Nonlinear Optimization of Multimodal 2D Map Alignment with Application to Prior Knowledge Transfer

Saeed Gholami Shahbandi[1], Martin Magnusson[2] and Karl Iagnemma[3]

***Abstract*— We propose a method based on a non-linear transformation for non-rigid alignment of maps of different modalities, exemplified with matching partial and deformed 2D maps to layout maps. For two types of indoor environments, over a data-set of 40 maps, we have compared the method to state-of-the-art map matching and non-rigid image registration methods and demonstrate a success rate of 80.41% and a mean point-to-point alignment error of 1.78 meters, compared to 31.9% and 10.7 meters for the best alternative method. We also propose a fitness measure that can quite reliably detect bad alignments. Finally we show a use case of transferring prior knowledge (labels/segmentation), demonstrating that map segmentation is more consistent when transferred from an aligned layout map than when operating directly on partial maps (95.97% vs. 81.56%).**

I. Introduction

The ability to build a map is a prerequisite for many robotic applications such as environment surveying whether it be for industrial automation or search and rescue, and service robots from home-care to industrial transportation. Such maps are the robot's internal representation of the world, an essential element of their autonomy. However these maps are sometimes partial, deformed or do not contain sufficient information for elaborate task planning. The ability to autonomously establish an association between different sources can considerably improve a robot's knowledge. A layout map (blueprint), for instance, carries prior knowledge that could be leveraged to improve the performance of Simultaneous Localization And Mapping (SLAM) upon architectural/structural information, or enable an elaborate task planning based on the semantic labels, and provide a mutual frame of reference for alignment and merging of partial maps in case of multi-agent mapping. Construction of a hybrid map by merging maps of different modalities, enables the robot to access all available modalities through individual maps.

Problem description: The focus of this work, as shown by the general flow of our proposed method in Fig. 1, is the alignment of robot (sensor) maps and layout maps. The different types and sizes of maps and their partial coverage are among the most important challenges in autonomous alignment of sensor and layout maps. Further challenges arise when the robot map is erroneous and not globally consistent (i.e deformed). A globally consistent map is a map that could be aligned with the ground truth with a similarity matrix, i.e. only rotation, translation and uniform scaling. It is desired to use the global consistency of the layout map to rectify deformation of the robot map, and therefore the solution must support nonlinear transformation. To that end, we use a decomposition-based map alignment technique from our previous work [2] to estimate an initial alignment, after which the problem becomes an optimization problem. Our method assumes that: i) the target map (layout) is globally consistent, ii) the source map (sensor) covers a subset of the target map, and iii) deformations of sensor maps are continuous, i.e. there is no "brokenness" in maps.

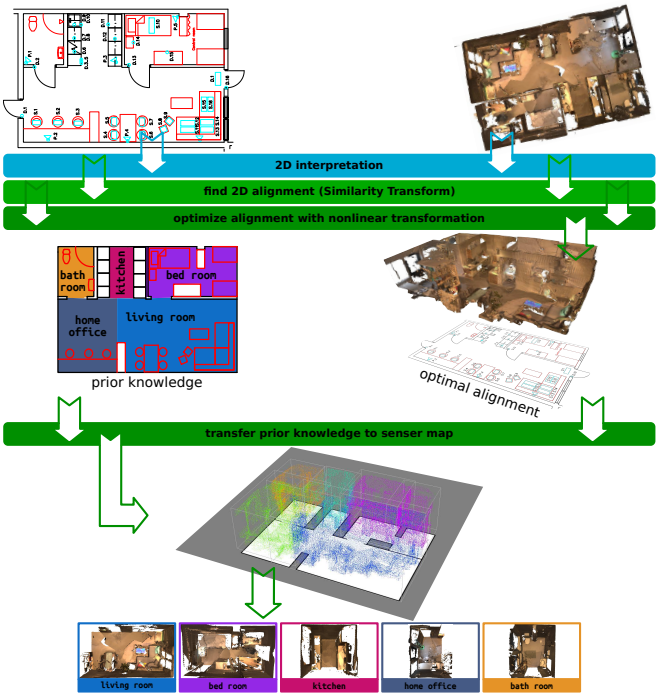


Fig. 1: In this work we present a method for optimizing the alignment, and show an example of using the alignment for transferring the prior knowledge from layout map to sensor map. Sensor maps are acquired with a Google Tango tablet as 3D meshes, and converted to 2D occupancy-like maps. This example is from Halmstad Intelligent Home [1].

Our approach: Although 2D grid maps can also be seen as images, we argue that the locations of occupied cells is more prominent information than the image intensity values. This argument will be further discussed in Sec. II-B and III-B. Accordingly, in this work, occupied cells are adopted as the basis of interpretation for data association. The occupied

This work was supported by the Swedish Knowledge Foundation.

[1]Saeed Gholami Shahbandi is with the Center for Applied Intelligent Systems Research, Halmstad University, Sweden saesha@hh.se

[2]Martin Magnusson is with the Center for Applied Autonomous Sensor Systems (AASS), Örebro University, Sweden martin.magnusson@oru.se

[3]Karl Iagnemma is with the Robotic Mobility Group, Massachusetts Institute of Technology, USA kdi@mit.edu

cells of the source map are sampled to an almost uniformly distributed point set, representing the structural outline of the environment. The target map underlies a fitness function that is highest at occupied cells and decreases by distance. Pinning down the representations to a point set and a fitness function, the formulation of data association simplifies to a local optimization over the fitness of the points. Additionally we impose a coherency condition to maintain the local consistency of the maps, A piece-wise affine transformation is employed to represent the solution. Sec. III presents the method in detail. The contributions of this work are:

- A method is proposed for the optimization of an alignment with a non-linear transformation, in order to simultaneously fine-tune the alignment and correct sensor map deformation.
- A simple and reliable measure of assessing the alignment quality is proposed.
- Finally a novel *strategy* for improving the consistency of region segmentation of partial maps is presented.

II. Related work

Those works most relevant to the objective of this paper are *map matching* from robot mapping (Sec. II-A), and *image registration* from the broader image processing topic (Sec. II-B.) In each category we present a few notable methods as examples that perform robustly in their related context, and review their shortcomings in solving the map alignment under the conditions specified in Sec. I.

A. Map matching

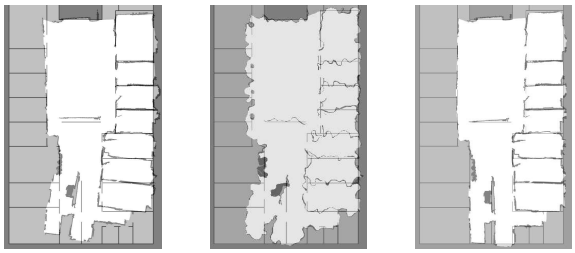
Two of the sub-problems in *graph theory* that are most relevant to map alignment are the Maximal Common Subgraph, and the error-tolerant sub-graph isomorphism. Some interesting map alignment methods based on graph theory have been proposed by Huang and Beevers [3], Wallgrün [4], Schwertfeger and Birk [5], Mielle et al. [6], and Kakuma et al. [7]. *Hough/Radon transform-based* map matching methods find the alignment by decomposing it into rotation and translation estimation. Such approaches are often deterministic, non-iterative, and fast, thanks to this decomposition. Carpin [8], Bosse and Zlot [9], Saeedi et al. [10] presented some inspiring work with this approach. In our previous work [2], we showed the challenges that most map alignment methods face in dealing with noisy maps, different scales, and maps of different types. Park et al. [11] proposed a map matching method for maps with uncertainties in scale, but assume that the maps have the same type. We proposed a decomposition-based map alignment method [2], and its advantages in handling noisy maps, supporting similarity instead of a rigid transformation, and handling discrepancy in representations, make the method suitable for aligning sensor maps with layout maps. Map deformity is another challenge in map alignment, that requires a non-linear transformation model. Addressing this challenge in particular is the main objective of this work. Bonanni et al. [12] perform a 3D map merging with *pose graphs* with a non-linear transformation, to account for distortions of the maps. However, their method would not be applicable when a pose graph is not available, as it is the case for layout maps.

B. Image registration

Image alignment methods such as Lucas-Kanade [13] and Enhanced Correlation Coefficient (ECC) Maximization [14] are from a category of image processing methods with linear transformation models. These methods fall short of solving the map alignment due to the discrepancy in data representation, i.e. different map types, and a lack of sufficient local information. *Point set registration* is another category, and they can be either shape-based such as Iterative Closest Point (ICP) [15] and Coherent Point Drift (CPD) [16], or feature-based such as Scale-Invariant Features Transform (SIFT) [17]. *Active Models* such as Active Shape Models [18] and Active Appearance Models [19] are examples of using domain knowledge to simplify the harder problem by building statistical shape models. This approach is not suitable for map alignment, since they require a distinct and consistent pattern to be represented by a model (as in faces, or leaves), and expects sufficient information in the images for training their statistical models.

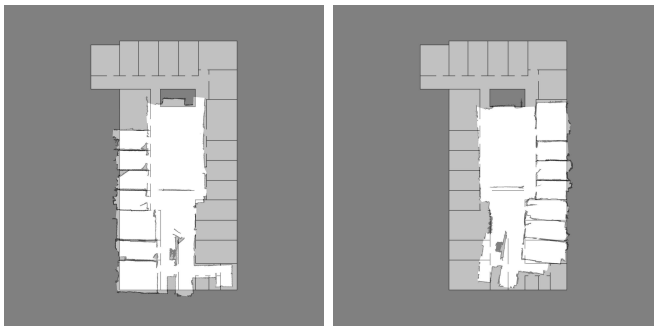
Free Form Deformation (FFD) field [20], often based on B-spline curves, is another approach to image registration that supports a nonlinear transformation model. These are most frequently used in medical image processing [21]. By supporting non-linearity, this category of methods takes on a very challenging problem with many parameters to estimate. As a consequence of this considerably big search space, these methods require a lot of local information for a successful convergence. Image registration methods based on FFD field [20] seem to be the most suitable alternatives to this work, since they locally optimize the alignment of two images, and support a non-linear transformation. We have studied some of the state-of-the-art "nonrigid image registration" techniques from the field of medical imaging [22]. The outcomes have been consistently unsatisfactory, with severe local deformations of the source maps [1]. Fig. 2 exemplifies the performance of such methods on occupancy maps, based on an implementation from the *ITK* library [23]. This is not an isolated example, and represents the general behavior of methods with the FFD field approach. The outcomes of operating on distance transform of the maps have been similar. The reason, we believe, is the fact that an image intensity-based optimization, in conjunction with a complex "non-rigid" transformation model, requires a higher level of local information. From an image processing perspective, occupancy maps are mostly patches of low information (open-space and unexplored areas), unlike most other vision signals (e.g. medical images) where the information is distributed more uniformly over the image. This makes the biggest challenge for employing most of the aforementioned image

[1] `BSplineTransform` and `DisplacementFieldTransform` for transformation model, `Correlation`, `MeanSquares` and `MattesMutualInformation` for similarity metric, and `Exhaustive`, `Gradient Descent` and `L-BFGS-B` for optimizer are some of the examples we studied.



(a) initial alignment (b) FFD field (c) this work

Fig. 2: Comparison between this work and an FFD field-based method [22], [23] on optimizing an initial alignment. Oversensitivity of the FFD field-based method to representation discrepancy and lack of sufficient local information can be observed in Fig. 2b.



(a) slight misalignment (b) map deformity

Fig. 3: Two examples where the initial alignments are correct, but suffer from minor defects.

processing techniques for map alignment. This also explains the appeal of abstract representations in map matching, such as Hough-spectra, Voronoi graphs and region decomposition, that benefit from the global structure of the maps.

III. Method

The main objective of this paper is to optimize an initial alignment between two maps. This initial alignment is provided via a decomposition based map alignment technique [2], which is outlined in Sec. III-A. This alignment approach, like most others [8], [10], is global and cannot guarantee a locally accurate solution in the presence of noise and map deformation. In Sec. III-B we present an optimization process, which provides a non-linear solution to the problem in form of a piece-wise affine transformation.

A. Model based alignment (decomposition-based)

Aligning sensor maps to layout maps includes the additional challenges of different map sizes, coverages and types. The decomposition-based map alignment method [2] specifically addresses the problem under those circumstances. The idea behind this method is to decompose the map into regions, and represent the decomposition with a Doubly-Connected Edge List (DCEL) data structure. The alignment solution is the best fitting hypothesis among all hypotheses generated from matching each region in one map to all regions in the other map. For the details of the decomposition process, the DCEL representation, the hypotheses generation, and the selection of best fitting hypothesis, please see our previous work [2].

B. Signal based optimization (occupancy map)

As the examples in Fig. 3 show, the initial alignment could be off from the optimal value, or the optimal alignment of a deformed sensor map is not achievable with a linear transformation. We remedy these deficiencies by optimizing the initial alignments and correcting the global inconsistency of the sensor map. The underlying problem which this optimization intends to solve involves data association, and the choice of data representation is crucial. The representations are expected to capture local information with highest level of fidelity from the environment that are mutual between layout and sensor map. Abstract models often lack details of the maps, Voronoi graphs are sensitive to clutter, and Hough-space does not have an explicit local representation. Accordingly, we base the objective function on the occupied cells of the maps, as they best satisfy the requirements.

Map interpretation: As shown with an example in Fig. 4, a collection of control points X are detected by the "Good Features to Track" [24] from the occupied cells of the source map (i.e. sensor map). The occupied cells of the target map (i.e. layout map) underlie a fitness function (map) M_f, as illustrated in Fig. 5, and the gradient map M_g is computed from M_f for the gradient ascent optimization

$$M_o \xmapsto[\text{transform}]{\text{distance}} M_d \xmapsto[\text{function}]{\text{Gaussian}} M_f \xmapsto{\text{gradient}} M_g$$

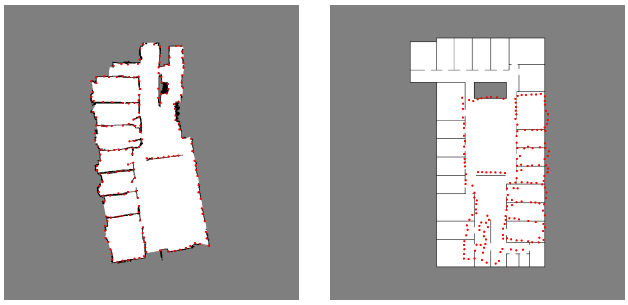
$$\begin{aligned} M_d &= \mathcal{DT}(M_o) \\ M_f &= [\exp(-d_i^2/2\sigma_f^2) \mid \forall d_i \in M_d] \\ M_g &= \tfrac{\partial M_f}{\partial x} + i\tfrac{\partial M_f}{\partial y} \end{aligned}$$

where σ_f defines the neighborhood of the fitness map, and $\mathcal{DT}(M_o)$ is a distance transform of the occupancy map which represents the distance of each open cell to its closest occupied cell. The fitness map M_f is a Radial Basis Function (Gaussian) applied to the distance value of each pixel in M_d, i.e. the farther a cell is from occupied points the lower its fitness value is. Fig. 5 shows an occupancy map with its distance, fitness, and gradient maps. The optimal value of σ_f depends on the structure of the environment, and more specifically the size of the open spaces. Based on our empirical observation, a value in the range of $\sigma_f = 1 \pm 0.4$ meter yields satisfactory results for home and office maps.

Optimization of the alignment: The control points of the source map X together with the fitness function of the target map M_f form the objective function of the optimization

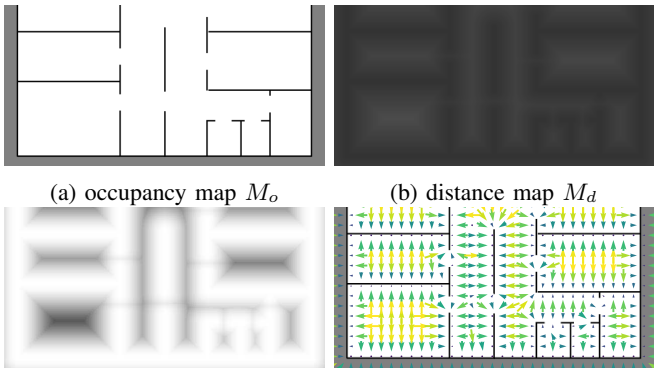
$$dX = \arg\max_{dX} \sum_{i=1}^{K} M_f(x_i + dx_i) \mid x \in X, dx \in dX$$

where K is the number of control points. The solution to this optimization is a motion matrix dX, where each row is a 2D motion vector corresponding to control points X. Like most conventional implementations of gradient ascent, at each iteration the control points are displaced according to incremental steps of dX which is computed by indexing the gradient map with the latest location of each point.



(a) control points X (b) X aligned with target map

Fig. 4: Interpretation of the source map (sensor) is a collection of points X, representing occupied cells.



(a) occupancy map M_o (b) distance map M_d

(c) fitness map M_f (d) gradient map M_g

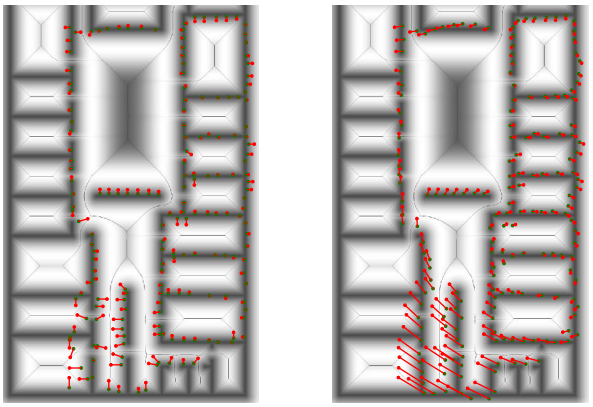
Fig. 5: Interpretations of the target map (layout), from occupancy to gradient.

Transformation model: The model to represent the optimized alignment is a piece-wise transformation. According to this model, the area enclosed by the convex hull of all the points is tessellated with a Delaunay triangulation. Each simplex of the tessellation is then assigned an affine transformation, that is estimated from the motion of its three vertices.

Coherency condition: The presented formulation of the optimization only incentivises the fitness of X with respect to M_f without any regard to the patterns of X. Fig. 6a shows an example of this optimization resulting in an incoherent motion of X. To assure the coherency of the motions, we modify the incremental dX by adjusting the motion of each control point to accord with its neighbors. To this end, the *coherent motion* of each control point is defined as a weighted average of its own and its neighbors' *uncorrelated motions* that are obtained directly from the gradient map. The averaging is weighted by a Gaussian function of the distance between two control points. Fig. 6 demonstrates the effect of this coherency adjustment. The coherent motion can be expressed as

$$dx'_i = \frac{1}{K} \sum_{j=1}^{K} dx_j . w_{ij}$$

where dx_j is the motion of point x_j obtained directly from the gradient map, and w_{ij} is the correlation between pairs



(a) independent motions dX (b) coherent motions dX'

Fig. 6: Motion dX is enforced to be coherent among neighboring points. Background image shows the magnitude of the gradient map. The result is from a completed optimization, not a single iteration.

of points according to their distances

$$dx_i = M_g(x_i) \mid x_i \in X$$
$$w_{ij} = \exp(-\|x_i, x_j\|^2 / 2\sigma_n^2) \mid x_i, x_j \in X$$

where $\| \, \|$ is the Euclidean distance between a pair of points. The parameter σ_n determines the locality scope of the coherency condition, where $\sigma_n = 0$ means no coherency and $\sigma_n = \infty$ means strict coherency resulting in a rigid transformation (translation and rotation). The optimal value of σ_n depends on the size of the map and its deformity. We expect a neighborhood of roughly 8 meters for our collection of maps, based on empirical observation, and any value in the range of $\sigma_n = 8 \pm 4$ is acceptable. The optimization procedure, including the coherency condition, is presented in Alg. 1.

Optimization termination criteria: Apart from the $max\_iteration$ that safeguards the process against infinite loops, min_motion is the only termination criterion that is a lower bound for the motions in dX'. Suggested values for these parameters are $min\_motion = 10^{-3}$ (1 millimeter), and $max\_iteration = 10^4$. As we will see in Fig. 7a from Sec. IV, the optimization of $3/36$ alignments fails with these parameters. However, the reason is that they converge to local minima (starting from poor initial alignments), which suggests that they would not have succeeded even with different values of termination parameters. We do not base any criterion on the fitness values, as the maximum fitness by definition results in motionless points (i.e. $M_g = 0$). On the other hand, the fitness will not be maximized when points become motionless due to the local minima from a wrong initial alignment (i.e. $M_f = 0, M_g = 0$). When the process converges to such an equilibrium, the process should be terminated even though the equilibrium does not correspond to the optimal solution. Therefore fitness based criterion can be subsumed by min_motion.

IV. Experimental Results and Verification

This section presents the data that we collected for the verification of the method's performance. An experiment that

Algorithm 1 Optimization

function OPTIMIZE(M_g, M_f, $X_{N\times 2}$, $W_{N\times N}$)
  $X' = X$
  for $iteration \in \{1, 2, \ldots, max\_iteration\}$ **do**
    $dX = M_g(X')$
    $dX' =$ WEIGHTEDAVERAGE(dX, W)
    $X' = X' + dX'$
    if $\max(\|dx'\| | \forall dx' \in dX') < min\_motion$ **then**
      break
    end if
  end for
  return X'
end function

function WEIGHTEDAVERAGE($dX_{N\times 2}$, $W_{N\times N}$)
  /* $(A \circ B)$: "Hadamard-Schur" product */
  $P = [dX, dX, ..., dX]_{N\times N\times 2} \circ [W, W]_{N\times N\times 2}$
  $dX'_{N\times 2} = \text{mean}(P_{N\times N\times 2})_{\text{along } 2^{nd} \text{ dimension}}$
  return dX'
end function

shows a strong correlation between fitness and alignment success, is presented in Sec. IV-A. We present the performance of map alignment in comparison with other techniques in Sec. IV-B. Finally, we present a use case of transferring prior knowledge (region segmentation) in Sec. IV-C, that improves segmentation consistency over sensor maps.

Setting of the parameters: All the parameters were set the same for all the experiments, home and office alike, with theses values: $\sigma_f = 1$, $\sigma_n = 8$ and $min\_motion = 10^{-3}$, all in meter, and $max\_iteration = 10^4$.

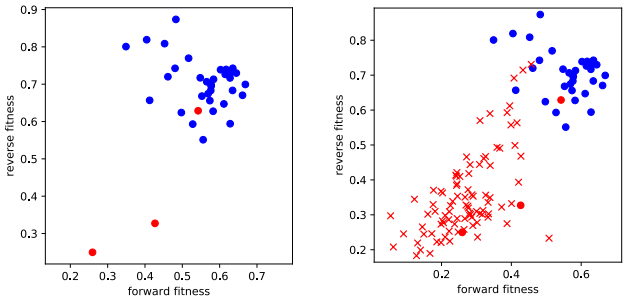
Data collection: We collected maps of four environments, two homes and two office buildings[2]. There are 36 sensor maps in total, 14 for each office and 4 for each home environment. Layout maps were obtained from CAD drawings, and there is a layout map for each environment. Sensor maps, most of them partial, were collected by a *Google Tango tablet* and the *Tango Constructor application* from Google. The 3D meshes were converted to occupancy-like maps through a ray-casting process. Due the absence of sensor's trajectory in 3D meshes, the locations of ray-casting are interactively chosen by the user. Each 3D mesh was sliced horizontally at different heights, to reflect the structural elements of the environment better, and avoid most of the overhanging objects (e.g. lamps) and clutters on the ground (e.g. chairs). A discrete representation that underlies the occupancy map, is constructed from a projection of most frequently sliced vertices.

A. Fitness and confidence metric

We define two variations of the fitness, namely *forward* and *reverse*, as an alignment quality measure

$$fitness := \text{mean}([M_f(x, y) \mid \forall (x, y) \in X]_{N\times 1})$$
$$forward: \quad M_f \leftarrow \text{layout map}, X \leftarrow \text{sensor map}$$
$$reverse: \quad M_f \leftarrow \text{sensor map}, X \leftarrow \text{layout map}$$

[2] https://github.com/saeedghsh/Halmstad-Robot-Maps/



(a) corresponding sensor/layout (b) all sensor to all layout

Fig. 7: The comparison of success and failure according to [forward and reverse] fitness. Blue and red markers represent the success and failure of the alignments. Circle and cross markers represent correct and wrong correspondence between sensor maps and layout maps. The failure and success classes are almost linearly separable by comparing our proposed forward and reverse fitness measure.

For the fitness function to better represent the quality of the alignment, M_f is computed with a stricter neighborhood of $\sigma_f = 0.1$ meter instead of that 1 ± 0.4 of the optimization process. This is because in the optimization process M_f requires a wider scope as it acts as a membership function of occupied cells, and underlies the gradient map. As an alignment quality metric, M_f evaluates the fitness of X with respect to the structure of the target map, and it is set narrower to penalize even minor deviations. Fig. 7a shows the fitness values of aligning each sensor map against their corresponding layout maps. Three failures are marked red. The one failing case that resides among successful points is a case were only one room from the source map ($\sim 10\%$ of the map) is stretched and covers two rooms in the layout, resulting in a *partial misalignment*. Fig. 7b also includes the fitness of aligning each sensor map with the layouts of *other* environments, marked with red *crosses*. The wrong alignments in the margin between success and failure, are cases where the sensor maps are from homes and they easily fit into sub-regions of office layouts. Despite these few degenerate cases, a strong correlation between success and fitness value can be observed.

B. Map alignment comparison

Our proposed optimization method assumes that the target map (layout) is globally consistent, and a super-set of the source map (sensor). This assumption cannot be guaranteed for sensor maps as target, and therefore our method is only viable for optimizing alignments of sensor maps to layout maps. On the other hand, most other map alignment techniques operate exclusively on sensor maps. Consequently, to establish a common ground for comparison, we use the decomposition based alignment [2] coupled with the proposed optimization method, and find the alignments of all sensor maps to their corresponding layout maps. Then a layout's frame of reference can be used as a link between sensor maps. The alignment of sensor maps to layout map, however, is not free of challenges. We have shown, in our previous work [2], the difficulties of most common map alignment approaches in dealing with maps of different types, scales and noise levels.

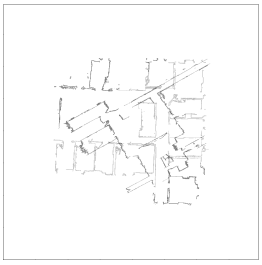
method	implementation	success rate (in %) home	office	total	error (meter) RMS	average time (and variance) in seconds home	office
Coherent Point Drift [16]	Python	0	6.04	5.6	10.02	NA	NA
Voronoi diagram-based [10]	Matlab	25.55	11.53	12.4	34.77	4.91(1.42)	50.20(19.84)
SIFT [17]	Python	8.33	23.07	22.1	124.8	0.20(0.05)	0.67(0.14)
Hough-based [8]	C++	91.66	23.07	27.31	13.06	$3.07e^{-4}(9.28e^{-5})$	$2.65e^{-4}(6.72e^{-5})$
ECC maximization [14]	Python	8.33	32.96	31.9	10.7	32.79(28.24)	73.46(85.46)
Decomposition-based [2]	Python	91.66	59.34	66.5	5.89	8.86(2.13)	41.86(41.92)
this work	Python	100	79.12	80.41	1.78	21.20(4.16)	49.63(20.62)

TABLE I: Success rates, RMS error, and computation times of different methods on aligning sensor maps. There are 182 and 12 pairs of sensor maps for two office buildings and two home environments respectively.

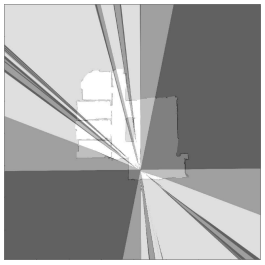
The experiment in this section compares the performance of the proposed map alignment through the layout map, with six other approaches. Three of these are image processing techniques adapted to the map alignment problem, namely i) image alignment with Enhanced Correlation Coefficient (ECC) Maximization [14], ii) image registration with Scale-Invariant Feature Transform (SIFT) [17] in combination with Fast Approximate Nearest Neighbors [25] for feature matching, and iii) treating each map as a set of occupancy points and employing the Coherent Point Drift (CPD) [16] for Point Set Registration. The other three are methods specifically designed for robot map alignment, namely i) map merging based on Hough-transform by Carpin [8], ii) map merging based on probabilistic generalized Voronoi diagram by Saeedi et al. (PGVD) [10], and iii) decomposition based map alignment from our previous work [2]. All methods work better and have been tested with M_o, except for ECC which achieved better results and has been tested using M_d.

The performance results are presented in Tab. I, where the success rate was measured by manually labeling successful alignments from visual inspection. For a more objective analysis, we annotated the maps with *key points* and their corresponding associations for measuring the accuracy of the alignments. The Euclidean distance between associated key points under an alignment is regarded as the error of the alignment. The Root Mean Square (RMS) errors of methods are presented in Tab. I, and Fig. 8d shows the distribution of this error for four of the best performing alignment methods. From these results, we note that the Hough-based method [8] is substantially faster than others, although compared to our method it has a lower success rate (27.31% vs. 80.41%) and a higher RMS error (13.06 vs. 1.78 meter). SIFT-based registration is similarly fast with low success rate. Closest to this work, in terms of success rate, is the decomposition-based method. Nevertheless, it still falls short in comparison to this work in terms of both success rate (66.5% vs. 80.41%) and RMS error (5.89 vs. 1.78 meter).

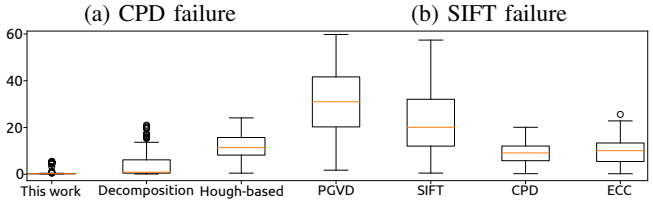
RMS error vs. success rate: It is important to note that the performance of each method must be evaluated with both the success rate and the RMS error. While success rate could be influenced by the subjective manner of visual inspection, RMS is also sensitive to the failure manner of each method. Different methods fail differently due to their different natures. For instance CPD will always keep the whole body of the source map inside the boundary of the target map, even in the failed cases as demonstrated in Fig. 8a. Failure of SIFT as is Fig. 8b, however, can easily return wild solutions, where control points are moved very far with no bounds. However, by visual inspection of all alignments we can see that SIFT is more successful than CPD ($\sim 22\%$ vs. $\sim 6\%$), while the RMS error from Fig. 8c suggests that CPD performs better than SIFT, due to failure type and bounded error of CPD. In conclusion, it should be noted that the RMS error can be misleading if the success rates of the methods are disregarded.



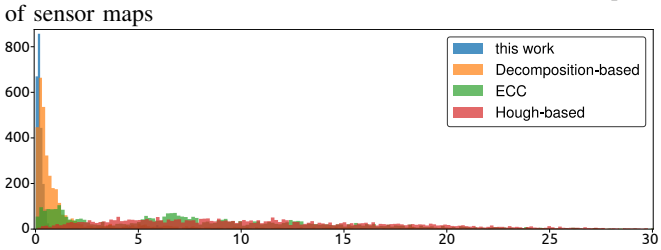
(a) CPD failure



(b) SIFT failure



(c) distribution of RMS error of each method for all the 194 pairs of sensor maps



(d) histogram distributions of sensor to sensor map alignments of four best performing methods (based on success rate)

Fig. 8: Error analysis of the alignment result. Figures. 8a and 8b compare the failures of CPD and SIFT methods in map alignment. While both are failed alignment, one has much higher impact on the RMS error. Fig. 8c shows the distribution of RMS error of each method for all the 194 pairs of sensor maps. Histogram distributions of all sensor to sensor map alignments of four best performing methods are presented in Fig. 8d. The error in Figures 8c and 8d is the Euclidean distance (in meters) between all pairs of associated key points from annotated ground truth.

Computation times: All the experiments were carried out on a computer with an Intel® Core™ i5-3340M CPU @ 2.70GHz ×4, and 8GiB SODIMM DDR3 Synchronous

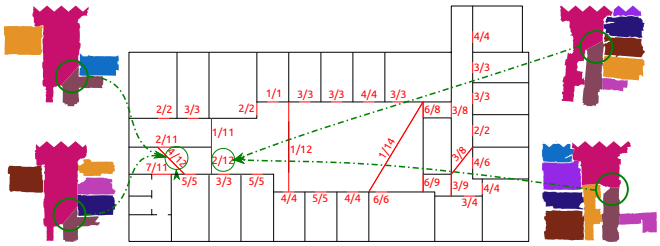


Fig. 9: "DuDe-2D" [28] is very robust in region segmentation, specifically for rooms. However, its results can be inconsistent over different noisy sensor maps. Segmented maps on the sides cover the same location, and they correspond to the left part of the layout (with 90° rotation). These results demonstrate the inconsistency of the region segmentation over the same corridor. The inconsistencies of border lines are quantified by a hit parameter (marked red in the layout), defined as the number of its appearance in segmentations divided by the number of sensor maps that cover this region.

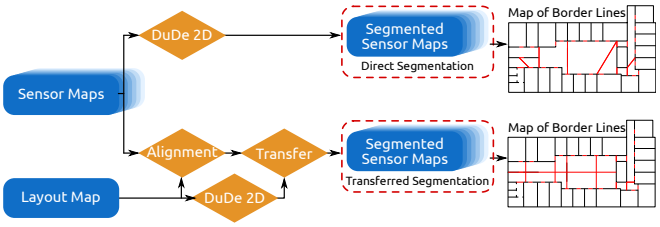


Fig. 10: Two ways of *direct* and *transferred* region segmentation. A list of border lines (marked red in the layout) is compiled for each method separately for measuring the consistency of each approach. They are compiled from the segmented sensor maps and not the segmented layout.

1600 MHz of memory, running Ubuntu 14.04. Our method has to perform two alignments for aligning one pair of sensor maps, which doubles its computation time. The time for each alignment is therefore the sum of decomposition-based alignment and optimization. The average time for *optimizing* one alignment is 1.31 seconds with a standard deviation of 1.08 (for both home and office maps). We replaced the original match score function from our previous work [2] with the fitness function in this work for hypothesis selection. This substitution yields better and faster results for sensor map to layout alignment, but performs worse for sensor map to sensor map. As this work is only concerned with sensor map to layout alignment, we benefit both in time and performance by this substitution. Please note that because of the different implementations used, the timings in Tab. I should only be taken as a rough indicator of relative computation time.

C. Segmentation transferring

Here we present an example of using alignments for transferring prior knowledge from layout to sensor maps, in which the prior knowledge is the region segmentation and regions' semantic labels. Fig. 1 shows the outline of transferring this knowledge from a layout map to the 2D sensor map and all the way to the 3D map, from which we obtained the 2D sensor maps.

State of the art in region segmentation: Bormann et al. [26] presented a very interesting review and comparison of the most common region segmentation methods, which covers four different approaches: morphological, distance transform-based, Voronoi graph-based, and feature-based. In a more recent work, Fermin-Leon et al. [27] has successfully applied the "Dual-Space Decomposition" (DuDe-2D) by Liu et al. [28] to robot maps. Region segmentation is often subjective, and the results vary depending on the definition of a region, the employed method and its settings. For instance, DuDe-2D [28], which has been shown to perform quite robustly on robot maps [27], results in an inconsistent segmentation with partially explored maps, as in Fig. 9.

Our strategy: In this work, we present a different *strategy* of region segmentation rather than a novel *technique*. As shown in Fig. 10, the idea is to perform region segmentation on the prior map (layout) and transferring the results to sensor maps through the estimated alignment. This approach has two main advantages, i) the sensitivity of the segmentation techniques to noise becomes irrelevant as they would operate on clean layout maps, ii) relying on a unique segmentation of the layout improves the region segmentation consistency across different sensor maps, regardless of their coverage and noise level. We use a region segmentation method to minimize human intervention. In a general setting, however, such information could be provided manually and as accurate and subjective as desired.

On the technical level, transferring the region segmentation could be achieved with different approaches, such as transforming region contours or boundary lines between regions from one map to the other. However, we noticed these approaches are not robust to noise and minor defects in sensor maps and alignments. Instead we realized the most robust approach is to detect *transition points* between regions from the Voronoi graphs of the layout map, transform them to the sensor map according to the alignment, and employ them as heuristic cues in junction with a morphological segmentation method [26]. The heuristic step works by padding the sensor map (M_o) with black disks (augmenting occupancy) at the position of each transition point with the radius being equal to the value from M_d at each transition point. This will enforce separation of regions based on prior knowledge, and fetching the radius from the distance map of the sensor map improves the robustness against noise.

Experiment: In the absence of a ground truth, the consistency, and not the quality, of region segmentation of the sensor maps is the object of comparison. To that end, for each segmentation approach, all border lines between segmented regions from each and all *sensor maps* are marked in the layout map, as in Fig. 9. Those border lines corresponding to the same segmentation (gauged by visual inspection) are grouped together. It should be noted that the segmentation of layout is irrelevant for measuring the consistency. That is to say, the border lines are compiled only from the segmentation of the sensor maps, and not the layout. A consistency measure is proposed based on the appearance consistency of these border lines. We define a hit number h for each border line as the number of its appearance in segmentations divided by the number of sensor maps that cover this region. For instance, consider a doorway that is covered in ten out

of twenty partial maps. A border line corresponding to that doorway has a hit value equal to $h = 0.8$, if it is segmented in eight maps. A border line is least consistent when it has a hit value of $h = 0.5$, and most consistent for a hit value of $h = 1$ and $h = 0$. Although, hit can never be zero, as $h = 0$ indicates that such a border line has never emerged in any of the segmentations. Accordingly we define the consistency of each border line as $c = |1 - 2h|$. The consistency of a region segmentation is defined as

$$consistency = \frac{1}{N} \sum_{i=1}^{N} |1 - 2h_i|$$

where N is the number of border lines, i.e. red lines in the layout map. According to this measure, direct region segmentation and transferring region segmentation are 81.56% and 95.97% consistent respectively.

V. Conclusion

In this work we present a method for optimizing a 2D alignment between a robot (sensor) map and a layout (blueprint) map. The optimization method proposed in this works fine-tunes an initial alignment, and simultaneously corrects potential deformations of the sensor map. The optimization of the alignment is achieved through an objective function that measures the alignment quality. Based on the assumption that the target map (layout) is globally consistent, and thanks to a non-linear transformation model, deformities of the sensor maps are also rectified through this process. The local consistency of the sensor map is maintained through the optimization by means of a coherency condition. We demonstrate that our method's result in aligning partial and deformed sensor maps to layout maps, could not be matched by any existing method. A simple and fast-to-compute fitness function is devised for the optimization, which is shown to strongly correlate with the quality the alignment. Finally we show an example of utilizing the optimized alignment for transferring prior knowledge, from the layout map to sensor map. For this example we employ a state-of-the-art region segmentation method for segmenting the layout map, and transfer the result to all aligned sensor maps. We show, through experimental results, that the consistency of the region segmentation could be improved by transferring the segmentation from the layout, in comparison to applying region segmentation directly on noisy sensor maps.

Future work: Assuming an initial alignment is provided, our method performs an optimization of that alignment based on only local information. An interesting feature would be to enable the method to measure the quality of the initial alignment on a structural level, so that the method becomes robust to errors in the initial alignment. We aim to detect and quantify errors in the initial alignment, which in turn requires the detection and quantification of errors in the maps. The motivation behind this feature rose from three failed optimization cases from Fig. 7a where the initial alignments were wrong. We are investigating means of incorporating the abstract models from our previous work [2] with the fitness measure presented in this work, into a unified framework of map and alignment quality measure.

References

[1] J. Lundström and e. al., *Halmstad Intelligent Home - Capabilities and Opportunities*. Springer International Publishing, 2016, pp. 9–15.

[2] S. Gholami Shahbandi and M. Magnusson, "2D map alignment with region decomposition," *arXiv*, 2017, under review for Autonomous Robots of Springer, available at https://arxiv.org/abs/1709.00309.

[3] W. H. Huang and K. R. Beevers, "Topological map merging," *IJRR*, vol. 24, no. 8, pp. 601–613, 2005.

[4] J. O. Wallgrün, "Voronoi graph matching for robot localization and mapping," in *TCS IX*. Springer Berlin Heidelberg, 2010, pp. 76–108.

[5] S. Schwertfeger and A. Birk, "Evaluation of map quality by matching and scoring high-level, topological map structures," in *IEEE ICRA*, May 2013, pp. 2221–2226.

[6] M. Mielle and e. al., "Using sketch-maps for robot navigation: Interpretation and matching," in *IEEE ISSSRR*, Oct 2016, pp. 252–257.

[7] D. Kakuma and e. al., "Alignment of occupancy grid and floor maps using graph matching," in *IEEE ICSC*, Jan 2017, pp. 57–60.

[8] S. Carpin, "Fast and accurate map merging for multi-robot systems," *Autonomous Robots*, vol. 25, no. 3, pp. 305–316, 2008.

[9] M. Bosse and R. Zlot, "Map matching and data association for large-scale two-dimensional laser scan-based slam," *IJRR*, vol. 27, no. 6, pp. 667–691, 2008.

[10] S. Saeedi and e. al., "Efficient map merging using a probabilistic generalized Voronoi diagram," in *IEEE/RSJ IROS*, Oct 2012, pp. 4419–4424.

[11] J. Park and e. al., "Map merging of rotated, corrupted, and different scale maps using rectangular features," in *IEEE/ION PLNS*, April 2016, pp. 535–543.

[12] T. M. Bonanni and e. al., "3D map merging on pose graphs," *IEEE RA-L*, vol. 2, no. 2, pp. 1031–1038, April 2017.

[13] S. Baker and I. Matthews, "Lucas-kanade 20 years on: A unifying framework," *IJCV*, vol. 56, no. 3, pp. 221–255, Feb 2004.

[14] G. D. Evangelidis and E. Z. Psarakis, "Parametric image alignment using enhanced correlation coefficient maximization," *IEEE PAMI*, vol. 30, no. 10, pp. 1858–1865, Oct 2008.

[15] P. J. Besl and N. D. McKay, "A method for registration of 3-d shapes," *IEEE PAMI*, vol. 14, no. 2, pp. 239–256, Feb 1992.

[16] A. Myronenko and e. al., "Non-rigid point set registration: Coherent point drift," in *ANIPS 19*. MIT Press, 2007, pp. 1009–1016.

[17] D. G. Lowe, "Object recognition from local scale-invariant features," in *IEEE ICCV*, vol. 2, 1999, pp. 1150–1157 vol.2.

[18] T. F. Cootes and e. al., "Active shape models: Their training and application," *CVIU*, vol. 61, no. 1, pp. 38–59, Jan. 1995.

[19] G. J. Edwards and e. al., "Interpreting face images using active appearance models," in *IEEE ICAFGR*, Apr 1998, pp. 300–305.

[20] T. W. Sederberg and S. R. Parry, "Free-form deformation of solid geometric models," *SIGGRAPH Comput. Graph.*, vol. 20, no. 4, pp. 151–160, Aug. 1986.

[21] G. K. Rohde and e. al., "The adaptive bases algorithm for intensity-based nonrigid image registration," *IEEE TMI*, vol. 22, no. 11, pp. 1470–1479, Nov 2003.

[22] W. R. Crum and e. al., "Non-rigid image registration: theory and practice," *BJR*, vol. 77, no. suppl_2, pp. S140–S153, 2004, pMID: 15677356.

[23] Kitware Inc., "The Insight Segmentation and Registration Toolkit."

[24] J. Shi and C. Tomasi, "Good features to track," in *IEEE CVPR*, Jun 1994, pp. 593–600.

[25] M. Muja and D. G. Lowe, "Fast approximate nearest neighbors with automatic algorithm configuration," in *ICCVTA*, 2009, pp. 331–340.

[26] R. Bormann, F. Jordan, W. Li, J. Hampp, and M. Hägele, "Room segmentation: Survey, implementation, and analysis," in *IEEE ICRA*, May 2016, pp. 1019–1026.

[27] L. Fermin-Leon and e. al., "Incremental contour-based topological segmentation for robot exploration," in *IEEE ICRA*, May 2017, pp. 2554–2561.

[28] G. Liu and e. al., "Dual-space decomposition of 2d complex shapes," in *IEEE CVPR*, June 2014, pp. 4154–4161.