2021 IEEE Intelligent Transportation Systems Conference (ITSC)
Indianapolis, USA. September 19-21, 2021

Straight Skeleton Based Automatic Generation of Hierarchical Topological Map in Indoor Environment

Qian Hou[1,2], Songyi Zhang[1,2], Shitao Chen[1,2], Zhixiong Nan[1] and Nanning Zheng[1†]

Abstract— **Real-time autonomous navigation and scheduling are regarded as the prerequisite capabilities for indoor mobile robots, while the application of prior information about the indoor map can contribute to reducing the level of computational complexity significantly. The indoor maps can be generated through manual marking or using automatic generation methods. The latter considerably reduces human intervention, which makes it more suitable for large indoor environments. However, most of the existing approaches to the automatic generation of indoor maps suffer such problems as imprecision and the lack of smoothness. These issues impede the generated map from meeting the requirement for safety and traceability. In order to address these issues, this paper proposes a novel approach to the automatic generation of the indoor hierarchical topological map (HTM) based on the straight skeleton. This method involves three steps. Firstly, the boundaries of walls and obstacles are extracted from the grid map or the CAD drawing. Secondly, the extracted boundaries are used to automatically generate the primary tracks in narrow corridors, the regions in large open areas, and the secondary tracks within these regions. Thirdly, for the robots that are incapable of taking a spin turn, the clothoid-based smoothing method is applied to ensure the curvature continuity of generated tracks. According to the experimental results, the proposed method is accurate in describing the topological structure of complex and changeable indoor environments and effective in providing smooth and easy-to-track trajectories.**

I. INTRODUCTION

With the extensive application of autonomous mobile robots in modern buildings, the navigation and scheduling in complex indoor environments play an increasingly significant role in the relevant fields of research [1]. The indoor map refers to a road network consisting of multiple stations that can provide the prior information required for navigation and scheduling. The pre-generated road network describes the topological structure of the indoor environment and reduces the computational complexity of online navigation searching. Meanwhile, the stations can store plenty of high-level interactive information about the surrounding environment, thus facilitating scheduling. However, the stations are not the focus of automatic indoor map generation, as they are often marked manually according to the real-world condition. Therefore, it is imperative to determine how to generate the road network automatically and accurately in an irregular indoor environment.

As for the existing solutions to the automatic generation of the indoor map [2]–[4], they are mainly reliant on Voronoi diagrams [5]. Despite the capability to generate the medial axis accurately in indoor environments, Voronoi-based methods are susceptible to measurement noise, thus resulting in jitters or glitches in the skeletons generated. This presents unnecessary challenges to the navigation and control of mobile robots. Besides, these methods perform poorly in describing the large open areas like rooms, where the direction of medial axis-based skeletons can get out of control.

In light of the existing problems encountered by Voronoi-based maps, a novel approach is proposed in this paper to generate a hierarchical topological map (HTM) based on the straight skeleton [6], where an HTM is comprised of primary tracks, secondary tracks, and regions. The proposed approach involves three major parts: boundary extraction, hierarchical skeleton generation, and connection smoothness (Fig. 1). The boundary extraction part is purposed to extract the boundary polygons of walls and obstacles from the grid map or CAD drawings. The hierarchical skeleton generation part is intended to generate straight skeleton-based tracks from polygons. Wall polygons are used to generate primary tracks, while the polygons of regions with obstacles are used to generate secondary tracks. Then, the primary tracks, secondary tracks, and regions are combined into the HTM for the multi-scale description of the scene. As for the connection smoothness part, the clothoid-based smoothing method is used in these tracks to ensure the continuity of their curvature, thus making them traceable for the robot that cannot take a spin turn.

To verify the effects of the proposed method, we apply the generated map into the indoor navigation, comparing it with the traditional Voronoi-based maps, and the experiment result shows the effectiveness of our approach.

To validate the proposed method, the generated map is applied to the indoor navigation for comparing against the traditional Voronoi-based maps. The experimental result is obtained to substantiate the effectiveness of our approach.

The contributions of this paper are summarized as follows. Firstly, a new method used to generate an HTM is proposed, which resolves the jitters and glitches caused by the use of skeleton-based methods. Secondly, on the coarse-scale, our approach can be taken to describe the environment with regions and tracks between them. On the fine-scale, inside regions, our approach can also provide the internal reference

*This work was supported by the National Natural Science Foundation of China(NO.61773312,61790563).

[1] Q. Hou, S. Zhang, S. Chen, Z. Nan, N. Zheng are with the Department of Electrical and Information Engineering, Xi'an Jiaotong University, Xi'an, Shaanxi 710049, P.R. China; Email: houqian, zhangsongyi, chenshitao@stu.xjtu.edu.cn; nzx2018, nnzheng@mail.xjtu.edu.cn

[2] Q. Hou, S. Zhang, S. Chen are with the Department of Shunan Academy of Artificial Intelligence, Ningbo, Zhejiang 315000, P.R. China

[†] N. Zheng is the corresponding author.

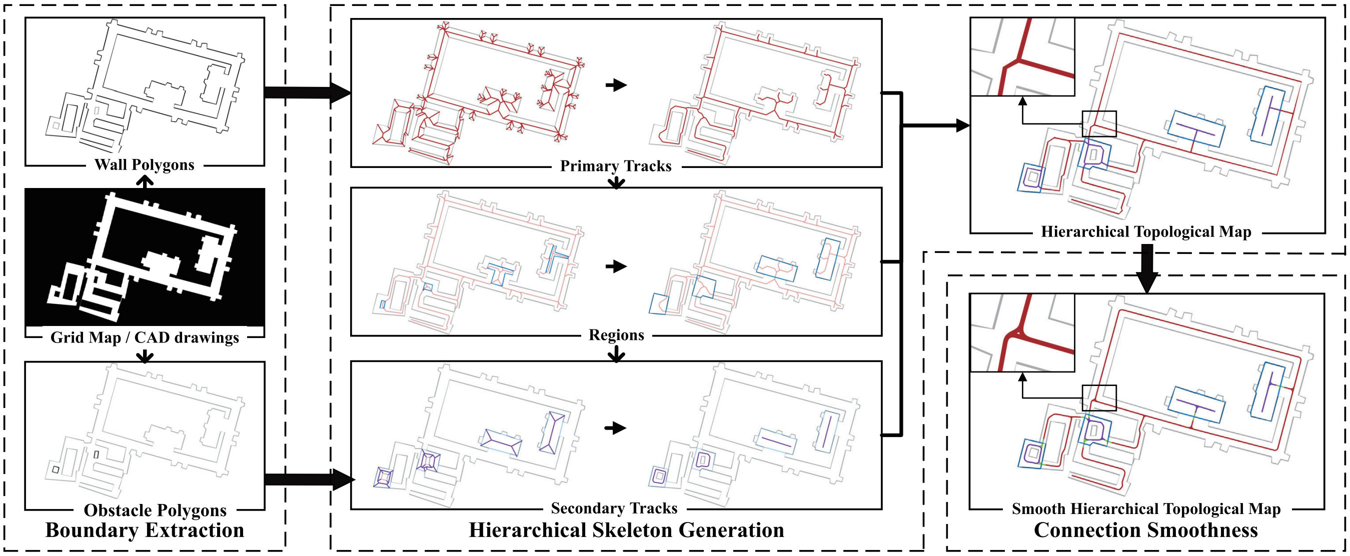


Fig. 1. The workflow of the proposed method of hierarchical topological map (HTM) generation, including three parts, boundary extraction, hierarchical skeleton generation, and connection smoothness. Primary tracks, secondary tracks, and regions are marked in red, purple, and blue, respectively.

tracks generated by the shape of obstacles. Thirdly, our map can be applied to the robots of various types, including the robots that cannot take a spin turn.

This paper is organized as follows. In Section II, a summary is made of the related work to indoor map generation. In Section III, a grid map based boundary extraction method is proposed. In Section IV, our method is introduced to build the HTM with the straight skeleton. In Section Section V, a connection smoothness method is introduced to avoid curvature jumping. In Section VI, our maps are compared with traditional Voronoi-based maps to demonstrate the effectiveness of the proposed approach.

II. Related Work

There have been two main approaches proposed with the focus placed on the generation of the grid map-based indoor map: manual marking and automatic generation. Manual marking methods are adopted to generate reference tracks through the addition of stations and connections between them. Yayan et al. [7] achieved the automatic navigation of intelligent wheelchair through an indoor map manually generated by the Java Open Street Map editor (JOSM). Vipo [8] developed a spatial-visual programming system for path and task editing, so as to coordinate the movement and workflow of robots in a factory. Despite many well-designed human-computer interfaces which can facilitate the generation of maps, it remains a challenge to generate an accurate centerline via manual marking-based maps due to the occurrence of deviation.

The automatic generation methods are mainly reliant on analyzing the topological structure of the grid map. Proposed by Kavraki et al. [9] in 1996, probabilistic roadmaps (PRM) can construct paths by connecting the randomly scattered points with no collision. It is easy for PRM to conduct a search for an optimal path from start to end. However, the path is probably too close to obstacles because of the randomness, thus resulting in the lack of safety.

Voronoi-based maps can be accurate in generating the medial axis in the indoor environment, but PRM-based maps cannot. As the most basic form in the Voronoi family, the Voronoi diagram (VD) is defined as the locus of points equidistant to two or more points. Such a definition ensures that the generated skeletons are farthest from all obstacles. By extending point set to obstacle set, the generalized Voronoi diagram (GVD) [2] is defined by the locus of points equidistant to two or more obstacles. Furthermore, by extending GVD to higher dimensions, Choset et al. [2] proposed the definition of generalized Voronoi graph (GVG), that is, the one-dimensional set of points equidistant to n or more obstacles in n-dimensions. In order to generate a lane graph that can be applied to intelligent vehicles, Dolgov et al. [3] performed local smoothing and global smoothing on GVD. Beeson et al. [4] revised GVG into the extended Voronoi graph (EVG), thus achieving the seamless transition from a skeleton of midline in corridors to a skeleton that follows walls in open rooms. In this way, the problem can be effectively solved when there is only a limited sensory horizon for robots. However, the reference tracks generated by Voronoi suffer problems in jitter and glitch due to the measurement noise.

This paper proposes a novel approach that can be taken to generate the straight skeleton-based HTM, including boundary extraction, hierarchical skeleton generation, and connection smoothness. The above-mentioned problems can be solved using this approach.

Aichholzer et al. [6] proposed the definition and algorithm of the straight skeleton based on simple polygons. On this basis, efforts were continued in [10]–[15] to expand and develop straight skeleton generation algorithms. Different from Voronoi, the straight skeleton is not defined by a distance function but the shrinkage of the polygon. That is to say, each edge of the polygon moves toward the interior along its normal direction at the same speed, while each vertex moves in the direction of the angular bisector of its incident edges. During the movement, the straight skeleton is obtained using the trajectory of vertices.

Some researchers have proposed the use of the straight skeleton. Haunert and Sester [16] applied the straight skeleton to generate the centerline of such geographic elements as roads and rivers automatically. With the Industry Foundation Classes (IFC) model, a widely-used standard format of Building Information Modeling (BIM), as input, Fu et al. [17] built the medial axis indoor navigation networks based on the straight skeleton. However, both of them encountered the challenge of dealing with large open areas in the process of straight skeleton generation. In order to achieve the objective of representing lakes by regions and rivers by centerlines, Haunert and Sester [16] manually gave a threshold for the width of the river during the shrinkage of the straight skeleton. When dealing with open rooms, Fu et al. [17] marked the shape and position of furniture manually to avoid the distortion of skeletons in large rooms. However, all of these solutions are not fully automatic.

III. Boundary Extraction of Grid Map

Boundary extraction is the premise of HTM generation. As boundaries can be obtained directly from CAD drawings, this section focus on the grid map.

A. Separation of Walls and Obstacles

The occupancy grid map consists of the occupied area, undeveloped area, and free area. With the focus placed on the binarization of the free area, there are three steps in separating the contours of walls and obstacles as follows:

1) Noise filtering: Since the gird maps generated by such methods as SLAM encounter problems due to measurement noise, the morphological opening is performed to filter out noise points.

2) Contours extraction: The boundary set, consisting of walls and obstacles, can be regarded as two-level contours. Among them, parent contours represent outer walls, while child contours represent the collection of inner walls and obstacles. The Suzuki contour tracing algorithm [18] is adopted to obtain such two-level contours.

3) Contours separation: Parent contours can be classified as walls. Nevertheless, it is necessary to further distinguish between walls and obstacles in child contours. Douglas-Peucker (DP) algorithm [19] is applied to extract the polylines of child contours. The contour, containing a large-scale polyline, belongs to walls, and otherwise, it is classed into obstacles. If the contour is degenerated into a straight line with an area of zero, it can also be classified as a wall since the contour is narrow and long. Then, it is possible to obtain the wall contour set $C_w = \{c_{w_i}\}_{i=1}^{N_W}$, and the obstacle contour set $C_o = \{c_{o_i}\}_{i=1}^{N_O}$, where c_{w_i} and c_{o_i} contains points in $\mathbb{R}^2$, while N_W and N_O means the total number of each set.

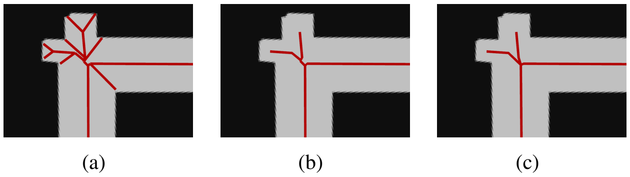


Fig. 2. Pruning of straight skeleton. (a) The straight skeleton generated by the boundary polygon. (b) Incident edge pruning of skeletons in Fig 2(a). (c) Short edge pruning of skeletons in Fig 2(b).

B. Boundary Fitting

Directly obtained from the contour information, C_w and C_o are susceptible to the boundary noise. The DP algorithm [19] with a threshold determined by the boundary noise is applied to simplify C_w to the wall polygon set $W = \{w_i | w_i \subseteq c_{w_i}, c_{w_i} \in C_w\}_{i=1}^{N_W}$, and C_o to the obstacle polygon set $O = \{o_i | o_i \subseteq c_{o_i}, c_{o_i} \in C_o\}_{i=1}^{N_O}$.

IV. Hierarchical Skeleton Generation

The hierarchical skeleton generation is a critical step in generating the HTM. The wall polygons and obstacle polygons obtained from the grid map will be parsed as a combination of primary tracks, secondary tracks, and regions.

A. Generation and Pruning of Straight Skeleton

Assume a straight skeleton graph $G = (V, E)$ with vertex set $V = \{v_i\}_{i=1}^{N_V}$ and edge set $E = \{e_i\}_{i=1}^{N_E}$, where $e_k = \{v_i, v_j\} (e_k \in E, v_i \in V, v_j \in V)$ represents an edge between v_i and v_j, N_V indicates the total number of the vertices, and N_E denotes the total number of the edges. The degree d_{v_i} of the vertex $v_i (v_i \in V)$ is defined as the number of edges associated with it. The vertex satisfying $d_{v_i} = 1$ or $d_{v_i} > 2$ is defined as an intersection point. E in G is rearranged by combining all the edges between two intersection points into a skeleton segment $s = \{e_i | e_i \in E\}_{i=1}^{N_e}$, where N_e refers to the number of edges in s. Then, the collection of skeleton segments can be expressed as $S = \{s_i\}_{i=1}^{N_S}$, where N_S represents the total number of skeletons segments in G, and the graph consisting of skeleton segments can be expressed as $G_s = (V, S)$.

As shown in Fig. 2(a), there are some branches in G_s irrelevant to the topological description of the indoor environment. The pruning operations are detailed as follows:

1) Incident edge pruning: As shown in Fig. 2(b), the skeleton segment $s_i (s_i \in S)$ whose endpoints coincide with the original polygon is supposed to be pruned by deleting all the edges in s_i and associated isolated vertices with $d_v = 0 (v \in e, e \in s_i)$.

2) Short edge pruning: As shown in Fig. 2(c), the skeleton segment $s_i (s_i \in S)$ whose length shorter than expected is pruned, and the two intersection points on s_i are merged.

B. Associating Regions and Tracks

Fig. 3(a) shows the circumstance where the tracks generated by straight skeleton become uncontrollable and meaningless. Compared with narrow corridors, skeletons are more susceptible to the boundary shape in the large open areas

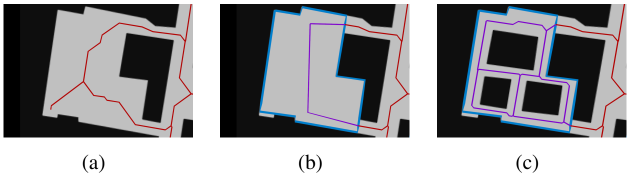

(a) (b) (c)

Fig. 3. (a) Straight skeleton in a large open area. (b) HTM without obstacles inside the region. Secondary tracks are generated by the maximum inscribed rectangle of the region. (c) HTM with obstacles inside the region. Secondary tracks are generated by the region with obstacles. Primary tracks, secondary tracks, and regions are marked in red, purple, and blue, respectively.

with a smaller aspect ratio. In this study, our consideration is given to describing such large open areas like rooms as regions for special treatment.

The shrinking-based straight skeleton generation method [6] is taken into consideration. As the edges of the polygon shrink inwardly, the disconnection and disappearance of regions continue to occur. The long and narrow corridors will first degenerate into a skeleton, while further shrinkage is required for large open areas. Based on this feature, regions and tracks are automatically divided.

Assume the shrinking step as t, the shrinking polygon set of W is expressed as $P(t) = \{p_i\}_{i=1}^{N_{P(t)}}$, where $N_{P(t)}$ means the total number of the shrinking polygons. It is expected to find a balance between the number and the convexity of elements in $P(t)$ during the skeleton shrinking. Suppose the equation

$$\begin{aligned}\mathcal{L}(P(t)) = \ & \mathcal{L}_h(P(t)) + k_a\mathcal{L}_a(P(t)) \\ & + k_c\mathcal{L}_c(P(t)) + k_s\mathcal{L}_s(P(t)),\end{aligned} \tag{1}$$

where $\mathcal{L}_h(P)$ represents the cost of the polygon with holes, $\mathcal{L}_a(P)$ refers to the cost of the small area, $\mathcal{L}_c(P)$ indicates the cost of the non-convex area, $\mathcal{L}_s(P)$ denotes the cost of the polygon number, while k_a, k_c, and k_s represent the corresponding weights.

Assume that regions like rooms are simple polygons without holes, $holes(p_i)$ is used to represent the number of holes in $p_i(p_i \in P(t))$. Then, $\mathcal{L}_h(p_i)$ can be expressed as

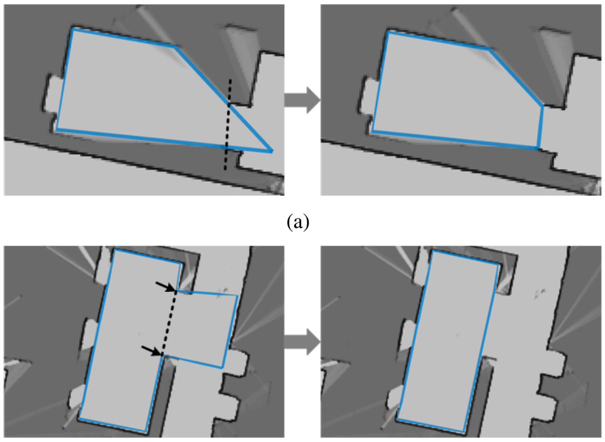
$$\mathcal{L}_h(p_i) = \begin{cases} \infty & \text{holes}(p_i) \neq 0 \\ 0 & \text{holes}(p_i) = 0 \end{cases}. \tag{2}$$

In the process of shrinkage, a region with a small area affected by boundary noise is undesirable. With area(p_i) used to represent the area of the polygon $p_i(p_i \in P(t))$, $\mathcal{L}_a(P(t))$ can be expressed as

$$\mathcal{L}_a(P(t)) = \frac{1}{N_{P(t)}} \sum_{i=1}^{N_{P(t)}} \frac{1}{\text{area}(p_i)}. \tag{3}$$

$\mathcal{L}_c(P(t))$ is purposed to make the region tend to be convex. With area$_c(p_i)$ used to represent the area of the convex hull of $p_i(p_i \in P(t))$, $\mathcal{L}_c(P(t))$ can be expressed as

$$\mathcal{L}_c(P(t)) = \frac{\sum_{i=1}^{N_{P(t)}} (\text{area}_c(p_i) - \text{area}(p_i))}{\sum_{i=1}^{N_{P(t)}} \text{area}_c(p_i)}. \tag{4}$$


(a)

(b)

Fig. 4. Region clipping. (a) Sharp corner clipping. A sharp angle between two adjacent polygon edges may cause a sharp corner in the region entrance. The polygon vertex around the corner can be used to clip the corner. (b) Outside door clipping. If the width of region entrance is greater than that of corridor, the region may contain part of the corridor. A pair of concave points with collinear on the associated region edges [20] are marked as door point to clip the region outside the door.

In the process of shrinkage, the number of regions shows a trend from small to large and then to small. When the shrinkage is close to the start, $P(t)$ almost contains the entire room. When the shrinkage is close to the end, $P(t)$ contains only a small area, neither of which is expected. With the shrinking result made not too close to the start or the end, $\mathcal{L}_s(P(t))$ is expressed as

$$\mathcal{L}_s(P(t)) = \frac{1}{N_{P(t)}}. \tag{5}$$

$\mathcal{L}(P(t))$ in Eq. 1 is calculated for each step in the inward shrinkage to obtain the optimal candidate region set $\hat{P} = P(\hat{t}) = \{\hat{p}_i\}_{i=1}^{N_{\hat{P}}}$, where $\hat{t}$ satisfies $\hat{t} = \arg\min_t \mathcal{L}(P(t))$ and $N_{\hat{P}}$ represents $N_{\hat{P}} = N_{P(\hat{t})}$.

Since the elements in $\hat{P}$ are inward shrinkage of walls, which are smaller than real rooms, the straight skeleton based external expansion with step $\hat{t}$ of each polygon in $\hat{P}$ needs to be calculated for obtaining the expanded region set $\hat{P}_e = \{p_{e_i}\}_{i=1}^{N_{\hat{P}}}$. To deal with $p_{e_i} \in \hat{P}_e$ that do not match the real room after external expansion, the clipping operations on on p_{e_i} including sharp corner clipping (Fig. 4(a)), outside door clipping (Fig. 4(b)) and outside wall clipping (reserve of $\hat{P}_e \cap W$) will be involved to obtain the clipped region set $\hat{P}_r = \{\hat{p_{r_i}}\}_{i=1}^{N_{\hat{P}}}$.

As shown in Fig. 3(b) and Fig. 3(c), the skeleton segments outside $\hat{P}_r$ generated by W and O_w can be reserved as primary tracks G_{sw}, where $O_w \subseteq O$ is the obstacles outside P_r. The skeleton segments inside $\hat{P}_r$ generated by $\hat{P}_r$ and O_o can be retained as the secondary tracks G_{so}, where $O_o \subseteq O$ represents the obstacles inside P_r. After connection between the endpoint in G_{sw} linked to the region and the closest point in G_{so} inside the region, a short dead-end skeleton with with $d_{v_i} = 1$ is pruned. Finally, an HTM $M = \{G_{sw}, G_{so}, \hat{P}_r\}$ can be obtained.

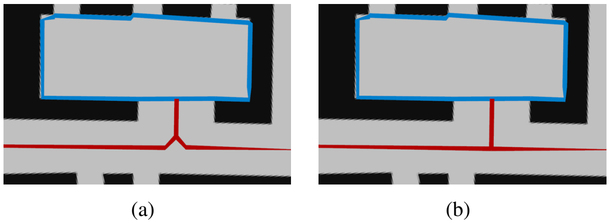


Fig. 5. Intersection adjustment. Primary tracks and regions are marked in red and blue, respectively. (a) Primary tracks and regions before intersection adjustment. (b) Primary tracks and regions after intersection adjustment.

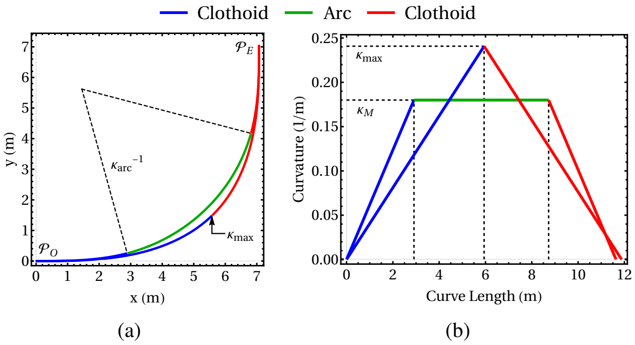


Fig. 6. Clothoid model. (a) The curve of a clothoid pair, comprised of two symmetrical clothoid segments, and the curve of a clothoid-arc-clothoid, comprised of a clothoid, a constant radius arc, and a clothoid. (b) Curvature diagram with the curve length of curves in Fig. 6(a).

C. Skeleton Optimization

In order to integrate the skeletons in G_{sw} and G_{so} into longer line segments, it is necessary to perform intersection adjustment and linear optimization. Fig. 5 shows the process of intersection adjustment. In order to reduce the influence when the region is located on the side of the main way, adjustment is made to the intersection point whose associated skeleton segment is similar to a line. The line similarity rate of two adjacent skeleton segments s_1, s_2 is defined as $\alpha = L(\ell)/L(v_1, v_2)$, where v_1 represents the end vertex on s_1, v_2 indicates the end vertex on s_2, $L(v_1, v_2)$ denotes the skeleton length between v_1 and v_2, and $L(\ell)$ refers to the skeleton length coincident with line v_1v_2. Given a threshold $\hat{\alpha}$, if $\alpha > \hat{\alpha}$ and there are no other conflicts arising, s_1 and s_2 can be simplified by projecting each point of s_1 and s_2 onto line v_1v_2. In the step of linear optimization, the DP algorithm [19] with a threshold as determined by the tolerance of centerline offset is applied in each skeleton segment.

V. Connection Smoothness

Clothoid based connection smoothness method is used for the robots that cannot take a spin turn, thus ensuring the G2 continuity of generated tracks.

A. Clothoid Model

Clothoid is referred to as a curve whose curvature changes linearly with its length. On this basis, a clothoid pair is defined as two symmetrical clothoids, and a clothoid-arc-clothoid is defined as two symmetrical clothoids with a constant radius arc in the middle [21], [22].

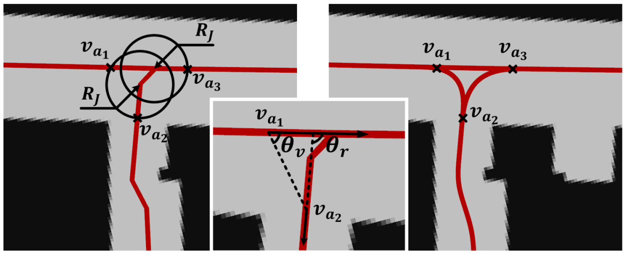


Fig. 7. Generation of curvature-continuos turn. Left shows tracks in HTM. Right shows tracks in smooth HTM. The figure in the middle is a calculation example from v_{a_1} to v_{a_2}.

As shown in Fig. 6, the track generated by the clothoid pair shows a large maximum curvature κ_{max} at a sharp turn, which is possible to cause the problem that the curve is incapable to be traced by the controller. It is assumed that the curvature κ on the track satisfy $|\kappa| < |\kappa_M|$, where κ_M represents the upper curvature limit of the robot. If $|\kappa_{max}| > |\kappa_M|$, a clothoid-arc-clothoid with $\kappa_{arc} = \kappa_M$ is applied for turning, where κ_{arc} refers to the curvature of the arc. Otherwise, if $|\kappa_{max}| \le |\kappa_M|$, the clothoid pair degenerated from the clothoid-arc-clothoid is used instead.

It is supposed that the distance between the start $\mathcal{P}_O$ and the end $\mathcal{P}_E$ is $2D$ and the deflection of $\overrightarrow{\mathcal{P}_O\mathcal{P}_E}$ is $\Theta(-\pi < \Theta < \pi)$. All unknown parameters can be obtained using quasi-Newton's method [23] to solve the following equation

$$\begin{cases} J = \dfrac{1}{k}\begin{bmatrix} F_C(kL) \\ \text{sign}(\sigma)F_S(kL) \end{bmatrix}^{\mathsf{T}} \begin{bmatrix} \cos\Theta \\ \sin\Theta \end{bmatrix} \\ D = \dfrac{1}{\kappa_M}\sin\left(\dfrac{1}{2}M\kappa_M\right) + J \\ \Theta = \dfrac{1}{2}\sigma L^2 + \dfrac{1}{2}M\kappa_M \\ \kappa_M = \sigma L \end{cases}, \tag{6}$$

where $k = \sqrt{\frac{|\sigma|}{\pi}}$, σ is clothoid sharpness, L represents the length of a single clothoid, M indicates the length of the arc, while F_C and F_S are Fresnel integrals that can be expressed as

$$\begin{cases} F_C(z) = \displaystyle\int_0^z \cos\left(\frac{1}{2}\pi t^2\right) dt \\ F_S(z) = \displaystyle\int_0^z \sin\left(\frac{1}{2}\pi t^2\right) dt \end{cases}. \tag{7}$$

B. Generation of Curvature-continuos Turn

The basic elements of each curvature-continuos turn include the straight line, the clothoid pair, and the clothoid-arc- clothoid. As shown in Fig. 7, for the vertex set V in the HTM M as obtained in Section IV, we aggregate the vertices within circles of radius R_J into a junction and choose points both on tracks and circles into anchor point set $V_a = \{v_{a_i}\}_{i=1}^{N_{V_a}}$, where N_{V_a} refers to the total number of elements in V_a. Focusing on two anchor points $v_{a_i} = \{x_i, y_i, \theta_i\}$ and $v_{a_j} = \{x_j, y_j, \theta_j\}$ in V_a, we project v_{a_j} to the plane with origin $\begin{bmatrix} x_i & y_i \end{bmatrix}^{\mathsf{T}}$ and deflection θ_i to obtain the relative point $v_r = \{x_r, y_r, \theta_r\}(-\pi \le \theta_r \le \pi)$. It

TABLE I
PERFORMANCE OF SKELETON GENERATION

Methods	Number of Nodes	Average Line Length Ratio	Generation Time(s)
GVD	4325	1.03958	0.7058
RGVG	855	1.05965	1.2540
EVG	930	1.07804	0.9621
HTM	166	1.03596	18.1789
smooth HTM	-	-	18.2262

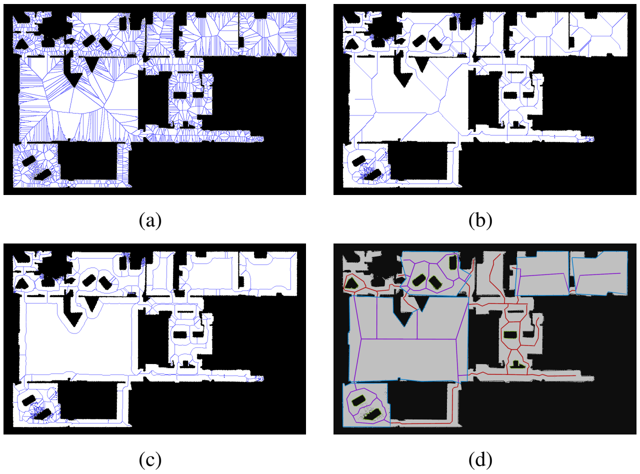

(a) (b) (c) (d)

Fig. 8. Comparison of skeleton generation, using the grid map obtained from [4]. (a) GVD. (b) RGVG. (c) EVG with a sensory horizon of $1.25m$. (d) HTM (our approach). Primary tracks, secondary tracks, and regions are marked in red, purple, and blue, respectively.

is assumed that $\theta_v = \tan^{-1}(y_r/x_r)$ and $\tilde{\theta_v}(-\pi \leq \tilde{\theta_v} \leq \pi)$ is the angle of θ_v flipped over 180°. We will connect the point pairs satisfying $\theta_v = \theta_r = 0$ with lines, the point pairs satisfying $\min(\tilde{\theta_v}, \theta_v) \leq \theta_r \leq \max(\tilde{\theta_v}, \theta_v)$ with two clothoid-arc-clothoid pairs, and the point pairs satisfying $\theta_r < \min(\tilde{\theta_v}, \theta_v)$ or $\theta_r > \max(\tilde{\theta_v}, \theta_v)$ with the combination of line and clothoid-arc-clothoid. A smooth HTM can be obtained by traversing all the anchor point pairs in V_a through the above operations.

VI. EXPERIMENT

To verify the effects of the proposed method, the HTM and smooth HTM are validated by means of skeleton generation and the global panner with an indoor topological map.

A. Skeleton Generation

The proposed approach is applied to the occupancy grid map from [4], and a comparison is performed with the existing methods, including GVD [24]–[26], reduced generalized Voronoi graph (RGVG) [27], and EVG [4]. The resolution of the input grid map is $0.05m$. The sensory horizon of the robot is limited to $1.25m$ in the EVG based map. Fig. 8 shows the indoor maps generated using each method.

The number of nodes and the average line length ratio are tested to validate the skeleton generation results of each map quantitatively. As tracks are stored differently in each map, node n should be defined to evaluate them fairly. For pixel-based methods like GVD, RGVG, and EVG, n is defined as branch point (degree $d_n > 2$) or endpoint (degree $d_n = 1$). For vector-based HTM, n is defined as the skeleton point in primary tracks or secondary tracks. Node set $\mathcal{N} = \{n_i\}_{i=1}^{N_{\mathcal{N}}}$ is defined as the collection of all nodes in the indoor map. The number of nodes $N_{\mathcal{N}}$ is defined as the total number of elements in $\mathcal{N}$.

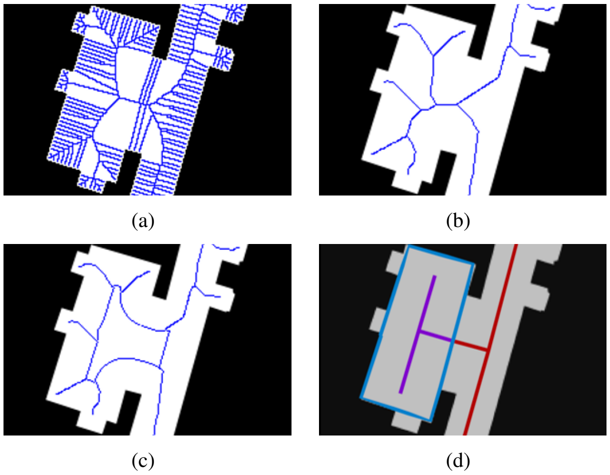

(a) (b) (c) (d)

Fig. 9. Comparison of skeleton generation with room on the side of the corridor. (a) GVD. (b) RGVG. (c) EVG with a sensory horizon of $1.25m$. (d) HTM (our approach). Primary tracks, secondary tracks, and regions are marked in red, purple, and blue, respectively.

The track segment t is defined as the skeleton segment between two nodes with $d_{n_i} \neq 2$. Track segment set $\mathcal{T} = \{t_i\}_{i=1}^{N_{\mathcal{T}}}$ is defined as the collection of all skeleton segment in the indoor map, where $N_{\mathcal{T}}$ represents the total number in $\mathcal{T}$. The average line length ratio is defined as

$$\sum_{i=1}^{N_{\mathcal{T}}} \left(L_r(t_i) * R(t_i)\right), \tag{8}$$

where $L_r(t_i)$ represents the weight of t_i, and $R(t_i)$, a parameter to measure the track jitter, represents the line length ratio of t_i. $L_r(t_i)$ and $R(t_i)$ can be expressed as

$$\begin{cases} L_r(t_i) = \dfrac{L(t_i)}{\sum_{i=1}^{N_{\mathcal{T}}} L(t_i)} \\ R(t_i) = \dfrac{L(t_i)}{L(t_i^s)} \end{cases}, \tag{9}$$

where $L(t_i)$ denotes the length of t_i, t_i^s is a simplified track keeping start point, turning point and end point in t_i. t_i^s can be calculated by DP algorithm [19] with a high threshold. A smaller $R(t_i)$ means fewer jitters and glitches in track segment t_i.

In the columns 2-3 of Table I, a summary is made of the number of nodes and the average line length ratio of each map shown in Fig. 8. As suggested by the results, the proposed HTM has the smallest number of nodes and average track length ratio. Compared with the other three Voronoi-based methods, the tracks generated using our approach are more concise and encounter fewer problems in jitter and glitch. The column 4 of Table I summarizes the time taken

TABLE II

Performance of Global Planner

Methods	Total Track Length(m)			Traceability (Lookahead Distance $\Delta = 0.1m$)			Average Angular Acceleration(rad/s^2) (Lookahead Distance $\Delta = 0.5m$)		
	Group 1	Group 2	Group 3	Group 1	Group 2	Group 3	Group 1	Group 2	Group 3
GVD	8.65330	8.41690	7.80624	-	-	-	1.51923	0.81406	0.76657
RGVG	7.94558	8.11630	7.69914	-	-	-	1.01868	0.86287	0.87651
EVG	7.94558	8.11630	7.74056	-	-	-	0.98288	0.91770	0.84280
HTM	7.50000	7.70000	6.97289	✓	✓	✓	0.76016	0.73369	0.70815
smooth HTM	7.50000	7.54384	6.97289	✓	✓	✓	0.76223	0.71592	0.70410

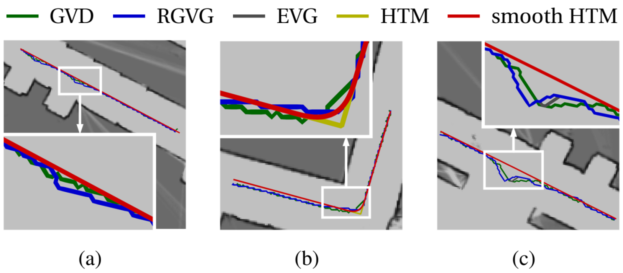


Fig. 10. Comparison of tracks generated by A* global planner. EVG based tracks coincide with RGVG based tracks within the sensory horizon. HTM based tracks coincide with smooth HTM when non-turning. (a) Tracks in the straight corridor (Group 1). (b) Tracks in the turning corridor (Group 2). (c) Tracks in the corridor with a room on the side (Group 3).

to generate each map in Fig. 8 running on Intel Core i7-8750H CPU. The results indicate that HTM and smooth HTM consume much more time during map generation than the other three maps. However, it is needless to be concerned about the generation time, as the map is generated offline.

Fig. 9 shows the case where a room is on the side of the corridor. Generated by GVG, RGVG, and EVG with a sensory horizon of $1.25m$, the skeletons in the corridor and that in the room are affected by each other, thus resulting in the bending of tracks at the connection, while our approach is the opposite.

B. Global Navigation

From the grid map with a resolution of $0.05m$, the maps generated by GVD, RGVG, EVG with a sensory horizon of $1.25m$, HTM, and smooth HTM are configured on a real robot. The robot used in this study is a two-wheel differential robot with a wheelbase of $0.415m$ and a wheel radius of $0.826m$. For each map, its global planner based on the A* uses the components obtained from the ROS Navigation stack. The performance of the global planner is validated on each map.

Fig. 10 shows the tracks generated by each global planner. There are three groups of starting and ending points chosen as input for the global planner. Group 1, Group 2, and Group 3 are involved to evaluate the tracks generated by each global planner in the straight corridor, the turning corridor, and the corridor with a room on the side, respectively. In the columns 2-4 of Table II, the total track length of each global planner in each group is summarized. According to the result, the global planner used in our approach has a shorter track length as compared to the other three methods under the identical conditions. The columns 5-7 of Table II summarize the traceability of generated tracks. Due to the jitter and glitch, the tracks based on GVD, RGVG, and EVG are incapable to be traced by the controller with a lookahead distance of $0.1m$. All of the tracks generated by each global planner are traceable when the lookahead distance is $0.5m$. The average angular acceleration of the robot is measured on each group, and it is expressed as

$$\frac{1}{N_t}\sum_{i=1}^{N_t}\frac{|w(i)-w(i-1)|}{t(i)-t(i-1)}, \tag{10}$$

where N_t represents the number of samples, $t(i)$ indicates the time on the i-th sampling, and $w(i)$ refers to the angular velocity of robot on the i-th sampling. $w(i)$ on a two-wheel differential robot can be approximately calculated as

$$w(i) = \frac{v_i^r - v_i^l}{L_b}, \tag{11}$$

where v_i^r represents the linear velocity of the right wheel on the i-th sampling, v_i^l denotes the linear velocity of the left wheel on the i-th sampling, and L_b indicates the wheelbase. A smaller average angular acceleration means that the robot is more stable during trajectory tracing. In the columns 8-10 of Table II, a summary is made of the result of average angular acceleration, suggesting that the robot with the global planner based on HTM and smooth HTM has the second smallest and the smallest average angular acceleration. In comparison with the other three Voronoi-based methods, the tracks generated by the global planner based on HTM and smooth HTM are smoother and easier to trace.

VII. Conclusion

In this paper, a novel method purposed to automatically generate an HTM is proposed, which is efficient in describing the indoor environment as a hierarchical topological structure. A comparative experiment is conducted against the traditional Voronoi-based methods to demonstrate that our approach encounters fewer jitter and glitch problems in the skeletons generated.

For future research, our plan is to test the proposed approach in a larger, more complex environment. Bedsides, a further research will be conducted on how to distinguish between measurement noise and such minor obstacles as chair legs. Moreover, consideration will be given to reducing the time complexity of the HTM generation.

References

[1] A. Pandey, S. Pandey, and D. Parhi, "Mobile robot navigation and obstacle avoidance techniques: A review," *Int Rob Auto J*, vol. 2, no. 3, p. 00022, 2017.

[2] H. Choset and J. Burdick, "Sensor-based exploration: The hierarchical generalized voronoi graph," *The International Journal of Robotics Research*, vol. 19, no. 2, pp. 96–125, 2000.

[3] D. Dolgov and S. Thrun, "Autonomous driving in semi-structured environments: Mapping and planning," in *2009 IEEE International Conference on Robotics and Automation*. IEEE, 2009, pp. 3407–3414.

[4] P. Beeson, N. K. Jong, and B. Kuipers, "Towards autonomous topological place detection using the extended voronoi graph," in *Proceedings of the 2005 IEEE International Conference on Robotics and Automation*. IEEE, 2005, pp. 4373–4379.

[5] F. Aurenhammer, "Voronoi diagrams - a survey of a fundamental geometric data structure," *ACM Computing Surveys (CSUR)*, vol. 23, no. 3, pp. 345–405, 1991.

[6] O. Aichholzer, D. Alberts, F. Aurenhammer, and B. Gärtner, "Straight skeletons of simple polygons," in *Proc. 4th Internat. Symp. of LIESMARS*, 1995, pp. 114–124.

[7] U. Yayan, B. Akar, F. Inan, and A. Yazici, "Development of indoor navigation software for intelligent wheelchair," in *2014 IEEE International Symposium on Innovations in Intelligent Systems and Applications (INISTA) Proceedings*. IEEE, 2014, pp. 325–329.

[8] G. Huang, P. S. Rao, M.-H. Wu, X. Qian, S. Y. Nof, K. Ramani, and A. J. Quinn, "Vipo: Spatial-visual programming with functions for robot-iot workflows," in *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, 2020, pp. 1–13.

[9] L. E. Kavraki, P. Svestka, J.-C. Latombe, and M. H. Overmars, "Probabilistic roadmaps for path planning in high-dimensional configuration spaces," *IEEE transactions on Robotics and Automation*, vol. 12, no. 4, pp. 566–580, 1996.

[10] O. Aichholzer and F. Aurenhammer, "Straight skeletons for general polygonal figures in the plane," in *International computing and combinatorics conference*. Springer, 1996, pp. 117–126.

[11] P. Felkel and S. Obdrzálek, "Improvement of oliva's algorithm for surface reconstruction from contours," in *In Proc. 15th Spring Conf. Comp. Graphics*. Citeseer, 1999.

[12] D. Eppstein and J. Erickson, "Raising roofs, crashing cycles, and playing pool: Applications of a data structure for finding pairwise interactions," *Discrete & Computational Geometry*, vol. 22, no. 4, pp. 569–592, 1999.

[13] S.-W. Cheng and A. Vigneron, "Motorcycle graphs and straight skeletons," in *Proceedings of the thirteenth annual ACM-SIAM symposium on Discrete algorithms*, 2002.

[14] S.-W. Cheng and A. Vigneron, "Motorcycle graphs and straight skeletons," *Algorithmica*, vol. 47, no. 2, pp. 159–182, 2007.

[15] S. Huber and M. Held, "Theoretical and practical results on straight skeletons of planar straight-line graphs," in *Proceedings of the twenty-seventh annual symposium on Computational geometry*, 2011, pp. 171–178.

[16] J.-H. Haunert and M. Sester, "Area collapse and road centerlines based on straight skeletons," *GeoInformatica*, vol. 12, no. 2, pp. 169–191, 2008.

[17] M. Fu, R. Liu, B. Qi, and R. R. Issa, "Generating straight skeleton-based navigation networks with industry foundation classes for indoor way-finding," *Automation in Construction*, vol. 112, p. 103057, 2020.

[18] S. Suzuki *et al.*, "Topological structural analysis of digitized binary images by border following," *Computer vision, graphics, and image processing*, vol. 30, no. 1, pp. 32–46, 1985.

[19] D. H. Douglas and T. K. Peucker, "Algorithms for the reduction of the number of points required to represent a digitized line or its caricature," *Cartographica: the international journal for geographic information and geovisualization*, vol. 10, no. 2, pp. 112–122, 1973.

[20] K. Joo, T.-K. Lee, S. Baek, and S.-Y. Oh, "Generating topological map from occupancy grid-map using virtual door detection," in *IEEE Congress on Evolutionary Computation*. IEEE, 2010, pp. 1–6.

[21] J. Henrie and D. Wilde, "Planning continuous curvature paths using constructive polylines," *Journal of Aerospace Computing, Information, and Communication*, vol. 4, no. 12, pp. 1143–1157, 2007.

[22] J. Funke, P. Theodosis, R. Hindiyeh, G. Stanek, K. Kritatakirana, C. Gerdes, D. Langer, M. Hernandez, B. Müller-Bessler, and B. Huhnke, "Up to the limits: Autonomous audi tts," in *2012 IEEE Intelligent Vehicles Symposium*. IEEE, 2012, pp. 541–547.

[23] D. F. Shanno, "Conditioning of quasi-newton methods for function minimization," *Mathematics of computation*, vol. 24, no. 111, pp. 647–656, 1970.

[24] B. Lau, C. Sprunk, and W. Burgard, "Improved updating of euclidean distance maps and voronoi diagrams," in *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2010, pp. 281–286.

[25] B. Lau, C. Sprunk, and W. Burgard, "Incremental updates of configuration space representations for non-circular mobile robots with 2d 2.5 d or 3d obstacle models." in *ECMR*, 2011, pp. 49–54.

[26] B. Lau, C. Sprunk, and W. Burgard, "Efficient grid-based spatial representations for robot navigation in dynamic environments," *Robotics and Autonomous Systems*, vol. 61, no. 10, pp. 1116–1130, 2013.

[27] H. Choset and K. Nagatani, "Topological simultaneous localization and mapping (slam): toward exact localization without explicit localization," *IEEE Transactions on robotics and automation*, vol. 17, no. 2, pp. 125–137, 2001.