A Vision-Based System for Robot Localization in Large Industrial Environments

Rémi Boutteau, Romain Rossi, Lei Qin, Pierre Merriaux, Xavier Savatier

Noname manuscript No.
(will be inserted by the editor)

A vision-based system for robot localization in large industrial environments

Rémi Boutteau · Romain Rossi · Lei Qin · Pierre Merriaux · Xavier Savatier

Received: date / Accepted: date

Abstract In this paper, we propose a vision-based system to localize mobile robots in large industrial environments. Our contributions rely on the use of fisheye cameras to have a large field of view and the associated algorithms. We propose several calibration methods and evaluate them with a ground-truth obtained by a motion capture system. In these experiments, we also evaluate the influence of the parameters as the number of points used for calibration, or the influence of the accuracy of these points. Our system is then experimented in a real industrial environment, where we localize blue an ROV (Remotely Operated underwater Vehicle) in a basin. As shown is this paper, this system can also localize wheeled mobile robots in the same way.

Keywords vision-based localization · fisheye cameras · robust pose estimation · extrinsic calibration

1 Introduction

Robots have long been used in industry, but they have long been confined to automated production tasks. In recent years, with the advent of the Factory of the Future (FoF) or Industry

R. Boutteau
Normandie Univ, UNIROUEN, ESIGELEC, IRSEEM, 76000 Rouen, France
E-mail: boutteau@esigelec.fr

R. Rossi
Normandie Univ, UNIROUEN, ESIGELEC, IRSEEM, 76000 Rouen, France
E-mail: rossi@esigelec.fr

L. Qin
Normandie Univ, UNIROUEN, ESIGELEC, IRSEEM, 76000 Rouen, France
E-mail: qin@esigelec.fr

P. Merriaux
Normandie Univ, UNIROUEN, ESIGELEC, IRSEEM, 76000 Rouen, France
E-mail: merriaux@esigelec.fr

X. Savatier
Normandie Univ, UNIROUEN, ESIGELEC, IRSEEM, 76000 Rouen, France
E-mail: savatier@esigelec.fr

4.0, they evolve to be more autonomous, interact with each other and collaborate with humans safely. The new technologies used in the Factory of the Future have brought out the concept of Cyber-Physical System (CPS), a system that interacts with its environment, extracts data, and processes them to control a physical process. In this context, Cyber-Physical Systems deployed in companies must be autonomous and embedded.

One of the main applications of mobile robotics in factories is logistics, through the use of AGV (Automatic Guided Vehicle), for example to move products or raw materials between different workplaces or storage spaces. These Cyber-Physical Systems can also perform inventory tasks, for example by moving in the factory while counting and locating objects with RFID tags [30].

To be able to navigate autonomously, a mobile robot must be able to localize itself accurately in its environment. In this purpose, several types of sensors can be used. The goal being to localize mobile robots in indoor environments, we can discard GPS-type technologies whose signals are not received in buildings. If we exclude proprioceptive sensors (e.g. odometers) that are generally not accurate enough for a reliable localization, two technologies are generally used: lidars and cameras.

Lidar (Light detection and ranging) sensors have the advantage of providing accurate and reliable data. Many localization systems work with these sensors [22, 24], but there are some limitations. The first is its high price, especially if it is necessary to use 3D lidars. This is specially true if a whole fleet of robots must be localized (each robot will have to embed one or more lidars). The second limitation is their electrical consumption since lidars are active sensors.

Vision sensors is the second technology mainly used for localization and has undeniable advantages: the cameras are passive sensors, and are inexpensive. Many approaches have been developed in recent years for robot or vehicle localization based on vision, the most advanced solutions being [23] and [28].

In this work, we are interested in a vision-based system for robot localization by instrumenting the factory rather than the robot itself. This makes it possible to locate a whole fleet of mobile robots without multiplying the number of cameras required. The application described in this paper is the remote control of an ROV (Remotely Operated underwater Vehicle) to recover industrial sludge in a silo, but this system can easily be used for the localization of AGVs in a factory. Our experimental results also illustrate this use case since we use wheeled mobile robots.

Our main contributions are:

- the use of omnidirectional vision theory, to develop a system for robots localization in large environments using a reduced number of cameras,
- the development of several calibration methods, depending on the constraints of the environment,
- the evaluation of these calibration methods and the localization accuracy by studying the influence of several parameters,
- the use of our localization system in a real industrial context.

The remainder of the paper is organized as follows. Section 2 presents the overall architecture of the system we have developed. Section 3 describes the Unified Projection Model (UPM) that we used to model fisheye cameras. Section 4 is dedicated to the various calibration methods that we have developed according to the operational constraints of the application (prior knowledge of the building, etc.). Section 5 presents our method of detection and localization. Section 6 presents our experimental results, both quantitative in the laboratory with a ground truth and qualitative in our industrial application. We present the results with

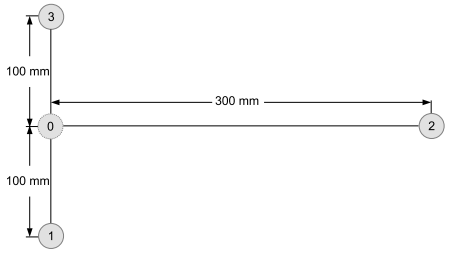

Fig. 1 Geometric layout of the LEDs on the mobile. The luminous pattern is composed of 3 or 4 LEDs (LED 0 is not mandatory).

the different calibration methods and we study the influence of the different parameters on the localization accuracy. Finally, Section 7 summarizes this paper and includes some ideas for future work.

2 System overview

Our localization system is based on the tracking of a light pattern embedded on the mobile. In our work, we used a luminous pattern composed of 3 LEDs forming a ”T” and spaced 300mm apart on one axis and 200mm apart on the second axis as shown in the Figure 1. This system can be embedded on a mobile robot or an AGV as illustrated by our experimental results and in Figure 8.

The industrial application of our work is localizing an ROV in a silo. This aim of our ROV is to collect industrial sludge in a silo using a delamination screw. To empty the silo, it is therefore necessary to make adjacent passages to remove the sludge to a depth of a few centimeters. To be able to do this, the operator must know accurately the position of the ROV because it does not have a visual feedback, the ROV being in a closed silo and therefore very dark.

The complete system we have developed for the ROV localization is described in Figure 2. One or two fisheye cameras (depending on the size of the silo) are positioned on the roof of the silo and observe the surface of the water. The ROV lies on the sludge and is totally immersed in the water. To be able to locate it, it is equipped with a buoy having three LEDs as shown in Figure 3. This buoy is attached to an articulated rod equipped with a sensor allowing to obtain its angle of inclination with respect to the ROV. A rangefinder provides us the distance between the roof of the silo and the surface of the water since the water level varies during the sludge extraction process.

The general idea is to detect the LED points in the image captured by the fish-eye camera and then to reconstruct their 3D positions. Concretely, the proposed system consists of the following components: calibration (intrinsic parameters, extrinsic parameters), LED detection and 3D reconstruction, localization and finally multiple camera fusion.

3 The Unified Projection Model (UPM)

Fisheye cameras present a very great interest in our application because we can cover a very large area with only one or two cameras. However, fisheye cameras do not obey the pinhole projection model commonly used in computer vision [13].

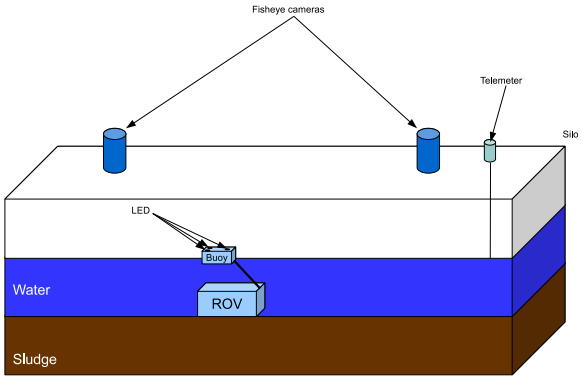

Fig. 2 System overview. The ROV lies on the sludge and is totally immersed in the water. To be able to locate it, it is equipped with a buoy having three LEDs. This buoy is attached to an articulated rod equipped with a sensor allowing to obtain its angle of inclination with respect to the ROV. A rangefinder provides us the distance between the roof of the silo and the surface of the water since the water level varies during the sludge extraction process.

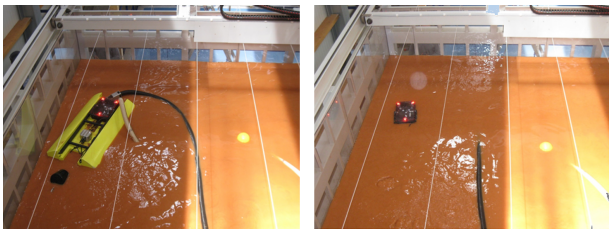

Fig. 3 The ROV in a test basin similar to the silo. On the left when it is on the surface, on the right in the working situation.

Many models have been proposed to represent a fisheye camera. These models can be classified into 3 categories.

The first category of models directly uses the optical models that were used by the manufacturer to build the optics. The projections commonly used are then stereographic, equiangular, sine-law or equi-solid depending on the reference of the optics [26].

The second category is based on the use of the pinhole model commonly used in computer vision. A 3D point is projected on the image plane and then distortions are added to the projected point to take into account the deformations induced by the fisheye. The mapping between the perspective projection and the fisheye point can be logarithmic [2], polynomial [25], or rational [6, 9].

Finally, the third category is based on the use of the Unified Projection Model (UPM) initially used for catadioptric sensors [1, 10, 20]. It has been demonstrated on numerous occasions that this model is also adapted to fisheye cameras [7, 8, 29].

The model used in this paper belongs to this third category. We used the model introduced by Mei and Rives [20] and inspired from those of Barreto [1] and Geyer [10]. In this section, the equations for projecting a 3D point to obtain the coordinates of the corresponding pixel are presented. The inverse projection, called lifting, allowing to pass from one pixel of the image to the direction of the corresponding light ray, is also presented.

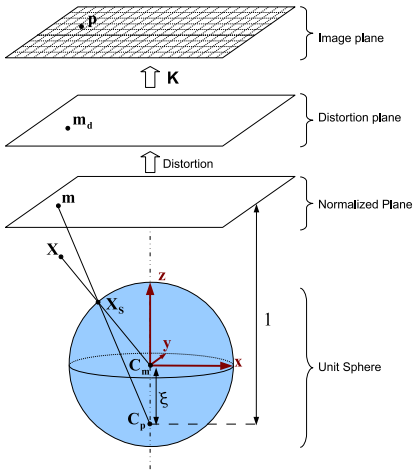

Fig. 4 The Unified Projection Model (UPM).

In general, points are expressed in a reference frame different from the one of the camera, called world frame. The two reference frames are related by a rigid transform $\mathbf{T}^{\mathbf{cam}}_{\mathbf{world}}$. If $\mathbf{X_{world}} = [X\ Y\ Z\ 1]^\top$ is the homogeneous coordinate vector of a point expressed in the world frame, and $\mathbf{X_{cam}}$ Is the coordinate vector of this same point expressed in the camera frame, then we can write

$$\mathbf{X_{cam}} = \mathbf{T}^{\mathbf{cam}}_{\mathbf{world}}\mathbf{X_{world}}. \tag{1}$$

If $\mathbf{C}$ represents the coordinates of the camera center in the world coordinate system, and $\mathbf{R}$ is the matrix expressing the orientation of the camera frame, then Equation (1) can be written

$$\mathbf{X_{cam}} = \begin{bmatrix} \mathbf{R} & -\mathbf{RC} \\ \mathbf{0}_{1\times 3} & 1 \end{bmatrix} \mathbf{X_{world}}. \tag{2}$$

There are several representations to express a rotation. In our model, we use quaternions. The rigid transform function, denoted by w, expressed in Equation (1), will be characterized by a vector of 7 elements (4 for the quaternion and 3 for the translation): $\mathbf{V_1} = [q_w\ q_x\ q_y\ q_z\ t_x\ t_y\ t_z]^\top$.

Once the coordinates of the point are expressed in the camera frame $\mathbf{X_{cam}} = [X_c\ Y_c\ Z_c\ 1]^\top$, the latter is projected onto the unit sphere using the following equation:

$$\mathbf{X_S} = \begin{bmatrix} X_S \\ Y_S \\ Z_S \end{bmatrix} = \begin{bmatrix} \frac{X_c}{\sqrt{X_c^2+Y_c^2+Z_c^2}} \\ \frac{Y_c}{\sqrt{X_c^2+Y_c^2+Z_c^2}} \\ \frac{Z_c}{\sqrt{X_c^2+Y_c^2+Z_c^2}} \end{bmatrix}. \tag{3}$$

This point undergoes a second projection on the normalized plane from a point situated at a distance ξ from the sphere center. The projection of $\mathbf{X_S}$ onto the normalized plane, denoted by $\mathbf{m} = [x\ y]^\top$, is therefore obtained by:

$$\mathbf{m} = \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} \frac{X_S}{Z_S+\xi} \\ \frac{Y_S}{Z_S+\xi} \end{bmatrix}. \tag{4}$$

These two transformations (projection on the sphere and then projection onto the normalized plane) are grouped together in the h function which depends only on a parameter: $\mathbf{V_2} = [\xi]$.

Distortions are then added to the point $\mathbf{m}$ by applying the distortion function d which involves five coefficients: three radial distortion coefficients (k_1, k_2 and k_5) and two tangential distortion coefficients (k_3 and k_4). Let $\mathbf{V_3} = [k_1\ k_2\ k_3\ k_4\ k_5]^\top$ be the parameter vector containing these coefficients, $\rho = \sqrt{x^2+y^2}$ and $\mathbf{m_d} = [x_d\ y_d]^\top$ the resulting point, its coordinates are obtained by:

$$\mathbf{m_d} = \mathbf{m} + d(\mathbf{m}), \tag{5}$$

and consequently:

$$\mathbf{m_d} = \begin{bmatrix} x(1+k_1\rho^2+k_2\rho^4+k_5\rho^6)+2k_3xy+k_4(\rho^2+2x^2) \\ y(1+k_1\rho^2+k_2\rho^4+k_5\rho^6)+2k_4xy+k_3(\rho^2+2y^2) \end{bmatrix}. \tag{6}$$

The final projection is obtained by the generalized projection matrix $\mathbf{K}$. The latter contains 5 parameters: the generalized focal lengths γ_u and γ_v, the coordinates of the principal point u_0 and v_0, and the skewness parameter α. Let k be this projection function and $\mathbf{V_4} = [\alpha\ \gamma_u\ \gamma_v\ u_0\ v_0]^\top$ the vector of parameters on which it depends, the projection $\mathbf{p} = [u\ v\ 1]^\top$ of the point $\mathbf{m_d}$ on the image plane is then given by:

$$\mathbf{p} = \mathbf{K} \begin{bmatrix} \mathbf{m_d} \\ 1 \end{bmatrix}, \tag{7}$$

$$\mathbf{p} = \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} \gamma_u & \gamma_u\alpha & u_0 \\ 0 & \gamma_v & v_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \mathbf{m_d} \\ 1 \end{bmatrix}. \tag{8}$$

It is common to simplify the matrix $\mathbf{K}$ by setting $\alpha = 0$, which corresponds to assuming that rows and columns of the pixels are orthogonal.

Let $\mathbf{V} = [\mathbf{V_1}^\top\ \mathbf{V_2}^\top\ \mathbf{V_3}^\top\ \mathbf{V_4}^\top]^\top$ be the vector containing the 18 parameters of the model. The global projection function of a point $\mathbf{X}$, denoted by $p(\mathbf{V},\mathbf{X})$ is obtained by composition of the different functions presented previously:

$$p(\mathbf{V},\mathbf{X}) = k \circ d \circ h \circ w(\mathbf{V},\mathbf{X}). \tag{9}$$

These steps are applied to obtain the projection onto the image plane of a 3D point knowing its coordinates in the 3D space.

The inverse projection, called lifting, consists in retro-projecting a point, i.e. knowing its pixel coordinates $\mathbf{p_{pix}} = [u\ v\ 1]^\top$, determining the direction of the corresponding light beam.

It is necessary to start by re-projecting the $\mathbf{p_{pix}}$ point on the normalized plane by performing an inverse perspective projection to obtain the point $\mathbf{m} = [x\ y\ 1]^\top$:

$$\mathbf{m} = \mathbf{K}^{-1}\mathbf{p_{pix}}, \tag{10}$$

hence:

$$\mathbf{m} = \begin{bmatrix} \frac{u-u_0}{\gamma_u} \\ \frac{v-v_0}{\gamma_v} \\ 1 \end{bmatrix}. \tag{11}$$

To obtain the corresponding $\mathbf{X_s}$ point on the sphere, we use the formula:

$$\mathbf{X_S} = \begin{bmatrix} \frac{\xi+\sqrt{1+(1-\xi^2)(x^2+y^2)}}{x^2+y^2+1}x \\ \frac{\xi+\sqrt{1+(1-\xi^2)(x^2+y^2)}}{x^2+y^2+1}y \\ \frac{\xi+\sqrt{1+(1-\xi^2)(x^2+y^2)}}{x^2+y^2+1} - \xi \end{bmatrix}. \tag{12}$$

4 Calibration

Two kinds of calibration have to be taken into account in such an application. First, we have to estimate the parameters $\mathbf{V_2}$, $\mathbf{V_3}$ and $\mathbf{V_4}$ of the unified projection model. These parameters are inherent properties of a camera and this kind of calibration is called *intrinsic calibration*. To do so, we used a calibration method described in [3, 4].

Once $\mathbf{V_2}$, $\mathbf{V_3}$,$\mathbf{V_4}$ are estimated, we are about to calibrate $\mathbf{V_1}$. Indeed, $\mathbf{V_1}$ encodes the geometric relation between the camera coordinate frame and the world coordinate frame. In this sense, the process for calibrating $\mathbf{V_1}$ is usually called *extrinsic calibration*.

For notation convenience, we denote hereafter the position of a LED point in the world coordinate frame as $\mathbf{X^{led}_{world}}$, the position of the camera in the world coordinate frame as $\mathbf{X^{cam}_{world}}$, the position of the camera in the ROV coordinate frame as $\mathbf{X^{cam}_{rov}}$ and the pixel coordinates of a LED point in the image as $\mathbf{p^{led}_{pix}}$. Similarly, we denote the orientation of the camera in the world coordinate frame as $\mathbf{R^{cam}_{world}}$ and the orientation of the camera in the ROV coordinate frame as $\mathbf{R^{cam}_{rov}}$.

The extrinsic parameters to be estimated, i.e. $\mathbf{V_1}$, can be decomposed into $\mathbf{R^{cam}_{world}}$ and $\mathbf{T^{cam}_{world}}$. After calibration, to reconstruct the 3D position of a point by its projection in the image, we also need the equation of the plane π in addition to the projection parameters $\mathbf{V}$. Depending on whether we have a set of precisely known 3D positions in the scene, we present in the following two methods for estimating $\mathbf{R^{cam}_{world}}$, $\mathbf{T^{cam}_{world}}$ and π.

4.1 Extrinsic calibration using points with known 3D positions

Estimating $\mathbf{R^{cam}_{world}}$ and $\mathbf{T^{cam}_{world}}$ requires a set of 3D-2D point correspondences. If we have a set of N points whose precise 3D positions in the world coordinate frame are known, we can take a picture of the scene and get the pixel locations of these points in the image. In this way, for the ith point in the set, both its 3D position in the world coordinate frame $\mathbf{X^i_{world}}$ and the 2D pixel location of its projection in the image $\mathbf{p^i_{pix}}$ are available.

The estimation of $\mathbf{R^{cam}_{world}}$ and $\mathbf{T^{cam}_{world}}$ consists in finding the parameters that minimize the sum of re-projection errors:

$$E(\mathbf{V_1}) = \frac{1}{2}\sum_{i}^{\mathbf{N}}(\mathbf{p^i_{pix}} - P(\mathbf{V_1}, \mathbf{X^i_{world}}))^2. \tag{13}$$

The parameters to be estimated have six degrees of freedom (three for the translation and three for the rotation). As each correspondence leads to two equations (x-axis and y-axis),

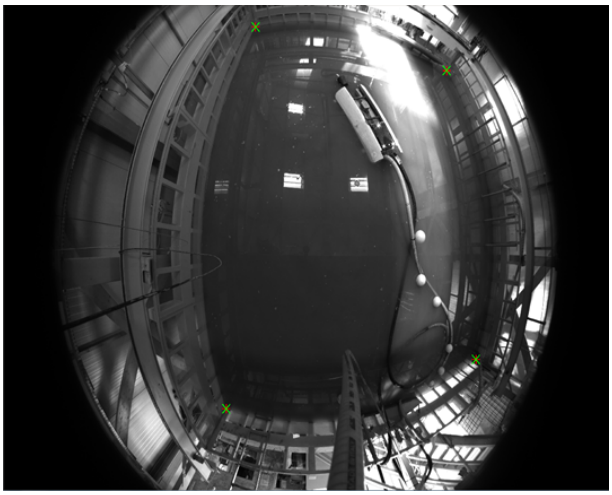

Fig. 5 Reprojection of the known points in the camera image. The red points are the selected (ground-truth) points, the green points are the reprojected points after extrinsic calibration. The error is less than one pixel per point.

N must be equal to (or greater than) 3. In practice, at least one supplementary point correspondence is needed in order to assess the quality of the optimization result. We use the Levenberg-Marquardt algorithm [17] to solve this nonlinear least-squares optimization problem (Fig. 5).

We can ask ourselves the impact of the accuracy of the 3D points coordinates (which are not always easy to determine) on the result of the extrinsic calibration. Similarly, the minimum number of points is 3, but does having more points have an impact on accuracy? All these questions are discussed in section 6.

4.2 Auto-Calibration by moving the ROV

In real environments, it can be difficult to measure 3D points very accurately. In our case, for example, it is difficult, if not impossible, to access the interior of the silo. Consequently, it may be difficult to obtain a set of precisely known 3D points for calibration. To circumvent this problem, we propose in this section an alternative calibration method by moving the ROV in the scene.

Thanks to the camera intrinsic parameters and the knowledge of the LEDs geometry, we can estimate the position of the ROV with respect to the camera. The method is exactly the same as the one presented in section 4.1 except that the known points are no longer those of the silo but the ones of the ROV LEDs. This problem is similar to what is known in literature as the PnP (Perspective-n-Point) problem [16]. The methods proposed in the literature are not directly applicable, however, since they are intended for conventional perspective cameras; in our case, we minimize the function given in Equation (13) to recover the pose of the ROV with respect to the camera.

We therefore have a set of ROV positions. It is consequently possible to estimate the plane parameters on which the ROV moves in the camera frame: its normal $\mathbf{n_{cam}} = [n_x\, n_y\, n_z]^\top$ and the orthogonal distance d between the camera and the plane. We know that in the silo

frame, this normal is vertical since the target floats on the water. We consequently have: $\mathbf{n_{world}} = [0\ 0\ 1]^\top$. To obtain the orientation of the camera, we look for the rotation matrix $\mathbf{R_{cam}^{world}}$ which allows us to switch from the normal expressed in the camera frame to the one expressed in the silo frame:

$$\begin{bmatrix} n_x \\ n_y \\ n_z \end{bmatrix} = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}. \tag{14}$$

Rotation around the water plane's normal (yaw) cannot be recovered. We thus only consider pitch and roll angles. The Denavit-Hartenberg [5, 12] parameterization of $\mathbf{R_{cam}^{world}}$ with these two angles leads to

$$\begin{bmatrix} n_x \\ n_y \\ n_z \end{bmatrix} = \begin{bmatrix} cos(\theta) & -sin(\theta)cos(\alpha) & sin(\theta)sin(\alpha) \\ sin(\theta) & cos(\theta)cos(\alpha) & -cos(\theta)sin(\alpha) \\ 0 & sin(\alpha) & cos(\alpha) \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}. \tag{15}$$

From (15), α (roll) and θ (pitch) can be easily recovered since

$$\begin{cases} \alpha = \arccos(n_z) \\ \theta = \frac{\arcsin(n_x)}{\sqrt{1-\cos^2(\alpha)}} \end{cases}. \tag{16}$$

With this method, it is therefore possible to find the attitude (roll and pitch) and the altitude of the camera relative to the plane of the water. To find an absolute pose, it is possible to use one of the ROV poses (for example the first one) as the origin of the world frame. The ROV can for example be positioned at a corner of the silo to set the origin.

5 Detection and Localization

In this section we present the processing chain that allows us to locate the ROV in the silo from the images. The points corresponding to the LEDs are detected in the images, then their 3D coordinates are calculated. A series of tests is carried out to reject possible false detection. Finally, the location of the ROV can be estimated. At the end of this section we also present the method used when the environment is large and requires the use of several cameras.

5.1 LEDs detection in the image

In order to get $\mathbf{p_{pix}^{led}}$, we need to detect the projections of the LEDs in the image. Here we do not use the background subtraction methods as in Section 4.2 due to speed considerations. Instead, we use a series of image processing techniques to detect the LEDs of the ROV in the image.

Given an image captured by the camera with the LEDs of the ROV in the scene, we first blur it to remove singular points. Then, we convert the image to binary by an intensity thresholding. Dark regions are rejected. Next, we apply a connected components labeling in the binary image to get a number of separated areas. In fact, each area represent a candidate LED point for further validation. We can reject the areas that are either too small or too large if we have some prior knowledge about the size of the LEDs in the image. Finally, we use the center location of the area to represent the pixel coordinates $\mathbf{p_{pix}}$ of a candidate point.

It is often the case that after this series of processing there are still a large number of candidate points, among which we need to select the real LED points. To tackle this problem, we propose to first reconstruct the 3D positions of the points and then to detect the LEDs in the reconstructed plane according to their geometrical configurations.

5.2 3D Position Reconstruction

We now present the method to reconstruct the 3D position $\mathbf{X_{world}}$ of a point $\mathbf{X}$, which is supposed to lie on the plane π, from its projection $\mathbf{p_{pix}}$ in the image.

First we lift $\mathbf{p_{pix}}$ back to the unit sphere to obtain $\mathbf{X_s}$ with equations (11) and (12). This point is then expressed in the world frame using the rigid transform $\mathbf{T_{cam}^{world}}$ obtained by the extrinsic calibration step. Let $\mathbf{A}$ be this point.

Obtaining the 3D coordinates of the LED point consists in computing the intersection of the line that connects $\mathbf{X_s}$ and the unit sphere center (see Figure 4) with the plane π. Let $\mathbf{B}$ be the position of the unit sphere center in the world coordinate frame.

The line that joins $\mathbf{A}$ and $\mathbf{B}$ can be expressed by a Plücker matrix [13] as:

$$\mathbf{L} = \mathbf{A}\mathbf{B}^\top - \mathbf{B}\mathbf{A}^\top. \tag{17}$$

Finally, the reconstructed 3D position is the intersection of the line $\mathbf{L}$ and the plane π. We get $\mathbf{X_{world}} = \mathbf{L} \cdot \pi$. Note that the obtained $\mathbf{X_{world}}$ is represented in homogeneous coordinates. We can get the original coordinates back by dividing $\mathbf{X_{world}}$ with its last element.

5.3 LEDs detection in the Reconstructed Plane

Let S be the set of the candidate points detected in the image and N be the number of LEDs of the ROV. If $size(S)$ is greater than N, we first reconstruct the 3D positions of the points in S using the method described in Section 5.2. Then, we reject the points that are out of the region of interest. In fact, we know a priori the area of the silo in the world coordinate frame. As the ROV can only appear in the silo, we reject from S all the points whose 3D positions are not within the silo area.

Then, we cluster the remaining 3D points into local groups by hierarchical clustering [18]. The distance threshold used for clustering is determined adaptively according to the geometrical configuration of the LEDs. Specifically, the longest distance between two LEDs in our system is 316 mm (LED_2 to LED_1 or LED_2 to LED_3 in Figure 1). We use a threshold of 500 mm (slightly greater than 316 mm) for clustering. After clustering, we remove from S the groups that have either too few or too many points. The groups with too few points consist of isolated light spots while the groups with too many points are reflections of light on the surface of the water.

Finally, we recognize the real set of the LEDs from the retained groups by verifying the internal geometrical characteristics, namely the distances between the points, the shape of the contour and the area. The distance constraint indicates that a plausible candidate should have some "support" points in S that are within the predefined distance tolerance range I_{dist} from it. For example, in our ROV configuration as depicted in Figure 1, a valid candidate for LED_0 should have two support points at about 100 mm and a third support point at about 300 mm. Similarly, a valid candidate for the LED_2 should have two support points at about 316 mm and a third support point at about 300 mm. The points that have no valid supports from other points in S are removed. This process is repeated and S shrinks. After stabilization, if

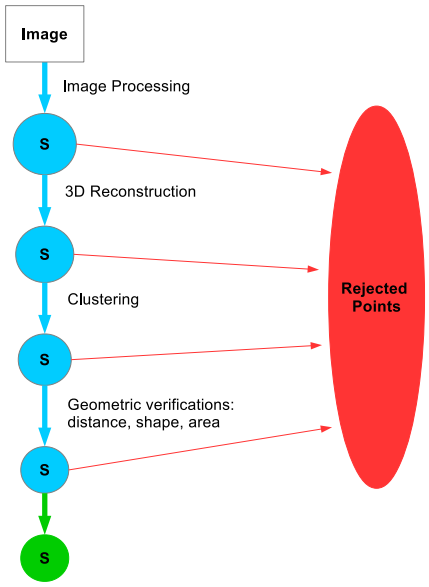

Fig. 6 The selection/rejection cascade to detect the LED points.

$size(S)$ is still greater than N, we decrease I_{dist} by a small amount and restart the shrinkage iteration until I_{dist} reaches the predefined minimum I_{dist_min} or $size(S)$ is less than or equal to N.

For the shape constraint, we first build a reference contour representing the geometrical layout of the LEDs in the ROV coordinate frame. Then for each local group, the points in the group form a candidate contour. We employ the Hu Moments [14] for comparing the shape of a candidate contour with the shape of the reference contour. Contours with comparing scores exceeding the predefined tolerance range, I_{shape}, are rejected. Their corresponding 2D points are removed from S. Similarly, for the area constraint we reject the candidates whose areas are out of the tolerance range, I_{area}, from the reference area. These three kinds of geometrical verification are coherent and complementary. Combining them together can help increasing robustness for recognizing the LEDs. We summarize the point selection process in Figure 6.

5.4 ROV Localization

The points remaining in S have successfully survived the rejection/selection cascade. If $size(S)$ is equal to N, we claim that the target is found. We can determine the order of the points by distance verification. The 3D position of the ROV can then be computed using the method described in Section 5.2.

Conversely, if $size(S)$ is either greater or less than N, we claim that the target is not found in the test image. Further developments can use the motion history of the ROV to help distinguishing the ROV from similar objects when $size(S)$ is greater than N.

5.5 Scalability: Multiple Cameras Fusion for Large Areas

In large areas, multiple cameras are needed to cover the entire terrain or to improve the localization accuracy. The purpose of fusion is twofold. First, the objective is to be able to estimate the pose of the ROV as soon as it is observable by one of the cameras. Second, when the ROV is observable by several cameras, the information is merged to optimize the estimation of its pose. The objective is therefore to obtain a more complete trajectory while avoiding jumps in the estimation of the ROV pose when switching from one camera to another. We treat this requirement as a state estimation and data fusion problem.

Several methods can be used for data fusion applied to localization: histogram filters, Kalman filters and particle filters [27]. Histogram filters only allow a discrete estimation of the state space, which can introduce a loss of precision depending on the chosen discretization step. In addition, their complexity increases exponentially with respect to the sampling step and size of the environment. Kalman filters have the advantage of estimating a continuous state space and are relatively easy to implement. Their main disadvantages are that they do not allow the estimation of a multimodal distribution and that their efficiency is quadratic with respect to the state space to be estimated. When it is necessary to estimate both the pose of a robot and the positions of the landmarks, this can become problematic and we then turn to the particle filter. Particle filters allow the estimation of a continuous state space and a multimodal distribution. In our case, we want to merge the position from the two cameras without having to estimate the map (this is a localization problem and not a SLAM problem), so the Kalman filter is the best compromise between ease of implementation and efficiency. In addition, it has been shown in some studies, for example in [11], that the particle filter is more robust but that the Kalman filter can be more accurate.

Since each camera measures the state of the ROV independently, we use the Kalman filter [15] to fuse the localization results of each camera and to estimate the state of the target. Specifically, at each time step, the Kalman filter predicts the state and each camera provides measurements to update this state. The state that gives the maximum a posteriori probability is used as the state of the target.

6 Experimental Evaluation

In this section, we evaluate the performance of the proposed localization system using different extrinsic calibration methods and different hypotheses on the input data. Quantitatively, we compute the position errors and the orientation errors with respect to the “ground truth” provided by a motion capture system (Vicon).

6.1 Experimental Settings

In order to have a practical evaluation of our algorithms, a dataset has been collected with a reliable ground-truth obtained by a motion capture system. The experiments have been conducted in a room equipped with a Vicon motion capture system composed of 20 Vicon T40S cameras. With such a system, we can assure a 6 DoF (Degrees of Freedom) localization of our system with a sub-millimetric accuracy as demonstrated in [19] and [21] and at a high framerate (500fps).

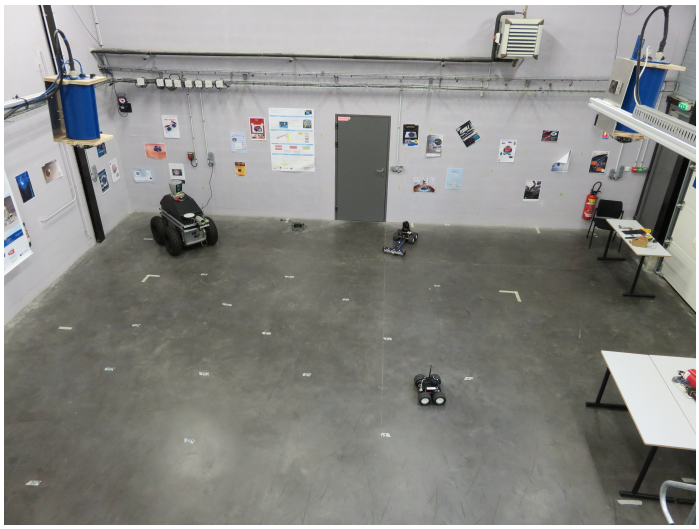

Fig. 7 The validation environment in our laboratory. The two blue cameras on top (about 3.60 meters from the ground) are the fish-eye cameras used for localization.

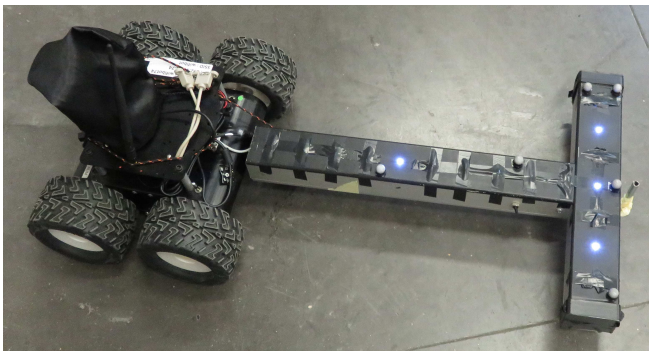

Fig. 8 The model ROV with 4 LEDs used for validation in our laboratory. The ROV is remotely operated thanks to the Wifibot attached to its front.

The cameras used in the experiments are two GEV-B4820M cameras from ImperX. These cameras have a high resolution (4904x3280) and a 3 fps framerate. They are equipped with Samyang 8 mm fisheye lenses which have a 180° diagonal field of view.

We implement the localization system in C++ with the OpenCV library. The validation environment in our laboratory is presented in Figure 7. The two blue cameras on top (about 3.60 meters from the ground) are the fish-eye cameras used for localization. The cameras on the wall are those of the Vicon system that provides the "ground truth" localization. The ROV with its LEDs on is emulated by a remotely operated Wifibot robot on the ground. A closer view of the robot with its 4 LEDs used in our laboratory for algorithm validation is shown in Figure 8. As a reminder, the real ROV in its environment is shown in Figure 3.

Our system takes less than 70 milliseconds for processing one frame when executing on a PC with an Intel Core i5-2520M CPU running at 2.50GHz and 8GB of memory.

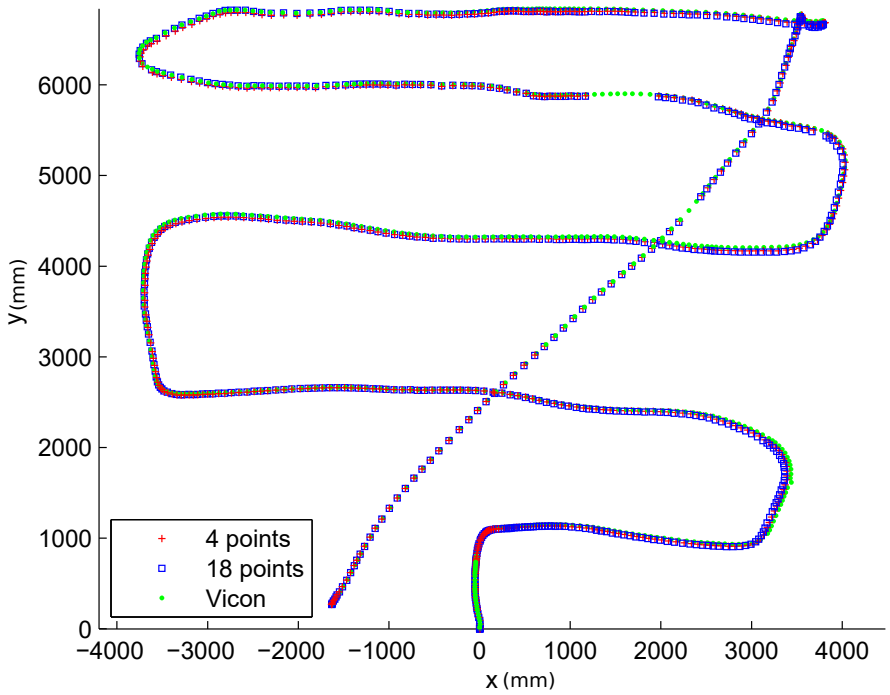

Fig. 9 Estimated trajectories (in red and blue) using different numbers of points with known 3D positions for the calibration step. The ground-truth provided by the Vicon is plotted in green.

Calibration Configurations	Position Error (mm)	Orientation Error (degree)
4 points	21.92 ± 11.42	0.54 ± 0.66
10 points	23.73 ± 15.36	0.54 ± 0.68
18 points	25.81 ± 19.06	0.54 ± 0.67

Table 1 Localization errors using different numbers of points with known positions for calibration

6.2 Calibration using points with known 3D positions

Firstly, we evaluate the calibration method using points with known 3D positions. We place some markers in the room and use the Vicon to get their 3D positions. To understand the influence of the number of points, we tested three configurations: one with 4 points, one with ten points and the last one with with 18 points. The resulting localization errors with respect to the results provided by the Vicon are summarized in Table 1. The estimated and real trajectories for the 4 and 18 points configurations are displayed in Figure 9. We see that as long as the 3D positions of the points are accurate and are geometrically scattered, using more points for calibration does not necessarily improve localization results. On the contrary, noise can be introduced using more points when calculating their pixel locations in the image, especially when the points are near the borders of the image.

Calibration Configurations	Position Error (mm)	Orientation Error (degree)
4 points + 50mm error	83.39 ± 64.13	0.77 ± 0.81
4 points + 100mm error	143.66 ± 109.52	1.37 ± 1.32
4 points + 200mm error	253.60 ± 185.04	4.06 ± 3.26

Table 2 Localization errors using 4 points with noisy 3D positions for calibration.

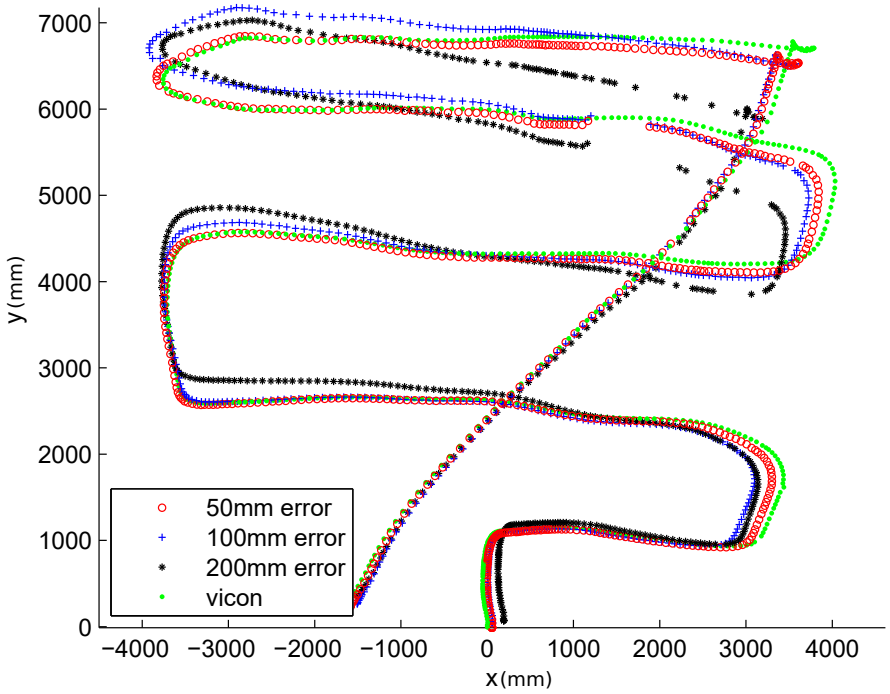

Fig. 10 Estimated trajectories using 4 points with noisy 3D positions for the calibration. The ground-truth is plotted in green.

6.3 Injecting errors to the 3D positions used for calibration

In order to assess the robustness of the localization system with respect to the accuracy of the 3D positions of the points used for calibration, we introduce errors to the positions of the points and recompute the position and orientation errors. Specifically, 50mm, 100mm and 200mm errors are injected to the position of each of the 4 points. The resulting localization errors are summarized in Table 2 and the estimated trajectories are displayed in Figure 10. We see a roughly linear relation between the introduced error and the resulting localization error.

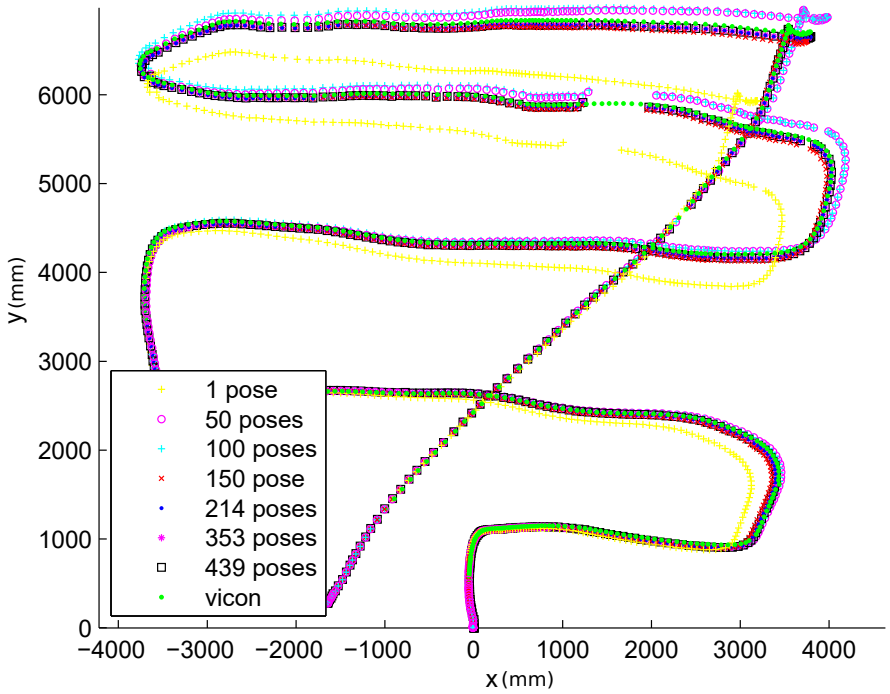

Fig. 11 Estimated trajectories using different numbers of ROV poses for the calibration. The ground-truth is plotted in green.

Calibration Configurations	Position Error (mm)	Orientation Error (degree)
1 pose	306.68 ± 276.71	2.47 ± 1.64
50 poses	58.57 ± 63.46	0.94 ± 0.99
100 poses	67.71 ± 69.88	0.90 ± 0.98
150 poses	49.07 ± 32.13	0.61 ± 0.76
214 poses	31.21 ± 16.32	0.56 ± 0.75
353 poses	28.14 ± 14.89	0.56 ± 0.75
439 poses	27.62 ± 14.85	0.55 ± 0.74

Table 3 Localization errors using different numbers of ROV poses for calibration

6.4 Calibration by moving the ROV

Next, we evaluate the localization performance of the calibration method by displacing the ROV. The localization errors are summarized in Table 3 and the trajectories are displayed in Figure 11. Note that for the "1 pose" case, we assume the equation of the plane to be horizontal, i.e its normal vector is assumed to be $[0\ 0\ 1]^\top$, as we do not have enough points to estimate it. For the other configurations, π is estimated by the RANSAC procedure using the poses of the ROV.

6.5 Fusing the localization results of multiple cameras

Finally, we present the localization results obtained by fusing the results of multiple cameras and compare them with those obtained by a single camera. As shown in Figure 7, we have two fisheye cameras that can perform localization independently at the same time. We have recorded a sequence for evaluation. The localized trajectories are presented in Figure 12, where the advantages of the fusion algorithm can be observed qualitatively.

As we can see, the ROV trajectories estimated by camera 1 or camera 2 are not complete. Indeed, there may be a occultation of the LEDs at certain times, or the system may not be able to estimate the position of the ROV (for example if it is too far from the camera). In the real situation, i.e. for the ROV in its silo as shown in Figure 3, the tether of the ROV can mask the LEDs at certain times, for example when it turns around.

We can see in the figure that the trajectory reconstructed by fusion is much more complete than those estimated by a single camera. The holes in the trajectory are filled by fusion. It is also important to note that the trajectory remains smooth, i.e. there are no jumps when switching from one camera to another or when we use both. This shows that the entire system is correctly calibrated and the estimates are accurate.

6.6 Qualitative evaluation in real conditions

In the previous subsections, we quantitatively assessed the performance of our system under different conditions using a ground truth provided by a motion capture system. These experiments were carried out in our laboratory and it is questionable how robust the system is under real conditions.

The approach developed is very robust to illumination changes. Although the approach was initially developed for a very dark environment (inside the silo), it worked well in our laboratory but also in a test silo. This test silo is a basin in a shed with a transparent roof. The system is therefore subjected to large lighting variations depending on the time of day and weather conditions. As we can see in Figure 3, there are reflections of the sun directly in the basin and shadows. The system has been in operation for more than a year now without any failure. This is due to the use of high power LEDs that represent strong light sources directly visible to the camera. The filtering algorithms explained in the article also make it possible to reject potential false detections.

7 Conclusion

In this paper, we proposed a system and the associated algorithms to accurately localize a mobile in a large industrial environment, whether it is a mobile robot, an AGV, or in our case an ROV. This system relies on the use of a fisheye camera observing LEDs placed on the mobile. We proposed two methods for calibrating the system: the first one uses known points in the scene, the second one is an auto-calibration method. We have studied the accuracy of the system as a function of the calibration, and by varying several parameters as the number of points used, the introduction of errors in the coordinates of the points, etc. We have shown that the number of points used does not matter, but it is important that the points are known accurately. The auto-calibration method makes it possible to overcome this constraint. We have proved both quantitatively thanks to a motion capture system in the laboratory and qualitatively on the actual site that our system makes it possible to localize a mobile with a

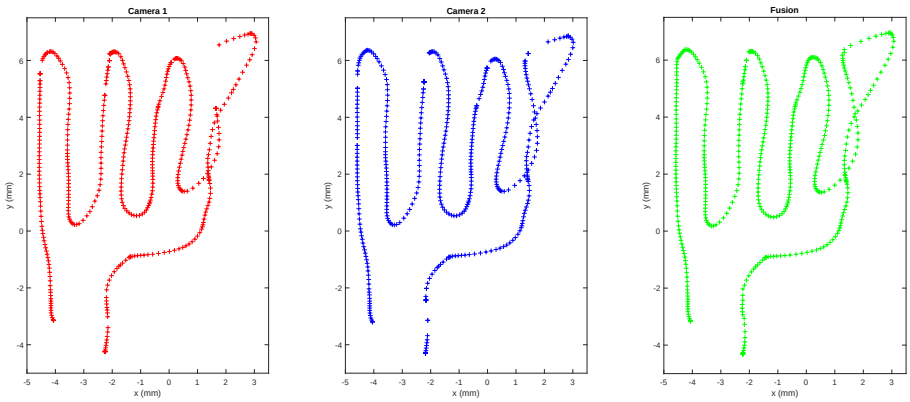

Fig. 12 Fusing the results of multiple cameras. Left: the estimated trajectory obtained by Camera 1. Middle: the estimated trajectory obtained by Camera 2. Right: the trajectory obtained by fusing the results of both cameras. As we can see, this method fills the holes in the trajectories while keeping a smooth trajectory.

high accuracy (up to 22mm). In our experiments, we successfully localized a mobile robot in our laboratory and an ROV operating in an industrial environment.

References

1. Barreto, J.: A unifying geometric representation for central projection systems. Computer Vision and Image Understanding (CVIU) **103**(3), 208–217 (2006)
2. Basu, A., Licardie, S.: Alternative models for fish-eye lenses. Pattern Recognition Letters (PRL) **4**(16), 433–441 (1995)
3. Boutteau, R., Savatier, X., Ertaud, J., Mazari, B.: An omnidirectional stereoscopic system for mobile robot navigation. Sensors and Transducers Journaland Transducers Journal, Special Issue on Robotic and Sensors Environments **5**, 3–17 (2009)
4. Boutteau, R., Savatier, X., Ertaud, J., Mazari, B.: Mobile Robots Navigation, chap. Chapter 1 : A 3D Omnidirectional Sensor For Mobile Robot Applications, pp. 1–23. In-Tech (2010). ISBN : 978-953-307-076-6
5. Boutteau, R., Sturm, P., Vasseur, P., Demonceaux, C.: Circular laser/camera-based attitude and altitude estimation: minimal and robust solutions. Journal of Mathematical Imaging and Vision (JMIV) **60**(3), 382–400 (2018)
6. Brauer-Burchardt, C., Voss, K.: A new algorithm to correct fish-eye-and strong wide-angle-lens-distortion from single images. In: Proceedings. International Conference on Image Processing (ICIP), vol. 1, pp. 225–228 (2001)
7. Courbon, J., Mezouar, Y., Eckt, L., Martinet, P.: A generic fisheye camera model for robotic applications. In: IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 1683–1688. IEEE (2007)
8. Courbon, J., Mezouar, Y., Martinet, P.: Evaluation of the unified model of the sphere for fisheye cameras in robotic applications. Advanced Robotics **26**(8-9), 947–967 (2012)
9. Fitzgibbon, A.: Simultaneous linear estimation of multiple view geometry and lens distortion. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR), vol. 1, pp. 125–132 (2001)
10. Geyer, C., Daniilidis, K.: A unifying theory for central panoramic systems and practical implications. In: European Conference on Computer Vision (ECCV), pp. 445–461. Springer, Berlin, Dublin, Ireland (2000)
11. Gutmann, J., Burgard, W., Fox, D., Konolige, K.: An experimental comparison of localization methods. In: IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 736–743 (1998)
12. Hartenberg, R., Denavit, J.: A kinematic notation for lower pair mechanisms based on matrices. Journal of Applied Mechanics **77**(2), 215–221 (1955)

13. Hartley, R., Zisserman, A.: Multiple View Geometry in Computer Vision, second edition edn. Cambridge University Press (2004)
14. Hu, M.: Visual pattern recognition by moment invariants. IRE Transactions on Information Theory **8**(2), 179–187 (1962)
15. Kalman, R.: A new approach to linear filtering and prediction problems. Transactions of the ASME - Journal of Basic Engineering **82**(Series D), 35–45 (1960)
16. Lepetit, V., Moreno-Noguer, F., Fua, P.: Epnp: An accurate o(n) solution to the pnp problem. International Journal of Computer Vision (IJCV) **81**(2), 155–166 (2009)
17. Levenberg, K.: A method for the solution of certain problems in least squares. Quarterly of Applied Mathematics **2**, 164–168 (1944)
18. Maimon, O., Rokach, L.: Data Mining and Knowledge Discovery Handbook. Springer US (2010)
19. Manecy, A., Marchand, N., Ruffier, F., Viollet, S.: X4-mag: A low-cost open-source micro-quadrotor and its linux-based controller. International Journal of Micro Air Vehicles (IJMAV) **7**(2), 89–110 (2015)
20. Mei, C., Rives, P.: Single view point omnidirectional camera calibration from planar grids. In: IEEE International Conference on Robotics and Automation (ICRA), pp. 3945–3950 (2007)
21. Merriaux, P., Dupuis, Y., Boutteau, R., Vasseur, P., Savatier, X.: A study of vicon system positioning performance. Sensors **17**, 1591 (2017)
22. Merriaux, P., Dupuis, Y., Boutteau, R., Vasseur, P., Savatier, X.: Robust robot localization in a complex oil and gas industrial environment. Journal of Field Robotics (JFR) **35**(2), 213–230 (2018). DOI 10.1002/rob.21735
23. Mur-Artal, R., Tardós, J.: Orb-slam2: an open-source slam system for monocular, stereo and rgb-d cameras. IEEE Transactions on Robotics **33**, 1255–1262 (2017)
24. Nobili, S., Dominguez, S., Garcia, G., Martinet, P.: 16 channels velodyne versus planar lidars based perception system for large scale 2d-slam. In: 7th Workshop on Planning, Perception and Navigation for Intelligeng Vehicles (2015)
25. Shah, S., Aggarwal, J.: Intrinsic parameter calibration procedure for a (high-distortion) fish-eye lens camera with distortion model and accuracy estimation. Pattern Recognition **29**(11), 1775–1788 (1996)
26. Sturm, P., Ramalingam, S., Tardif, J., Gasparini, S., Barreto, J.: Camera models and fundamental concepts used in geometric computer vision. Foundations and Trends in Computer Graphics and Vision **6**(1-2), 1–183 (2011)
27. Thrun, S., Burgard, W., Fox, D.: Probabilistic robotics. MIT Press (2005)
28. Wang, R., Schworer, M., Cremers, D.: Stereo dso: Large-scale direct sparse visual odometry with stereo cameras. In: IEEE International Conference on Computer Vision (ICCV), pp. 3903–3911 (2017)
29. Ying, X., Hu, Z.: Can we consider central catadioptric cameras and fisheye cameras within a unified imaging model. In: European Conference on Computer Vision (ECCV), pp. 442–455. Springer (2004)
30. Zhang, J., Lyu, Y., Roppel, T., Patton, J., Senthilkumar, C.: Mobile robot for retail inventory using rfid. In: IEEE International Conference on Industrial Technology (ICIT), pp. 101–106 (2016)