Adapting a Real-Time Monocular Visual SLAM from Conventional to Omnidirectional Cameras

Daniel Gutierrez, Alejandro Rituerto, J.M.M. Montiel, J.J. Guerrero
Departamento de Informática e Ingeniería de Sistemas(DIIS)-Instituto de Investigación en Ingeniería de Aragón(I3A)
Universidad de Zaragoza, Spain
dangu87@gmail.com, {arituerto, josemari, jguerrer}@unizar.es

Abstract

The SLAM (Simultaneous Localization and Mapping) problem is one of the essential challenges for the current robotics. Our main objective in this work is to develop a real-time visual SLAM system using monocular omnidirectional vision. Our approach is based on the Extended Kalman Filter (EKF). We use the Spherical Camera Model to obtain geometric information from the images. This model is integrated in the EKF-based SLAM through the linearization of the direct and the inverse projections. We introduce a new computation of the descriptor patch for catadioptric omnidirectional cameras which aims to reach rotation and scale invariance. We perform experiments with omnidirectional images comparing this new approach with the conventional one. The experimentation confirms that our approach works better with omnidirectional cameras since features last longer and constructed maps are bigger.

1. Introduction

The SLAM [25] problem tries to build a map of the surrounding and localize an autonomous robot relative to this map using only partial measurements of the environment. SLAM is usually formulated in a probabilistic way, *i.e*. the estimate of the robot position and map are computed as a probability distribution. Two main approaches are used for the computation of the probability distribution: the extended Kalman filter (EKF) [25] and the particle filter [2].

In Visual SLAM applications, image projections of relevant points known as local features are used as measurements. To extract and store the features on the image an extractor and descriptor are used. The feature extractor process the image and detects the key-points on it. The image processing is a high time-consuming step, which is critical for a real time application like SLAM. Rosten *et al*. [20] developed the feature extraction algorithm FAST (Features Accelerated Segment Test). They benchmark their FAST extractor with other widely used feature extractors showing that FAST outperforms them in computational cost and in repeatability when viewing the scene from different positions. The descriptor provides an identifier to an extracted point so that it can be recognised in future measurements. The most basic descriptor is a patch of a certain size centered in the key-point, although there exists more kinds of descriptors like SIFT [13], SURF [4], LBP [12], etc.

Since the seminal work of Davison [9], monocular SLAM has been a fertile research field. In this work we propose to combine state of the art robust EKF SLAM [7] with an omnidirectional sensor. Visual SLAM using omnidirectional cameras has been proposed in [8], [15], [24] and [22].

Due to the 360º FOV of omnidirectional cameras, features last longer on the image than in the case of conventional cameras, specially in big camera rotations. The increased lifespan of the features on the image translates in a better estimation of the position of the features on the map, a lower need to initialise new features and a increased robustness.

However the omnidirectional images involve a more complex projection model, important image deformation, distortion and variable scale in the image. So, the feature descriptor should be modified for catadioptric cameras. In this way, Svoboda and Padjla [23] propose the use of patches with variable size and shape (active windows). Their experiments show that active windows provide best matching results than square windows. Ieng *et al*. [5] propose the computation of patches of different angular apertures for the same feature to overcome the matching problems derived from the varying resolution of the camera. Scaramuzza *et al*. [21] take advantage of the projection of vertical lines of the world as radial lines on the image. They propose a method to extract and match vertical lines with rotation invariant descriptors and apply this method to an EKF-SLAM. In [1] Andreasson *et al*. propose a modified SIFT feature with no scale invariance. To obtain rotation invariance they rotate each patch to the same global orientation. Lu and Zheng [14] combine the rotation invariant patch by Andreasson with a FAST extractor and a CS-LBP descriptor and they compare it with the SIFT algorithm.

Besides that, omnidirectional images require a more complex projection model to obtain geometric information from them. One of the most used is the Spherical Camera Model [10], [3], which has been integrated in a monocular SLAM by Rituerto *et al*. in [19].

In this work we build on state of the art robust EKF monocular SLAM [7]. We integrate the Spherical Camera Model in a Real Time application. The main differences with the work developed in [19] is that now we use image patches instead of SIFT descriptors and our solution includes robust detection of spurious, operating at video sampling rate. Besides that we develop a patch for catadioptric cameras which considers rotation and scale invariance in function of mirror parameters. To reach rotation invariance we base on Andreasson proposal [1]. For scale invariance we develop a formulation of the scale factor in function of the mirror parameters which can be applied in any kind of central camera and, in particular, in a hiper-catadioptric system compound by a hiperbolic mirror coupled with a perspective camera.

The paper is structured as follows. Spherical Camera Model is described in Section 2. The SLAM problem is presented in Section 3 together with the Spherical Camera Model adaptation for the EKF. In section 4 our patch for the omnidirectional camera is formulated. Finally the results of the experiments with the new patch are presented in Section 5, and conclusions are presented in Section 6.

2. The Spherical Camera Model

First of all we describe the projection model for the omnidirectional catadioptric systems presented in [10] and extended in [3]. We start with a 3D point expressed in homogeneus coordinates $\mathbf{X} = [x, y, z, 1]$. Its projection on the image is divided in the following steps:

1) Point $\mathbf{X}$ is mapped into a projective ray $\mathbf{x}$ in the camera reference frame. This is done by P, a conventional projection matrix $\mathbf{x} = \mathsf{P}\mathbf{X}$.

2) The ray $\mathbf{x}$ is projected onto the unit sphere centered in the origin $\mathbf{O}$. The intersection point is projected to a virtual projection plane π through the virtual projection center $\mathbf{C_P} = (0, 0, -\xi)^\mathsf{T}$ yielding the point $\mathbf{x}'$. This step is coded by the non-linear function $\hbar$:

$$\mathbf{x}' = \hbar(\mathbf{x}) = \begin{pmatrix} x \\ y \\ z + \xi\sqrt{x^2 + y^2 + z^2} \end{pmatrix} \tag{1}$$

3) The virtual plane π is transformed in the image plane π_{IM} through a homographic transformation $\mathsf{H_c}$

$$\mathbf{x}'' = \mathsf{H_c}\mathbf{x}' \tag{2}$$

$$\mathsf{H_c} = \mathsf{K_c}\mathsf{R}\mathsf{M_c} \tag{3}$$

where $\mathsf{K_C}$ includes the camera parameters, $\mathsf{M_C}$ includes the mirror parameters [10] and R is the rotation matrix between camera and mirror. By assuming a pin-hole camera model and $\mathsf{R} = \mathsf{I}$, the transformation $\mathsf{H_C}$ yields:

$$\mathsf{H_C} = \begin{bmatrix} \eta f & 0 & u_0 \\ 0 & \eta f & v_0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} \gamma & 0 & u_0 \\ 0 & \gamma & v_0 \\ 0 & 0 & 1 \end{bmatrix} \tag{4}$$

where $\gamma = \eta f$ is the generalized focal lenght of the camera-mirror system with η a mirror parameter and f the focal length of the camera.

4) Finally image coordinates are calculated by dividing $\mathbf{x}''$ by its z'' coordinate:

$$\mathbf{p} = \begin{pmatrix} u \\ v \\ 1 \end{pmatrix} = f_u(\mathbf{x}'') = \begin{pmatrix} \frac{x''}{z''} \\ \frac{y''}{z''} \\ \frac{z''}{z''} \end{pmatrix} \tag{5}$$

The parameter of the model, ξ depends only on the system modeled and the geometry of the mirror. For conventional cameras $\xi = 0$. $\xi = 1$ for catadioptric systems with parabolic mirror and orthographic camera, and $0 < \xi < 1$ with hyperbolic mirror and perspective camera.

With this model it is also possible to estimate the 3D ray from where the image point comes. That projection is named the inverse projection model. It starts with the point in image coordinates $\mathbf{p} = (u,\ v)^\mathsf{T}$, being $\mathbf{x}'' = (u,\ v,\ 1)^\mathsf{T}$. The equations of the inverse projection model are:

$$\mathbf{x}' = {\mathsf{H_c}}^{-1}\mathbf{x}'' \tag{6}$$

$$\mathbf{x} = \hbar^{-1}(\mathbf{x}') = \begin{pmatrix} x' \\ y' \\ z' - \frac{\xi(x'^2 + y'^2 + z'^2)}{\xi z' + \chi} \end{pmatrix} \tag{7}$$

where $\chi = \sqrt{(1 - \xi^2)(x'^2 + y'^2 + z'^2)}$

3. Simultaneous Localisation And Mapping

The most used SLAM algorithms are based on the Kalman Filter, a filter that predicts the state of linear systems. As the geometry imposes non-linear relations, the Extended Kalman Filter (EKF) [25] is used. The EKF linearize the non-linear functions by approximating them to its first order Taylor series. The EKF is divided into two parts. In the first part, *Prediction* , the new state of the system is estimated from the previous time step state through the motion model. The second part of the algorithm, *Update* , uses the measurements of the environment to improve the new state prediction. The full state vector, composed of both the map and last camera location, is modelled as a multidimensional Gaussian distribution coded by its mean vector and covariance matrix.

The state of the system is given by the state vector $\mathbf{x}$

$$\mathbf{x} = (\underbrace{\mathbf{r},\ \mathbf{q},\ \mathbf{V},\ \omega}_{\text{Camera state}}, \underbrace{x_i, y_i, z_i, \theta_i, \phi_i, \rho_i, ...}_{\text{3D points (IDP)}}, \underbrace{X_j, Y_j, Z_j, ...}_{\text{3D points}}) \tag{8}$$

where $\mathbf{r}_{(3\times1)}$ is the camera pose, $\mathbf{q}_{(4\times1)}$ is the quaternion of its orientation and $\mathbf{V}_{(3\times1)}$ and $\omega_{(3\times1)}$ are its linear and angular velocities, respectively. The state size of the map features depends on the depth uncertainty they have. Features with large depth uncertainty are parametrised in inverse depth parametrisation (IDP) [6]. This parametrization

is used for recently initialised features. They are initialised with an arbitrary depth prior of ρ_{0i} with large uncertainty. In succesive observations of the feature, depth estimation is gradually refined. If the depth uncertainty of a feature decreases under a certain treshold then the state of the feature is given by its cartesian coordinates in the world reference frame. Since $\mathbf{x}$ has n dimensions the state covariance matrix P is a squared $n \times n$ matrix.

3.1. The Spherical Camera Model for the EKF

The EKF algorithm requires the first derivative of the measurement equation. So, the jacobian of the Spherical Camera Model must be computed [19].

$$\mathsf{J} = \mathsf{J}_{\mathsf{fu}} \mathsf{H}_{\mathsf{C}} \mathsf{J}_{\hbar} \tag{9}$$

$$\mathsf{J}_{\mathsf{fu}} = \begin{bmatrix} \frac{1}{z''} & 0 & -\frac{x''}{z''^2} \\ 0 & \frac{1}{z''} & -\frac{y''}{z''^2} \end{bmatrix} \tag{10}$$

$$\mathsf{J}_{\hbar} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ \frac{\xi x}{\rho} & \frac{\xi y}{\rho} & 1 + \frac{\xi z}{\rho} \end{bmatrix} \tag{11}$$

where $\rho = \sqrt{x^2 + y^2 + z^2}$

To initialize new features, the inverse jacobian of the model is also required:

$$\mathsf{J}^{-1} = \mathsf{J}_{\hbar^{-1}} \mathsf{H}_{\mathsf{C}}{}^{-1} \tag{12}$$

$$\mathsf{J}_{\hbar^{-1}} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ -\frac{\xi x'}{\chi} & -\frac{\xi y'}{\chi} & 1 - \frac{\xi(z' - \xi \frac{x'^2 + y'^2 + z'^2}{\xi z' + \chi})}{\chi} \end{bmatrix} \tag{13}$$

where $\chi = \sqrt{(1 - \xi^2)(x'^2 + y'^2 + z'^2)}$

3.2. Data Association and Map Management

Robust 1-point-RANSAC [7] based active search is applied. At each prediction step, an elliptical search region is computed from the measurement prediction and its corresponding covariance innovation. Correlation is computed for every pixel inside the search region. The pixel scoring highest is selected as putative match. In a second stage, joint scene rigidity is check for all the putative matches and spurious matches are detected. Active search is both efficient, because only a reduced fraction of the image is searched, and robust because of the reduced false positive rate in the putative matching computation. New point features are initiliased when the number matched map points are under a threshold. To improve geometrical condition map features have to be spread all over the image. FAST key-points scoring higher are searched in feature depleted image areas. From these key-points new map features are initialized. In order to keep complexity low, map features that are repeatedly no detected are marginalized out from the state vector.

4. New patch formulation

In this section we develop the formulation of a patch for catadioptric cameras invariant to rotation and scale.

Figure 1: Rotation transformation computed from $\Delta\theta = \theta_{pred} - \theta_{ini}$ is applied to a big patch. New patch for correlation is extracted from the warped patch.

4.1. Rotation invariance

For the rotation invariance we inspire on the idea proposed by Andreasson *et al.* [1]. A squared oriented patch is extracted by bilinear interpolation in the radial direction from the principal point to the feature. This patch is then rotated to a fixed orientation and stored as descriptor.

However, in the used SLAM application matching is done by active search in an elliptical region. So, if we use Andreasson's approach each candidate patch inside the search region should be determined by bilinear interpolation in the non-natural radial and polar directions of the image, which would be time consuming.

To avoid this, we combine this idea with the implementation existing in the SLAM application [17]. A bigger patch is extracted during feature initialisation. Before the matching process, big patch is warped by an homographic transformation [11] to predict how the appearance patch varies depending of the variation of the position of the camera respect to the position in which the feature was initialised. New patch for correlation is extracted from the center of the warped big patch. This way patches for correlation are always determined in the horizontal and vertical directions of the image and bilinear interpolation is only computed during big patch warping.

For its use with omnidirectional cameras, instead of computing the homography, we transform the patch by a rotation transformation given by the variation of the polar angle ($\Delta\theta$) of the feature in the image between the current prediction of its projection and the position where it was first observed and initialised (Fig. 1).

4.2. Scale invariance

To reach scale invariance, we develop the simple idea of scaling the patch by a given scale factor. To consider the variable resolution in the catadioptric image, a theorical formula was obtained in function of the mirror parameters

and the image position.

To obtain it, first we define a point in the 3D space in homogeneus coordinates at a distance or depth D from the camera with an azimut ϕ and an elevation of θ. Due to the rotational symetry of the mirror and for the sake of simplicity an azimuth angle of $\phi = 0$ is taken without loss of generality. So the coordinates of the 3D point yield $\mathbf{X_0} = (D\cos\theta,\ 0,\ D\sin\theta,\ 1)^{\mathsf{T}}$ According to the spherical camera model, this point is projected on the image plane as $\mathbf{p_0} = (\frac{\gamma\cos\theta}{\xi+sin\theta},\ 0,\ 1)^{\mathsf{T}}$ taking the reference frame attached to the principal point ($u_0 = v_0 = 0$ in the matrix $\mathsf{H_C}$). The norm of the projected point is the distance from the principal point R_{im}:

$$R_{im} = \|\mathbf{p_0}\| = \frac{\gamma\cos\theta}{\xi + sin\theta} \tag{14}$$

Projection of the points in the neighbourhood of $\mathbf{X_0}$ can be approximated by a linear mapping from the 3D scene to the image plane given by the projection jacobian of (9) computed in $\mathbf{X_0}$, which after some calculations and algebraic manipulation yields:

$$\mathsf{J}_{\mathbf{X}=\mathbf{X_0}} = \frac{\gamma}{D(\xi+S_\theta)^2}\begin{bmatrix} S_\theta(1+\xi S_\theta) & 0 & -C_\theta(1+\xi S_\theta) \\ 0 & \xi+S_\theta & 0 \end{bmatrix} \tag{15}$$

where $S_\theta = \sin\theta$ and $C_\theta = \cos\theta$. This jacobian maps points from a 3D to a 2D euclidean space. To extend it to a projective transformation in the projective space we do:

$$\mathsf{P}_{\mathsf{J}(3\times4)} = \begin{bmatrix} \mathsf{J}_{\mathbf{X}=\mathbf{X_0}} & \mathbf{0} \\ \mathbf{0}^{\mathsf{T}} & 1 \end{bmatrix} \tag{16}$$

Now lets take a sphere of radius $r << D$ centred on $\mathbf{X_0}$. It is parameterised by a quadratic form with matrix:

$$\mathsf{Q}_{(4\times4)} = \begin{bmatrix} \mathsf{I} & \mathbf{0} \\ \mathbf{0}^{\mathsf{T}} & -r^2 \end{bmatrix} \tag{17}$$

And its projection is computed as follows:

$$\mathsf{C} = (\mathsf{P_J}\mathsf{Q}^{-1}\mathsf{P_J^{\mathsf{T}}})^{-1} = \begin{bmatrix} \frac{\gamma^2(1+\xi S_\theta)^2}{D^2(\xi+S_\theta)^4} & 0 & 0 \\ 0 & \frac{\gamma^2}{D^2(\xi+S_\theta)^2} & 0 \\ 0 & 0 & \frac{-1}{r^2} \end{bmatrix} \tag{18}$$

where C is the matrix which determines the quadratic form of a ellipse with major and minor semi-axis:

$$r_{im}^{+} = \gamma\frac{r}{D}\frac{1+\xi\sin\theta}{(\xi+\sin\theta)^2} \tag{19}$$

$$r_{im}^{-} = \gamma\frac{r}{D}\frac{1}{\xi+\sin\theta} \tag{20}$$

The previous steps are shown schematically in figure 2.

From (14) for θ we can calculate $\sin\theta$ as a function of R_{im} and the parameters ξ and γ, which we call $f(\xi, \frac{R_{im}}{\gamma})$. After some substitutions and manipulation we get a second order equation with unknown $\sin\theta$. By solving the equation and selecting the solution with physical meaning, we obtain:

$$S_\theta = f(\xi, \frac{R_{im}}{\gamma}) = \frac{\sqrt{1+(\frac{R_{im}}{\gamma})^2(1-\xi^2)} - \xi(\frac{R_{im}}{\gamma})^2}{1+(\frac{R_{im}}{\gamma})^2} \tag{21}$$

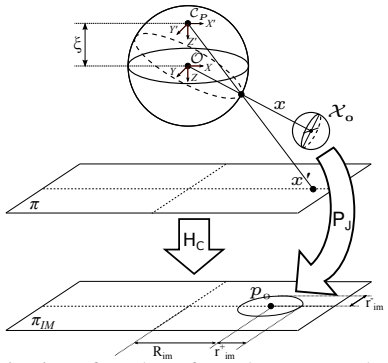


Figure 2: Projection of a sphere from the scene to the image plane by the jacobian computed on its centre $\mathbf{X_0}$

According to the obtained formulas for the semiaxis it is deduced that the scale of a feature on the image depends on the following parameters:

- Real size of the feature (r)
- Distance of the feature to the camera (D)
- Camera-mirror parameters ξ and γ
- Distance in the image to the principal point (R_{im})

To compute the scale factor the real size of the feature is not relevant since it does not change between frames.

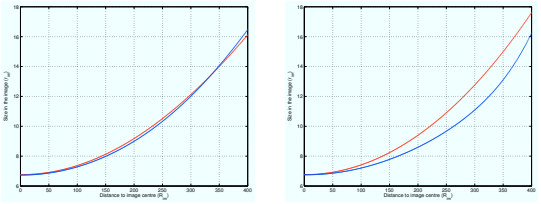
Concerning the contribution of R_{im} and the mirror parameters, to apply a uniform scale factor, one of the two formulas (19) and (20) must be selected. A sensibility test to the the unmodelled image distortion and the linearization error induced by the jacobian is lead. The test involves a simulation of the projection of a set of spheres with $D = 6\ m$, $r = 0.1\ m$ and a gradual shift on elevation angle using a real camera calibration with 5 distortion parameters [16]. From the simulation results, empirical data is obtained for the dependency on R_{im} of the semiaxis in the radial and tangential directions of the projected ellipses. These functions are compared with the functions for r_{im}^{+} and r_{im}^{-} respectively (Fig. 3). For the major semiaxis the maximum absolute and relative errors are 2 $pixels$ and a 15% respectively, while for the minor semiaxis the maximum errors are 0.3 $pixels$ and a 2%.

Therefore we select r_{im}^{-} to calculate the scale factor, which yields:

$$k = \frac{r_{im2}^{-}}{r_{im1}^{-}} = \frac{D_1}{D_2}\frac{\xi + f(\xi, \frac{R_{im1}}{\gamma})}{\xi + f(\xi, \frac{R_{im2}}{\gamma})} \tag{22}$$

The depth of the feature in the scene D is the most problematic contribution since it is not observable in one image. As explained in Section 3, new detected features are initialized in IDP with an arbitrary depth value with high uncertainty, which is not reliable to calculate the scale factor. For this reason, the application of the whole scale factor can only be considered with fully initialised features, which are assumed to have a reasonable depth uncertainty.

For features with a high depth uncertainty, the application of a partial scaling droping the depth terms can be considered. To evaluate it, we consider that the camera moves



(a) (b)

Figure 3: Comparison of the theoretical formulas to calculate the ellipse semiaxis in which the sphere is projected (red) with the results of a simulation using a camera model with distortion parameters (blue). Figure (a) for the minor semiaxisis. Figure (b) for the major semiaxis

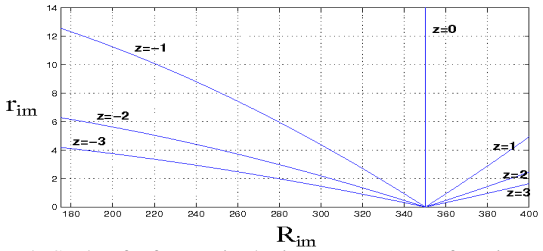


Figure 4: Scale of a feature in the image (r_{im}) as a function of the distance to the principal point (R_{im}) and the distance in meters (z) to the plane where the camera moves.

in a plane, so relative movement of the tracked features takes place in parallel planes. By making $D = \frac{z}{\sin\theta}$ in (20), we obtain the dependence of r_{im} on $\sin\theta$, and so on R_{im}, at different distances z from the plane where the camera moves (Fig. 4). For features below the camera ($z < 0$) their scale decreases until 0 in the line at infinity (given by the circunference with radius $R_{im} = R_\infty = \frac{\gamma}{\xi}$); while for features above the camera its scale increases from R_∞.

However, the contribution of the shape of the mirror always increases with R_{im} (Fig. 3). So, assuming movement in a plane, the use of a scale factor without depth estimate only makes sense when the tracked features are above the camera (*i.e.* $R_{im} > R_\infty$). As it only suposes a fraction of the image, the implementation of the partial scale factor for IDP features was eventually not considered.

4.3. Computation of patch transformation

Before computing the patch transformation, it must be checked that the descriptor patch will be fully contained in the warped big patch. The limit situation arises when the warped patch "touches" the corners of the descriptor patch (Fig. 5). In this case, the scale factor is:

$$k = \sqrt{2}\frac{h_P}{h_{BP}}\cos(\frac{\pi}{4} - \mod(\Delta\theta, \frac{\pi}{2})) \quad (23)$$

where h_{BP} and h_P are the half of the sizes of the big patch and the descriptor patch respectively and $\Delta\theta$ is the variation of the polar angle used to construct the transformation. If we add a security margin of 0.1 we obtain an expression for the limit of the scale factor:

$$k_{lim} = \sqrt{2}\frac{h_P}{h_{BP}}\cos(\frac{\pi}{4} - \mod(\Delta\theta, \frac{\pi}{2})) + 0.1 \quad (24)$$

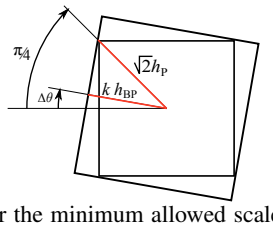


Figure 5: Limit for the minimum allowed scale factor ($kh_{BP} = \sqrt{2}h_P\cos(\frac{\pi}{4} - \mod(\Delta\theta, \frac{\pi}{2}))$)

With this previous consideration, the computation of the warping is done in the following steps:

1) Check the condition $k > k_{lim}$. If k does not fill this condition it is set to k_{lim}.

2) Calculation of the transformation matrix:

$$\mathsf{H} = \mathsf{H}_{\mathsf{tr}}\mathsf{H}_{\mathsf{S}}{\mathsf{H}_{\mathsf{tr}}}^{-1} \quad (25)$$

$$\mathsf{H}_{\mathsf{S}} = \begin{bmatrix} k\cos(\Delta\theta) & -k\sin(\Delta\theta) & 0 \\ k\sin(\Delta\theta) & k\cos(\Delta\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (26)$$

with H_{S} the transformation matrix which combines the rotation transformation $\mathsf{R}_{\Delta\theta}$ with the scale factor k and H_{tr} is a translation matrix to translate the patch coordinate frame to the center of the patch.

3) Computation of the warped patch by doing the inverse mapping to the original big patch and performing a bilinear interpolation.

$$\mathbf{X_{BP}} = {\mathsf{H}_{\mathsf{S}}}^{-1}\mathbf{X_{WP}} \quad (27)$$

4) Extraction of the patch for correlation from the center of the warped patch.

5. Experiments

In this section experiments to evaluate the omnidirectional visual SLAM and the new patch are presented. The images used to lead the experiments were taken from one of the image databases provided by The Rawseeds Project[1]. This database consists of a sequence with more than 32000 frames acquired by a robot equiped with a hyper-catadioptric camera.

5.1. Experiment 1

We carried three tests out where we decoupled the matching process from the SLAM algorithm to make a preliminary evaluation of the rotation and the scale factor transformation. The set up of the three tests was quite similar. First, we extracted corners on the first frame with the FAST extractor. Among the extracted corners, we selected some features and stored their locations and their patches. Like the matching proccess has been decoupled from the SLAM algorithm, the true locations of the features were manually selected on each frame (Fig. 6) and the search region was fixed to a 50x50 pixels square. The matching in the selected frames is done by obtaining for each feature the best correlation inside the search region, as is done in the SLAM application.

[1] HTTP://www.rawseeds.org

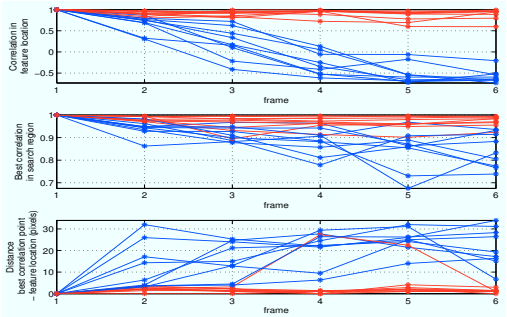


Figure 7: Matching results of test 1. In red, with oriented patch. In blue, with not oriented patch

For each feature and frame, we have defined the following variables to be measured:

- Correlation in true feature location
- Best correlation in search region
- Distance between true feature location and best correlation location

5.1.1 Test 1: Rotated patch and 180º rotation

We evaluate the rotation of the patches in a sequence in which the robot rotates 180º. From this sequence we have extracted 6 frames spaced by 20 frames between them. We have selected 9 features to carry the test out (Fig. 6).

The results show that rotated patches provide a better correlation value on the true feature location, which confirms that they are more rotation invariant than the non rotated patches (Fig. 7). The rotated patch also provides by far better values for the best correlation inside the search region (all of them above 0.9) as well as a very low distance between true feature location and matched location.

5.1.2 Test 2: Rotated patch and translation

Performance of rotated patches is evaluated in a sequence only containing camera translation. 6 frames were extracted with intervals of 20 frames between them and the number of selected features was 6 (Fig. 8).

The results (Fig. 10) reveal that slightly better correlation values are obtained using a rotated patch. This is due to the relative motion of the map features along lines which are projected as conic curves in a catadioptric image. So, the rotated patch initially intended to improve the matching results during camera rotations, can also deal with translations better than a non rotated patch. It can be seen too, that as the distance the robot has translated increases, the matchings tend to be made in the wrong location for both patches. One posible reason is that as the robot translates the features change their scale and their point of view and can become hidden by other scene objects.

5.1.3 Test 3: Scaled patch

We evaluate the performance of the matching process with respect to the scale changes. Due to the decoupling from SLAM, it is not possible to determine the depth of the patches extracted and the contribution of the depth to the scale factor is not considered. So, an image adquisition

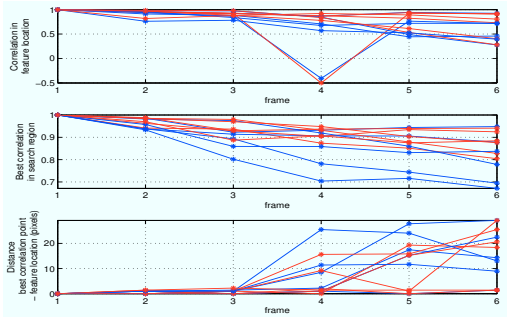


Figure 10: Matching results of test 2. In red, with oriented patch. In blue, with not oriented patch

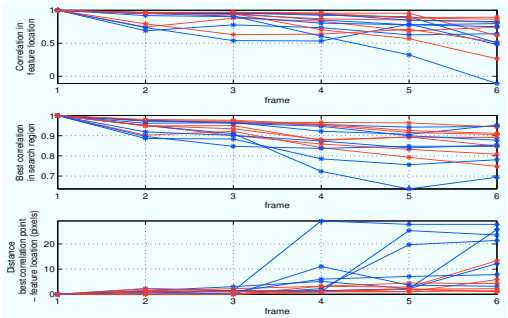


Figure 11: Matching results of test 3 (scale decrease). In red, with scaled patch. In blue, with unscaled patch

without depth changes in the features has been made following the next steps:

- Select a zone in the scene with potential patch richness and situated far enough from the camera so that $D \to \infty$.

- Capture images while camera rotates so that the selected zone moves only along the radial direction. As infinite distance has been assumed, little camera displacements during capture are not problematic.

A sequence of 6 images was taken for the test. The number of selected features was 7. To evaluate the performance of the patches under scale change, the features were selected in a zone with no orientation change in the image (Fig. 9). Two cases have been carried out to prove the performance under scale decrease $k < 1$ and scale increase $k > 1$. In the scale decrease case, the extraction of the features was made in the image where the zone of extraction was the furthest from the image centre. For the scale increase case the order of the images has been inverted.

The results of the tests show that in scale decrease (Fig. 11) the patch with scaling offers a better performance than a normal patch while in the case of scale increase(Fig.12) both patches perform in a similar way, due to the impossibility of extracting new information by increasing the scale of an image.

5.2. Experiment 2

After testing the new transformations applied to the patch, we evaluated it integrated in our visual SLAM approach for omidirectional cameras with the Real-Time application developed by Davison *et al*. For the evaluation

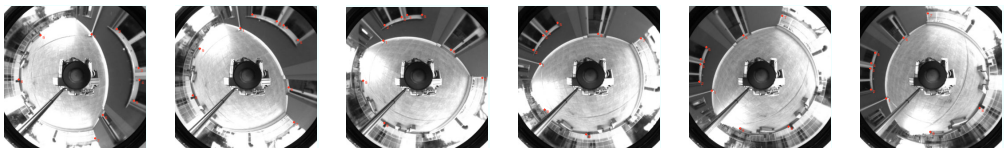


Figure 6: Image sequence taken for test 1 (180^o rotation). Selected corners for matching are shown in red

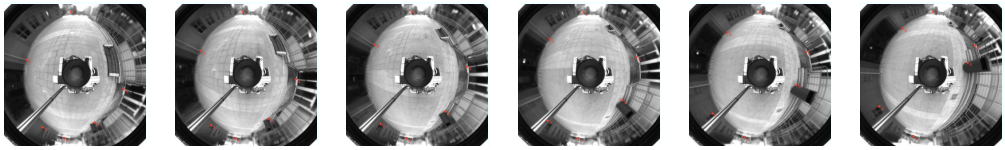


Figure 8: Image sequence taken for test 2 (translation). Selected corners for matching are shown in red

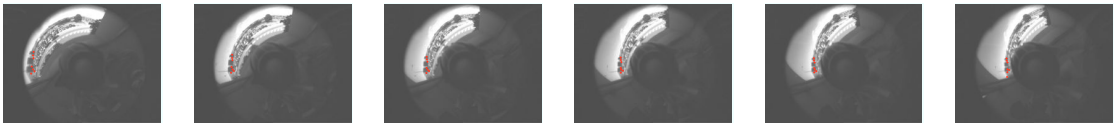


Figure 9: Image sequence taken for test 3 (scale change). Selected corners for matching are shown in red

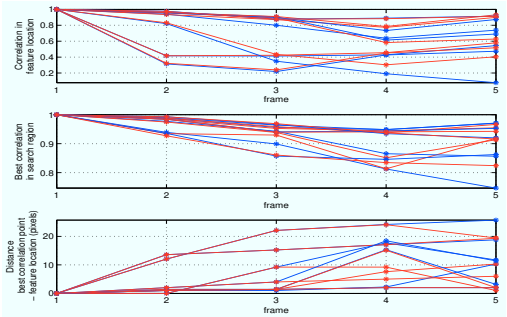


Figure 12: Matching results of test 3 (scale increase). In red, with scaled patch. In blue, with unscaled patch.

we selected a long outdoor sequence from the database provided by the Rawseeds Project.

To compare our warped patch with the normal patch we ran the sequence using both patches with different correlation thresholds for matching and we have measured three variables:

- Total number of features initialised (FI) .
- Matchings per feature ratio: ($R_m = \frac{Total\ matchings}{FI}$) .
- Features in map per feature initialised ratio: ($R_f = \frac{Final\ map\ size}{FI}$)

The results in Table 1 show that the warped patch for omnidirectional cameras performs better than the patch with no warping, as SLAM initialises less features and is obtains more information for SLAM per initialised feature. On the other, the correlation threshold may have more influence on the measured variables than the kind of patch, but at the expense of reducing the *a priori* precision of SLAM. In Fig. 13 the projections on the XY plane and the YZ plane of the trajectory obtained with the new patch and a correlation threshold of 0.8 are shown. Note that although the MonoSLAM application estimates 3D camera motion, being not bounded to a 2D plane, according to the obtained trajectory the camera is moving on the ground plane.

Table 1: Total number of initialised fatures (FI), Matchings per initialised feature (R_m) and features in map per initialised feature(R_f)

Correlation threshold	Warped patch			Normal patch		
	FI	R_m	R_f	FI	R_m	R_f
0.8	8648	22.31	0.1	8923	19.75	0.087
0.9	9834	19.38	0.062	10854	16.09	0.046
0.95	13189	13.31	0.027	14970	10.56	0.019

Finally we compared this trajectory with respect to the ground truth provided by the GPS data. As scale is not observable by one single camera, for the comparison we scaled the trajectory and aligned it with the trajectory obtained with the GPS (Fig. 14). To evaluate the accuracy of the SLAM trajectory numerically we have calculated the mean error of the distance between the corresponding points of both trajectories. The mean error is $\mu_{err} = 3.44\ m$ with a standard deviation of $\sigma = 1.93\ m$ and a maximum error of $max_{err} = 6.73\ m$. Dividing by the trajectory lenght, we obtain a relative mean error of 1%.

6. Conclusion

In this work we have developed a Visual SLAM for omnidirectional cameras building on state of the art EKF monocular SLAM [7] for conventional cameras. Two main modifications have been made: the implementation of the Spherical Camera Model for projection and the formulation of a new patch for omnidirectional cameras which aims to be rotation and scale invariant. Then we have lead experiments to compare the new patch with a conventional patch. First we have tested the matching process decoupled from the SLAM. Once the supperiority of the new patch has been proven we have run in real time the SLAM algorithm in a 340 meters long trajectory. Results have shown that the obtained trajectory estimation is quite accurate, which encourages us to make experiments with longer trajectories in the future.

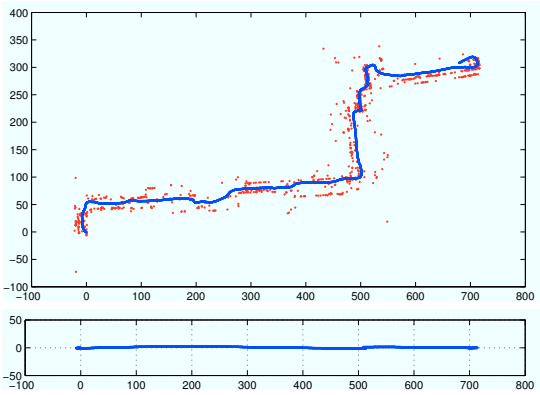


Figure 13: SLAM trajectory with correlation threshold 0.8 using the warped patch projected on the XY plane (up) and on the YZ plane (down) The red dots are the map features

Figure 14: GPS trajectory (red) and SLAM trajectory (green) superposed on the satellite image of the Campus of Bovisa (Milan) where the sequences were adquired

Acknowledgment

This work has been supported by the project DPI2009-14664-C02-01. Thanks to the I3A Fellowship Program. We thank A. J. Davison and all the people involved in the development of the MonoSLAM application used in this work.

References

[1] H. Andreasson, A. Treptow, and T. Duckett. Self-localization in non-stationary environments using omni-directional vision. *Robot. Auton. Syst.*, 55:541–551, July 2007. 1, 2, 3

[2] M. S. Arulampalam, S. Maskell, and N. Gordon. A tutorial on particle filters for online nonlinear/non-gaussian bayesian tracking. *IEEE Transactions on Signal Processing*, 50:174–188, 2002. 1

[3] J. Barreto and H. Araujo. Issues on the geometry of central catadioptric image formation. In *CVPR, 422–427*, 2001. 2

[4] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool. Speeded-up robust features (SURF). *Comput. Vis. Image Underst.*, 110:346–359, June 2008. 1

[5] R. Chang, S. Ieng, and R. Benosman. Auto-organized visual perception using distributed camera network. 57(11):1075–1082, November 2009. 1

[6] J. Civera, A. J. Davison, and J. M. M. Montiel. Inverse depth parametrization for monocular SLAM. *IEEE Transactions on Robotics*, 24(5):932–945, October 2008. 3

[7] J. Civera, O. G. Grasa, A. J. Davison, and J. M. M. Montiel. 1-Point RANSAC for EKF Filtering: application to real-time structure from motion and visual odometry. *Journal of Field Robotics*, 27(5):609–631, October 2010. 1, 2, 3, 7

[8] P. Corke, D. Strelow, and S. Singh. Omnidirectional visual odometry for a planetary rover. In *IROS*, 2004. 1

[9] A. J. Davison. Real-time simultaneous localisation and mapping with a single camera. In *ICCV*, 2003. 1

[10] C. Geyer and K. Daniilidis. A unifying theory for central panoramic systems and practical applications. In *ECCV (2)*, 445–461, 2000. 2

[11] R. Hartley and A. Zisserman. *Multiple View geometry in Computer vision*. Cambridge university press, 2000. 3

[12] M. Heikkila, M. Pietikainen, and C. Schmid. Description of interest regions with local binary patterns. *Pattern Recognition*, 42(3):425–436, March 2009. 1

[13] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *Int. J. Comput. Vision*, 60:91–110, 2004. 1

[14] H. Lu and Z. Zheng. Two novel real-time local visual features for omnidirectional vision. *Pattern Recogn.*, 43:3938–3949, December 2010. 2

[15] C. Mei. *Laser-Augmented Omnidirectional Vision for 3D Localisation and Mapping*. PhD thesis, INRIA Sophia Antipolis, Project-team ARobAS, 2007. 1

[16] C. Mei and P. Rives. Single view point omnidirectional camera calibration from planar grids. In *ICRA*, April 2007. 4

[17] N. Molton, A. Davison, and I. Reid. Locally planar patch features for real-time structure from motion. In *British Machine Vision Conference*, 2004. 3

[18] A. Rituerto, L. Puig, and J. J. Guerrero. Comparison of omnidirectional and conventional monocular systems for visual slam. In *10th OMNIVIS with RSS,* , 2010.

[19] A. Rituerto, L. Puig, and J. J. Guerrero. Visual slam with an omnidirectional camera. In *20th ICPR, 348–351*, 2010. 2, 3

[20] E. Rosten, R. Porter, and T. Drummond. Faster and better: A machine learning approach to corner detection. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 32:105–119, 2010. 1

[21] D. Scaramuzza, N. Criblez, A. Martinelli, and R. Siegwart. Robust Feature Extraction and Matching for Omnidirectional Images. In *6th International Conference on Field and Service Robotics* , 2007. 1

[22] D. Scaramuzza, F. Fraundorfer, and R. Siegwart. Real-time monocular visual odometry for on- road vehicles with 1-point ransac. In *ICRA*, 4293–4299, 2009. 1

[23] T. Svoboda and T. Pajdla. Matching in catadioptric images with appropriate windows, and outliers removal. In *9th CAIP, 733-740*, 2001. 1

[24] J.-P. Tardif, Y. Pavlidis, and K. Daniilidis. Monocular visual odometry in urban environments using an omnidirectional camera. In *IROS,2531-2538*, 2008. 1

[25] S. Thrun, W. Burgard, and D. Fox. *Probabilistic Robotics*. MIT Press, 2005. 1, 2