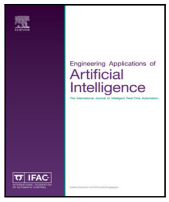
Engineering Applications of Artificial Intelligence 107 (2022) 104539

Contents lists available at ScienceDirect

Engineering Applications of Artificial Intelligence

journal homepage: www.elsevier.com/locate/engappai



Efficient probability-oriented feature matching using wide field-of-view imaging

María Flores [a,*], David Valiente [b], Arturo Gil [a], Oscar Reinoso [a], Luis Payá [a]

[a] Department of Systems Engineering and Automation, Miguel Hernandez University, Avenida de la Universidad, s/n, Elche, 03202, Spain
[b] Communications Engineering Department, Miguel Hernandez University, Avenida de la Universidad, s/n, Elche, 03202, Spain

ARTICLE INFO

Keywords:
Feature matching
Dynamic visual model
Adaptive probability-oriented feature matching
Fisheye lenses
Omnidirectional images
Visual localization

ABSTRACT

Feature matching is a key technique for a wide variety of computer vision and image processing applications such as visual localization. It permits finding correspondences of significant points within the environment that eventually determine the localization of a mobile agent. In this context, this work evaluates an Adaptive Probability-Oriented Feature Matching (APOFM) method that dynamically models the visual knowledge of the environment in terms of the probability of existence of features. Several improvements are proposed to achieve a more robust matching in a visual odometry framework: a study on the classification of the matching candidates, enhanced by a nearest neighbour search policy; a dynamic weighted matching that exploits the probability of feature existence in order to tune the matching thresholds; and an automatic false positive detector. Additionally, a comparison of performance is carried out, considering a publicly available dataset composed of two kinds of wide field-of-view images: catadioptric and fisheye. Overall, the results validate the appropriateness of these contributions, which outperform other well-recognized implementations within this framework, such as the standard visual odometry, a visual odometry method based on RANSAC, as well as the basic APOFM. The analysis shows that fisheye images provide more visual information of the scene, with more feature candidates. Contrarily, omnidirectional images produce fewer feature candidates, but with higher ratios of feature acceptance. Finally, it is concluded that improved precision is obtained when the location problem is solved by this method.

1. Introduction

In recent years, the creation of visual models of environments has received a great attention by the scientific community, due to the numerous applications it has in a variety of areas such as in mobile robotics (Harapanahalli et al., 2019; Patruno et al., 2020; Taheri and Xia, 2021; Kostavelis et al., 2016). When a robot has to operate in an 'a priori' unknown scenario (Alatise and Hancke, 2020), modelling efficiently this environment is a crucial requisite. Nowadays, vision systems sustained by computer vision and image processing techniques are widely acknowledged to this purpose. In particular, feature matching (Jiang et al., 2013; Liu et al., 2021) permits finding, modelling and tracking relevant visual information from the environment. Once the previous task is achieved, the mobile robot will be able to solve the mapping and localization problems with robustness (Hou et al., 2020).

The present work continues the research line started in Valiente et al. (2018), where the Adaptive Probability-Oriented Feature Matching (APOFM) technique is proposed to obtain a robust local feature correspondence search in presence of outliers. This method comprises a feedback loop that accounts for the existence of previous matches in the 3D space. Such information corresponds to a 3D probability distribution provided by a Gaussian Process (GP). Finally, once the local feature points are detected in the next iteration, the 3D probability distribution of features existence aids in the selection of candidate points for the definitive matching.

The APOFM can be used in many applications where feature matching is needed (e.g. object tracking Xiao et al., 2012, detection Jakubović and Velagić, 2018, mapping Zivkovic et al., 2005 and localization Wu et al., 2011 of mobile robots). Among them, we have focused on the localization problem. Sometimes the presence of dynamic elements can cause errors in the pose estimation and a robust matching framework is required. Thence, considering the benefits of the previous method in that context, its implementation in a visual odometry algorithm can improve the solution to this problem.

This work presents several improvements to the APOFM which provide a more precise localization estimation comparing to the basic APOFM (Valiente et al., 2018). The main contributions of this work are fourfold:

* Corresponding author.
E-mail addresses: m.flores@umh.es (M. Flores), dvaliente@umh.es (D. Valiente), arturo.gil@umh.es (A. Gil), o.reinoso@umh.es (O. Reinoso), lpaya@umh.es (L. Payá).

https://doi.org/10.1016/j.engappai.2021.104539
Received 10 March 2021; Received in revised form 29 October 2021; Accepted 31 October 2021
Available online 15 November 2021

(a) The matching candidates selection has been improved by means of a k-nearest neighbour classifier based on different distance metrics.
(b) The spatial probability distribution is used to perform a weighted and dynamic search of feature correspondences, under static and adaptive thresholds.
(c) An automatic false positive detector is implemented, based on the distance between pixel points and their 3D projection.
(d) An extended comparison of the efficiency of the proposal is performed, using not only a catadioptric vision system but also a fisheye one. To that end, two open-source and publicly available image datasets (Robotics and Perception Group, University of Zurich, Switzerland, 2013) have been used to benchmark the proposal with other well-acknowledged implementations such as a Standard Method (SM) (Hartley and Zisserman, 2003), SM using RANdom SAmple Consensus (RANSAC) (Nister, 2003; Scaramuzza, 2011), as well as the basic APOFM (Valiente et al., 2018).

The remainder of this paper is structured as follows. Section 2 presents an overview of related works. In Section 3, the two types of vision system and the camera model are described. The method to recover the relative pose from a pair of images is outlined in Section 4, whereas Section 5 presents how this method concretizes based on the vehicle model. In Section 6, all the steps of the improved APOFM are explained. Finally, the results achieved during the experiments are shown in Section 7. Section 8 presents the conclusions of this work.

2. Related work

Modelling the environment consists in creating a representation. Three main approaches can be found in the related literature: topological (Cebollada et al., 2019; Román et al., 2020), metric (Andert and Goormann, 2007; Liu et al., 2020) and hybrid (Yuan et al., 2018). One of the most usual representation is the occupancy grip map (Gil et al., 2015), which discretizes the environment into cells to define free or occupied (presence of an obstacle) regions. However, the classical occupancy grid approaches have some limitations such as the fact that the structural correlations between points on the map are not considered. For this reason, new techniques, such as Gaussian Process (GP) (Rasmussen and Williams, 2006), have been applied to overcome them. This learning method is a Bayesian nonparametric approach designed to solve regression and probabilistic classification problems. GP is a powerful tool to accurately identify a complex mathematical model from experimental data. Among the variety of suitable properties of GP, its main advantage is that it deals with the noise in the system, as well as with the uncertainty in the model. In O'Callaghan and Ramos (2012), the authors present an algorithm that creates a continuous occupancy representation of the environment by GP, denominated Gaussian Process Occupancy Mapping (GPOM). Ghaffari et al. (2017) extend this algorithm to create a semantic map. To this purpose, they formulate the semantic mapping as a multi-class classification problem instead of a binary classification. The GP technique is not only applied to build a model of the environment. It has recently become popular in the research community since it can be used to solve a wide range of problems in the field of robotics (Song et al., 2018; Polymenakos et al., 2020; Sun et al., 2018; Park et al., 2018; Dalla Libera et al., 2019; Nutalapati et al., 2019; Li et al., 2020). For example, Nguyen et al. (2019) employ the GP to infer remaining wall thickness at unseen pipe sections for a mobile robot which moves inside a pipeline with the objective of inspecting at the location of a break.

Both the mapping and the localization tasks can be carried out as long as the mobile robot acquires information from its environment. To this purpose, many types of sensors (e.g. sonar, lidar, encoders, global position system) can be mounted on the mobile robot. Among them, vision sensors have become a source of countless research contributions in recent years due to the several attractive features that they present, such as the richness of information captured, low weight, power consumption, size, and cost (Reinoso and Payá, 2020). Cameras are versatile since they can be utilized for navigation both in outdoor and indoor environments. Nevertheless, the most interesting advantage is the amount of information from the environment that an image contains, such as colour, luminance, shape and texture. The use of these sensors increases the scope of applications of mobile robots. The type of information provided by them not only permits solving the localization and mapping problems, but it can also be used for other tasks, for instance, road detection (Zhang et al., 2018), traffic sign recognition (Jung et al., 2016), and obstacle identification (Emani et al., 2019). The amount of information available in an image is related to the field of view of the camera that captured it. The wider the field of view is, the higher the amount of information from the environment. According to this, this type of vision systems can be classified, in broad lines, into conventional monocular or omnidirectional cameras.

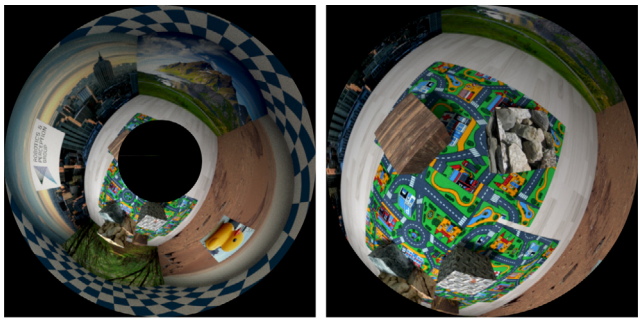
Comparing to conventional monocular, omnidirectional vision systems have more advantages thanks to their wide field of view. A single image captured by this type of camera can provide a 360° view from the environment around the mobile robot (Amorós et al., 2020). Therefore, omnidirectional cameras permit obtaining exhaustive models of the environment with a reduced number of views (Payá et al., 2017). There are different alternatives to get an omnidirectional vision system (Scaramuzza, 2014; Li, 2006). The most extended ones are dioptric and catadioptric systems. These are the configurations used in the present work. The first one consists in combining a conventional camera with a shaped wide-angle lens (such as fisheye). This vision system provides a hemispherical view, so a pair of cameras pointing to opposite sides is required to acquire a full spherical view (Gao and Shen, 2017). The second way to create an omnidirectional system is the combination of a spherical (Barone et al., 2018), conic (Marcato Junior et al., 2016), hyperbolic (Boutteau et al., 2010), parabolic, or elliptic mirror and a pinhole camera.

For some applications (e.g. autonomous aerial robots), the fisheye cameras are better than the catadioptric ones since they achieve an omnidirectional coverage with lower weight (Gao et al., 2020). Nowadays, the automotive industry is very interested in providing vision perception to the drivers, concretely a 360° view around the vehicle. To that end, the vehicles are equipped with four fisheye cameras which are placed in a way that the coverage is optimized. In this application, the coverage obtained using a catadioptric vision system is less effective since the majority of information captured in the image will be sky and body car. For instance, Lee et al. (2013) have mounted four fisheye cameras (looking front, rear, left and right) on a vehicle to implement their structureless pose-graph loop-closure algorithm.

The formulation of the localization problem typically depends on the type of sensor used. It can be classified into global (i.e. global position system) or local (i.e. wheel, inertial, laser, radar, or vision systems mounted onboard) localization. In Mohamed et al. (2019), the authors provide a general overview of the state-of-the-art about the localization methods using these latter sensors.

In the case of onboard vision systems, the technique to solve the local localization problem is also known as visual odometry, which incrementally estimates the motion of an agent. The difference is that the vision-based odometry obtains the relative pose through the changes that the movement induces in the images (Fraundorfer and Scaramuzza, 2012). This way, this method overcomes the main limitations of the wheel odometry (such as wheel slippage and uneven terrain). Besides, comparing with other traditional approaches (i.e. GPS, inertial, laser and radar), visual odometry is an inexpensive and relatively accurate alternative technique that can be employed both in outdoor and indoor environments, and its use is not only limited to ground vehicles.

Depending on the process chosen to extract the information from the images, the different methods of visual odometry can be classified into feature-based, appearance-based, or hybrid approaches (Poddar et al., 2018). In Valiente García et al. (2012), a comparison between both appearance- and feature-based visual odometry methods is carried out, using omnidirectional images.



(a) (b)

Fig. 1. Two examples of wide field of view images which are extracted in the same indoor scenario (Robotics and Perception Group, University of Zurich, Switzerland, 2013). (a) An image from a catadioptric system composed of a hyperbolic mirror and a camera. (b) An image from a system composed of a camera and a fisheye lens.

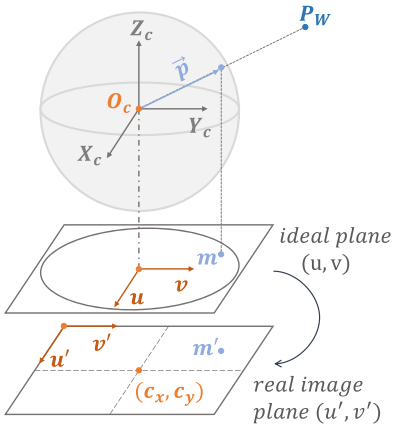


Fig. 2. A scene point P_W is projected onto the unit sphere surface. This way, the 3D unit vector $\vec{p}$ is obtained. Then, it is mapped to a point $m = [u, v]$ on the ideal plane through a function. This ideal plane point is transformed to a point $m' = [u', v']$ in the real image plane (pixel coordinates) by an affine transformation.

3. Catadioptric and fisheye vision systems

In this work, two different visions systems are used: one catadioptric system (omnidirectional camera) and one camera with a fisheye lens. Fig. 1 shows two images of the same scene captured by each of these systems.

The mathematical model of a catadioptric or fisheye camera is more complex than a standard perspective camera. The lens causes refraction, and the mirror produces reflection, so the model should take these effects into account. There are many works in the literature to estimate the model of an omnidirectional camera. The first unified model for central catadioptric systems, that is, cameras using a parabolic, hyperbolic, or elliptical mirror, was proposed by Geyer and Daniilidis (2000). They determine that this type of camera can be modelled by a projection of the 3D scene point onto a unit sphere centred in the effective viewpoint, followed by a perspective projection onto a plane. This model was developed specifically for central catadioptric cameras, so it is not valid for fisheye cameras. Ying and Hu (2004) presented an extension of this model that can be used to model fisheye cameras as well. With respect to Scaramuzza et al. (2006a) all central catadioptric cameras can be represented through an exact parametric function. Still, the projective model depends on the lens field-of-view and varies from camera to camera in the case of the fisheye lenses. Therefore, the approximation of Ying and Hu (2004) for a fisheye camera through a catadioptric one, only works with limited accuracy (Siegwart et al., 2011). To overcome this problem, Scaramuzza et al. (2006a, 2006b) proposed a new unified model for catadioptric and fisheye cameras. In this case, the authors use a Taylor polynomial, whose coefficients and degree are found through a calibration phase.

As mentioned at the beginning of this section, we use catadioptric and fisheye images, so we use this model due to its suitability for both types of cameras (Scaramuzza et al. 2006a, 2006b). Fig. 2 shows the projection following this unified model proposed in (Scaramuzza et al. 2006a, 2006b). A scene point P_W, expressed in the world reference frame can be expressed in the fisheye/mirror reference frame P_C by using the extrinsic parameters. This 3D point is projected onto the unit sphere surface obtaining the unit vector $\vec{p}$ emanating from the centre of the reference frame O_c. Then, the pixel point m is obtained through an imaging function g (see Eq. (1)) and an affine transformation (Scaramuzza et al. 2006a, 2006b).

$$\lambda \cdot g(m) = \lambda \begin{bmatrix} u \\ v \\ f(u,v) \end{bmatrix} = P_c = [\mathbf{R}|\vec{t}]P_w \tag{1}$$

4. Relative pose estimation

Estimating the relative pose between two images taken from different positions is a crucial problem in visual navigation. This technique is known as Visual Odometry (Fraundorfer and Scaramuzza, 2012; Scaramuzza and Fraundorfer, 2011).

To solve the feature-based visual odometry problem, the algorithm can be principally decomposed into three different blocks: (1) feature detection and description, (2) feature matching (or tracking), and (3) motion estimation. The first step consists in identifying points of interest in the image and representing the region around each one as a compact vector, named descriptor, which is used to compare features in different images. The second step consists in detecting the pixel points corresponding to the same 3D point in the pair of images (i.e. finding the matches). Finally, the third step consists in estimating the relative camera motion between the pair of images taken at different times (Scaramuzza and Fraundorfer, 2011). Depending on the dimension of the feature correspondences, there are three techniques to carry out this last step (Yousif et al., 2015): motion estimation from 3D feature correspondences (3D to 3D); from 3D feature and 2D image feature correspondences (3D to 2D); and from 2D image feature correspondences (2D to 2D). In the last method, both feature correspondences are specified in 2D image coordinates, so the relative motion is recovered by the epipolar geometry (see Fig. 3), concretely by the essential matrix **E**. In this work, Standard Method (SM) refers to an algorithm composed only of the three blocks mentioned at the beginning of the previous paragraph, where the technique employed in the motion estimation block is the epipolar geometry (2D to 2D).

The essential matrix depends only on the camera motion parameters that can be recovered only up to a scale factor. This matrix encodes the relative motion parameters between a pair of images and, in consequence, can be defined as:

$$\mathbf{E} = [t]_x \mathbf{R} \tag{2}$$

where $\mathbf{R}$ is the rotation matrix and $[t]_x$ is the skew-symmetric matrix of the translation vector $\vec{t} = [t_x, t_y, t_z]$. After following the process described in Hartley and Zisserman (2003), the relative pose is recovered.

The relative pose can be expressed using angular parameters. Firstly, the coordinates (t_x, t_y, t_z) of the translation vector can be transformed into spherical coordinates. In other words, the relative position between the two camera poses is determined by a radial distance ρ (from the centre of the camera frame at the first pose to the camera centre at the second pose), an elevation angle β and an azimuth angle ϕ.

$$\phi = atan2(t_y, t_x) \tag{3}$$

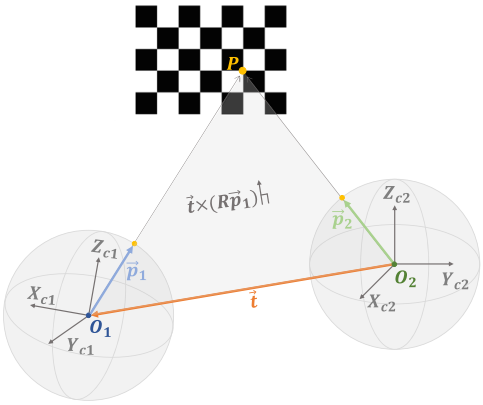


Fig. 3. Epipolar geometry of cameras with wide field of view. A scene point P is projected on the sphere surface of the first and second camera as $\vec{p}_1$ and $\vec{p}_2$, respectively. A rotation matrix and translation vector relate both camera reference systems. Therefore, $\vec{t}$, $\mathbf{R}\vec{p}_1$ and $\vec{p}_2$ lie in the epipolar plane (coplanarity condition).

$$\beta = atan2(t_z, \sqrt{t_x^2 + t_y^2}) \tag{4}$$

$$\rho = \sqrt{t_x^2 + t_y^2 + t_z^2} \tag{5}$$

Secondly, the orientation **R** can be defined by using the Euler angles: yaw (rotation θ around the Z-axis), pitch (rotation γ around the Y-axis) and roll (rotation α around the X-axis). In short, the relative pose is given by six parameters, which can be seen in Fig. 4, five of them are angles (θ, γ, α, ϕ, β) and the remaining one is a scale factor (ρ). This work focuses on the estimation of the angular parameters.

5. Relative pose estimation based on the vehicle model

Some steps of this method require that the relation between the camera frame and the world frame is well-known since the objective is to obtain a 3D model of the environment. Consequently, the mapping from pixel to world coordinates, and vice-versa, will be carried out. In this work, we try to solve the visual odometry problem for a mobile robot that navigates without knowing its following pose, therefore, the camera pose is not known. Nevertheless, assuming that the camera is on-board of a mobile robot, then an approximation of the next camera pose can be obtained by using the probabilistic odometry motion model presented by Thrun et al. (2005). Since the ground truth is available, these data can be modelled as odometry data (by adding some amount of noise), and, after that, the next pose can be estimated. The mobile robot moves from t to $t + 1$, and then the image I_{t+1} is captured. It means that the odometry information, which is usually provided by wheel sensors, is available when the image is processed. Therefore, the odometry-based motion model can be used as an estimation of the relative pose, but it is only used to map the 3D model and image points.

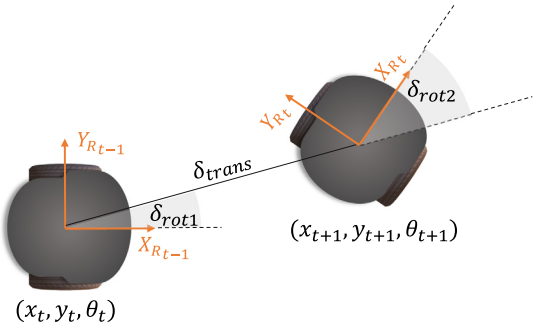


Fig. 5. Parameters of the odometry-based motion: first rotation δ_{rot1}, translation δ_{trans} and second rotation δ_{rot2}.

5.1. Odometry motion model

For a planar environment, the mobile robot state $\vec{x}$ is represented by a point (x, y) and a rotation angle θ that determines the orientation. The odometry-based motion model describes the movement of a mobile robot between two consecutive poses (from $\vec{x}_t = (x_t, y_t, \theta_t)$ to $\vec{x}_{t+1} = (x_{t+1}, y_{t+1}, \theta_{t+1})$) as a sequence of three steps: an initial rotation δ_{rot1}, followed by a straight line motion (translation) δ_{trans} and final rotation δ_{rot2} as illustrated in Fig. 5.

After obtaining $\vec{x}_t$ and $\vec{x}_{t+1}$ from the ground truth data, the parameters of the odometer model can be computed as:

$$\delta_{rot1} = atan2(y_{t+1} - y_t, x_{t+1} - x_t) - \theta_t \tag{6}$$

$$\delta_{trans} = \sqrt{(x_{t+1} - x_t)^2 + (y_{t+1} - y_t)^2} \tag{7}$$

$$\delta_{rot2} = \theta_{t+1} - \theta_t - \delta_{rot1} \tag{8}$$

In the ideal case, these values would be the same as the ones obtained using the odometer readings, but it does not happen in a real operation. In that case, the measurements provided by the odometer are given by the true motion with independent noises for each one of these motion parameters. The noise is modelled as a zero-mean Gaussian distribution with variance σ and it is denoted as $\epsilon(\sigma)$. Then, the measured parameters are:

$$\hat{\delta}_{rot1} = \delta_{rot1} + \epsilon(\alpha_1 \delta_{rot1} + \alpha_2 \delta_{trans}) \tag{9}$$

$$\hat{\delta}_{trans} = \delta_{trans} + \epsilon(\alpha_3 \delta_{trans} + \alpha_4 (\delta_{rot1} + \delta_{rot2})) \tag{10}$$

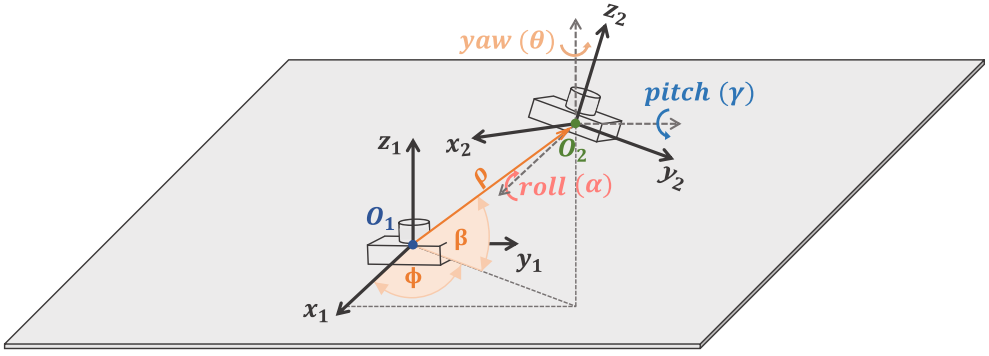


Fig. 4. The relative pose recovered from **E** can be described by six parameters. The orientation can be defined as three successive rotations: around the Z-axis $\mathbf{R}(\theta)$, Y-axis $\mathbf{R}(\gamma)$ and X-axis $\mathbf{R}(\alpha)$. The position is given by two angles (β and ϕ) and a scale factor (ρ).

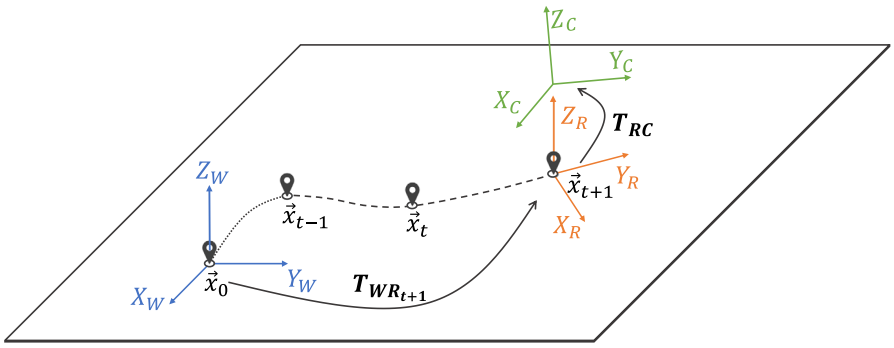


Fig. 6. Coordinate Systems: world frame, mobile robot frame and camera frame.

$$\hat{\delta}_{rot2} = \delta_{rot2} + \epsilon(\alpha_1 \delta_{rot2} + \alpha_2 \delta_{trans}) \tag{11}$$

where the α_1, α_2, α_3 and α_4 parameters model the noise caused by drifts and slipping (translation and rotation). Finally, the initial odometry can be calculated as:

$$\hat{x}_{t+1} = x_t + \hat{\delta}_{trans} \cos(\theta_t + \hat{\delta}_{rot1}) \tag{12}$$

$$\hat{y}_{t+1} = y_t + \hat{\delta}_{trans} \sin(\theta_t + \hat{\delta}_{rot1}) \tag{13}$$

$$\hat{\theta}_{t+1} = \theta_t + \hat{\delta}_{rot1} + \hat{\delta}_{rot2} \tag{14}$$

The odometry is a relative positioning technique (Aqel et al., 2016), so there is no fixed mapping between the coordinates used by the robot's internal odometry and the world coordinates. To solve it, the world reference system has been fixed in the initial state of the mobile robot $\vec{x}_0$, then the relative pose of the mobile robot at the instant $t + 1$ with respect to the world frame $\mathbf{T}_{\mathbf{WR_{t+1}}}$ is given by a rotation matrix around the Z-axis $\mathbf{R_z}(\hat{\theta}_{\mathbf{t+1}})$ and a translation in the XY plane $\vec{t} = (\hat{x}_{t+1}, \hat{y}_{t+1}, 0)$:

$$\mathbf{T}_{\mathbf{WR_{t+1}}} = \begin{bmatrix} \cos\hat{\theta}_{t+1} & -\sin\hat{\theta}_{t+1} & 0 & \hat{x}_{t+1} \\ \sin\hat{\theta}_{t+1} & \cos\hat{\theta}_{t+1} & 0 & \hat{y}_{t+1} \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{15}$$

Consequently, the relationship between the camera and world coordinate system $\mathbf{T}_{\mathbf{WC_{t+1}}}$ can be computed assuming that the position of the camera with respect to the mobile robot $\mathbf{T}_{\mathbf{RC}}$ is fixed and well-known.

$$\mathbf{T}_{\mathbf{WC_{t+1}}} = \mathbf{T}_{\mathbf{WR_{t+1}}} \cdot \mathbf{T}_{\mathbf{RC}} \tag{16}$$

In other words, $\mathbf{T}_{\mathbf{WC_{t+1}}}$ is the matrix that transforms the points from the camera frame into the world frame. Fig. 6 shows these reference systems.

6. Adaptive probability-oriented feature matching (APOFM)

This section synthesizes the basis of the APOFM (Valiente et al., 2018) and the improvements proposed in the present paper to improve its performance. In each iteration, the corresponding points (matches) between the images are obtained. A pair of feature points (m_1 and m_2) are considered matched points if their feature descriptors are similar. Therefore, this means that these feature points are the projection of the same 3D scene point. Consequently, if this point appears projected on the next images, and providing it continues to be considered a matching in other iterations, it presents a high probability in the model. Hence, the associated probability with each point is updated at every iteration. Fig. 7 shows the block diagram with the most representative steps of this process.

The model is obtained by using the GP (Rasmussen and Williams, 2006) that is defined as a collection of random variables, a finite number of which have a joint Gaussian distribution, whose input is a set of 3D points (Section 6.2). Hence, it is necessary to recover, previously, the 3D coordinates of each pair of correspondences (Section 6.1). This problem is known as triangulation (Hartley and Sturm, 1997). After that, the environment model is updated using the matches between the previous and current image, and the pose of the next image with respect to the current one is calculated.

The problem of visual odometry is solved following the algorithm described in Section 4. However, some steps have been added and modified in order to improve the matching search. For instance, some new steps have been inserted between the feature detection and feature matching search. Now the search of the feature matchings is not performed with all the feature points detected in the image corresponding to the next pose. The search is only focused on these points considered as candidates. In broad lines, after detecting the feature points in the image taken at the next pose I_{t+1}, using SURF (Bay et al., 2006), the coordinates of the output of the GP are expressed into the frame of the camera at the next time instant $t + 1$. To do it, we use the transformation between world and camera frame calculated in Section 5.1 by the odometry motion model. Next, these points are projected on the image using the calibration parameters (Section 6.3). The next step consists in determining how many detected SURF points are candidates, based on their proximity to a projected probability point (Section 6.4). After that, the search for matches can be carried out (Section 6.5).

In the first iteration, that is, to estimate the relative pose using the images I_0 and I_1, the method employed is SM since there is no information about matching features, that is, all SURF points of I_1 have the same probability of finding a correspondence in I_0. After that, the triangulation problem is solved with the matched features of this iteration, and these 3D points are the input to the GP. This way, the scene model is available from the second iteration, and the proposed model can be employed from then on.

6.1. Triangulation and false positive record

As already mentioned, the triangulation problem essentially consists in calculating the position of a point in the space, given its projection on at least two views, and the calibration parameters and pose estimation. The basic method to solve this problem is to find the intersection of the lines of sight whose origins are the camera centres (O_1 and O_2) and their direction vectors are given by the projections of the image points on the unit surface sphere ($\vec{p}_1$ and $\vec{p}_2$). To recover the coordinates of the 3D point in the world frame, the centres of the camera and the direction vectors must be expressed in the world reference system. The necessary information to do it can be extracted from the estimated transformation matrix that has been calculated using Eq. (16).

However, the rays may not intersect in the 3D space as a consequence of the presence of noise in the matching of image points. This noise can be produced by lens distortion or errors in the calibration parameters. The first one affects the 3D to 2D mapping, whereas the

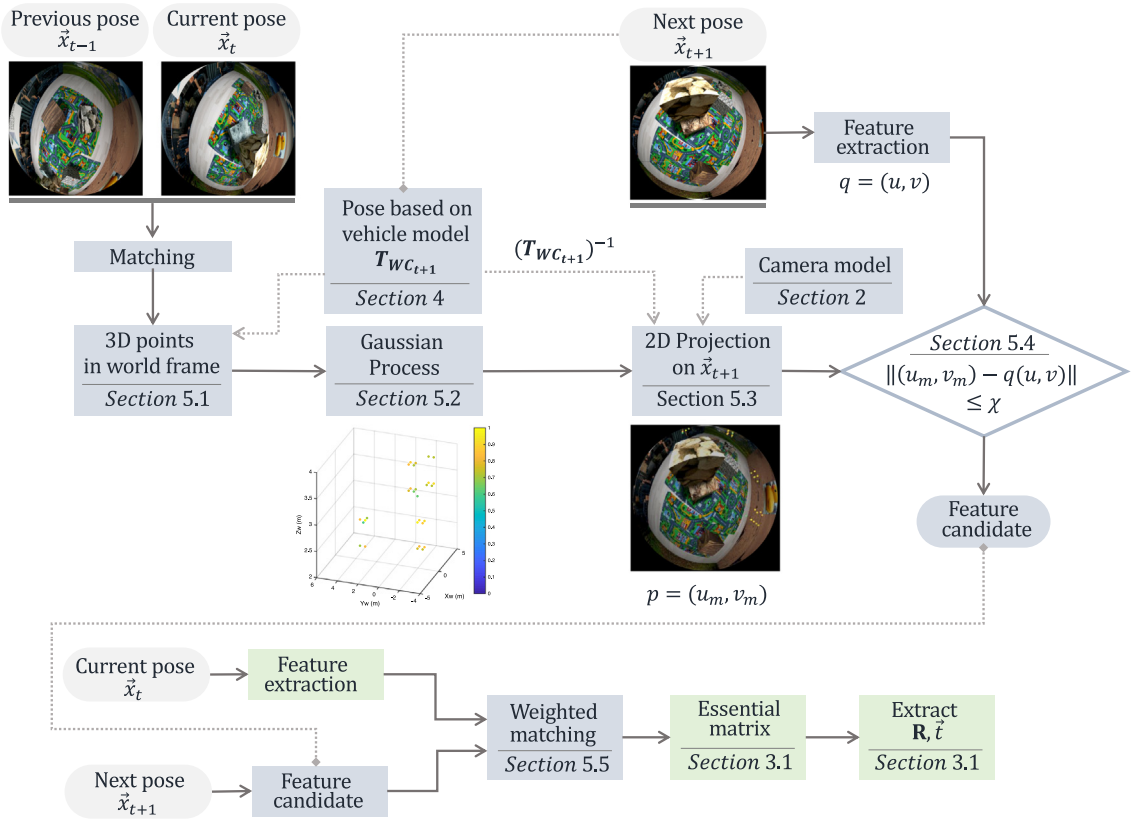


Fig. 7. Block diagram that shows the main parts of the algorithm to create a model of the environment, detailing the sections of the paper in which each part of the algorithm is presented.

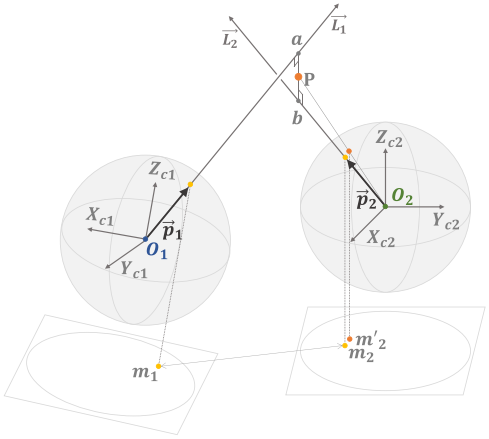


Fig. 8. Triangulation problem: the mid-point approach to recover the 3D coordinates of a point using its projection on a pair of images.

second one makes the 2D to 3D mapping be not precise since the camera model is used in this step. Moreover, some noise may appear due to image processing, such as interest points detection error or the presence of outliers in the correspondence detection. As a consequence, the solution to the triangulation problem becomes nontrivial. In the literature, there are different methods to find the best solution; some of them are described in Nair and Nair (2020). In this work, we adopt the mid-point method proposed in Beardsley et al. (1994). Thus the 3D scene point, is approximated by the midpoint of the segment which is perpendicular to both rays with the shortest distance.

Fig. 8 shows that for the first camera there is a ray defined by the origin O_1 and the direction vector $\vec{p}_1$, so its corresponding equation is $\vec{L}_1 = O_1 + \lambda_1 \cdot \vec{p}_1$. Similarly, there is a ray equation, whose equation is $\vec{L}_2 = O_2 + \lambda_2 \cdot \vec{p}_2$, for the second camera. The first step for the computation of the intersection point is to obtain the points $\vec{a}$ and $\vec{b}$. These points are the intersection of the common perpendicular $\vec{ab}$ with the line $\vec{L}_1$ and $\vec{L}_2$ respectively. In other words, the point $\vec{a}$ satisfies the equation of the first ray, so $\vec{a} = O_1 + \lambda_1 \cdot \vec{p}_1$, and the point $\vec{b}$ satisfies the equation of the second ray, hence $\vec{b} = O_2 + \lambda_2 \cdot \vec{p}_2$.

Due to the fact that the segment ab is perpendicular to both rays, the dot product of its direction vector, and the corresponding of each ray is equal to zero.

$$(\vec{b} - \vec{a}) \cdot \vec{p}_1 = (O_2 - O_1) \cdot \vec{p}_1 + \lambda_2 \cdot \vec{p}_2 \cdot \vec{p}_1 - \lambda_1 \cdot \vec{p}_1 \cdot \vec{p}_1 = 0 \tag{17}$$

$$(\vec{b} - \vec{a}) \cdot \vec{p}_2 = (O_2 - O_1) \cdot \vec{p}_2 + \lambda_2 \cdot \vec{p}_2 \cdot \vec{p}_2 - \lambda_1 \cdot \vec{p}_1 \cdot \vec{p}_2 = 0 \tag{18}$$

After solving the equation system, the unknowns λ_1 and λ_2 are obtained. The intersection point is the average of the points $\vec{a}$ and $\vec{b}$.

$$P = \frac{\vec{a} + \vec{b}}{2} = \frac{(O_1 + \lambda_1 \cdot p_1) + (O_2 + \lambda_2 \cdot p_2)}{2} \tag{19}$$

As mentioned above, sometimes the set of corresponding points may contain wrong matches, named false positives. Fig. 9 depicts this problem. The detected feature point in the first image m_1 and the detected feature point in the second image m_2 are considered as a pair of corresponding features during the matching search. Then, the triangulation problem is solved, and the result is the 3D point P. If P is re-projected on the second image m'_2, it can be observed that its projection is not near the detected feature point in the second image m_2. This means that the feature points are not the projection of the same 3D point (false positive). As a matter of fact, this can be noticed in this figure, where the true 3D point of each feature point is shown. To extend this example, Fig. 10 shows two pairs of matched features where one is a false positive and the other is a true positive.

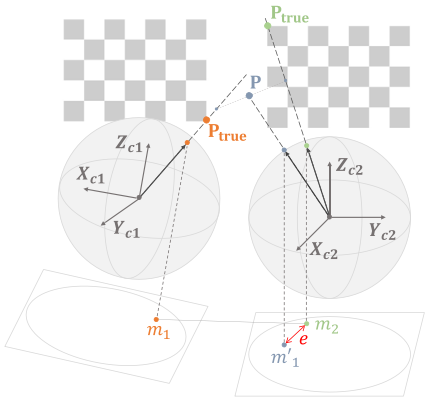


Fig. 9. The detected feature point in the first image • and the detected feature point in the second image • have been considered as a pair of corresponding features. However, it can be observed that the feature points are not the projection of the same 3D point so it is a false positive. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

It is important to note that this contribution aims to serve as a tool for quantifying false positive ratios, and thus evaluating the efficacy of the probability-based matching.

6.2. GaussIan process

Once the 3D coordinates corresponding to each matching feature have been recovered, the next step is to create the probability model with them using the GP.

A GP can be seen as a generalization of the Gaussian probability distribution to function spaces. It means that a probability distribution describes random variables, whereas a GP is a distribution over functions $f(x)$. Therefore, if a Gaussian distribution is given by its mean and covariance, then a GP is formed by a mean function $f_m(x)$ and covariance function $k(x, x')$. So, the GP can be written as:

$$f(x) \sim \mathcal{GP}(f_m(x), k(x, x')) \tag{20}$$

where $x \in \mathbb{R}^d$ and $x' \in \mathbb{R}^d$, are the training and test (query) input points respectively.

The algorithm used to obtain the probabilistic model of the environment is the one proposed by Ghaffari et al. (2018). They developed a technique for occupancy mapping using the GP. As presented in Fig. 11, the algorithm is composed of three main modules: (1) GP regression (Section 6.2.1). (2) Logistic regression classifier that squashes the output of the prior module into probabilities (Section 6.2.2) and leads to the local map. (3) Bayesian Committee Machine (BCM) (Tresp, 2000), which updates the global map incrementally. The output of this algorithm is a probability distribution which is shown in Fig. 12(a) and the modules that compose it are described in depth in the following subsections.

6.2.1. GaussIan process regression

Given a set of n training input points $X = \{x_1, x_2, \dots, x_n | x_i \in \mathbb{R}^3\}$, their corresponding output values arranged as a vector $y = \{y_1, y_2, \dots, y_n | y_i \in \mathbb{R}\}$ and a set of n_t test points $X_* = \left\{x_{*1}, x_{*2}, \dots, x_{*n_t} | x_i \in \mathbb{R}^3\right\}$. The mean and covariance of the predictive conditional distribution for test data $f_* | X, y, X_* \sim \mathcal{N}(\bar{f}_*, cov(f_*))$ can be computed as follows:

$$\bar{f}_* = K(X, X_*)^T (K(X, X) + \sigma_n^2 I)^{-1} y \tag{21}$$

$$cov(f_*) = K(X_*, X_*) - K(X, X_*)^T (K(X, X) + \sigma_n^2 I)^{-1} K(X, X_*) \tag{22}$$

where σ_n^2 is the variance of the observation noise and $\mathbf{K}(\cdot, \cdot)$ denotes the covariance matrix of the variables $(\cdot, \cdot)$, for instance, $\mathbf{K}(\mathbf{X}, \mathbf{X}_*)$ is the $n \times n_t$ matrix of the covariances evaluated at all pairs of training X and test points X_*.

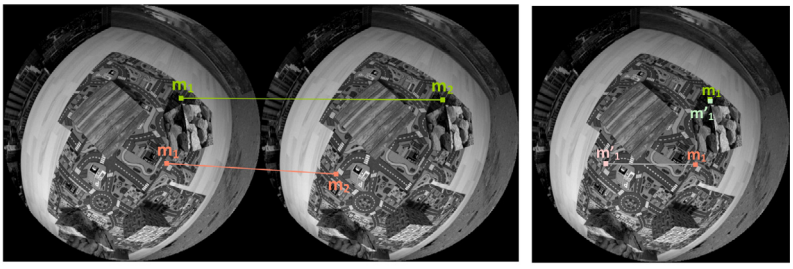
In this work, the training input data X are the 3D points P obtained after solving the triangulation problem for each pair of feature correspondences. The target value assigned to each training input point is one ($y_i = 1$) indicating that the projections of this point on the images at $t-1$ and t have been considered as a pair of matched features. Therefore, the training output data y is a vector of ones $y = \{1, 1, \dots, 1\}$. Finally, the test data are the set of spatial coordinates to build the map on. In other words, the motion space of the mobile robot with existing points is evaluated. This map consists of a three-dimensional grid represented by the vectors $X_m = \left\{x_1 : i : x_{n_x}\right\}$, $Y_m = \left\{y_1 : i : y_{n_y}\right\}$, and $Z_m = \left\{z_1 : i : z_{n_z}\right\}$ that are defined by a starting and ending value, and an increment i between their elements, which is denominated the step of the grid ($\Delta grid$). The number of test points is given by the length of these vectors so $n_t = n_x \cdot n_y \cdot n_z$.

6.2.2. Logistic regression classifier

Since the goal is to obtain a probabilistic representation of the environment, the output of the GP regression, that is the prediction (μ_*, σ_*^2) at a test point x_*, must be squashed into the range $[0, 1]$. Hence, a logistic function is used.

$$p(y_* = 1 | X, y) = \frac{1}{1 + exp(-\gamma \omega_i)} \tag{23}$$

where $\omega_i = \mu_{*i} \lambda^{1/2}$ is the weighted mean, $\lambda = \sigma_{min}^2 / \sigma_{*i}^2$ denotes the bounded information associated to each location, σ_{min} is the minimum predicted variance by the GP regression and γ is a positive constant parameter to control the sigmoid shape.



(a) (b)

Fig. 10. Detection of false positive: (a) Two pairs of features matches can be seen in this figure. Each feature is symbolized by m_i where the subscript i indicates to which image it belongs. Furthermore, each pair of matched features is represented by a different colour. (b) The calculated 3D points are projected on the first image m'_1 after solving the triangulation problem for each pair of correspondences shown in (a). It can be observed that the projected point m'_1 is nearby the feature point in the case of the green correspondence. Nevertheless, it does not happen the same for the orange matching since this is a false positive. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

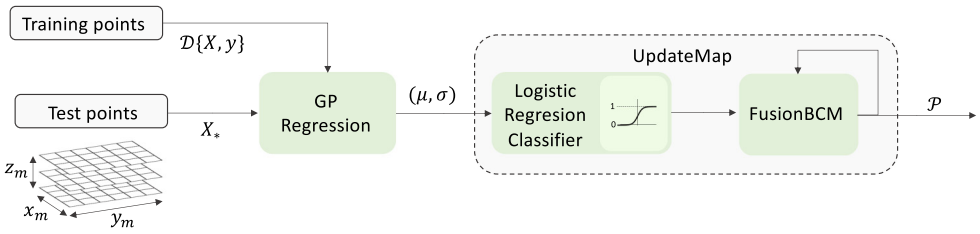


Fig. 11. Block diagram of the algorithm that calculates the probabilistic model of the environment using GP.

6.3. Projection of the model points on the 2D image

A 3D probability distribution has been obtained in the previous section. Given that it determines the probability that the projection of a 3D scene point is a feature correspondence, the output of the GP must be projected on the image at the next pose. In this manner, relevant areas are obtained over the image. If a feature point is detected in one of these areas, then it will probably be a matching feature.

The first step is to express each 3D point of the probability distribution ${}_W P = \{{}_W p_1, {}_W p_2, \dots, {}_W p_n | {}_W p_i \in \mathbf{R}^3\}$ in the camera coordinate system:

$$ {}_{C_{t+1}} p_i = \mathbf{T}_{\mathbf{C_{t+1}W}} \cdot {}_W p_i \tag{24} $$

where $\mathbf{T}_{\mathbf{C_{t+1}W}}$ is the matrix that transforms the points from world to camera frame at $t + 1$. To move the GP output to the next frame pose $t + 1$, an estimation of the relationship between the world and camera frames must be available. This estimation es calculated from the vehicle model (Section 5). In this case, we obtain the matrix that transforms the points from the camera to the world frame $\mathbf{T}_{\mathbf{WC_{t+1}}}$ using Eq. (16). Therefore, taking this into account, the previous equation can be written as:

$$ \begin{aligned} {}_{C_{t+1}} p_i = \mathbf{T}_{C_{t+1}W} \cdot {}_W p_i = \mathbf{T}^{-1}_{\mathbf{WC_{t+1}}} \cdot {}_W p_i = \\ \begin{bmatrix} \mathbf{R}^T_{\mathbf{C_{t+1}W}} & -\mathbf{R}^T_{\mathbf{C_{t+1}W}} \cdot \vec{t}_{C_{t+1}W} \end{bmatrix} \cdot {}_W p_i \end{aligned} \tag{25} $$

where $\mathbf{R}_{\mathbf{C_{t+1}}W}$ is the rotation matrix that describes the orientation of the camera frame with respect to the world frame and $\vec{t}_{C_{t+1}W}$ is the distance vector from world to camera expressed in world frame. The next step is to calculate the pixel coordinates of each point using the camera model Eq. (1).

Fig. 12(a) shows the 3D probability distribution of feature existence expressed in the world frame. Fig. 12(b) shows these same points with their associated probability in the image at $t + 1$ after performing the transformation between frames and mapping the 3D points in the camera frame to 2D image points.

6.4. Determining candidate features

The last step of this method is to determine which of the detected feature points will be considered as a possible matching candidate from the probability points projected.

The feature points will be candidates if they are near projected points with an associated probability, besides they will be assigned the probability of the nearest point. To carry out this, the Nearest Neighbour (Cover and Hart, 1967) method has been used, which calculates the distances between the test data and each of the training data in order to identify the nearest neighbour.

Given a set of training data $p_1, p_2, \dots, p_n$, and a distance function d, the nearest neighbour search permits finding the closest point in the training dataset to each query point q according to Eq. (26). In the APOFM, the training points are the projected points with an associated probability, whereas the set of query points are the feature points detected.

$$ NN(q) = \arg\min_{p_i} d(p_i, q) \tag{26} $$

There are several types of distance functions which have been used in the literature (Chomboon et al., 2015), such as Euclidean, Mahalanobis, Manhattan, Minkowsky, City-block, and Chebyshev. In this paper, two of these distance metrics have been employed. In the first place, the Mahalanobis distance, whose search of the nearest neighbour has been carried out using the exhaustive method. This search method finds the distance from each detected feature point to all n projected points with an associated probability. In the second place, the City-block distance has been employed to find the nearest neighbour using the Kd-tree algorithm (Bentley, 1975). Finally, each feature point is classified using the distance between itself and its nearest neighbour. Then, a specific threshold χ is imposed on the maximum distance for a feature point to be considered as a candidate. The value of χ is given by the chi-square inverse cumulative distribution function, with n_{dof} degrees of freedom, evaluated at a probability value. In this work, n_{dof} is equal to 2 since this is the dimension of the image points. The probability value is chosen as the one that provides the best results, according to the experiments performed in Section 7.1.

In summary, a feature point is classified as a candidate only if $d(p_i, q) < \chi$. On the contrary, the feature points which do not satisfy this requirement are classified as not candidates and are not taken into account in the next step.

Fig. 12(c) shows the projected points with an associated probability and the detected SURF feature points in the image taken at $t + 1$. After solving the classification problem, the detected SURF feature points which are classified as candidates are represented in Fig. 12(d), with a specific colour based on its probability.

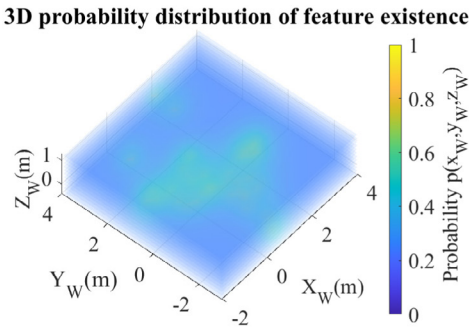
6.5. Image matching

This step consists in searching for similar features between a pair of images, that is, two-dimensional features that are the re-projection of the same 3D point across two different frames. A common approach to this task is to compare all feature descriptors in the first image to all other feature descriptors in the second image. After comparing all feature descriptors using a similarity measure, the correspondence of a feature is established by finding the nearest neighbour in the descriptor space.

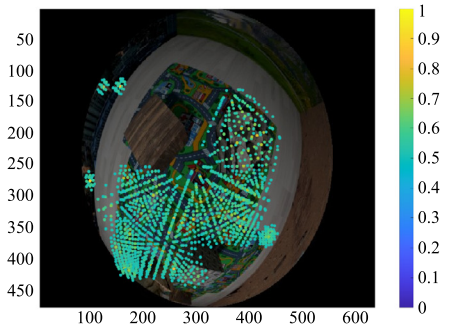
The problem of image matching can be formulated as follows (Hassaballah et al., 2016): after finding a set of interest points and extracting the feature descriptors around each one as a vector of length M, suppose that q_1^1 is one of these points in the first image I_1, and $\mathbf{F_1^1} = [f_1^1(1), f_1^1(2) \dots f_1^1(M)]$ is its feature descriptor. The aim is to find the best matching point q_2^j from the set of N feature points detected in the second image I_2 so, $j = 1, 2 \dots N$. To this end, the feature vector $\mathbf{F_1^1}$ is compared with each keypoint descriptor extracted $\mathbf{F^{j_2}} = [f_2^j(1), f_2^j(2) \dots f_2^j(M)]$ from I_2 by means of a distance function such as the Euclidean.

$$ d_j(\mathbf{F_1^1}, \mathbf{F^{j_2}}) = \left\| \mathbf{F_1^1} - \mathbf{F^{j_2}} \right\| = \sqrt{\sum_{i=1}^{M} (f_2^j(i) - f_1^1(i))^2} \tag{27} $$

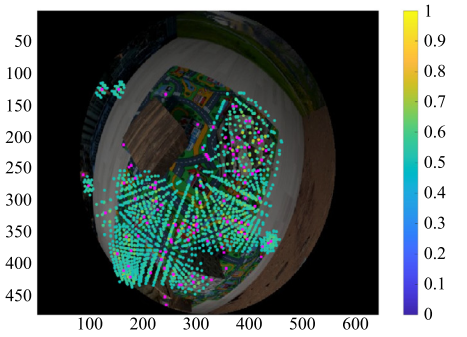
where $j = 1, 2 \dots N$ and N is the number of keypoints in I_2. Once all the distances are calculated, the nearest neighbour is searched, that is, the one with the minimum distance d_{1st}. The feature point associated is



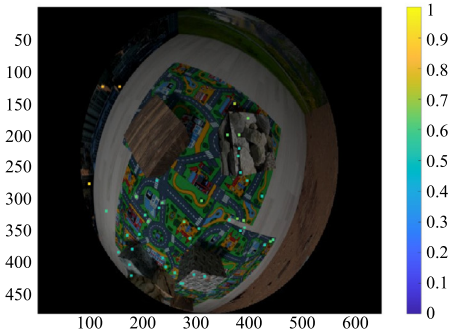
(a) Probability model of the environment. 3D probability distribution of feature existence (Section 6.2)



(b) Projected probability on the image at the next pose I_{t+1} (Section 6.3)



(c) Detected SURF points are plot on the image as squares in magenta colour ■.



(d) Candidate points with an associated probability (Section 6.4).

Fig. 12. The 3D points with a probability (a) are transformed from the world frame to the camera frame at pose $t + 1$. Then, they are projected on the image and their pixel coordinates are obtained (b). Once the projected points and the detected SURF points are expressed in the image at $t + 1$ (c), the process to extract SURF points as candidate is carried out. Finally, we obtain a set of SURF points (candidates) with an associated probability (d). (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

accepted as a correspondence of q_1^1 only if this distance is smaller than a threshold.

However, this requirement is not enough to discard ambiguous matches, and this is due to the fact that some descriptors are much more discriminative than others. For this reason, another condition based on Nearest Neighbour Distance Ratio (NNDR) (Lowe, 2004; Mikolajczyk and Schmid, 2005) has been used to find the best match. This method considers a matching is reliable only if the closest neighbour is significantly closer than the closest incorrect match. Thus, the distance ratio between the nearest $\mathbf{F_2^{1st}}$ and the second nearest $\mathbf{F_2^{2nd}}$ image descriptor is used.

$$NNDR = \frac{d_{1st}}{d_{2nd}} = \frac{\left\|\mathbf{F_1} - \mathbf{F_2^{1st}}\right\|}{\left\|\mathbf{F_1} - \mathbf{F_2^{2nd}}\right\|} \leq th_{ratio} \tag{28}$$

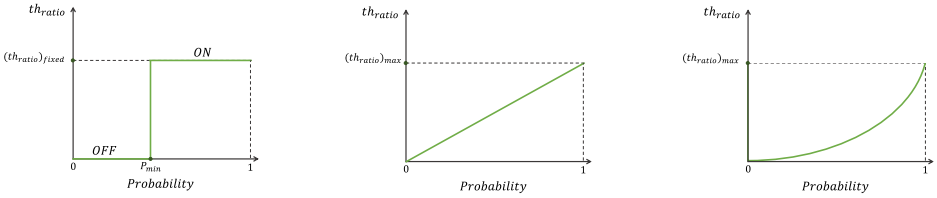
where d_{1st} and d_{2nd} are the Euclidean distances to the nearest and second nearest neighbour respectively. A correct match will have a distance ratio lower than a specific threshold, whereas an ambiguous match or an incorrect match will have a distance ratio close to one (Hassaballah et al., 2019).

Taking all this information into consideration, the feature point associated to the nearest feature descriptor (the one with minimum Euclidean distance) is considered as the best match only if this distance is lower than a matching threshold ($th_{matching}$) and the ratio between the nearest and the second closest match is smaller than a ratio threshold (th_{ratio}).

As presented in the previous section, the APOFM employs the 3D probability distribution to obtain the set of candidate points. In the present paper, we propose using this probability information to weigh the matching search as well. On account of that, the improved APOFM employs a weighted and dynamic matching evaluated under three custom functions, as presented in Fig. 13. The value of the $th_{matching}$ is constant; by contrast, the th_{ratio} value will depend on the function used (step, linear or square). In the weighted matching with a step function, which is shown in Fig. 13(a), only the projected probability points whose associated probability is higher than a threshold (P_{min}) are considered. The points whose probability is lower than P_{min} are not considered in the matching search. In the weighted matching with a linear function, which is shown in Fig. 13(b), all the projected probability points are taken into account and the value of the th_{ratio} is established according to the associated probability and a linear function. In the weighted matching with a square function, which is shown in Fig. 13(c), all the projected probability points are taken into account and the value of the th_{ratio} is established according to the associated probability and this function.

7. Results

In order to have objective evidences of the performance of this work, this section presents results evaluated in a publicly available dataset, with the inclusion of a benchmark of the different methods introduced in Table 1. As stated in the introduction, one of the goals of this work is to compare the performance of the improved APOFM (Section 6) solving the visual odometry with a SM (Hartley and Zisserman, 2003) described in Section 4. In addition to this, given that the main feature of the APOFM is the optimization of the matching search regarding to false positives, we have also compared SM with outlier rejection by means of RANSAC (Scaramuzza, 2011). The code implemented for this purpose is an open-source available in Yan (2011) which has been adapted to estimate the essential matrix. After performing a study, the values of the RANSAC parameters have been optimized to obtain the best estimation of the relative pose. We denote it as



(a) Step. (b) Linear. (c) Square.

Fig. 13. Value of th_{ratio} based on (a) step, (b) linear or (c) square function.

Table 1
Summary of the different methods and variations employed during the experiments.

Identification	Method	Function th_{ratio}	Parameters of the function
SM	Standard method (Hartley and Zisserman, 2003)	–	–
SM+RANSAC	Standard method and RANSAC to remove outliers (Scaramuzza, 2011)	–	–
WM-SF0.6	Improved APOFM	Step (Fig. 13(a))	$(th_{ratio})_{fixed} = 0.4$ and $P_{min} = 0.6$
WM-SF0.7	Improved APOFM	Step (Fig. 13(a))	$(th_{ratio})_{fixed} = 0.4$ and $P_{min} = 0.7$
WM-LF	Improved APOFM	Linear (Fig. 13(b))	$(th_{ratio})_{max} = 0.4$
WM-SqF	Improved APOFM	Square (Fig. 13(c))	$(th_{ratio})_{max} = 0.4$

SM+RANSAC. These comparisons are carried out using images taken with two different types of wide field of view cameras: fisheye and catadioptric.

In this regard, we have used the image dataset available in Robotics and Perception Group, University of Zurich, Switzerland (2013) and Zhang et al. (2016) composed by synthetic images generated with Blender. These images were rendered with two different camera models (fisheye and catadioptric) that were moving along the same trajectory in an indoor pixels for the fisheye model (180° FOV), and another sequence with the same number of images and resolution for the catadioptric model have been obtained.

On this matter, two plots have been obtained for each experiment, one using images captured by the catadioptric camera (they will be on the left side of the figures in the following subsections) and the other with images captured by the fisheye camera (these will be on the right side) and a comparative evaluation is performed. Altogether, six methods are considered which are summarized in Table 1. The first of them is the Standard Method (SM) and the remaining ones are variations of the improved APOFM, denoted in this section as WM (Weighted Matching). Focusing attention on the latter, the changes are related to the feature matching search step (Section 6.5). The second method (WM-SF0.6) considers a Step Function (SF) (Fig. 13(a)) to set th_{ratio}, with $P_{min} = 0.6$. The third method (WM-SF0.7) considers the same function with $P_{min} = 0.7$. The fourth method (WM-LF) uses a Linear Function (LF) to set th_{ratio} (Fig. 13(b)) with $(th_{ratio})_{max} = 0.4$ and, finally, method 5 (WM-SqF) makes use of a Square Function (SqF) (Fig. 13(c)) with $(th_{ratio})_{max} = 0.4$.

7.1. Parameters: $\Delta grid$ and χ

The APOFM depends mainly on two parameters. The first one is the value of χ, as mentioned in Section 6.4. This parameter is the threshold that determines if a detected feature point is a candidate to be a matching feature according to the distance between itself and the nearest projected probability point. The second parameter delimits the number of test points n_t that the GP has to treat.

Thus, the first experiment tries to evaluate the influence of these parameters upon the localization error and computation time. Thereby, the experiment will permit selecting optimum values for both parameters: χ and $\Delta grid$, so that the localization error is small and the computation time is admissible. Given that the third method (WM-SF0.7) is the most restrictive in comparison with the other proposed methods (i.e. fewer feature points are candidates), it has been employed for this experiment.

Therefore, the algorithm will be run for different values of χ and $\Delta grid$ whereas the values of the other parameters are fixed. A range of possible values for χ and for the $\Delta grid$ has been defined. For each possible combination of values of these parameters, the relative pose between each image of the dataset (t) and its three successive ones ($t + 1$, $t + 2$, $t + 3$) has been calculated. Then, the mean value of all localization errors obtained at each iteration is calculated. Fig. 14 shows the translation error (i.e. the error when the azimuth ϕ angle is estimated) which is shown with a specific colour based on its value. It is worth highlighting that, in some cases, it is not possible to estimate the relative pose because the number of correspondence pairs are not enough to obtain the essential matrix. These cases are represented with white colour in Figs. 14 and 15. It usually happens when (a) the value of χ is small and (b) the value of the $\Delta grid$ is high. The first condition denotes that χ is more restrictive in terms of distance and, consequently, the number of feature points considered candidates will be fewer. As a result of the second condition, the probability model of the environment will be represented by a low number of points, and the result is a loss of 3D information.

As Fig. 14 shows, the behaviour is different for each type of camera. In the case of the catadioptric vision system (Fig. 14(a)), the error is smaller when the value of the $\Delta grid$ is low, and the value of χ is high; in other words, when the method is less restrictive (i.e. there are more points to represent the scene and more feature points are considered as candidates). In the case of the fisheye camera (Fig. 14(b)), the smallest error is obtained when both parameters take values in the middle of the range of possible values.

Next, Fig. 15 shows the computation time of the process. In this case, the influence on the calculation time is the same, regardless of the camera type. In both cases, the time is shorter as the $\Delta grid$ is increased. This result makes sense given that the higher the value of the $\Delta grid$, the lower the number of test points is. This means that the number of points that the GP has to treat is lower and, as a consequence, the computation time is also lower. As regards the χ parameter, it can be observed that when it increases, so does the computation time. We could expect this fact since this means that more SURF points are considered as candidates, so both the matching search and the GP (training points) have to process more data in terms of points. However, the increment of the computation time is small compared to the one caused by the $\Delta grid$. Therefore, we can say that the GP has more influence on the computation time than the other parts of this process (e.g. the image processing), and it does not depend on the camera type.

Taking all the above information into account, the values of χ and $\Delta grid$ have been chosen to obtain a good balance between error and time for both kinds of cameras. As for the value of the $\Delta grid$, the

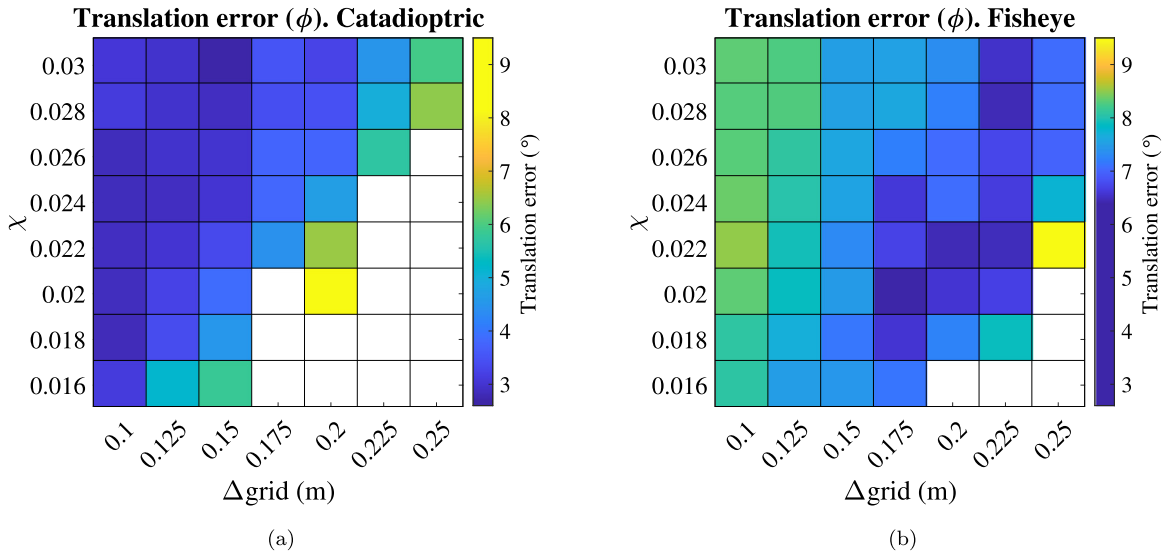


Fig. 14. The influence of the values of the $\Delta grid$ and χ upon the translation error when using either (a) a catadioptric or (b) a fisheye camera.

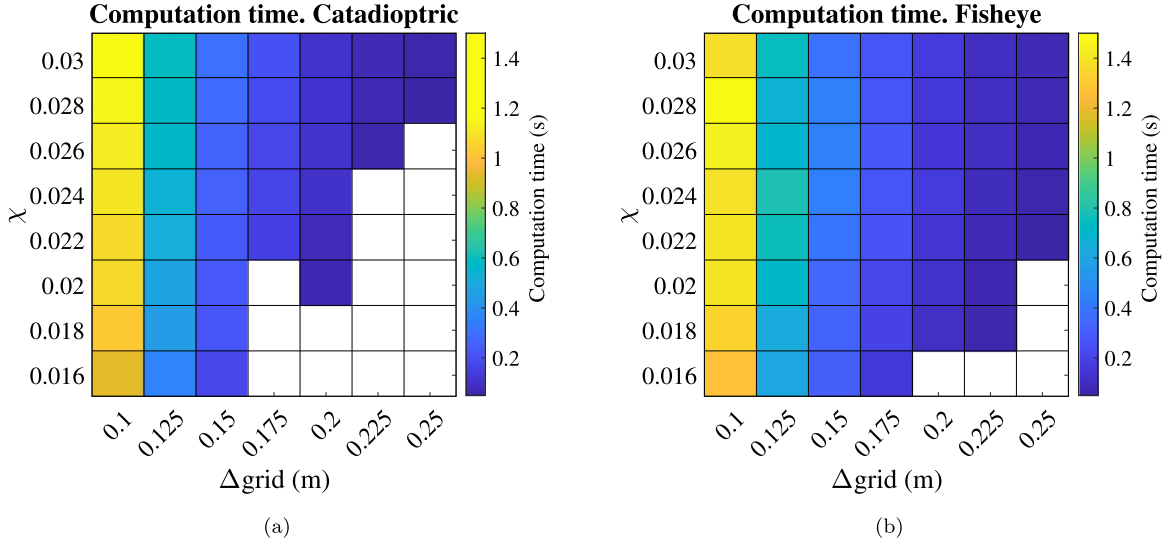


Fig. 15. The influence of the values of the $\Delta grid$ and χ upon the computation time when using either (a) a catadioptric or (b) a fisheye camera.

selected value is 0.15, since the results obtained in both cases show a good balance. While it is true that they are better for the catadioptric camera, higher values could lead to a case in which it is not possible to estimate the pose. Additionally, according to Fig. 15(b), the best result for this $\Delta grid$ occurs when χ is 0.018, so in the following sections, the experiments are carried out with these optimum values.

7.2. Number of feature matches

This subsection studies the number of SURF feature points corresponding to the next image I_{t+1} that have been considered in the search of matching features, and how many of them have found matches in the current image I_t. Fig. 16 shows the number of these sets of points considering images captured in different times, labelled as d_1, d_2, d_3. The first distance specified as d_1 (Fig. 16(a) and (b)) denotes that the algorithm considers the images I_t and I_{t+1}. The second distance specified as d_2 (Fig. 16(c) and (d)) means that the algorithm estimates the relative pose between the images I_t and I_{t+2}. Finally, in the case of d_3, the images taken at t and $t+3$ have been employed (Fig. 16(e) and (f)). In each sub-figure, the columns denote number of points (left axis). The first column corresponds to the SM, so it represents all SURF points detected in the next image I_{t+i} (where $i = 1, 2, 3$) and the number of them which have been found as a match in the current image I_t. In the second column, the same results are represented but employing RANSAC to estimate the essential matrix. The other columns show the results when the improved APOFM with specific variations is employed (Table 1). In these cases, the points considered in the searching of matching features are the candidate points (Section 6.4), therefore the number of these points and how many of them have been found as match are represented in each one of these columns.

Firstly, we analyse the results of the SM with each type of camera. Even though the number of detected SURF points is higher for the fisheye camera, the results show that the number of feature matches is higher for the catadioptric camera. This effect is likely to appear when the field of view is higher. Comparing to SM, the number of matches with SM+RANSAC is lower, as expected, since it removes those matches that do not fit well the model. It leads to lower values of matching ratio, especially with fisheye images.

Secondly, about the APOFM, comparing the number of candidate points, it can be said that more points have been determined as candidates with the fisheye camera than with the catadioptric one. However, if we calculate the ratio between them and the number of matching

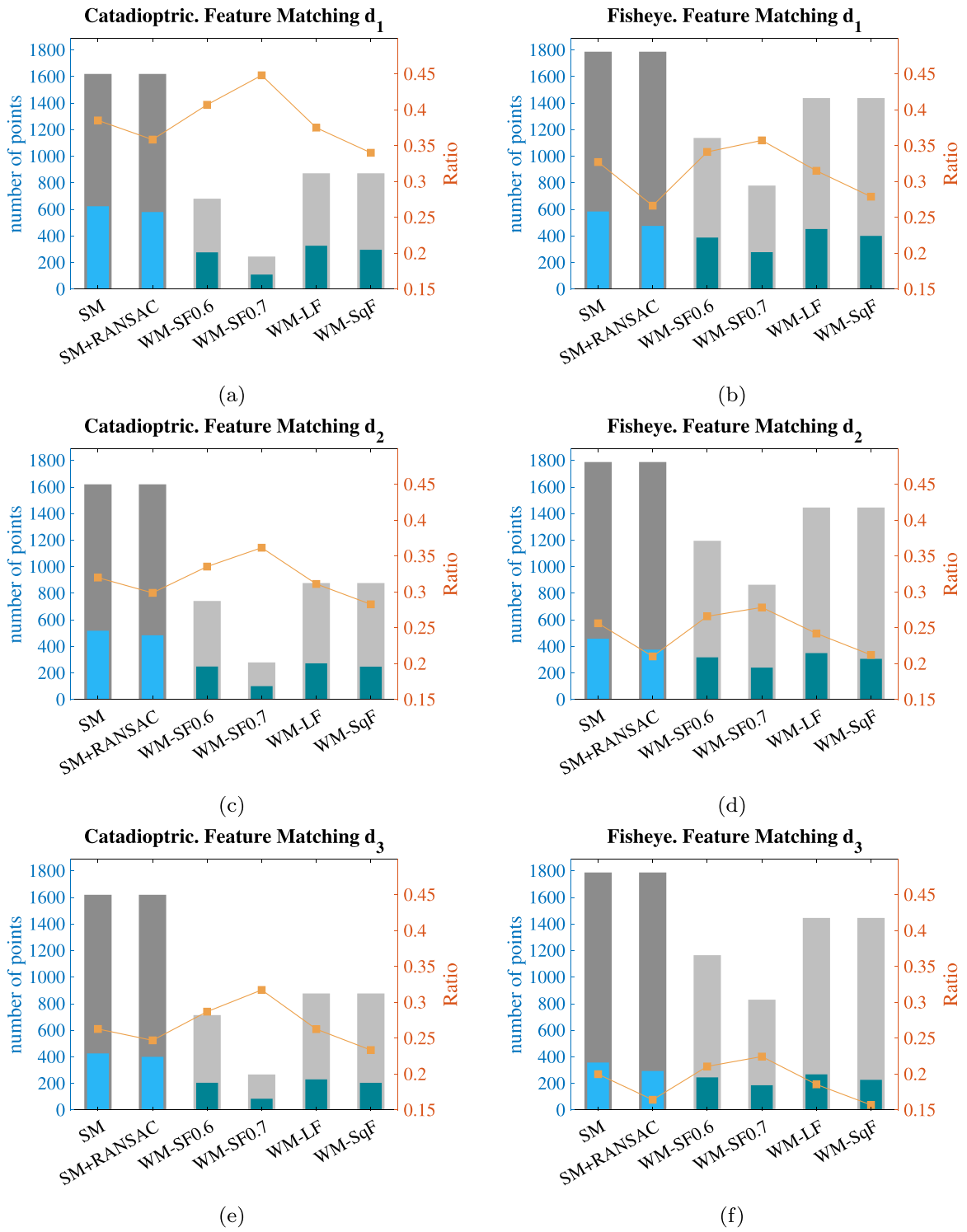
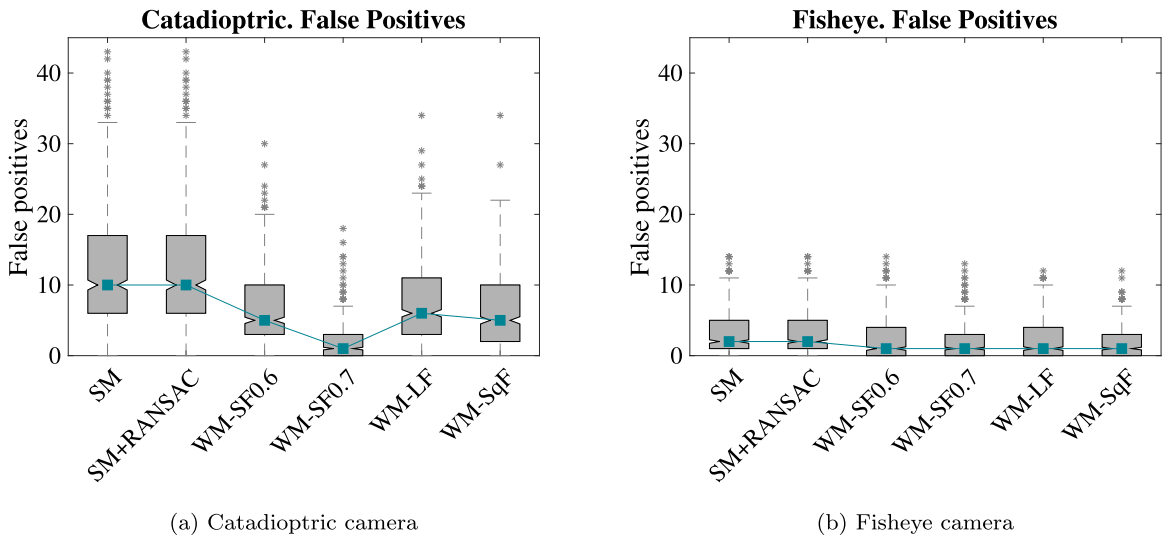


Fig. 16. Number of SURF points and ratio between the number of considered and matched points in the next image I_{i+i}, with (a), (b) $i = 1$; (c), (d) $i = 2$ and (e), (f) $i = 3$. The left axis shows the total number of points (num SURF ■); the number of them that have found a match in the current image I_i using SM (Standard Matching ■); the number of points considered as candidates by the APOFM (num candidates ■) and how many of these latter have found matches (Proposed Matching ■). The right y-axis shows the ratio ■ between the number of feature points used during the matching step and the number of them that have found a match. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

points (this ratio is represented as an orange tendency whose values are shown in the right axis of Fig. 16), the catadioptric camera provides better results. In other words, many candidate points are not found as correspondence in the case of the fisheye camera. This may be due to the fact that these candidates are extracted by metric distance to projected points with an associated probability (pixel frame). However, given the nature of the fisheye images, the reprojected rays of these candidates might be practically coincident with more than one 3D point. As a result, several 3D points might be associated with the same pixel location, thus losing the coincidence of their visual descriptors. Finally, the matching discards these points since it does not find corresponding visual descriptor.



(a) Catadioptric camera (b) Fisheye camera

Fig. 17. Distribution of the number of false positives in the images captured by (a) catadioptric and (b) fisheye camera. The SM and the variations of the improved APOFM (WM-SF0.6, WM-SF0.7, WM-LF and WM-SqF) have been compared.

Considering the number of feature points, the third method WM-SF0.7 provides the best results for both types of cameras since the ratio is high in this case. Even so, it is necessary to study its behaviour regarding the false positives and the localization error before determining if this method is the best one.

To complete this section, we discuss the effect of the distance between the images in the number of matched points (i.e. d_1, d_2, d_3). After observing the results achieved for each distance with the catadioptric camera (Fig. 16(a), (c) and (e)), and with the fisheye camera (Fig. 16(b), (d) and (f)), the conclusion is that, regardless the type of camera, the number of matches is lower when the distance between the images is higher. This difference of matches is lower in the case of the catadioptric camera.

7.3. False positives

A high number of feature matches does not indicate *per se* that a specific method is more effective. It is true that the higher the number of matches, the more information about the relative motion, and consequently, the localization error is expected to be smaller. However, as mentioned in Section 6.5, some of these matches may be false positives and may lead to a wrong estimation of the relative camera pose.

In this sense, Fig. 17 shows the number and distribution of false positives with each method by means of a boxplot. Each one represents all the false positives between the current image I_t and the three successive ones I_{t+1}, I_{t+2} and I_{t+3}. Once the plots have been observed, the conclusion is that the range of the number of false positives is greater using the SM than using the variations proposed in this work. In particular, the third proposal WM-SF0.7 demonstrates a more condensed distribution, fact that implies a lower number of false positives, but also less dispersion.

Even though the results for the SM and SM+RANSAC seem similar, there is a small decrease in the number of false positives. Still, this difference is much more significant with the improved APOFM.

After keeping the result obtained in this subsection as well as the previous ones in mind, it can be said that the feature matches are more robust in all cases in which the improved APOFM has been employed, since the number of false positives is smaller, though this implies that some true positives have also been eliminated. For this reason, it is also necessary to study the localization error, based on the method employed.

Regarding the type of camera, there are fewer false positives in the images taken by a fisheye camera than by a catadioptric camera. Furthermore, the lower whisker of the boxplots does not exist for the improved APOFM since the median is near to zero. In the majority of the iterations to obtain the relative pose, the number of false positives obtained is between zero and a small value.

7.4. Localization error

One of the aims of the work is to solve the localization problem; hence it is necessary to make a study about the error obtained after estimating the relative pose with each of the methods. Two different distance measures have been applied to determine the candidate feature points. As mentioned in Section 6.4, they are the Mahalanobis distance (exhaustive search algorithm) and City-Block distance (kd-tree search algorithm). Figs. 18 and 19 show the angular error made in the estimation of the relative pose by means of a bar graph, where the first bar corresponds to the error using the SM. Each variation of the improved APOFM has two bars. The first one shows the error using the Mahalanobis distance to determine candidate feature points; the second bar represents the angular error when the distance used is City-Block. Fig. 18 shows the angular error after estimating the translation vector (ϕ and β), whereas Fig. 19 shows the orientation error (θ, γ and α).

After analysing Fig. 18, it can be deduced that the translation error is smaller using a fisheye camera than a catadioptric camera with the SM. In contrast to this, the improvement with regard to the translation error can be appreciated better when the improved APOFM (all variations) is employed with the images taken by a catadioptric camera. The translation error using RANSAC (SM+RANSAC) is lower than without it (SM) and even similar to WM-SF0.6 for the fisheye images. Nevertheless, the use of the improved APOFM provides a more precise solution for all remaining cases.

With respect to the distance metric, it can be observed that the City-Block with the fisheye camera leads to a smaller translation error, independently of the proposed method case. In this sense, the behaviour with the catadioptric camera is different, the Mahalanobis distance seems to be better to the second (WM-SF0.6) and fourth (WM-LF) case, whereas the City-Block distance outputs a smaller error in the third (WM-SF0.7) and fifth (WM-SqF) case. All the same, a considerable difference of error between both distance measures can only be seen in the third case when the feature points with an associated probability higher than 0.7 are screened.

Finally, the angular error after estimating the orientation is studied. As Fig. 19 shows, the rotation error behaves very similarly to the translation error. However, it is worth highlighting that the low error

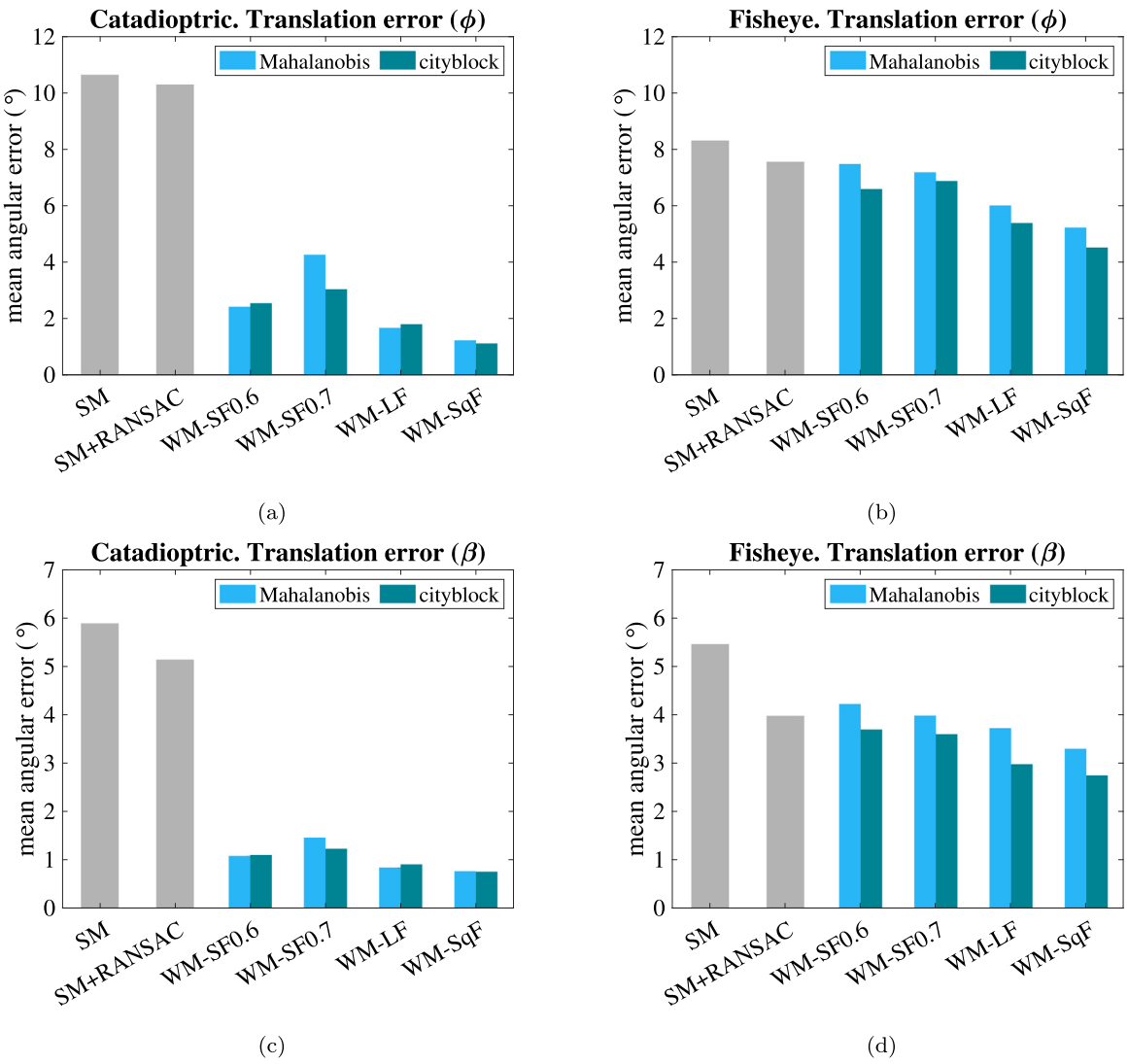


Fig. 18. Translation error. Each subfigure shows the angular error employing SM ( ) and the variations of the improved APOFM (Table 1: WM-SF0.6, WM-SF0.7, WM-LF and WM-SqF) based on the distance used (Mahalanobis ■ and cityblock ■). The error estimating ϕ with a catadioptric camera is shown in (a) and with a fisheye lens in (b). The error estimating β with a catadiotric camera appears in (c) and with a fisheye lens in (d). (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

achieved when the orientation is estimated, in contrast with the one obtained in the translation vector estimation, being this mean angular error below a remarkable value of 1°.

8. Conclusion

In this work, the localization problem is solved using visual information. The basis of this method relies on the former approach proposed in Valiente et al. (2018), which presented a visual information fusion approach for Adaptive Probability-Oriented Feature Matching (APOFM). Despite the fact that Valiente et al. (2018) exploited the potential of GP to produce a 3D representation with probability of feature existence towards obtaining a robust and adaptive matching, several aspects have been improved in the present work.

The main goal of this work was to improve the former method and to extend its application to images captured by a catadioptric and by fisheye camera, so as to produce a consistent comparison between two well-recognized vision systems within the field of visual localization. In this context, we have benchmarked the improved APOFM against: (a) a Standard Method (SM) (Hartley and Zisserman, 2003), (b) this SM with outlier rejection by means of RANSAC (SM+RANSAC) (Scaramuzza, 2011) and (c) the basic APOFM (Valiente et al., 2018) (WM-SF0.6 and WM-SF0.7). This comparative evaluation has comprised several variations associated to new contributions. This analysis appraises efficiency and computation when using these two types of vision systems under a publicly available dataset.

Additionally, we have presented an improved search method for matching candidates with the support of a k-nearest neighbour classifier which matches the nearest projected point on the images (with an associated probability) with a feature point, resulting in a matching candidate. It has been implemented as a Kd-tree algorithm using the City-block distance, and compared to an exhaustive search using the Mahalanobis distance. Next, we have improved the use of visual information in terms of the spatial probability distribution. In contrast to the previous method, which only used such information for filtering within a maximum probability those feature points allowed to result in matching candidates, now we enhance its use to achieve a weighted search of features, which dynamically adapts to the current probability of feature existence by applying three probability-weighted variations of the improved APOFM. Finally, a more reliable detection of false positives has been introduced. It is supported by a design that evaluates the pixel coincidence of the projected 3D point onto the image, assumed as a matched point between two images. Thus, a match is tagged as a false positive if the projection of its 3D point does not converge towards

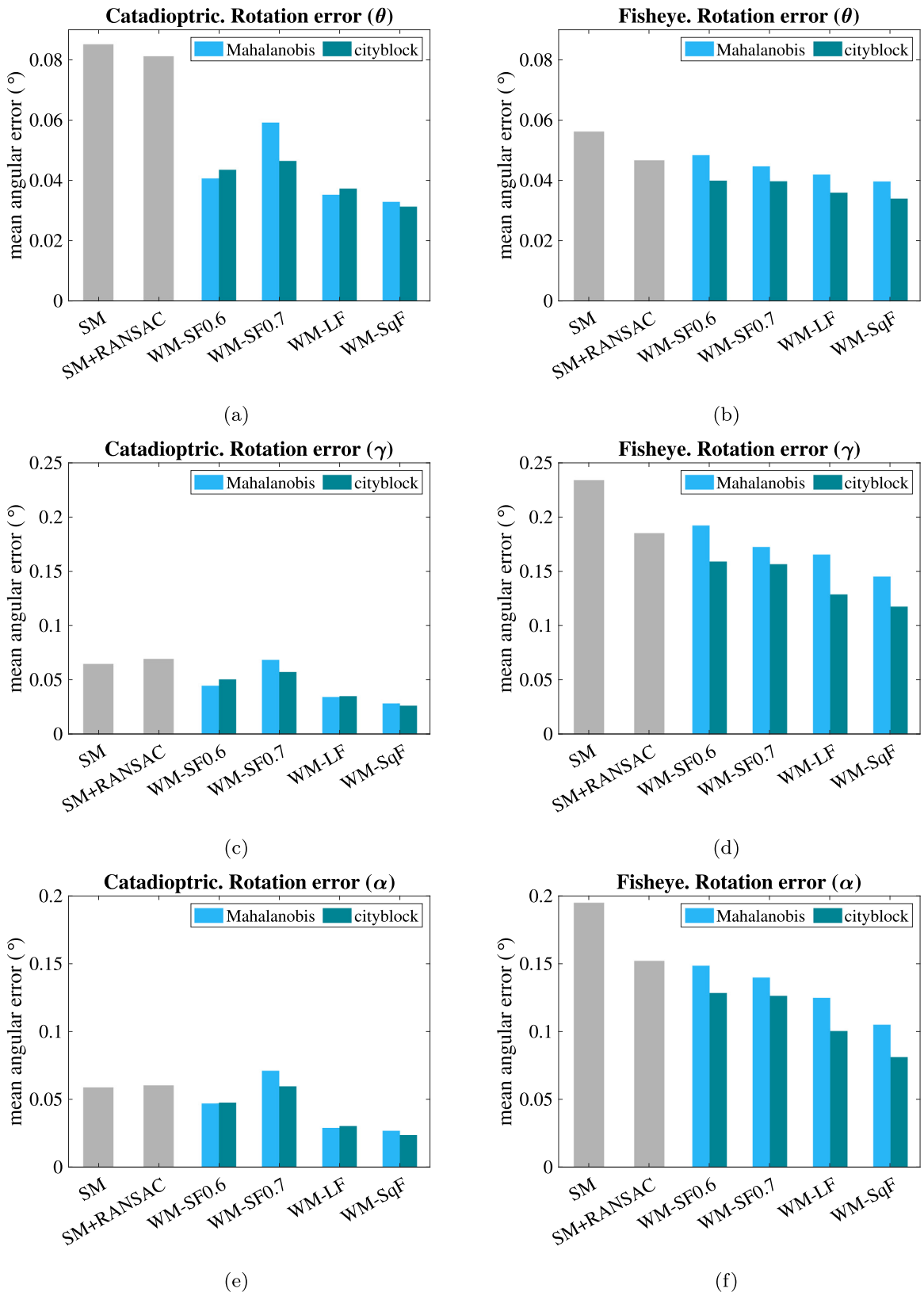


Fig. 19. Rotation error. Each subfigure shows the angular error employing SM ( ) and the variations of the improved APOFM (Table 1: WM-SF0.6, WM-SF0.7, WM-LF and WM-SqF) based on the distance used (Mahalanobis and cityblock). The error estimating θ with a catadioptric camera is shown in (a) and with a fisheye lens in (b). The error estimating γ with a catadiotric camera appears in (c) and with a fisheye lens in (d). The error estimating α with a catadiotric camera appears in (e) and with a fisheye lens in (f). (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

the same pixel point which was initially marked as a SURF point, within a certain distance threshold.

Before producing comparative results, a preliminary experiment was carried out in order to extract an optimum set of parameters for the

GP computation. The execution of the method constraints the accuracy with the computation load. This study reveals that a trade-off setup can be established between the spatial resolution of the 3D testing points in the probabilistic model ($\Delta grid$), and the distance threshold that discerns whether a feature point is considered as a matching candidate (χ). Although both vision systems, the catadioptric and the fisheye, should be tuned with their specific trade-off setups, these experiments considered the same value of $\Delta grid$ and χ for both vision systems test-bed, in order to ensure an acceptable balance.

After inspecting the comparative results between the catadioptric and fisheye images, it can be confirmed that the improved APOFM provides an enhanced accuracy and efficiency, comparing to SM, regardless the sort of vision system employed, either catadioptric or fisheye.

Regarding the performance of the three variations of the improved APOFM, the results corroborate several benefits of these contributions. First, they confer higher ratios of detected matching candidates, versus the total amount of feature points, in comparison with the SM. Particularly, the method WM-SF0.7 returns the highest ratio. Notably, the fisheye camera produces more matching candidates, however, due to its nonlinear nature and field of view, the final set of matched points is more reduced than the one provided by the catadioptric system.

As for the false positives detector, the proposed variations demonstrate to outperform significantly both the SM and the SM+RANSAC. The fisheye images perform better in this sense, fact that is justified by the lower number of matched points in the last stage.

Finally, focusing on the accuracy of the visual localization, it can be confirmed that the error associated to the relative pose estimation is lower when a weighted matching search with the square function is employed. Moreover, the best performance is obtained when the City-block distance is used to establish which feature point is the nearest to a certain projected point with an associated probability, and thus obtaining a valid matching candidate. The outputs of the experimental set lead to deduce that the catadioptric vision system produces lower errors with all the methods with the improved APOFM approach. It is noteworthy that the translation error, which typically is the worst affected by noise and non-linearities of these lenses, is bounded by a value under 1 degree (mean error) with the proposed variation of the method (linear and square).

In summary, this work has validated the appropriateness of the proposed contributions to deal with the visual localization problem. The estimated relative pose is defined by five angular parameters, three for the orientation and two for the translation, so this method presents the inconvenient the translation vector is obtained with the exception of the scale factor. Taking all these facts into account, the results have evidenced to outperform the SM, as well as SM+RANSAC. Furthermore, the suitability of these implementations have been extensively tested against a publicly dataset, which at the same time permitted producing an extended evaluation and comparison over two of the most commonly used vision systems in visual localization, within mobile robotics.

The evaluation has been carried out in an indoor environment. As future work, it will be interesting to employ this method using images taken from an outdoor environment. Another future work could be to compare the robustness of this method using a camera with a fisheye lens and a vision system composed of two fisheye cameras pointing to opposite sides (a full 360 degrees of view).

CRediT authorship contribution statement

María Flores: Methodology, Software, Investigation, Writing – original draft, Data curation. **David Valiente:** Methodology, Software, Supervision, Writing – review & editing, Validation. **Arturo Gil:** Methodology, Resources, Software, Writing – review & editing. **Oscar Reinoso:** Resources, Conceptualization, Validation, Project administration. **Luis Payá:** Conceptualization, Supervision, Writing – review & editing, Validation, Project administration.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

This work is part of the project PID2020-116418RB-I00 funded by MCIN/AEI/10.13039/501100011033, of the project AICO/2019/031 funded by Generalitat Valenciana, Spain, and of the grant ACIF/2020/141 funded by Generalitat Valenciana, Spain and Fondo Social Europeo, European Union.

References

Alatise, M.B., Hancke, G.P., 2020. A review on challenges of autonomous mobile robot and sensor fusion methods. IEEE Access 8, 39830–39846. http://dx.doi.org/10.1109/ACCESS.2020.2975643.

Amorós, F., Payá, L., Mayol-Cuevas, W., Jiménez, L.M., Reinoso, O., 2020. Holistic descriptors of omnidirectional color images and their performance in estimation of position and orientation. IEEE Access 8, 81822–81848. http://dx.doi.org/10.1109/ACCESS.2020.2990996.

Andert, F., Goormann, L., 2007. Combined grid and feature-based occupancy map building in large outdoor environments. In: Proceeding of 2007 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2007). pp. 2065–2070. http://dx.doi.org/10.1109/IROS.2007.4399086.

Aqel, M.O.A., Marhaban, M.H., Saripan, M.I., Ismail, N.B., 2016. Review of visual odometry: types, approaches, challenges, and applications. SpringerPlus 5 (1), 1897. http://dx.doi.org/10.1186/s40064-016-3573-7.

Barone, S., Neri, P., Paoli, A., Razionale, A., 2018. Catadioptric stereo-vision system using a spherical mirror. Procedia Struct. Integr. 8, 83–91. http://dx.doi.org/10.1016/j.prostr.2017.12.010.

Bay, H., Tuytelaars, T., Van Gool, L., 2006. SURF: Speeded up robust features. In: Proceedings of Computer Vision–ECCV 2006: 9th European Conference on Computer Vision. 3951, pp. 404–417. http://dx.doi.org/10.1007/11744023_32.

Beardsley, P.A., Zisserman, A., Murray, D.W., 1994. Navigation using affine structure from motion. In: Proceedings of Computer Vision – ECCV '94: Third European Conference on Computer Vision. 801, pp. 85–96. http://dx.doi.org/10.1007/BFb0028337.

Bentley, J.L., 1975. Multidimensional binary search trees used for associative searching. In: Ashenhurst, R.L. (Ed.), Commun. ACM 18 (9), 509–517. http://dx.doi.org/10.1145/361002.361007.

Boutteau, R., Savatier, X., Ertaud, J.-Y., Mazari, B., 2010. A 3D omnidirectional sensor for mobile robot applications. In: Barrera, A. (Ed.), Mobile Robots Navigation. InTech, http://dx.doi.org/10.5772/9001.

Cebollada, S., Payá, L., Mayol, W., Reinoso, O., 2019. Evaluation of clustering methods in compression of topological models and visual place recognition using global appearance descriptors. Appl. Sci. 9 (3), http://dx.doi.org/10.3390/app9030377.

Chomboon, K., Chujai, P., Teerarassammee, P., Kerdprasop, K., Kerdprasop, N., 2015. An empirical study of distance metrics for k-nearest neighbor algorithm. In: Proceedings of the 3rd International Conference on Industrial Application Engineering 2015 (ICIAE2015). pp. 280–285. http://dx.doi.org/10.12792/iciae2015.051.

Cover, T., Hart, P., 1967. Nearest neighbor pattern classification. IEEE Trans. Inform. Theory 13 (1), 21–27. http://dx.doi.org/10.1109/TIT.1967.1053964.

Dalla Libera, A., Tosello, E., Pillonetto, G., Ghidoni, S., Carl, R., 2019. Proprioceptive robot collision detection through Gaussian process regression. In: Proceedings of 2019 American Control Conference (ACC). pp. 19–24. http://dx.doi.org/10.23919/ACC.2019.8814361.

Emani, S., Soman, K.P., Sajith Variyar, V.V., Adarsh, S., 2019. Obstacle detection and distance estimation for autonomous electric vehicle using stereo vision and DNN. In: Wang, J., Reddy, G.R.M., Prasad, V.K., Reddy, V.S. (Eds.), Soft Computing and Signal Processing, Vol. 898. Springer Singapore, Singapore, pp. 639–648. http://dx.doi.org/10.1007/978-981-13-3393-4_65.

Fraundorfer, F., Scaramuzza, D., 2012. Visual odometry: Part II: Matching, robustness, optimization, and applications. IEEE Robot. Autom. Mag. 19 (2), 78–90. http://dx.doi.org/10.1109/MRA.2012.2182810.

Gao, W., Shen, S., 2017. Dual-fisheye omnidirectional stereo. In: Proceeding of 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2017). pp. 6715–6722. http://dx.doi.org/10.1109/IROS.2017.8206587.

Gao, W., Wang, K., Ding, W., Gao, F., Qin, T., Shen, S., 2020. Autonomous aerial robot using dual-fisheye cameras. J. Field Robot. 37 (4), 497–514. http://dx.doi.org/10.1002/rob.21946.

Geyer, C., Daniilidis, K., 2000. A unifying theory for central panoramic systems and practical implications. In: Proceedings of Computer Vision – ECCV 2000: 6th European Conference on Computer Vision. 1843, pp. 445–461. http://dx.doi.org/10.1007/3-540-45053-X_29.

Ghaffari, M., Gan, L., Parkison, S.A., Li, J., Eustice, R.M., 2017. Gaussian processes semantic map representation. arXiv:1707.01532 [Cs].

Ghaffari, M., Valls Miro, J., Dissanayake, G., 2018. Gaussian processes autonomous mapping and exploration for range-sensing mobile robots. Auton. Robot. 42 (2), 273–290. http://dx.doi.org/10.1007/s10514-017-9668-3.

Gil, A., Juliá, M., Reinoso, O., 2015. Occupancy grid based graph-SLAM using the distance transform, SURF features and SGD. Eng. Appl. Artif. Intell. 40, 1–10. http://dx.doi.org/10.1016/j.engappai.2014.12.010.

Harapanahalli, S., Mahony, N.O., Hernandez, G.V., Campbell, S., Riordan, D., Walsh, J., 2019. Autonomous navigation of mobile robots in factory environment. Procedia Manuf. 38, 1524–1531. http://dx.doi.org/10.1016/j.promfg.2020.01.134.

Hartley, R.I., Sturm, P., 1997. Triangulation. Comput. Vis. Image Underst. 68 (2), 146–157. http://dx.doi.org/10.1006/cviu.1997.0547.

Hartley, R., Zisserman, A., 2003. Multiple View Geometry in Computer Vision, second ed. Cambridge University Press, Cambridge, UK.

Hassaballah, M., Abdelmgeid, A.A., Alshazly, H.A., 2016. Image features detection, description and matching. In: Awad, A.I., Hassaballah, M. (Eds.), Image Feature Detectors and Descriptors, Vol. 630. Springer International Publishing, Cham, pp. 11–45. http://dx.doi.org/10.1007/978-3-319-28854-3_2.

Hassaballah, M., Alshazly, H.A., Ali, A.A., 2019. Analysis and evaluation of keypoint descriptors for image matching. In: Hassaballah, M., Hosny, K.M. (Eds.), Recent Advances in Computer Vision, Vol. 804. Springer International Publishing, Cham, pp. 113–140. http://dx.doi.org/10.1007/978-3-030-03000-1_5.

Hou, J., Yu, L., Fei, S., 2020. A highly robust automatic 3D reconstruction system based on integrated optimization by point line features. Eng. Appl. Artif. Intell. 95, 103879. http://dx.doi.org/10.1016/j.engappai.2020.103879.

Jakubović, A., Velagić, J., 2018. Image feature matching and object detection using brute-force matchers. In: Proceedings of ELMAR 2018: 60th International Symposium ELMAR-2018. pp. 83–86. http://dx.doi.org/10.23919/ELMAR.2018.8534641.

Jiang, Y., Xu, Y., Liu, Y., 2013. Performance evaluation of feature detection and matching in stereo visual odometry. Neurocomputing 120, 380–390. http://dx.doi.org/10.1016/j.neucom.2012.06.055.

Jung, S., Lee, U., Jung, J., Shim, D.H., 2016. Real-time traffic sign recognition system with deep convolutional neural network. In: Proceedings of URAI 2016: 13th International Conference on Ubiquitous Robots and Ambient Intelligence (URAI). pp. 31–34. http://dx.doi.org/10.1109/URAI.2016.7734014.

Kostavelis, I., Charalampous, K., Gasteratos, A., Tsotsos, J.K., 2016. Robot navigation via spatial and temporal coherent semantic maps. Eng. Appl. Artif. Intell. 48, 173–187. http://dx.doi.org/10.1016/j.engappai.2015.11.004.

Lee, G.H., Fraundorfer, F., Pollefeys, M., 2013. Structureless pose-graph loop-closure with a multi-camera system on a self-driving car. In: Proceeding of 2013 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2013). pp. 564–571. http://dx.doi.org/10.1109/IROS.2013.6696407.

Li, S., 2006. Full-view spherical image camera. In: Proceedings of 18th International Conference on Pattern Recognition (ICPR'06), Vol. 4. pp. 386–390. http://dx.doi.org/10.1109/ICPR.2006.585.

Li, B., Wang, Y., Zhang, Y., Zhao, W., Ruan, J., Li, P., 2020. GP-SLAM: laser-based SLAM approach based on regionalized Gaussian process map reconstruction. Auton. Robot. 44 (6), 947–967. http://dx.doi.org/10.1007/s10514-020-09906-z.

Liu, Y., Chen, J., Bai, X., 2020. An approach for multi-objective obstacle avoidance using dynamic occupancy grid map. In: Proceedings of 2020 IEEE International Conference on Mechatronics and Automation (ICMA). pp. 1209–1215. http://dx.doi.org/10.1109/ICMA49215.2020.9233760.

Liu, Y., Li, Y., Dai, L., Yang, C., Wei, L., Lai, T., Chen, R., 2021. Robust feature matching via advanced neighborhood topology consensus. Neurocomputing 421, 273–284. http://dx.doi.org/10.1016/j.neucom.2020.09.047.

Lowe, D.G., 2004. Distinctive image features from scale-invariant keypoints. Int. J. Comput. Vis. 60 (2), 91–110. http://dx.doi.org/10.1023/B:VISI.0000029664.99615.94.

Marcato Junior, J., Tommaselli, A.M.G., Moraes, M.V.A., 2016. Calibration of a catadioptric omnidirectional vision system with conic mirror. ISPRS J. Photogramm. Remote Sens. 113, 97–105. http://dx.doi.org/10.1016/j.isprsjprs.2015.10.008.

Mikolajczyk, K., Schmid, C., 2005. A performance evaluation of local descriptors. IEEE Trans. Pattern Anal. Mach. Intell. 27 (10), 1615–1630. http://dx.doi.org/10.1109/TPAMI.2005.188.

Mohamed, S.A.S., Haghbayan, M.-H., Westerlund, T., Heikkonen, J., Tenhunen, H., Plosila, J., 2019. A survey on odometry for autonomous navigation systems. IEEE Access 7, 97466–97486. http://dx.doi.org/10.1109/ACCESS.2019.2929133.

Nair, N.S., Nair, M.S., 2020. On evolutionary computation techniques for multi-view triangulation. Mach. Vis. Appl. 31 (4), 29. http://dx.doi.org/10.1007/s00138-020-01077-2.

Nguyen, L., Miro, J.V., Shi, L., Vidal-Calleja, T., 2019. Gaussian mixture marginal distributions for modelling remaining metallic pipe wall thickness. In: Proceedings of the IEEE 2019: 9th International Conference on Cybernetics and Intelligent Systems (CIS), Robotics, Automation and Mechatronics (RAM). pp. 257–262. http://dx.doi.org/10.1109/CIS-RAM47153.2019.9095851.

Nister, D., 2003. An efficient solution to the five-point relative pose problem. In: Proceedings of 2003 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'03). 2, pp. II–195–202. http://dx.doi.org/10.1109/CVPR.2003.1211470.

Nutalapati, M.K., Arora, L., Bose, A., Rajawat, K., Hegde, R.M., 2019. Model free calibration of wheeled robots using Gaussian process. In: Proceedings of 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). pp. 29–35. http://dx.doi.org/10.1109/IROS40897.2019.8967569.

O'Callaghan, S.T., Ramos, F.T., 2012. Gaussian process occupancy maps. Int. J. Robot. Res. 31 (1), 42–62. http://dx.doi.org/10.1177/0278364911421039.

Park, S., Huang, Y., Goh, C.F., Shimada, K., 2018. Robot model learning with Gaussian process mixture model. In: Proceedings of 2018 IEEE: 14th International Conference on Automation Science and Engineering (CASE). pp. 1263–1268. http://dx.doi.org/10.1109/COASE.2018.8560452.

Patruno, C., Colella, R., Nitti, M., Renò, V., Mosca, N., Stella, E., 2020. A vision-based odometer for localization of omnidirectional indoor robots. Sensors 20 (3), 875. http://dx.doi.org/10.3390/s20030875.

Payá, L., Gil, A., Reinoso, O., 2017. A state-of-the-art review on mapping and localization of mobile robots using omnidirectional vision sensors. J. Sensors 2017, 1–20. http://dx.doi.org/10.1155/2017/3497650.

Poddar, S., Kottath, R., Karar, V., 2018. Evolution of visual odometry techniques. arXiv:1804.11142 [Cs].

Polymenakos, K., Laurenti, L., Patane, A., Calliess, J.-P., Cardelli, L., Kwiatkowska, M., Abate, A., Roberts, S., 2020. Safety guarantees for planning based on iterative Gaussian processes. arXiv:1912.00071 [Cs, Stat].

Rasmussen, C.E., Williams, C.K.I., 2006. Gaussian Processes for Machine Learning. MIT Press, Cambridge, Mass.

Reinoso, O., Payá, L., 2020. Special issue on visual sensors. Sensors 20 (3), 910. http://dx.doi.org/10.3390/s20030910.

Robotics and Perception Group, University of Zurich, Switzerland, 2013. The "multi-fov" synthetic datasets. (accessed 22 October 2021), http://rpg.ifi.uzh.ch/fov.html.

Román, V., Payá, L., Cebollada, S., Reinoso, O., 2020. Creating incremental models of indoor environments through omnidirectional imaging. Appl. Sci. 10 (18), 6480. http://dx.doi.org/10.3390/app10186480.

Scaramuzza, D., 2011. 1-point-RANSAC structure from motion for vehicle-mounted cameras by exploiting non-holonomic constraints. Int. J. Comput. Vis. 95 (1), 74–85. http://dx.doi.org/10.1007/s11263-011-0441-3.

Scaramuzza, D., 2014. Omnidirectional camera. In: Ikeuchi, K. (Ed.), Computer Vision. Springer US, Boston, MA, pp. 552–560. http://dx.doi.org/10.1007/978-0-387-31439-6_488.

Scaramuzza, D., Fraundorfer, F., 2011. Visual odometry [tutorial]. IEEE Robot. Autom. Mag. 18 (4), 80–92. http://dx.doi.org/10.1109/MRA.2011.943233.

Scaramuzza, D., Martinelli, A., Siegwart, R., 2006a. A flexible technique for accurate omnidirectional camera calibration and structure from motion. In: Proceedings of the Fourth IEEE International Conference on Computer Vision Systems (ICVS 2006). p. 45. http://dx.doi.org/10.1109/ICVS.2006.3.

Scaramuzza, D., Martinelli, A., Siegwart, R., 2006b. A toolbox for easily calibrating omnidirectional cameras. In: Proceedings of 2006 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). pp. 5695–5701. http://dx.doi.org/10.1109/IROS.2006.282372.

Siegwart, R., Nourbakhsh, I.R., Scaramuzza, D., 2011. Introduction to Autonomous Mobile Robots. MIT Press, Cambridge, Mass.

Song, X., Cao, Z., Gao, H., 2018. Local Gaussian processes for identifying complex mobile robot system. In: Chen, X., Zhao, Q. (Eds.), Proceedings of the 37th Chinese Control Conference. pp. 3796–3802. http://dx.doi.org/10.23919/ChiCC.2018.8483251.

Sun, K., Saulnier, K., Atanasov, N., Pappas, G.J., Kumar, V., 2018. Dense 3-D mapping with spatial correlation via Gaussian filtering. In: Proceedings of 2018 Annual American Control Conference (ACC). pp. 4267–4274. http://dx.doi.org/10.23919/ACC.2018.8431777.

Taheri, H., Xia, Z.C., 2021. SLAM; definition and evolution. Eng. Appl. Artif. Intell. 97, 104032. http://dx.doi.org/10.1016/j.engappai.2020.104032.

Thrun, S., Burgard, W., Fox, D., 2005. Probabilistic Robotics. In: Intelligent robotics and autonomous agents, MIT Press, Cambridge, Mass.

Tresp, V., 2000. A Bayesian committee machine. Neural Comput. 12 (11), 2719–2741. http://dx.doi.org/10.1162/089976600300014908.

Valiente, D., Payá, L., Jiménez, L., Sebastián, J., Reinoso, O., 2018. Visual information fusion through Bayesian inference for adaptive probability-oriented feature matching. Sensors 18 (7), 2041. http://dx.doi.org/10.3390/s18072041.

Valiente García, D., Fernández Rojo, L., Gil Aparicio, A., Payá Castelló, L., Reinoso García, O., 2012. Visual odometry through appearance- and feature-based method with omnidirectional images. J. Robot. 2012, 1–13. http://dx.doi.org/10.1155/2012/797063.

Wu, B.-F., Lu, W.-C., Jen, C.-L., 2011. Monocular vision-based robot localization and target tracking. J. Robot. 2011, 1–12. http://dx.doi.org/10.1155/2011/548042.

Xiao, Q., Liu, X., Liu, M., 2012. Object tracking based on local feature matching. In: Proceedings of ISCID 2012: Fifth International Symposium on Computational Intelligence and Design (ISCID 2012). pp. 399–402. http://dx.doi.org/10.1109/ISCID.2012.106.

Yan, K., 2011. RANSAC algorithm with example of finding homography. MATLAB central file exchange. (accessed 22 October 2021), https://es.mathworks.com/matlabcentral/fileexchange/30809-ransac-algorithm-with-example-of-finding-homography.

Ying, X., Hu, Z., 2004. Can we consider central catadioptric cameras and fisheye cameras within a unified imaging model. In: Proceedings of Computer Vision – ECCV 2004: 8th European Conference on Computer Vision. 3021, pp. 442–455. http://dx.doi.org/10.1007/978-3-540-24670-1_34.

Yousif, K., Bab-Hadiashar, A., Hoseinnezhad, R., 2015. An overview to visual odometry and visual SLAM: Applications to mobile robotics. J. Intell. Syst. 1 (4), 289–311. http://dx.doi.org/10.1007/s40903-015-0032-7.

Yuan, M., Yau, W.-Y., Li, Z., 2018. Lost robot self-recovery via exploration using hybrid topological-metric maps. In: Proceedings of TENCON 2018: IEEE Region 10 Conference. pp. 188–193. http://dx.doi.org/10.1109/TENCON.2018.8650236.

Zhang, H., Hernandez, D., Su, Z., Su, B., 2018. A low cost vision-based road-following system for mobile robots. Appl. Sci. 8 (9), 1635. http://dx.doi.org/10.3390/app8091635.

Zhang, Z., Rebecq, H., Forster, C., Scaramuzza, D., 2016. Benefit of large field-of-view cameras for visual odometry. In: Proceedings of 2016 IEEE International Conference on Robotics and Automation (ICRA 2016). pp. 801–808. http://dx.doi.org/10.1109/ICRA.2016.7487210.

Zivkovic, Z., Bakker, B., Krose, B., 2005. Hierarchical map building using visual landmarks and geometric constraints. In: Proceedings of 2005 IEEE/RSJ International Conference on Intelligent Robots and Systems. http://dx.doi.org/10.1109/IROS.2005.1544951, 2480–2485.