



2D map alignment with region decomposition

Saeed Gholami Shahbandi¹ · Martin Magnusson²

Received: 21 June 2017 / Accepted: 10 July 2018
© The Author(s) 2018

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 benefit from all available information in both maps, the robot must find the correct alignment between the two maps. There exist many approaches to address this challenge, however, most of the previous 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 propose a decomposition-based method for 2D spatial map alignment which does not rely on those assumptions. Our proposed method is validated and compared with other approaches, including generic data association approaches and map alignment algorithms. Real world examples of four different environments with thirty six sensor maps and four layout maps are used for this analysis. The maps, along with an implementation of the method, are made publicly available online.

Keywords Mobile robots · Mapping · Map alignment · Decomposition · 2D · Sensor map · Robot map · Layout map · Emergency map · Region segmentation · Similarity transformation

1 Introduction

There are many applications in which it is beneficial for a robot to merge its map with any of a number of existing maps. For example, in environment surveying, a blueprint layout map could be introduced to give the robot a head start in terms of exploration. Such a prior map could also improve global consistency of SLAM algorithms by exploiting the global consistency of the prior map. Another example is to integrate semantic information or traffic flow data into a central map 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 using 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 human intervention.

A 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 this example, an important challenge is the difference in map formats (Fig. 1). In such cases the prior map of the environment is a blueprint, and the robot maps are often discrete such as occupancy grid maps. *This paper addresses 2D map alignment where the maps share no frame of reference, overlap only partially, have different amounts of clutter, and have different modalities.*

✉ Saeed Gholami Shahbandi
saesha@hh.se

Martin Magnusson
martin.magnusson@oru.se

¹ Center for Applied Intelligent Systems Research, Intelligent Systems Lab, Halmstad University, Halmstad, Sweden

² Mobile Robotics and Olfaction Lab, Centre for Applied Autonomous Sensor Systems (AASS), Örebro University, Örebro, Sweden

1.1 Problem description

We define the map alignment as a data association problem across two representations of the same environment (two maps). The solution to this problem is a transformation function between the coordinate frames of the input maps. The objective is then to find the optimal transformation under

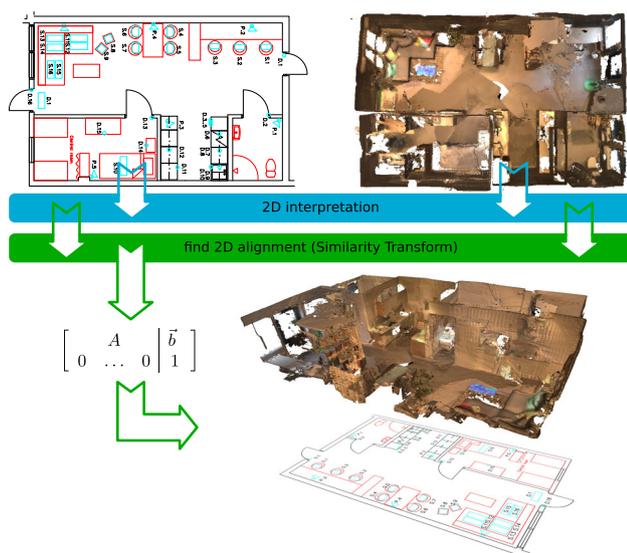


Fig. 1 An example of a pair of layout and sensor maps. Sensor maps are acquired with a Google Tango tablet as 3D meshes, and later are converted to 2D occupancy-like maps. This example is from the Halmstad Intelligent Home (Lundström et al. 2016). Other maps also include office environments (see “Appendix”)

which the distances between corresponding elements in the two representations are minimized. The challenges of map alignment we intend to address in this work are:

- Multi-modality and representation discrepancy between the maps (e.g. blueprint versus sensor map),
- Scale mismatch between input maps (e.g. due to different modalities),
- Repeating patterns and the associated problem of local minima in the alignment objective function (auto-isomorphic graphs in a topological sense, and shape correspondence problem in a geometric sense).

By the following set of assumptions we contain the problem description to a more specific application domain:

- The environments are:
 - Well-structured, that is to say their maps could be modeled (abstracted) with a set of 2D geometric curves,
 - Composed of meaningful regions that can be segmented (e.g. room, corridor, etc.,).
- The maps are:
 - Spatial, geometric and 2-dimensional (or could be represented in a 2D plane),
 - Globally consistent (not “broken”) and uniformly scaled.

The restrictions and limitations caused by these assumptions are further discussed in Sect. 5.

1.2 Approaches

In this general formulation, the map alignment problem proves to be more challenging than the relaxed versions such as scan matching or image alignment problems. When the displacement between two frames (maps or scans) are small, optimization based algorithms such as point set registration (Besl and McKay 1992; Gold et al. 1998; Tsing and Kanade 2004; Myronenko et al. 2007; Myronenko and Song 2010) or image registration (Baker and Matthews 2004) are suitable solutions. However, these algorithms are vulnerable to local minima, especially in the absence of an initial guess. This pitfall is exacerbated when the input maps contain repetitive patterns that increase the number of local minima.

Another approach is to employ the Hough transform to structure the search space and decompose the transformation into separate operations of rotation and translation (Carpin 2008; Saedi et al. 2012). However, these approaches are limited to rigid transformations (i.e. Euclidean transformation that includes translation, rotation and no scaling) and expect homogeneity of the input signal (same modality).

A more common approach to the map 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 (Huang and Beevers 2005; Schwertfeger and Birk 2013; Kakuma et al. 2017). Consequently, geometric and/or topological similarities of the vertices and/or edges are used to find a match between two maps. When maps are of different types, such an interpretation plays an important role. This interpretation abstracts the input maps into a *shared instantiated representation*, which makes the search for similarities between maps feasible.

A more thorough review of related work is presented in Sect. 2, after which detailing our method and the experimental validation of it are presented in Sects. 3 and 4.

1.3 Our approach

In this work, we propose a method based on the aforementioned concept of *shared instantiated representation*. The underlying representation of our method is a geometric decomposition that outlines the segmentation of regions, namely “arrangement” (Agarwal and Sharir 2000) in 2D. 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-spaces. In addition, it implicitly captures the connectivity of the open-spaces through the regions’ adjacency. Our proposal is to (i) abstract input maps via region decomposition into a shared geometric representation, i.e. 2D arrangement, (ii) search for all potential

alignments that match the regions of the maps, (iii) select the best alignment according to a proposed quality measure. The details of the method, the 2D arrangement, and further details are provided in Sect. 3. The differentiating characteristic of our method is its representation and the consequent 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 missing the correct answer. This is crucial in the case of noisy maps and maps of different modalities. To our knowledge, no algorithm has been developed for solving the alignment problem in this manner. The main contributions of this work are as follows:

- An algorithm for map alignment that does not rely on rigid transformations, available initial guess or similar representation between maps.
- An abstract representation (2D arrangement), which is capable of interpreting maps of different modalities. We use this interpretation for region segmentation, and for solving the alignment problem. This interpretation results in a hierarchical representation of maps, where the abstract models on the top level are readily available for other geometric processing and manipulations after alignment.
- A publicly available collection of maps, containing forty maps of four different environments. The source code to our implementation of the proposed method is also made available online. For the links to these online repositories please see Sect. 4.

2 Related work

The main underlying problem in map alignment 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 their 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, i.e. map alignment.

Motivations of sensor to prior map alignment There are different motivations for fusing prior maps and sensor maps.

For instance, Sanchez and Branaghan (2009) argue that abstract maps are easier to learn, and accordingly, Georgiou et al. (2017) state that a correspondence between an abstract human readable map and robot's sensor map is desired to facilitate collaborative tasks between humans and robots. Bowen-Biggs et al. (2016) claim that sensor maps are not "natural" for many high level tasks, especially those including semantics or with human in the loop. In their work, they present a method of fusing 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, either through exploiting the structure of the prior map, or by aligning local maps to build a global map. Georgiou et al. (2017) formulated the "structural information from architectural drawings" as "informative Bayesian mapping priors" in order to improve the performance of the SLAM algorithm. 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 (2017) 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. (2017a) proposed 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 enhance the quality of the rough prior map by fusing it with sensor map.

2.1 Graph matching approaches

Topological structure of the open spaces 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 (Huang and Beevers 2005).

Huang and Beevers (2005) 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, followed by a second stage in which the geometric consistencies of the match hypotheses are evaluated. 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 (2010) 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 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 radii of the maximal inscribed circles 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 (2013) have developed an interesting method for map alignment. Their method captures the high-level spatial structures of the maps through Voronoi graphs, and represents with topological graphs that contain the angles between edges and the length of edges. The map alignment is done by finding similar vertices of the graphs and "identification of sub-graph isomorphisms through wave-front propagation" (Schwertfeger and Birk 2013). With experimental results, they show the robustness of their method by detecting brokenness in sensor maps.

In another intriguing work, Mielle et al. (2016) proposed a map alignment method based on graph matching, which enables robots to follow navigation orders specified in sketch maps. Their method converts the Voronoi skeleton 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 2004) graph matching algorithm based on the normalized Levenshtein edit-distance (LED) (Yujian and Bo 2007). By skipping the absolute position values, the interpretation becomes insensitive to noise and inconsistency of the map. Consequently their method doesn't require global consistency and uniform scaling of the maps.

In order to benefit from semantic information available in floor maps for high level task execution, Kakuma et al. (2017) proposed a graph matching based method for the alignment of sensor maps to floor plans of the buildings. Their method constructs a graph from segmented regions of the occupancy map. Graph matching is carried out with minimizing a matching cost function based on a variation of graph edit distance (GED) (Sanfeliu and Fu 1983; Hu-Moments Hu 1962).

2.2 Hough/Radon transform approaches

Hough(/Radon) transform maps the input signal from the Cartesian to a *parametric* space. This parametric space has the advantage of capturing the salient, thought maybe latent, structure of the 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, and work best on maps with same modalities.

For merging partial maps in a multi-robots application, Carpin (2008) proposed a method that first finds the rotation alignment via a correlation between the Hough spectra of the two maps. After the orientation alignment, the translation parameters are estimated from a x - y projection of the maps. 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 (2008) tackle the problem of global mapping by merging local maps. Their method also decouples the rotation and translation estimations, but with some twists in their transformation techniques. They use an "orientation histogram of the scan normals" (yields an output similar to a Hough transform) to determine the orientation alignment. Then a "weighted projection histograms created from the orthogonal projections" (somewhat equivalent to radiography) is used for estimating the translation between the orientationally aligned data.

Saeedi et al. (2012, 2014a, b) 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.

2.3 Optimization approaches

One of the most popular categories of techniques for data associations in robot mapping is based on optimization. A famous example is the iterative closest point (ICP; Besl and McKay 1992) 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 (1981), Baker and Matthews (2004) and enhanced correlation coefficient (ECC) Maximization (Evangelidis and Psarakis 2008), 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 (2005), Birk and Carpin (2006), and Carpin et al. (2005). Their approach minimizes a dissimilarity function (overlapping quality index) over the transformation parameters, with a stochastic process (random walk), used for the optimization. An interesting feature of this method is its ability to robustly handling unstructured environments.

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 (2001), Dedeoglu and Sukhatme (2000), and Williams et al. (2002) are good examples in this category. These methods are based on the assumption that the input maps are from the same modality. With a similar application, i.e. multi-robot exploration, some researchers have developed methods to determine the relative transformation between robots’ partial maps when the robots can physically meet in the environment. Examples of the methods based on “rendezvous” or “mutual observation” are proposed by Howard et al. (2006), Howard (2004), Fox et al. (2006), Zhou and Roumeliotis (2006), and Konolige et al. (2003). These methods are based on the robots’ ability to meet and generate transformation hypotheses from a rendezvous, which is unfeasible for off-line methods. Erinc et al. (2013) proposed a method to annotate heterogeneous maps with WIFI signal that provides cues for data association between maps. This means two essentially different maps are annotated by a shared landmark, which provides a seamless data association cue. Boniardi et al. (2015) developed a method for localizing and navigating directly in a sketch map, without the map alignment. Partial map alignment is an essential component of map merging. Saeedi et al. (2016) provided a thorough review of the multi-robot SLAM field which covers a broad range of such methods. However, most of these methods, being specifically developed to improve multi-robot mapping applications, are dependent on sources of information that are specific to robot map and not accessible for any arbitrary map (e.g. layout maps). The work by Bonanni et al. (2017) on merging 3D maps, and its earlier

version on 2D maps targeting the problem of merging “partially consistent maps” (Bonanni et al. 2014), require the pose graph to be available (or computable) for both maps. Jiang et al. (2017) proposed a method based on “motion averaging” for merging multiple local maps. Their approach is to find transformation between all local map, construct a graph of the inter-map “motion” and optimize such motions for optimal alignment. However, the core of this method is the optimization of alignment between several local maps, and therefore not suitable for aligning a pair of maps.

3 Method

The essence of our method, as depicted in Fig. 2, is to abstract the representation of input maps in order to facilitate the search for alignment. 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 map into separate regions with a 2D arrangement. We have shown in our earlier works (Gholami Shahbandi et al. 2014, 2015), how this representation could be used for semantic annotation and place categorization of occupancy grid maps. Section 3.1 describes the 2D

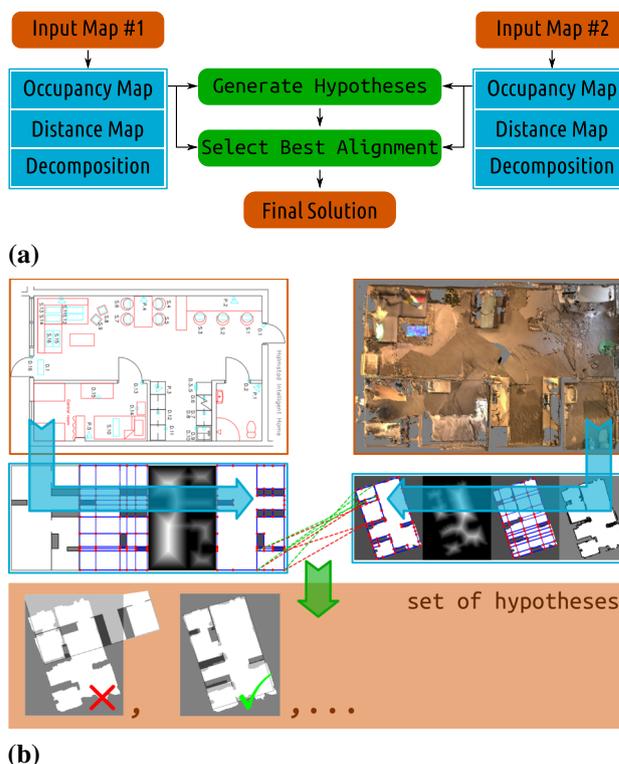


Fig. 2 The outline of our method in **a**, and a concrete example demonstrating the process on real maps in **b**. **a** The orange blocks represent inputs and output, blue blocks are the intermediate representations, and the green blocks are the alignment processes. **b** An example of the process on a pair real world maps (Color figure online)

Algorithm 1 Map Interpretation

```

function INTERPRET(Map)
   $\mathcal{T} = \text{DETECTGEOMETRICTRAITS}(\text{Map})$ 
   $A(\mathcal{T}, \mathcal{P}, \mathcal{F}) = \text{ARRANGEMENT}(\mathcal{T})$ 
  /* Pruning */
  //  $M_d$ : normalized Distance Transform of Map
  //  $N_p(e)$ : neighboring cells (pixels) to an edge  $e$ 
  //  $V(e) = \frac{1}{\text{size}(N_p(e))} \sum_{p \in N_p(e)} M_d(p)$ 
  //  $\text{thr}_e$ : wall/gateway detection threshold ( $\sim 0.075$ )
   $E(\mathcal{P}) = E(\mathcal{T}) - \{e \mid e \in E(\mathcal{T}) \wedge V(e) < \text{thr}_e\}$ 
  UPDATE( $A(\mathcal{T}, \mathcal{P}, \mathcal{F})$ )
return  $A$ 
end function

```

```

function ARRANGEMENT( $\mathcal{T} = \{t_i\}$ )
  /* construct prime graph  $\mathcal{P}$  from traits  $\mathcal{T}$  */
   $V(\mathcal{P}) : \{v_i(x, y) \mid (\exists t_j, t_k \in \mathcal{T})[v_i \in t_j \wedge v_i \in t_k]\}$ 
   $E(\mathcal{P}) : \{e_i(t_k, v_s^i, v_e^i) \mid (\nexists v_m \in e_i)[v_m \neq v_s^i \wedge v_m \neq v_e^i]\}$ 
  /* identify irreducible faces  $\mathcal{F}$  in  $\mathcal{P}$  */
   $\mathcal{F} : \{f_i := \{e_j\} \mid (\forall e_j \in f_i)[\exists! e_k \in f_i \mid v_s^j = v_e^k]\}$ 
   $N(\mathcal{F}) : \{(f_i, f_j) \mid (\exists e_k)[e_k \in f_i \wedge e_k \in f_j]\}$ 
return  $A(\mathcal{T}, \mathcal{P}, \mathcal{F})$ 
end function

```

arrangement representation. Furthermore, we explain how this representation is adjusted to capture *meaningful* regions, i.e. adjusting the structural decomposition to region segmentation. Section 3.2 describes the alignment procedure, that is the matching of regions in the maps and estimation of alignment transformation for each match, resulting 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. We introduce a “match score” in Sect. 3.3, that is used for comparing the quality of alignments, which in turn is used to pick the best hypothesis.

3.1 Map interpretation

First step towards map alignment is the modeling of maps with an abstract representation, i.e. arrangement (Agarwal and Sharir 2000). Algorithm 1 outlines the process of this abstraction, composed of geometric trait detection, decomposition (arrangement), and pruning of the arrangement from a structural decomposition to a region segmentation. An arrangement partitions a 2D plane according to a set of geometric objects (such as, but not limited to, lines and circles), referred to as geometric traits and traits for short. A set of geometric traits \mathcal{T} will result in a unique arrangement A identified by a prime graph \mathcal{P} , and a set of faces \mathcal{F} . The prime graph \mathcal{P} is the result of intersecting all traits \mathcal{T} , and faces are *irreducible* closed-regions (“Jordan Curve”) bounded to edges from the prime graph $V(\mathcal{P})$. Neighborhood $N(\mathcal{F})$ is an attribute associated with the set of faces \mathcal{F} , defined as a set of tuples of faces where each tuple identifies a pair of neigh-

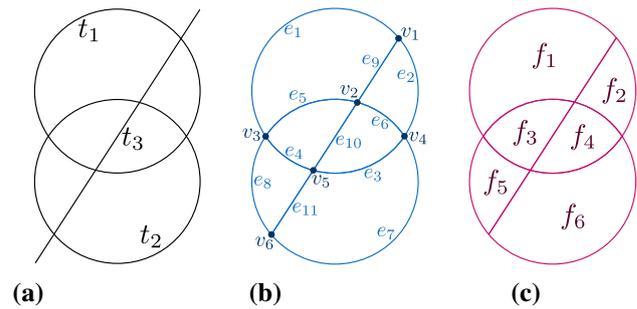


Fig. 3 An arrangement $\mathcal{A} := (\mathcal{T}, \mathcal{P}, \mathcal{F})$. This example involves straight lines and circles to demonstrate the ability to handle geometric traits beyond straight lines. However, due to the nature of buildings in our maps, our map interpretation with arrangement only relies on straight lines. **a** Geometric traits, **b** prime graph, **c** faces

boring faces. Figure 3 demonstrates an arrangement and its components on a toy example. For technical details of the arrangement algorithm, see Agarwal and Sharir (2000).

3.1.1 Geometric traits

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. Accordingly, an arrangement manifests a *dual* characteristic: (i) \mathcal{F} is a geometric representation of the open-space and its boundaries, and (ii) $N(\mathcal{F})$ captures the topology of open-space. The detection of the geometric traits from 2D maps could be achieved by common algorithms such as *Generalized Hough Transform* (Ballard 1981) and *Radon Transform* (Radon 1986). Given that all maps used in our experiments could be modeled with only straight lines, in this work we use *radiography*, which is a variation of the aforementioned algorithms. Radiography operates as a Radon transform that is filtered by the oriented gradient of the image. That is, the projection of each point is weighted by the magnitude of the image’s gradient at that point, multiplied by the difference between the orientation of the image’s gradient and the direction of the Radon projection. We have shown previously (Gholami Shahbandi et al. 2014) that radiography is more robust in modeling physical elements of the environment (e.g. walls) that suffer from a discrepancy in their continuity or too much noise. Nonetheless, the arrangement representation is neither limited to straight lines, nor dependent on the trait detection technique.

3.1.2 Abstraction compatibility and arrangement pruning

Despite its merits in detecting discontinuous traits in noisy maps and capturing the global structure of environments (Gho-

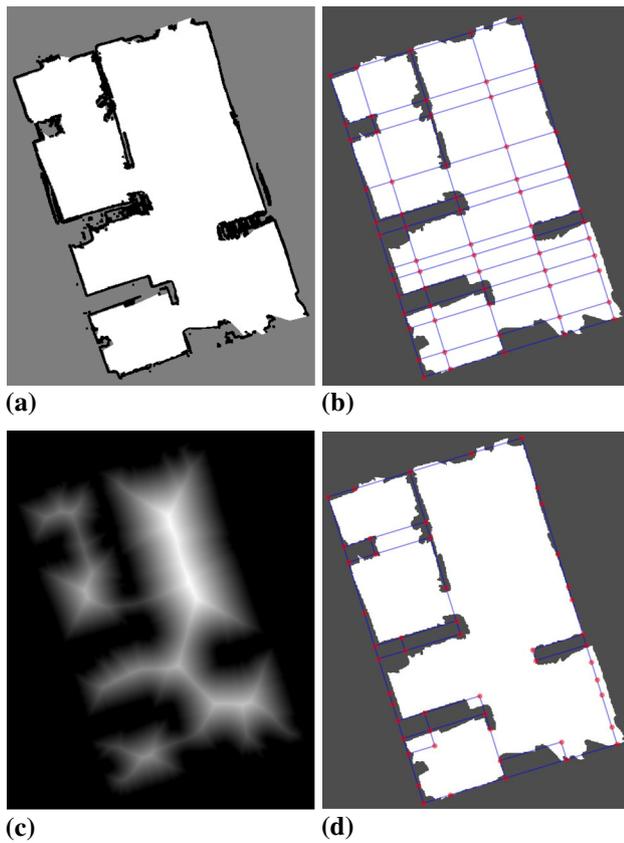


Fig. 4 An Occupancy Map in **a**, its original decomposition in **b**, and a cleaned-up version of the arrangement (pruned) in **c**, based on the distance map M_d in **d**. **a** Occupancy map, **b** arrangement, **c** distance image, **d** after pruning

lami Shahbandi et al. 2014), radiography detects unbounded traits (e.g. infinite lines instead of line segments). Consequently the partitioning of the space is not equivalent to a plausible region segmentation, due to over-decomposition of areas that are conceptually a single region (e.g. a kitchen or an office). Figure 4 demonstrates the over-decomposition of a real map. This inconsistency of region segmentation is non-deterministic, depends on the noise, partiality and inconsistencies of the maps, and could vary from sensor maps to layout maps. Since the essence of our alignment method is to match corresponding regions, it is crucial for the maps to have representations on the same level of abstraction (regions segmentation), what we call *abstraction compatibility*. That is to say, if a single face represents a room in one map, the same room must be represented by a similar face in the other map. Based on empirical observations, the success rate of our method seems to be most sensitive to this compatibility assumption. However, the sensitivity of the alignment method to abstraction compatibility is not critically obstructive, and not every corresponding region should have compatible abstraction in the maps.

We remedy this challenge by *pruning* the arrangement to a more plausible region segmentation, presented in the function INTERPRET in Algorithm 1. This pruning is the process of removing all face boundaries (i.e. $e_i \in E(\mathcal{P})$) which do not correspond to a *wall* or *gateway*, followed by merging all adjacent faces whose shared boundaries are removed. As described in Algorithm 1, an edge is considered a gateway or wall if the average value of the pixels $V(e)$ from the distance map M_d (Fig. 4c) within a neighborhood of the that edge $N_p(e)$ is below a certain threshold thr_e . Since the distance map M_d is normalized (scaled to $[0, 1]$), the threshold thr_e over the averaged pixel values $V(e)$ is independent of the map scale. The result of a pruning process applied to the over-decomposed example of Fig. 4b, is presented in Fig. 4d.

Our pruning of an over-decomposed arrangement to region segmentation is a variation of “Morphological Region Segmentation” as presented in the “Room Segmentation” survey by Bormann et al. (2016). In more elaborate scenarios, one could employ other region segmentation methods presented in Bormann et al. (2016), or more recent works by Fermin-Leon et al. (2017) and Mielle et al. (2017b). Nevertheless, we observed that the arrangement pruning, along with an approximation of faces with *Oriented Minimum Bounding Boxes* (details in Sect. 3.2), satisfies the abstraction compatibility assumption.

3.2 Alignment procedure

A hypothesis in the context of this work is a transformation function between the coordinate frames of the two maps. Hypothesis generation is the process of proposing such plausible transformations. According to the *uniform scaling assumption* stated in Sect. 1.3, the transformations estimation is restricted to “similarity” (i.e. 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. The *shape descriptor* is an ordered sequence of vertex-edge tuples

$$D(f) := ((v_i, e_l), \dots, (v_j, e_k))$$

in which, all the entries (vertices and edges) are ordered counter clock-wise from an arbitrary reference point. Since the choice of reference point in each face is arbitrary, the descriptor also contains all its “cyclic shifts”. For example, the descriptor for face f_1 of the arrangement from the Fig. 3 with all its s -step shifts are

$$D(f_1^{s=1}) = ((v_2, e_9), (v_1, e_1), (v_3, e_5))$$

$$D(f_1^{s=2}) = ((v_3, e_5), (v_2, e_9), (v_1, e_1))$$

$$D(f_1^{s=3}) = ((v_1, e_1), (v_3, e_5), (v_2, e_9))$$

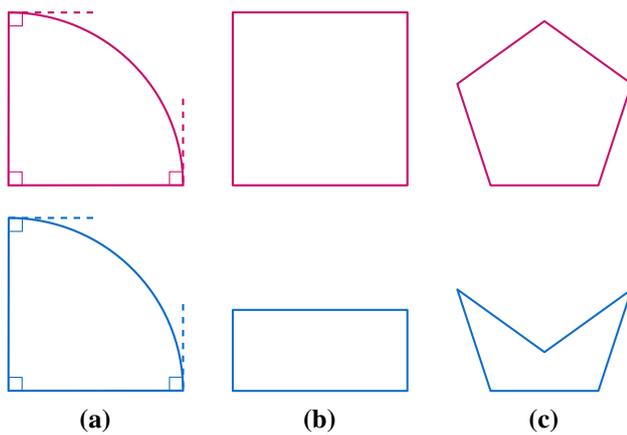


Fig. 5 Examples where a missing feature results in wrong matches. In the absence of *edge length ratio* those faces in **a** could have three matches instead of one, and the faces in **b** could have four matches instead of zero. In the absence of *corner angle* those faces in **c** could have five matches instead of none

In the context of the shape descriptor, vertices denote corners, where corners are defined as vertices with any internal angles other than π . The features of the descriptor are the internal angles of vertices (i.e. *corner angle*), and the length of edges normalized to the perimeter of the face (i.e. *edge length ratio*). Figure 5 demonstrates the necessity of these features, through examples where the absence of these two features would result in false matches. Descriptor size equivalency is the first necessary condition for a potential match. A match is then identified as a s -step “circular shift” of one descriptor, so that all corresponding entries in the descriptors of the faces are equivalent. After a correspondence between the two point sets (face corners) is proposed via face matching, a transformation between the two point sets is estimated based on the “Least-squares estimation” method proposed by Umeyama (1991), which uses the singular value decomposition of a covariance matrix of the data points.

Algorithm 2 presents the map alignment procedure in pseudo-code, and Fig. 2b depicts two examples of correct (in green) and wrong (in red) association and their consequent transformation.

Simplified Alignment Procedure 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 the 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 Fig. 4d. Alternatively, due to such cases, we simplify shapes with their *Oriented Minimum Bounding Boxes* (OMBB). This counts as an interpretation of the “well-structured environments” assumption stated in Sect. 1.3. This substitution of the shapes with OMBB renders the descriptor size and corner angles

Algorithm 2 Map Alignment Procedure

Input: Map₁, Map₂
Output: Alignment (similarity transformation)

```

/* Map Interpretation */
A1(T1, P1, F1) = INTERPRET(Map1)
A2(T2, P2, F2) = INTERPRET(Map2)

/* Face Matching & Hypotheses Generation */
// f.c : corners of the face f
// D(f) : shape descriptor of face f
// fs := s-step circular shift of corners and D(f)
H = ∅
for fi ∈ F1, fj ∈ F2 | SIZE(fi.c) = SIZE(fj.c) do
  for s := 1 to SIZE(fi.c), step=1 do
    if D(fis) = D(fj) then
      H = H ∪ {ESTIMATETRANSFORM(fis.c, fj.c)}
    end if
  end for
end for

/* Selecting The Best Hypotheses */
Alignment = arg maxh ∈ H MATCHSCORE(h, A1, A2)

```

Algorithm 3 Map Alignment Procedure with simplified hypotheses generation

Input, Output (see Algorithm 2)

```

/* Map Interpretation (see Algorithm 2) */

/* Generate Hypotheses (without face matching) */
// f̂ = ORIENTEDMINIMUMBOUNDINGBOX(f)
H = ∅
for fi ∈ F1, fj ∈ F2 do
  for s := 1 to 4, step=1 do
    H = H ∪ {ESTIMATETRANSFORM(f̂is.c, f̂j.c)}
  end for
end for

/* Reject False Positives Hypotheses */
// h.sx, h.sy: scales of transformation in x/y directions
// thrs: acceptable ratio between scales (~ 1.2)
for h ∈ H | -(1/thrs < (h.sx/h.sy) < thrs) do
  reject h
end for

/* Selecting The Best Hypotheses (see Algorithm 2) */

```

redundant, and consequently, the only relevant shape feature would be the aspect-ratio of the OMBBs (i.e. edge length ratio). We have noticed that it is computationally cheaper to replace this feature (i.e. edge length ratio), with an equivalent process of *False Positive rejection* which is based on the uniform scaling assumption. Accordingly, the uniform scale assumption is relaxed, where all the transformations are estimated with an affine model, and then any transformation that does not qualify as a similarity transformation is rejected. Algorithm 3 is the modified version of the alignment procedure, which reflects the simplification of the shape descriptor and the rejection of non-uniformly scaled transfor-

mations. The result of the False Positive rejection can be seen in Table 2, where $\sim 90\%$ of initial hypotheses are rejected. We have carefully monitored the hypotheses pools of cases where the method failed to find the correct alignment. We can report that a correct alignment was never generated. In other words, we can safely assure that a rejection of potentially correct hypotheses has never been a cause of failure.

3.3 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 between two arrangements A_1 and A_2 , under the transformation 1T_2 , is defined as

$$S_A(A_1, A_2, {}^1T_2) = \sum_{\substack{f_i \in \mathcal{F}_1 \\ f_j \in \mathcal{F}_2}} \min(w(f_i), w(f_j)) \times s_f(f_i, f_j)$$

where $w(f)$ 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:

$$w(f_i) = \frac{\text{area}(f_i)}{\text{area}(A)}, \quad \text{area}(A) = \sum_{f_k \in \mathcal{F}} \text{area}(f_k)$$

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

$$s_f(f_i, f_j) = \begin{cases} \frac{e^{\left(\frac{f_i \cap f_j}{f_i \cup f_j}\right)} - 1}{e - 1} & \text{if } (f_i, f_j) \in \text{association} \\ 0 & \text{otherwise} \end{cases}$$

where $f_i \cap f_j$ is the surface area of the faces' intersection and $f_i \cup f_j$ is the surface area of the faces' union. The match score of a face with itself (perfect match) equals one, and the match score of two non-intersecting faces equals zero. The exponential expression rewards slight improvements close to perfect match more than the slight improvements close to a bad match.

The *association* represents pairs of faces from two arrangements that are associated (not just overlapping) under the transformation. We define *association* based on three conditions:

$$\begin{aligned} \text{association} &: \{(f_i, f_j) \mid \forall f_i \in \mathcal{F}_1, f_j \in \mathcal{F}_2, c_1 \wedge c_2 \wedge c_3\} \\ c_1 &: \text{center}(f_i) \in f_j \wedge \text{center}(f_j) \in f_i \\ c_2 &: \nexists f_k \in \mathcal{F}_2 \mid \text{center}(f_k) \in f_i, d(f_i, f_j) > d(f_i, f_k) \\ c_3 &: \nexists f_k \in \mathcal{F}_1 \mid \text{center}(f_k) \in f_j, d(f_i, f_j) > d(f_j, f_k) \end{aligned}$$

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 . First, for the two faces f_i and f_j to be associated, they must enclose each other's center. We define the center of a face as the "centroid" (geometric center) of the vertices of the face. Condition number two assures a one to one association where a face overlaps with multiple faces from the other arrangement. In such cases, among all faces of \mathcal{F}_2 with their centers enclosed by $f_i \in \mathcal{F}_1$, the face ($f_j \in \mathcal{F}_2$) with most similar size (surface area) is associated with f_i . And the third condition is symmetric to the second condition, i.e. vice versa for f_i to f_j .

This match score is devised only for the comparison of different hypotheses for a single pair of maps. That is to say, the alignment of different sensory maps over a layout map could not be compared with this score, nor is it suitable to detect the layout to which a sensor map belongs (i.e. layout recognition), and neither is it suitable as a quantified match accuracy measure. This matter is better observed in Figs. 8 and 9 from Sect. 4, where it is discussed with experimental observations.

The challenge of face center The center points of the faces can be defined differently according to the context of the application, such as "center of gravity", "Chebyshev centers" and "polygon centroid". However, it proves to be very hard to lay down a definition that guarantees to be enclosed by the region and unique. If a face is non-convex as in Fig. 6a, there is no guarantee that the center of gravity would be enclosed by the face. Chebyshev centers are defined as the center of either (i) the minimal-radius circle enclosing a region, or (ii) the maximal-radius inscribed circle inside the region. The example of Fig. 6a shows that the minimal-radius enclosing circle is susceptible to the same problem as the center of gravity, and Fig. 6b presents an example where the maximal-radius inscribed circle is not necessarily unique. Another example of tackling this challenge that we have explored, is to extract the Generalized Voronoi Diagram, and picking a point on the skeleton of the influence zone (SKIZ) that has the minimum sum of distance to all other points. This definition is also prone to degenerate cases, such the example in Fig. 6c shows. A variety of definitions could be considered for the center of a region that guarantee to be enclosed by the region. Alas, they would ultimately depend on the interpretation of "center point" with respect to the domain of application, and most are prone to degenerate cases where such a point is not unique. Ultimately, we have observed that in the setting of our problem such degenerate cases are not so frequent to

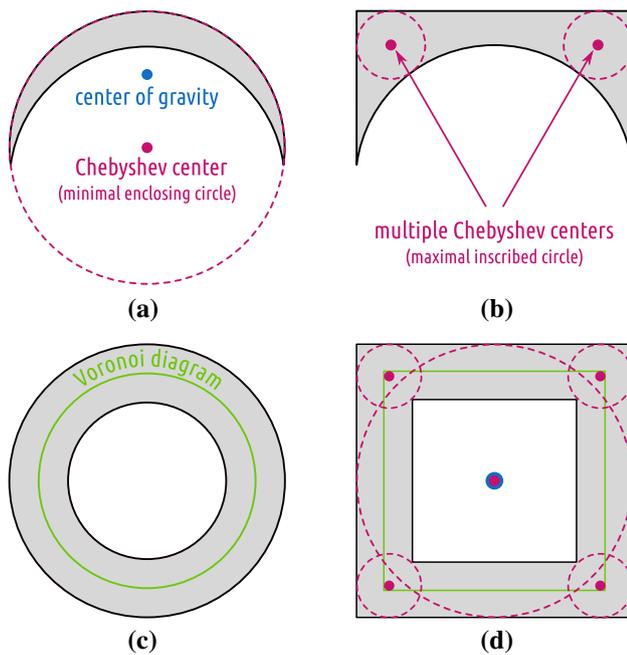


Fig. 6 Examples where “center of gravity”, “Chebyshev centers” and “Voronoi-based” definitions fail to identify a center point of a region that is both enclosed by the region *and* unique. **a**, **b** and **c** highlight the failures of each definition under different circumstances. While those three may not seem probable in a real world scenario, **d** presents the failure of such definitions in a more realistic example

disturb the performance of the method. Either of these definition would satisfy the requirements of our method as long as it guarantees uniqueness, and we chose the centroid of the vertices.

4 Experimental results and verification

In this section we present the results from a series of experiments, on a data-set of forty maps collected specifically for this task. The experiments are designed to show the method’s performance, under different circumstances and in comparison with other methods. All the experiments are based on an implementation of the method in Python, using many libraries (Jones et al. 2001; van der Walt et al. 2011, 2014; Hunter 2007; Meurer et al. 2017; Bradski 2000; Rädler 2015; Hagberg et al. 2008). The source code to our implementation is also available online.¹

4.1 Data collection

To evaluate our method, we collected maps of four different environments in two modalities, of CAD drawings and sensor

maps. All the maps are available online,² and presented in “Appendix”.

Modalities A series of sensor maps were collected by a *Google Tango tablet*, and the *Tango Constructor application* from Google. The 3D meshes were sliced horizontally and converted to an occupancy-like bitmap, where all the space is open except for the vertices of the mesh. From there, we generated a pseudo-occupancy map through an interactive ray-casting process. Detection of the geometric traits from foregoing maps were done via a variation of the Radon transform, namely radiography (Gholami Shahbandi et al. 2014).

As for the other modality, the layout maps were obtained from CAD drawings in portable document format (PDF). These CAD 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 Fig. 2b from Sect. 3. It should be mentioned that the simplified version of the layout map is not tailored to accurately reflect the real layout and what is captured by the sensor map, with no intention to benefit the alignment method. For instance, walls are represented with single lines in layout maps (width = ~ 1 –2 pixel), while they are much wider in the sensor maps (~ 5 –10 pixels). The drawings were converted to scalable vector graphics (SVG) and the geometric traits were obtained directly by parsing the SVG files (Port 2017). 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 also be automated. 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.

While all the sensor maps have the same scale, that is, they could be correctly aligned with each other under a rigid transformation, layout maps have different scales compared to the sensor maps, and a rigid transformation could not correctly align sensor maps to layout maps.

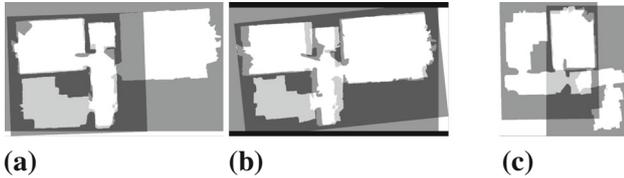
Environment types We collected data from four different environments, two of which are homes and the other two are office buildings. Table 1 lists the number of available maps for each environment, and all the maps can be found in “Appendix”. In total there are forty maps, four of which are

¹ <https://github.com/saeedghsh/Map-Alignment-2D/>.

² <https://github.com/saeedghsh/Halmstad-Robot-Maps/>.

Table 1 A list of all maps of four different environments

Environment	Type	# Sensor maps	# Layout maps
HH_E5	Office	14	1
HH_F5	Office	14	1
HH	Home	4	1
KPT	Home	4	1

**Fig. 7** Examples of correct, defected, and wrong alignments. Both correct and defected are considered as successful alignments. **a** Correct, **b** defected, **c** wrong

layout maps and the rest are sensor maps. Most sensor maps are partial and vary in their coverage of the environment.

Maps that violate our assumptions The map collection contains maps that violate some of the initial assumptions. For instance, maps HH_E5_2, HH_E5_3, HH_E5_4 and HH_F5_2 only cover corridors and halls and do not contain any room, and therefore there are not enough segment-able regions for hypotheses generation. Other examples include HH_E5_12 and HH_F5_1 which are bent (deformed) and violate the global consistency assumption. There exist further minor defects in some other maps. Consequently, the performance results presented here are not the representative of the method's performance under all the assumptions. Nevertheless, we include these maps to better observe the dependency of the method on the aforementioned assumptions, and provide a more inclusive performance result under different conditions.

Evaluations are based on success rate The performance of each method is provided as success rate, which is a percentage of successful alignments. We skip a *quantified accuracy measure* for the alignment. It proved very hard (impossible for our data) to provide a per map *alignment accuracy*, due to: (i) the lack of ground truth for the sensor maps, (ii) the inaccuracy of layout maps, and (iii) the presence of noise and global inconsistency of the sensor maps. Figure 7 illustrates our quality assessment of the alignments.

4.2 Experiments and results

The performance of the method is evaluated under three different experimental setups:

- *Sensor map to layout map alignment*, which is the main objective of the proposed method.
- *Sensor map to sensor map alignment*, where we observe how partial coverage, noise and inconsistency of sensor maps affect the performance.
- *Evaluation of alignment match score*, where the match score is studied for the alignments of intra and inter environment maps. Accordingly every sensor map is aligned to all other layout maps, whether from the same environment or not.

4.2.1 Sensor map to layout map alignment

Table 2 presents the performance of the method in aligning sensor maps to layout maps (within the same environment). The column *initial* represents the number of initial estimated transformations, *after rejection* represents the number of remaining hypotheses after rejecting non-uniformly scaled transformations ($\sim 90\%$ are rejected), and the last column marks the success of each alignment. In total, the method has successfully aligned all maps of the home environments, and yielded 83% in success rate for the office buildings. According to our investigations, failures are mainly due to the violation of the prior assumptions, such as global inconsistency and not enough segment-able regions in sensor maps.

4.2.2 Sensor map to sensor map alignment

Table 3 compares the success rate of the method in aligning sensor maps to sensor maps, versus aligning sensor maps to layout maps. It can be observed that the success rate of the method drops in aligning sensor maps to sensor maps. There are two main reasons for this drop; (i) many sensor maps are partial and consequently they overlap with each other marginally, (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 the noise and inconsistencies are amplified.

4.2.3 Evaluation of the alignment match score

Figure 8 represents the match score of the *winning hypotheses* for all pairs of sensor maps (includes pairing sensor maps of different environment). Gray-scale encodes the value of the match score ($0 \leq S_A \leq 1$). The cells on diagonal (marked with blue borders) represent the alignment of sensor maps versus the layout maps, and the red lines separate different environments. Green and red dots mark the success and failure of the alignments respectively. The squares on diagonal, corresponding to intra environment alignments, are slightly brighter compared to the rest of the matrix which corresponds to inter environment alignments. However, this is not conclusive enough to employ this measure across different

Table 2 Performance of the method in aligning sensor maps to layout maps

Map	Number of hypotheses		Result
	Initial	After rejection	
HHI_01	24	6	✓
HHI_02	48	8	✓
HHI_03	24	6	✓
HHI_04	36	4	✓
KPT_01	256	14	✓
KPT_02	128	10	✓
KPT_03	144	10	✓
KPT_01	160	10	✓
HH_E5_01	9152	726	✓
HH_E5_02	5280	458	×
HH_E5_03	6688	704	×
HH_E5_04	5984	458	×
HH_E5_05	5808	368	✓
HH_E5_06	2992	178	✓
HH_E5_07	3696	214	✓
HH_E5_08	4400	352	×
HH_E5_09	9152	616	✓
HH_E5_10	9152	794	✓
HH_E5_11	5808	470	✓
HH_E5_12	8624	644	✓
HH_E5_13	4928	336	×
HH_E5_14	3344	328	✓
HH_F5_01	1292	128	✓
HH_F5_02	1088	82	✓
HH_F5_03	816	86	✓
HH_F5_04	680	70	✓
HH_F5_05	544	56	✓
HH_F5_06	408	28	✓
HH_F5_07	476	26	✓
HH_F5_08	3604	158	✓
HH_F5_09	952	78	✓
HH_F5_10	680	56	×
HH_F5_11	680	264	✓
HH_F5_12	680	46	✓
HH_F5_13	1020	92	✓
HH_F5_14	1088	158	✓

The column *initial* represents the number of initial estimated transformations, *after rejection* represents the number of remaining hypotheses after rejecting non-uniformly scaled transformations ($\sim 90\%$ are rejected), and the last column marks the success of each alignment

environments and to identify a layout map to which a sensor map belongs. Under scrutiny it can be seen that maps of a smaller environment (KPT) align with a maps of 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, even though some are successfully aligned.

Table 3 The success rate of the method in aligning sensor maps to sensor maps, versus aligning sensor maps to layout maps

Environment	Sensor versus layout	Sensor versus sensor
HH_E5	64.28% (9/14)	50.54% (46/91)
HH_F5	92.85% (13/14)	68.13% (62/91)
HHI	100% (4/4)	100% (6/6)
KPT	100% (4/4)	83.33% (5/6)

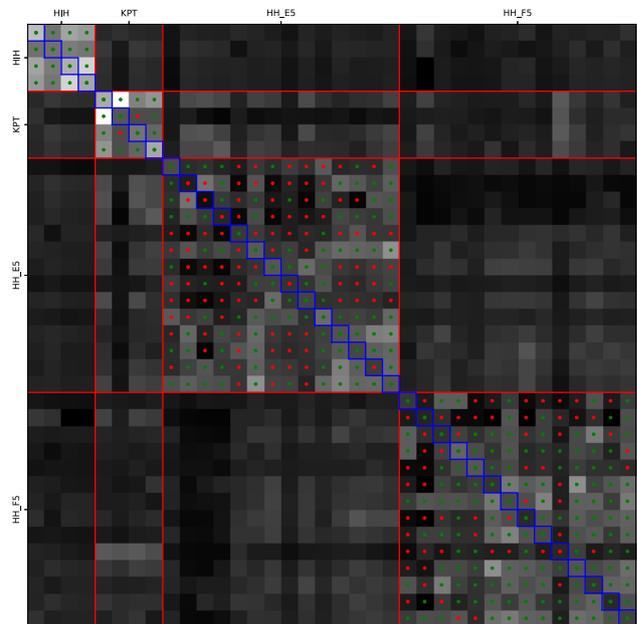


Fig. 8 The match score of the *winning hypotheses* for all pairs of sensor maps (includes pairing sensor maps of different environment). Gray-scale encodes the value of the alignment match score ($0 \leq S_A \leq 1$). The cells on diagonal (marked with blue borders) represent the alignment of sensor maps versus the layout maps, and the red lines separate different environments. Green and red dots mark the success and failure of the alignments respectively (Color figure online)

Figure 9 presents a box plot of the alignment match score for *all hypotheses* in aligning sensor maps to layout maps. The winning hypotheses are marked red and green, representing the failure and success of each alignment. There seems to be a cut-off point on the match score value across all maps (~ 0.15), which separates successful alignments from failures. However there is no reliable margin to this cut-off to be used as a threshold between success and failure. The take away message here is that the value of match score is not a reliable indicator of the alignment success.

In conclusion we can say, even though the *match score* has proven useful in selecting the winning alignment among all hypotheses, yet it is not conclusively reliable to detect to which layout map a sensor map belongs, nor to autonomously detect a successful alignment.

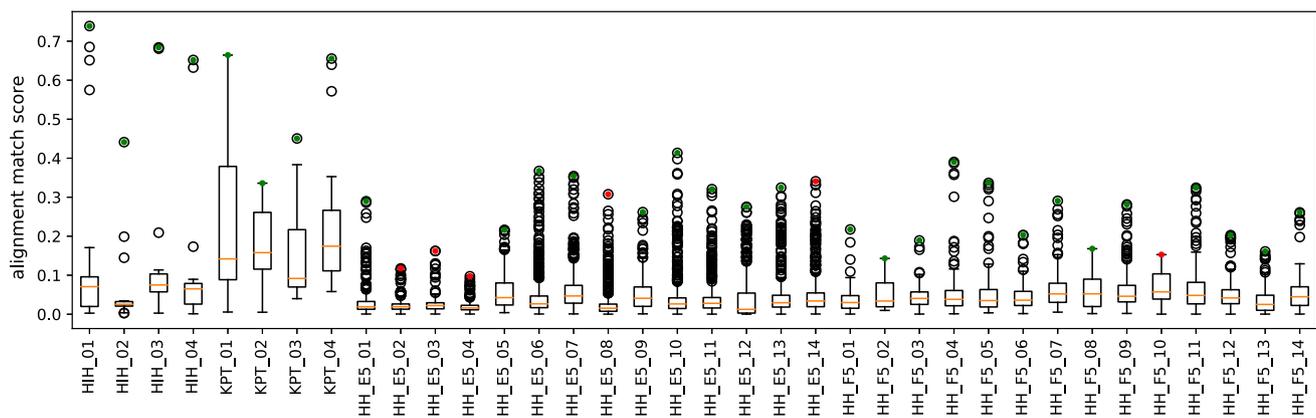


Fig. 9 The alignment match score for *all hypotheses* in sensor maps to layout maps alignments. The winning hypotheses are marked red and green, representing the failure and success of each alignment (Color figure online)

Table 4 Success rates (in %) of different methods on map alignment

Method	Sensor to layout					Sensor to sensor				
	HH_E5	HH_F5	HHI	KPT	Total	HH_E5	HH_F5	HHI	KPT	Total
ECC maximization (Evangelidis and Psarakis 2008)	0.0	7.14	0.0	0.0	2.77	37.36	29.67	0.0	16.67	31.9
SIFT (Lowe 1999)	21.43	50	0.0	0.0	27.7	17.58	28.57	16.67	0.0	22.1
CPD (Myronenko et al. 2007; Myronenko and Song 2010)	0.0	0.0	0.0	0.0	0.0	8.79	3.3	0.0	0.0	5.6
Saeedi et al. (PGVD, 2012)	0.0*	0.0*	25*	0.0*	2.77*	10.98	12.08	33.33	16.66	12.4
Carpin (2008)	0.0*	7.14*	0.0*	0.0*	2.77*	16.48	29.67	100	83.33	27.31
Our method	64.28	92.85	100	100	83.3	50.54	68.13	100	83.33	66.5

Numbers marked with an asterisk (*) refer to the methods that are not able to handle scaling. In those cases, the experiments were performed on manually scaled maps

Table 5 Average (and standard deviation) of the computation times (in seconds) of different methods, separated to home and office environments

Method	Implementation	Time in seconds	
		Home	Office
ECC maximization (Evangelidis and Psarakis 2008)	Python & C++	32.79 (28.24)	73.46 (85.46)
SIFT (Lowe 1999)	Python & C++	0.20 (0.05)	0.67 (0.14)
Saeedi et al. (PGVD, 2012)	Matlab	4.91 (1.42)	50.20 (19.84)
Carpin (2008)	C++	3.07×10^{-4} (9.28×10^{-5})	2.65×10^{-4} (6.72×10^{-5})
Our method	Python	8.86 (2.13)	41.86 (41.92)

4.3 Comparison with other methods

Our initial investigation and experiments towards map alignment lead to methods which we categorize into two groups. First the generic approaches in data association, such as image alignment, image registration and point set registration. And second the map alignment methods, such as Hough transform-based algorithms. The performance of all methods in terms of success rate is available in Table 4. And finally, a brief account of computational costs, presented in Table 5, will follow the performance evaluation.

4.3.1 Generic data association methods

Here the performances of three generic data association methods in map alignment are presented: (i) image alignment with enhanced correlation coefficient maximization (ECC) (Evangelidis and Psarakis 2008), (ii) image registration with scale-invariant feature transform (SIFT) (Lowe 1999), and (iii) point set registration with coherent point drift (CPD) (Myronenko et al. 2007; Myronenko and Song 2010). All the performance results are available in Table 4. We observed that the methods based on ECC and SIFT perform slightly better

when they are applied to the distance transform of the maps, instead of the occupancy maps. Accordingly the presented results are based on distance images of the maps.

ECC maximization performs worse on aligning sensor maps to layouts, due to its higher sensitivity to data-level similarity. A detailed review of the failed cases in aligning sensor maps to sensor maps reveals that the main causes of failure are the global inconsistencies of the sensor maps and small overlaps between the maps.

Image registration with SIFT (Lowe 1999) was tested in combination with fast approximate nearest neighbors (Muja and Lowe 2009) for feature matching. This method works best on maps with unique patterns represented by a unique constellation of “key points”, 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, this method yields better results in aligning sensor maps to layouts of bigger environments thanks to higher overlaps.

For the experiments with CPD, point sets have been generated from the occupied cells of the maps. CPD is superior to iterative closest point (ICP) as it supports affine transformations. However, it is computationally expensive and memory demanding, so that the original point sets had to be subsampled. This in turn conceals the structural patterns of the maps, and becomes more sensitive to local minima.

4.3.2 Map alignment methods

We have chose the works by Carpin (2008) and Saeedi et al. (PGVD) (2012) as the representatives of this category. Implementations of both methods are made publicly available by the authors. The performances of these methods are presented in Table 4.

One interesting aspect of these methods is their independence from the assumption of maps’ “segmentable regions”. Therefore they could be considered to have a broader 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 method. Also, thanks to the underlying decoupling of rotation and translation estimation, they could be relatively faster than other methods, specifically the method proposed by Carpin (2008). 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, they still failed at 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 noise, the global inconsistencies, and the repetitive patterns of our maps are the top challenges for such methods.

These methods are limited to rigid transformation, and as a result they could solve only the alignment of sensor to sensor map. Therefore, we manually adjusted scales of the layout maps, so that these methods could be evaluated over the alignments of sensor to layout maps. These results (manually adjusted scales) are marked with asterisks in Table 4. They score very low, which we believe is due to the significant disparity in the representations (i.e. different modalities). When contrasted with our method, this is an interesting result. Compared to the alignment of sensor to layout maps, our method scores lower when both maps are sensor maps. This is mainly due to the amplification of noise, inconsistency and partial coverage when both maps are sensor maps. On the other hand, these methods (Carpin 2008; Saeedi et al. 2012) perform worse when aligning sensor maps to layouts, due to their sensitivity to representation disparity.

On a final note, it is important to note that due to a lack of proper insight to the implementations of these methods, we could not fine-tune them, to maximize their performances in the setting of our experiments. Therefore we would like to point out, that the success rates of the methods presented in Table 4 might not represent their best performances, but rather they provide an insight into advantages and drawbacks of each method.

4.3.3 Computation time

The timings of all methods are provided in Table 5. All the experiments were carried out on a computer with an Intel® Core™ i5-3340M CPU @ 2.70 GHz × 4, and 8 GiB SODIMM DDR3 Synchronous 1600MHz of memory, running Ubuntu 14.04. The timings of experiments are separated into home and office building, which provides a sense of methods’ scalability with respect to the size of maps. The average map size for home environments is 2.2×10^5 pixels, and it is 1.0×10^6 pixels for office buildings (roughly 5 times bigger).

Since CPD is expensive and not scalable, the original point sets were reduced from 1.2×10^4 points on average in small maps and 3.3×10^4 points 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 some other approaches in terms of computational cost. Specifically, those methods designed for real-time applications such as Carpin’s method (2008) for multi-robot mapping, are extremely fast and hard to beat. Our method is based on the decomposition of the space and requires an interpretation through abstract models, which is in general computationally more expensive than signal based interpretations such

as Hough-spectra. However, if one intends to exploit the fast speed of the Hough transform-based methods in combination with our method, there is a trade-off between thoroughness of the hypotheses generation and computational time. In conclusion we speculated that, under certain assumptions (such as orthogonally structured environments), one can create a set of constraints imposed on the hypotheses generation to narrow down the search space. Although, a better understanding of such potential combination requires further development and more experiments.

At the end, we would like to emphasize that the timings of each method 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.

5 Conclusion

In this paper, we present our work and findings on solving the map alignment problem, for 2D spatial maps. Many interesting approaches have been proposed to address this problem. However, existing algorithms hinge on assumptions that are not valid in (a number of) interesting use cases, such as aligning partial maps of different modalities. 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, and consequently are sensitive to 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*. Such assumptions do not hold where maps of different modalities, such as sensor maps and layout maps, are to be aligned. 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, ECC), and some map alignment methods (designed for aligning maps of same modalities) in solving the problem in our experimental setup. We have compared the performance of our method with that of other methods both for sensor to sensor map alignment and sensor to layout map alignment. Except for few examples of similar performance, our method outperforms other methods. In aligning sensor to sensor maps, we observed that the presence of noise and global inconsistency

has been the main challenge for most other approaches. The representation disparity between maps of different modalities has been even more challenging for those methods in aligning sensor to layout maps. For the latter experiment, the layout maps were manually scaled to match the sensor maps in size, since other map alignment methods are limited to rigid transformation. Our 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 regions 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.

5.1 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. We do not claim, or believe, that our method is superior to other approaches in a generic problem formulation of data association and map alignment. 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.

Advantages Apart from the higher success rate of our proposed method, we would like to point out some other interesting features of it. One important aspect, and one of the main motivations behind this work, is the ability to align maps of different modality, and specifically sensor maps to layout maps. 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 (success rate 83.3% compared to the best alternative 27.7%). We have developed a region 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 spa-

tial and could be segmented into 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 (Gholami Shahbandi et al. 2015, 2014), and it is not alignment-specific.

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 improves the map alignment under difficult circumstances, 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 a maze-like environment, unless an appropriate region segmentation algorithm is employed. Partial maps which don't cover multiple regions (e.g. a map of only one room), in applications such as scan matching and incremental mapping, violate one of the initial assumptions and would cause our method to fail. We speculate that this is a domain where other methods such as the ones proposed by Carpin (2008) and Saeedi et al. (2012) would outperform our method, given the maps are from the same modality. As stated before, in Sect. 4, not all the maps satisfy our initial assumptions such as global consistency. 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, even if they violate the assumptions of our method. Other conditions that make our 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.

Model regression Random sample consensus (RANSAC) is a powerful regression technique in estimating a model from noisy data. However, our empirical observation suggests that RANSAC is not a suitable replacement for the components of our method. The first possibility is to employ RANSAC for hypothesis generation, i.e. estimating a transformation between faces with known correspondences. We have found Umeyama's method to be a better fit as a non-iterative deterministic method for this objective. Alternatively RANSAC could be considered for solving the alignment on the map level with unknown correspondences. However, the point

sets from our representation (vertices of the prime graph) are sparse and do not reflect the skeletal structure. This challenge is exaggerated with relatively high level of noise and partiality of the maps. We have experimented with RANSAC in this manner, with a simple setup and a denser sampling of the occupied points, the result of which has not been satisfactory. This led to our experimentation with iterative closest points (ICP) and Coherent point drift (CPD), a continuation of the attempt in relying on the shape of the distribution of occupied points. The results of CPD have been included in this manuscript as a representative of this category of approaches. Despite the inadequacy of RANSAC in estimating the alignment, we speculate that such regression techniques could be beneficial in estimating other models as a part of a more elaborate method. For instance, RANSAC can be used for the regression of the *geometric coherency of hypotheses*. That is to say, assuming a correct alignment is represented with multiple hypotheses, the pool of hypotheses is expected to contain clusters of similar transformations. RANSAC can be used for estimating the geometric coherency of hypotheses and rejecting outliers. We ran experiments with this idea, although with a clustering algorithm (DBSCAN (Ester et al. 1996)) and not RANSAC. The challenge is that not always the correct alignment has multiple representatives, specially for small, deformed, and partial maps. This idea needs further investigation, since treating the pool of hypotheses has to be done carefully with additional considerations.

5.2 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 an *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. Furthermore, we intend to carry out more challenging experiments and with other modalities to inspect the performance of the proposed alignment approach 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 relying on the structure of the layout map.

Acknowledgements This work was funded by the Swedish Knowledge Foundation (KK-Stiftelsen), and the European Union's Horizon 2020

research and innovation programme under grant agreement No 732737 (ILIAD). The authors would like to express their gratitude to Dr. Karl Iagnemma and the anonymous reviewers who helped us improve the quality of the manuscript with their valuable and constructive inputs.

Open Access This article is distributed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits unrestricted use, distribution, and reproduction in any medium, provided you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license, and indicate if changes were made.

Appendix

Map data-set

See Figs. 10, 11, 12 and 13.

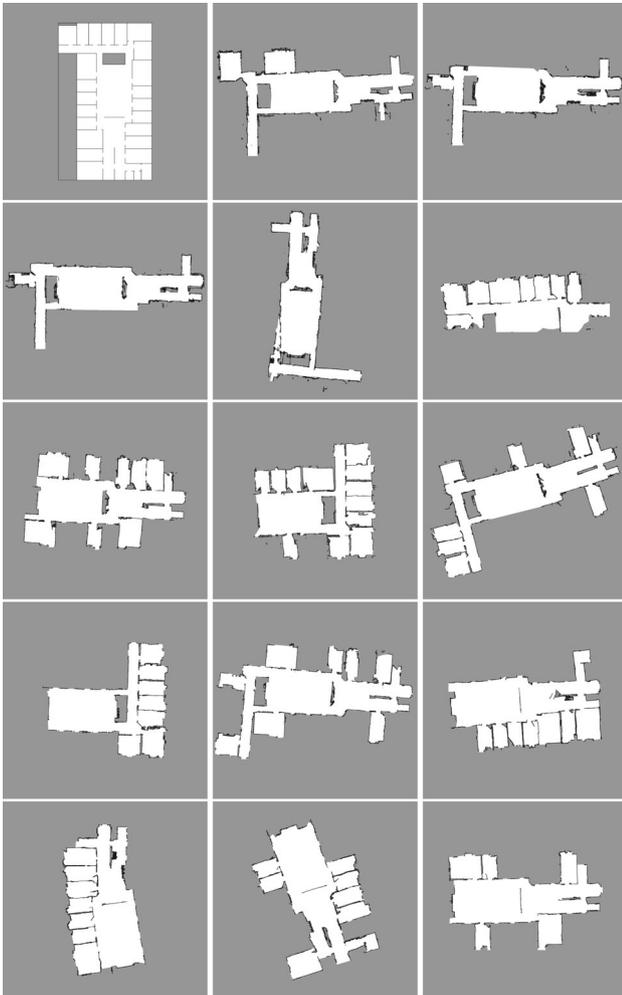


Fig. 10 HH_E5 (office building in Halmstad, Sweden)

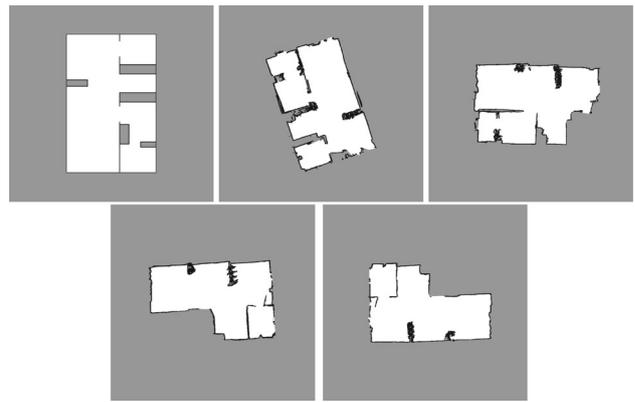


Fig. 11 HII (Halmstad Intelligent Home (Lundström et al. 2016))

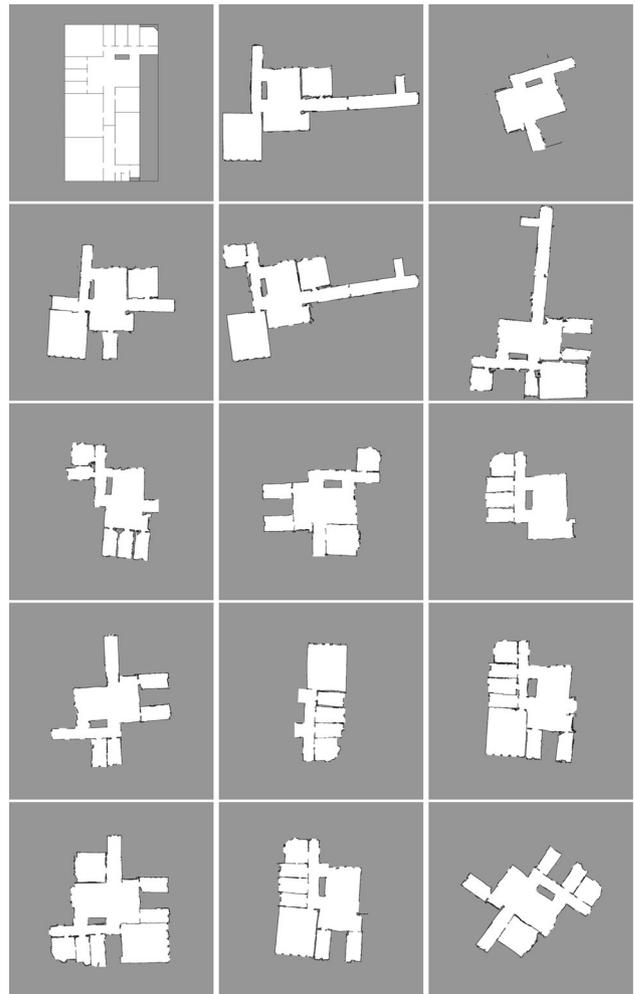


Fig. 12 HH_F5 (office building in Halmstad, Sweden)

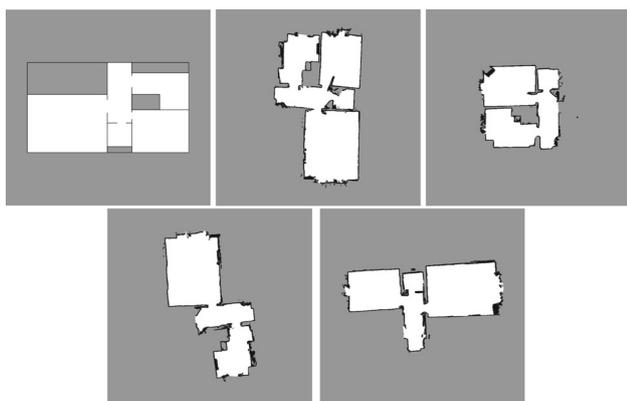


Fig. 13 KPT (apartment in Halmstad, Sweden)

References

- Agarwal, P. K., & Sharir, M. (2000). Chapter 2—Arrangements and their applications*. In J. R. Sack & J. Urrutia (Eds.), *Handbook of computational geometry* (pp. 49–119). Amsterdam: North-Holland.
- Baker, S., & Matthews, I. (2004). Lucas–Kanade 20 years on: A unifying framework. *International Journal of Computer Vision*, 56(3), 221–255.
- Ballard, D. H. (1981). Generalizing the hough transform to detect arbitrary shapes. *Pattern Recognition*, 13(2), 111–122.
- Besl, P. J., & McKay, N. D. (1992). A method for registration of 3-D shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(2), 239–256.
- Birk, A., & Carpin, S. (2006). Merging occupancy grid maps from multiple robots. *Proceedings of the IEEE*, 94(7), 1384–1397.
- Bonanni, T. M., Corte, B. D., & Grisetti, G. (2017). 3-D map merging on pose graphs. *IEEE Robotics and Automation Letters*, 2(2), 1031–1038.
- Bonanni, T. M., Grisetti, G., & Iocchi, L. (2014). *Merging partially consistent maps* (pp. 352–363). Cham: Springer.
- Boniardi, F., Behzadian, B., Burgard, W., & Tipaldi, G. D. (2015). Robot navigation in hand-drawn sketched maps. In *2015 European conference on mobile robots (ECMR)*(pp. 1–6).
- Bormann, R., Jordan, F., Li, W., Hamp, J., & Hägele, M. (May 2016). Room segmentation: Survey, implementation, and analysis. In *2016 IEEE international conference on robotics and automation (ICRA)*(pp. 1019–1026).
- Bosse, M., & Zlot, R. (2008). Map matching and data association for large-scale two-dimensional laser scan-based slam. *The International Journal of Robotics Research*, 27(6), 667–691.
- Bowen-Biggs, L., Dazo, S., Zhang, Y., Hubers, A., Rueben, M., Sowell, R. T., Smart, W. D., & Grimm, C. (2016). Sketched floor plans versus SLAM maps: A comparison. [arxiv:1606.04836](https://arxiv.org/abs/1606.04836).
- Bradski, G. (2000). The OpenCV library. *Dr. Dobbs's Journal of Software Tools*. Retrieved from <https://github.com/opencv/opencv/wiki/CiteOpenCV>.
- Carpin, S. (2008). Fast and accurate map merging for multi-robot systems. *Autonomous Robots*, 25(3), 305–316.
- Carpin, S., & Birk, A. (2005). *Stochastic map merging in rescue environments* (pp. 483–490). Berlin: Springer.
- Carpin, S., Birk, A., & Jucikas, V. (2005). On map merging. *Robotics and Autonomous Systems*, 53(1), 1–14.
- Dedeoglu, G., & Sukhatme, G. S. (2000). *Landmark-based matching algorithm for cooperative mapping by autonomous robots* (pp. 251–260). Tokyo: Springer Japan.
- Erinc, G., Balaguer, B., & Carpin, S. (2013, Nov). Heterogeneous map merging using wifi signals. In *2013 IEEE/RSJ international conference on intelligent robots and systems* (pp. 5258–5264).
- Ester, M., Kriegel, H.-P., Sander, J., & Xu, X. (1996). A density-based algorithm for discovering clusters a density-based algorithm for discovering clusters in large spatial databases with noise. In *Proceedings of the 2nd international conference on knowledge discovery and data mining, KDD'96* (pp. 226–231). AAAI Press.
- Evangelidis, G. D., & Psarakis, E. Z. (2008). Parametric image alignment using enhanced correlation coefficient maximization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(10), 1858–1865.
- Fermin-Leon, L., Neira, J., & Castellanos, J. A. (2017, May). Incremental contour-based topological segmentation for robot exploration. In *2017 IEEE ICRA* (pp. 2554–2561).
- Fox, D., Ko, J., Konolige, K., Limketkai, B., Schulz, D., & Stewart, B. (2006). Distributed multirobot exploration and mapping. *Proceedings of the IEEE*, 94(7), 1325–1339.
- Georgiou, C., Anderson, S., & Dodd, T. (2017). 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.
- Gholami Shahbandi, S., Åstrand, B., & Philippsen, R. (2014). Sensor based adaptive metric-topological cell decomposition method for semantic annotation of structured environments. In *2014 13th International conference on control automation robotics vision (ICARCV)* (pp. 1771–1777).
- Gholami Shahbandi, S., Åstrand, B., & Philippsen, R. (2015). Semi-supervised semantic labeling of adaptive cell decomposition maps in well-structured environments. In *2015 European conference on mobile robots (ECMR)* (pp. 1–8).
- Gold, S., Rangarajan, A., Lu, C.-P., Pappu, S., & Mjolsness, E. (1998). New algorithms for 2d and 3d point matching: Pose estimation and correspondence. *Pattern Recognition*, 31(8), 1019–1031.
- Hagberg, A. A., Schult, D. A., & Swart, P. J. (2008, August). Exploring network structure, dynamics, and function using NetworkX. In *Proceedings of the 7th python in science conference (SciPy2008)*, Pasadena, CA USA (pp. 11–15).
- Howard, A. (2004, April). Multi-robot mapping using manifold representations. In *Proceedings of the 2004 IEEE international conference on robotics and automation, 2004. ICRA'04* (Vol. 4, pp. 4198–4203).
- Howard, A., Parker, L. E., & Sukhatme, G. S. (2006). Experiments with a large heterogeneous mobile robot team: Exploration, mapping, deployment and detection. *The International Journal of Robotics Research*, 25(5–6), 431–447.
- Hu, M.-K. (1962). Visual pattern recognition by moment invariants. *IRE Transactions on Information Theory*, 8(2), 179–187.
- Huang, W. H., & Beevers, K. R. (2005). Topological map merging. *The International Journal of Robotics Research*, 24(8), 601–613.
- Hunter, J. D. (2007). Matplotlib: A 2d graphics environment. *Computing in Science Engineering*, 9(3), 90–95.
- Jiang, Z., Zhu, J., Li, Y., Li, Z., & Lu, H. (2017, June). *Simultaneous merging multiple grid maps using the robust motion averaging*. ArXiv e-prints.
- Jones, E., Oliphant, T., & Peterson, P., et al. (2001). *SciPy: Open source scientific tools for Python*. [Online; Accessed May, 2017].
- Kakuma, D., Tsuichihara, S., & Ricardez, G. A. G., Takamatsu, J., Ogasawara, T. (Jan 2017). Alignment of occupancy grid and floor maps using graph matching. In *2017 IEEE 11th international conference on semantic computing (ICSC)* (pp. 57–60).
- Konolige, K., Fox, D., Limketkai, B., Ko, J., & Stewart, B. (2003, Oct). Map merging for distributed robot navigation. In *Proceedings 2003 IEEE/RSJ international conference on intelligent robots and systems (IROS 2003) (Cat. No.03CH37453)* (Vol. 1, pp. 212–217).

- Lowe, D. G. (1999). Object recognition from local scale-invariant features. In *Proceedings of the seventh IEEE international conference on computer vision* (Vol. 2, pp. 1150–1157).
- Lucas, B. D., & Kanade, T. (1981). An iterative image registration technique with an application to stereo vision. In *Proceedings of the 7th international joint conference on artificial intelligence, IJCAI'81*, San Francisco, CA, USA (Vol. 2, pp. 674–679). Morgan Kaufmann Publishers Inc.
- Lundström, J., De Morais, W. O., Menezes, M., Gabrielli, C., Bentes, J., Sant'Anna, A., et al. (2016). *Halmstad intelligent home—Capabilities and opportunities*. Cham: Springer.
- Meurer, A., Smith, C. P., Paprocki, M., Čertk, O., Kirpichev, S. B., Matthew Rocklin, A., et al. (2017). Sympy: Symbolic computing in python. *PeerJ Computer Science*, 3, e103.
- Mielle, M., Magnusson, M., Andreasson, H., & Lilienthal, A. J. (2017). *SLAM auto-complete using an emergency map*. [arxiv:1702.05087](https://arxiv.org/abs/1702.05087).
- Mielle, M., Magnusson, M., & Lilienthal, A. J. (2016). Using sketch-maps for robot navigation: Interpretation and matching. In *2016 IEEE international symposium on safety, security, and rescue robotics (SSRR)* (pp. 252–257).
- Mielle, M., Magnusson, M., & Lilienthal, A. J. (2017). A method to segment maps from different modalities using free space layout—MAORIS: Map of ripples segmentation. [arxiv:1709.09899](https://arxiv.org/abs/1709.09899).
- Muja, M., & Lowe, D. G. (2009). Fast approximate nearest neighbors with automatic algorithm configuration. In *VISAPP international conference on computer vision theory and applications* (pp. 331–340).
- Myronenko, A., & Song, X. (2010). Point set registration: Coherent point drift. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(12), 2262–2275.
- Myronenko, A., Song, X., & Carreira-Perpi nán, M. Á. (2007). Non-rigid point set registration: Coherent point drift. In P. B. Schölkopf, J. C. Platt, & T. Hoffman (Eds.), *Advances in neural information processing systems* (Vol. 19, pp. 1009–1016). Cambridge: MIT Press.
- Neuhaus, M., & Bunke, H. (2004). *An error-tolerant approximate matching algorithm for attributed planar graphs and its application to fingerprint classification* (pp. 180–189). Berlin: Springer.
- Port, A. (2017). *Svgpathtools; a collection of tools for manipulating and analyzing SVG path objects and Bezier curves*. <https://github.com/mathandy/svgpathtools>.
- Rädler, J. (2015). *Polygon: Python bindings to the 'general polygon clipping library' (GPC) and additional functions*. <https://www.j-raedler.de/projects/polygon/>.
- Radon, J. (1986). On the determination of functions from their integral values along certain manifolds. *IEEE Transactions on Medical Imaging*, 5(4), 170–176.
- Saeedi, S., Paull, L., Trentini, M., Seto, M., & Li, H. (2012, Oct). Efficient map merging using a probabilistic generalized voronoi diagram. In *2012 IEEE/RSJ international conference on intelligent robots and systems* (pp. 4419–4424).
- Saeedi, S., Paull, L., Trentini, M., Seto, M., & Li, H. (2014). Group mapping: A topological approach to map merging for multiple robots. *IEEE Robotics Automation Magazine*, 21(2), 60–72.
- Saeedi, S., Paull, L., Trentini, M., Seto, M., & Li, H. (2014). Map merging for multiple robots using hough peak matching. *Robotics and Autonomous Systems*, 62(10), 1408–1424.
- Saeedi, S., Trentini, M., Seto, M., & Li, H. (2016). Multiple-robot simultaneous localization and mapping: A review. *Journal of Field Robotics*, 33(1), 3–46.
- Sanchez, C. A., & Branaghan, R. J. (2009). The interaction of map resolution and spatial abilities on map learning. *International Journal of Human-Computer Studies*, 67(5), 475–481.
- Sanfeliu, A., & Fu, K. S. (1983). A distance measure between attributed relational graphs for pattern recognition. *IEEE Transactions on Systems, Man, and Cybernetics, SMC-13*(3), 353–362.
- Schwertfeger, S., & Birk, A. (2013, May). Evaluation of map quality by matching and scoring high-level, topological map structures. In *2013 IEEE international conference on robotics and automation* (pp. 2221–2226).
- Thrun, S. (2001). A probabilistic on-line mapping algorithm for teams of mobile robots. *The International Journal of Robotics Research*, 20(5), 335–363.
- Tsin, Y., & Kanade, T. (2004). *A correlation-based approach to robust point set registration* (pp. 558–569). Berlin: Springer.
- Umeyama, S. (1991). Least-squares estimation of transformation parameters between two point patterns. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(4), 376–380.
- van der Walt, S., Colbert, S. C., & Varoquaux, G. (2011). The numpy array: A structure for efficient numerical computation. *Computing in Science Engineering*, 13(2), 22–30.
- van der Walt, S., Schönberger, J. L., Nunez-Iglesias, J., Boulogne, F., Warner, J. D., Yager, N., et al. (2014). and the scikit-image contributors. scikit-image: image processing in Python. *PeerJ*, 2, e453, 6.
- Vysotska, O., & Stachniss, C. (2017). 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.
- Wallgrün, J. O. (2010). Voronoi graph matching for robot localization and mapping. In M. L. Gavrilova, C. J. K. Tan, & F. Anton (Eds.), *Transactions on computational science IX: Special issue on voronoi diagrams in science and engineering* (pp. 76–108). Berlin: Springer.
- Williams, S. B., Dissanayake, G., & Durrant-Whyte, H. (2002). Towards multi-vehicle simultaneous localisation and mapping. In *Proceedings 2002 IEEE international conference on robotics and automation (Cat. No.02CH37292)* (Vol. 3, pp. 2743–2748).
- Yujian, L., & Bo, L. (2007). A normalized levenshtein distance metric. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(6), 1091–1095.
- Zhou, X. S., & Roumeliotis, S. I. (2006). Multi-robot slam with unknown initial correspondence: The robot rendezvous case. In *2006 IEEE/RSJ international conference on intelligent robots and systems* (pp. 1785–1792).

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Saeed Gholami Shahbandi obtained degrees in Electronics and Robotics from Mazandaran University (Iran, 2010) and Ecole Centrale de Nantes (France, 2012). He then joined the School of Information Technology at Halmstad University, Sweden, and he is a Ph.D. candidate with the Center for Applied Intelligent Systems Research (CAISR). His research focuses on interpretation, modeling and semantic annotations of robot maps, which encompasses topics in computer vision,

machine learning and robot mapping.



Martin Magnusson received his M.S. degree in computer science from Uppsala University, Sweden, in 2004 and Ph.D. degree from Örebro University, Sweden, in 2009. He is currently a lektor (senior lecturer) in the MRO Lab of the Center of Applied Autonomous Sensor Systems (AASS), Örebro University, Sweden. Dr. Magnusson has been vice-chair of the working group for the IEEE/RAS standards for 2D and 3D map representations and is deputy chair for the eu- Robotics

topic group on robots for logistics and transport. Dr. Magnusson's research interests include 3D perception (including efficient and versatile 3D surface representations), creation and usage of robot maps that go beyond mere geometry, and methods for making use of heterogeneous maps with high uncertainty.