix, and the minimization is iterated until convergence.

Using a small fraction of salient pixels (i.e., 60k pixels), the localization algorithm runs flawlessly at 45 Hz with an input image of dimensions  $800 \times 600$  pixels. Table I shows the timings for each step of the algorithm obtained on a standard Core i7 laptop.

Since the localization algorithm is based on an iterative minimisation, the number of iterations can be directly adjusted with the desired camera frame-rate. As was highlighted in Handa, Newcombe, Angeli, and Davison (2012), a high camera frame-rate reduces the interframe camera motion, allowing us to perform fewer iterations until convergence. In our experiments, the maximum number of iterations for the interframe rotation estimation was set to 10. The 6 DOF localization is performed using a three-level image pyramid with, respectively, three, five, and eight iterations for each level of the pyramid (from coarse to fine).

## 6.2. Storage Cost and Memory Management

As detailed in Section 3, each sphere contains a photometric image  $I_S \in \mathbb{R}^{+n \times m}$ , which is directly converted into luminance and stored in a 16-bit gray-level image, a set of pixel coordinates  $\mathbf{p}_S \in \mathbb{R}^{2 \times n \times m}$ , which is the same for all spheres, a 32-bit floating point depth map  $\mathbf{Z}_S \in \mathbb{R}^{+n \times m}$ , and a saliency map  $\mathbf{W}_S \in \mathbb{N}^m$ , which contains a list of 32-bit unsigned integer indices. For each sphere, a three-level multiresolution pyramid is also precomputed. With a  $2,048 \times 1,024$  base resolution, the cost of one sphere is 26.25 MB (without compression). For the large-scale graph (310 spheres) reconstructed in Section 6.3.1, this represents 7.72 GB of memory. A better memory compression could be achieved using a lossless image compression algorithm. During online localization, only a subset of the entire graph of spheres is loaded in RAM. While the camera moves in the environment, a parallel CPU thread is used to stream out the farthest spheres and to stream in new spheres with respect to the current camera pose. This allows us to navigate seamlessly in very large maps.

## 6.3. Localization and Mapping Results

### 6.3.1. Dense Mapping—INRIA Sophia Antipolis

A sequence of  $7,364 \times 6$  images was acquired at 45 Hz at the INRIA Sophia-Antipolis site with an electric Cycab ve-

hicle and the onboard spherical system described in Section 4. The path traveled by the vehicle was approximately 1.5 km in length and is representative of a wide variety of outdoor environments, including wide open spaces, buildings, canyons, parked vehicles, vegetation, trees, tight turns, and varying altitudes. The aim of this is to test extensively the robustness and accuracy of the 6 DOF estimates of the vehicle pose. The learning phase, which involved creating the augmented spherical panoramas and positioning them within a graph, was calculated offline at about 1 Hz due to the large amount of data to be handled.

The selection of specific key-frames and the detection of loop closures allowed us to reduce the 7,364 initial images to 310 reference spheres with a compression rate of 96%. Figure 11 shows the trajectory that was obtained after detecting loop closures and correcting the graph. Some of the key-frame images are also shown.

### 6.3.2. Photorealistic Virtual Navigation

The final spherical representation can be used to synthesise photorealistic virtual images, using a technique similar to those developed in image-based rendering (Debevec, Taylor, & Malik, 1996; Gortler, Grzeszczuk, Szeliski, & Cohen, 1996; Levoy & Hanrahan, 1996). To demonstrate this, a real-time application was made with the OpenGL graphics rendering library, which allows us to navigate virtually within the key-frame graph by generating and rendering realistic novel views in real-time. This basic way to interact with the proposed world representation illustrates the richness and quality of the model and also suggests possible alternative applications. Given a photorealistic real-time visualization system, the next sections will show how this type of model can be used to perform real-time pose estimation and autonomous navigation of a vehicle.

To generate images, a virtual camera is controlled manually by the user (keyboard and mouse) to define the 6 DOF pose within the graph. The closest RGB-D sphere to the camera is then used to generate the view of the virtual camera using novel view synthesis. Figure 12 shows a virtual view that has been synthesized from a key-frame sphere. It can be noted that each sphere is valid for a local domain (in practice around  $5 \text{ m}^3$ ) until errors in the depth map begin to create visual distortion. It is therefore possible to generate an infinity of virtual images locally around the acquired nodes of the graph without having ever observed them before.

### 6.3.3. Asymmetric Real-time Localization

One of the main advantages of a spherical key-frame graph is that any type of online sensor can be used to estimate the relative pose with respect to this environment model. In particular, whether the online sensor is monocular, omnidirectional, spherical RGB-D, or other, it is possible to define a warping function that allows us to align each sensor data

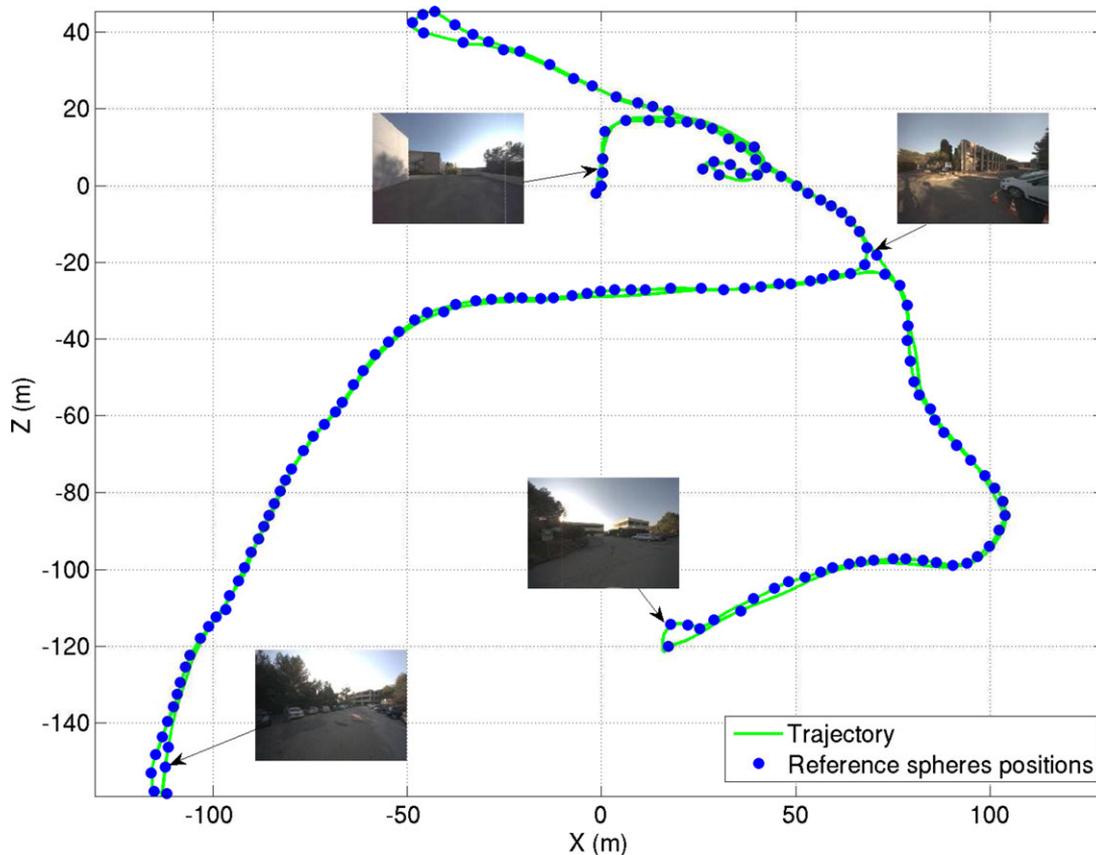


Figure 11. A spherical key-frame graph covering 1.5 km.

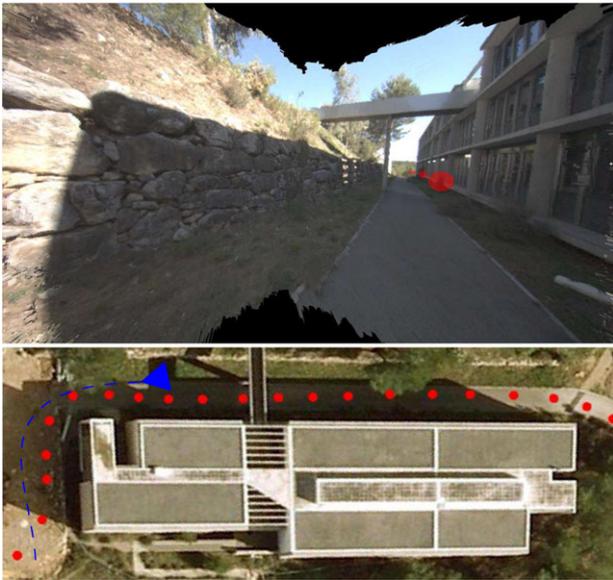
type with a spherical RGB-D graph. Here the term “asymmetric” is used to highlight this advantage. In practice, both monocular and stereo online camera configurations were tested in order to perform asymmetric 6 DOF localization in real-time with respect to a previously acquired spherical key-frame graph. The monocular camera is advantageous since it is low cost and only requires warping a single image (in exchange for faster computation but less robustness). The stereo camera pair, on the other hand, requires an extra image but is more robust to occlusions. Several real-time experiments were performed to validate the performance (robustness, precision, computational efficiency) of this asymmetric online localization algorithm.

The first experiment was conducted using a single key-frame graph in order to test the domain of validity of a single sphere. A monocular camera with a resolution of  $800 \times 600$  pixels was manually moved around the sphere, and online tracking was performed at 45 Hz. Figure 13 shows the trajectory of the camera with respect to the reference sphere. It can be seen that using a spherical image allows us to localize the camera in all viewing directions. In this practical setting (with given resolution and camera and scene config-

uration), high-quality localization can still be performed up to 3.5 m away.

The algorithm was then validated on a subset graph of 12 spheres extracted from the results presented in Section 6.3.1. These images were acquired from the same monocular camera configuration while mounted on an electric Cycab vehicle navigating locally within the graph.

Figure 14 shows the trajectory that was obtained by the localization algorithm. In the experiment, the starting point of the vehicle is at coordinates  $(X = -0.4, Z = -0.1)$  corresponding to Figure 14(a). The vehicle then advances in the positive  $Z$  direction until it reaches the point  $(X = -1.8, Z = 20.5)$ . The vehicle then reverses (green part) until it reaches the coordinates  $(X = 3.3, Z = 18.8)$ , at which point it then returns in the forward direction toward the starting point. It can be seen that the use of this representation allows us to instantaneously capture all viewing directions, which is particularly useful for locating a vehicle that may navigate in two or more directions along a road. What is more is that the depth component allows localization of the vehicle from trajectories not previously seen, therefore allowing vehicles to deviate from the learned path.



**Figure 12.** Top: Virtual image generated from an augmented spherical image. Bottom: An aerial view illustrating the global graph of key-frames.

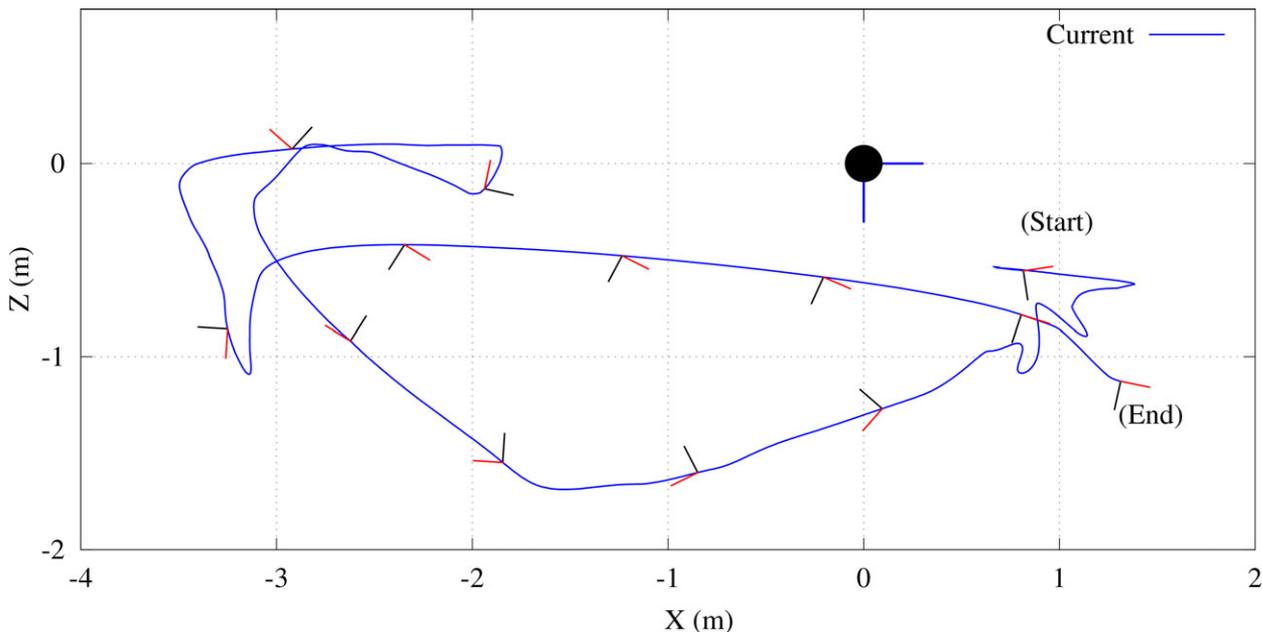
#### 6.3.4. Localization in the XIIth District of Paris

The real-time 6 DOF localization method was also validated on large-scale stereo sequences captured in uncontrolled ur-

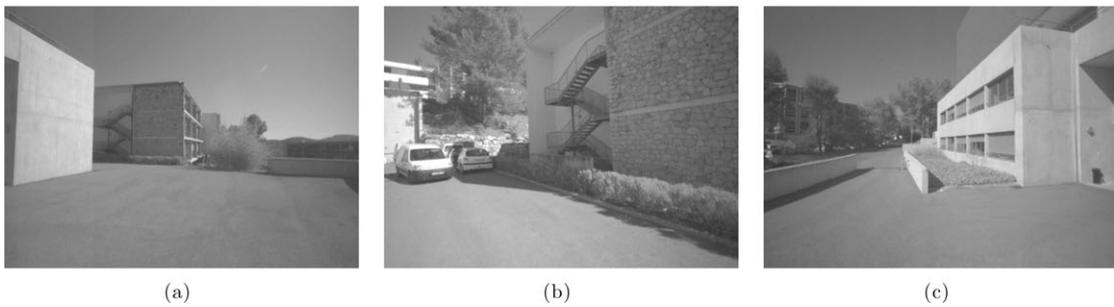
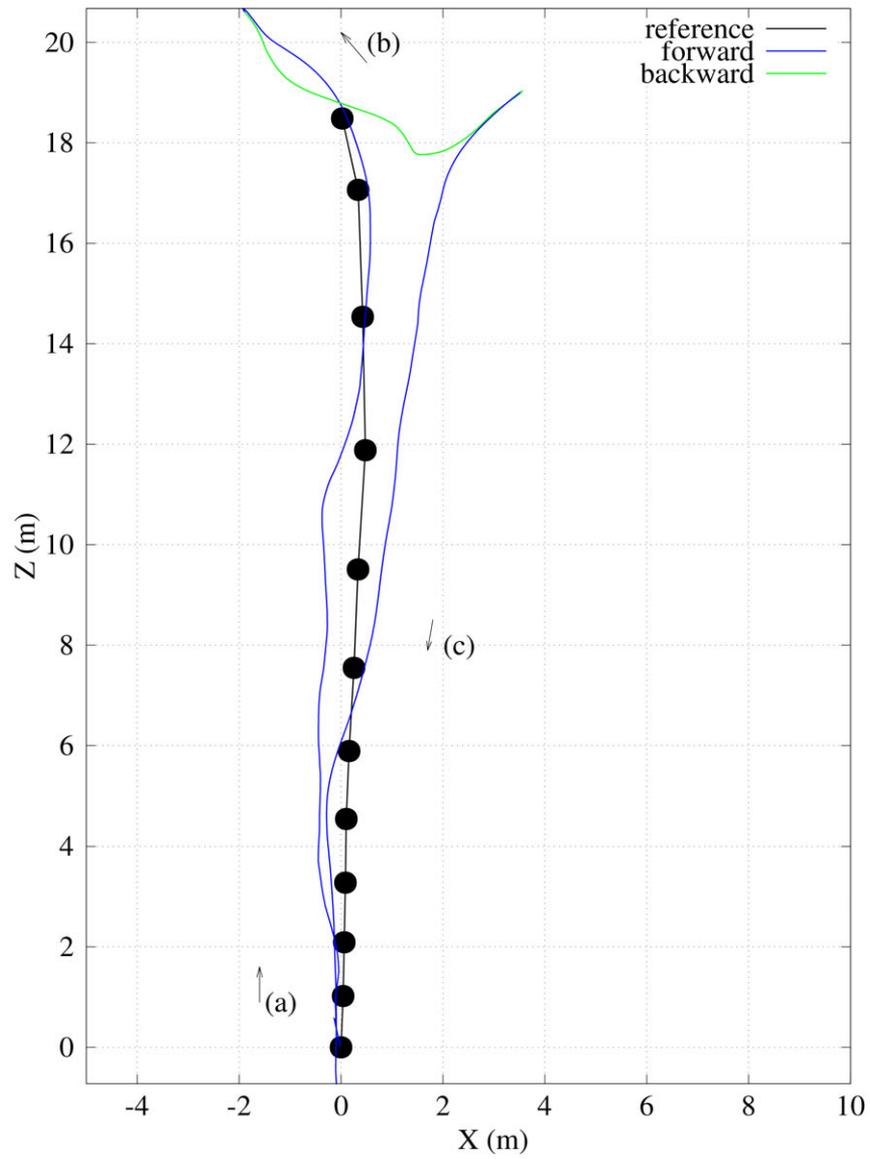
ban environments in the XIIth district of Paris for the purpose of the French national ANR-CityVIP project. While it can be argued that this type of environment is structured, the structure of this type of environment has not been explicitly exploited in the proposed method, which also works in natural settings. Furthermore, the sequences were acquired on a vehicle operating normally among the flow of traffic (i.e., 50 km/h). The cameras used have a resolution image of  $800 \times 600$  pixels, and acquisition is performed at 15 Hz. At this frequency, the displacement between successive images can be quite large and challenging, particularly when there are large rotations when the car takes a corner.

The topological key-frame graph was learned via a first pass covering about 1 km in length. This resulted in a graph of 441 reference images. Since only stereo key-frames are used, a larger number of reference images is generated in the corners than the spherical model requires. This ensures sufficient overlap between the nodes of the graph (cf. Figure 15).

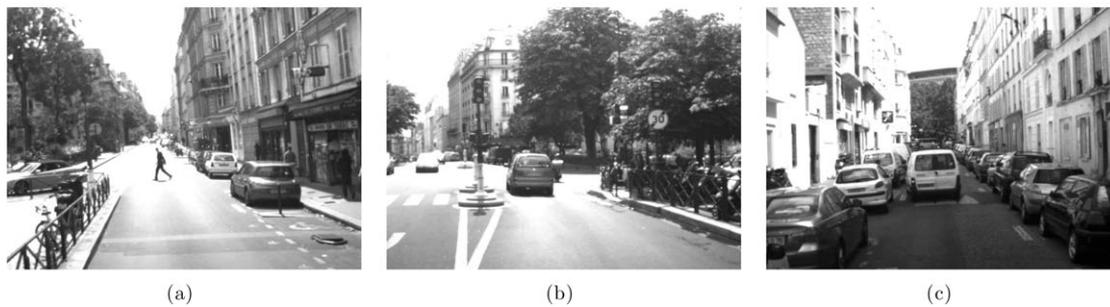
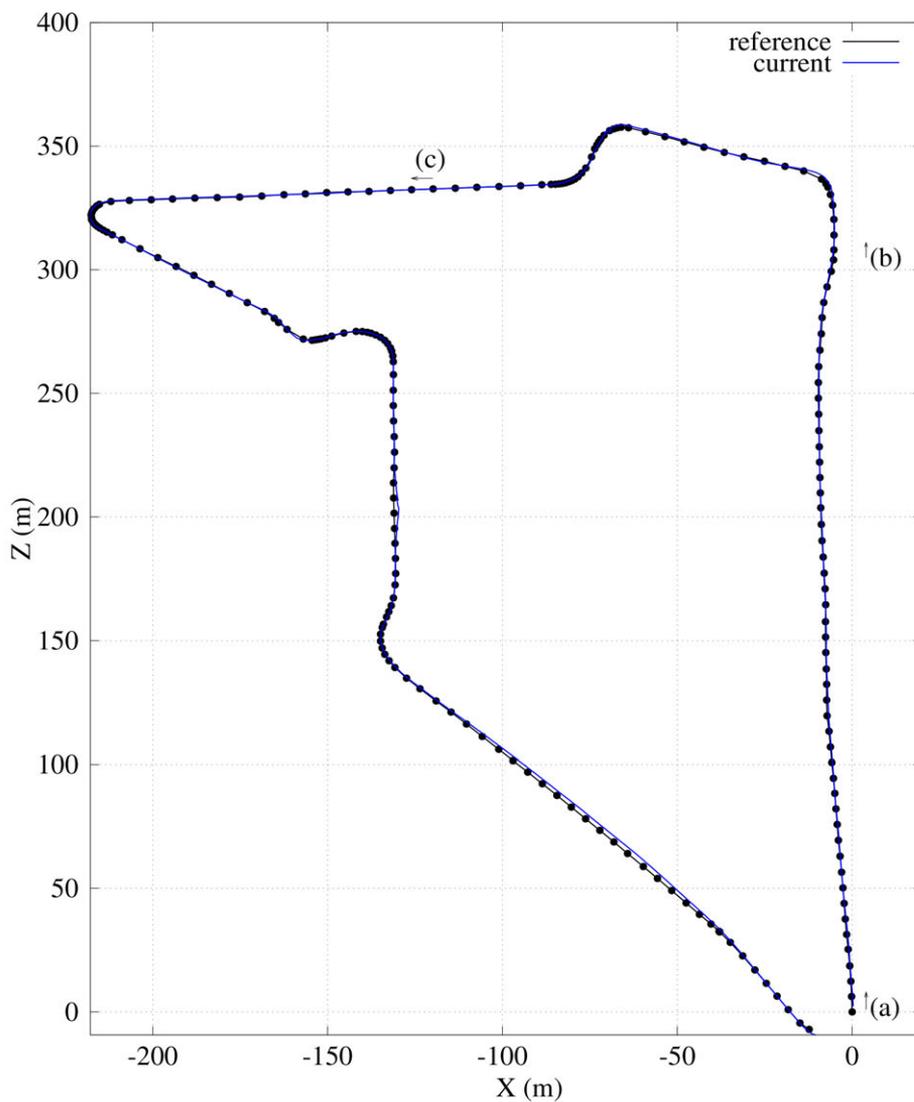
Live localization was performed from a second sequence of images that was recorded via a second pass that was significantly different from the first one due to the moving traffic, the different trajectory taken on the road, along with different illumination conditions. Figure 15 shows the obtained trajectory and the position of the reference images. Although the vehicle is following the same road, the path is locally different from the learned trajectory (the car overtakes, the turns are taken much more widely, etc.). As can be



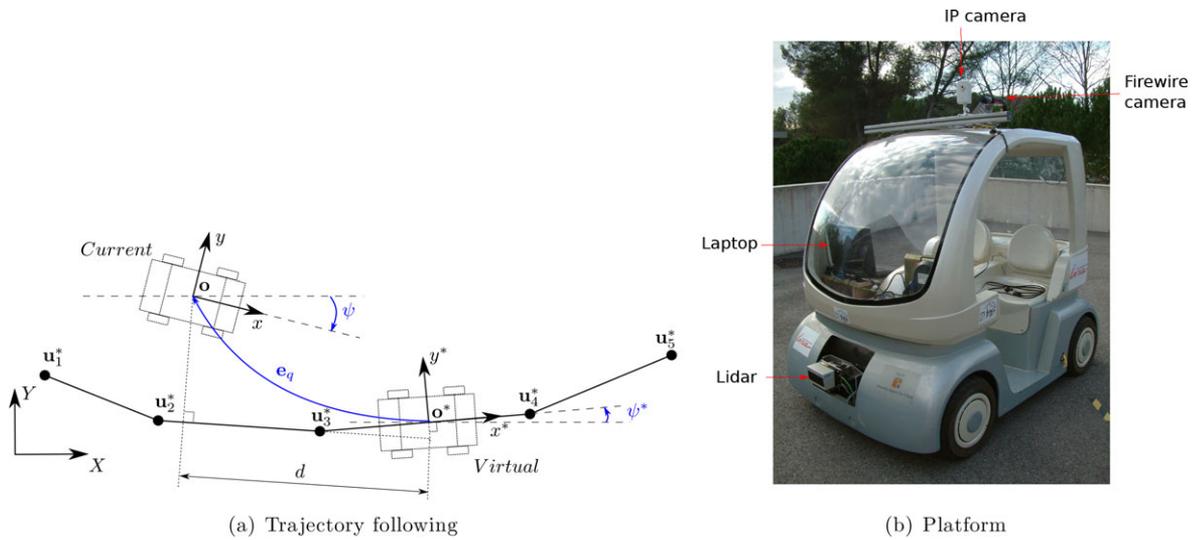
**Figure 13.** 6 DOF localization around a spherical image. In blue: the trajectory of the estimated camera. Some of the orientations are displayed in red and black; the optical axis is in red.



**Figure 14.** Localization in a graph of RGB-D spheres. In blue: the vehicle is moving forward. In green: the vehicle is reversing. Parts (a), (b), and (c) are images acquired during the live real-time tracking phase.



**Figure 15.** Localization from a key-frame graph in the XIIth district of Paris. In blue: the trajectory that was estimated by the real-time localization algorithm. In black, the position of the reference images (only one out of two reference images is shown). Parts (a), (b), and (c) are images that were acquired during the online localization phase.



**Figure 16.** (a) Regulated error for visual servoing. The current vehicle position is projected onto the closest reference trajectory’s edge. A reference position is selected at a distance  $d$ , to generate longitudinal and angular errors  $\{e_q, e_\psi\}$ . (b) Cycab vehicle with its sensors. The Firewire camera is placed on the top of the vehicle. The Lidar sensor is only used to detect obstacles.

seen in Figures 14(a), 14(b), and 14(c), the environment contains a lot of vegetation, illumination change, moving cars, and pedestrians that are not in the database. Thanks to the M-estimator outlier rejection and the hybrid minimization, the proposed localization algorithm remains very robust to large changes in the scene. Furthermore, the utilization of interframe rotation estimation, defined in Section 5.2, allows us to efficiently deal with large motion between the images (e.g., an interframe rotation of  $5^\circ$  generates a motion of 50 pixels in the image).

#### 6.4. CityVIP Autonomous Driving Challenge

This section reports the autonomous navigation results obtained during the final challenge of the French ANR CityVIP project. This project was conducted by several leading French research teams from June 2008 to December 2011, and its aim was to develop autonomous transportation vehicles for urban environments. The goal of the challenge was to autonomously follow a reference trajectory over large scales in a completely uncontrolled environment. The entry presented here was based on a learned environment map and trajectory acquired during a manual teaching phase. While the navigation path was mapped using a key-frame approach, the environment remained highly unstructured and included traffic (cars, bicycles, trams), pedestrians, etc. In a first part, the platform will be presented before introducing the control law that was used for trajectory following. In a second part, the preliminary trials and preparations are first presented before detailing the final results for the autonomous driving challenge.

##### 6.4.1. Platform

The platform used for the experiments is an electric Cycab vehicle [see Figure 16(b)], which can be controlled using longitudinal velocities and steering angle. A Lidar is placed in the front of the vehicle and is only used to detect imminent collisions. A single Firewire camera, placed on the top of the vehicle, is used for online localization. All the computations and control commands are performed on a Core i7 standard laptop, embedding the precomputed graph of augmented spheres. Additionally, an IP camera is used to broadcast a video to a ground station.

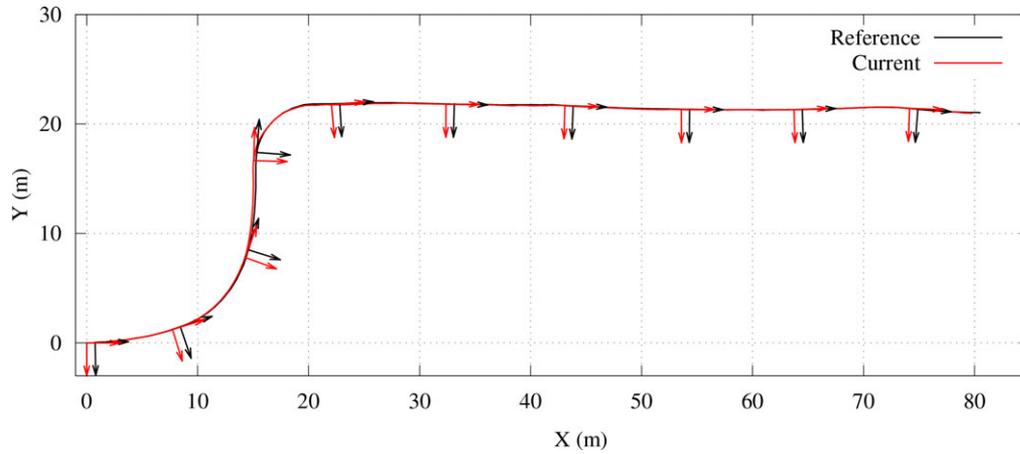
##### 6.4.2. Control Aspects

For autonomous driving, the aim is to follow automatically a reference trajectory  $\mathcal{U}$  generated locally around the learned graph. The trajectory  $\mathcal{U} = \{\mathbf{u}_1^*, \mathbf{u}_2^*, \dots, \mathbf{u}_n^*\}$  contains  $n$  input vectors such that

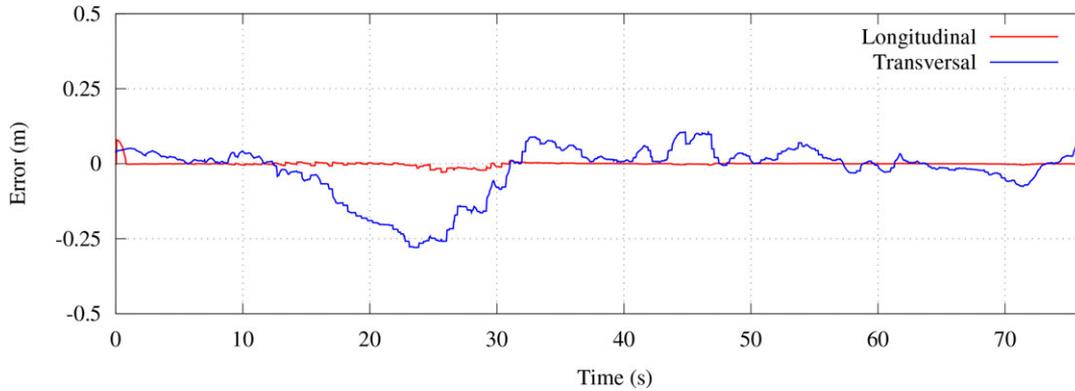
$$\mathbf{u}^* = \{x^*, y^*, \psi^*, U^*, \dot{\psi}^*\}, \quad (21)$$

where the point  $\mathbf{o}^* = \{x^*, y^*\}$  is a desired position,  $\psi^*$  is the yaw angle,  $U^*$  is the longitudinal velocity, and  $\dot{\psi}^*$  is the desired angular velocity.

The control problem can be formulated as detailed in Benhimane, Malis, Rives, and Azinheira (2005). In the proposed case, a *virtual* vehicle is followed and used to generate an error in translation and orientation that can be regulated using state feedback. Longitudinal velocity is controlled using a proportional feedback on the longitudinal error, and steering angle depends on yaw and transversal errors. These errors are obtained by projecting the current



**Figure 17.** Trajectory autonomously followed at INRIA Sophia Antipolis. A subset of reference and current poses is plotted. The offset between the current and reference poses is related to the distance  $d$  used to generate a longitudinal error. Corresponding longitudinal and transversal error can be found in Figure 18.



**Figure 18.** Longitudinal and transversal errors. The longitudinal error is centred around  $d$ .

vehicle position onto the closest reference trajectory’s edge [see Figure 16(a)]. The reference position is then selected by translating the projected position along the trajectory by a distance  $d$ . The translation error between the reference point and the current position is then defined by

$$\mathbf{e}_d = \begin{bmatrix} e_x \\ e_y \end{bmatrix} = \mathbf{R}_{\psi^*}^T (\mathbf{o} - \mathbf{o}^*) = \mathbf{R}_{\psi^*}^T \begin{bmatrix} x - x^* \\ y - y^* \end{bmatrix}, \quad (22)$$

where the rotation matrix of  $\psi^*$  can be written by

$$\mathbf{R}_{\psi^*} = \begin{bmatrix} \cos(\psi^*) & -\sin(\psi^*) \\ \sin(\psi^*) & \cos(\psi^*) \end{bmatrix}. \quad (23)$$

The angular error is directly defined as

$$e_\psi = \psi - \psi^*, \quad (24)$$

and the control law, derived from Benhimane et al. (2005), is

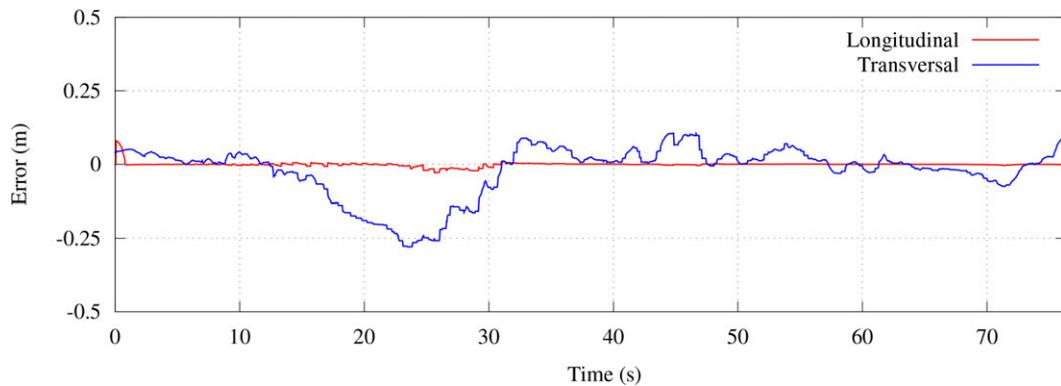
$$\begin{cases} U = U^* - k_x(|U^*| + \epsilon)e_x, \\ \dot{\psi} = \dot{\psi}^* - k_y|U^*|e_y - k_\psi|U^*| \tan(e_\psi), \end{cases} \quad (25)$$

where the gains  $k_x$ ,  $k_y$ ,  $k_\psi$ , and  $\epsilon$  are positive scalars.

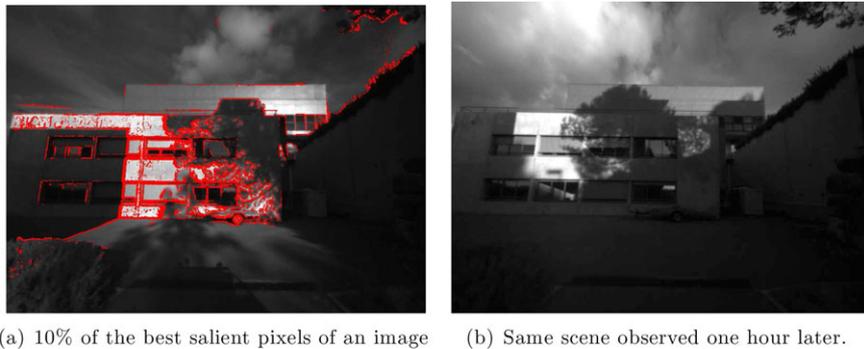
### 6.4.3. Challenge Preparation and Testing

Preliminary autonomous navigation results were conducted at INRIA Sophia Antipolis in order to test the entire autonomous navigation pipeline. The following result reports autonomous driving on a small trajectory of 100 m, which was manually generated around a prelearned key-frame graph. As can be seen in Figure 17, the trajectory contains two 90° turns. The reference vehicle longitudinal velocity was set to 1.4 m/s.

Figure 18 plots the longitudinal and transversal errors with respect to the time. It can be seen that the vehicle was



**Figure 19.** Autonomous navigation. Top: A 490 m trajectory followed autonomously; the reference trajectory is shown in black. The current trajectory is shown in red and some virtual/current vehicle poses are plotted. (a)–(f) Images captured during the navigation.



**Figure 20.** Shadows example. Image (b) was acquired 1 h after image (a). Strong shadows are selected as salient pixels.

able to follow the reference trajectory, keeping a longitudinal error close to zero, while the transversal error is less than 25 cm. Note that the peaks in transversal errors correspond to the 90° turns, which can be avoided with a better setting of the control gains of Eq. (25).

#### 6.4.4. Final CityVIP Challenge

The following section will present one of the many experiments conducted during the final CityVIP challenge, which lasted one whole week at Place Jaude in the city center of Clermont Ferrand, France. Over the course of one week, various trials were conducted in a large range of highly varying situations, allowing validation of the proposed approach. The intensive battery of tests included realistic, dynamic, and large-scale environments involving wide open spaces, narrow corridors, vegetation, trees, and “hostile” moving objects such as pedestrians, bicycles, and other vehicles.

A learning phase was performed along a 490 m trajectory by manually driving an electric Cycab vehicle equipped with the spherical acquisition system described in Section 4. To ensure an admissible path for the online navigation phase (i.e., without static obstacles), the trajectory obtained during the learning phase was used as input for the online navigation. Even so, it should be highlighted that the proposed localization method is capable of deviating from the learned path and can accurately estimate the camera pose within a local region around the graph, as was demonstrated in Section 6.3.3. The reference longitudinal velocity was set to 1.2 m/s for the whole sequence.

Figure 19 shows the desired trajectory in black, and the trajectory followed autonomously by the vehicle in red. The vehicle starts at position ( $X = 0, Y = 0$ ) and begins to move along the  $Y$  axis. The experiment finishes at position ( $X = -62, Y = 61$ ). The vehicle was able to follow autonomously the whole sequence using only the monocular camera for localization. As can be seen in Figures 18(a)–18(c), the accuracy of the localization method enables navigation in narrowed corridors, while the M-estimator ensures robustness to occlusions such as pedestrians. Figures 18(d) and 18(f)

show the vehicle navigating in much larger areas (open place). This kind of environment has displayed some limitations of vision-based-only navigation. Since geometric information is far from the camera (building façades), accurate estimations of translations are degraded (infinite points are invariant to translations). These effects can be seen on the red trajectory around the landmark [Figure 18(e)]. However, this lack of precision could be overcome using additional sensors, such as GPS and inertial measurements.

#### 6.4.5. Discussions on the Experiments

As mentioned in the previous section, visual navigation in large open-spaces is challenging since translations are not accurately estimated. This is particularly a problem when using wide-angle cameras since a small translation of the sensor will not generate any changes in the images. On the other hand, wide-angle cameras provide more robustness. It could be interesting to combine a wide-angle camera for robustness with a long focal camera for accuracy.

A second scenario has revealed one limitation of the visual localization approach. In Figure 19, 10% of the best salient pixels of the image are shown in red. We can see that the strong gradients generated by the shadows are selected by the saliency selection algorithm. In Figure 19, it can be seen that one hour later, the shadows are completely different. Even though the localization approach is extremely robust, the minimization can be perturbed by those false gradients. A simple solution could be to recompute a saliency selection online with respect to the outlier rejection. A second possibility could be to detect the shadows and use an invariant representation as in Corke, Paul, Churchill, and Newman (2013).

## 7. CONCLUSIONS AND FUTURE WORK

The approach described in this paper proposes a complete mapping, localization, guidance, and control framework for autonomous navigation of robots in large unstructured environments. The mapping method allows to reconstruct

dense visual maps of large-scale 3D environments, using a novel spherical key-frame and topological graph representation combining photometric color and depth information (RGB-D). It has been shown how to acquire this model using a learning approach and subsequently synthesizing photometrically accurate views locally around the learned graph. Reconstructed spheres acquired along a trajectory are used as input for a robust dense spherical tracking algorithm, which estimates the spheres' positions.

During online navigation, an efficient direct registration technique is employed to accurately localize a monocular camera navigating within the graph. The robustness of the localization method has been validated in challenging noncontrolled sequences containing a lot of dynamic objects, including traffic, pedestrians, lighting variation, and other outliers. The proposed spherical environment map is asymmetric in that it can be used not only for the localization of standard monocular cameras but also for stereo or omnidirectional systems. The proposed approach is, however, purely vision-based in order to demonstrate the potential of such a low-cost system, however the system could easily be fused with other types of sensors such as IMU or GPS.

Future work will be aimed at improving the database construction, by georeferencing the spherical graph in a GIS (Geographic Information System), which may contain higher-level information such as free space. This geolocation will allow us to use advanced path-planning algorithms. To further improve the autonomous navigation solution, it should be interesting to fuse vision-based localization with GPS signal and inertial measurements.

An aspect that has not been addressed in this work is the region of validity of a single sphere. As was shown in the experimental results, the proposed approach is able to perform online localization with respect to a reference sphere up to several meters. It would be interesting to quantify this region in order to improve it, for example by taking into account the changes in resolution in the warping functions. This could also allow a better compression of the graph of spheres.

Another important feature that will be studied is life-long mapping. Since the environment is not static, it is necessary to update the database with new geometric and photometric information gathered during the online phase. This could be achieved using an online SLAM technique.

## ACKNOWLEDGMENTS

This work has been supported by ANR (French National Agency) CityVIP project under Grant No. ANR-07\_TSFA-013-01. The authors would like to thank Alexandre Chapoulie for helpful discussions and for his work on loop closure detection.

## APPENDIX A

### A1 Warping Functions

#### Perspective projection

A 3D point  $\mathbf{v} \in \mathbb{R}^3$  is projected onto the normalized image plane such that

$$\bar{\mathbf{p}} = \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \frac{\mathbf{M}\bar{\mathbf{v}}}{\mathbf{e}_3^T \mathbf{M}\bar{\mathbf{v}}}, \quad (\text{A.1})$$

where  $\bar{\mathbf{v}} = (x, y, z, 1)$  is the homogeneous version of  $\mathbf{v}$ ,  $\mathbf{e}_3 = [0 \ 0 \ 1]^T$  is a unit vector allowing to normalize the pixel coordinates with respect to their third component, and  $\mathbf{M}$  is the  $3 \times 4$  perspective projection matrix defined as

$$\mathbf{M} = \mathbf{K} [\mathbf{R} \ \mathbf{t}], \quad (\text{A.2})$$

with  $\mathbf{K}$  the  $3 \times 3$  intrinsic matrix of the camera and  $\mathbf{T} = (\mathbf{R}, \mathbf{t}) \in \mathbb{SE}(3)$  a rigid transformation.

#### Spherical projection

A 3D point  $\mathbf{v} \in \mathbb{R}^3$  can be projected onto the unit sphere by

$$\mathbf{q}_E = \begin{bmatrix} x_S \\ y_S \\ z_S \end{bmatrix} = \frac{\mathbf{R}\mathbf{v} + \mathbf{t}}{\|\mathbf{R}\mathbf{v} + \mathbf{t}\|}, \quad (\text{A.3})$$

where  $\mathbf{T} = (\mathbf{R}, \mathbf{t}) \in \mathbb{SE}(3)$  is a rigid transformation. The point  $\mathbf{q}_E$  can therefore be converted to spherical coordinates by

$$\mathbf{q}_S = \begin{bmatrix} \theta \\ \phi \\ \rho \end{bmatrix} = \begin{bmatrix} \arctan(z_S/x_S) \\ \arctan\left(y_S/\sqrt{x_S^2 + z_S^2}\right) c \\ \sqrt{z_S^2 + y_S^2 + x_S^2} \end{bmatrix}. \quad (\text{A.4})$$

### A2 Jacobians

The Jacobian matrix of error (11) can be decomposed in three Jacobian matrices:

$$\mathbf{J}(\bar{\mathbf{x}}) = \mathbf{J}_I \mathbf{J}_w \mathbf{J}_T, \quad (\text{A.5})$$

where  $\mathbf{J}_T = \frac{\partial \mathbf{T}(\mathbf{x})}{\partial \mathbf{x}}$  is the derivative of Eq. (1) with respect to  $\mathbf{x}$  of dimensions  $12 \times 6$ ,  $\mathbf{J}_w = \frac{\partial \mathbf{w}(\cdot)}{\partial \mathbf{T}}$  is the derivative of the spherical warping with respect to  $\mathbf{T}$  of dimensions  $3 \text{ nm} \times 12$ , and  $\mathbf{J}_I = \frac{\partial \mathbf{I}}{\partial \mathbf{q}_S}$  is the spherical image gradient of dimensions  $\text{nm} \times 3 \text{ nm}$ .

For the multisphere localization of Eq. (19), the global Jacobian matrix is obtained by stacking the Jacobian of each error, leading to

$$\mathbf{J}_G = \begin{bmatrix} \mathbf{J}_0 \\ \mathbf{J}_1 \mathbf{J}_v(\mathbf{T}_G^1) \end{bmatrix}, \quad (\text{A.6})$$

where  $J_v(\mathbf{T}_G^1)$  is the adjoint map associated with the pose  $\mathbf{T}_G^1$ , and it is defined by

$$\mathbf{J}_v(\mathbf{T}) = \begin{bmatrix} \mathbf{R} & \mathbf{t} \times \mathbf{R} \\ \mathbf{0}_3 & \mathbf{R} \end{bmatrix}. \quad (\text{A.7})$$

### A3 Robust Estimation

The robust diagonal matrix  $\mathbf{D}$  is defined such that

$$\mathbf{D} = \begin{bmatrix} w_1 & 0 & \cdots & 0 \\ 0 & w_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & w_n \end{bmatrix}, \quad (\text{A.8})$$

where the weights  $w_i \in [0; 1]$  are obtained by

$$w_i = \frac{\psi(\delta_i/\sigma)}{\delta_i/\sigma}, \quad \psi(u) = \begin{cases} u & \text{if } |u| \leq a, \\ a \frac{u}{|u|} & \text{if } |u| > a, \end{cases} \quad (\text{A.9})$$

$\delta_i$  is the centered residual of each component of the error vector  $\mathbf{e} = [e_1, e_2, \dots, e_{nm}]^\top$ , with  $\delta_i = e_i - \text{median}(\mathbf{e})$ , and  $\psi$  is the Huber influence function. The proportionality factor for the Huber function is  $a = 1.345$ , which represents 95% efficiency under a Gaussian noise.

The values  $\delta_i$  are normalized by a robust measure of the scale of the distribution (MAD) such that

$$\sigma = \frac{1}{\Phi^{-1}(0.75)} \text{median}[|\delta_i - \text{median}(\delta)|], \quad (\text{A.10})$$

where  $\Phi(\cdot)$  is the cumulative distribution function and  $1/\Phi^{-1}(0.75) = 1.48$  is the standard deviation for a normal distribution.

### A4 Multimedia File

The video attached to this paper illustrates each aspect of the proposed dense mapping and autonomous navigation approach as follows:

- (08–14 s) Spherical acquisition system mounted onto an electric Cycab vehicle.
- (14 s–1 m) Incremental sphere mapping process. Left, current RGB-D sphere. Top right, estimated trajectory and reference sphere positions. Bottom right, new sphere selection criterion.
- (1–2 m) Photorealistic virtual navigation (see Section 6.3.2).
- (2 m–3 m 18 s) Online localization results in the XIIth district of Paris (see Section 6.3.4). Top right, current monocular image. Top middle, closest reference key-frame. Bottom right, virtual view of the model rendered at the current camera pose. Bottom left, current camera pose and key-frames poses.
- (3 m 18 s–4 m 53s) Autonomous navigation results in Clermont Ferrand (see Section 6.4.4).

## REFERENCES

- Anguelov, D., Dulong, C., Filip, D., Frueh, C., Lafon, S., Lyon, R., Ogale, A., Vincent, L., & Weaver, J. (2010). Google street view: Capturing the world at street level. *Computer*, 43.
- Arican, Z., & Frossard, P. (2007). Dense disparity estimation from omnidirectional images. In *Proceedings of the IEEE Conference on Advanced Video and Signal Based Surveillance (AVSS)*, London.
- Audras, C., Comport, A. I., Meilland, M., & Rives, P. (2011). Real-time appearance-based slam for rgb-d sensors. In *Proceedings of the Australian Conference on Robotics and Automation (ACRA)*, Melbourne.
- Avidan, S., & Shashua, A. (1997). Novel view synthesis in tensor space. In *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, San Juan, Puerto Rico.
- Baker, P., Fermuller, C., Aloimonos, Y., & Pless, R. (2001). A spherical eye from multiple cameras. In *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, Kauai, HI.
- Baker, S., & Matthews, I. (2001). Equivalence and efficiency of image alignment algorithms. In *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, Kauai, HI.
- Benhimane, S., Ladikos, A., Lepetit, V., & Navab, N. (2007). Linear and quadratic subsets for template-based tracking. In *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, Minneapolis, MN.
- Benhimane, S., & Malis, E. (2004). Real-time image-based tracking of planes using efficient second-order minimization. In *Proceedings of the IEEE International Conference on Intelligent Robots and Systems (IROS)*, Sendai, Japan.
- Benhimane, S., Malis, E., Rives, P., & Azinheira, J. (2005). Vision-based control for car platooning using homography decomposition. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, Barcelona, Spain.
- Brown, D. C. (1971). Close-range camera calibration. *Photogrammetric Engineering*, 37(8), 855–866.
- Burt, P. J., & Adelson, E. H. (1983). A multiresolution spline with application to image mosaics. *ACM Transactions on Graphics*, 2, 217–236.
- Cappelle, C., El Najjar, M., Charpillat, F., & Pomorski, D. (2011). Virtual 3d city model for navigation in urban areas. *Journal of Intelligent and Robotic Systems*, 66(3), 1–23.
- Caron, G., Dame, A., & Marchand, E. (2014). Direct model-based visual tracking and pose estimation using mutual information. *Image and Vision Computing*, 32(1), 54–63.
- Caron, G., Marchand, E., & Mouaddib, E. (2011). Tracking planes in omnidirectional stereovision. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, Shanghai, China.
- Chapoutie, A., Rives, P., & Filliat, D. (2011). A spherical representation for efficient visual loop closing. In *Proceedings of the 11th Workshop on Omnidirectional Vision*,

- Camera Networks and Non-classical Cameras (OMNIVIS), Barcelona, Spain.
- Cobzas, D., Zhang, H., & Jagersand, M. (2003). Image-based localization with depth-enhanced image map. In Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), Taipei, Taiwan.
- Comport, A. I., Malis, E., & Rives, P. (2007). Accurate quadrifocal tracking for robust 3d visual odometry. In Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), Rome, Italy.
- Comport, A. I., Malis, E., & Rives, P. (2010). Real-time quadrifocal visual odometry. *The International Journal of Robotics Research*, 29(2-3), 245–266.
- Comport, A. I., Marchand, E., Pressigout, M., & Chaumette, F. (2006). Real-time markerless tracking for augmented reality: The virtual visual servoing framework. *IEEE Transactions on Visualization and Computer Graphics*, 12(4), 615–628.
- Corke, P., Paul, R., Churchill, W., & Newman, P. (2013). Dealing with shadows: Capturing intrinsic scene appearance for image-based outdoor localisation. In Proceedings of the IEEE International Conference on Intelligent Robots and Systems (IROS), Tokyo, Japan.
- Courbon, J., Mezouar, Y., & Martinet, P. (2008). Indoor navigation of a non-holonomic mobile robot using a visual memory. *Autonomous Robots*, 25, 253–266.
- Courbon, J., Mezouar, Y., & Martinet, P. (2009). Autonomous navigation of vehicles from a visual memory using a generic camera model. *Intelligent Transport System*, 10, 392–402.
- Craciun, D., Paparoditis, N., & Schmitt, F. (2010). Multi-view scans alignment for 3d spherical mosaicing in large-scale unstructured environments. *Computer Vision and Image Understanding*, 114(11), 1248–1263. Special issue on Embedded Vision.
- Cummins, M., & Newman, P. (2008). FAB-MAP: Probabilistic Localization and Mapping in the Space of Appearance. *The International Journal of Robotics Research*, 27(6), 647–665.
- Dame, A., & Marchand, E. (2010). Accurate real-time tracking using mutual information. In Proceedings of the IEEE/ACM ACM International Symposium on Mixed and Augmented Reality (ISMAR), Seoul, Korea.
- Davison, A., & Murray, D. (2002). Simultaneous localization and map-building using active vision. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(7), 865–880.
- Debevec, P. E., Taylor, C. J., & Malik, J. (1996). Modeling and rendering architecture from photographs: A hybrid geometry- and image-based approach. In Proceedings of the Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH), New Orleans, LA.
- Dellaert, F., & Collins, R. (1999). Fast image-based tracking by selective pixel integration. In Proceedings of the International Conference on Computer Vision Workshop on Frame-Rate Vision (ICCV-W), Kerkyra, Corfu, Greece.
- Drummond, T., Society, I. C., & Cipolla, R. (2002). Real-time visual tracking of complex structures. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24, 932–946.
- Durrant-Whyte, H., & Bailey, T. (2006). Simultaneous localisation and mapping (slam): Part i, the essential algorithms. *IEEE Robotics and Automation Magazine*, 13(2), 99–110.
- Earthmine (2009). Earthmine spherical sensor. <http://www.earthmine.com>.
- Furukawa, Y., & Ponce, J. (2010). Accurate, dense, & robust multiview stereopsis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32, 1362–1376.
- Gallegos, G., Meilland, M., Rives, P., & Comport, A. I. (2010). Appearance-based slam relying on a hybrid laser/omnidirectional sensor. In Proceedings of the IEEE International Conference on Intelligent Robots and Systems (IROS), Taipei, Taiwan.
- Geiger, A., Roser, M., & Urtasun, R. (2010). Efficient large-scale stereo matching. In Proceedings of the Asian Conference on Computer Vision (ACCV), Daejeon, Korea.
- Goncalves, N., & Araujo, H. (2004). Projection model, 3d reconstruction and rigid motion estimation from non-central catadioptric images. In Proceedings of the International Symposium on 3D Data Processing, Visualization and Transmission (3DPVT), Thessaloniki, Greece.
- Gorski, K. M., Hivon, E., Banday, A. J., Wandelt, B. D., Hansen, F. K., Reinecke, M., & Bartelman, M. (2005). Healpix—A framework for high resolution discretization, & fast analysis of data distributed on the sphere. *The Astrophysical Journal*, 622, 759–773.
- Gortler, S. J., Grzeszczuk, R., Szeliski, R., & Cohen, M. F. (1996). The lumigraph. In Proceedings of the Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH), New Orleans, LA.
- Grisetti, G., Grzonka, S., Stachniss, C., Pfaff, P., & Burgard, W. (2007). Efficient estimation of accurate maximum likelihood maps in 3d. In Proceedings of the IEEE International Conference on Intelligent Robots and Systems (IROS), Rome, Italy.
- Grisetti, G., Stachniss, C., Grzonka, S., & Burgard, W. (2009). Toro—Tree-based network optimizer. <http://openslam.org/toro.html>.
- Guizzo, E. (2011). Google car. <http://spectrum.ieee.org/automaton/robotics/artificial-intelligence/how-google-self-driving-car-works>.
- Hager, G., & Belhumeur, P. (1998). Efficient region tracking with parametric models of geometry and illumination. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(10), 1025–1039.
- Hammoudi, K., Dornaika, F., Soheilian, B., & Paparoditis, N. (2010). Generating raw polygons of street facades from a 2d urban map and terrestrial laser range data. In Proceedings of the SSSI Australasian Remote Sensing and Photogrammetry Conference (ARSPC), Alice Springs, NT, Australia.
- Handa, A., Newcombe, R. A., Angeli, A., & Davison, A. J. (2012). Real-time camera tracking: When is high frame-rate best? In Proceedings of the European Conference on Computer Vision (ECCV), Florence, Italy.

- Harris, C., & Stephens, M. (1988). A combined corner and edge detector. In *Proceedings of the 4th Alvey Vision Conference*.
- He, L., Luo, C., Geng, Y., Zhu, F., & Hao, Y. (2007). Reliable depth map regeneration via a novel omnidirectional stereo sensor. In *Proceedings of the 3rd International Conference on Advances in Visual Computing—Volume Part I*.
- Henry, P., Krainin, M., Herbst, E., Ren, X., & Fox, D. (2010). Rgb-d mapping: Using depth cameras for dense 3d modeling of indoor environments. In *Proceedings of the International Symposium on Experimental Robotics (ISER)*, New Delhi and Agra, India.
- Hirschmuller, H. (2008). Stereo processing by semiglobal matching and mutual information. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30, 328–341.
- Howard, A. (2008). Real-time stereo visual odometry for autonomous ground vehicles. In *Proceedings of the IEEE International Conference on Intelligent Robots and Systems (IROS)*, Nice, France.
- Huber, P. (1981). *Robust statistics*. New York: Wiley.
- Irani, M., & Anandan, P. (1998). Robust multi-sensor image alignment. In *International Conference on Computer Vision*, Bombay, India.
- Irani, M., & Anandan, P. (2000). About direct methods. *Vision Algorithms: Theory and Practice*, 1883, 267–277.
- Irschara, A., Zach, C., Frahm, J.-M., & Bischof, H. (2009). From structure-from-motion point clouds to fast location recognition. In *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, Ft. Collins, CO.
- Jogan, M., & Leonardis, A. (2000). Robust localization using panoramic view-based recognition. In *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, Hilton Head, SC.
- Kitt, B., Geiger, A., & Lategahn, H. (2010). Visual odometry based on stereo image sequences with ransac-based outlier rejection scheme. In *Proceedings of the IEEE Intelligent Vehicles Symposium (IV)*, La Jolla, CA.
- Klein, G., & Murray, D. (2007). Parallel tracking and mapping for small ar workspaces. In *Proceedings of the IEEE/ACM International Symposium on Mixed and Augmented Reality (ISMAR)*, Nara, Japan.
- Klingner, B., Martin, D., & Roseborough, J. (2013). Street view motion-from-structure-from-motion. In *Proceedings of the International Conference on Computer Vision (ICCV)*, Sydney, Australia.
- Kolmogorov, V., & Zabih, R. (2001). Computing visual correspondence with occlusions using graph cuts. In *Proceedings of the International Conference on Computer Vision (ICCV)*, Vancouver, Canada.
- Konolige, K., & Agrawal, M. (2008). Frameslam: From bundle adjustment to realtime visual mapping. *IEEE Transactions on Robotics*, 24(5), 1066–1077.
- Kutulakos, K. N., & Seitz, S. M. (2000). A theory of shape by space carving. *International Journal of Computer Vision*, 38, 307–314.
- Lafarge, F., & Mallet, C. (2011). Building large urban environments from unstructured point data. In *Proceedings of the International Conference on Computer Vision (ICCV)*, Barcelona, Spain.
- Levoy, M., & Hanrahan, P. (1996). Light field rendering. In *Proceedings of the Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH)*, New Orleans, LA.
- Lothe, P., Bourgeois, S., Dekeyser, F., Royer, E., & Dhome, M. (2010). Monocular slam reconstructions and 3d city models: Towards a deep consistency. *Computer Vision, Imaging and Computer Graphics. Theory and Applications*, 68, 201–214.
- Lovegrove, S., & Davison, A. (2010). Real-time spherical mosaicing using whole image alignment. In *Proceedings of the European Conference on Computer Vision (ECCV)*, Heraklion, Crete, Greece.
- Lowe, D. (2004). Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2), 91–110.
- Lowe, D. G. (1991). Fitting parameterized three-dimensional models to images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13, 441–450.
- Lucas, B. D., & Kanade, T. (1981). An iterative image registration technique with an application to stereo vision. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*, Vancouver, Canada.
- Lui, W. L. D., & Jarvis, R. (2010). Eye-full tower: A gpu-based variable multibaseline omnidirectional stereovision system with automatic baseline selection for outdoor mobile robot navigation. *Robotics and Autonomous Systems*, 58(6), 747–761.
- Malis, E. (2004). Improving vision-based control using efficient second-order minimization techniques. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, New Orleans, LA.
- Marchand, E., Bouthemy, P., & Chaumette, F. (2001). A 2d-3d model-based approach to real-time visual tracking. *Image and Vision Computing*, 19(13), 941–955.
- Mei, C., Benhimane, S., Malis, E., & Rives, P. (2006). Constrained multiple planar template tracking for central catadioptric cameras. In *Proceedings of the British Machine Vision Conference (BMVC)*, Edinburgh, UK.
- Mei, C., Sibley, G., Cummins, M., Newman, P., & Reid, I. (2010). Rslam: A system for large-scale mapping in constant-time using stereo. *International Journal of Computer Vision*, 94, 198–214. Special issue of BMVC.
- Meilland, M., Comport, A. I., & Rives, P. (2010). A spherical robot-centered representation for urban navigation. In *Proceedings of the IEEE International Conference on Intelligent Robots and Systems (IROS)*, Taipei, Taiwan.
- Meilland, M., Comport, A. I., & Rives, P. (2011a). Dense visual mapping of large scale environments for real-time localisation. In *Proceedings of the IEEE International Conference on Intelligent Robots and Systems (IROS)*, San Francisco.

- Meilland, M., Comport, A. I., & Rives, P. (2011b). Real-time dense visual tracking under large lighting variations. In Proceedings of the British Machine Vision Conference (BMVC), Dundee, UK.
- Meilland, M., Rives, P., & Comport, A. I. (2012). Dense rgb-d mapping for real-time localisation and navigation. In Proceedings of the IEEE Intelligent Vehicles Workshop on Navigation, Positioning and Mapping (IV-W), Alcalá de Henares, Spain.
- Micusik, B., & Pajdla, T. (2004). Autocalibration 3d reconstruction with non-central catadioptric cameras. In Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition (CVPR), Washington, DC.
- Montemerlo, M., Thrun, S., Koller, D., & Wegbreit, B. (2002). FastSLAM: A factored solution to the simultaneous localization and mapping problem. In Proceedings of the National Conference on Artificial Intelligence (AAAI), Edmonton, Canada.
- Mouragnon, E., Lhuillier, M., Dhome, M., Dekeyser, F., & Sayd, P. (2006). Real time localization and 3d reconstruction. In Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition (CVPR), New York.
- Nayar, S. (1997). Catadioptric omnidirectional camera. In Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition (CVPR), San Juan, Puerto Rico.
- Newcombe, R. A., Izadi, S., Hilliges, O., Molyneaux, D., Kim, D., Davison, A. J., Kohli, P., Shotton, J., Hodges, S., & Fitzgibbon, A. (2011a). Kinectfusion: Real-time dense surface mapping and tracking. In Proceedings of the IEEE/ACM ACM International Symposium on Mixed and Augmented Reality (ISMAR), Basel, Switzerland.
- Newcombe, R. A., Lovegrove, S., & Davison, A. J. (2011b). Dtm: Dense tracking and mapping in real-time. In Proceedings of the International Conference on Computer Vision (ICCV), Barcelona, Spain.
- Nistér, D., Naroditsky, O., & Bergen, J. (2004). Visual odometry. In Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition (CVPR), Washington, DC.
- Ogale, A. S., & Aloimonos, Y. (2005). Shape and the stereo correspondence problem. *International Journal on Computer Vision*, 65(3), 147–162.
- Olson, E., Leonard, J., & Teller, S. (2006). Fast iterative alignment of pose graphs with poor initial estimates. In Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), Orlando, FL.
- Ragot, N., Rossi, R., Savatier, X., Ertaud, J. Y., & Mazari, B. (2008). 3d volumetric reconstruction with a catadioptric stereovision sensor. In Proceedings of the IEEE International Symposium on Industrial Electronics (ISIE), Cambridge, UK.
- Royer, E., Lhuillier, M., Dhome, M., & Chateau, T. (2005). Localization in urban environments: Monocular vision compared to a differential gps sensor. In Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition (CVPR), San Diego, CA.
- Shan, Q., Adams, R., Curless, B., Furukawa, Y., & Seitz, S. M. (2013). The visual Turing test for scene reconstruction. In Proceedings of the International Conference on 3D Vision (3DIMPV), Seattle.
- Shi, J., & Tomasi, C. (1994). Good features to track. In Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition (CVPR), Seattle.
- Sibley, G., Matthies, L., & Sukhatme, G. (2008). A sliding window filter for incremental SLAM, *Lecture Notes in Electrical Engineering* (vol. 8). Springer.
- Silveira, G., Malis, E., & Rives, P. (2008). An efficient direct approach to visual SLAM. *IEEE Transactions on Robotics*, 24(5), 969–979.
- Spinello, L., & Arras, K. O. (2011). People detection in rgb-d data. In Proceedings of the IEEE International Conference on Intelligent Robots and Systems (IROS), San Fransisco.
- Szeliski, R. (2006). Image alignment and stitching: A tutorial. *Foundations and Trends in Computer Graphics and Vision*, 2(1), 1–104.
- Tardif, J.-P., George, M., Laverne, M., Kelly, A., & Stentz, A. (2010). A new approach to vision-aided inertial navigation. In Proceedings of the IEEE International Conference on Intelligent Robots and Systems (IROS), Taipei, Taiwan.
- Thrun, S. (2002). Probabilistic robotics. *ACM Communications*, 45, 52–57.
- Tola, E., Lepetit, V., & Fua, P. (2010). Daisy: An efficient dense descriptor applied to wide baseline stereo. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(5), 815–830.
- Triggs, B., Mclauchlan, P. F., Hartley, R. I., & Fitzgibbon, A. W. (1999). Bundle adjustment – a modern synthesis. In Proceedings of the International Workshop on Vision Algorithms: Theory and Practice (ICCV-W). Kerkyra, Corfu, Greece.
- Vacchetti, L., Lepetit, V., & Fua, P. (2004). Stable real-time 3D tracking using online and offline information. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(10), 1385–1391.
- Viola, P., & Wells, W. M. I. (1995). Alignment by maximization of mutual information. In Proceedings of the International Conference on Computer Vision (ICCV), Boston.