CAD2SLAM: Adaptive Projection Between CAD Blueprints and SLAM Maps

Martín Bayón-Gutiérrez, Natalia Prieto-Fernández, María Teresa García-Ordás, José Alberto Benítez-Andrades, Héctor Alaiz-Moretón, and Giorgio Grisetti

***Abstract*—Robotic mobile platforms are key building blocks for numerous applications and cooperation between robots and humans is a key aspect to enhance productivity and reduce labor cost. To operate safely, robots typically rely on a custom map of the environment that depends on the sensor configuration of the platform. In contrast, blueprints stand as an abstract representation of the environment. The use of both CAD and SLAM maps allows robots to collaborate using the blueprint as a common language, while also easing the control for non-robotics experts. In this work we present an adaptive system to project a 2D pose in the blueprint to the SLAM map and vice-versa. Previous work in the literature aims at morphing a SLAM map to a previously available map. In contrast, *CAD2SLAM* does not alter the internal map representation used by the SLAM and localization algorithms running on the robot, preserving its original properties. We believe that our system is beneficial for the control and supervision of multiple heterogeneous robotic platforms that are monitored and controlled through the CAD map. Finally, we present a set of experiments that support our claims as well as open-source implementation.**

***Index Terms*—Software architecture for robotic and automation, computational geometry, mapping computational geometry.**

I. Introduction

ACCURATE and reliable mapping of the environment in which a robot operates is fundamental for the use of such platforms. To gather an initial map, in which to plan the missions for the robot, many systems have been proposed since the origin of the so-called Simultaneous Localization and Mapping (SLAM) techniques in the 1980s [1]. SLAM systems use a series of sensors, such as cameras, laser rangefinders or wheel encoders, to estimate the ego-motion of the robot while a map of the environment traversed so far is constructed [2], [3], [4].

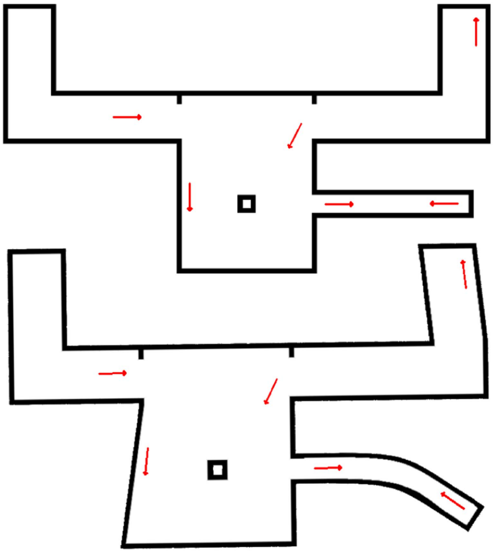

Fig. 1. CAD2SLAM adaptive projection in the *DEMO* environment. Poses in the CAD blueprint (top) are projected to their corresponding pose in the SLAM map (bottom).

For autonomous navigation, the map requires to be both topologically consistent and show an accurate local geometry. State-of-the-art SLAM pipelines provide locally accurate representations of the navigation environment and by revisiting places seen in the past they enhance the map's global consistency through the loop closures. However, depending on the shape of the environment or operational restrictions in steering the robot, loop closures may not be possible. In these cases the map might exhibit the effects of the increasing drift affecting the open loop estimator. The map built by a SLAM system is typically targeted at localization, and it depends on the sensors used: in case a Visual SLAM system is used, the map consists of 3D points with the attached appearance, while LiDAR maps are typically represented as occupancy grids. Furthermore, SLAM maps are typically cluttered with noise and occlusions, that appear as artifacts and render the map less intuitive to non-robotics experts, or might not even resemble blueprints.

In contrast, Computer Aided Design (CAD) blueprints or emergency evacuation plans are part of the common knowledge, since people are typically exposed to these types of representation of buildings, and may contain useful semantic information about the environment. However, CAD blueprints may lack the

Received 30 August 2024; accepted 10 December 2024. Date of publication 26 December 2024; date of current version 6 January 2025. This article was recommended for publication by Associate Editor H. Yang and Editor J. Civera upon evaluation of the reviewers' comments. This work was supported by Programa Propio de Investigación de la Universidad de León 2021. *(Corresponding author: Martín Bayón-Gutiérrez.)*

Martín Bayón-Gutiérrez, Natalia Prieto-Fernández, María Teresa García-Ordás, José Alberto Benítez-Andrades, and Héctor Alaiz-Moretón are with the Department of Electric, Systems and Automatics Engineering, Universidad de León, 24071 León, Spain (e-mail: martin.bayon@unileon.es; nprif@unileon.es; mgaro@unileon.es; jbena@unileon.es; halam@unileon.es).

Giorgio Grisetti is with the Department of Computer, Control, and Management Engineering "Antonio Ruberti", Universidad de León, 00185 Rome, Italy (e-mail: grisetti@diag.uniroma1.it).

This letter has supplementary downloadable material available at https://doi.org/10.1109/LRA.2024.3522838, provided by the authors.

Digital Object Identifier 10.1109/LRA.2024.3522838

metrical accuracy needed for a robot to autonomously navigate in the environment [5], [6].

The concurrent use of a SLAM map and a CAD blueprint allows the robot to use path-planning and navigation tools in its own internal representation of the environment while the operator is able to visualize the state of the robot and issue commands in an easy-to-understand blueprint. Moreover, the use of a common CAD blueprint provides an excellent opportunity to facilitate the collaboration of multiple robots that operate in the same environment, each with its own sensor configuration, SLAM map and navigation stack. By projecting the blueprint into each of the robots internal map, and vice-versa, the CAD blueprint can be established as the baseline for the collaboration among the mobile platforms.

Previous works have studied the combination of two representations of the environment by rigidly overlaying a blueprint onto the SLAM map [7] or by the definition of localization strategies using the robot sensor readings on a sketch of the environment [8]. However, these approaches fail to accurately map the environment when large distortions or noise are present on the SLAM map. In alternative to these methods, we propose an adaptive projection system which is able to robustly mitigate effects such as skewing, bending or shortening of corridors, typical in SLAM maps, while preserving local consistency. In contrast to other methods, our approach does not aim at compensating the distortion of the map, but rather at seeking a function that maps corresponding poses between different representations of the environment. We represent the environment with a grid-mesh covering the free space. The mesh itself is modeled as a non-rigid body where the poses of the nodes are elements of $SE(2)$ and nearby nodes are connected by soft $SE(2)$ constraints that preserve the initial configuration of the mesh. The mesh can be warped by imposing the position of specific control points, that are chosen to represent the same location in the SLAM maps and in the blueprint. The problem of warping the mesh is modeled as a Pose-Landmark Graph Optimization problem (PLGO) [9], [10], [11] where the poses of the mesh represent the poses along a virtual trajectory of the robot and the control points are modeled as fixed landmarks. To use our approach, the installer is only required to provide a set of corresponding points in the CAD blueprint and the SLAM map. Since no specific assumptions are made on the SLAM map, its representation can be arbitrary as long as it is possible to identify correspondences in it, hence it is sensor and robot independent. This facilitates, for example, the collaboration between different types of robots, e.g., shelf carrier robots and automated forklifts, equipped with LiDAR sensors at different heights, thus perceiving different representations of the same environment.

Qualitative and quantitative experiments have demonstrated that our system is able to accurately and quickly project any pose in the blueprint to the SLAM and vice-versa, which may be useful in several environments that require the collaboration between several robots and/or with human operators.

Fig. 1 presents a blueprint and its corresponding SLAM map, in which the red arrows represent robot poses that are projected from the CAD blueprint to the SLAM map, while preserving its local consistency.

The main contributions of this work are the introduction of the problem of adaptive projection between two maps that represent the same environment and the introduction of ***CAD2SLAM***, a methodology for its resolution, based on the use of a set of correspondence points in both maps.

An open-source implementation of CAD2SLAM, as well as the dataset used, extended experiments and additional information is available at the project web page at: https://cad2slam.martinbayon.com.

II. Related Work

In the literature, the integration of a SLAM map with other sources of information has been mainly addressed by using these sources as a prior to be fed to the SLAM pipeline. Most of the authors decided to adapt the SLAM map to a simple sketch of the environment, with just a few details such as the position of the corners of a room or some furniture elements.

One of the first works on the matter was proposed by Setalaphruk et al. [12], who described a navigation method aimed to control a mobile robot by issuing commands on a floor map of the environments. This system is based on the use of a topological map in the form of a Voronoi diagram that allows the robot to localize in the environment by detecting the intersection between corridors. Subsequently, Mielle et al. [13] presented a similar system to match a sketch map and a robot map.

Parekh et al. [14] presented an initial work on the adaptation of SLAM and a sketch of an environment by modeling a correspondence system for the objects present in the environment. The authors made the assumption that the number of objects in the sketch is the same as in the SLAM map, and do not provide a method for mapping the free space of the environment, but only the objects.

Behzadian et al. [8], Boniardi et al. [15] and Bowen-Biggs et al. [5] investigated the use of hand-drawn maps of an indoor environment for robot localization. Behzadian and Boniardi modeled the sensor readings of the robot to the user map and used Monte Carlo localization (MCL) to localize the robot. This system is able to distinguish the room in which the robot stays, but is unable to provide accurate localization. Alternatively, Bowen-Biggs manually selected a set of corresponding points in the blueprint and the SLAM map to optimize a system that projects points in the reference frame of the blueprint to the SLAM map reference frame. This system is used to specify a no-go zone on the blueprint that is transformed to the robot internal representation of the environment. However, the authors do not provide details on the system, and just introduce it as a one-to-one mapping with fairly minimal inaccuracies. Additionally, the authors present a user study that supports the idea that non-trained users are more comfortable working with a blueprint than with a SLAM map of the environment.

Contrary to previous works, Boniardi et al. [6] aimed to combine the CAD blueprint with SLAM by online augmenting the floor map with a pose-graph LiDAR SLAM map, allowing the robot to also navigate on environments that differ from the blueprint.

Kakuma et al. [16] and Li et al. [17] also proposed the use of graph structures to align floor maps and SLAM maps. The former used region segmentation and graph matching to enrich the SLAM map with semantic information from the floor map, while the latter develops a topometric map from the environment that only requires high accuracy at selected regions e.g., intersections of corridors.

A topometric approach was also employed by Mielle et al. [18]. Authors developed a custom SLAM system that benefits from matching between the corners and walls in an emergency map and those detected by the robot to merge both means of information into the resulting SLAM map. Authors also proposed a subsequent work [19] in which they segment a sketch map and SLAM map and match the corresponding regions using environment descriptors.

More recently, Wang et al. [7] used Visible Light Positioning (VLP) landmarks to align a layout map with a SLAM map. The authors described the use of a rigid transformation matrix that could translate, rotate and scale the layout map and result in the alignment of both maps, which could be insufficient for large environments where the SLAM map distortion is uneven along the environment.

Shahbandi et al. [20], [21] presented a methodology for the alignment of two maps of the environment by means of region decomposition and a non-linear transformation. However, the computational burden of the system renders it unsuitable for real-world applications, as acknowledged by the authors.

Other topometric methods have proven to be locally accurate for the localization of mobile robots [22], [23], [24]. However, these techniques do not take into account the possibility of using external sources of information, such as CAD blueprints, and are limited to the information provided by the robot during the map construction phase.

In contrast to previous works, our proposed method does not directly combine the CAD and SLAM data, nor does it interfere with existing SLAM pipelines, so it can even be used with existing SLAM maps. Additionally, we are not limited by the topological shape of the environment, as CAD2SLAM provides a pose to pose projection.

Our contribution is in the domain of the adaptive projection of a CAD blueprint onto a SLAM map and vice-versa. We present a novel paradigm to quickly and reliably project a pose in the reference frame of a blueprint to its corresponding pose in the reference frame of a SLAM map, and provide a set of quantitative experiments to evaluate the accuracy of the projection.

III. Our Approach

We want to compute a function $\mathbf{f} : SE(2) \rightarrow SE(2)$, that projects a pose in the blueprint onto a pose in the SLAM map. This projection is sufficient to issue location goals to the platform on the blueprint map, and let its internal algorithms operate on the richer and more up-to-date SLAM map. If we want to monitor the current location of the platform on a blueprint map, we need to apply the inverse mapping. We represent $\mathbf{f}(\cdot)$ as a set of samples $\mathcal{X} = \{\langle \mathbf{X}_i^{\mathrm{b}}, \mathbf{X}_i^{\mathrm{s}} \rangle\}$. Here $\mathbf{X}^{\mathrm{b}} \in SE(2)$, and $\mathbf{X}^{\mathrm{s}} \in SE(2)$ are two corresponding poses in the blueprint and in the SLAM map respectively. We represent the elements $\mathbf{X} \in SE(2)$ with homogeneous transformation matrices $[\mathbf{R} \;\; \mathbf{t}]$, where we omit the last row $[\mathbf{0} \;\; 1]$ for convenience.

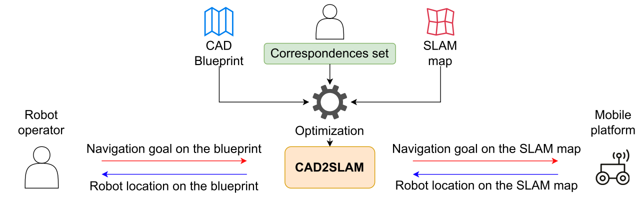
CAD Blueprint
Correspondences set
SLAM map
Robot operator
Optimization
Mobile platform
Navigation goal on the blueprint
CAD2SLAM
Navigation goal on the SLAM map
Robot location on the blueprint
Robot location on the SLAM map

Fig. 2. Diagram of the proposed system. On top, the CAD Blueprint, the SLAM map and the user-selected reference points are provided to the optimizer, that produces the CAD2SLAM projector. On the bottom we represent the pipelines for issuing navigation goals to the robot (red arrows) and announcing the current position of the robot (blue arrows).

To evaluate the value of $\mathbf{X}^{\mathrm{s}} = \mathbf{f}(\mathbf{X}_q^{\mathrm{b}})$ at a query point $\mathbf{X}_q^{\mathrm{b}}$ we proceed by interpolation. More in detail, we consider all samples $\mathcal{X}_q \subseteq \mathcal{X}$ whose Euclidean distance from the query is below a threshold as:

$$\mathcal{X}_q = \left\{ \langle \mathbf{X}_i^{\mathrm{b}}, \mathbf{X}_i^{\mathrm{s}} \rangle \in \mathcal{X} \mid \left\| \mathbf{t}_i^{\mathrm{b}} - \mathbf{t}_q^{\mathrm{b}} \right\| \langle \mathcal{E} \right\} \tag{1}$$

Once the neighbors $\mathcal{X}_q$ of the query point $\mathbf{X}_q$ are computed, we determine the pose in the target map with respect to each element in the set as:

$$\mathbf{X}_i^{\mathrm{s}*} = \mathbf{X}_i^{\mathrm{s}} \cdot \mathbf{X}_i^{\mathrm{b}^{-1}} \mathbf{X}_q^{\mathrm{b}} \tag{2}$$

That is, we compute the query pose in the reference frame of the sample $\mathbf{X}_i^{\mathrm{b}}$ in the domain, and we apply this transformation to the transformed element of the codomain $\mathbf{X}_i^{\mathrm{s}}$.

We compute the final solution as the weighted mean of the $\{\mathbf{X}_i^{\mathrm{s}*}\}$. The weights are inversely proportional to the Euclidean distance between the query point and the sample $\mathbf{X}_i^{\mathrm{b}}$. The translation of the result will be the weighted average of the translations, whereas for the orientations we use circular averaging [25].

A similar procedure can be applied to obtain a pose in the blueprint reference frame given a pose in the SLAM map as illustrated in Fig. 2.

Since $\mathbb{R}^2$ is a subspace of the $SE(2)$ group, the same procedure can be applied to project 2D points in the environment, by neglecting the θ component.

With such a simplistic yet powerful representation we can model arbitrary distortions that include shears, rotations or shortening of corridors, among other common phenomena present in SLAM maps, while maintaining local and global consistency on the projection.

To the best of the authors' knowledge, this is the first paper to present a methodology for the adaptive projection between CAD blueprints and SLAM maps representing the same environment that relies on a graph structure and is supported by a set of correspondence points.

A. CAD2SLAM Optimization

As presented in Section III, the value of the $\langle \mathbf{X}_i^{\mathrm{b}}, \mathbf{X}_i^{\mathrm{s}} \rangle$ pairs is the controlling parameter of the projection in the neighborhood of a pose. In this work, we deterministically sample a number

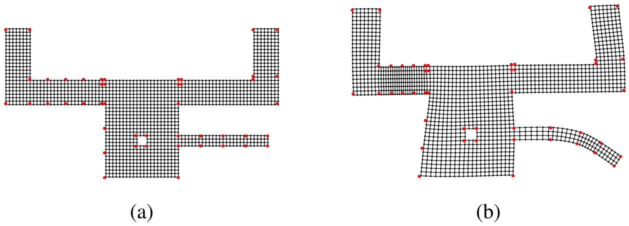

Fig. 3. The CAD blueprint is used to define a non-rigid mesh (a) that is augmented with anchor points in both maps (red circles). After optimization (b), the mesh resembles the SLAM map and a pose in one map can be projected to the other map.

of $\mathbf{X}_i^{\mathrm{b}}$ poses on the blueprint and we propose the use of some reference points in both maps to account for distortions (See Fig. 3).

To obtain the parameters of the set $\mathcal{X}$ we conduct an optimization process in which the value of $\{\mathbf{X}_i^{\mathrm{s}}\}$ is modeled as a parameter of the optimization, under two types of soft constraints:

- The *shape preserving constraints*, that ensure the local structure to the poses $\{\mathbf{X}_i^{\mathrm{s}}\}$ to be similar to the structure to their counterparts $\{\mathbf{X}_i^{\mathrm{b}}\}$ in the blueprint.
- The *anchor constraints*, that enforce pairs of corresponding control points to be the same after the projection is applied.

For convenience, we employ the optimization paradigm of the factor graph [26]. In our problem, the variables of the factor graph correspond to the poses in the blueprint $\{\mathbf{X}_i^{\mathrm{b}}\}$ and the factors are of two types corresponding to the constraints mentioned above.

B. Shape Preserving Constraints

We use a rasterized image of the blueprint and define a non-rigid grid-like mesh for the free space, where each element of the grid is included in the factor graph as $\{\mathbf{X}^{\mathrm{b}}\}$.

To avoid large distortions of the global shape of the corridors and rooms represented in the blueprint, we include a soft constraint to enforce neighboring variables to preserve their relative orientation. Fig. 3 presents an example of the grid mesh on the blueprint and the constraints. We consider 4-neighbors connectivity and model these constraints as pose-pose factors of a pose-graph optimization problem [27] as follows:

$$\mathbf{e}\left(\mathbf{X}_i, \mathbf{X}_j\right) = \log\left(\mathbf{Z}_{ij}^{-1}\mathbf{X}_i{}^{-1}\mathbf{X}_j\right). \tag{3}$$

Here $\mathbf{X}_i$ and $\mathbf{X}_j$ represent two neighboring variables in the grid representation and $\mathbf{Z}_{ij} = \mathbf{X}_i^{\mathrm{b}^{-1}}\mathbf{X}_j^{\mathrm{b}}$ denotes the relative pose of $\mathbf{X}_j^{\mathrm{b}}$ in the reference frame of $\mathbf{X}_i^{\mathrm{b}}$ in the blueprint, while $\log(\cdot) \in \mathbb{R}^3$ represents the pose vector $[x\ y\ \theta]$ extracted from the transformation matrix in the argument. These constraints penalize the configuration of variables that during optimization are moved too far apart w.r.t. their initial relative configuration modeled by $\mathbf{Z}_{ij}$.

For each of these factors, we can also specify an information matrix $\mathbf{\Omega}_{ij}$ that models the error along different directions depending on $\mathbf{Z}_{ij}$. We found it beneficial to allow the factors to be softer along the tangential direction of the factor $\mathbf{t}_{ij}$, and stiffer along the orthogonal direction $\mathbf{t}_{ij}^{\perp}$.

To this end, let λ_x, λ_y and $\boldsymbol{\lambda}_\theta$ be the longitudinal, transversal and rotation components of an information matrix. We compute the information matrix $\mathbf{\Omega}_{ij}$ as:

$$\mathbf{\Lambda} = \mathrm{diag}(\boldsymbol{\lambda}_x, \boldsymbol{\lambda}_y)$$

$$\mathbf{R}_{ij} = \frac{1}{\|\mathbf{t}_{ij}\|}\begin{bmatrix}\mathbf{t}_{ij} & \mathbf{t}_{ij}^{\perp}\end{bmatrix}$$

$$\mathbf{\Omega}_{ij} = \begin{bmatrix}\mathbf{R}_{ij}\mathbf{\Lambda}\mathbf{R}_{ij}^{\intercal} & 0\\ 0 & \boldsymbol{\lambda}_\theta\end{bmatrix}$$

Experimental tests allowed us to validate that the use of this information matrix can limit the occurrence of undesired artifacts, such as bending of the corridors, when $\lambda_x = 1$ and $\lambda_y = \boldsymbol{\lambda}_\theta = 10$.

Fig. 3 presents the *DEMO* environment graph before and after optimization.

C. Anchor Constraints

To account for arbitrary distortions between the blueprint and the SLAM map, we rely on a set of reference points in both maps that will constrain the optimization. Given rasterized images of the blueprint and the SLAM map, an operator can provide a number of points in both images that represent the same position in the real world, such as corners or columns. The more correspondence points in a region of the environment, the more constrained the mapping in that region will be and, consequently, more accurate. Based on this simple heuristic the operator can add more points in those areas where the robot has to carry on specific tasks that require the interaction with other appliances or robots. Moreover, the reference points can be added/removed dynamically, to control the outcome of the optimization.

Visible Light Positioning (VLP) systems [28], fiducial markers [29] or corner detection systems [30], [31] could be used to provide the reference points automatically.

Let $\mathbf{c} = \langle \mathbf{p}^{\mathrm{b}}, \mathbf{p}^{\mathrm{s}} \rangle$ be a pair of 2D reference points, where $\mathbf{p}^{\mathrm{b}}$ denotes the point in the blueprint map and $\mathbf{p}^{\mathrm{s}}$ its corresponding point in the SLAM map. Let $\mathbf{X}_i^{\mathrm{b}}$ be the pose variable on the graph of the blueprint closest to $\mathbf{p}^{\mathrm{b}}$, and let $\mathbf{X}_i$ be the variable in the SLAM map corresponding to $\mathbf{X}_i^{\mathrm{b}}$. If the two maps were identical, the points expressed in the reference frame of the observing poses would be the same. Hence we can add a factor to the graph modeling this constraint, as follows:

$$\mathbf{e}_q\left(\mathbf{X}_i\right) = \mathbf{X}_i{}^{-1}\mathbf{p}^{\mathrm{s}} - \mathbf{X}_i^{\mathrm{b}^{-1}}\mathbf{p}^{\mathrm{b}} \tag{4}$$

As a result, the reference points pairs will enforce the optimization process to account for the deformations present in the SLAM map with respect to the blueprint map. Fig. 3 presents the correspondence points pairs for the *DEMO* environment.

IV. Experimental Evaluation

Once we have constructed our factor graph structure, we employed the SRRG Solver [32] for the optimization by means of Least Squares (LS), which results in the configuration of the variables $\mathbf{X}_k$ that minimizes (3) and (4).

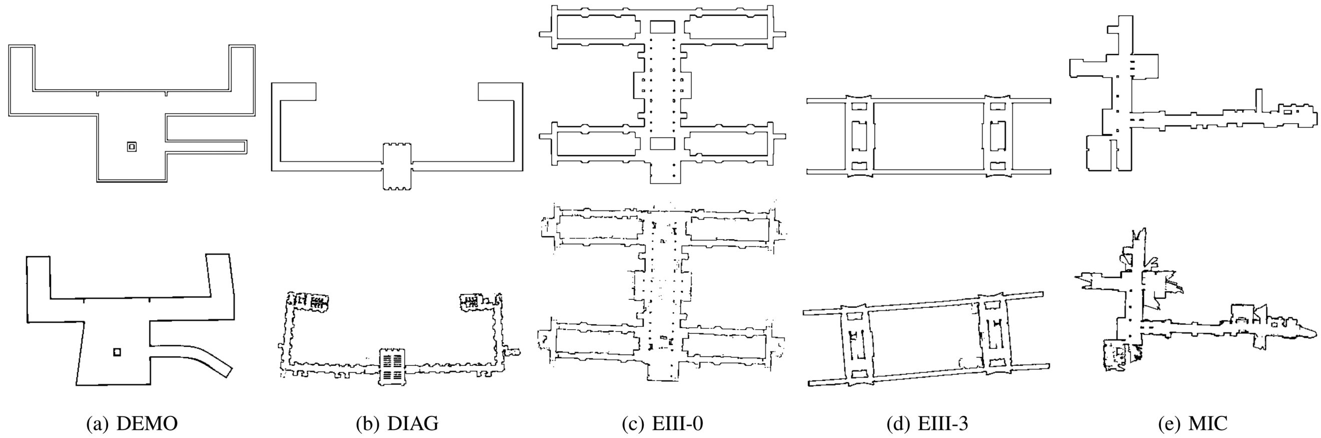

Fig. 4. Test environments used for the experiments. Top row presents the CAD blueprints of the environments, while the bottom row corresponds to the SLAM maps of the environments.

We have conducted a series of experiments to measure the accuracy and validate the usefulness of our proposed method, and we present both quantitative and qualitative results in several test environments.

We equipped a mobile robot with a 2D LiDAR scan and we evaluated CAD2SLAM in the following test scenarios:

- *DEMO:* A synthetically generated environment, designed to prove the effectiveness of our system even in the most challenging scenarios with arbitrary distortions.
- *DIAG:* Dipartimento di Ingegneria informatica, automatica e gestionale, Second floor, Sapienza Università di Roma, Italy
- *EIII-0:* Escuela de Ingeniería Industrial, Informática y Aeroespacial, Ground floor, Universidad de León, Spain
- *EIII-3:* Escuela de Ingeniería Industrial, Informática y Aeroespacial, Third floor, Universidad de León, Spain
- *MIC:* Módulo de Investigación Cibernética, First floor, Universidad de León, Spain

We employed Plug-and-Play SLAM [33] on the real environments to generate the SLAM maps, which are locally consistent. However, its global consistency is affected by drift, generating a bending effect that is more noticeable in the *DIAG* environment because the environment does not have loops while constructing the SLAM map.

Sapienza Università di Roma provided a printed CAD blueprint for DIAG environment, while Universidad de León provided digital CAD blueprints for EIII-0, EIII-3 and MIC environments. We preprocessed the CAD blueprints to only include those regions of the building that are represented in the SLAM map. Please note that for some of the environments, the blueprint might lack much of the details of the SLAM map, due to the environment being cluttered with objects that are not represented in the blueprint, e.g., columns, chairs, cabinets, ornamentation, etc. CAD blueprints and SLAM maps for these environments are presented in Fig. 4.

A. Mapping Accuracy

Although we cannot directly measure the accuracy of our system since neither the blueprints nor SLAM maps can be defined as ground truths, we can indirectly measure the local accuracy for a mapping pair $\mathbf{X}^{\mathrm{s}} = \mathbf{f}(\mathbf{X}^{\mathrm{b}})$ by comparing two virtual scans generated from the corresponding positions in each map. The rationale behind this experiment is that, under perfect circumstances, one should get identical observations, hence measuring their difference gives a hint of the accuracy of the mapping $\mathbf{f}(\cdot)$. Additionally, we seek to recreate a realistic scenario for the use of CAD2SLAM in which a robot is to be controlled by a human operator by issuing goals in the CAD blueprint that are projected to the robot SLAM map.

In this experiment, we ask a human operator to select $\langle \mathbf{p}_r^{\mathrm{b}}, \mathbf{p}_g^{\mathrm{b}} \rangle$, two poses on the CAD blueprint that correspond to the robot current and goal positions respectively. We employ CAD2SLAM as indicated in (2) to obtain $\langle \mathbf{p}_r^{\mathrm{s}}, \mathbf{p}_g^{\mathrm{s}} \rangle$, the projection of those positions in the SLAM map. This process would be sufficient for the robot to navigate to the desired goal using its own navigation stack and SLAM map.

We use an off-the-shelf path planning algorithm to predict the robot path on the blueprint map and applied again CAD2SLAM for every intermediate position $\mathbf{p}_i^{\mathrm{b}}$ of the path to acquire the projection of this path on the SLAM map, $\mathbf{p}_i^{\mathrm{s}}$ as presented in Fig. 5.

Given the path in both maps, we simulate $\mathcal{S}^{\mathrm{b}}$ and $\mathcal{S}^{\mathrm{s}}$, two LiDAR scans from the blueprint and SLAM maps given at $\mathbf{p}_i^{\mathrm{b}}$ and $\mathbf{p}_i^{\mathrm{s}}$ respectively.

We apply again CAD2SLAM for every point in $\mathcal{S}^{\mathrm{b}}$ to obtain $\mathcal{S}^{\mathrm{b}*}$, the LiDAR scan of the blueprint in the SLAM map reference frame, and we compare that to $\mathcal{S}^{\mathrm{s}}$ by using a nearest neighbor criterion:

$$\mathbf{E} = \sum_{1:N} \left\| \mathbf{s}_i^{\mathrm{b}*} - \mathbf{s}_i^{\mathrm{s}} \right\|^2 \tag{5}$$

Please note that the CAD blueprint and SLAM map may differ significantly due to the environment being cluttered with objects that are not represented in the CAD blueprint. To account for this event, we employ a threshold value on the point-to-point comparison to reject the impact of outliers values in the evaluation.

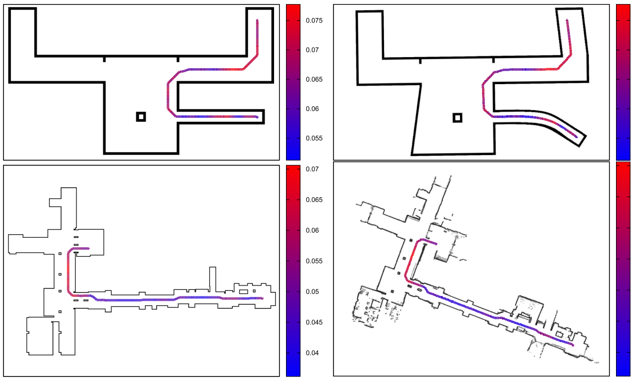

Fig. 5. Robot path on the *DEMO* and *MIC* environments. The path is planned on the CAD blueprint (left) and projected using CAD2SLAM to the SLAM map (right). LiDAR readings are simulated along the path on both maps and the point-to-point error is calculated.

TABLE I
SUMMARY OF THE EXPERIMENTS PERFORMED IN EACH TEST ENVIRONMENTS

Environment	Experiments	Mean LiDAR scans
DEMO	7	551
DIAG	5	2570
EIII-0	9	754
EIII-3	6	2152
MIC	6	665

The number of LiDAR scans depends on the size of the environment and the resolution of the CAD blueprint.

TABLE II
SUMMARY OF THE EXPERIMENTS PERFORMED FOR EACH THRESHOLD VALUE

Threshold (m)	Mean (m)	Std deviation (m)	Outliers
0.01	0.0065	0.0019	355
0.05	0.0326	0.0057	327
0.1	0.0613	0.0106	278
0.5	0.2128	0.0772	107

Lower threshold values result in a higher number of points to be consider outliers.

Table I presents an overview of the number of paths that were used for evaluation in each environment and the mean number of LiDAR scans simulated for each environment, while in Table III we include the aggregated mean and standard deviation values of the point-to-point comparison for every environment, as well as the mean number of points that were considered outliers depending on the threshold value.

For reference, Fig. 5 shows the result of one of the experiments, evaluated on the *DEMO* environment with a point-to-point threshold of 0.1 m and a resolution of 0.05 m/px. In this environment, it is clearly visible a lower error in the proximity of anchor points, as indicated in Fig. 3. However, the maximum error along the entire path stays below 0.077 m, which demonstrates the effectiveness of our method.

Table II summarizes the experiments performed with each threshold. One can clearly note that the mean point-to-point error increases as the threshold increases. This is due to discrepancies between the real world, the CAD blueprint and the SLAM map, as well as small imperfections on the maps. This effect is more noticeable in the *DIAG* environment while *EIII-3* offered the worst results of the environments that were tested, because of it representing an environment with two long corridors without distinguishable elements to use as anchor points. We consider the 0.1 m outlier rejection threshold to be a good trade-off between the outliers counter and the point-to-point error, allowing us to robustly evaluate our method.

TABLE III
POINT-TO-POINT DISTANCE OF THE PROJECTED SIMULATED SCANS

Environment	Threshold (m)	Mean (m)	Std deviation (m)	Outliers
DEMO	0.01	0.0071	0.0016	358
	0.05	0.0374	0.0050	327
	0.1	0.0638	0.0076	245
	0.5	0.1570	0.0301	19
DIAG	0.01	0.0067	0.0020	353
	0.05	0.0315	0.0052	333
	0.1	0.0595	0.0093	278
	0.5	0.02195	0.0389	58
EIII-0	0.01	0.0066	0.0020	357
	0.05	0.0321	0.0055	331
	0.1	0.0634	0.0121	278
	0.5	0.1892	0.0371	58
EIII-3	0.01	0.0066	0.0021	358
	0.05	0.0338	0.0063	341
	0.1	0.0651	0.0100	304
	0.5	0.2635	0.0977	168
MIC	0.01	0.0059	0.0014	355
	0.05	0.0296	0.0024	271
	0.1	0.0518	0.0084	189
	0.5	0.1232	0.0288	39

The complete experiments and the instructions on how to replicate the results are available on the public project web page, while in this paper we report the summary results.

B. Maps Overlay

To illustrate the usefulness of our method and the use cases in which it might be of interest, we have designed a qualitative evaluation in which we overlay the SLAM map with the CAD blueprint of the environment and the projected version of the CAD blueprint using CAD2SLAM.

As introduced previously, a number of phenomena can happen during the construction of SLAM maps, resulting in an altered representation of the environment that does not correspond to the real world nor the CAD blueprint. This effect is even more noticeable when different robots traverse the same environment, each of them with its own SLAM map. This situation is presented on Fig. 6.

Similarly to the previous experiment, we applied CAD2SLAM to project every point in the SLAM maps onto the CAD reference system, so the CAD blueprint can be considered the common interface of all the robots in the environment. As the SLAM map is directly projected to the CAD reference frame, no manual translation, rotation or scaling of the map is needed. Our system also allows for the inverse procedure. The results of this process are presented in the bottom part of Fig. 6.

This map overlay representation is a byproduct of our adaptive projection system, but it is not the target of our system. As indicated previously, our system only requires the correspondence points and the CAD blueprint to compute the projection,

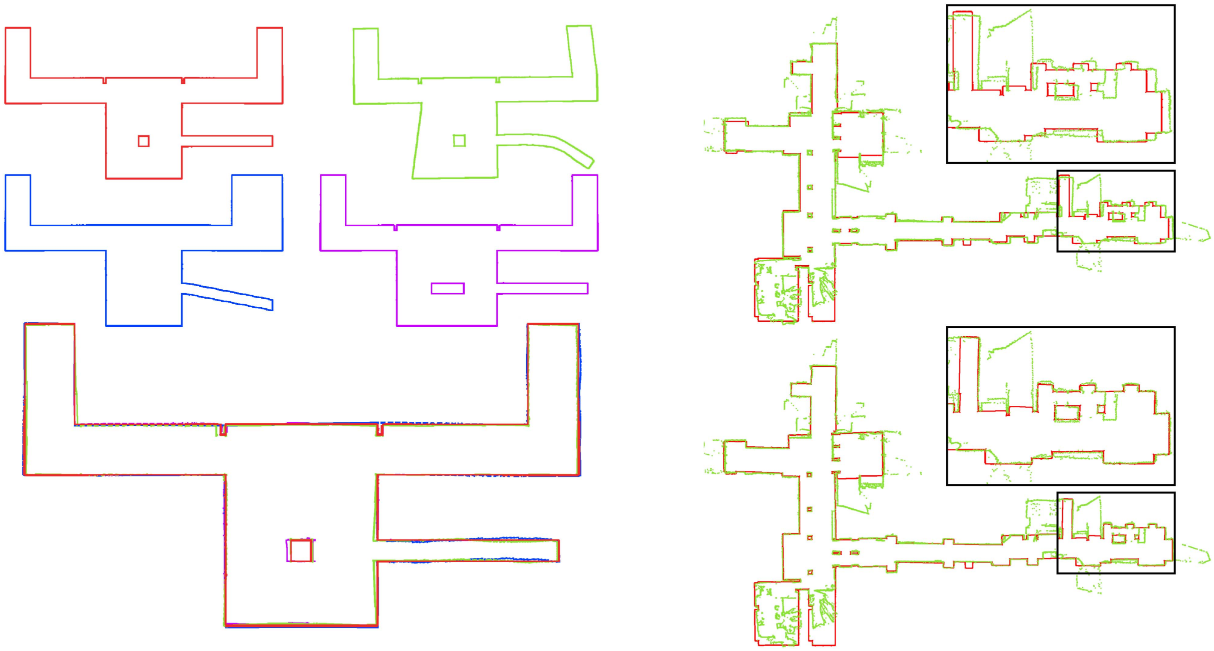

Fig. 6. Illustration of CAD2SLAM in two test environments. On the left, the CAD blueprint (red) is presented along 3 SLAM maps (blue, green, purple) that are affected from different heavy distortions. The SLAM maps are projected onto the CAD reference frame and overlaid on top of the blueprint on the bottom of the image. On the right, the CAD blueprint (red) is projected to the SLAM map (green) reference frame. The proposed method keeps both local and global consistency when projecting every point of the CAD blueprint onto the SLAM map reference frame, or vice-versa, as can be observed in the zoomed region.

TABLE IV
RUNNING TIME FOR THE SETUP OF THE SYSTEM AND DURING ITS NORMAL OPERATION

Environment	**# Variables**	**Graph construction (ms)**	**Graph optimization (ms)**	**Projection mean time (ms)**	**Projection Std. deviation time (ms)**
DEMO	1442	13.975	86.960	0.0429	0.0045
DIAG	2534	72.910	155.14	0.0647	0.0055
EIII-0	715	29.557	29.338	0.0189	0.0019
EIII-3	1888	66.917	91.471	0.0503	0.0049
MIC	3752	59.866	542.05	0.1072	0.0119
Mean	2066	48.645	180.99	0.0568	0.0057

so the proposed method could work even if no SLAM map of the environment is available. This experiment is intended to showcase the ability of CAD2SLAM to adequately project a pose in the CAD blueprint to the SLAM map even in the most challenging environments.

C. Running Time

We performed a running time evaluation using an Intel Core i7-9700 CPU (8 cores @ 3.00GHz) and 32GB of RAM.

The projection time, i.e., the time needed to transform a pose in the blueprint to its corresponding pose in the SLAM map, is critical in this system, as a slow projection could result in important delays in the control and supervision of the robotic platform. We evaluated the running time of our system by measuring the time it takes to project 1000 random poses in the blueprint to their corresponding pose in the SLAM map for each test environment and we obtained the results presented in Table IV. This time is negligible compared to the execution time of the path planning algorithm, which can take up to 290 ms in large real-world environments [34], hence using our projector has virtually no impact on typical robotic platforms and is suitable for online operation.

In contrast to the projection time, the graph construction and optimization times are not relevant for the evaluation of the method online, as it is a one-time operation. Still a slow runtime might hinder the user experience when adding/removing points to the GUI. Our current implementation allows to setup CAD2SLAM in a few seconds even in the most challenging environment, as presented in Table IV.

The time required in these stages clearly depends on the number of variables and factors, that is controlled by the mesh density (see Section III-B) and by the size of the environment. Intuitively, a smaller tile size results in a higher resolution of the projection, at the expense of increased graph construction and optimization times. Similarly, the time required to optimize the graph grows with the number of iterations of the Least Squares solver [32]. For the *MIC* environment, we performed double the iterations of the LS solver with respect to the other environments, to exhibit that even under double the iterations, the optimization is completed in a few hundred milliseconds.

V. CONCLUSION AND FUTURE WORK

In this paper we introduce the problem of the adaptive projection of a CAD blueprint onto a SLAM map and present an alternative solution to the problem of combining different means of representation of an environment. To the best of our knowledge, this is the first work to propose a system which is able to combine a CAD blueprint with a number of SLAM maps from different robots in a robust and efficient manner. We present a novel methodology and an open-source implementation that has been validated through our experiments. Our method robustly compensates common phenomena present in SLAM maps and provides an accurate projection between the two representations of the environment in real time, even when large distortions are present. In future work we will explore the use of corner detection systems, VLP systems or fiducial markers as anchors for our proposed method.

We have developed CAD2SLAM, an easy-to-use pipeline that allows for the seamless projection of poses in the CAD reference frame to the SLAM reference frame and vice-versa. Our system accounts for any arbitrary distortion that may exist between the CAD and SLAM maps even in the most challenging situations and solves the problem of keeping a common reference frame when human-robot or robot-robot cooperation is desired.

We consider that our work presents a novel research question that deserves attention to further investigate the use of a common map to which several systems can project their own maps, in an efficient and robust manner.

References

[1] H. Durrant-Whyte and T. Bailey, "Simultaneous localization and mapping (SLAM): Part I," *IEEE Robot. Automat. Mag.*, vol. 13, no. 2, pp. 99–110, Sep. 2006.

[2] J. Aulinas, Y. Petillot, J. Salvi, and X. Lladó, "The SLAM problem: A survey," in *Proc. Int. Conf. Catalan Assoc. Artif. Intell.*, 2008, pp. 363–371.

[3] C. Campos, R. Elvira, J. J. G. Rodríguez, J. M. M. Montiel, and J. D. Tardós, "ORB-SLAM3: An accurate open-source library for visual, visual–Inertial, and multimap SLAM," *IEEE Trans. Robot.*, vol. 37, no. 6, pp. 1874–1890, Dec. 2021.

[4] T. Taketomi, H. Uchiyama, and S. Ikeda, "Visual SLAM algorithms: A survey from 2010 to 2016," *IPSJ Trans. Comput. Vis. Appl.*, vol. 9, pp. 1–11, 2017.

[5] L. Bowen-Biggs et al., "Sketched floor plans versus SLAM maps: A comparison," *IEEE Commun. Surv. Tut.*, early acess, 2016, doi: 10.1109/COMST.2024.3471950.

[6] F. Boniardi, T. Caselitz, R. Kümmerle, and W. Burgard, "Robust LiDAR-based localization in architectural floor plans," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2017, pp. 3318–3324.

[7] Y. Wang, B. Hussain, and C. P. Yue, "VLP landmark and SLAM-Assisted automatic map calibration for robot navigation with semantic information," *Robotics*, vol. 11, no. 4, 2022, Art. no. 84.

[8] B. Behzadian, P. Agarwal, W. Burgard, and G. D. Tipaldi, "Monte Carlo localization in hand-drawn maps," in *Proc. 2015 IEEE/RSJ Intl. Conf. Intell. Robots Syst.*, 2015, pp. 4291–4296.

[9] G. Grisetti, R. Kümmerle, C. Stachniss, and W. Burgard, "A tutorial on graph-based SLAM," *IEEE Trans. Intell. Transp. Syst. Mag.*, vol. 2, no. 4, pp. 31–43, Winter 2010.

[10] F. Schuster, C. G. Keller, M. Rapp, M. Haueis, and C. Curio, "Landmark based radar SLAM using graph optimization," in *Proc. 2016 IEEE 19th Int. Conf. Intell. Transp. Syst.*, 2016, pp. 2559–2564.

[11] B. Alsadik and S. Karam, "The simultaneous localization and mapping (SLAM)-An overview," *J. Appl. Sci. Technol. Trends*, vol. 2, no. 02, pp. 147–158, 2021.

[12] V. Setalaphruk, A. Ueno, I. Kume, Y. Kono, and M. Kidode, "Robot navigation in corridor environments using a sketch floor map," in *Proc. 2003 IEEE Int. Symp. Comput. Intell. Robot. Automat.*, 2003, pp. 552–557.

[13] M. Mielle, M. Magnusson, and A. J. Lilienthal, "Using sketch-maps for robot navigation: Interpretation and matching," in *Proc. 2016 IEEE Int. Symp. Saf., Secur., Rescue Robot.*, 2016, pp. 252–257.

[14] G. Parekh, M. Skubic, O. Sjahputera, and J. M. Keller, "Scene matching between a map and a hand drawn sketch using spatial relations," in *Proc. 2007 IEEE Int. Conf. Robot. Automat.*, 2007, pp. 4007–4012.

[15] F. Boniardi, B. Behzadian, W. Burgard, and G. D. Tipaldi, "Robot navigation in hand-drawn sketched maps," in *Proc. 2015 Eur. Conf. Mobile Robots*, 2015, pp. 1–6.

[16] D. Kakuma, S. Tsuichihara, G. A. G. Ricardez, J. Takamatsu, and T. Ogasawara, "Alignment of occupancy grid and floor maps using graph matching," in *Proc. 2017 IEEE 11th Int. Conf. Semantic Comput.*, 2017, pp. 57–60.

[17] J. Li, C. L. Chan, J. Le Chan, Z. Li, K. W. Wan, and W. Yun Yau, "Cognitive navigation for indoor environment using floorplan," in *Proc. 2021 IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2021, pp. 9030–9037.

[18] M. Mielle, M. Magnusson, and A. J. Lilienthal, "The auto-complete graph: Merging and mutual correction of sensor and prior maps for SLAM," *Robotics*, vol. 8, no. 2, 2019, Art. no. 40.

[19] M. Mielle, M. Magnusson, and A. J. Lilienthal, "URSIM: Unique regions for sketch map interpretation and matching," *Robotics*, vol. 8, no. 2, 2019, Art. no. 43.

[20] S. Shahbandi, M. Magnusson, and K. Iagnemma, "Nonlinear optimization of multimodsal 2D map alignment with application to prior knowledge transfer," *IEEE Robot. Automat. Lett.*, vol. 3, no. 3, pp. 2040–2047, Jul. 2018.

[21] S. G. Shahbandi and M. Magnusson, "2D map alignment with region decomposition," *Auton. Robots*, vol. 43, no. 5, pp. 1117–1136, 2019.

[22] L. Montano-Oliván, J. A. Placed, L. Montano, and M. T. Lázaro, "G-loc: Tightly-coupled graph localization with prior topo-metric information," *IEEE Robot. Automat. Lett.*, vol. 9, no. 11, pp. 9167–9174, Nov. 2024.

[23] H. Badino, D. Huber, and T. Kanade, "Real-time topometric localization," in *Proc. 2012 IEEE Int. Conf. Robot. Automat.*, 2012, pp. 1635–1642.

[24] M. Mazuran, F. Boniardi, W. Burgard, and G. D. Tipaldi, "Relative topometric localization in globally inconsistent maps," in *Robotics Research*. Cham, Switzerland: Springer, 2018, pp. 435–451.

[25] J. Cremers and I. Klugkist, "One direction? a tutorial for circular data analysis using r with examples in cognitive psychology," *Front. Psychol.*, vol. 9, 2018, Art. no. 2040.

[26] F. Dellaert and M. Kaess, "Factor graphs for robot perception," *Foundations Trends Robot.*, vol. 6, no. 1/2, pp. 1–139, 2017.

[27] L. Carlone and G. Calafiore, "Convex relaxations for pose graph optimization with outliers," *IEEE Robot. Automat. Lett.*, vol. 3, no. 2, pp. 1160–1167, Apr. 2018.

[28] Z. Zhu, Y. Yang, M. Chen, C. Guo, J. Cheng, and S. Cui, "A survey on indoor visible light positioning systems: Fundamentals, applications, and challenges," 2024, *arXiv:2401.13893*.

[29] M. Kalaitzakis, B. Cain, S. Carroll, A. Ambrosi, C. Whitehead, and N. Vitzilaios, "Fiducial markers for pose estimation," *J. Intell. Robot. Syst.*, vol. 101, 2021, Art. no. 71.

[30] N. Prieto-Fernández, S. Fernández-Blanco, A. Fernández-Blanco, J. A. Benítez-Andrades, F. Carro-De-Lorenzo, and C. Benavides, "Conditional weighted linear fitting for 2D-LiDAR-mapping of indoor SLAM," *IEEE Trans. Ind. Informat.*, vol. 20, no. 7, pp. 9579–9587, Jul. 2024.

[31] J.-H. Im, S.-H. Im, and G.-I. Jee, "Vertical corner feature based precise vehicle localization using 3D LIDAR in urban area," *Sensors*, vol. 16, no. 8, 2016, Art. no. 1268.

[32] G. Grisetti, T. Guadagnino, I. Aloise, M. Colosi, B. D. Corte, and D. Schlegel, "Least squares optimization: From theory to practice," *Robotics*, vol. 9, no. 3, 2020, Art. no. 51.

[33] M. Colosi et al., "Plug-and-play SLAM: A unified SLAM architecture for modularity and ease of use," in *Proc. 2020 IEEE/RSJ Intl. Conf. Intell. Robots Syst.*, 2020, pp. 5051–5057.

[34] S. Macenski, M. Booker, and J. Wallace, "Open-source, cost-aware kinematically feasible planning for mobile and surface robotics," 2024, *arXiv:2401.13078*.