# Comparison of omnidirectional and conventional monocular systems for visual SLAM

Alejandro Rituerto[1], Luis Puig[2], J. J. Guerrero[3]

[1]*RoPeRT*
*Aragón Institute of Engineering Research (I3A).*
*University of Zaragoza, Mariano Esquillor s/n, 50018, Zaragoza, Spain.*
*Tel. +34976762707, Fax +34976762043, e-mail: aleritu@gmail.com*

[2]*RoPeRT*
*Aragon Institute of Engineering Research (I3A).*
*University of Zaragoza, Mariano Esquillor s/n, 50018, Zaragoza, Spain.*
*Tel. +34976762707, Fax +34976762043*

[2]*RoPeRT*
*Aragon Institute of Engineering Research (I3A).*
*University of Zaragoza, Mariano Esquillor s/n, 50018, Zaragoza, Spain.*
*Tel. +34976762707, Fax +34976762043*

## Abstract

The SLAM (Simultaneous Localization and Mapping) problem is one of the essentials challenges for the current robotics. Our main objective in this work is to perform a comparison between a visual SLAM system using monocular omnidirectional and conventional vision. Our approach is based on the Extended Kalman Filter (EKF). We use the Spherical Camera Model to obtain geometric information from the images. To integrate this model in the EKF-based SLAM we linearize the direct and the inverse projections. The approach also uses the inverse depth parameterization of the measurements and SIFT points as measurement image points. We perform experiments with images of both systems corresponding to the same challenging trajectories. This experimentation confirms that the omnidirectional system gives a much better trajectory and orientation estimation than the conventional vision system.

**Keyword:** Visual-SLAM, Omnidirectional vision, Spherical Camera Model.

# 1 Introduction

The use of omnidirectional systems in robotics applications has increased quickly in the recent years. The main reason for this rise is the wide Field Of View (FOV) of this kind of systems. There are many types of omnidirectional systems. The rotating camera is a conventional camera with a mechanic system that allows capturing images all around the scene. There are systems compound of many cameras that focus to different parts of the surrounding, all the images acquired by these cameras are joined together to build a panoramic image of the scene. Other omnidirectional systems are the dioptric systems that uses special lenses with wide FOV such as fish-eye lens. In robotics applications the most used omnidirectional system is the catadioptric one, composed by a curved mirror and a conventional camera. This type of system has been used in many applications such as surveillance [Scotti et al., 2005], robot navigation [Chahl & Srinivasan, 2000], telepresence [Nagahara et al., 2006] and 3D reconstruction [Lhuillier, 2007]. In this work we use a hyper-catadioptric monocular system.

An autonomous robot should be able to navigate through a previously unknown environment. The SLAM [Thrun et al., 2005] 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. There exists several types of sensors to obtain information from the environment, e.g., radar, laser, sonar and vision. Among the visual SLAM approaches that use conventional vision we can find Davison et al. [Davison et al., 2007] where they present a real-time algorithm to recover 3D trajectory of a camera moving through an unknown scene. Pupilli and Calway [Pupilli & Calway, 2006] who describe a particle filtering SLAM extension which provides resilience to erratic motion. However just a few SLAM approaches use omnidirectional systems. Gaspar et al. [Gaspar et al., 2000] transform the omnidirectional images into bird's eye images and propose two navigation modalities: Topological navigation and Visual Path following. Huang and Song propose in [Huang & Song, 2008] a switching method of visual reference scans that can reduce the computation complexity of the Extended Kalman Filter. Kim and Chung [Kim & Chung, 2003] present a SLAM algorithm based on a vision sensor robust to the matching problem with cooperation of structure from motion and stereo vision.

When a vision system is used as measurement sensor a projection model is needed. Through this projection model we can obtain geometric information from the images. There are different projection models that describe central catadioptric systems [Kang, 2000], [Svoboda & Pajdla, 2002], [Scaramuzza et al., 2006], [Toepfer & Ehlgen, 2007], however the most used is the Spherical Camera Model [Geyer & Daniilidis, 2000], [Barreto & Araujo, 2001].

The omnidirectional systems can capture in just one image the $360^o$ of a scene. That makes possible to keep track of image points in all directions, while conventional systems only keeps track of image points inside a narrow FOV. Until our knowledge this is the first comparison of omnidirectional and conventional cameras for SLAM. That comparison confirms the advantages of the omnidirectional systems over the conventional systems.

The structure of this paper is the following. The Spherical Camera Model is described in Sec. 2. The SLAM problem is presented in Sec. 3 together with the Spherical Camera Model adaptation for the EKF. Finally the results of the comparison are presented in Sec. 4, and conclusions are presented in Sec. 5.

This work extends the previous work [Rituerto et al., 2010] accepted for the 20th International Conference on Pattern Recognition and has been accepted in the 2010 Robotics: Science and Systems Conference (RSS), in the workshop OMNIVIS 2010: The 10th Workshop on Omnidirectional Vision, Camera Networks and Non-classical Cameras.

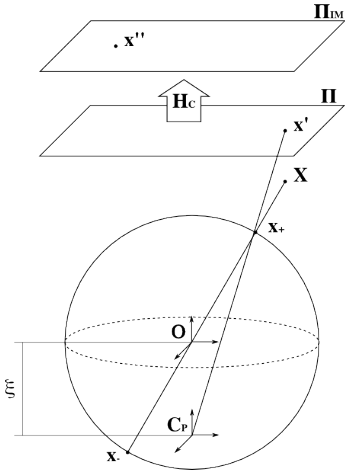


Figure 1: Spherical Camera Model Projection

## 2 The Spherical Camera Model projection

First of all we present the projection model for the omnidirectional catadioptric systems. In [Geyer & Daniilidis, 2000] Geyer and Daniilidis present a unified model of projection for any catadioptric system with a single viewpoint. Barreto and Araujo extend this method in [Barreto & Araujo, 2001]. This model of projection is known as the Spherical Camera Model and it is widely used to model omnidirectional vision systems.

The projection of a 3D point in the image is divided in three steps (see Fig. 1). The world reference system coincides with the camera reference system. The first step is the projection of the scene point $\mathbf{X}$ onto the unit sphere centered in the origin $\mathbf{O}$. There are two intersection points, $\mathbf{x}_+$ and $\mathbf{x}_-$, but just one is physically true. These points are projected to a virtual projection plane $\pi$ through the virtual projection center $\mathbf{C}_P = (0,\ 0,\ -\xi)^\mathsf{T}$. The new point is $\mathbf{x}'$. These two steps are coded in one equation:

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

The next step transforms the virtual plane $\pi$ in the image plane $\pi_{IM}$ through a homographic transformation $\mathsf{H}_C$.

$$\mathbf{x}'' = \mathsf{H}_C\mathbf{x}' \tag{2}$$

$$\mathsf{H}_C = \mathsf{K}_C\mathsf{R}\mathsf{M}_C \tag{3}$$

$$\mathsf{K}_C = \begin{pmatrix} f_x & s_{skew} & u_0 \\ 0 & f_y & v_0 \\ 0 & 0 & 1 \end{pmatrix} \tag{4}$$

$$\mathsf{M}_C = \begin{pmatrix} \psi-\xi & 0 & 0 \\ 0 & \xi-\psi & 0 \\ 0 & 0 & 1 \end{pmatrix} \tag{5}$$

where $\mathsf{K}_C$ includes the camera parameters, $\mathsf{M}_C$ includes the system parameters [Barreto & Araujo, 2001] and $\mathsf{R}$ is the rotation matrix of the camera with respect to the mirror.

Finally, the image coordinates are estimated by the next function:

$$\mathbf{u} = \begin{pmatrix} u \\ v \end{pmatrix} = f_u(\mathbf{x}'') = \begin{pmatrix} \frac{x''}{z''} \\ \frac{y''}{z''} \end{pmatrix} \tag{6}$$

The parameter of the model, $\xi$ 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 image coordinates $\mathbf{u} = (u,\ v)^{\mathsf{T}}$, being $\mathbf{x}'' = (u,\ v,\ 1)^{\mathsf{T}}$. The equations of the inverse projection model are

$$\mathbf{x}' = \mathsf{H}_C{}^{-1}\mathbf{x}'' \tag{7}$$

$$\mathbf{x} = \hbar(\mathbf{x}')^{-1} = \begin{pmatrix} x' \\ y' \\ z' - \frac{\xi(x'^2+y'^2+z'^2)}{\xi z'+\chi} \end{pmatrix} \tag{8}$$

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

Since the Spherical Camera Model deals with catadioptric systems and conventional cameras, our visual SLAM application can consider both vision systems just modifying the camera calibration parameters.

## 3 Simultaneous Localization and Mapping (SLAM)

An autonomous robot must be able to navigate through unknown environments. With SLAM techniques a robot can build a map of the environment and can keep track of its pose in this map using noisy measurements of the environment. In Visual SLAM these measurements are usually characteristic points in the images.

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) [Thrun et al., 2005] 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. Every state variable has a Gaussian distribution and it is represented by its mean and its covariance.

The system state in our EKF algorithm is coded in $\mathbf{x}$,

$$\mathbf{x} = (\underbrace{\mathbf{r},\ \mathbf{q},\ \mathbf{V},\ \omega}_{\text{Camera state}},\ \underbrace{x_i,\ y_i,\ z_i,\ \theta_i,\ \phi_i,\ \rho_i}_{\text{3D points}} ...)^{\mathsf{T}} \tag{9}$$

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 scene points are coded by inverse depth parameterization [Civera et al., 2008], $(x_i,\ y_i,\ z_i)$ is the pose of the camera the first time it observes the point $i$, $\theta_i$ and $\phi_i$ are the angles that determine the corresponding 3D ray, and $\rho_i$ is the inverse depth of the point. As correlation must be considered, if $\mathbf{x}$ has $n$ dimensions the state covariance matrix $\mathsf{C}$ is a squared $n \times n$ matrix.

### 3.1 The matching process

An essential step of the *Update* part of the EKF is the measurement matching. We use SIFT [Lowe, 2004] points technique to detect and match image features. The SIFT descriptor is orientation and scale invariant, which adapts to multiview matching. The matching is made in two steps for each new image. First, we match these 3D points with the new image features. The matching only uses the 3D points which were predicted to be inside the FOV of the current image. An 3D point is matched to a new image feature if the distance between their descriptors multiplied by a threshold is not greater than the distance from the 3D point descriptor to all the other new image features descriptor. The value of this threshold is 1.5. In second step the matched point must be inside the 95% confidence region predicted by the SLAM algorithm to be considered valid.

### 3.2 The Spherical Camera Model for the EKF

The EKF algorithm requires the first derivative of the measurement equation. In order to adapt the Spherical Camera Model to the EKF algorithm we have formulated the model derivatives, for the direct and the inverse projection.

- *Jacobian of the Sphere Camera Model projection*

$$\mathsf{J} = \mathsf{J}_{\mathsf{f_u}} \mathsf{H}_C \mathsf{J}_\hbar \tag{10}$$

$$\mathsf{J}_{\mathsf{f_u}} = \begin{pmatrix} \frac{1}{z''} & 0 & -\frac{x''}{z''^2} \\ 0 & \frac{1}{z''} & -\frac{y''}{z''^2} \end{pmatrix} \tag{11}$$

$$\mathsf{J}_\hbar = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ \frac{\xi x}{\rho} & \frac{\xi y}{\rho} & 1 + \frac{\xi z}{\rho} \end{pmatrix} \tag{12}$$

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

To initialize new features we require a way to estimate the covariance of the new feature from the covariance of the image point. This estimation can be done through the Jacobian of the Spherical Camera Model inverse projection.

- *Jacobian of the Sphere Camera Model inverse projection*

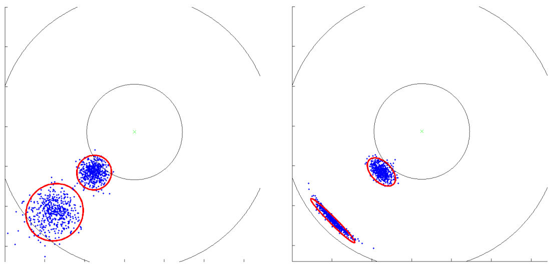
$$\frac{\partial \mathbf{x}}{\partial \mathbf{x}''} = \frac{\partial \mathbf{x}}{\partial \mathbf{x}'} \frac{\partial \mathbf{x}'}{\partial \mathbf{x}''} \tag{13}$$

$$\frac{\partial \mathbf{x}'}{\partial \mathbf{x}''} = \mathsf{H_C}^{-1} \tag{14}$$

$$\frac{\partial \mathbf{x}}{\partial \mathbf{x}'} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ \frac{\partial z}{\partial x'} & \frac{\partial z}{\partial y'} & \frac{\partial z}{\partial z'} \end{pmatrix} \tag{15}$$

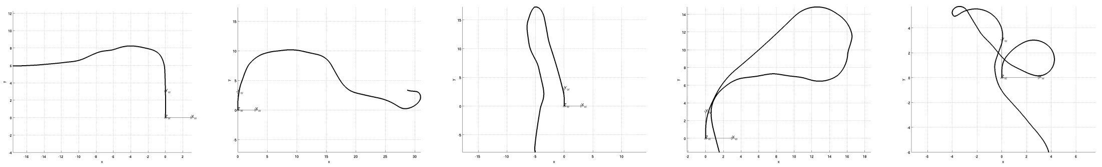
$$\frac{\partial z}{\partial x'} = -\frac{2\xi x'(\xi z' + \chi) - \frac{(1-\xi^2)\xi x' \rho^2}{\chi}}{(\xi z' + \chi)^2} \tag{16}$$

$$\frac{\partial z}{\partial y'} = -\frac{2\xi y'(\xi z' + \chi) - \frac{(1-\xi^2)\xi y' \rho^2}{\chi}}{(\xi z' + \chi)^2} \tag{17}$$



(a) (b)

Figure 2: Simulation of the projection of 3D points in a omnidirectional image. The black asterisks are the projection of the 3D central points, and the blue points are the projection of the cloud points around them. The red ellipses are the 95% confidence zone of the point according to the linearized model.



Trajectory 1 Trajectory 2 Trajectory 3 Trajectory 4 Trajectory 5

Figure 3: Selected trajectories

$$\frac{\partial z}{\partial z'} = -\frac{2\xi z'(\xi z' + \chi) - (\xi + \frac{z'}{\chi})\xi\rho^2}{(\xi z' + \chi)^2} \tag{18}$$

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

To verify the developed Jacobians and to show the size and shape of the uncertainty in the omnidirectional images we perform the next simulation. We simulate central 3D points and a Gaussian cloud of 500 points around each of them. After that we project those points through the Spherical Camera Model in a omnidirectional image. Two cases are considered in Fig. 2. In both cases two 3D points situated at the same distance from the camera are considered. In fig. 2(a) the points have a 3D uncertainty, so the cloud of points form a sphere. In Fig. 2(b) a 2D uncertainty in a horizontal plane perpendicular to the optical axis is considered, to simulate points extracted from the ground plane. In both cases the uncertainty near the center of the image is smaller than in the outer part of the image because of the non-homogeneous resolution of the omnidirectional image. In the case of the 2D uncertainty the width of the ellipse in the radial direction is smaller than in tangential direction, specially in the outer part of the image.

## 4 Experiments

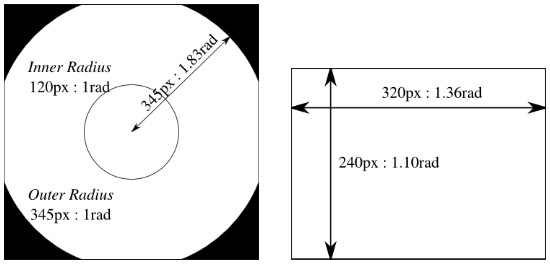
The SLAM approach used to perform the comparison is based on a basic visual SLAM[1] using the EKF algorithm and codifying the measurements points by inverse depth parametrization [Civera et al., 2008]. We have added the Spherical Camera Model in the measurement equation and the SIFT features are used as image points. A bucketing is also used to allocate features all over the image. We use two image sequences provided by The Rawseeds Project[2]. They have been acquired by a robot with a hyper-catadioptric camera and

[1] http://www.robots.ox.ac.uk/~SSS06/Website/Practicals/~SSS06.Prac2.MonocularSLAM.tar.gz

[2] http://www.rawseeds.org

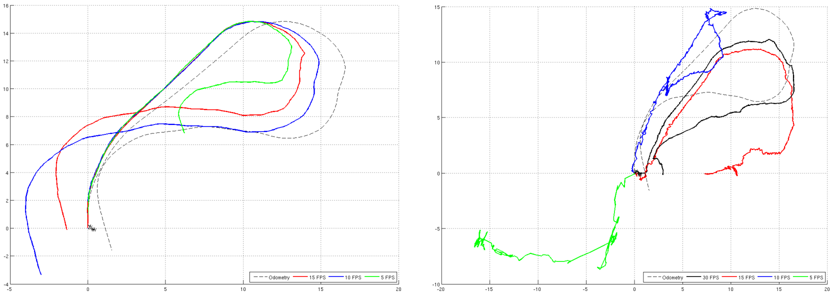
Table 1: Mean of the orientation error in radians for different values of the framerate, Trajectory 4

| Framerate | Omnidirectional camera | Conventional camera |
|---|---|---|
| 30 FPS | – | 0.22 |
| 15 FPS | 0.07 | 0.63 |
| 10 FPS | 0.07 | 0.26 |
| 5 FPS | 0.10 | 0.54 |



(a) (b)

Figure 4: Angular resolution of the cameras



(a) (b)

Figure 5: Results of the framerate analysis of the trajectory 4 for the omnidirectional system, (a), and for the conventional system, (b).

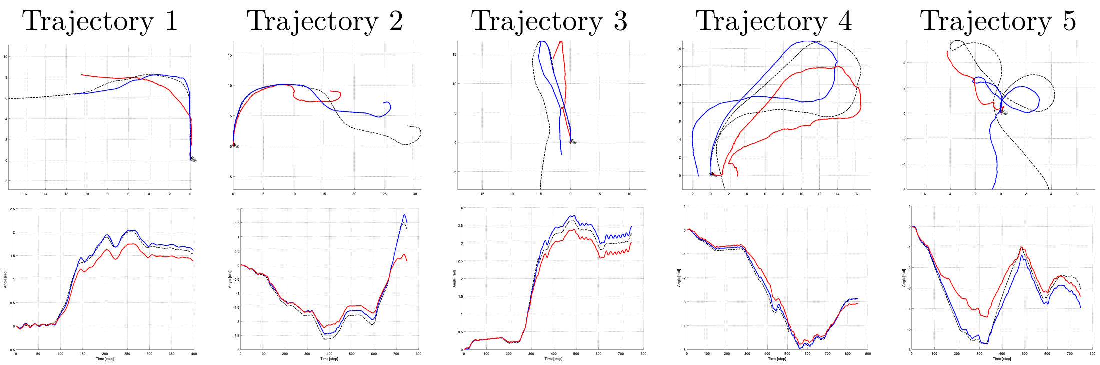


Figure 6: The upper row shows the *xy* trajectory results of the experimentation and the lower row shows the orientation results for the five selected trajectories. The blue and red line are the estimated data with the omnidirectional camera and the conventional one, respectively. The black dashed line is the real data given by the odometry.

a conventional one. The ground truth of the robot trajectory is given by an improved odometry. To better analyze the behavior of these two approaches we have split the whole trajectory in partial trajectories wich result hard to follow for the conventional SLAM. We choose five trajectories (Fig. 3) to perform the comparison. All of these trajectories have been acquired outdoor with natural lighting and include people and vehicles moving. These trajectories are sorted in increasing difficult order. The first trajectory includes a simple right turn while the last trajectory is compound by many turns.

The resolution of the images is 640x640 pixels for the omnidirectional camera and 320x240 pixels for the conventional one. The values of the angular resolution of both cameras (Fig. 4) are 235.05 $\frac{pixels}{rad}$ and 218.26 $\frac{pixels}{rad}$ for the conventional camera in horizontal and vertical direction respectively, and a mean of 232 $\frac{pixels}{rad}$ in tangential direction and 188.25 $\frac{pixels}{rad}$ in radial direction for the omnidirectional camera. We consider both vision sytems equivalent in angular resolution.

The calibration parameters for both systems is provided by The Rawseeds Project. For the conventional camera we add the Spherical Camera Model parameter $\xi = 0$ to the original calibration. For the hyper-catadioptric system we have performed a new calibration [Bastanlar et al., 2008] for better results.

Initially we tested the effect of the framerate for both visual systems. An example of the estimated trajectories are shown in Fig. 5 for the omnidirectional camera and for the conventional one. The Mean Absolute Error (MAE) of the orientation for each experiment is shown in Table 1. The best results for both systems are the ones obtained with the maximum framerate. From now we use a framerate of 15 FPS for omnidirectional camera and 30 FPS for conventional camera in order to obtain the best results.

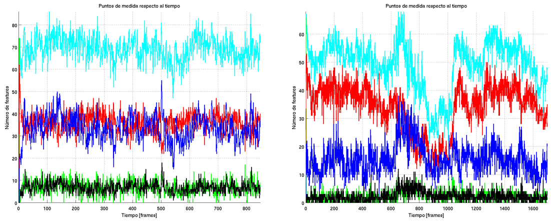
### 4.1 Trajectories comparison

Using monocular vision we cannot estimate the real scale of the 3D points. Therefore, we decide to use the orientation angle to compare the results since camera translation and 3D coordinates of points are affected by scale. The estimated results are compared to those given by the odometry. This scale problem has been solved in the literature using stereo vision or using additional sensors. In order to plot the real and the estimated trajectories in just one figure we compute a scale coefficient for each experiment. In Fig. 6 we show the trajectory results and the orientation results for each tested trajectory.

At first sight the results corresponding to the omnidirectional camera are better than

Table 2: Mean of the orientation error in radians

| Trajectory | Omnidirectional camera | Conventional camera |
|---|---|---|
| 1 | 0.04 | 0.16 |
| 2 | 0.12 | 0.26 |
| 3 | 0.11 | 0.15 |
| 4 | 0.07 | 0.22 |
| 5 | 0.29 | 0.62 |



(a) (b)

Figure 7: Features history for the trajectory 4. The cyan line is the number of active points in the state vector. The red line is the number of features matched and the blue one the non-matched features. The green line is the number of initialized features and the black line the number of removed features in each frame. (a) using omnidirectional camera, and (b) using conventional camera.

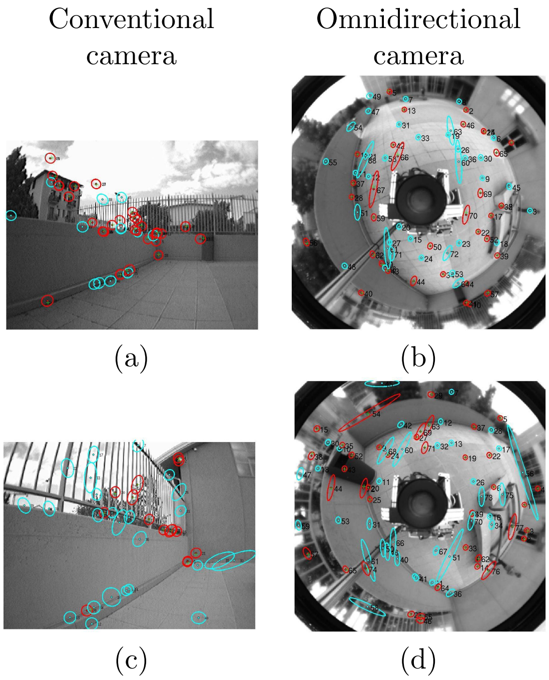
the conventional camera results. Every estimated trajectory for the omnidirectional camera follows the real trajectory and takes all the turns. The estimation follows the real movement, but it does not fit the real dimension of the real trajectory. That is clear for the fifth section, Fig. 6(e), where the omnidirectional estimated trajectory takes all the turns of the real trajectory, but the distances are not the same. The conventional camera estimation for this case does not follow the real trajectory, however the estimation of the orientation (Fig. 6(e)) is similar to the omnidirectional one.

For the third trajectory results (Fig. 6(c)) the scale coefficient is not the same through the whole trajectory. The estimated trajectory with the omnidirectional camera goes up from the start point following the real trajectory, but after the $180^o$ turn the estimation travels a shorter distance than the real trajectory. The scale coefficient has been computed for the upper point of the trajectory, and changes after the turn.

The values of the mean absolute error for the orientation in every trajectory are shown in the Table 2. The omnidirectional camera gives much better orientation estimation than using conventional camera, even using double number of frames per second with the conventional camera.

## 4.2 Influence of FOV in feature history

To study the advantages of the FOV of the omnidirectional cameras over the conventional cameras we analyze the features history of the SLAM results. Fig. 7 shows the features history over the time corresponding to the trajectory number 4. In this figure we can observe how the number of active features decreases significantly between the steps 600 and 800. The number of features removed from the state vector in this interval increases. Our SLAM approach removes features after 4 consecutive steps of not being matched. If we look at the omnidirectional camera feature history we observe that in the omnidirectional correspondent interval, 300 to 400 frame steps, the number of active features does not



Conventional camera

Omnidirectional camera

(a) (b)

(c) (d)

Figure 8: Frames of interest of the fourth trajectory execution. The cyan ellipses correspond to the confidence zone of not matched features, and the red ones correspond to the matched features. (a) and (b) correspond to the initial image of the analyzed zone for the conventional and the omnidirectional system, respectively. The (c) and (d) images are the last frames for this analyzed zone for both systems.

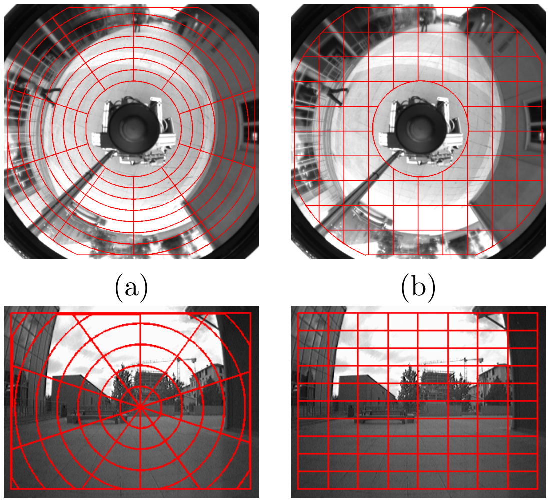
Table 3: Mean life of the features in seconds

| Trajectory | 1 | 2 | 3 | 4 | 5 | Mean |
|---|---|---|---|---|---|---|
| Omnidirectional camera/Polar buck. | 0.69 | 0.70 | 0.83 | 0.74 | 0.80 | 0.75 |
| Omnidirectional camera/Rect. buck. | 0.67 | 0.72 | 0.85 | 0.72 | 0.77 | 0.75 |
| Conventional camera/Polar buck. | 1.10 | 0.84 | 0.96 | 0.78 | 0.60 | 0.86 |
| Conventional camera/Rect. buck. | 0.69 | 0.52 | 0.60 | 0.55 | 0.48 | 0.57 |

decrease. To better analyze what is happening here we study the particular images (Fig. 8). In the conventional images we see how the camera approximates to an homogeneous wall that hides the further houses and trees. The visible floor and sky is also homogeneous. All the feature points are similar or not visible behind the wall, moreover the homogeneous scene makes hard to initialize new ones. The omnidirectional images do not suffer of this problem. Though it also gets closer to the wall it keeps track of features behind the robot.

The bucketing is used to initialize features all over the image, avoiding the problem of having many features in a small area of the image. We test two bucketing alternatives for each vision system, a polar bucketing and a rectangular one. In Fig. 9 we show the two bucketing alternatives tested for the omnidirectional and the conventional images. In Table 3 we show the mean of the life time of features in seconds for each trajectory using both bucketing systems.

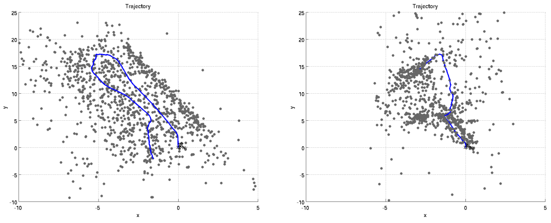
It has been argued that omnidirectional vision can maintain tracked points for a longer time due to its wide FOV. This happens with the rectangular bucketing where the mean of life time of the features is bigger for the omnidirectional camera than for the conventional one. However with the polar bucketing the mean of the life for the conventional camera is bigger than for the omnidirectional system. The reason is that with a circular bucketing



(a) (b)

(c) (d)

Figure 9: Bucketing systems used for the omnidirectional and the conventional camera. Polar, (a), and rectangular, (b), bucketing for the omnidirectional camera. For the conventional images (c) and (d) shows the polar and rectangular bucketing, respectively.



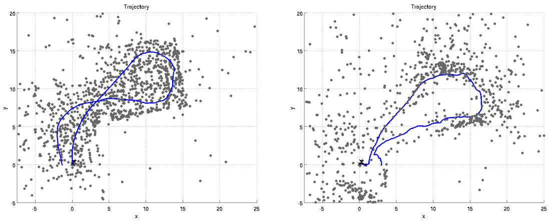
(a) (b)

Figure 10: 3D reconstruction along trajectory 3. The blue line is the estimated trajectory and the gray points are the cenital view of the estimated 3D points. (a) using omnidirectional camera, and (b) using conventional camera.

for conventional vision there are a lot of features in the center of the image which remains visible while the camera moves straight in the direction of the focal axis. In the fifth trajectory the camera takes many turns so it does not move in the direction of the focal axis which explains the differences. This makes hard to the conventional camera to track the features, while the omnidirectional camera can keep track of the features all over the scene. However we have tested that a bigger mean life of the features does not make a better SLAM solution.

### 4.3 Reconstruction of the scene

We do not have ground truth for the 3D structure of the scene. To illustrate these results in Fig. 10 and Fig. 11 we show the cenital view of the reconstruction of the scenes for the trajectories 3 and 4, respectively. The features showed are those with a life bigger than 20 frames. In Fig. 10(a) we can observe that the features form a straight line parallel to the trajectory in its right. This line corresponds to a wall present in the real scene. The conventional camera features reconstruction (Fig. 10(b)) does not form that wall so clearly. Similar differences can be observed for the fourth trajectory, Fig. 11, where the 3D reconstruction using the omnidirectional camera is much better than the other.



(a) (b)

Figure 11: 3D reconstruction along trajectory 4. The blue line is the estimated trajectory and the gray points are cenital view of the the estimated 3D points. (a) using omnidirectional camera, and (b) using conventional camera.

Table 4: Mean of the absolute error of the estimation of the vertical coordinate $z$ in meters

| Trajectory | Omnidirectional camera | Conventional camera |
|---|---|---|
| 1 | 0.18 | 0.19 |
| 2 | 0.26 | 0.34 |
| 3 | 0.18 | 0.44 |
| 4 | 0.11 | 0.34 |
| 5 | 0.10 | 0.38 |

### 4.4 Analysis of the vertical coordinate drift

Our SLAM approach estimates the 3D motion of the camera, but the camera moves on a planar ground, where the vertical coordinate should be zero. Both cameras have been calibrated by the Rawseeds Project to be aligned with the acquisition platform axis. The MAE of the estimation of $z$ coordinate of the camera motion is shown in Table 4. In all the experiments the estimation using the omnidirectional camera is much better than using the conventional one.

## 5 Conclusion

In this work we have compared a visual SLAM system using omnidirectional and conventional cameras. To do this we have formulated the Spherical Camera Model direct and inverse projection Jacobians needed by the EKF algorithm. We use a SIFT detector for the matching of the image points. We have tested different image sequences of a hyper-catadioptric system and a conventional perspective camera. The comparison of the results obtained in this experimentation has shown the superiority of the omnidirectional systems in SLAM. The estimations of trajectory and orientation are better with the omnidirectional camera than with the conventional one, and the 3D reconstruction of the scene obtained with the omnidirectional system is also clearly better than the obtained with conventional vision.

## Acknowledgment

This work has been supported by the project DPI2009-14664-C02-01. Thanks to the I3A Fellowship Program. We thank J. Civera and J.M. Montiel for their basic SLAM tool and fruitful discussions.

# References

[Barreto & Araujo, 2001] Barreto, J. P. & Araujo, H. (2001). Issues on the geometry of central catadioptric image formation. In *In CVPR* (pp. 422–427).

[Bastanlar et al., 2008] Bastanlar, Y., Puig, L., Sturm, P., Guerrero, J. J., & Barreto, J. (2008). Dlt-like calibration of central catadoptric cameras. In *OMNIVIS: Workshop on Omnidirectional Vision, Camera Networks and Non-Classical Cameras.*

[Chahl & Srinivasan, 2000] Chahl, J. S. & Srinivasan, M. V. (2000). A complete panoramic vision system, incorporating imaging, ranging, and three dimensional navigation. In *OMNIVIS '00* (pp. 104).: IEEE Computer Society.

[Civera et al., 2008] Civera, J., Davison, A. J., & Montiel, J. M. M. (2008). Inverse depth parametrization for monocular slam. *IEEE Transactions on Robotics*, 24(5), 932–945.

[Davison et al., 2007] Davison, A. J., Reid, I. D., Molton, N. D., & Stasse, O. (2007). Monoslam: Real-time single camera slam. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(6), 1052–1067.

[Gaspar et al., 2000] Gaspar, J., Winters, N., & Santos-Victor, J. (2000). Vision-based navigation and environmental representations with an omnidirectional camera. *IEEE Transactions on Robotics and Automation*, 16, 890–898.

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

[Huang & Song, 2008] Huang, F.-S. & Song, K.-T. (2008). Vision slam using omnidirectional visual scan matching. In *International Conference on Intelligent Robots and Systems, 2008* (pp. 1588–1593).

[Kang, 2000] Kang, S. B. (2000). Catadioptric self-calibration. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, volume 1 (pp. 201–207).

[Kim & Chung, 2003] Kim, J.-H. & Chung, M. J. (2003). Slam with omni-directional stereo vision sensor. In *International Conference on Intelligent Robots and Systems, 2003*, volume 1 (pp. 442–447).

[Lhuillier, 2007] Lhuillier, M. (2007). Toward flexible 3d modeling using a catadioptric camera. In *Computer Vision and Pattern Recognition* (pp. 1–8).

[Lowe, 2004] Lowe, D. G. (2004). Distinctive image features from scale-invariant keypoints. *Int. J. Comput. Vision.*

[Nagahara et al., 2006] Nagahara, H., Yagi, Y., & Yachida, M. (2006). Super wide field of view head mounted display using catadioptrical optics. *Presence: Teleoper. Virtual Environ.*, 15(5), 588–598.

[Pupilli & Calway, 2006] Pupilli, M. & Calway, A. (2006). Real-time visual slam with resilience to erratic motion. In *Computer Vision and Pattern Recognition*, volume 1 (pp. 1244–1249).

[Rituerto et al., 2010] Rituerto, A., Puig, L., & Guerrero, J. J. (2010). Visual slam with an omnidirectional camera. In *IEEE Conference on Computer Vision and Pattern Recognition.*

[Scaramuzza et al., 2006] Scaramuzza, D., Martinelli, A., & Siegwart, R. (2006). A toolbox for easily calibrating omnidirectional cameras. In *IROS*.

[Scotti et al., 2005] Scotti, G., Marcenaro, L., Coelho, C., Selvaggi, F., & Regazzoni, C. (2005). Dual camera intelligent sensor for high definition 360 degrees surveillance. *Vision, Image and Signal Processing*, 152(2), 250–257.

[Svoboda & Pajdla, 2002] Svoboda, T. & Pajdla, T. (2002). Epipolar geometry for central catadioptric cameras. *Int. J. Comput. Vision*, 49(1), 23–37.

[Thrun et al., 2005] Thrun, S., Burgard, W., & Fox, D. (2005). *Probabilistic Robotics (Intelligent Robotics and Autonomous Agents)*. The MIT Press.

[Toepfer & Ehlgen, 2007] Toepfer, C. & Ehlgen, T. (2007). A unifying omnidirectional camera model and its applications. In *Proceedings of the 11th International Conference on Computer Vision, ICCV 2007.* (pp. 1–5).