IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received December 23, 2019, accepted January 2, 2020, date of publication January 6, 2020, date of current version January 15, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2964317

A Novel Method to Estimate the Position of a Mobile Robot in Underfloor Environments Using RGB-D Point Clouds

CRISTOBAL PARRA[1], SERGIO CEBOLLADA[1], LUIS PAYÁ[1], MATHEW HOLLOWAY[2], AND OSCAR REINOSO[1], (Senior Member, IEEE)

[1]Department of Systems Engineering and Automation, Miguel Hernandez University, 03202 Elche, Spain
[2]School of Design Engineering, Imperial College London, London SW7 1AL, U.K.

Corresponding author: Luis Payá (lpaya@umh.es)

This work was supported by the Spanish government through the project DPI 2016-78361-R (AEI/FEDER, UE): ''Creación de mapas mediante métodos de apariencia visual para la navegación de robots''; by the Generalitat Valenciana through the project AICO/2019/031: ''Creación de modelos jerárquicos y localización robusta de robots móviles en entornos sociales'' and by the European Union's Horizon 2020 research and innovation program under grant agreement No. 698607.

ABSTRACT This paper is focused on the design of a mobile robot whose objective is to apply thermal insulation spray in underfloor voids, to improve the energy efficiency of buildings. Solving robustly the mapping and localization problems is crucial to achieve a high degree of autonomy during the development of this task. Nevertheless, underfloor voids constitute specially challenging environments mainly owing to the extreme unevenness of the terrain and the changes the environment experiences as the insulation process is carried out. Taking these issues into account, this work presents the implementation of the localization module of the robot, which is equipped with a laser scanner and an RGB (Red, Green and Blue) camera. The data captured by both sensors is combined to build point clouds that describe the appearance of the environment. While the robot traverses the *a priori* unknown environment, several point clouds are built and an alignment between each pair of consecutive clouds is carried out. From this information, the current position of the robot is estimated with respect to the previous one. The method has been tested with several datasets captured in real underfloor environments (building crawl spaces) and under real operating conditions.

INDEX TERMS Building crawls, global appearance descriptors, localization, point clouds, registration.

I. INTRODUCTION

Along the past few years, the use of mobile robots has extended to a wide range of applications thanks mainly to the improvement of their perception and processing abilities. Numerous examples can be found in the literature, such as in search and rescue applications [1], [2], social assistive robots [3], mobile manipulation [4], navigation in densely crowded environments [5] and material handling in manufacturing systems [6].

The present work is part of a wider project whose objective is improving the energy efficiency of those buildings which have voids between floor and foundations, which are relatively common in many areas in Europe and around the world due to building methods [7]. The energy efficiency of such buildings can be improved through under-floor insulation. However, gaining access to such voids is quite difficult or even impossible for people and thus, traditional methods to perform the insulation are very disruptive for the occupants of the building because they require removing floorboards, applying rigid panels or rolls of insulation and putting everything back together. Considering this, Holloway *et al.* [8] developed a small mobile robotic platform which is designed to make the insulation process quicker and less disruptive. The present work is based on the autonomous surveying robot architecture presented in this reference, and it is specifically developed to adapt to the particularities of this vehicle. The robot is equipped with all the necessary actuators to move and spray foam insulation in the underfloor voids. Initially, this task can be performed successfully in a tele-operated way, driven by an expert operator who recognizes the environment, supervises the task and makes decisions about the trajectory of the robot and the ejection of foam insulation [9]. Notwithstanding that, increasing the degree of autonomy of

The associate editor coordinating the review of this manuscript and approving it for publication was Zhenhua Guo.

the robot would improve the performance and speed while reducing cost. This is the main motivation of the work.

Mapping and localization are two key abilities that a mobile robot must have to carry out the task it has been designed for in a truly autonomous way. The map must be both complete and compact to make it possible that the robot performs its task accurately and with a reasonable computational time. This way, when the robot starts moving through one *a priori* unknown environment, it should create a model of this environment and estimate its position making use of this model. This is a classical problem in robotics known as SLAM (*Simultaneous Localization And Mapping*). It is considered a fundamental challenge in the robotics field because estimating the position of the robot requires having an accurate model of the environment and building a model requires knowing accurately the position of the robot. Therefore, both the robot trajectory and the model need to be estimated and updated simultaneously. A variety of SLAM algorithms [10] have been proposed in the literature, using different kinds of sensors, either range sensors (such as laser-range scanners [11]), vision sensors (such as monocular cameras [12], stereo cameras [13] or omnidirectional cameras [14], [15]), or even a fusion of both kinds [16].

As far as vision based SLAM is concerned, in order to build a 3D map of the surrounding environment, relevant information must be extracted from images. Therefore, methods based on local appearance, such as FAST, Harris corners, SIFT or SURF [17], are widely used to detect and/or describe the visual information. These local features or landmarks are detected in each frame and then matched along a sequence of consecutive frames considering their visual descriptors. If this process is not robust or many outliers are present while the features are matched, the SLAM algorithm may fail and not converge to a correct map. This is a problem that typically arises in unstructured and changing environments, and those which are prone to present visual aliasing.

More recently, RGB-D (Red, Green, Blue and Depth) sensors, such as Microsoft Kinect, have received much attention in a number of works related to mapping, segmentation and recognition [3], [18]. These sensors provide the robot with both color and depth information from the environment and they present a relatively low cost and weight. In this project, an RGB-D acquisition system is used and a novel approach has been implemented to solve the localization problem, using the data captured by this system, when the robot moves within complex environments which are especially challenging. This kind of sensor permits combining and exploiting the synergies of both the metric information provided by the 3D depth measurements and the complete and varied information provided by the images.

As pointed out before, this work has been motivated by the need of estimating the position of the robot in real indoor environments, such as underfloor voids, where human access is difficult or even impossible. Consequently, mobile robots constitute a powerful solution to carry out this kind of task autonomously. Such environments present some difficulties that make the localization especially challenging when traditional state-of-the-art methods are used [19], [20].

Among these complexities, four can be highlighted. First, underfloor voids tend to be very unstructured and the lack of recognizable objects in the scene hinders the feature extraction and matching processes. Second, the terrain tends to be extremely uneven due to the presence of construction remains and debris which can even change their position as the robot moves, modifying the geometry of the environment. Third, the lack of light makes it difficult to extract robust features from the scenes and the installation of artificial light sources may produce severe shadows. Fourth, since the robot ejects foam insulation onto the bottom of the floor, the appearance of the environment significantly changes while it moves. Taking these issues into account, the combination of visual and depth data may be especially interesting to extract robust information from the environment and it is the basis of the approach proposed in this work. As the robot moves through the initially unknown environment, some point clouds (or local maps) that contain both kinds of information are captured and an alignment between them is calculated to estimate the position of the robot in the environment.

In the present work, the mobile robot is equipped with an RGB-D sensor that captures data while the robot follows a trajectory that covers the environment to model, which is initially unknown, and a novel framework is proposed to estimate this trajectory. The trajectory is defined as a set of adjacent poses traversed consecutively by the robot, and from each pose the robot captures a set of points (depth information) and a set of images that cover a field of view of 360 degrees around the robot. In the present work, the distance between consecutive poses is relatively high, so the pose estimation algorithm must work well considering this additional constraint. This is mainly due to the fact that the data acquisition process is quite time consuming, as Julia *et al.* [9] show. We propose an algorithm which estimates each new pose Q with respect to the previous one P using a registration approach. However, considering that the target environments to model in this work are especially challenging, as pointed out in the previous section, and the distance between consecutive poses is expected to be relatively high, the preliminary experiments presented a large number of incorrect matches, what leaded to high localization errors.

To try to overcome these problems, the visual information is used initially to achieve robust matches. Firstly, every image of the first pose is paired up with the most similar image in the second pose using a global-appearance approach. Secondly, SURF features are detected [21], [22], and matches are searched within the previously paired-up images. Finally, the matched keypoints are identified into the two point clouds data, and the transformation matrix is calculated using only these previously selected points. An SVD-based estimation of the transformation is carried out to align these selected points. Therefore, using this transformation matrix, the second pose of the robot can be estimated with respect to the first one. The previous selection of visual information is expected to provide robust points to match during the registration process.

The remainder of this paper is organized as follows. The second section makes a brief review of the state-of-the-art techniques. After that, the third section presents a detailed description of the acquisition system. Next, the mapping and localization algorithms are described in depth in the fourth section and their results are discussed. Finally, the conclusion and future research lines are outlined.

II. RELATED WORK

Building a map of an unknown environment autonomously is a problem which has received a great deal of attention from the computer vision and robotics research communities. In this field, the SLAM problem is considered a core question and many algorithms have been proposed to address it [23], [24].

When visual information is used to extract the necessary information from the environment, choosing the optimal features' extraction and description algorithms is one of the most important issues that have an impact upon the convergence of the algorithm [25]. A number of algorithms on this topic can be found in the literature, such as the work presented by Gil *et al.* [22], who performed a comparative evaluation of different local features detectors and descriptors considering a variety of circumstances such as changes in the scale, point of view and lighting conditions. Their experiments concluded that SURF presents some characteristics that make it a good choice to solve mapping and localization tasks. Nevertheless, most of such comparisons and improvements are significantly relevant to specific data or environment types in such a way that a specific description method may perform successfully in some cases but not as properly as expected under different conditions. Therefore, these conclusions must be taken as specific suggestions that may not be applicable to solve any general case.

Due to the emergence of new 3D sensors such as RGB-D cameras or 3D laser scanners, new possibilities have appeared in 3D mapping. RGB-D sensors directly provide depth information and color images and the combination of both kinds of information permits building point clouds that describe the environment around the robot in great detail. Considering these representations of the environment, many modern approaches apply the ICP (Iterative Closest Point) algorithm in their works on mapping and localization, such as those developed by Rusinkiewicz and Levoy [26] or Segal *et al.* [27]. In these works, the robot goes through the environment to map (either using any exploration algorithm or in a tele-operated way) and captures some sets of point clouds from several positions. If the initial position is known, the position of the robot when each new observation is captured can be estimated by means of an alignment between the current and the previous point cloud. Once these positions are known, all the observations or local maps can be combined to obtain a global map of the environment. Tiar *et al.* [28] also present a method based on ICP for local mapping, with the objective of solving the problem of SLAM. They use the data provided by a laser system to recognize the environment. Cho *et al.* [29] propose the use of ICP matching methods based on line features and compare it with the classical ICP method using feature-point based SLAM, achieving better results.

The ICP algorithm was first proposed in 1990 [30], and since then, a great number of variants that try to improve the performance of the algorithm have been published, such as GICP (Generalized-ICP) [27], 3D-NDT [31] and AICP (Adaptive ICP) [32], which are widely used in the splicing and registration processes. On the one hand, the inputs of ICP are two point clouds captured from two poses (position and orientation). The 3D position of the points in the clouds is known (it can be calculated using the depth values of the RGB-D camera). On the other hand, the output of ICP is a transformation matrix that defines the relative rotation and translation of the robot between the two poses. Pomerleau *et al.* [33], [34], present a comparison between ICP variants, considering a broad range of input data. However, when this kind of standard ICP variants are used in the target environments (building crawls), in which small overlapping may exist between different areas, the results are not successful.

Therefore, traditional ICP algorithms tend to perform well and provide effective alignments when the two point clouds to compare are relatively similar. However, when significant differences between the clouds exist, a good initial estimation is needed so that the ICP algorithm converges to a proper transformation matrix. If no estimation is available, traditional ICP algorithms are prone to fail under these circumstances and some of the robot poses may not be estimated with enough accuracy. As a result, the robot is expected to have many difficulties in creating the map and estimating consecutively each new position Q with respect to the previous one P. More recent works try to cope with such difficulties including some additional constraints in the algorithm. For example, Feng *et al.* [35] present an algorithm to detect multiple planes in 3D point clouds. They construct a graph of planes in the point cloud, whose nodes and edges represent a group of points in the plane and their neighbourhood relations, respectively. Such graph can be used to refine the registration process. Also, Grant *et al.* [36] first extract planes from the point clouds and second point features are detected and matched within pairs of corresponding planes.

Some research works propose using visual features to complement the data. Khoshelham *et al.* [37] present an epipolar search method for accurate transformation of the keypoints from 2D to the 3D space, achieving more accurate 3D correspondences. Also, Yousif *et al.* [38] present a framework that concatenates the estimated camera transformation between sequential frames obtaining a global camera pose estimated with respect to a fixed reference frame in environments which are dark and with poor illumination. Some authors, such as Kim *et al.* [39], identify geometric correspondences among a series of scans to extract an initial alignment. After that, they compute the final alignment using the overlapped area, through a standard ICP algorithm. These methods are tested in outdoor construction environments, in which the visual features are much more distinctive that those obtained from the building crawls. Also, Xin *et al.* [40] present a SLAM system for indoor office environments, from a point cloud acquired with a Kinect sensor. First, depth images and RGB images are captured by the sensor. Then, feature detection and matching are carried out, using the ORB algorithm. With these matched points, and considering the depth information, a PROSAC (PROgressive Sample Consensus) algorithm is used to get more accurate inlier matching points in motion transformation estimation. Our proposal follows a similar philosophy, but adapting it to the complexities of the target environments, where better results have been achieved by using SURF features, selecting only the most robust matches, and obtaining their depth information to obtain the final point clouds that are used to carry out registration. Pandey *et al.* [41] present an automatic registration of 3D point clouds using visual features and depth information. They employ visual features to establish an initial correspondence between the two poses (seed transformation). This initial estimation is subsequently refined by means of the ICP framework, using all the points of the original point clouds. In the present work, only the points obtained after matching visual features are used to obtain the final point clouds, and the registration process is carried out with the clouds composed by these points, once the depth information has been included. The present work presents an additional complexity because a conventional camera mounted on a rotating turret is used to acquire the visual information, so it is necessary to perform a previous pairing-up between the images obtained from both poses, as the next section describes.

Therefore, in the present paper, we develop a method to solve the registration problem in building crawl spaces robustly. The method tries to overcome both issues: (a) the complexities of underfloor voids and (b) the large distance between consecutive poses. The main contribution of this paper is the development of a novel framework to improve the performance of the registration algorithm, specifically developed considering the characteristics of the robot [8] and the acquisition system and acquisition process [9]. The method performs a drastic reduction of the number of points in the original clouds, by means of a visual algorithm that only selects points that lead to robust matches in a set of images which have been previously paired-up through their global-appearance. The registration is subsequently performed with these clouds that contain a reduced number of points.

III. DESCRIPTION OF THE DATA ACQUISITION SYSTEM

The system is composed of a small mobile robotic platform with a 3D scanner mounted on it. This scanner is constituted by a monocular camera and a 2D-laser scanner whose beams are contained in a vertical plane. The camera and the vertical laser are mounted on a turret, which can rotate 360° around the x-axis of the robot reference system. Fig. 1 shows the robotic platform with the two sensors, and the robot frame of reference used along the work.

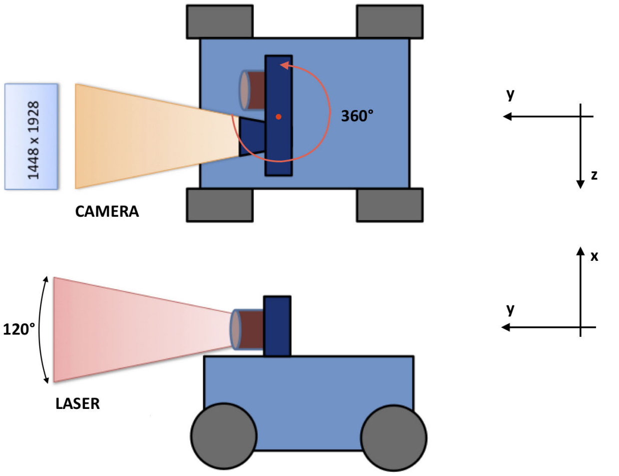


FIGURE 1. Top and side view of the robotic platform with the 3D-scanner mounted on its top. It is composed of a monocular camera and a vertical laser mounted on a turret. The robot frame of reference is depicted.

To acquire complete visual and range information from the surroundings of the robot, the turret spins a whole revolution while both sensors capture data. During this process the robot is motionless at a position P. After that, the depth and color data captured by the two sensors are assembled into a 3D textured point cloud $\mathcal{M}_P$. The next subsections give more details about this operation.

A. 3D SCANNER SYSTEM

The turret rotates by means of a stepper motor which has N even steps, what implies having a resolution equal to $360/N$ degrees. The vertical laser provides a set of M range readings, ρ_i, $i = 1, \ldots, M$, from a set of angles θ_i, $i = 1, \ldots, M$, measured in the laser reference frame, which cover a complete field of view of 120°. These readings can be expressed as 3D points in the laser reference frame $p_i^{[L]} = [\rho_i \cos\theta_i, \rho_i \sin\theta_i, 0]^T \in \mathbb{R}^3$, $i = 1, \ldots, M$, where the superscript $[L]$ indicates that the coordinates of this point are expressed with respect to the laser reference system. The maximum number of points that can be captured during a complete rotation of the turret is equal to $M \times N$. They constitute a point cloud of distance readings that can be expressed in the robot frame of reference $[R]$ using eq. 1.

$$p_{i,j}^{[R]} = R(\Phi_j) T_{laser} p_{i,j}^{[L]}; \quad i = 1, \ldots, M;\ j = 1, \ldots, N \tag{1}$$

where T_{laser} is the transformation matrix from the laser frame of reference to the robot frame of reference (obtained after a calibration process) and $R(\Phi_j)$ is the rotation matrix that expresses that the turret has rotated an angle Φ_j around the x-axis of the robot frame. In this work, the number of steps of the stepper motor that rotates the turret is $N = 2400$ steps per revolution and the number of range readings provided by the laser sensor is $M = 334$ range readings per step.

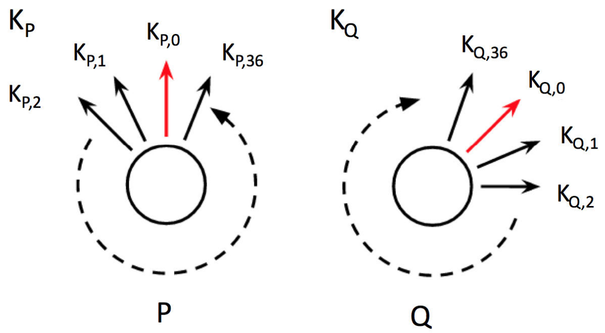


FIGURE 2. From each position *P*, *Q* of the robot, the image acquisition system provides 37 images captured from a set of evenly distributed orientations that cover the full circumference.

Simultaneously to the acquisition of the laser data, 37 RGB images are acquired by the camera while the turret rotates a complete revolution, with orientations evenly distributed around the x-axis of the robot frame. The number of 37 RGB images per pose is chosen to produce an overlapping at low capture distances of, at least, two thirds of the image between consecutive captures. A significant overlapping is necessary so that the pairing-up algorithm described in section IV-A works correctly despite the shadows produced by the artificial light sources. These images are $K_P = \{K_{P,0}, K_{P,1}, \ldots, K_{P,36}\} = \{K_{P,l}\}, l = 0, \ldots, 36$, where P refers to the position where the robot is while the turret rotates (fig. 2). The orientation of the first capture and the direction in which the turret spins may change between two different capture positions P and Q. The resolution of each image is $N_x \times N_y = 1448 \times 1928$ pixels. The system is fully calibrated, in such a way that an RGB value can be associated to the majority of the points that compose the point cloud [9]. To carry out this task, first, every point in the cloud is projected onto the images and subsequently, using a subpixel mapping process, the color value of the point is estimated from the information contained in the set of acquired images. It is important to highlight the fact that after this process, some of these 3D points of the cloud may not have any color value associated. This happens when they fall out of the field of view of the images.

A 3D point $p_{i,j}^{[R]}$ expressed in the robot frame can be projected to image coordinates $u \in \Omega \subset \mathbb{R}^2$ using eq. 2. To do that, it is necessary to consider the angle Φ_l that the turret has rotated when the image $K_{P,l}$ is acquired.

$$u_{\{i,j\},l} = \pi\left(CT_c^{-1}R_{\Phi_l}^T p_{i,j}^{[R]}\right) \tag{2}$$

where R_{Φ_l} is the rotation matrix corresponding to the angle Φ_l the turret has when the image $K_{P,l}$ is acquired, T_c^{-1} is the transformation matrix that defines the calibrated camera pose with respect to the robot frame of reference, C is the calibrated camera matrix, and $u = \pi(x)$ is a function that performs the dehomogenisation of $x \in \mathbb{R} = (x, y, z)$ in order to obtain the image coordinates. The same point $p_{i,j}^{[R]}$ can be projected onto different image coordinates depending on the angle Φ_l.

Additionally, as a part of the localization method, it has been necessary to implement an algorithm to estimate the depth of an image pixel. This is a relevant part of our implementation that constitutes a crucial part of the alignment algorithm, as described in the next section. Using the method described in the previous paragraphs, when evaluating the pixels of a particular image $K_{P,l}$, there is no directly available depth information for all of them, since the angular resolution of the camera is higher than the laser's. Considering this, the next steps are proposed to estimate the depth associated to a specific pixel in the image $K_{P,l}$:

1) Projecting every point of the cloud $\mathcal{M}_P$ onto the images $K_{P,l},\ l = 0, \ldots, 36$. Equation 2 can be used to project all the points and obtain image coordinates. All points which lie outside the limits of the images or which are projected from behind the scene are discarded. The rest of the points are saved on an array with the tuples $\{u_{n,l}, d_{n,l}\}$, where $d_{n,l}$ is the z coordinate of the point n in camera coordinates for image $K_{P,l}$.
2) Searching for adjacent pixels within a specific radius. A kd-tree structure is used with this aim.
3) Interpolating depths. The depth value of the target pixel is calculated by means of an interpolation using a Gaussian filter with the neighbouring pixels' depth.

In this way it is possible to estimate the depth of the pixels of a specific image. This feature is necessary in the method proposed in the next section to carry out the alignment.

After this process, the result is a local map of the environment, captured from the position P. This map contains the point cloud $\mathcal{M}_P$ which includes RGB information and the set of images $K_{P,l},\ \ l = 0, \ldots, 36$ that include depth information.

IV. VISUAL ALIGNMENT

In this work, the localization process is addressed as a problem to align the information captured from two consecutive poses P and Q. This section describes the method we propose to select robust points from the point clouds $\mathcal{M}_P$ and $\mathcal{M}_Q$ acquired from two consecutive poses P and Q as the robot goes through the underfloor environment. Previous research works have presented some proposals that make use of the Iterative Closest Point (ICP) algorithm to calculate such alignments [42]. In this work, a novel alternative is proposed, which estimates this alignment using mainly the information provided by the images.

Therefore, the objective of this section is to estimate the transformation matrix T_{PQ} between the poses P and Q. If the pose P is known, then, once T_{PQ} has been calculated, the pose Q can be estimated and integrated into the model, along with the point cloud captured from it.

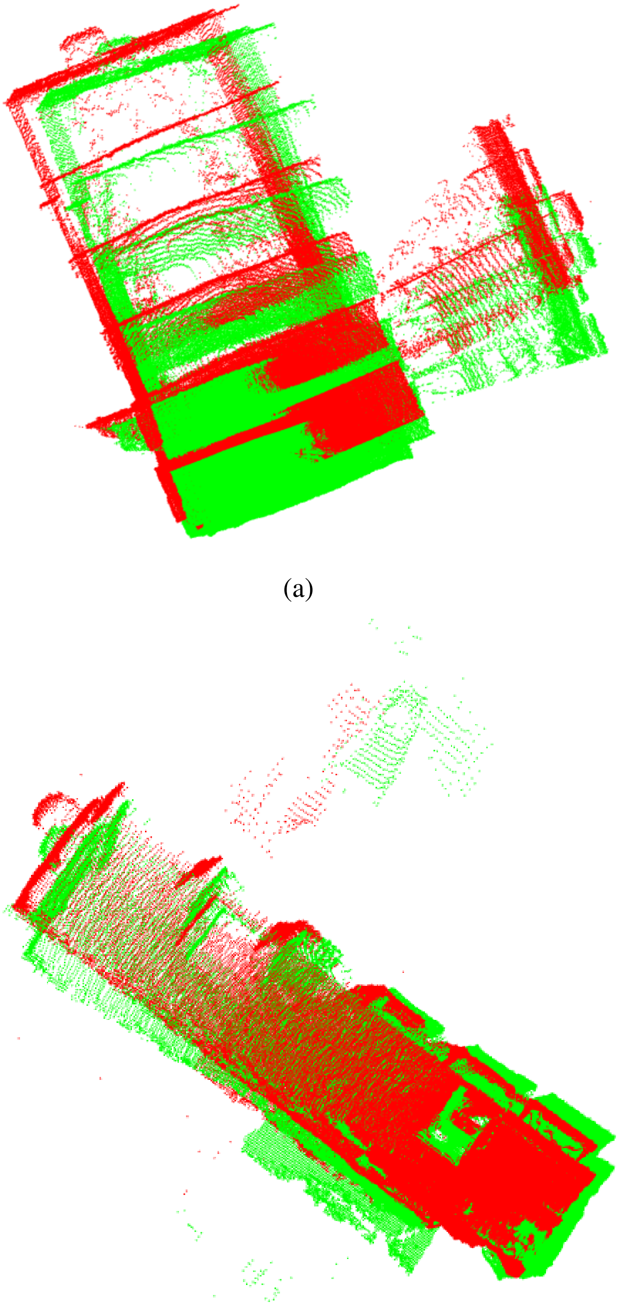
One possibility to estimate the visual alignment is to find the transformation $T_{PQ} \in \mathbb{SE}_3$ that minimizes the error function:

$$E(T_{PQ}) = \sum_{m=1}^{N_{PQ}} \left((T_{PQ} \cdot \mathcal{M}_P(m) - \mathcal{M}_Q(m)) \cdot \vec{n}_{\mathcal{M}_P}\right)^2 \tag{3}$$

where $\mathcal{M}_P(m)$ and $\mathcal{M}_Q(m)$ represent the m-th corresponding point of the clouds $\mathcal{M}_P$ and $\mathcal{M}_Q$ after ICP. Also, N_{PQ} is the number of correspondences between both point clouds, and $\vec{n}_{\mathcal{M}_P}$ is the normal vector to point. This minimization is carried out through an iterative linearisation process [43]. In this method, the speed and convergence of the algorithm strongly depend on the reliability of the initial estimation for the transformation matrix and the robustness of the correspondences. Also the number of points included in the point cloud has a great influence upon the final result (transformation matrix T_{PQ}).

Considering the difficulties of building crawl spaces, as presented in previous sections, we propose introducing some changes to the general ICP method in this paper. Preliminary experiments have confirmed the negative impact of these complexities upon the estimation of the transformation matrix. Additionally, the large distance between two consecutive poses P and Q which typically exists in the trajectories produces many mistakes if all the $3D$ points are considered. It leads to excessive errors in the alignment and high computing time. As an example, fig. 3 shows the result of aligning two point clouds using the traditional ICP algorithm. The two clouds were acquired from two consecutive poses in a real operating environment, $\mathcal{M}_P$ in green color and $\mathcal{M}_Q$ in red color. The alignment is clearly unsuccessful and these are the typical results obtained when using raw ICP in underfloor voids and with large distances between consecutive poses. Taking these facts into account, the proposal we present tries to obtain a more robust set of matches. This reduced set of matched points will be used subsequently to estimate the alignment matrix between both point clouds. The visual information acquired by the camera is used to obtain robust matches. Considering this, the method proposed in this paper to estimate T_{PQ} consists of the following steps:

1) *Pairing up the images in the set $K_{P,l}$ with the images in the set $K_{Q,m}$, $l, m = 0, \ldots, 36$.* For each image in the first set, the most similar image in the second set is calculated and paired up with the first image. It is necessary to take it into account that the turret performs a uniform and constant rotation of 360° while the images of each set are captured.
2) *Matching of visual features*. For each of the pairs of images resulting from the previous step, visual features are extracted and described, and correspondences between these features are established. As a result, each pair of images provides us with a list of visual correspondences.
3) *Performing the alignment through depth information*. The depth of each corresponding point is calculated using the method presented in subsection "Depth information extraction and alignment". As a result, two new point clouds with a significantly reduced number of points are generated. These point clouds are expected to provide a robust alignment because they are built using only points that have proved to have a reliable correspondence. This step finishes with the estimation of the transformation matrix T_{PQ}.


(a)

(b)

FIGURE 3. Sample results of the alignment of two point clouds acquired from a real operating environment, using raw ICP. The points in the cloud $\mathcal{M}_P$ are shown in green color and the points in $\mathcal{M}_Q$ are shown in red. (a) Bird's eye view and (b) lateral view.

The complete algorithm is represented in fig. 4. Also, each of the three steps is detailed in the following subsections.

A. MATCHING IMAGES THROUGH GLOBAL APPEARANCE

The objective of the first step consists in pairing up the images captured from the first position P with the images

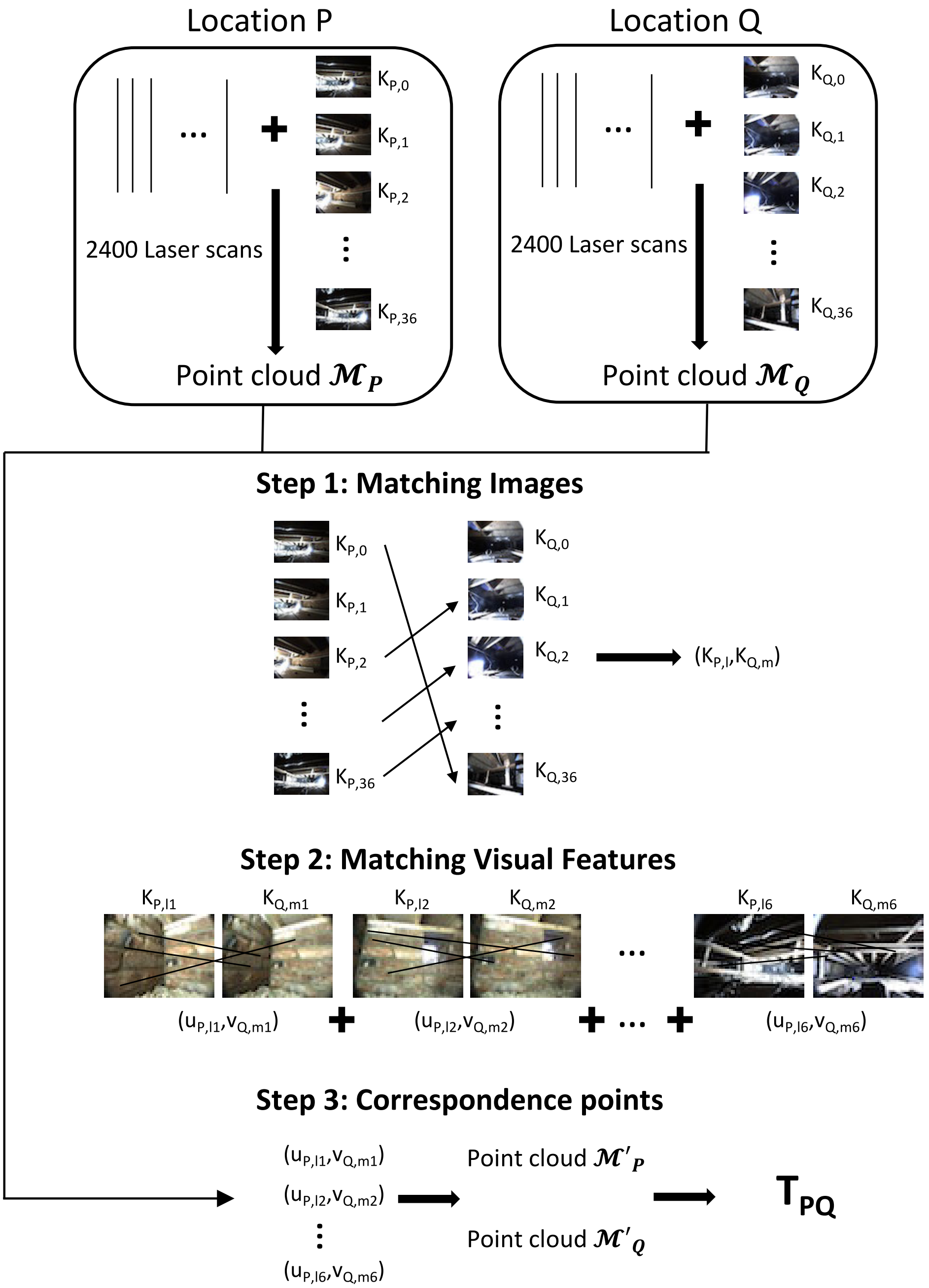


FIGURE 4. Schematic overview of the alignment algorithm. Initially, the images of both sets are paired up. After that, visual features are extracted and matched considering each pair of images. Finally, 3D points are generated from the matches and the transformation T_{PQ} is obtained.

captured from the second position Q. Once the robot has moved from P to Q, it is not possible to know with accuracy which orientation the robot has when the acquisition process starts from Q. This is due to the fact that the odometry of the robot is not reliable in this kind of underfloor environments, since it tends to accumulate large errors. The reason is that, very often, these environments contain debris on the floor, which may cause that the robot sleeps or even that the floor changes as these little objects may be moved. As a consequence, it is necessary to implement an algorithm that makes use of the visual information to find out which image in the set $K_{Q,m}$ is most similar to each image in the set $K_{P,l}$, $l, m = 0, \ldots, 36$. To determine this correspondence, global appearance descriptors have been used to extract the most relevant information from the images [44]–[46], to compare them pairwise and to make the pairing process. In particular, the Fourier Signature (FS), proposed initially by Menegatti *et al.* [47] has been used.

Starting from an image with N_x rows and N_y columns, the Fourier Signature consists in obtaining the one-dimensional Discrete Fourier Transform (1D-DFT) of each row. This way, each row x of the original image

$r_x = \{r_{x,0}, r_{x,1}, r_{x,2}, ..., r_{x,N_y-1}\}$, $x = 0, \ldots, N_{x-1}$ is transformed into the sequence of complex numbers $F_x = \{F_{x,0}, F_{x,1}, F_{x,2}, ..., F_{x,N_y-1}\}$, $x = 0, \ldots, N_{x-1}$ using eq. 4 [48]:

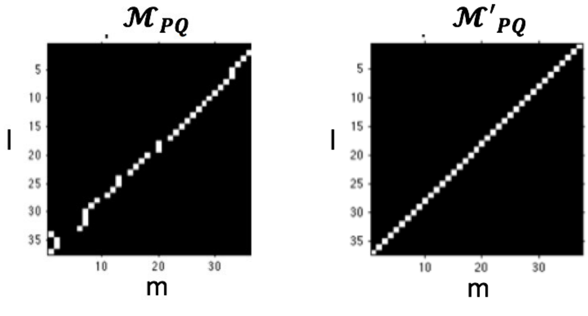
$$F_{x,k} = \sum_{n=0}^{N_y-1} r_{x,n} \cdot e^{-j(2\pi/N_y)kn}, \quad k = 0, \ldots, N_y - 1, \ x = 0, \ldots, N_x - 1 \qquad (4)$$

After transforming the whole image, the result is a complex matrix $F(v, y)$, where v is the frequency variable, expressed in *cycles/pixel*. In this matrix, the most relevant information is concentrated on the low frequency components. Therefore, a compression effect is achieved if a reduced number of such components is retained. Also, the high frequency components tend to be more corrupted by the possible presence of noise in the scenes. This way, the last columns of the matrix can be discarded and only the first N_k columns might be retained, resulting the matrix $F(v, y) \in \mathbf{C}^{N_x \times N_k}$, which is named *Fourier Signature*. $F(v, y)$ can be decomposed into a magnitudes matrix $A(v, y)$ and an arguments one. $A(v, y) \in \mathbf{R}^{N_x \times N_k}$ contains non localized information on the global appearance of the scene, so it can be considered as a global descriptor of the appearance of the original image. Taking these facts into account, all the images in the sets $K_{P,l}$ and $K_{Q,m}$ are individually transformed to obtain their global appearance descriptors (i.e. the magnitudes matrix of their Fourier Signature), leading to the sets of matrices $A_{P,l}$ and $A_{Q,m}$ respectively.

Once the descriptors of all the images are available, we can pair up each image in the set $K_{P,l}$ with one image in the set $K_{Q,m}$. To do it, the Euclidean distance between each descriptor $A_{P,l}$ and all the descriptors in the second set $A_{Q,m}$ is calculated, and the one which presents a minimum distance is paired up with $A_{P,l}$. According to this process, it is possible that the same image in the set $K_{Q,m}$ is assigned to several images in the set $K_{P,l}$. When the match-ups between each descriptor in the first set and all the descriptors of the second set is calculated, a sparse 37 × 37 matrix M_{PQ} is obtained. Fig. 5(a), makes a depiction of a sample pairings matrix M_{PQ}. In this figure, the horizontal axis represents the number of image m in the second set $K_{Q,m}$ and the vertical axis represents the number of image l in the first set $K_{P,l}$. In this matrix, the components that indicate a match-up are assigned unit value.

In the sample case shown in this figure, the image $K_{P,33}$ is paired up with the image $K_{Q,0}$ (since this is the image in the set $K_{Q,m}$ whose descriptor presents a minimum Euclidean distance to the descriptor of the image $K_{P,33}$). However, the image $K_{P,36}$ is paired up with the image $K_{Q,0}$ too. Considering the image acquisition system described in the section III-A, there must be only some offset in the order or acquisition of the images from both poses. Also, the direction in which the turret spins can be different. Therefore, the correct association between two images:

$$K_{P,l} \rightarrow K_{Q,m} \qquad (5)$$



(a) (b)

FIGURE 5. **Matrix of pairings. Figure 5(a) M_{PQ} shows the initial matrix of pairings between the images in the sets $K_{P,l}$ and $K_{Q,m}$. Figure 5(b) M'_{PQ} shows the final matrix of pairings used to establish the actual pairings.**

depends on two variables (s, t). The first one, $s = \pm 1$, indicates the spinning direction (clockwise or counterclockwise) and t represents the relative offset between the first image of each set. This way, it is necessary to estimate these two values s and t from M_{PQ}. To perform this estimation in a robust way, once M_{PQ} has been built we compare it with all the $2 \cdot 37$ possible matrices of pairings, considering $s = [-1, 1]$ and $t = [0, 1, \ldots, 36]$. The algorithm compares M_{PQ} with all the possible solutions and selects the most similar one (using the Hadamard product as the criterion to obtain the degree of similitude between matrices), which is named M'_{PQ}. Fig. 5(b) shows the matrix of parings which is the most similar M_{PQ} (fig. 5(a)) among all the possible solutions. This method has proved to be a robust and accurate way to estimate s and t in all the experiments developed. Once these values are known, the images of both sets are paired up according to the final matrix of match-ups M'_{PQ}.

Mathematically, once t and s are known, the pairings can be calculated through the next expression. The image $K_{P,l}$ is paired up with the image $K_{Q,m}$, where:

$$m = \begin{cases} (l + s \cdot t) \mod 37, & \textit{if} \quad s = 1 \\ (37 - l - s \cdot t) \mod 37, & \textit{if} \quad s = -1 \end{cases} \qquad (6)$$

B. MATCHING OF VISUAL FEATURES

Once the images in the sets $K_{P,l}$ and $K_{Q,m}$ have been paired-up, the next step consists in carrying out the detection, description and matching of features considering each pair of images. OpenCV is used with these aims [49].

First of all, the visual features of each image are detected and described by means of the Speeded Up Robust Features (SURF) algorithm [17] because the preliminary experiments proved its robustness in the target environments. To tune the SURF detector, a constant value on the Hessian matrix has been employed. During the extraction, a self-adjusting threshold is used which tries to keep the number of detected keypoints roughly constant, because a high number of features may lead to too many false positives during the matching process.

Finally, once the visual features have been obtained and described, a matching process is performed. For each pair of images resulting from the previous step (subsection IV-A), the SURF visual features are matched. This process is not very demanding even when artificial lights produce shadows in the images, because this search for potential matches is carried out between the images that have been previously paired-up. As a result, a set of matched visual features is obtained, for each pair of images. Two sample images captured from two consecutive poses are shown in fig. 6. These figures have been previously paired-up using the algorithm of subsection IV-A and the matched keypoints are also shown.

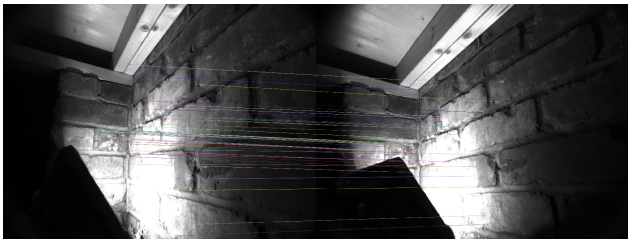


FIGURE 6. SURF keypoints extraction and matching between two previously paired-up images.

Considering that the images acquired from each robot pose have a high degree of overlapping among them, to create a robust and non-overlapping point cloud from each pose, the process described in this subsection is carried out using just 6 pairs of equally spaced images ($K_{P,l\cdot i}, K_{Q,m\cdot i}$) with $i = 1...6$. This way, we avoid adding redundant information (see fig. 4).

C. DEPTH INFORMATION EXTRACTION AND ALIGNMENT

Once the matches between the visual features (keypoints) of each pair of images have been obtained, in the third step, the 3D information of each keypoint is recovered. To do it, both the image in which the keypoint is present and the initial point cloud are considered. Once the 3D coordinates of all the matched keypoints are available, two new point clouds are created, one corresponding to the pose P and the other to the pose Q. These two point clouds are further processed to remove from them these points that belong to the ceiling and to the floor. It is important to highlight the fact that the ceiling may undergo some changes as the robot is ejecting foam isolation during the operation. Also, some changes may happen on the floor, due to the movement of existing debris while the robot moves on it. Therefore, the points in these two planes are expected to contain some inconsistencies and this is the reason to remove them. The final point clouds of the poses P and Q are named $\mathcal{M}'_P$ and $\mathcal{M}'_Q$ respectively. These point clouds have a significantly lower number of points compared to the original clouds $\mathcal{M}_P$ and $\mathcal{M}_Q$.

Using the Point Cloud Library (PCL) [30] with the two new point clouds $\mathcal{M}'_P$, $\mathcal{M}'_Q$, the transformation matrix that represents the alignment T_{PQ} (relative rotation and translation of the pose Q with respect to P) can be obtained. This algorithm iteratively examines some possible transformations to minimize the distance from the target point cloud $\mathcal{M}'_Q$ to the reference one $\mathcal{M}'_P$, dealing with the possible presence of outliers, until it finds an optimal result. As the number of points in each cloud is relatively small, the necessary time to reach a solution is reduced.

The benefit of the proposed method is twofold. First, since the keypoints have been previously selected from similar images, the process ensures that the subsequent correspondences are more robust and reliable. This feature is specially relevant in challenging environments, such as building crawl spaces, which are the target environments in this work. Second, the number of correspondences used to estimate the alignment matrix is substantially lower, what improves the calculation time. The next section presents the experimental evaluation that we have carried out to prove that the proposed algorithm is robust and relatively quick, comparing to some benchmark methods.

V. EVALUATION

This section presents the results of the alignment framework described in the previous sections. Three different real environments have been used to evaluate the performance of the algorithm. All the experiments have been carried out on a 2 × 2.66 GHz Dual-Core Intel Xeon CPU with 10 GB of memory. The acquisition of these three data-sets has been made with the data acquisition system detailed in section III-A, which was mounted on the four wheeled robotic platform designed by Holloway *et al*, as described in [8].

All the data have been captured within the three environments under real operating conditions. All of them are underfloor voids and they must be modeled to make it possible a subsequent foam insulation process. They present some features which are representative of the type of environments in which this robot has to move. On the one hand, the environment 1 is especially challenging because of the poor lighting conditions and the lack of objects in the scenes, what makes it especially prone to visual aliasing and complicates the detection and correct matching of visual features. Within this environment, 9 locations are considered and data are captured from these locations. As a result, 9 point clouds and 333 RGB images are available to model the environment (37 images per position). On the other hand, the environments 2 and 3 cover a wider area, whose approximate size is 2.5 × 2.5 meters. Environment 2 contains 703 RGB images and 19 point clouds acquired from 19 different poses and environment 3 contains 777 RGB images and 21 point clouds acquired from 21 different poses.

This section is structured in two subsections. The algorithm to pair-up images using their global appearance is tested in the first subsection and the performance of the alignment algorithm is assessed in the second subsection.

A. EVALUATION OF THE ALGORITHM TO PAIR-UP THE IMAGES USING GLOBAL APPEARANCE

To start with, every pair of consecutive locations in each data set are considered as poses P and Q and the visual alignment algorithm is run for every case, in such a way that

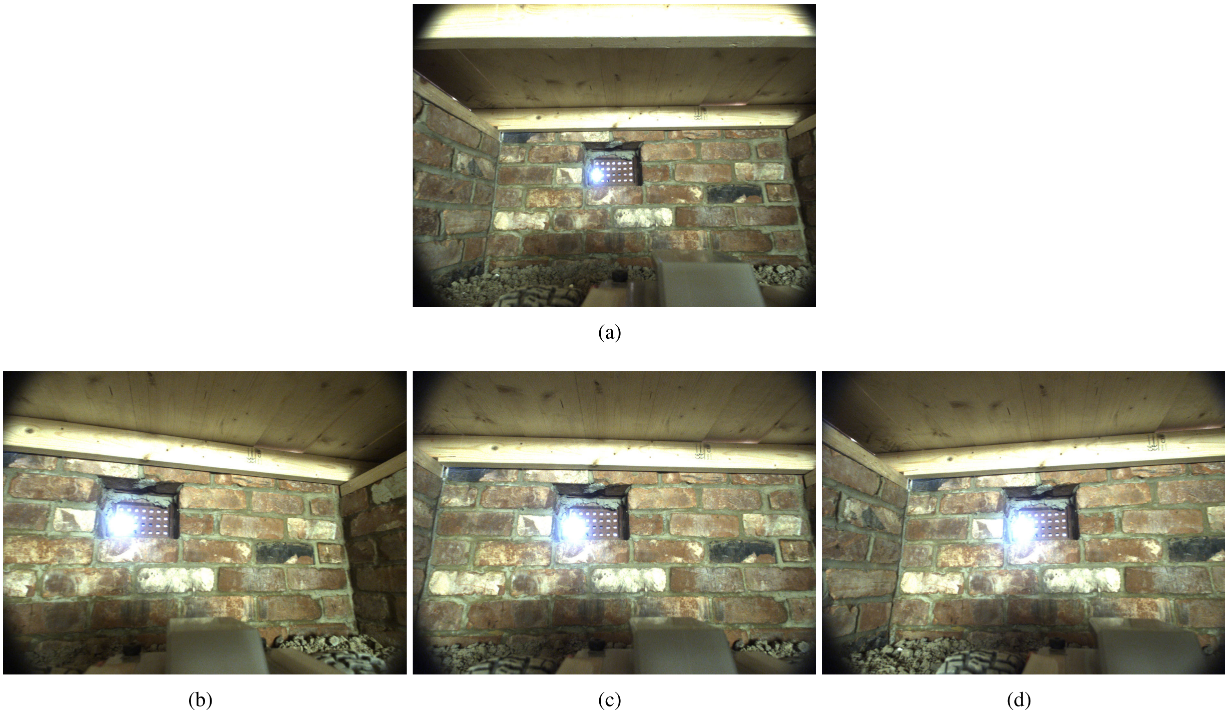
TABLE 1. Results of the image pairing-up process.

Position		Environment 1		Environment 2		Environment 3	
P	Q	t	s	t	s	t	s
0	1	36	-1	1	-1	2	-1
1	2	0	1	36	-1	35	-1
2	3	36	-1	0	-1	1	-1
3	4	36	-1	0	-1	36	-1
4	5	0	1	0	1	36	-1
5	6	0	1	36	1	35	-1
6	7	36	-1	0	1	0	-1
7	8	36	-1	1	1	0	-1
8	9	-	-	36	1	0	1
9	10	-	-	36	1	36	1
10	11	-	-	0	1	36	1
11	12	-	-	24	1	36	-1
12	13	-	-	4	1	36	-1
13	14	-	-	35	1	36	-1
14	15	-	-	33	1	0	-1
15	16	-	-	36	1	36	-1
16	17	-	-	35	1	36	-1
17	18	-	-	35	1	1	1
18	19	-	-	-	1	31	-1
19	20	-	-	-	1	10	-1

the coordinates of the position Q in the ground plane are estimated with respect to the coordinates of the position P. First, the results of the algorithm to pair-up the images captured from each pose are shown. Table 1 shows the values of the variables t and s, obtained after running the algorithm presented in section IV-A.

Let's suppose that the image K_{P,l_1} has been matched up with the image K_{Q,m_1}. Taking the acquisition process into account, the results are considered valid if the orientation of the turret when the image K_{P,l_1} was acquired is included between the orientations of the images K_{Q,m_1-1} and K_{Q,m_1+1}.

Considering this criterion, all the pairings provided by the algorithm turn out to be correct, despite the challenging properties of the three environments. An example of this pairing-up process can be seen in fig. 7 for the environment 2, positions $P = 9$, $Q = 10$. The result of the image pairing process is $t = 36$, $s = 1$. For example, this means that the image $K_{P,l_1} = K_{9,18}$ is matched-up with the image $K_{Q,m_1} = K_{10,17}$ where $m_1 = (l_1 + s \cdot t) \mod 37$ (eq. 6). Fig. 7(a) shows the image $K_{9,18}$, and the images $K_{10,16}$, $K_{10,17}$ and $K_{10,18}$ are show on figures 7(b), 7(c) and 7(d) respectively. These figures show that the pairing provided by


(a)

(b)

(c)

(d)

FIGURE 7. Results of an image pairing-up process. Figure 7(a) is the image 18 acquired from location (pose) $P = 9$ (named image $K_{9,18}$). Figures 7(b), 7(c) and 7(d) are the images 16, 17 and 18 acquired from pose $Q = 10$ (named, respectively, $K_{10,16}$, $K_{10,17}$ and $K_{10,18}$).

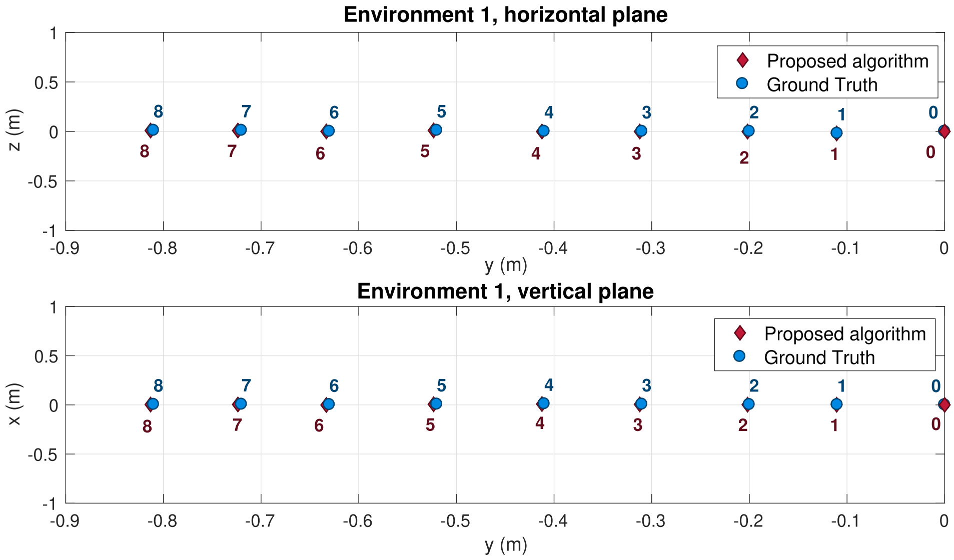


FIGURE 8. Results of the proposed alignment method in environment 1. The position of the robot obtained with the proposed algorithm and the ground truth are shown, separately, in the horizontal (yz) plane and in the vertical (xy) plane.

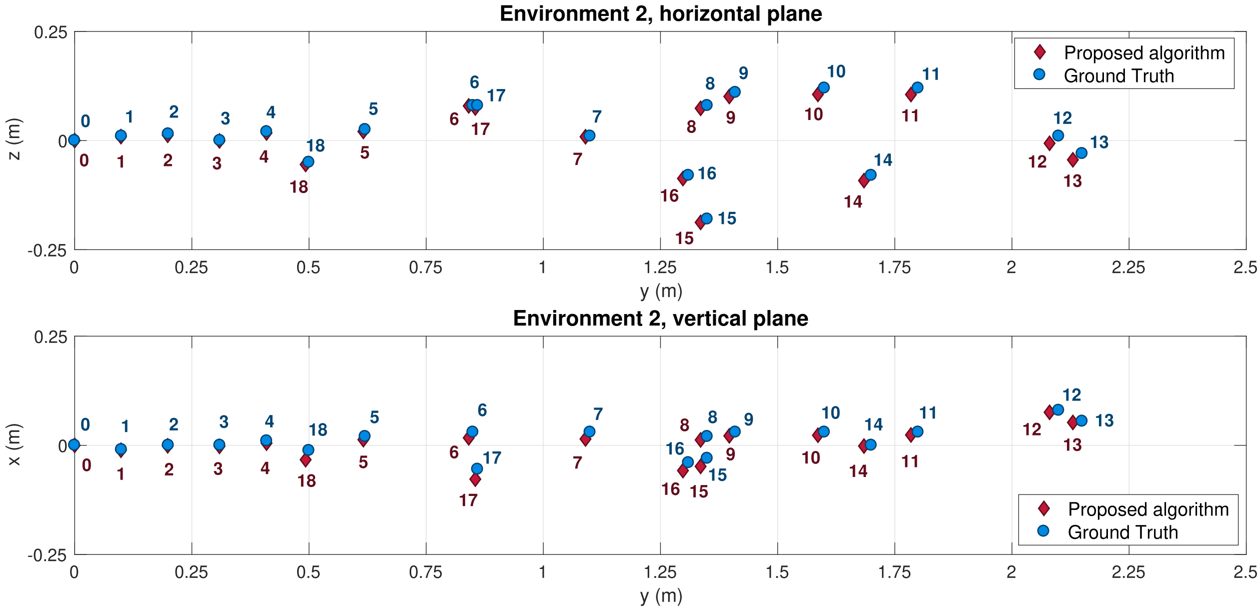


FIGURE 9. Results of the proposed alignment method in environment 2. The position of the robot obtained with the proposed algorithm and the ground truth are shown, separately, in the horizontal (yz) plane and in the vertical (xy) plane.

the algorithm is successful, since the orientation of the image $K_{9,18}$ is between the orientations of images $K_{10,16}$ and $K_{10,18}$.

The experiments have shown that the average necessary time to pair up the images of pose P with the images of pose Q is equal to 90 milliseconds.

B. EVALUATION OF THE ALIGNMENT ALGORITHM

This subsection assesses the performance of the alignment algorithm presented in subsection IV-C. For every pair of consecutive locations P, Q, the second pose can be estimated with respect to the first one by using the transformation matrix calculated with the proposed alignment algorithm T_{PQ}. After considering each par of consecutive locations and the three experimental environments, the results are shown in figures 8, 9 and 10. In these figures, both the position of the robot calculated with the algorithm and the ground truth are represented. For clarity purposes, the position of the robot in the horizontal (yz) plane and in the vertical (xy) plane is shown separately.

Fig. 11 represents the localization error at each position of the robot (millimeters), considering each environment separately. The environments 1 shows relatively accurate

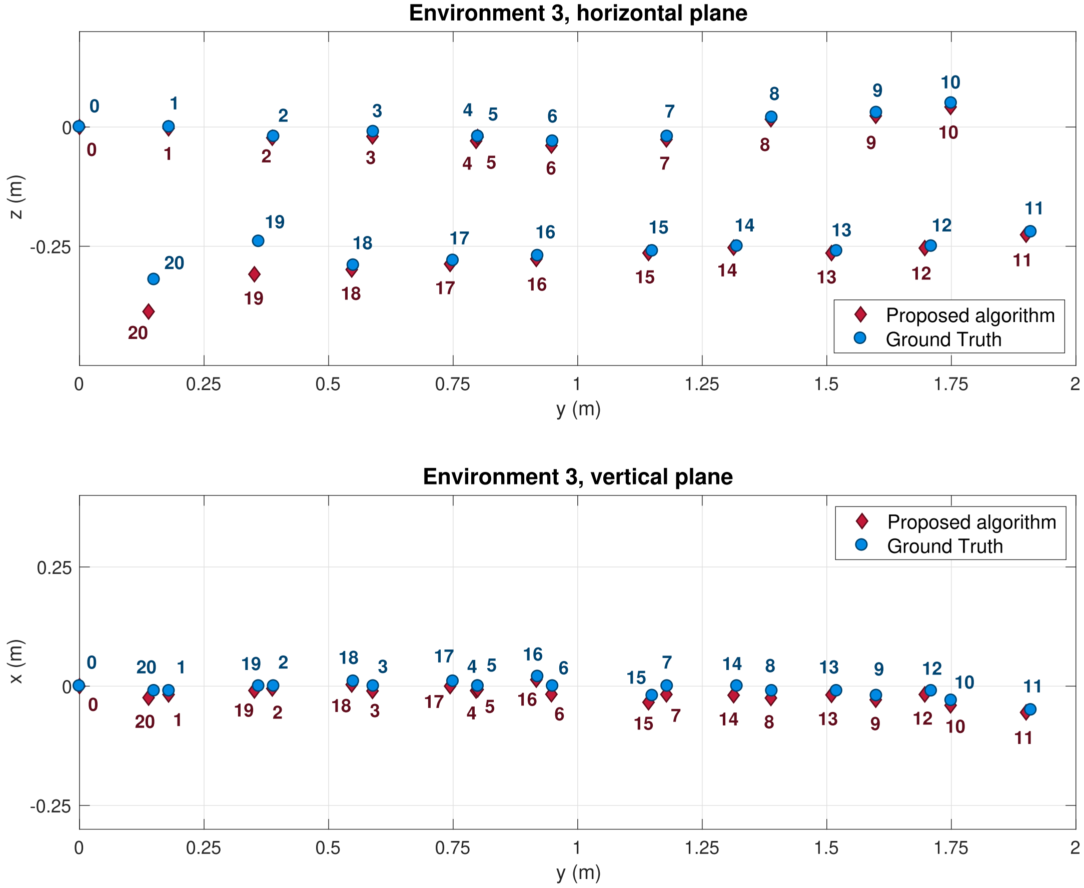


FIGURE 10. Results of the proposed alignment method in environment 3. The position of the robot obtained with the proposed algorithm and the ground truth are shown, separately, in the horizontal (yz) plane and in the vertical (xy) plane.

results with error values below 4.5 mm, comparing to the ground truth (fig. 11(a)). As far as the environment 2 is concerned, it presents slightly less accurate results, compared to environment 1, with a maximum global error around 25 mm in the pose 12 (fig. 11(b)). Finally, the results in environment 3 show that the algorithm produces good results (global error lower than 20 mm), except for the two last poses (19 and 20), where the method fails (figures 11(c) and 11(d)). Globally, considering all the results, the method provides relatively accurate position estimations, and it is not successful only in two cases.

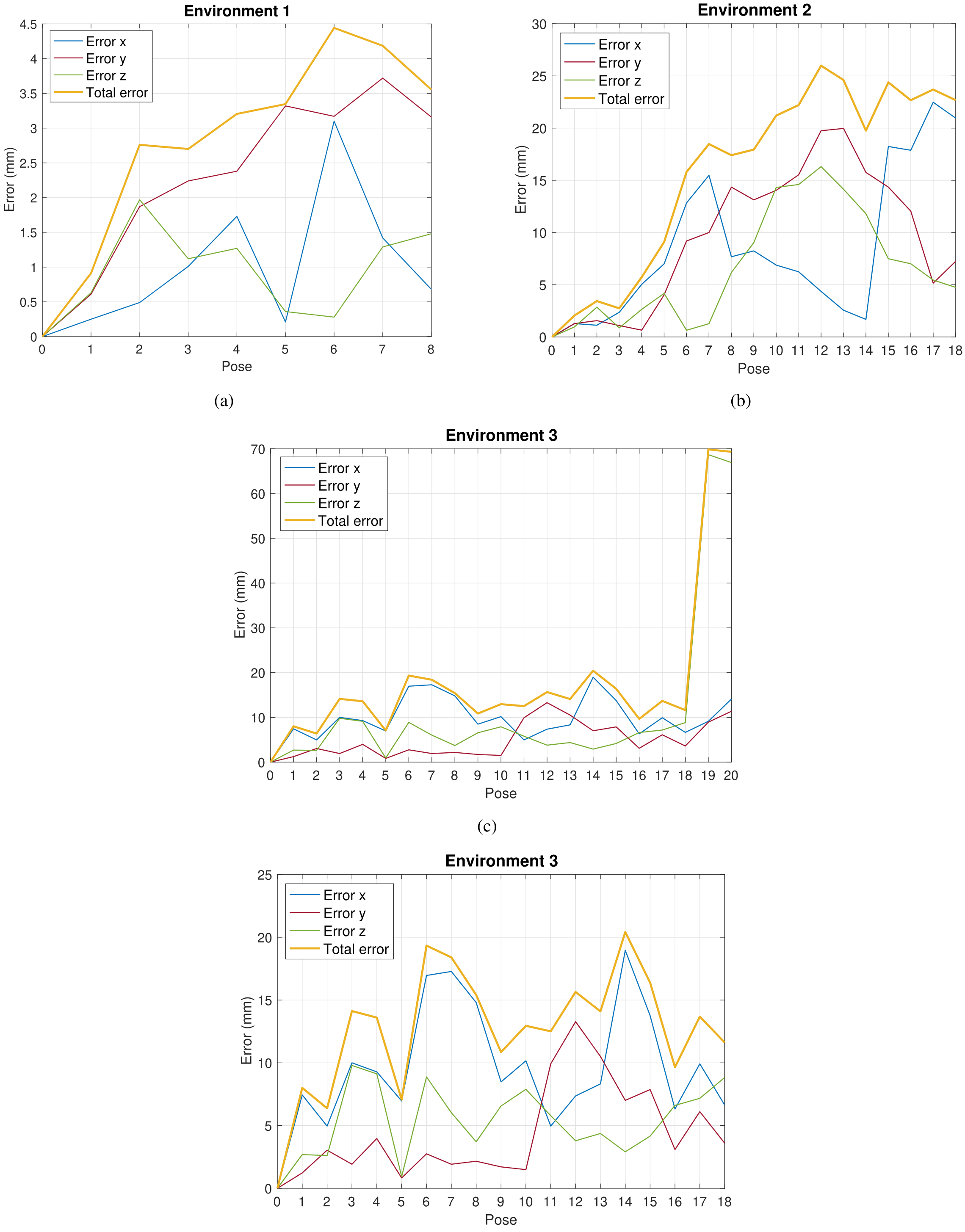
The results confirm that even in these especially challenging underfloor environments, the proposed approach presents a relatively accurate behaviour to estimate the alignment between two consecutive poses and thus estimate the location of the robot from an initial pose.

Finally, table 2 shows some relevant pieces of information about each environment (Env1, Env2 and Env3). The parameter #Features specifies the total number of visual keypoints matched between images. The second parameter, #Correspondences, indicates the number of points used by the system to do the alignment and estimate the transformation matrix once outliers have been removed. Finally, %Correspondences is the ratio between the two previous parameters.

In case of obtaining an unsuccessful alignment, it would be very important to have any indicator of this circumstance, as it would permit running an additional algorithm to try to recalculate the unsuccessfully aligned pose. The results show that the parameter %Correspondences permits knowing if the alignment is correct or not. Table 2 shows that the estimation of pose 19 with respect to pose 18 in environment 3 presents 20% correspondences, which is the lowest value of all the experiments. This is the pose where the algorithm starts to fail. In all the other cases, this percentage is 42% or higher.

The experiments show that a threshold can be set around 40%, in such a way that if this indicator is over this threshold, the result can be considered correct. Otherwise, the alignment must be considered unsuccessful and the matrix T_{PQ} must not be used to estimate the second pose of the robot. This way, subsequent poses should not be aligned with respect to this unsuccessful one to avoid spreading this error.

This part of the process (searching the correspondences, obtaining the depth of these correspondences and aligning two point clouds) has an average computing time of 760 milliseconds. Considering it together with the necessary time to pair-up the images, the average total time to obtain the transformation matrix between two consecutive poses is equal to 850 milliseconds.



(a)

(b)

(c)

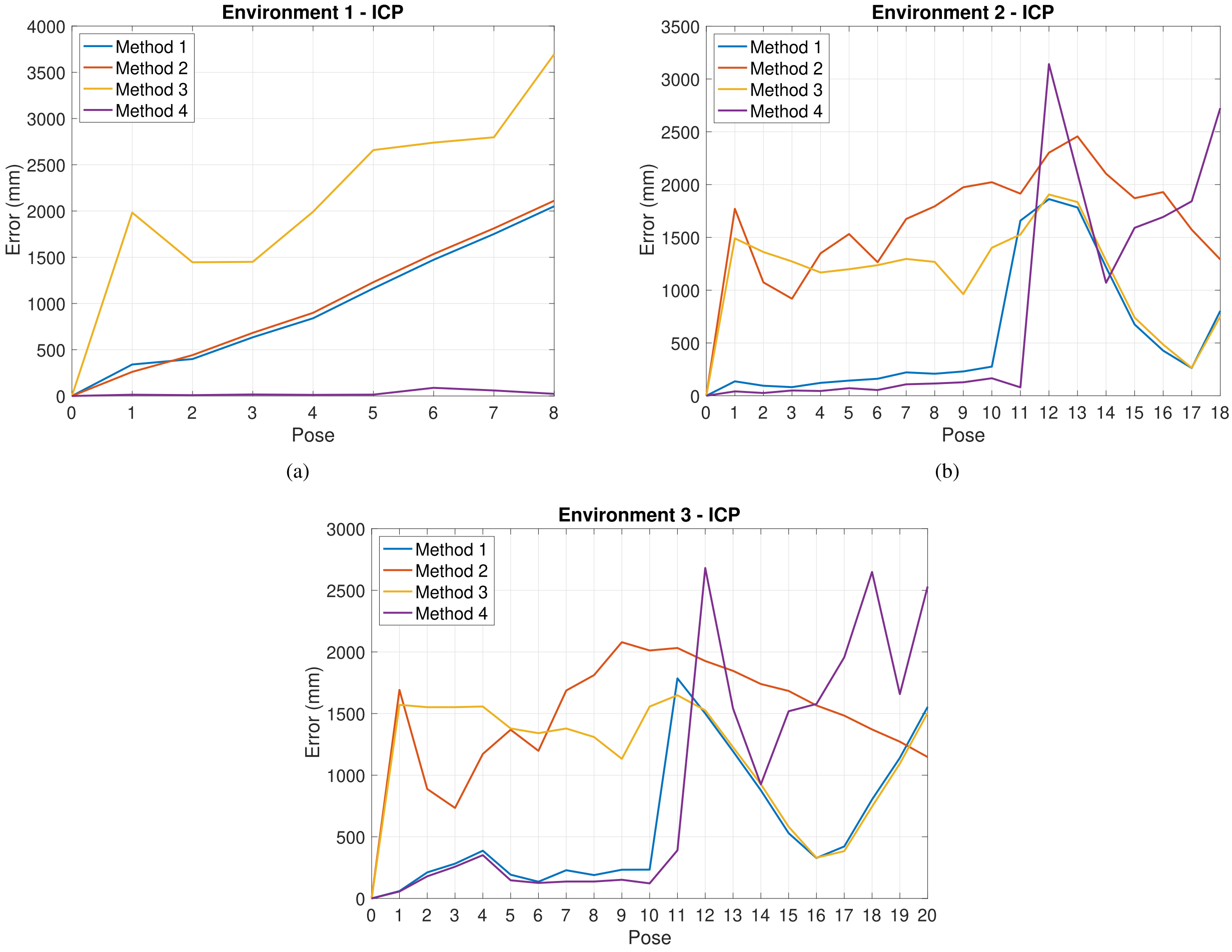
(d)

FIGURE 11. Error obtained along every axis for every pose in each environment and total error, expressed in mm. Fig. 11(a): environment 1; fig. 11(b): environment 2 and fig. 11(c): environment 3. Fig. 11(d) shows again the results obtained in environment 3 but removing the two last poses (which have proved to be unsuccessfully estimated).

To conclude the experimental section, we have run some benchmark methods, with the purpose of completing the experiments, for comparative purposes, and to prove the validity of the proposed algorithm in underfloor voids. We use three algorithms in this section: traditional ICP, CPD (Coherent Drift Point) [50] and NDT (Normal-Distributions Transform) [51]. On the one hand, CPD is a probabilistic registration algorithm designed for estimation of non-linear and non-rigid transformations. It is a global registration method that can be used to obtain an initial transformation estimate.

TABLE 2. Results of the alignment using the previously selected 3D points.

Poses	#Features			#Correspondences			%Correspondences		
P - Q	Env1	Env2	Env3	Env1	Env2	Env3	Env1	Env2	Env3
0-1	99	102	117	152	138	146	65	74	80
1-2	107	105	106	155	160	133	59	66	80
2-3	99	84	87	161	105	124	61	80	70
3-4	99	130	90	155	162	129	63	80	70
4-5	106	91	170	161	142	272	65	64	63
5-6	95	65	108	159	97	155	60	67	70
6-7	127	87	111	178	115	153	71	76	73
7-8	108	107	99	168	164	117	64	65	85
8-9	-	79	85	-	105	102	-	75	83
9-10	-	126	183	-	206	207	-	61	88
10-11	-	86	20	-	112	29	-	77	69
11-12	-	141	126	-	204	147	-	69	86
12-13	-	98	137	-	132	168	-	74	82
13-14	-	33	104	-	64	133	-	52	78
14-15	-	47	111	-	90	145	-	52	77
15-16	-	102	93	-	122	124	-	84	75
16-17	-	30	121	-	71	147	-	42	82
17-18	-	50	91		85	118	-	59	77
18-19	-	-	12	-	-	59	-	-	20
19-20	-	-	87	-	-	147	-	-	59



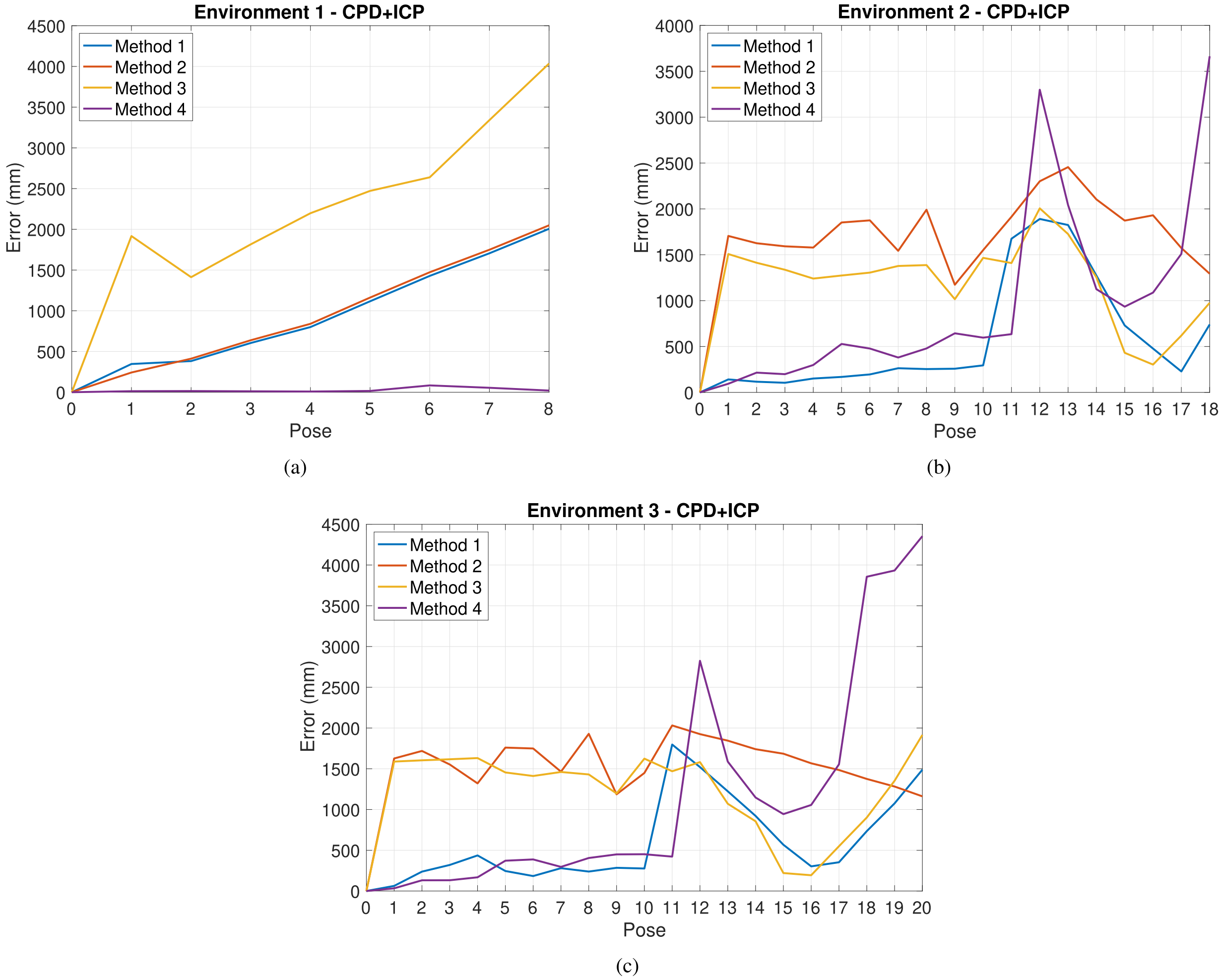
(a)

(b)

(c)

FIGURE 12. Results of the benchmark experiment 1. Total error obtained for every pose in each environment, expressed in mm. Fig. 12(a): environment 1; fig. 12(b): environment 2 and fig. 12(c): environment 3.

On the other hand, NDT is a laser scan matching algorithm that does not rely on specific correspondences between points so it is expected to be robust in presence of outliers and missing points. It is a local registration method that relies on an initial estimation of the transformation matrix. Since traditional ICP is also a local registration approach that relies



(a)

(b)

(c)

FIGURE 13. Results of the benchmark experiment 2. Total error obtained for every pose in each environment, expressed in mm. Fig. 13(a): environment 1; fig. 13(b): environment 2 and fig. 13(c): environment 3.

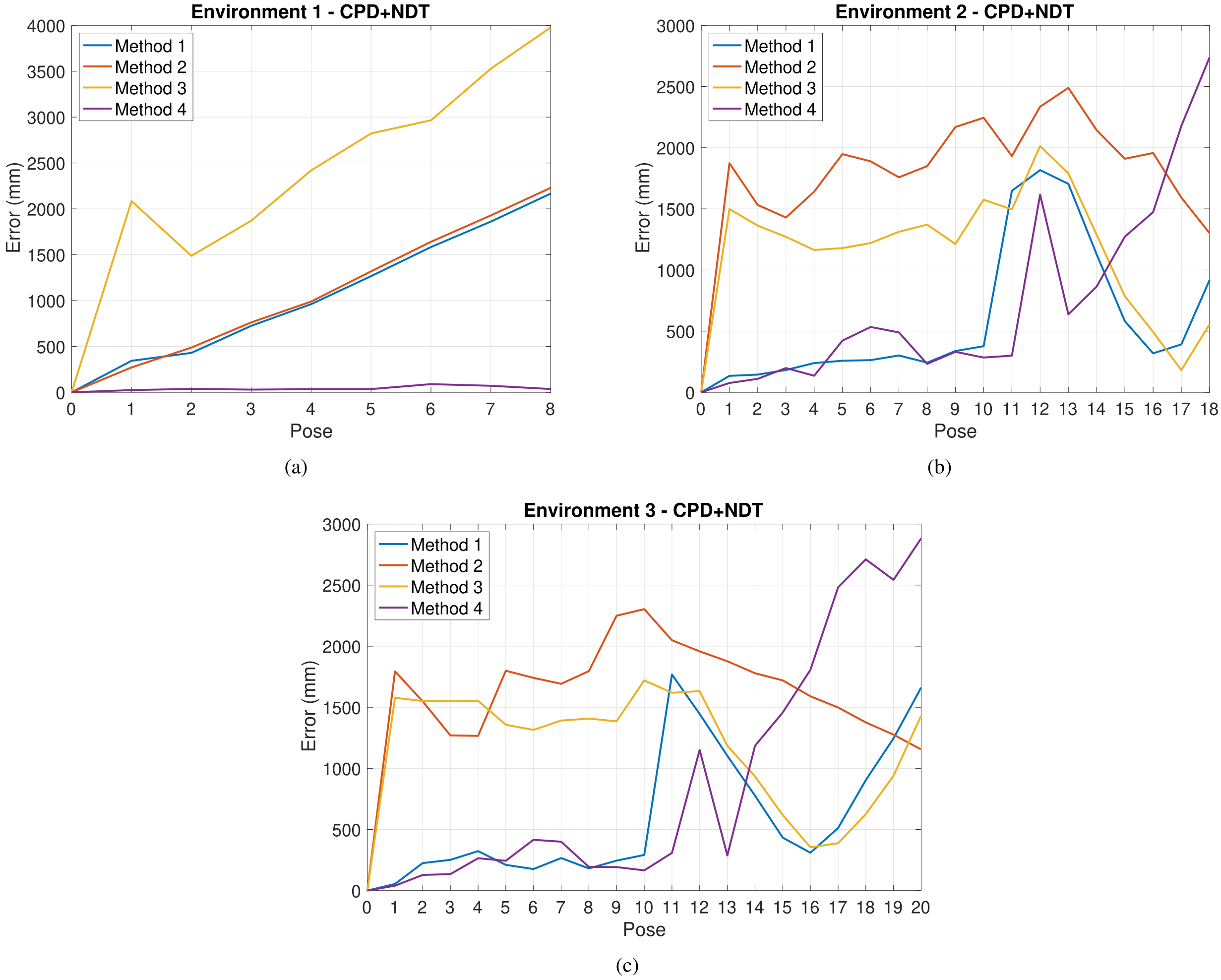
on an initial transform estimate, the benchmark experiments have been structured as follows:

1) Benchmark experiment 1. A traditional ICP method is run, considering different minimization metrics, downsampling and removal of some planes, to try to optimize the performance. Specifically, four methods are tested:

   a) **Method 1**: Traditional ICP, with minimization metric 'point to point'. All the points in the original clouds are considered.
   b) **Method 2**: Traditional ICP, with minimization metric 'point to plane'. All the points in the original clouds are considered.
   c) **Method 3**: The point clouds are previously downsampled, and 80% of points are retained. After that, traditional ICP is run, with minimization metric 'point to point'.
   d) **Method 4**: The point clouds are previously downsampled, retaining 80% of points. Subsequently, the points that belong to the ceiling and floor planes are removed (only the points belonging to the walls are kept, which are expected to lead to the most robust matches). After that, the resulting clouds are aligned using traditional ICP, with minimization metric 'point to point'.

   The results of the benchmark experiment 1 are shown in fig. 12. This figure represents the total localization error at each position of the robot (millimeters), considering each environment separately. In general terms, the method 4 tends to present relatively good results, comparing to the other three methods. In the environment 1, the error of the method 1 is around 10 mm. until pose 6, in which the error takes its maximum value, around 85 mm. In environments 2 and 3, the method 4 also presents relatively good results until the pose 11, from which the error substantially increases. In this benchmark experiment, the average time to complete the calculation of each pose is equal to 2.5 sec.

2) Benchmark experiment 2. This experiment consists in using, first, the CPD algorithm to obtain an initial estimate of the transformation between consecutive



(a)

(b)

(c)

FIGURE 14. Results of the benchmark experiment 3. Total error obtained for every pose in each environment, expressed in mm. Fig. 14(a): environment 1; fig. 14(b): environment 2 and fig. 14(c): environment 3.

poses and, second, the ICP algorithm to obtain the final estimate. ICP is configured according to method 4 in benchmark experiment 1, since it presented the best results. Several configurations are tested when using CPD to try to optimize the performance. Specifically, four methods are tested:

a) **Method 1**: The point clouds are previously downsampled, retaining 80% of the points.
b) **Method 2**: The point clouds are previously downsampled, retaining 80% of the points and the clouds are subsequently voxelized.
c) **Method 3**: It consists of the same steps than method 2 but, additionally, the clouds are denoised to remove outliers.
d) **Method 4**: It consists of the same steps than method 3 but, additionally, the points that belong to the ceiling and floor planes are removed (only the points belonging to the walls are kept).

The results of the benchmark experiment 2 are shown in fig. 13. In general terms, the results present the same tendencies than the benchmark experiment 1. Using CPT to obtain an initial estimate of the transformation between consecutive poses does not improve substantially the results comparing to the use of only ICP. Only slight improvements can be appreciated in environments 1 and 3. In this benchmark experiment, the average time to complete the calculation of each pose is equal to 7.6 sec.

3) Benchmark experiment 3. This experiment consists in using, first, the CPD algorithm to obtain an initial estimate of the transformation between consecutive poses and, second, the NDT algorithm to obtain the final estimate. To try to optimize the performance of the algorithm, the same configurations than in benchmark experiment 2 are tested.

The results of the benchmark experiment 3 are shown in fig. 14. Again, in general terms, the results present the same tendencies than the previous benchmark experiments. Using NDT to refine the estimate of the transformation matrix is not advantageous in this kind of building crawl spaces. In this benchmark experiment, the average time to complete the calculation of each pose is equal to 7.1 sec.

Studying together the results of the proposed method (fig. 11) and the results of the benchmark methods (figs. 12, 13 and 14), the proposed method clearly improves the results of the benchmark methods and is able to cope with the complexities of building crawl spaces.

VI. CONCLUSIONS

In this paper, a novel approach has been presented to solve the localization problem of a mobile robot which moves through underfloor voids. To solve this problem, the robot is equipped with an RGB-D sensor. Our approach extracts visual keypoints from sets of color images and matches them. The point clouds to align are built using only these matches, which constitute robust points because they have a reliable correspondence. Taking these point clouds as input, we use the PCL library to robustly estimate the transformations between two consecutive poses of the mobile robot. Finally, the approach is evaluated quantitatively using different RGB-D datasets acquired in real underfloor environments. The experiments in these environments show that the framework presents successful results in position estimation, comparing to some benchmark methods based on ICP, CPD and NDT.

During the evaluation of our system, considering globally all the experiments, two poses have not been properly aligned mainly owing to the long distance between the capture positions and the challenging properties of the operating environments. Since we have proved that it is possible to detect when such unsuccessful alignments occur, as a future research line, it would be interesting to implement an algorithm to recover these locations, for example by comparing them with other poses correctly located, either previously or subsequently.

ACKNOWLEDGMENT

The authors declare that there are no competing interests regarding the publication of this paper.

REFERENCES

[1] Y. Nevatia, T. Stoyanov, R. Rathnam, M. Pfingsthorn, S. Markov, R. Ambrus, and A. Birk, ''Augmented autonomy: Improving human-robot team performance in Urban search and rescue,'' in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Sep. 2008, pp. 22–26.

[2] D. Belter, M. Nowicki, P. Skrzypczyński, K. Walas, and J. Wietrzykowski, ''Lightweight RGB-D SLAM system for search and rescue robots,'' in *Proc. Prog. Automat., Robot. Measuring Techn.* Cham, Switzerland: Springer, 2015, pp. 11–21, doi: 10.1007/978-3-319-15847-1_2.

[3] L. Meng, C. W. De Silva, and J. Zhang, ''3D visual SLAM for an assistive robot in indoor environments using RGB-D cameras,'' in *Proc. 9th Int. Conf. Comput. Sci. Educ.*, Aug. 2014, pp. 32–37.

[4] K.-T. Song, S.-Y. Jiang, C.-J. Wu, M.-H. Lin, C. H. Wu, Y.-F. Chiu, C.-H. Lin, C.-Y. Lin, and C.-H. Liu, ''Mobile manipulation and visual servoing design of a configurable mobile manipulator,'' in *Proc. CACS Int. Autom. Control Conf. (CACS)*, Dec. 2013, pp. 239–244.

[5] R. Kümmerle, M. Ruhnke, B. Steder, C. Stachniss, and W. Burgard, ''A navigation system for robots operating in crowded urban environments,'' in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2013, pp. 3225–3232.

[6] Z. Miljković, N. Vuković, M. Mitić, and B. Babić, ''New hybrid vision-based control approach for automated guided vehicles,'' *Int. J. Adv. Manuf. Technol.*, vol. 66, no. 1, pp. 231–249, Apr. 2013, doi: 10.1007/s00170-012-4321-y.

[7] S. Pelsmakers, B. Croxford, and C. Elwell, ''Suspended timber ground floors: Measured heat loss compared with models,'' *Building Res. Inf.*, vol. 47, no. 2, pp. 127–140, Feb. 2019, doi: 10.1080/09613218.2017.1331315.

[8] M. Holloway, M. Julia, and P. R. N. Childs, ''A robot for spray applied insulation in underfloor voids,'' in *Proc. ISR 47th Int. Symp. Robot.*, Jun. 2016, pp. 1–7.

[9] M. Julia, M. Holloway, O. Reinoso, and P. R. N. Childs, ''Autonomous surveying of underfloor voids,'' in *Proc. ISR 47th Int. Symp. Robot.*, Jun. 2016, pp. 1–7.

[10] S. Thrun and J. J. Leonard, ''Simultaneous localization and mapping,'' in *Springer Handbook of Robotics*. Berlin, Germany: Springer, 2008, doi:10.1007/978-3-540-30301-5_38.

[11] L. Teslić, I. Škrjanc, and G. Klančar, ''EKF-based localization of a wheeled mobile robot in structured environments,'' *J. Intell. Robot. Syst.*, vol. 62, no. 2, pp. 187–203, May 2011, doi: 10.1007/s10846-010-9441-8.

[12] V. Prasad, S. Singh, N. Pareekutty, B. Ravindran, and K. M. Krishna, ''SLAM-Safe planner: Preventing monocular SLAM failure using reinforcement learning,'' *CoRR*, vol. abs/1607.07558, Jun. 2016. [Online]. Available: http://arxiv.org/abs/1607.07558

[13] H. Oleynikova, D. Honegger, and M. Pollefeys, ''Reactive avoidance using embedded stereo vision for MAV flight,'' in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2015, pp. 50–56.

[14] Y. Berenguer, L. Payá, M. Ballesta, and O. Reinoso, ''Position estimation and local mapping using omnidirectional images and global appearance descriptors,'' *Sensors*, vol. 15, no. 10, pp. 26368–26395, Oct. 2015, doi: 10.3390/s151026368.

[15] D. Valiente, L. Payà, L. Jiménez, J. Sebastián, and Ó. Reinoso, ''Visual Information Fusion through Bayesian inference for adaptive probability-oriented feature matching,'' *Sensors*, vol. 18, no. 7, p. 2041, Jun. 2018.

[16] H. Kwon, K. M. Ahmad Yousef, and A. C. Kak, ''Building 3D visual maps of interior space with a new hierarchical sensor fusion architecture,'' *Robot. Auto. Syst.*, vol. 61, no. 8, pp. 749–767, Aug. 2013.

[17] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool, ''Speeded-up robust features (SURF),'' *Comput. Vis. Image Underst.*, vol. 110, no. 3, pp. 346–359, Jun. 2008, doi: 10.1016/j.cviu.2007.09.014.

[18] J. W. Li, D. F. Zheng, Z. H. Guan, C. Y. Chen, X. W. Jiang, and X. H. Zhang, ''Indoor 3D scene reconstruction for mobile robots using Microsoft kinect sensor,'' in *Proc. 35th Chin. Control Conf. (CCC)*, Jul. 2016, pp. 6324–6328.

[19] D. Belter, M. Nowicki, and P. Skrzypczyński, ''Accurate map-based RGB-D SLAM for mobile robots,'' in *Proc. Robot, 2nd Iberian Robot. Conf., Adv. Robot.*, vol. 2. Springer, 2016, pp. 533–545, doi: 10.1007/978-3-319-27149-1_41.

[20] F. Endres, J. Hess, N. Engelhard, J. Sturm, D. Cremers, and W. Burgard, ''An evaluation of the RGB-D SLAM system,'' in *Proc. IEEE Int. Conf. Robot. Automat.*, May 2012, pp.1691–1696.

[21] M. Montemerlo, S. Thrun, D. Koller, and B. Wegbreit, ''FastSLAM: A factored solution to the simultaneous localization and mapping problem,'' in *Proc. AAAI Nat. Conf. Artif. Intell.*, Edmonton, AB, Canada, 2002.

[22] A. Gil, O. M. Mozos, M. Ballesta, and O. Reinoso, ''A comparative evaluation of interest point detectors and local descriptors for visual SLAM,'' *Mach. Vis. Appl.*, vol. 21, no. 6, pp. 905–920, Oct. 2010, doi: 10.1007/s00138-009-0195-x.

[23] A. Nuchter, K. Lingemann, J. Hertzberg, and H. Surmann, ''6D SLAM with approximate data association,'' in *Proc. ICAR 2nd Int. Conf. Adv. Robot.*, Oct. 2006, pp. 242–249.

[24] G. Grisetti, S. Grzonka, C. Stachniss, P. Pfaff, and W. Burgard, ''Efficient estimation of accurate maximum likelihood maps in 3D,'' in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Oct. 2007, pp. 3472–3478, doi: 10.1109/iros.2007.4399030.

[25] Z. Shi, Z. Liu, X. Wu, and W. Xu, ''Feature selection for reliable data association in visual SLAM,'' *Mach. Vis. Appl.*, vol. 24, no. 4, pp. 667–682, May 2013, doi: 10.1007/s00138-012-0440-6.

[26] S. Rusinkiewicz and M. Levoy, ''Efficient variants of the ICP algorithm,'' in *Proc. 3rd Int. Conf. 3-D Digit. Imag. Modeling*, 2001, pp. 145–152.

[27] A. Segal, D. Hähnel, and S. Thrun, ''Generalized-ICP,'' in *Robotics: Science and Systems*, J. Trinkle, Y. Matsuoka, and J. A. Castellanos, Eds. Cambridge, MA, USA: MIT Press, 2009. [Online]. Available: http://dblp.uni-trier.de/db/conf/rss/rss2009.html

[28] R. Tiar, M. Lakrouf, and O. Azouaoui, ''Fast ICP-SLAM for a bi-steerable mobile robot in large environments,'' in *Proc. Int. Conf. Adv. Robot. (ICAR)*, Jul. 2015.

[29] H. Cho, E. K. Kim, and S. Kim, ''Indoor SLAM application using geometric and ICP matching methods based on line features,'' *Robot. Auto. Syst.*, vol. 100, pp. 206–224, Feb. 2018. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0921889017301367

[30] P. J. Besl and N. D. McKay, "A method for registration of 3-D shapes," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 14, no. 2, pp. 239–256, Feb. 1992, doi: 10.1109/34.121791.

[31] M. Magnusson, A. Lilienthal, and T. Duckett, "Scan registration for autonomous mining vehicles using 3D-NDT," *J. Field Robot.*, vol. 24, no. 10, pp. 803–827, Oct. 2007.

[32] J.-D. Lee, S.-S. Hsieh, C.-H. Huang, L.-C. Liu, C.-T. Wu, S.-T. Lee, and J.-F. Chen, "An adaptive icp registration for facial point data," in *Proc. 18th Int. Conf. Pattern Recognit. (ICPR)*, 2006, pp. 703–706.

[33] F. Pomerleau, F. Colas, R. Siegwart, and S. Magnenat, "Comparing ICP variants on real-world data sets," *Auton. Robots*, vol. 34, no. 3, pp. 133–148, Apr. 2013.

[34] F. Pomerleau, F. Colas, and R. Siegwart, "A review of point cloud registration algorithms for mobile robotics," *Found. Trends Robot.*, vol. 4, no. 1, pp. 1–104, 2015.

[35] C. Feng, Y. Taguchi, and V. R. Kamat, "Fast plane extraction in organized point clouds using agglomerative hierarchical clustering," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2014, pp. 6218–6225.

[36] W. S. Grant, R. C. Voorhies, and L. Itti, "Finding planes in LiDAR point clouds for real-time registration," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Nov. 2013, pp. 4347–4354.

[37] K. Khoshelham, D. R. Dos Santos, and G. Vosselman, "Generation and weighting of 3D point correspondences for improved registration of RGB-D data," in *Proc. ISPRS Ann. Photogramm. Remote Sens. Spat. Inf. Sci.*, vol. 5, Oct. 2013, pp. 127–132.

[38] K. Yousif, A. Bab-Hadiashar, and R. Hoseinnezhad, "3D registration in dark environments using RGB-D cameras," in *Proc. Int. Conf. Digit. Image Comput., Techn. Appl. (DICTA)*, Nov. 2013, pp. 1–8.

[39] P. Kim, J. Chen, and Y. K. Cho, "Automated point cloud registration using visual and planar features for construction environments," *J. Comput. Civil Eng.*, vol. 32, no. 2, Mar. 2018, Art. no. 04017076, doi: 10.1061/(asce)cp.1943-5487.0000720.

[40] G.-X. Xin, X.-T. Zhang, X. Wang, and J. Song, "A RGBD SLAM algorithm combining ORB with PROSAC for indoor mobile robot," in *Proc. 4th Int. Conf. Comput. Sci. Netw. Technol. (ICCSNT)*, Dec. 2015, pp. 71–74.

[41] G. Pandey, S. Savarese, J. R. Mcbride, and R. M. Eustice, "Visually bootstrapped generalized ICP," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2011, pp. 2660–2667.

[42] D. Holz, A. E. Ichim, F. Tombari, R. B. Rusu, and S. Behnke, "Registration with the point cloud library: A modular framework for aligning in 3-D," *IEEE Robot. Autom. Mag.*, vol. 22, no. 4, pp. 110–124, Dec. 2015.

[43] K.-L. Low, *Linear Least-Squares Optimization for Point-to-Plane ICP Surface Registration*, vol. 4. Chappel Hill, NC, USA: Univ. North Carolina, 2004.

[44] L. Payá, A. Peidró, F. Amorós, D. Valiente, and O. Reinoso, "Modeling environments hierarchically with omnidirectional imaging and global-appearance descriptors," *Remote Sens.*, vol. 10, no. 4, p. 522, Mar. 2018.

[45] S. Cebollada, L. Paya, V. Roman, and O. Reinoso, "Hierarchical localization in topological models under varying illumination using holistic visual descriptors," *IEEE Access*, vol. 7, pp. 49580–49595, 2019.

[46] S. Cebollada, L. Payá, W. Mayol, and O. Reinoso, "Evaluation of clustering methods in compression of topological models and visual place recognition using global appearance descriptors," *Appl. Sci.*, vol. 9, no. 3, p. 377, Jan. 2019.

[47] E. Menegatti, T. Maeda, and H. Ishiguro, "Image-based memory for robot navigation using properties of omnidirectional images," *Robot. Auto. Syst.*, vol. 47, no. 4, pp. 251–267, Jul. 2004.

[48] L. Payá, O. Reinoso, Y. Berenguer, and D. Úbeda, "Using omnidirectional vision to create a model of the environment: A comparative evaluation of global-appearance descriptors," *J. Sensors*, vol. 2016, 2016, Art. no. 1537891.

[49] A. Bradski, *Learning OpenCV—Computer Vision With OpenCV Library: Software That Sees*, G. Bradski and A. Kaehler, Eds., 1st ed. Sebastopol, CA, USA: O'Reilly Media, 2008.

[50] A. Myronenko and X. Song, "Point set registration: Coherent point drift," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 12, pp. 2262–2275, Dec. 2010.

[51] P. Biber and W. Strasser, "The normal distributions transform: A new approach to laser scan matching," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Jul. 2004, pp. 2743–2748.

CRISTOBAL PARRA received the M.Eng. degree in industrial engineering from Miguel Hernández University, Elche, Spain, in 2014. He is currently working with the company Robotics and Vision Technologies (RVT), which is highly specialized in artificial vision, robotics, and industrial automation. He is also a Researcher with Miguel Hernández University, Spain. His research interests include omnidirectional vision and global appearance algorithms, map building, and localization of mobile robots.

SERGIO CEBOLLADA received the M.Eng. degree in telecommunication engineering from Miguel Hernández University (UMH), in 2014. Since 2015, he has been with Miguel Hernández University as a Ph.D. candidate student. The topic of research is focused on omnidirectional vision and global appearance algorithms, map building and localization of mobile robots, and deep learning. Since 2017, he had a PhD-Candidate Scholarship supported by the Valencian Government (ACIF/2017/146).

LUIS PAYÁ received the M.Eng. degree in industrial engineering, in 2002, and the Ph.D. degree in industrial technologies, in 2014. He currently works as an Associate Professor with the Department of Systems Engineering and Automation, Miguel Hernández University, Spain. He teaches some subjects related to the fields of automatic control, electronics, and robotics. He is the author of several books, papers, and communications in the cited topics. His current research interests include omnidirectional vision and global appearance algorithms, topological map building and localization of mobile robots, and also implementation and testing of remote laboratories.

MATHEW HOLLOWAY is currently the CEO of Q-Bot, an innovative technology company using robotics, AI, and 3D mapping to enable new solutions for the inspection, maintenance, and management of buildings. He is also a Researcher with Imperial College London. He has spent last 15 years of his career developing new services for use in the public sector and housing industry. His ideas have reached global audiences and have received recognition through awards from Ashden, CIBSE, HSBC, Allianz, and CNBC. Mathew studied Engineering and Design at Bath University (Meng), Imperial College London (MSc) and the Royal College of Art (MA).

OSCAR REINOSO received the industrial engineering and Ph.D. degrees from the Polytechnic University of Madrid (UPM), in 1991 and 1996, respectively. From 1994 to 1997, he was with the Research and Development Department of Protos Desarrollo in a visual inspection system. Since 1997, he has been with the Miguel Hernández University, as Professor in control, robotics, and computer vision. He is the author of several books, papers, and communications in the cited topics. His research interests include robotics, teleoperated robots, climbing robots, visual servoing, and visual inspection systems. He is a member of the CEA-IFAC.