

Matching Paths in Topological Maps

Sören Schwertfeger* Tianyan Yu**

* *ShanghaiTech University, No. 8 Building, 319 Yueyang Road, Shanghai 200031, China (e-mail: soerensch@shanghaitech.edu.cn)*

** *ShanghaiTech University, (e-mail: yuty@shanghaitech.edu.cn).*

Abstract: Topological maps have many applications in robotics. Matching two topological maps from the same environment can be used for map merging, place detection, map evaluation and other purposes. In this paper we present an approach to match two corresponding edges from two Topology Graphs to each other based on the actual path with which the vertices of the edges are connected in the underlying 2D grid maps. We perform experiments with two artificial maps as well as with four maps from the RoboCup Rescue WorldCup 2010.

Keywords: Mobile robots, Mapping, Topological map, Path matching, Voronoi graph

1. INTRODUCTION

In many robotic applications mapping is an essential task for mobile robot systems. The generated maps are models of the environment which are often represented as 2D grid maps. This image like map format is quite detailed. Topology Graphs are more abstract representations which only comprise of places and connections between them.

There are many applications for topological maps, for example for map merging Saeedi et al. (2014), place detection Beeson et al. (2005), or planning Thrun (1998). There are also different ways to generate topological representations from 2D grid maps, for example based on thinning methods Ko et al. (2004) or Voronoi Diagrams Kolling and Carpin (2008), Lau et al. (2010).

All maps have some degree of error which should be measured Schwertfeger et al. (2011). Recent work on map quality assessment matches the topology graph of a ground truth map to the topology graph of robot generated maps Schwertfeger and Birk (2015a), Schwertfeger and Birk (2013). Those algorithms base the matching on a similarity metric of the vertices and a common sub-graph isomorphism. In this work we explore the possibility to match the edges of the Topology Graphs instead of the vertices. The resulting edge matching is useful not only for map evaluation but also to other applications like map merging or place recognition.

This paper is structured as follows: Section 2 provides a short review of related work while Section 3 introduces definitions and the algorithm. The experimental validation is presented in Section 4 and is followed by the conclusions.

2. RELATED WORK

Previous work on path or skeleton matching mainly follows three different approaches:

- Map matching with shortest geodesic paths
- Contour partitioning and skeleton pruning
- Shape matching using Bayesian formula with skeleton similarity

In Bai and Latecki (2008), only the endpoints of a path are considered. As the endpoints are always on the contour, they are indexed in a clockwise orientation. The least distances through the skeleton path for each endpoint pair, which is called "shortest geodesic path", are calculated, and vertices are sampled from the path equidistantly. A normalized vector is used to save the radii of each sampled vertex's maximal disc. With this vector, the similarity between two paths can be calculated. For skeleton matching, the similarity matrix between a certain vertex v_i from map A and another vertex v'_j from map B is calculated by going through all shortest geodesic paths which are connected to the vertex v_i from map A and calculating the similarity with all shortest geodesic paths which are connected to the vertex v'_j from map B.

By applying optimal subsequence bijection (OSB), some outlier endpoints are filtered out, and the dissimilarity between two endpoints are obtained. A dissimilarity matrix is then generated by going through all endpoints in both map A and map B. By using the Hungarian algorithm, each endpoint from map A finds its best match in map B. It has to be noted that in this approach the matching of connection vertices which are connected to 2 edges and the junction vertices which are connected to 3 or more edges are not taken into account because the endpoint which is connected to only 1 edge is always on the contour, which saves a lot of information about the shape of the object and this is very important for matching.

If applied to the figures like horses or humans, whose skeletons have many branches on the contour, and each vertex on the algorithm finds the shortest geodesic paths to the other on the contour, the approach works well. For robot maps, there may be very few dead-end vertices on the contour, and the paths in the maze-like map are also complicated. So the shortest geodesic path between vertices on the contour seems not suitable for robotic map matching.

In their another approach Bai et al. (2007), skeleton maps are used for contour partition. The Discrete Curve Evolution (DCE) Latecki and Lakämper (2000) is applied to the

contours of objects in digital images. It uses the contour pixels to form a polygon and each time a contour vertex with the least contribution to the contour is removed. At the same time, skeleton pruning is carried out, in which only the skeleton points whose generating points belong to different contour partitions are reserved. By setting a certain threshold to stop the pruning iteration and removing concave vertex on the contour and the skeleton edge connected to it, several main skeleton edges and the partition of the contour are obtained.

Since only the main part of the skeleton map and its convex hull (in the form of polygon) is preserved, this approach can be further used for shape classification (by matching the pruned skeleton map).

In this approach, the figure always has a contour (such as leaf), contour partitioning and skeleton pruning is used to get the main skeleton of it. In contrast, the skeleton of robotic maps is much more complicated. Some paths are in a room, others connect one room to another. This approach can only be applied to the paths in a certain room in the map, since room has a contour, but it can not be applied on the whole map.

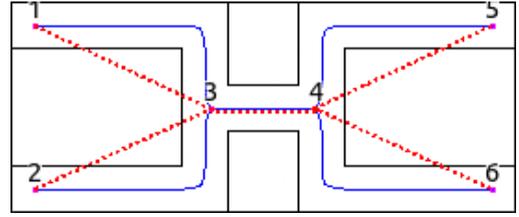
Yang and Sze (2007) use the longest weighted path in a directed acyclic graph to match two graphs, and find the top K suboptimal paths in polynomial time. This approach is used in biology, and matches the abstract graph and paths.

3. ALGORITHMS

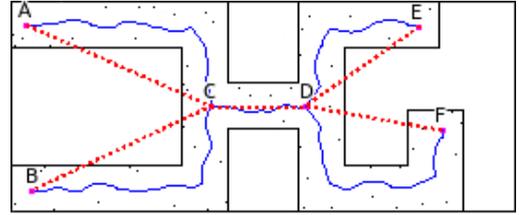
First we define a terminology for the following sections.

- **Graph:** A graph is an ordered pair $\mathbf{G} = (\mathbf{V}, \mathbf{E})$ comprising a set \mathbf{V} of vertices or nodes together with a set \mathbf{E} of edges or links.
- **Topology Graph:** A graph in which vertices represent locations and edges which represent the fact that there exists a drivable route between two vertices.
- **Vertex:** A node in the graph. It is attributed with the metric location as x, y coordinates.
- **Edge:** A (drivable) connection between vertices in the graph. Each edge is attributed with a metric Path.
 - Half Edge: In a Doubly Connected Edge List, a half edge is a directed connection between two vertices.
 - Twin: Each half edge has one twin. A half edge's source vertex is its twin's target vertex and vice versa.
- **Path:** Every edge is attributed with exactly one path. The path represents the metric information of a free (drivable) way between two vertices. A path is a small directed graph consisting of a series of edges which are connected in a (possibly curved) line between the source vertex and the target vertex of the parent edge. Each vertex in the path is attributed with its metric x, y coordinates.

The path matching works on the Topology Graphs introduced in detail in Schwertfeger and Birk (2015a) and Schwertfeger and Birk (2013). There, a Voronoi Diagram is used to generate a topological graph from a 2D grid map. After some filtering of vertices and edges, the topological graph can directly be used to match two Topology Graphs



(a) Map I: Artificial ground truth map.



(b) Map II: Artificial map with man-made noise and difference around E and F.

Fig. 1. Artificial 2D grid maps (black) with Topology Graphs (labeled vertices: purple, edges: red dotted, paths attributed to edges: blue)

with each other. In that work the matching was based on similarity metric of the vertices and a common sub-graph isomorphism. In contrast, this paper explores the matching based on the edges of a topology graph, or more precisely based on the path attributed to each edge.

Instead of using a Voronoi Diagram other methods of generating the Topology Graph are possible, for example thinning methods Ko et al. (2004). The Topology Graph generation used here works best in relatively confined environments. In open areas graphs often differ between different maps and it is thus difficult to match those graphs. Our work on generating Topology Graphs which represent such open areas as vertices in the graph ("room detection") and thus alleviate this problem will be published soon. Also it should be noted that the paths used here are not paths that the robot actually traversed. Our edges/ paths are connections between vertices have been found to be possible to drive for a robot - by evaluating the metric grid maps that are the sole input to this algorithm.

Please observe the definition: An edge is a topological graph element directly connecting two vertices ("in a straight line"), while the path is a metric attribute of an edge which forms (an often curved) line in the free space of the environment between the two vertices of the edge. For example in Figure 1(a) there is an edge between vertices 1 and 3 (red dotted line) and a curved path attributed to this edge (blue line). In the other figures the red dotted edges are omitted. Also note that the topology of the graph is undirected, but the graph is represented using directed half edges and each half edge has a corresponding twin in which the source and target are switched.

3.1 Path Matching

In the following algorithms we assess the dissimilarity of two paths and thus also the dissimilarity of their associated edges. The path dissimilarity is calculated by sampling each path equidistantly. The result is a list of (x, y) 2D coordinates. After sampling, the 2D Horn's algorithm

Horn (1987) is applied in order to find the best rotation, translation and scaling between the vertices sampled from two different paths that minimizes the error between them. The $L2$ Norm is used to define the error. For each path from the map generated by the robot, the least-error path in the ground truth map is found, which is recognized as a match.

Generally, since the matching result will be used in later stages of the map matching algorithm, within a topological graph, the long and curvy paths are more distinctive than the straight and short ones. Thus we are filtering out short and straight paths (which are quite common) and concentrate on the longer and more curvy paths.

3.2 Full Path Comparison

When matching two maps A and B we generate the two corresponding Topology Graphs $g(A)$ and $g(B)$. The edges in the graphs are all attributed with one path each.

Each path is cut into $N - 1$ pieces, and each one has the same length of $1/N$ of the path's whole length. In total we will have sampled N points per path (including source and target location). We employ 2D Horn's for calculating the similarity between two paths K and L . Horn's works with matched point lists. So the i -th point K_i from K is matched to the i -th point L_i from L for all points from $i = 0$ to $i = N$. The $L2$ norm as calculated by Horn's algorithm is the residual error and then used as our dissimilarity value.

For the paths of all half edges of $g(A)$ we calculate the dissimilarity to the paths of all half edges of $g(B)$, resulting in a matrix with I rows (I is the number of edges in $g(A)$) and J columns (J is the number of edges in $g(B)$).

From each row in this table, a minimal value of the $L2$ norm and the corresponding path in map $g(B)$ can be found so that each path from $g(A)$ can be matched to a certain path in B .

3.3 Partial Path Comparison

There can be paths in $g(A)$ that are shorter or longer than the corresponding path in $g(B)$ because the map is only similar around one of the vertices but differs in the second. In this case, if the $L2$ norm of Section 3.2 is used, most likely the corresponding path cannot be found.

To avoid this, the difference between the lengths of a path pair can be calculated by dividing the length of the longer path by the length of the shorter path.

In the partial path comparison $1/N$ of the length of the shorter path is used to cut both paths for sampling the vertices. That means only the subpath starting from the source of the longer path with the same length of the shorter path is used for matching. After sampling, the 2D Horn's method is used to calculate the partial $L2$ norm. Again a matrix with the dissimilarity values of all edges between the two maps is created.

3.4 Filtering

Straight paths always have a very low $L2$ norm in comparison and are thus not very discriminative. Also paths which

Path	Length	Curviness	Path	Length	Curviness
1→3	44.9	1.00	18→20	64.2	0.80
2→2	110.7	0.00	19→21	45.2	0.81
2→3	95.3	0.65	20→24	93.8	0.54
2→9	63.3	0.97	21→22	37.0	0.96
3→5	45.4	0.96	23→24	26.0	0.98
4→5	76.4	0.99	24→27	79.4	0.90
4→6	24.8	0.99	25→30	99.2	0.97
4→7	71.8	0.60	26→32	78.2	0.71
5→13	39.4	0.97	27→29	25.8	0.98
8→9	48.8	0.99	27→30	38.1	1.00
9→21	117.6	0.78	28→32	41.6	1.00
10→15	32.6	0.96	30→31	11.3	1.00
11→20	69.0	0.98	30→33	41.7	0.96
12→15	31.4	0.88	32→33	178.9	0.43
13→16	16.2	1.00	32→35	93.1	0.85
13→17	89.4	0.85	33→34	20.3	0.99
14→16	19.8	1.00	35→36	24.5	0.98
15→17	12.9	1.00	35→38	24.1	0.96
16→19	59.5	0.78	36→37	40.0	0.66
17→18	64.2	0.47	36→39	23.2	0.99
18→19	139.1	0.82			

Table 1. Length and curviness of edge paths from the ground truth Map III (Figure 2(a))

Path	Length	Curviness	Path	Length	Curviness
1→3	134	0.79	AC	138	0.79
2→3	133	0.79	BC	139	0.77
3→4	57	1.00	CD	53	0.96
4→5	134	0.79	DE	97	0.77
4→6	133	0.79	DF	132	0.57

Table 2. Length and curviness of the edge paths from the artificial maps. (Figure 1)

are very short are typically very similar. That is why we filter short paths out before applying the path comparison.

In human made environments we often have straight paths. Straight paths cannot be differentiated by their shape and we thus are also filtering out paths which have a low curviness. The curviness c of a path is calculated by dividing the length of the geometric distance d between the source and the target vertices of an edge by the length l along the path. Since we work in a planar 2D geometry the following always holds: $d \leq l$. Thus $c = \frac{d}{l}$ is guaranteed to always be between 1 (straight) and 0 (very curvy).

4. EXPERIMENTS

We do experiments with two sets of maps. Figure 1 features Map I and II which were artificially created to showcase certain features of the algorithms. Maps III through VI (Figure 2) in contrast are real world maps created during the RoboCup Rescue Worldcup 2010 in Singapore Jacoff et al. (2012). They were created using systems described in Pellenz and Paulus (2010), Milstein et al. (2011) and Kleiner and Dornhege (2011). Those maps are also used in Schwertfeger (2012) for map evaluation.

Table 1 shows the length and curviness of the ground truth Map III (Figure 2(a)).

4.1 Artificial maps

We created a ground truth Map I (Figure 1(a)) and a modified Map II to showcase some of the features of the path dissimilarity. We added some noise to Map II to show the feasibility of the approach with imperfect data and also changed the topology on the right side. The top right part of Map II is shortened to highlight the properties of the partial path matching while the bottom right part has approximately the same length with only the path being

bend. In the following experiments we always sample 100 points per path (N in Section 3 is set to 100).

Please note that we use Arabic numbers for the vertices in the ground truth map and upper case letters for the vertices in the other maps.

Table 2 shows the length and curviness for Map I and Map II. Note that only one of the twins is included in the table - the other twin has the exact same values. The curviness values correspond to the expectations: $3 \rightarrow 4$ are perfectly straight (value 1) while CD is only close to 1 (0.96) because of the noise. DF is much more curvy (0.57) than the other paths (about 0.79).

The first fact to notice when looking at the dissimilarity matrix (Table 3) of the paths of Map I and II is that there are certain symmetries. The shape of path AC matches $1 \rightarrow 3$ as well as $6 \rightarrow 4$ (when rotated 180°). The path similarity is calculated in a rotation independent way. This is then also reflected in the results in Table 3 where AC has very low dissimilarities for those two paths and actually matched to the true one.

For DE and ED as well as DF and FD we can observe significantly higher dissimilarities because their path shape differs from the ground truth map.

For the partial path comparison only the path up to the length of the shorter of the two paths are compared - starting from the source vertex. We can note that especially the values for CD, DC as well as the corresponding $3 \rightarrow 4$, $4 \rightarrow 3$ have very good matches to other paths. This is because only a very short part of the paths is compared - due to the short length of the center edge (about 40%). For the further discussion we thus ignore those four paths.

The advantage of the partial path comparison is that DE can be matched better to the symmetric matches $3 \rightarrow 2$ and $4 \rightarrow 5$ when compared to the full path matching (dissimilarity value L2 of 1.98 versus 4.76). The twin ED however does not match well in any case (7.82 partial and 4.77 full) - which is expected because it starts at the shortened end of the path.

In summary we have shown that the dissimilarity value L2 works quite well - it matches the incompatible path shape DF, FD very badly when compared to the compatible paths AC, CA and BC, CB (L2 of about 6.8 versus < 3.0). This is true even though there is considerable amount of noise in Map II. Furthermore we have shown that the partial map comparison has better results for DE but not ED. We have also given good justification for filtering out short and straight edges.

4.2 RoboCup Rescue Maps

Figure 2 shows the maps used for the experiments with real robot generated maps. They are from the RoboCup Rescue WorldCup 2010 in Singapore. Map III is the ground truth map to which we try to match the other three maps.

Because of the limited size of this paper we cannot show experiments on more maps or include the full dissimilarity matrices here. Those matrices are available in the dataset on <http://robotics.shanghaitech.edu.cn/static/data/IAV2016/dataset.pdf>. In the experi-

	1→3	3→1	2→3	3→2	3→4	4→3	4→5	5→4	4→6	6→4
AC	2.01	16.40	15.27	9.41	3.85	3.85	9.33	15.01	16.65	2.20
CA	16.38	2.01	9.41	15.29	3.85	3.85	15.03	9.33	2.20	16.63
BC	15.86	10.12	1.20	17.49	4.41	4.41	17.23	1.11	10.18	16.11
CB	10.12	15.88	17.48	1.20	4.41	4.41	1.10	17.2	16.13	10.18
CD	11.41	7.33	7.79	12.10	1.42	1.42	11.87	7.61	7.51	11.65
DC	7.31	11.42	12.08	7.80	1.42	1.42	7.62	11.85	11.66	7.49
DE	7.61	17.84	18.85	4.76	4.92	4.92	4.85	18.58	18.10	7.58
ED	17.83	7.61	4.77	18.88	4.93	4.93	18.61	4.86	7.58	18.09
DF	21.41	6.94	7.84	21.83	6.91	6.91	21.56	7.86	6.78	21.70
FD	6.95	21.43	21.81	7.84	6.91	6.91	7.86	21.54	21.72	6.79
AC	2.98	15.55	14.64	9.40	1.12	1.12	9.29	14.23	15.97	2.99
CA	16.59	1.78	9.44	15.55	3.82	3.82	15.37	9.38	1.99	16.80
BC	14.74	10.05	2.15	16.52	1.11	1.11	16.09	2.15	10.16	15.16
CB	10.05	16.56	18.04	1.16	2.73	2.73	1.30	17.88	16.71	10.11
CD	1.33	1.89	1.33	2.57	1.33	1.33	2.81	1.33	1.80	1.33
DC	1.33	3.12	1.33	1.44	1.33	1.33	1.55	1.33	2.89	1.33
DE	7.90	14.86	8.91	1.98	2.98	2.98	1.91	8.91	14.95	7.90
ED	8.87	3.52	7.82	15.17	1.50	1.50	15.11	7.82	3.35	8.87
DF	21.16	6.80	7.85	21.63	2.00	2.00	21.46	7.86	6.49	21.16
FD	7.08	21.34	21.32	7.66	5.06	5.06	7.77	21.31	21.53	7.08

Table 3. L2 norm for the full path comparison between Map I and Map II at the top and for the partial path comparison at the bottom.

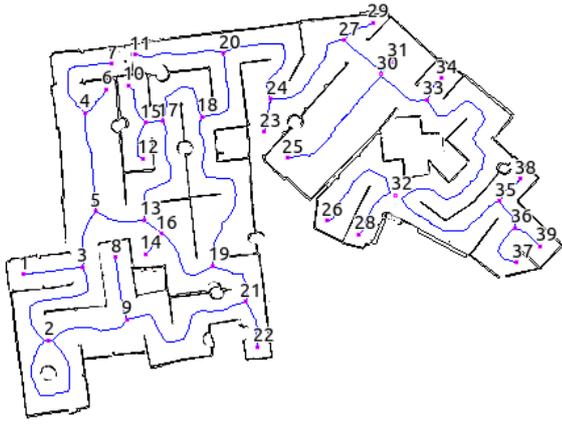
ments we applied the filter: Paths shorter than 40 are filtered out as well as paths with a curviness of higher than 0.95(for Map IV) and 0.9(for Map V and VI).

The following tables show the two best matches of the unfiltered path from the robot maps (Map IV to VI) to the ground truth Map III and the actual corresponding path in Map III. The robot maps are smaller and in certain areas also different (erroneous) from Map III. This is why there are paths in the robot map that don't have a correspondence in the ground truth map. For those no match nor similarity is given in the tables. For paths which are filtered in Map III, but where the corresponding path is not filtered out in the robot map, no similarity value for the correct (but unfindable) match is given.

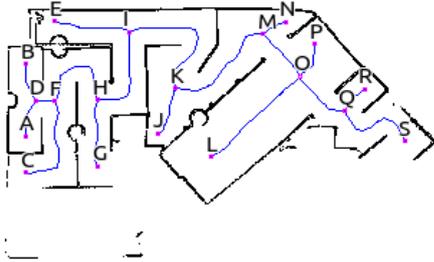
In Table 4 we can see the best matches of Map IV with the ground truth Map III. We get 66% correct matches with the full path comparison and 83% correct matches with the partial path comparison. For the full path matching FC, SQ, and their twins are mismatched. However, in the partial matching only 2 out of 12 paths are mismatched. Only CF, SQ are mismatched, but their twins match correctly. It can be noticed that in both approaches all the mismatched paths are parts of certain long paths in the ground truth map, which do not exist in the other map due to mapping errors. The partial matching approach matches with the correctly represented starting vertices, but of course fails for the other direction.

If we apply a hard threshold on the resulting dissimilarity (L2) of lower than 0.7 we can achieve a 100% true positive rate and 0 false negatives for both approaches without filtering away any good match.

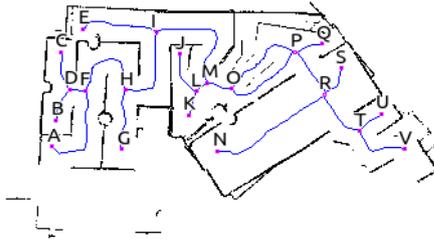
The results for the full path matching for Map V are shown in Table 5. Path TV, FA, PO(there are two edges/ paths between P and O - up and down) and their twins are matched wrongly but differ significantly from the ground truth due to errors in the map. IM and its twin are also wrong - the correct match is the 2nd best match which is very close in dissimilarity value. The result for the partial matching is similar. When applying the 0.7 threshold to the full matching method we get 100% true positive matches and 0 false negatives. For the partial method we get a 80% true positive rate with 0 false negatives.



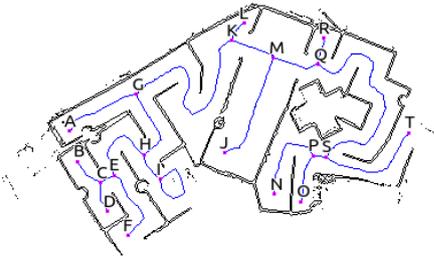
(a) Map III: Ground truth map.



(b) Map IV



(c) Map V



(d) Map VI

Fig. 2. RoboCup 2D grid maps (black) with Topology Graphs (labeled vertices purple, edges blue)

Map VI covers the most area of all three robot maps and has thus also the most number of edges (Table 6). EF and its twin are incorrectly matched because this path is much longer in the ground truth map. It has to be noticed that vertex 24 from Map III has no correspondence in Map VI, such that GK and KG are not correctly matched in the full approach. However, in the partial approach the beginning of GK is correctly matched to $24 \rightarrow 20$! SQ and its twin are correctly matched in the full approach, but in the partial approach much shorter paths get better results. We are

	Best L2	Best Edge	2nd Best L2	2nd Best Edge	Correct L2	Correct Edge
HF	0.27	18 \rightarrow 17	2.17	36 \rightarrow 37	0.27	18 \rightarrow 17
FH	0.27	17 \rightarrow 18	2.17	37 \rightarrow 36	0.27	17 \rightarrow 18
HI	0.24	18 \rightarrow 20	1.09	21 \rightarrow 19	0.237	18 \rightarrow 20
IH	0.24	20 \rightarrow 18	1.09	19 \rightarrow 21	0.236	20 \rightarrow 18
CF	0.78	18 \rightarrow 20	1.27	21 \rightarrow 19	4.62	13 \rightarrow 17
FC	0.78	20 \rightarrow 18	1.27	19 \rightarrow 21	4.61	17 \rightarrow 13
QS	3.28	19 \rightarrow 21	3.81	27 \rightarrow 24	16.1	33 \rightarrow 32
SQ	3.28	21 \rightarrow 19	3.81	24 \rightarrow 27	16.1	32 \rightarrow 33
IK	0.22	20 \rightarrow 24	1.89	17 \rightarrow 18	0.22	20 \rightarrow 24
KI	0.22	24 \rightarrow 20	1.89	18 \rightarrow 17	0.22	24 \rightarrow 20
MK	0.17	27 \rightarrow 24	2.00	19 \rightarrow 21	0.17	27 \rightarrow 24
KM	0.17	24 \rightarrow 27	1.89	21 \rightarrow 19	0.17	24 \rightarrow 27
HF	0.39	18 \rightarrow 17	0.89	36 \rightarrow 37	0.39	18 \rightarrow 17
FH	0.27	17 \rightarrow 18	0.96	19 \rightarrow 21	0.27	17 \rightarrow 18
HI	0.32	18 \rightarrow 20	0.84	36 \rightarrow 37	0.32	18 \rightarrow 20
IH	0.34	20 \rightarrow 18	1.07	20 \rightarrow 24	0.34	20 \rightarrow 18
CF	0.98	36 \rightarrow 37	1.32	18 \rightarrow 20	5.96	13 \rightarrow 17
FC	0.68	17 \rightarrow 13	1.48	26 \rightarrow 32	0.68	17 \rightarrow 13
QS	0.26	33 \rightarrow 32	0.98	21 \rightarrow 19	0.26	33 \rightarrow 32
SQ	0.79	21 \rightarrow 19	0.93	36 \rightarrow 37	6.36	32 \rightarrow 33
IK	0.66	20 \rightarrow 24	0.90	20 \rightarrow 18	0.66	20 \rightarrow 24
KI	0.37	24 \rightarrow 20	1.61	16 \rightarrow 19	0.37	24 \rightarrow 20
MK	0.13	27 \rightarrow 24	2.69	21 \rightarrow 19	0.13	27 \rightarrow 24
KM	0.27	24 \rightarrow 27	1.06	21 \rightarrow 19	0.27	24 \rightarrow 27

Table 4. Result of paths from Map IV to Map III. Full path: top; partial path: bottom.

	Best L2	Best Edge	2nd Best L2	2nd Best Edge	Correct L2	Correct Edge
HF	0.21	18 \rightarrow 17	2.11	36 \rightarrow 37	0.21	18 \rightarrow 17
FH	0.22	17 \rightarrow 18	2.11	37 \rightarrow 36	0.22	17 \rightarrow 18
IH	0.47	20 \rightarrow 18	1.03	19 \rightarrow 21	0.47	20 \rightarrow 18
HI	0.47	18 \rightarrow 20	1.03	21 \rightarrow 19	0.47	18 \rightarrow 20
TV	0.82	21 \rightarrow 19	1.25	18 \rightarrow 20	23.5	33 \rightarrow 32
VT	0.82	19 \rightarrow 21	1.25	20 \rightarrow 18	23.6	32 \rightarrow 33
FA	1.03	19 \rightarrow 21	1.05	26 \rightarrow 32	4.57	17 \rightarrow 13
AF	1.03	21 \rightarrow 19	1.07	32 \rightarrow 26	4.59	13 \rightarrow 17
MI	1.65	7 \rightarrow 4	1.69	24 \rightarrow 20	1.69	24 \rightarrow 20
IM	1.65	4 \rightarrow 7	1.69	20 \rightarrow 24	1.69	20 \rightarrow 24
PO(up)	2.21	21 \rightarrow 19	2.46	18 \rightarrow 20	/	/
OP(up)	2.22	19 \rightarrow 21	2.45	20 \rightarrow 18	/	/
PO(down)	1.97	19 \rightarrow 21	2.37	20 \rightarrow 18	/	/
OP(down)	1.97	21 \rightarrow 19	2.38	18 \rightarrow 20	/	/
HF	0.15	18 \rightarrow 17	0.66	36 \rightarrow 37	0.15	18 \rightarrow 17
FH	0.47	17 \rightarrow 18	0.74	19 \rightarrow 21	0.47	17 \rightarrow 18
IH	0.33	20 \rightarrow 18	1.05	20 \rightarrow 24	0.33	20 \rightarrow 18
HI	0.56	18 \rightarrow 20	0.78	36 \rightarrow 37	0.56	18 \rightarrow 20
TV	0.90	21 \rightarrow 19	1.02	33 \rightarrow 32	1.02	33 \rightarrow 32
VT	0.68	19 \rightarrow 21	1.36	17 \rightarrow 18	2.23	32 \rightarrow 33
FA	1.06	20 \rightarrow 18	1.31	26 \rightarrow 32	1.70	17 \rightarrow 13
AF	0.93	32 \rightarrow 26	1.40	36 \rightarrow 37	5.63	13 \rightarrow 17
MI	0.96	16 \rightarrow 19	1.71	24 \rightarrow 20	1.71	24 \rightarrow 20
IM	0.86	20 \rightarrow 18	1.95	20 \rightarrow 24	1.95	20 \rightarrow 24
PO(up)	0.852	21 \rightarrow 19	1.88	36 \rightarrow 37	/	/
OP(up)	1.29	19 \rightarrow 21	1.36	17 \rightarrow 13	/	/
PO(down)	1.13	26 \rightarrow 32	1.77	17 \rightarrow 13	/	/
OP(down)	1.59	21 \rightarrow 19	1.77	32 \rightarrow 35	/	/

Table 5. Result of paths from Map V to Map III. Full path: top; partial path: bottom.

thus thinking about adding a certain penalty for too big length differences.

When applying the 0.7 threshold we get a 80% true positive rate with 2 false positives and 2 false negatives for the full approach and a 100% true positive rate and 0 false negatives for the partial approach.

The runtime of the algorithm is dominated by the generation of the Topology Graph. Including the Topology Graph generation the calculation of all path similarities between any of the experiment maps takes less than a second (Map III to Map IV: 0.741 seconds).

5. CONCLUSIONS

In this paper we introduced an algorithm to determine the dissimilarity of two paths in Topology Graphs. The

	Best L2	Best Edge	2nd Best L2	2nd Best Edge	Correct L2	Correct Edge
<i>HE</i>	0.17	18 → 17	2.13	36 → 37	0.17	18 → 17
<i>EH</i>	0.17	17 → 18	2.13	37 → 36	0.17	17 → 18
<i>GH</i>	0.28	20 → 18	1.11	19 → 21	0.28	20 → 18
<i>HG</i>	0.28	18 → 20	1.10	21 → 19	0.28	18 → 20
<i>PN</i>	0.79	32 → 26	1.30	21 → 19	0.79	32 → 26
<i>NP</i>	0.76	26 → 32	1.30	19 → 21	0.76	26 → 32
<i>SQ</i>	0.45	32 → 33	2.07	36 → 37	0.45	32 → 33
<i>QS</i>	0.37	33 → 32	2.06	37 → 36	0.37	33 → 32
<i>EF</i>	0.66	20 → 18	1.36	19 → 21	4.58	17 → 13
<i>FE</i>	0.66	18 → 20	1.36	21 → 19	4.59	13 → 17
<i>II</i>	9.96	37 → 36	9.97	36 → 37	/	/
<i>GK</i>	3.63	36 → 37	5.42	16 → 19	19.0	20 → 24
<i>KG</i>	3.63	37 → 36	5.42	19 → 16	19.0	24 → 20
<i>ST</i>	0.48	32 → 35	1.13	36 → 37	0.48	32 → 35
<i>TS</i>	0.43	35 → 32	1.14	37 → 36	0.43	35 → 32
<i>HE</i>	0.48	18 → 17	1.03	36 → 37	0.48	18 → 17
<i>EH</i>	0.61	17 → 18	0.63	19 → 21	0.61	17 → 18
<i>GH</i>	0.43	20 → 18	1.10	20 → 24	0.43	20 → 18
<i>HG</i>	0.23	18 → 20	0.71	21 → 19	0.23	18 → 20
<i>PN</i>	0.38	32 → 26	0.96	36 → 37	0.38	32 → 26
<i>NP</i>	0.72	20 → 18	1.37	20 → 24	1.64	26 → 32
<i>SQ</i>	1.89	19 → 21	1.91	37 → 36	2.00	32 → 33
<i>QS</i>	0.86	21 → 19	1.33	33 → 32	1.33	33 → 32
<i>EF</i>	0.40	17 → 13	1.31	20 → 18	0.40	17 → 13
<i>FE</i>	0.70	18 → 20	0.80	36 → 37	5.82	13 → 17
<i>II</i>	1.05	36 → 37	3.36	18 → 17	/	/
<i>GK</i>	0.47	20 → 24	0.88	20 → 18	0.47	20 → 24
<i>KG</i>	2.36	21 → 19	3.12	36 → 37	14.5	24 → 20
<i>ST</i>	0.49	32 → 35	2.39	32 → 33	0.49	32 → 35
<i>TS</i>	0.43	35 → 32	1.00	20 → 18	0.43	35 → 32

Table 6. Result of paths from Map VI to Map III. Full path: top; partial path: bottom.

algorithm comes in two variants: the full path comparison and the partial path comparison. On average we achieve a true positive rate above 90%. This is very good since for applying it in a framework for matching Topology Graphs, such as in Schwertfeger and Birk (2015a), other factors such as neighborhoods, vertex similarities and sub-graph isomorphisms are also applied to determine a match between two maps. This framework will then resolve the ambiguities of similar looking paths, which might occur more often in bigger maps. The partial path comparison is especially suited for determining a vertex similarity, because we can see how well the paths which start at that vertex match the paths of the corresponding vertex.

In the future we plan to include the path dissimilarity in a graph matching framework. In the long term we also want to apply the matching to 3D Topology Graphs, for example for 3D map evaluation such as in Schwertfeger and Birk (2015b).

REFERENCES

Bai, X. and Latecki, L.J. (2008). Path similarity skeleton graph matching. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 30(7), 1282–1292.

Bai, X., Latecki, L.J., and Liu, W.Y. (2007). Skeleton pruning by contour partitioning with discrete curve evolution. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 29(3), 449–462.

Beeson, P., Jong, N., and Kuipers, B. (2005). Towards autonomous topological place detection using the extended voronoi graph. In *Robotics and Automation, 2005. ICRA 2005. Proceedings of the 2005 IEEE International Conference on*, 4373 – 4379.

Horn, B.K.P. (1987). Closed-form solution of absolute orientation using unit quaternions. *Journal of the Optical Society of America*, 4(4), 629–642.

Jacoff, A., Sheh, R., Virts, A.M., Kimura, T., Pellenz, J., Schwertfeger, S., and Suthakorn, J. (2012). Using

competitions to advance the development of standard test methods for response robots. In *Proceedings of the Workshop on Performance Metrics for Intelligent Systems*, 182–189. ACM.

Kleiner, A. and Dornhege, C. (2011). Mapping for the support of first responders in critical domains. *Journal of Intelligent & Robotic Systems*, 64(1), 7–31.

Ko, B.Y., Song, J.B., and Lee, S. (2004). Real-time building of a thinning-based topological map with metric features. In *Intelligent Robots and Systems, 2004. (IROS 2004). Proceedings. 2004 IEEE/RSJ International Conference on*, volume 2, 1524 –1529.

Kolling, A. and Carpin, S. (2008). Extracting surveillance graphs from robot maps. In *Intelligent Robots and Systems, 2008. IROS 2008. IEEE/RSJ International Conference on*, 2323–2328.

Latecki, L.J. and Lakämper, R. (2000). Shape similarity measure based on correspondence of visual parts. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 22(10), 1185–1190.

Lau, B., Sprunk, C., and Burgard, W. (2010). Improved updating of euclidean distance maps and voronoi diagrams. In *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*, 281–286.

Milstein, A., McGill, M., Wiley, T., Salleh, R., and Sammut, C. (2011). Occupancy voxel metric based iterative closest point for position tracking in 3d environments. In *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, 4048–4053. IEEE.

Pellenz, J. and Paulus, D. (2010). Stable mapping using a hyper particle filter. In *RoboCup 2009: Robot Soccer World Cup XIII*, 252–263. Springer.

Saeedi, S., Paull, L., Trentini, M., Seto, M., and Li, H. (2014). Group mapping: A topological approach to map merging for multiple robots. *Robotics Automation Magazine, IEEE*, 21(2), 60–72.

Schwertfeger, S. (2012). *Robotic Mapping in the Real World: Performance Evaluation and System Integration*. mastersphd, Jacobs University Bremen. URL <http://d-nb.info/1035268981/34>.

Schwertfeger, S. and Birk, A. (2013). Evaluation of map quality by matching and scoring high-level, topological map structures. In *Robotics and Automation (ICRA), 2013 IEEE International Conference on*, 2221–2226. IEEE.

Schwertfeger, S. and Birk, A. (2015a). Map evaluation using matched topology graphs. *Autonomous Robots*, 1–27.

Schwertfeger, S. and Birk, A. (2015b). Using fiducials in 3d map evaluation. In *IEEE International Symposium on Safety, Security, Rescue Robotics (SSRR)*. IEEE Press.

Schwertfeger, S., Jacoff, A., Pellenz, J., and Birk, A. (2011). Using a fiducial map metric for assessing map quality in the context of robocup rescue. In *IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR)*, 1–6. IEEE Press.

Thrun, S. (1998). Learning metric-topological maps for indoor mobile robot navigation. *Artificial Intelligence*, 99(1), 21 – 71.

Yang, Q. and Sze, S.H. (2007). Path matching and graph matching in biological networks. *Journal of Computational Biology*, 14(1), 56–67.

Preprint for the 9th Symposium on Intelligent Autonomous Vehicles IAV – IFAC.

Schwertfeger, Sören, and Yu, Tianyan, "Matching Paths in Topological Maps", *9th Symposium on Intelligent Autonomous Vehicles (IAV), IFAC*: IFAC, 2016.

Publication Date: July 2016

Provided by Sören Schwertfeger
ShanghaiTech Advanced Robotics Lab
School of Information Science and Technology
ShanghaiTech University

<http://robotics.shanghaitech.edu.cn/people/soeren>
<http://robotics.shanghaitech.edu.cn>
<http://sist.shanghaitech.edu.cn>
<http://www.shanghaitech.edu.cn/eng>

File location

<http://robotics.shanghaitech.edu.cn/publications>