Abstract—In many applications of autonomous mobile robots, the following problem is encountered. Two maps of the same environment are available, one a prior map and the other a sensor map built by the robot. To exploit the available information in both maps to the full capacity, the robot must find the correct association between the frames of the input maps. There exists many approaches to address this challenge, however, most of those methods rely on assumptions such as similar modalities of the maps, same scale, or existence of an initial guess for the alignment. In this work we proposed a decomposition based method for 2D spatial map alignment which do not rely on aforementioned assumptions. The proposed method is validated and compared with other approaches from generic data association approaches to map alignment specific algorithms, over real world examples of four different environment with thirty six sensor maps and four layout maps.

I. INTRODUCTION

Consider the task of environment surveying for applications such as industrial automation, search and rescue, or handling hazardous situations. Or imagine the scenarios in which an available map is introduced as prior knowledge to the robot for different objectives. Examples of such objectives and prior knowledge can be an elaborate task planning in the presence of a semantically annotated prior map, improving the performance of SLAM algorithms by exploiting the global consistency of the prior map, and incorporating information such as traffic flow that a single agent could not obtain alone. Furthermore, a hybrid map constructed from merging maps of different modalities, enables access to all included modalities through each individual map. For instance assume that the semantic labels are provided by one map and the robot can localize itself based on another sensor modality. The robot can become aware of each region’s semantic label merely by localizing itself in one map and exploiting the association between maps. All the aforementioned applications share the need for a map alignment procedure. Solving the autonomous map alignment problem has interesting upshots. A seamless map alignment procedure improves the autonomy of robotic services by reducing the demand for the human intervention. The need for map alignment could rise from different motivations. For instance, an autonomous map alignment and merging of partial maps is desired in a multi-agent mapping scenarios. Another scenario in which the map alignment is of particular interest is where a robot is expected to employ a prior map of the environment in addition to its own capacity to create maps. In the latter example, an important challenge is the difference in map formats. In such cases the prior map of the environment is in a CAD drawing format, and the robot maps are often discrete such as Occupancy-like Maps. This example is from the Halmstad Intelligent Home [32]. Other maps also include office environments (see Appendix).

Fig. 1: An example of a pair of layout and sensor maps. Sensor maps are acquired with a Google Tango tablet as 3D meshes, and later are converted to 2D Occupancy-like Maps. This example is from the Halmstad Intelligent Home [32]. Other maps also include office environments (see Appendix).

A. Problem Description

The terms merging and matching appear frequently in this paper, and it is important to avoid a potential confusion. The objective of this work is not to address the “map merging” problem per se (as we do not deal with the fusion of the maps). The map merging is one of the processes that could benefit from the solution to our alignment problem. The very specific objective of this paper is to address an alignment problem between maps. The problem of map alignment is regarded as a data association or registration problem, and more specifically could be stated as “how to find the correct alignment between two input maps in form of a
A common approach to the alignment problem is to interpret the input maps with an abstract representation that enables a search on the similarity of instances. For example, graphs capture the canonical points of the open space as vertices, and the connectivity of the open space is represented by the edges between vertices. Consequently, to find a match between two maps, geometric or topological similarities of the vertex and/or edge instances of the graphs are exploited. When maps are from different [sensor] modalities, such interpretations plays an important role. This interpretation abstracts the input maps into a shared instantiated representation, which makes the search for similarities between maps feasible. Graphs are among the popular representations undertaking this role [24], [50], [27]. Some of the works more relevant to the objective of this paper and some other influential works on this topic are reviewed in Section [V] with more details.

C. Our approach

In this work, we propose a method based on the aforementioned concept of shared instantiated representation. Our proposal relies on “2D-arrangement” [2], which is a geometric interpretation of the maps. When modeling the occupied regions of the maps (corresponding to the physical elements in the environment), the 2D-arrangement explicitly represents both the boundaries and the regions of the open-space. In addition, it implicitly captures the connectivity of the open-space through the neighborhood of the regions. Our proposal is i) interpreting input maps into a shared geometric representation, ii) search for all potential candidates suggested by matching “arrangement” instances (regions) of the two maps, and iii) select the best candidate according to an alignment quality measure. The details of the method, the arrangement representation, and further descriptions are provided in Section [VI]. The main characteristic of our proposed method is in the search approach. Most
approaches generate hypotheses from some initial cues and follow along the progress of those cues. In contrast, we exhaust the search space to avoid following only wrong cues and missing the correct solution. Evidently, this is crucial in case of noisy maps and maps of different modalities. To address the problem of tractability, we constraint the search space and simplify the hypotheses generation and quality evaluation processes. To our knowledge, no algorithms has been developed for solving the alignment problem in this manner. Main contributions of this work are:

- An algorithm for map alignment. Although other algorithms have been proposed for map alignment, however, each operates under specific assumptions, such as supporting only rigid transforms, expecting an initial guess, expecting similar data type on input (same modality), etc. Our algorithms do not rely on any such assumption.
- We propose to use an abstract representation, namely 2D arrangement, that is applicable to interpret maps of different modalities. We use this interpretation for region segmentation, and solving the alignment problem. This interpretation results in a hierarchical representation of maps, where the models on the abstract level are readily available after alignment for other geometric processing and manipulations.
- Finally we share our collection of maps, containing forty maps of four different environments. More details in Section III.

II. METHOD

The essence of our method, as depicted in Figure 2, is to represent input maps with an abstract interpretation that facilitates the search for the alignment solution. This abstraction consists of modeling the physical entities of the environment with 2D geometric objects (such as lines and circles.) These models are then used to partition the environment into separate meaningful regions via an arrangement representation. We have shown, in our earlier works [51], [52], how this interpretation could be used for semantic annotation and region-place categorization of Occupancy Grid Maps. Section II-A describes this interpretation and the process of abstraction. Furthermore, we describe in Section II-B how such an abstract interpretation is adjusted to capture meaningful regions. Matching different regions from one map to another and estimating a transformation between them will result in a pool of plausible hypotheses. While each hypothesis is estimated from matching only two regions, they are evaluated based on how well they align the two maps as a whole. In Section II-D we describe the “match score” metric that is used for evaluating the correctness of the alignments, which in turn is used to pick the winning hypothesis.

A. Abstract interpretation of the maps

We lay the abstract representation used by our method on a 2D-arrangement [2] as an interpretation of the spatial maps. An arrangement represents the 2D-plane via partitioning it according to a set of geometric objects (such as, but not limited to, lines and circles). From this point onward we refer to these geometric objects as geometric traits, and traits for short. An arrangement is identified by a determined configuration of a set of foregoing traits and partitioned atomic regions called faces. Faces of an arrangement are irreducible closed-regions (known as “Jordan Curves”) bounded to a set of geometric traits. Figure 3 demonstrates an arrangement and its components on a toy example.

A set of geometric traits $T$ will result in a unique arrangement $A$ identified by a prime graph $P$, and a set of faces $F$. The prime graph $P$ is constructed through intersecting all traits in $T$. The resulting intersection points construct the set of vertices of the prime graph $V(P)$. Edges of the prime graph $E(P)$ are segments of traits that connect the vertices. As a consequence, the prime graph is a multi-graph, meaning two vertices could be connected via different edges (i.e. traits). The set of faces of the arrangement $F$ are the irreducible closed-regions that are bounded by edges from the prime graph $V(P)$. A neighborhood is an attribute associated with the set of faces $F$. Neighborhood $N(F)$ is defined as a set of tuples of faces. Each tuple in this set identifies a neighborhood relation between a pair of faces. Neighborhood is defined as true for a pair of faces if they share at least an edge from the prime graph $E(P)$.
Fig. 3: An arrangement \( A := (T, \mathcal{P}, \mathcal{F}) \). This example involves straight lines and circles to demonstrate the ability to handle geometric traits beyond straight lines. However, due to the nature of buildings in our maps, our map interpretation with arrangement only relies on straight lines.

A := (T : traits, \mathcal{P} : prime graph, \mathcal{F} : faces)
V(\mathcal{P}) := \{e_i : p_i(x, y) | (\exists j, k \in T)[p_i \in t_j \land p_i \in t_k]\}
E(\mathcal{P}) := \{e_i | e_j = (t_i, v_s, v_e), t_i \in T\}
\mathcal{F} := \{f_i = \{e_j\} | e_j \in E(\mathcal{P})\}
N(\mathcal{F}) := \{(f_i, f_j) | (\exists e_k)[e_k \in f_i \land e_k \in f_j]\}

(a) geometric traits (b) prime graph (c) faces

a) What do geometric traits represent?: We model the physical elements of the buildings (e.g. occupied pixels in occupancy maps) with geometric traits, which represent the boundary between open spaces and occupied (or unexplored) areas. Detection of the geometric traits from 2D-maps could be achieved by common algorithms such as Generalized Hough Transform [4] and Radon Transform [44]. Given that all maps of interest to us could be fairly modeled with only straight lines, in this work we use radiography which is a variation of aforementioned algorithms. It has been shown [51] that radiography is more robust in modeling physical elements of the environment (e.g walls) that suffer from a discrepancy in their continuity. Nonetheless, the arrangement representation is not limited to straight lines. As the physical entities of a spatial map are modeled with geometric traits (e.g. walls with straight lines), an arrangement of a map is interpreted as i) edges of the prime graph as the boundaries of open space, and ii) faces capturing a decomposition of regions of the open-space. This dual characteristic of the arrangement, i.e. representation of a space by its regions and their boundaries, has the merit of capturing two important aspects of the information available in the spatial maps, namely the geometric shape of the regions (the open-space), and topological structure of the open-space through the connectivity between faces of the arrangement.

B. Abstraction compatibility and arrangement pruning

In modeling the spatial maps with geometric traits, we employ a radiography algorithm. This approach has merits of handling noisy data, tackling the challenge of discrepancy in the continuity of such traits, and capturing the global structure of environments [51]. One characteristic of this method is that the traits (in this work, straight lines) are not locally bounded. That is to say, not necessarily every point on a trait corresponds to a physical entity in the environment. A consequence of using such traits in our interpretation is the over-decomposition of areas that are conceptually a single region (e.g. a kitchen or an office). Figure 4 demonstrates the over-decomposition on a real map. On the other hand, the search for transformation hypotheses is performed by matching the faces from two maps. For the hypotheses to be plausible, it is crucial for the input maps to have representations on the same level of abstraction, which we call “abstraction compatibility”. That is to say, if a room is represented by a single face in one map, the same room must be represented by a single face in the other map. This means the same room should not be represented by more than one face, or be merged with a neighboring area. It is this fundamental assumption that makes the generation of hypotheses from “face matching”, a reasonable approach. Based on empirical observations, along all contributing factors, the success rate of our proposed method seems to be most sensitive to this compatibility assumption. We overcome this challenge by pruning the arrangement to a more meaningful representation. The pruning process consists of detecting irrelevant edges of the prime graph \( E(\mathcal{P}) \) that do not correspond to a meaningful notion in the underlying maps. We define meaningful notions as either; i) a physical obstruction between regions (e.g. walls), or ii) a gateway between neighboring regions (e.g. doorway). While the justification for physical obstruction is obvious, the explanation for gateways is that it is not desired for all the open space in a map to be merged to a single faces. In another word, the arrangement must represent a region segmentation. In order to identify whether an edge represents either of these notions, we employ a more concrete (non-abstract) representation of the input maps, namely the distance transform or distance map. The distance map corresponding to the previous example is presented in Figure 4. To identify whether an edge is relevant, the average pixels values from the distance map in a neighborhood of the edge is assigned to that edge as relevance metric. Based on empirical observations we concluded that a simple threshold over this values is sufficient to identify the relevant edges. After this process, those faces neighboring by the means of irrelevant edges are merged. The result of a pruning process applied to the over-decomposed example of Figure 4 is presented in Figure 5. It should be noted that, depending on the context of the application, many other approaches could be devised other than the distance transform to tackle the over-decomposition problem and deliver a reliable region segmentation. The region segmentation could be much more elaborate and advanced, but we found this simple technique satisfactory for our map alignment method. That is because the sensitivity of the alignment method to abstraction compatibility is not critically obstructive. That means, not every region is expected to become a single face, nor every corresponding regions should have compatible abstraction in the maps. The hypotheses generation would satisfy the expectations as
long as there are enough reliable correspondence between faces of the two arrangements. We have observed that our approach of rectifying the arrangement’s over-decomposition problem based on the distance map satisfies the abstraction compatibility assumption.

C. Search for alignment

A hypothesis in the context of this work is a transformation that could potentially align one map to another. Hypothesis generation is the process of proposing such plausible transformation matrices. Relying on the uniform scaling assumption we stated in the Section I-C we restrict the transformations to “similarity” group, that is a composition of translation, rotation and uniform scaling. To propose hypotheses, faces of the open space regions with similar shapes are associated and a transformation is estimated for each pair of faces with similar shapes. These transformations are estimated based on the “Least-squares estimation” method proposed by Umeyama [55]. We define shapes as they appear within the arrangement representation. In this limited sense, the geometric shape of a region (face) is defined as the number of corners and their corresponding angles in each region. Since the transformations are restricted to “similarity”, the angles of vertices are preserved in the transformation. In Figure 25 two examples of correct (in green) and wrong (in red) association, and their consequent transformation are depicted.

a) Oriented Minimum Bounding Boxes: In the presence of too much noise in a map, the pruning of the arrangement might not return clean-cut shapes desired for face matching. One wrong corner missed in pruning process will render the shape of that region useless for matching, if the same error does not occur in the other map for the corresponding region. One example of missed corners is visible at the bottom of Figure 4d. Alternatively, due to such cases, we use Oriented Minimum Bounding Boxes as the shape of regions. This counts as an interpretation of the “well-structured environments” assumption stated in Section I-C.

b) Rejecting false positive hypotheses: Choosing a rudimentary descriptor for the shape matching results in too many false positive hypotheses. For instance, from every rectangle in one map there are four possible transformation to every rectangle in the other map. Our approach to avoid this pitfall lies in the uniform scale assumption. We relax this assumption while estimating transformations between regions and allow them to be affine transforms. For instance, when the affine transformations between two similar rectangles are estimated, only two out of four will have uniform scaling. As for rectangles with different width to height ratios, none of the transformations would be uniformly scaled. Thereafter we reject any transformation that does not qualify as a similarity transform. We observed this simple tactic to be very effective and the number of hypotheses are reduced drastically. This can be seen in Section III Table II where ~ 90% of initial hypotheses are rejected.

D. Alignment match score

To select the winning hypothesis, each hypothesis is evaluated based on how well the arrangements of the two maps ($A_1$ and $A_2$) are aligned under that transformation. To this end, an arrangement match score ($S_A$) is defined to measure the alignment quality of each hypothesis. The arrangement match score is defined as

$$S_A(A_1, A_2) = \sum_{f_i \in F} \min(w(f_i), w(f_j)) \times s_f(f_i, f_j)$$

where, $w(f_i)$ is a weight assigned to individual faces, and $s_f$ is the face match score. The weight is defined as the relative surface area of faces to the surface area of the whole arrangement they belong to

$$w(f_i) = \frac{\text{area}(f_i)}{\text{area}(A)}$$

The larger a face is, the higher weight it will have in the arrangement match score. The face match score $s_f$ is defined as

$$s_f(f_i, f_j) = \begin{cases} \left( \frac{\text{area}(f_i)}{\text{area}(f_j)} \right)^{\frac{1}{e-1}} & \text{if } (f_i, f_j) \in \text{association} \\ 0 & \text{otherwise} \end{cases}$$
where \( f_i \wedge f_j \) is the surface area of faces’ intersection and \( f_i \lor f_j \) is the surface area of faces’ union. Match score of a face with itself (perfect match) equals to one, and the match score of two non-intersecting faces equals to zero. The exponential expression rewards slight improvements close to perfect match more than the slight improvements close to a bad match. And \( \text{association} \) represents pairs of faces from two arrangements that are associated (overlapping) under the transformation. We define \( \text{association} \) based on three conditions. First, for the two faces \( f_i \) and \( f_j \) to be associated, they must enclose each other’s center. The center of a face is identified by the “center of gravity” of the vertices of the face. Condition number two assures a one to one association for cases where multiple faces from one arrangement overlap with a single face from the other arrangement. In such cases, among all faces of \( F_2 \) with their centers enclosed by \( f_i \in F_1 \), the face \( (f_j \in F_2) \) with most similar size (surface area) is associated with \( f_i \). And the third condition is symmetric to the second condition, vice versa for \( f_i \) to \( f_j \). Accordingly, \( \text{association} \) could be defined by the following expression

\[
\text{association} : \{\{f_i, f_j\} | \forall f_k \in F_1, f_j \in F_2, c_1 \wedge c_2 \wedge c_3\}
\]

\[
c_1 : \text{center}(f_i) \in f_j \wedge \text{center}(f_j) \in f_i
\]

\[
c_2 : \exists f_k \in F_2 \mid \text{center}(f_k) \in f_i, d(f_i, f_j) > d(f_i, f_k)
\]

\[
c_3 : \exists f_k \in F_1 \mid \text{center}(f_k) \in f_j, d(f_i, f_j) > d(f_j, f_k)
\]

where \( \text{center}(f_i) \) is the center of \( f_i \) and \( d(f_i, f_j) \) is the difference in surface area of \( f_i \) and \( f_j \).

It is important to note that the presented “arrangement match score” is devised only for the comparison of different environments, two of which are homes and the other two are office buildings. Table I lists the number of available sensor maps and layout maps for each environment.

| Environment | Type | # Sensor Maps | # Layout Maps |
|-------------|------|---------------|---------------|
| HH_E5       | office | 14            | 1             |
| HH_F5       | office | 14            | 1             |
| HH_JIH      | home  | 4             | 1             |
| KPT4A       | home  | 4             | 1             |

TABLE I: A list of all maps of four different environments

That these drawings had to be manually simplified before further processing, due to the presence of furniture and other common appliances. The process involved removing all elements of the drawings, except for the building’s elements (i.e. mainly walls). This simplification can be observed in Figure 2b from Section III. The drawings were converted to Scalable Vector Graphics (SVG) and the geometric traits were obtained directly by parsing the SVG files [42]. In order to acquire segmented regions and for the sake of convenience, the SVG files were converted to bitmap format (PNG) and the same process of decomposition and arrangement pruning based on distance transform has been employed. However, if CAD drawings of the layouts are accessible in a richer format (e.g. DXF or DWG), the process of simplification and parsing could be automated too. Furthermore, if the regions are accessible in such formats, there would not be a need for conversion to bitmap and distance transform for region segmentation.

b) Environment types: We collected data from four different environments, two of which are homes and the other two are office buildings. Table I lists the number of available maps for each environment, and all the maps can be found in Appendix. In total there are forty maps, four of which are layout maps and the rest are sensor maps. Most sensor maps are partial and vary in their coverage of the environment.

c) Maps that violate our assumptions: It is important to note that in some cases the sensor maps violate some of our initial assumptions. For instance, maps HH_E5_2, HH_E5_3, HH_E5_4 and HH_F5_2 only cover corridor and hall areas and do not contain any rooms, and therefore there are not enough segment-able regions for hypotheses generation. Or HH_E5_12 and HH_F5_1 suffer from a lack of global consistency as they are visibly bent. There exists further minor defects in some other maps. Consequently, the performance results as presented in this section are not the representative of the method’s performance under all those stated assumptions. Nevertheless, we included these maps to better explain the effects of aforementioned assumptions on the method and portray a fair picture of the method’s performance under different conditions.

d) Evaluations are based on success rate: In advance to presenting the performance of our proposed method and its comparison to other approaches, we must mention that we only provide a success rate as the performance of each method. We skip a quantified accuracy measure for the alignment. It proved very hard (impossible for our data) to provide a per map “alignment accuracy” measure, due to i) the lack of ground truth for the sensor maps which is an obstruction enough in the way of an alignment accuracy metric, and ii) the presence of severe noise and varying level
Fig. 5: Examples of correct, correct with minor defect, and wrong alignments. In calculating the success rate of different methods, both “correct”, “minor defect” are counted as successful alignments.

of global inconsistency of the sensor maps. And as for a per-map alignment accuracy, there is no established method for that to our knowledge. Figure 5 illustrates our quality assessment of the alignments.

B. Implementation

All our experiments were based on an implementation of the proposed method in Python (publicly available here [https://github.com/saeedghsh/Map-Alignment-2D/]). Our implementation is built on many libraries from the SciPy [26] eco-system, namely NumPy [56], Matplotlib [25], SymPy [33], scikit-learn [41] and scikit-image [57], Open-CV library [10], Polygon package [43] (a Python wrapping of the General Polygon Clipper library (GPC) [37]) and NetworkX library [20] are the other included libraries.

C. Experiments and results

We carried out experiments in three different setups and with different objectives:

- **sensor map to layout map alignment**: to demonstrate the performance of the proposed method on aligning sensor maps to layout maps, which is the main objective of the proposed method.
- **sensor map to sensor map alignment**: we observe how partial coverage, noise and inconsistency of sensor maps are amplified, and this results in a lower success rate compared to the alignments that involve layout maps. Since some of the important methods chosen for comparison only operate on same modality maps and can only estimate rigid transformations, this setup also makes the comparisons with those methods possible.
- **evaluation of alignment match score**: where we align every sensor map to all other sensor maps, whether from the same environment or not. In this experiment we attempt to study the “alignment match score” in alignment of intra-environment and inter-environment maps.

a) **Sensor map to layout map alignment**: Table I presents the performance of the method in aligning sensor maps to layout maps of the same environments, with intermediate details. The columns are “initial”: the number of remaining hypotheses after rejecting non-uniformly scaled transformations (∼90% are rejected), The last column states whether if the alignment was successful or not. In total, the method has successfully aligned all maps for home environments, and 78% of maps for office buildings. We believe the failures are mainly due to the violation of the prior assumptions, such as global inconsistency and lack of enough segmentable regions.

b) **Sensor map to sensor map alignment**: Table II compares the success rate of aligning sensor maps to sensor maps and layout maps in different environments. We can observe from the results of sensor to sensor map alignment, that the success rate is lower than the experiments with layout maps. There are two main reasons for this drop; i) many sensor maps are partial and consequently their overlap with each other is partial, ii) the violation of initial assumptions. In the presence of layout map there is one source of noise and global inconsistency. But in case of aligning two sensor maps, such noise and inconsistencies are amplified.

c) **Evaluation of the alignment match score**: In the last set of experiments we try to assess the behavior of alignment match score. Figure 6a represents the match score of the winning hypotheses for every pair of sensor to sensor map (includes pairing sensor maps of different environment).
Gray-scale values represent the alignment match score of the winning hypotheses between each pair of maps ($0 \leq S_A \leq 1$). The blue bordered cells on diagonal represent the alignment of sensor maps versus the layout maps, and the red lines separate different environments. Green and red dots, respectively, mark the success and failure of alignments. As expected, the square matrices on the diagonal of the main matrix which represent the alignment match scores of maps from the same environments are slightly brighter. However, this is not conclusive enough to exploit this measure across different environments. Under scrutiny it can be observed that a smaller environment (KPT4A) could align with a bigger environment (HH\_E5) with a high score. Also, some maps of the same environment have low match score due to the small overlap or violation of the prior assumptions, even though they are successfully aligned. In the same line of argument, Figure 6b presents a box plot of the alignment match score for all hypotheses for each pair of sensor-layout maps. In this figure, the score of the winning hypotheses are marked red and green, representing the failure and success of the alignment. There seems to be a cut-off point on the match score value across all maps, which separates successful alignments from failures. However the margin of this cut-off is not wide enough to be used as a threshold between success and failure. The take away message of the Figure 6b is, as we suspected, the value of match score is not a reliable indicator of alignment success. In conclusion we can say, although the proposed alignment match score has proven useful in finding the correct solution among hypotheses of pairs of maps, yet it is not conclusively reliable for detecting whether if the input maps are from the same environment, nor to autonomously detect a successful alignment.

D. Comparison with other approaches

Prior to development of the presented method, we have experimented with different approaches to solve the alignment problem with our maps. These methods belong to two categories. The first contains generic approaches in data association and alignment, such as image alignment, image registration and point set registration. The other category consists of approaches that are specifically designed for map alignment, such as Hough transform based algorithms. While an in-depth review of these approaches is provided in Section [IV], here we assess the performance of some exemplary algorithms from each approach. The success rate for each method is available in Table [IV]. Furthermore, a brief account of computational cost will follow, and presented in Table. [V]

### TABLE III: Success rate of the method in different environments.

| Environment | Sensor vs Layout | Sensor vs Sensor |
|-------------|-----------------|-----------------|
| HH\_E5      | 64.28% (9/14)   | 50.54% (49/98)  |
| HH\_F5      | 92.85% (13/14)  | 68.13% (62/91)  |
| HH\_HH      | 100% (4/4)      | 100% (6/6)      |
| KPT4A       | 100% (4/4)      | 83.33% (5/6)    |

(a) Match score of the winning hypothesis for each alignment. Diagonal entries: sensor-layout alignment, otherwise: sensor-sensor alignment.

(b) Match score for all hypotheses for each pair of sensor-layout maps.

Fig. 6: A detailed overview of the match score.

a) **Generic data association methods**: Enhanced Correlation Coefficient (ECC) Maximization [16] was used for image alignment approaches. Image registration was tested with Scale-Invariant Feature Transform [30] in combination with Fast Approximate Nearest Neighbors [36] for feature matching. The implementations of these methods were done in Python and based on the Open-CV library [10]. We observed that these algorithms perform slightly better if they are applied to the distance transform of the maps, instead of directly operating on the occupancy maps. And the presented results are accordingly based on distance images of the maps. For the experiments with Point Set Registration we converted the occupied cells to a point set, and employed the Coherent Point Drift (CPD) [39] [38], based on a Python implementation of the algorithm [28]. Since the three above-mentioned algorithms are capable of handling affine transforms (including scaling), we performed the experiments on both sensor to sensor map alignment,
and sensor to layout map alignment. All the results are available in Table [V] The approach based on SIFT features works best on maps with unique visual patterns ordered in a unique constellation, and consequently has a slightly better performance on bigger maps with more “key points”. Although the data-level similarity between sensor maps is in favor of resulting more similar features, however, it yields best results in aligning sensor maps to layouts of bigger environments due to a higher overlap ratio. On the other hand, ECC maximization based approach performs worse on aligning sensor maps to layouts, due to its higher sensitivity to data-level similarity. Nevertheless, a detail review of the results reveals that the main causes of ECC failure in aligning sensor maps is the global inconsistency of the sensor maps and small overlap between the maps. Also, one contributing factor to its success was the cases where the initial alignment of the sensor maps was close to the correct solution. That is to say, ECC is more likely to succeed if the displacement between maps is not drastic. As for the CPD, it is superior to Iterative Closest Point (ICP) method as it handles affine transformation estimations. However, it is computationally very expensive and demands high memory. Consequently we had to employ this method on a subset of the original point set. This in turn result in further concealing of the structural pattern of the maps, and becoming more sensitive to local minima.

b) Map alignment specific methods: In this category we selected the works by Carpin [11] and Saeedi et. al. (PGVD) [45]. Implementations of both algorithms are made publicly available by the authors. Hough based methods are limited to rigid transformation, and as a result we only experimented with sensor to sensor map alignment (no layouts). One interesting aspect of Hough based methods is their independency of the assumption of map’s “segment-able regions”. Therefore they could be considered to have a broader scope of target applications. For instance, we have observed that such methods perform better on maps that mostly contain corridors, which is a challenge for the region segmentation phase of our proposed method. Also, thanks to the underlying decoupling of rotation and translation estimation, they could be relatively faster than other methods. However, these advantages come with a price in performance, while these methods perform better on particular cases, they do have a lower overall success rate over our collection of maps. By inspecting individual results, we observed that many of the failures were due to a wrong orientation alignment. And many of those cases which survived the orientation estimation with a correct result, they still failed the translation estimation. Fundamentally, these methods exploit the structural similarities in maps, by finding similarity is Hough spectra and cross correlating the maps after orientation alignment. We believe the severe noise, global inconsistencies, and repetitive patterns of our maps are among what challenges the above-mentioned foundations of Hough based methods. It is important to note that due to a lack of proper insight to the implementations of these methods, we could not adjust the parameters to maximize their performance in the setting of our experiments. Therefore we would like to point out that the success rate of those algorithms presented in Table [IV] shall not be counted as their best performance, but rather providing an insight into advantages and drawbacks of each method.

c) Time comparison: The timings of all methods are provided in Table [V] All the experiments were ran on a computer with a 4 × 2.7 GHz for CPU, and 8 GB of memory, running Ubuntu 14.04. They are separated to home and office building categories, which provides a sense of scalability with respect to map size. The average map size for home environments is 2.2e^5 pixels, and it is 1.0e^6 pixels for office buildings (roughly 5x bigger). Since CPD is expensive and not scalable, the original point sets were reduced from 1.2e^4 points on average in small maps and 3.3e^4 points on average in bigger maps, to 500 (close to memory limit of the algorithm on our hardware.) Therefore a meaningful computation time could not be provided here.

In comparison, our method falls behind most other approaches in terms of computational cost. Specifically those methods designed for real-time applications such as Carpin’s method [11] for multi-robot mapping, are extremely fast and hard to beat. At the end, we would like to emphasize that the timings of each algorithm provided here can portray a rough scale, and should not be taken as an accurate computational cost comparison. This is mainly due to the heterogeneity of the implementations (C++, Python, Matlab). Furthermore, some of the algorithms are borrowed from other context (e.g. CPD, ECC) and applied to map alignment problem. Some are intended for offline applications with not much concern for computational time, while others were specifically designed to be fast for real-time applications. As a result, these computation times are not sufficient to generalize on the performance of each approach.

IV. RELATED WORK

The objective of this work is to address the problem of map alignment with specific assumptions, as described in previous sections. The main underlying challenge in this line of work is data association, which manifests in a variety of forms according to the application context. Few examples are image registration (e.g. stereo vision correspondence, optical flow, and visual odometry), laser scan matching and point cloud registration in Simultaneous Localization and Mapping (SLAM), and the correspondence problems in SLAM such as loop closure and partial map merging. While above-mentioned problems share the underlying challenge of data association, different methods formulate their underlying problem differently depending on the context of the application, data type, and prior assumptions. While we try to point out some of the seminal works with formulations other than those related to our work, we turn the focus of the literature review to those closest to ours.

a) Alignment of prior map to sensor map, and motivations: There are different motivations for fusing prior maps and sensor maps. For instance, Sanchez and Branaghan argue that abstract maps are easier to learn [48], and accordingly,
TABLE IV: Success rates (in %) of different methods on map alignment. Not Applicable (NA) refers to the methods that are not able to handle scaling.

| method                  | sensor to layout | KPT4A | sensor to sensor |
|-------------------------|------------------|-------|------------------|
| ECC maximization [16]   | 0.00             | 37.36 | 37.36            |
| SIFT [30]               | 21.43            | 8.79  | 0.00             |
| CPD [39] [38]           | 0.00             | 10.98 | 0.00             |
| Saeedi et. al. (PGVD) [45] | NA              | 16.48 | 0.00             |
| Carpin [11]             | 64.28            | 50.54 | 0.00             |

TABLE V: Average (and standard deviation) of computation time (in seconds) of different methods, separated for home and office environment.

| method                  | implementation | time in seconds |
|-------------------------|----------------|-----------------|
| ECC maximization [16]   | Python         | 32.79(28.24)    |
| SIFT [30]               | Python         | 0.20(0.05)      |
| Saeedi et. al. (PGVD) [45] | Matlab    | 4.91(1.42)      |
| Carpin [11]             | C++            | 3.07e-4(9.28e-5)|
| our method              | Python         | 8.86(2.13)      |

Georgiou et. al. [18] state that a correspondence between an abstract human readable map and robot’s sensor map is desired to facilitate collaborative tasks between human and robots. Bowen-Biggs et. al. claim that sensor maps are not “natural” for many high level tasks [9], especially those including semantics or human in the loop. In their work, they present a method of fusing the two sensor and floor maps and using the combination for accomplishing elaborate tasks. However, in their work the map to map correspondence is established manually. In other examples the prior map is exploited towards improving the performance of SLAM algorithms [18], either through exploiting the structure of the prior map, or by aligning local maps to build a global map. Georgiou et. al. formulated the “structural information from architectural drawings” as an “informative Bayesian mapping priors” in order to improve the performance of the SLAM algorithm [18]. Although, this work does not address the map alignment problem per se, instead the SLAM output is structured according to the prior information embedded into the SLAM algorithm. Vysotska and Stachniss [58] proposed an approach to improve SLAM performance by generating constraints from the correspondence between the building information from OpenStreetMap and the robot’s perception of its surroundings. They also benefit from the “localizability” information available in the OpenStreetMap. Mielle et. al. [35] propose a method for applications with extreme conditions (e.g. with dust or smoke) where the information from a “rough prior” is incorporated in order to improve the SLAM performance, and enhances the quality of the rough prior map by fusing with sensor map.

b) Graph matching approaches: The topology of the open space’s structure is one of the most salient information in the maps, and it is natural that the graph representation of the aforementioned structure draws much attention as a fitting representation. Two of the sub-problems in graph theory that are most relevant to map alignment are the Maximal Common Sub-graph (MCS) problem, and the error-tolerant sub-graph isomorphism [24]. Huang and Beevers [24] proposed a method for merging partial maps based on the embedded topological maps. Their approach is based on a graph matching process inspired by maximal common sub-graph (MCS) and image registration, and followed by a second stage that evaluates the geometric consistency of the match hypotheses. The vertices of the topological map are embedded in a metric space, along with a minimal amount of metric information (e.g., orientation of edges at each vertex and path length for each edge). Therefore, their method benefits from both the geometric and topological information of the open-spaces. In another work with a similar approach, Wallgrün [59] proposes a map alignment technique with a graph matching method based on the Voronoi graph of the maps. The objective of his work is localization and mapping, and the underlying data association model of his proposed method is based on an inexact graph matching with graph edit distance, over annotated graphs generated from the Voronoi graphs. Nodes are annotated with the radius of the maximal inscribed circle used to generate the Voronoi graph, and the edges are annotated with their relative length, the shape of the Voronoi curve beneath the edge, and the edge’s traversability. By assigning such attributes to the elements of the graph, he incorporates geometric constraints into the matching process. In order to develop an automated process for map quality assessment, Schwertfeger and Birk have developed an interesting method for map alignment [50]. Their approach is based on capturing the high-level spatial structures of maps through Voronoi graphs and represented with topological graphs that contain the angles between edges and the length of edges. The map alignment is done by finding similarities between vertices of the graphs and “identification of sub-graph isomorphisms through wave-front propagation” [50]. With experimental results, they show the robustness of their method by detecting brokenness in sensor maps. In another intriguing work, Mielle et. al. [34] proposed a map alignment method based on graph matching, towards enabling robots to follow navigation orders specified.
in sketch maps. The Voronoi skeleton is converted to a graph, where vertices are the bifurcation and ending points of the skeleton. Vertex type (dead-end or junction) and an ordered list of edges are attributed to the graph’s vertices in the matching process. To find the error-tolerant maximal common sub-graph (ETMCS), they developed a modified version of Neuhaus and Bunke’s [40] graph matching algorithm based on the normalized Levenshtein edit-distance (LED) [61]. By skipping the absolute position values, the interpretation becomes insensitive to noise and inconsistency of the map and their method doesn’t require global consistency, that it to say non-uniform scaling is allowed to handle sketch maps. In order to benefit from semantic information available in “floor maps” for high-level task execution, Kakuma et. al. [27] proposed a graph matching based approach for the alignment of sensor maps to floor plans of the buildings. In their proposed method, an occupancy map is segmented into regions and a graph is constructed from such regions. Graph matching is carried out with minimizing a matching cost function based on a variation of Graph Edit Distance (GED) [49] and Hu-Moments [23].

c) Hough/Radon transform approaches: Hough (/Radon) transform maps the input signal from the Cartesian coordinate frame to a parametric space. This parametric space has the advantage of capturing the salient, thought maybe latent, structure of the robot maps. The core of those methods based on Hough transform is to decompose the alignment problem into rotation and translation estimation. Such approaches are often deterministic, non-iterative, and fast, thanks to this decomposition. However, methods in this category are limited to rigid transformation estimation, and work best on maps of similar modalities. For merging maps in a multi-robot application, Carpin proposed a method [11] that starts by computing the correlation between the Hough spectra of two maps and finding the rotation alignment, and then estimating the translation parameters from a x-y projection of the maps after the orientation alignment. One of the interesting features of this method is that the estimated transformations are weighted and such weights could be treated as uncertainties. With a conceptually similar approach Bosse and Zlot [8] tackle the problem of global mapping by merging local maps. Consistent with the other Hough transform approaches, their proposed method also decouples the rotation and translation estimation, with some twists in their transformation techniques. They use an “orientation histogram of the scan normals” (yields output similar to a Hough transform) to determine the orientation alignment and a “weighted projection histograms created from the orthogonal projections” (somewhat equivalent to radiography) for estimating the translation between orientationally aligned data. Saeedi et. al. [45], [47], [46] proposed a novel technique to represent the topology of the open space with a probabilistic Voronoi graph. Even though they employ a graph representation, they do not solve the matching problem by graph matching techniques. First a Radon transform is employed to find the relative orientation between maps, followed by an edge matching technique based on a 2D cross correlation over graphs’ edges to find the translation. One of the very interesting features of their method is the propagation of the uncertainty from input map to the Voronoi graph, and accounting for this uncertainty in the fusion process.

d) Optimization approaches: One of the most popular categories of techniques for data associations in robot mapping is to decompose the alignment problem into rotation and translation estimation. A famous example is the Iterative Closest Point (ICP) [5] which is a point set registration and finds a rigid transformation between two point sets. Such an approach is inherently susceptible to the problem of local minima. They are only suitable to problems where a [rough] initial estimate of the displacement between input data is available. While this is a reliable assumption in incremental mapping, it is not a valid assumption in map alignment. Furthermore, such methods work on same modality input data. Other similar approaches in image alignment, such as Lucas-Kanade algorithm [31], [3] and Enhanced Correlation Coefficient (ECC) Maximization [16], also work under similar assumptions and consequently they are prone to similar pitfalls. One example of optimization based method applied to the map merging problem is presented by Carpin and Birk [12], [13], [6]. Their approach minimizes a dissimilarity function (overlapping quality index) over the transformation parameters, with a stochastic process, the random walk, used for the optimization. An interesting feature of this method is the ability to robustly handling unstructured environments.

e) What else?: It is good to mention some other interesting approaches, even though we did not find them particularly relevant in order to investigate them in detail and experiment with them. Among those are methods from the multi-robot mapping applications where the alignment of individual maps are determined by localizing each robot in the partial maps of other robots. Works by Thrun [53], Dedeoglu and Sukhatme [14], and Williams et. al. [60] are good examples in this category. These methods are based on the assumption that the input maps are from the same modality. With similar application, i.e. multi-robot exploration, some researchers have developed methods to determine the relative transformation between robots’ partial maps when they can physically meet in the environment. Examples of the methods based on “rendezvous” or “mutual observation” are proposed by Howard et. al. [22], Howard [21], Fox et. al. [17], Zhou and Roumeliotis [62], and Konolige et. al. [29]. These methods are based on the robots’ ability to meet and generate transformation hypotheses from a rendezvous, an infeasible assumption for off-line methods. Erinc et. al. proposed a method [15] to annotate heterogeneous maps with WIFI signal that provides cues for data association between maps. This means two essentially different maps as annotated by a shared landmark which provides a seamless association between maps. Boniardi et. al. [7] developed a method for localizing and navigating directly in a sketch map, without the map alignment.
Rigid transformations are limited to modality have similar scale, and as a result, such methods inconsistencies of sensor maps. In addition, maps of the same data similarity based methods are prone to severe noise and on sensor level similarity of the input maps. Sensor-level where maps are from similar modalities, hence they rely setup. Most often they are designed to perform map merging where maps are from similar modalities, hence they rely on sensor level similarity of the input maps. Sensor-level data similarity based methods are prone to severe noise and inconsistencies of sensor maps. In addition, maps of the same modality have similar scale, and as a result, such methods are limited to rigid transformations estimation. Such assumptions do not hold where maps of different modality, such as sensor map and layout maps, are to be aligned. Sensor maps and layout map have essential difference on data-level. Also, the scaling from one map to the other adds a new dimension to the search space and the desired solution becomes a similarity transformation rather than a rigid transformation.

We have shown, with experimental results, the insufficiency of generic data association methods (e.g. SIFT features, ECC maximization), and some map alignment specific methods (designed for aligning maps of same modalities) in our experimental setup. Due to inability of methods that only estimate rigid transformation to handle scaling, the comparison of such method to our proposed method had to be limited to sensor to sensor map alignment. Nevertheless, we observed that the presence of noise and global inconsistency has been very challenging for most other approaches. Our proposed method relies on the notion that most human built environments are composed of regions. Accordingly, our method finds the correct alignment by associating regions and selecting the best hypothesis among all candidates. By exploiting the notion of region and founding our method on spatial decomposition, our alignment method operates on a higher level of abstraction. As a consequence, the method is more robust to dissimilarity and heterogeneity of the sensor-level data. Furthermore, the approach of aligning regions rather than associating sensor-level data enables our method to handle the scaling factor like any other transformation parameter.

A. Discussion

In the result section we tried to provide a thorough performance comparison between our proposed method and other approaches to solve the map alignment problem. It should be noted that, we partly credit our method’s out performing other alternatives to the fact that our method is designed to solve the map alignment problem with specific assumptions and experimental settings. We do not claim, or believe, that the method proposed in this paper is superior to other approaches in a generic problem formulation of data association. Rather, we tend to emphasize the particular characteristics and advantages that this method offers over alternatives in specific challenges, namely aligning maps from different modality, severe data level noise, and maps of different scales. However, there might be some other objectives close to the core of the map alignment problem that our method falls short of. Examples of such applications are, aligning maps of unstructured environment and real-time applications.

a) Advantages: Apart from the higher success rate of our proposed method over the presented experimental setup, we would like to point out some other interesting features of the methods. One important aspect, and one of the main motivation behind this work, is the ability to align maps of different modality and specifically sensor maps to layout map. As stated earlier, such a task demands a method that is indifferent to heterogeneity and different scales of input maps. Our proposed method shows a considerable performance for such cases. Although we have developed a regions segmentation method based on the arrangement representation and distance transform, but the general framework of our alignment method is not dependent on any specific region segmentation technique. Our decomposition based algorithm would be able to find the alignment as long as the input maps are effectively interpreted by the arrangement of the 2D plane. That is to say, as long as the input maps are spatial and could be segmented to meaningful regions, the proposed method in this work could be employed to find the alignment. We speculate that an improved region segmentation will have a positive effect on the performance of this alignment approach. It is worth mentioning that the implementation of our proposed method, and the accompanied experiments presented in this paper, convert both maps to occupancy-like bitmaps in advance. However it is not a requirement of the proposed alignment algorithm, but rather it was a convenient choice. And finally, the intermediate representation that is constructed for alignment, by itself is a useful representation for different objectives [52], [51]. This representation is also compatible with the IEEE Standard for Robot Map Data Representation [1], which means not all intermediate interpretations are alignment-specific.

b) Drawbacks and limitations: The main disadvantage of the proposed method is the computation time. This means that this method is not suitable for real-time applications. While exploiting the notion of meaningful regions comes with aforementioned advantages, it also limits the applicability of the method. Dependency on the region segmentation means it most likely will fail in maps of environments cluttered with furniture, or in maze-like environment, unless an appropriate region segmentation algorithm is employed. As stated before, in Section III, not all the maps satisfy our initial assumptions such as global consistency. Consequently, the performance results as presented are not the representative of the method’s performance under all stated assumptions. Nevertheless, we included these maps to better explain the effects of aforementioned assumptions.
on the method and portray a fair picture of the method’s performance under different conditions. Other conditions that makes the proposed method unsuitable occurs when the prior assumptions are violated. Examples are non-uniformly scaled maps like sketch maps, and maze-like environments such as underground tunnels and alike where the notion of meaningful regions might not apply.

B. Future work

In the continuation of this work we intend to address some interesting questions which were raised during the development of this work. One of those questions is the challenge of autonomous detection of successful alignments. This problem can be translated to a classification task, where alignment match score could be a multidimensional vector based on other sources of information in addition to arrangement based match score, such as graph matching metrics (e.g. GED), and data level distance between maps. Towards that objective, we intended to enrich our collection of maps with a wider variety of environments. Further more, we intend to carry out more challenging experiments and with other modalities to inspect the performance of the proposed alignment framework under different circumstances.

The direction of our future work is towards merging maps after alignment. Specific examples of features to contain in a merging process would be the transferring of semantic labels from layout map to sensor map for high level task planning, and detecting and compensating global inconsistencies in sensor map by exploiting the consistent structure of layout maps.

REFERENCES

[1] . IEEE standard for robot map data representation for navigation. 1873-2015 IEEE Standard for Robot Map Data Representation for Navigation, pages 1–54, Oct 2015.

[2] Pankaj K. Agarwal and Micha Sharir. Chapter 2 - arrangements and their applications*. In J.R. Sack and J. Urrutia, editors, Handbook of Computational Geometry, pages 49 – 119, North-Holland, Amsterdam, 2000.

[3] Simon Baker and Iain Matthews. Lucas-kanade 20 years on: A unifying framework. International Journal of Computer Vision, 56(3):221–255, 2004.

[4] D.H. Ballard. Generalizing the hough transform to detect arbitrary shapes. Pattern Recognition, 13(2):111 – 122, 1981.

[5] P. J. Besl and N. D. McKay. A method for registration of 3-d shapes. IEEE Transactions on Pattern Analysis and Machine Intelligence, 14(2):239–256, Feb 1992.

[6] A. Birk and S. Carpin. Merging occupancy grid maps from multiple robots. Proceedings of the IEEE, 94(7):1384–1397, July 2006.

[7] F. Boniardi, B. Bezhadian, W. Burgard, and G. D. Tipaldi. Robot navigation in hand-drawn sketched maps. In 2015 European Conference on Mobile Robots (ECMR), pages 1–6, Sept 2015.

[8] Michael Bosse and Robert Zlot. Map matching and data association for large-scale two-dimensional laser scan-based slam. The International Journal of Robotics Research, 27(6):667–691, 2008.

[9] Leo Bowen-Biggs, Suzanne Dazo, Yili Zhang, Alexander Hubers, Matthew Rueben, Ross T. Sowell, William D. Smart, and Cindy Grimm. Sketched floor plans versus SLAM maps: A comparison. CoRR, abs/1606.04836, 2016.

[10] G. Bradski. OpenCV library. Dr. Dobb’s Journal of Software Tools, 2000.

[11] Stefano Carpin. Fast and accurate map merging for multi-robot systems. Autonomous Robots, 25(3):305–316, 2008.

[12] Stefano Carpin and Andreas Birk. Stochastic Map Merging in Rescue Environments. pages 483–490. Springer Berlin Heidelberg, Berlin, Heidelberg, 2005.

[13] Stefano Carpin, Andreas Birk, and Viktoria Jucikas. On map merging. Robotics and Autonomous Systems, 53(1):1 – 14, 2005.

[14] Göksel Dedeoglu and Gaurav S. Sukhatme. Landmark-based Matching Algorithm for Cooperative Mapping by Autonomous Robots, pages 251–260. Springer Japan, Tokyo, 2000.

[15] G. Erine, B. Balaguer, and S. Carpin. Heterogeneous map merging using wifi signals. In 2013 IEEE/RSJ International Conference on Intelligent Robots and Systems, pages 5258–5264, Nov 2013.

[16] G. D. Evangelidis and E. Z. Psaraks. Parametric image alignment using enhanced correlation coefficient maximization. IEEE Transactions on Pattern Analysis and Machine Intelligence, 30(10):1858–1865, Oct 2008.

[17] D. Fox, J. Ko, K. Konolige, B. Limketkai, D. Schulz, and B. Stewart. Distributed multirobot exploration and mapping. Proceedings of the IEEE, 94(7):1325–1339, July 2006.

[18] Christina Georgiou, Sean Anderson, and Tony Dodd. Constructing informative bayesian map priors: A multi-objective optimisation approach applied to indoor occupancy grid mapping. The International Journal of Robotics Research, 36(3):274–291, 2017.

[19] Steven Gold, Anand Rangarajan, Chien-Ping Lu, Suguna Pappu, and Eric Mjolsness. New algorithms for 2d and 3d point matching: pose estimation and correspondence. Pattern Recognition, 38(8):1019 – 1031, 1998.

[20] Aric A. Hagberg, Daniel A. Schult, and Pieter J. Swart. Exploring network structure, dynamics, and function using NetworkX. In Proceedings of the 7th Python in Science Conference (SciPy2008), pages 11–15, Pasadena, CA USA, August 2008.

[21] A. Howard. Multi-robot mapping using manifold representations. In Robotics and Automation, 2004. Proceedings. ICRA ’04. 2004 IEEE International Conference on, volume 4, pages 4198–4203 Vol.4, April 2004.

[22] Andrew Howard, Lynne E. Parker, and Gaurav S. Sukhatme. Experiments with a large heterogeneous mobile robot team: Exploration, mapping, deployment and detection. The International Journal of Robotics Research, 25(5-6):431–447, 2006.

[23] Ming-Kuei Hu. Visual pattern recognition by moment invariants. IRE Transactions on Information Theory, 8(2):179–187, February 1962.

[24] Wesley H. Huang and Kristopher R. Beavers. Topological map merging. The International Journal of Robotics Research, 24(8):601–613, 2005.

[25] J. D. Hunter. Matplotlib: A 2d graphics environment. Computing in Science Engineering, 9(3):90–95, May 2007.

[26] Eric Jones, Travis Oliphant, Pearu Peterson, et al. SciPy: Open source scientific tools for Python. 2001–. [Online; accessed May 2017].

[27] D. Kakuma, S. Tsuichihara, G. A. G. Ricardez, J. Takamatsu, and T. Ogawasara. Alignment of occupancy grid and floor maps using graph matching. In 2017 IEEE 11th International Conference on Information and Automation Computing (ICIC), pages 57–60, Jan 2017.

[28] Siavash Khallaghi. Pure numpy implementation of the coherency enhanced correlation coefficient maximization. IEEE Transactions on Intelligent Robots and Systems, 2008.

[29] K. Konolige, D. Fox, B. Limketkai, J. Ko, and B. Stewart. Map merging for distributed robot navigation. In Proceedings 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2003) (Cat. No.03CH37453), volume 1, pages 212–217 vol.1, Oct 2003.

[30] D. G. Lowe. Object recognition from local scale-invariant features. In Proceedings of the Seventh IEEE International Conference on Computer Vision, volume 2, pages 1150–1157 vol.2, 1999.

[31] Bruce D. Lucas and Takeo Kanade. An iterative image registration technique with an application to stereo vision. In Proceedings of the 7th International Joint Conference on Artificial Intelligence - Volume 2, IJCAI’81, pages 674–679, San Francisco, CA, USA, 1981. Morgan Kaufmann Publishers Inc.

[32] Jens Lundström, Wagner O. De Morais, Maria Menezes, Celeste Nugent, Gabriela Joao Bentas, Anita Sant’Anna, Jonathan Symnott, and Chris Nugent. Halmstad Intelligent Home - Capabilities and Opportunities, pages 9–15. Springer International Publishing, Cham, 2016.

[33] Aaron Meurer, Christopher P. Smith, Mateusz Paprocki, Ondej Čertk, Sergey B. Kirpichev, Matthew Rocklin, AMiT Kumar, Sergiu Ivanov, Jason K. Moore, Sartaj Singh, Thilina Rathnayake, Sean Vig, Brian E. Granger, Richard P. Muller, Francesco Bonazzi, Harsh Gupta, Shivan Vats, Fredrik Johansson, Fabian Pedregosa, Matthew J. Curry, Andy R. Terrel, tpn Rouka, Aishithosh Saboo, Isuru Fernando, Sunil Kulal, Terrel, tpn Rouka, Ashutosh Saboo, Isuru Fernando, Sumith Kulal, Terrel, tpn Rouka, Ashutosh Saboo, Isuru Fernando, Sumith Kulal,
Robert Cimrman, and Anthony Scopatz. SymPy: symbolic computing in python. PeerJ Computer Science, 3:e103, January 2017.

[34] M. Mielle, M. Magnusson, and A. J. Lilienthal. Using sketch-maps for robot navigation: Interpretation and matching. In 2016 IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR), pages 252–257, Oct 2016.

[35] Malcolm Mielle, Martin Magnusson, Henrik Andreasson, and Achim J. Lilienthal. SLAM auto-complete using an emergency map. CoRR, abs/1702.05087, 2017.

[36] Marius Muja and David G. Lowe. Fast approximate nearest neighbors with automatic algorithm configuration. In In VISAPP International Conference on Computer Vision Theory and Applications, pages 331–340, 2009.

[37] Alan Murta. GPC: General polygon clipper library. http://www.cs.man.ac.uk/toby/alan/software/, 1997–.

[38] A. Myronenko and X. Song. Point set registration: Coherent point drift. IEEE Transactions on Pattern Analysis and Machine Intelligence, 32(12):2262–2275, Dec 2010.

[39] Andriy Myronenko, Xubo Song, and Miguel Á. Carreira-Perpiñán. Non-rigid point set registration: Coherent point drift. In P. B. Schölkopf, J. C. Platt, and T. Hoffman, editors, Advances in Neural Information Processing Systems 19, pages 1009–1016. MIT Press, 2007.

[40] Michel Neuhaus and Horst Bunke. An Error-Tolerant Approximate Matching Algorithm for Attributed Planar Graphs and Its Application to Fingerprint Classification, pages 180–189. Springer Berlin Heidelberg, Berlin, Heidelberg, 2004.

[41] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. Journal of Machine Learning Research, 12:2825–2830, 2011.

[42] Andy Port. Svgpathtools; a collection of tools for manipulating and analyzing svg path objects and bezier curves. https://github.com/mathandy/svgpathtools, 2017.

[43] Jörg Rädler. Polygon: Python bindings to the ‘general polygon clipping library’ (gpc) and additional functions. https://www.j-raedler.de/projects/polygon/, 2015.

[44] J. Radon. On the determination of functions from their integral values along certain manifolds. IEEE Transactions on Medical Imaging, 5(4):170–176, Dec 1986.

[45] S. Saeedi, L. Paull, M. Trentini, M. Seto, and H. Li. Efficient map merging using a probabilistic generalized voronoi diagram. In In 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems, pages 4419–4424, Oct 2012.

[46] S. Saeedi, L. Paull, M. Trentini, M. Seto, and H. Li. Group mapping: A topological approach to map merging for multiple robots. IEEE Robotics Automation Magazine, 21(2):61–72, June 2014.

[47] Sajad Saeedi, Liam Paull, Michael Trentini, Mae Seto, and Howard Li. Map merging for multiple robots using hough peak matching. Robotics and Autonomous Systems, 62(10):1408 – 1424, 2014.

[48] Christopher A. Sanchez and Russell J. Branaghan. The interaction of map resolution and spatial abilities on map learning. International Journal of Human-Computer Studies, 67(5):475 – 481, 2009.

[49] A. Sanfelio and K. S. Fu. A distance measure between attributed relational graphs for pattern recognition. IEEE Transactions on Systems, Man, and Cybernetics, SMC-13(3):353–362, May 1983.

[50] S. Schwertfeger and A. Birk. Evaluation of map quality by matching and scoring high-level, topological map structures. In In 2013 IEEE International Conference on Robotics and Automation, pages 2221–2226, May 2013.

[51] S. Umeyama. Least-squares estimation of transformation parameters between two point patterns. IEEE Transactions on Pattern Analysis and Machine Intelligence, 13(4):376–380, Apr 1991.

[52] S. van der Walt, S. C. Colbert, and G. Varoquaux. The numpy array: a structure for efficient numerical computation. Computing in Science Engineering, 13(2):22–30, March 2011.

[53] Stéfan van der Walt, Johannes L. Schönberger, Juan Nunez-Iglesias, François Boulogne, Joshua D. Warner, Neil Yager, Emmanuelle Gouillart, Tony Yu, and the scikit-image contributors. scikit-image: image processing in Python. PeerJ, 2:e453, 6 2014.

[54] Olga Vysotska and Cyrill Stachniss. Improving slam by exploiting building information from publicly available maps and localization priors. PFG – Journal of Photogrammetry, Remote Sensing and Geoinformation Science, 85(1):53–65, 2017.

[55] Jan Oliver Wallgrün. Voronoi graph matching for robot localization and mapping. In Marina L. Gavrila, C. J. Kenneth Tan, and François Anton, editors, Transactions on Computational Science IX: Special Issue on Voronoi Diagrams in Science and Engineering, pages 76–108. Springer Berlin Heidelberg, Berlin, Heidelberg, 2010.

[56] S. B. Williams, G. Dissanayake, and H.Durant-Whyte. Towards multi-vehicle simultaneous localisation and mapping. In Proceedings 2002 IEEE International Conference on Robotics and Automation Cat. No.02CH37292, volume 3, pages 2743–2748, 2002.

[57] L. Yujian and L. Bo. A normalized levenshtein distance metric. IEEE Transactions on Pattern Analysis and Machine Intelligence, 29(6):1091–1095, June 2007.

[58] X. S. Zhou and S. I. Roumeliotis. Multi-robot slam with unknown initial correspondence: The robot rendezvous case. In 2006 IEEE/RSJ International Conference on Intelligent Robots and Systems, pages 1785–1792, Oct 2006.
Fig. 7: HH_E5 (office building in Halmstad, Sweden)

Fig. 8: HH_F5 (office building in Halmstad, Sweden)
Fig. 9: HH_HIH (Halmstad Intelligent Home [32])

Fig. 10: KPT4A (apartment in Halmstad, Sweden)