Proceeding of the IEEE
International Conference on Robotics and Biomimetics
Dali, China, December 2019

Three-Dimensional Truss Modelling for Biped Climbing Robots

Weilin Lin, Guozhen Cai, Shangbiao Wei, Shichao Gu, Haifei Zhu* and Yisheng Guan

Biomimetic and Intelligent Robotics Lab (BIRL)
School of Electromechanical Engineering
Guangdong University of Technology
Guangzhou, Guangdong Province, China

***Abstract*— Recognizing and modelling the three-dimensional (3D) truss-style environment, consisting of cylinder members, is a challenging task for climbing robots. State-of-the-art segmentation algorithms always suffer from incomplete member-segmentation. To solve this problem, in this paper we propose a truss environment modelling (TEM) method using a sliding bounding box algorithm. TEM can obtain accurate poses of the members in truss environments by two strategies. First, the remaining incomplete segmented members are localized by a sliding bounding box along the axis of the members. After that, members are merged based on the new endpoints detected after projecting points in the new bounding box onto the axis, reducing the uncertainty and complexity of members merging. Our proposed method was verified by experiments with steady performance, which satisfied the requirements of the climbing robots.**

***Index Terms*— 3D modelling, truss-environment, objecte segmentation, climbing robots.**

I. INTRODUCTION

Biped climbing robots have great advantages in carrying out high-risk tasks in the large and complex truss environments with a variety of abilities such as climbing gait, good obstacle crossing, and transition between cylinder members. While performing tasks, what they need for autonomous path planning is the results of modeling and expressing an environment [1].

In the past, Computer-Aided Design (CAD), a memory-efficient parametric expression [2] is a typical choice for autonomous path planning in a known environment. But most of the time, climb robots have to work in an unknown environment. At this time, it is difficult to get the whole environment model. When it comes to the unknown environment, an ideal alternative option is to use pointcloud such as some environment modelling works based on the pointcloud [3] [4] but it is more complicated than CAD method.

As for the unknown 3D truss-style environment, there are two main approaches to modelling. The first one is to get the pose of one of the cylinder members then repeat this process again and again. Compared with those complicated works, it would be simpler to use object-segmentation algorithms based on the pointcloud, such as RANdom SAmple Consensus (RANSAC) [5], Hough transform [6] to get one member at a time. Their improved methods have better performance than the original by overcoming flaws such as inefficiency or having too many parameters to tune to fit a better model, but still cannot meet the requirement of modelling results for climbing robots. RANSAC processed the original pointcloud by using random sampling that requires the least number of data points to detect the shape. Due to the dependence on the initially selected point [7], the final results were often nonoptimal parameters under sensor data with noise. Hough transform uses the accumulative-vote method to detect the cylinder in Lidar scanning pointcloud data [8]. Meanwhile, Hough transform needs less time than RANSAC.

(a) Guangzhou South Railway Station (b) Scaffolds in the construction site

Fig. 1: Two examples of truss environments

However, all of the above works can only segment single members instead of all in the truss environment.

The second approach is to get poses of all members at one time. In order to solve the problem of multicylinder extraction, an automatic-detection approach for cylindrical objects in buildings [9] was proposed with two-dimensional Hough transform used to detect the pipe section by projecting the points onto the orthogonal plane. The two-dimensional circular section was merged to obtain the complete pipe-reel-forming model. Although this method solves the problem of low time efficiency in the three-dimensional Hough transform, it only works when objects are parallel to the orthogonal axis, so it cannot simulate inclined core and accessory regions [10]. In [11], an improved method of using

*Corresponding author (hfzhu@gdut.edu.cn). This work is supported in part by the National Natural Science Foundation of China (Grant No. 51605096, 51705086) and the Frontier and Key Technology Innovation Special Funds of Guangdong Province (Grant No. 2017B050506008, 2019A050503011).

978-1-7281-6321-5/19//$31.00 ©2019 IEEE

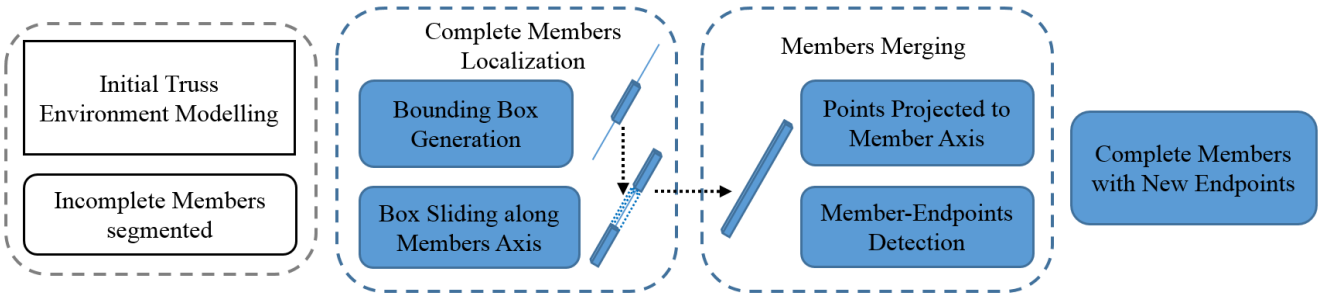


Fig. 2: System framework of truss-environment-modelling (TEM) method overview.

RANSAC for preprocessing and Hough transform to extract the cylinder model from the data was successfully used to segment and extract the pipelines of the whole building. However, the high complexity of the whole method led to low memory efficiency and was not suitable for processing visual sensor data with high noise. These works could extract cylinders from the pointcloud, but none of them proposed a complete modelling system applied to biped climbing robots working in truss environments (Figure 1).

Aiming at this, a complete fast and memory-efficient truss-environment modelling method with more accurate precision for climbing robot systems is proposed that can simply be applied to a pointcloud from the simultaneous localization and mapping (SLAM) by just using an RGB-D sensor, no longer needing any known CAD model.

This paper is divided as follows: Section II states the main member-segmentation problems we have met in the point-cloud of the truss environment and outlines the framework of the whole system. Section III describes the implementation of the algorithm improving member-extraction accuracy. In Section IV, the TEM system is evaluated by the collected datasets, and experiments showed it is feasible for our autonomous climbing.

II. PROBLEM STATEMENT

A. Problem of Initial Modelling

To build an autonomous climbing system, a truss modeling and recognition system was proposed and applied to autonomous climbing in our previous work [12]. Although the truss parametric extraction algorithm (TPEA) in the system is efficient and has low memory consumption, it still has the limitation of member length being segmented incompletely when there were multiple members that had to be segmented in the truss environment. There were two reasons for the limitation. First, for members in the truss, it was unavoidable to have them connected by the linkers. After mapping the whole environment with SLAM, all of these linkers were included in the dense pointcloud model. With TPEA, the plane could be removed from the pointcloud before member segmentation, but the linkers that had a noncylinder surface were still in the pointcloud. Just like other existing algorithms, there was the problem of when members were segmenting by RANSAC or Hough transform, the points with the noncylinder surface gave different votes. Second, the truss pointcloud definitely had cracks at member points once there was occlusion by each member in the pointcloud. In the middle of the member pointcloud surface, there was another member pose. Thus, the whole length of the members could not be detected after segmentation.

Therefore, to provide more accurate member poses in the truss environment for the autonomous path planning of the climbing robot systems, these inevitable segmented shortcomings leading to incomplete members segmented from the previous algorithm had to be solved.

B. System-Overview Framework

Therefore, after initial modelling, a TEM method using a sliding bounding box algorithm was applied to recognize members more completely in the truss, which could represent the parametric expression of the truss environment under the realistic challenging conditions mentioned above. The framework of the system is shown in Figure 2, mainly consisting of two parts. One is how complete members are localized based on results from the initial truss modelling, and the other is a simpler way to detect the endpoints of the members in the pointcloud for the purpose of complete members merging. Both parts are our contributions that can be applied to other algorithm to enhance their segmentation performance, in which cylinder members are segmented incompletely.

III. ALGORITHM IMPLEMENTATION

A. Complete Member Localization

1) Bounding Box Generation: With the results of the initial environment modelling, the parametric expression of members in the truss can be used as input for the box-sliding algorithm. Assuming that there are $\boldsymbol{N}$ members in the envelope cubes,

$$\boldsymbol{C}^{member} = \{A, B, r\} \tag{1}$$

where the $A, B \in \boldsymbol{R}^3, r \in \boldsymbol{R}$ output from the truss pointcloud was incompletely segmented by the occlusion of members or linkers, which actually always happens. With all points of the envelope cube, there is no need for executing another Oriented-Bounding Box (OBB) algorithm because boxes can easily be generated from them (Figure 3 (a)). Thus, every segmented member is surrounded by bounding box $\boldsymbol{\Theta}$, preparing for the next step of our box-sliding algorithm.

(a) Bounding box $\boldsymbol{\Theta}$ generation in red.

(b) Box $\boldsymbol{\Theta}$ sliding along axis of members with two directions; then, two new boxes $\boldsymbol{\Theta}^{new}$ were duplicated in blue by $\boldsymbol{\Theta}$.

(c) Endpoints detected in green by gap $\boldsymbol{G}$ with projecting points on the axis.

(d) Members merged into one by two endpoints e.

Fig. 3: Implementation of box-sliding algorithm.

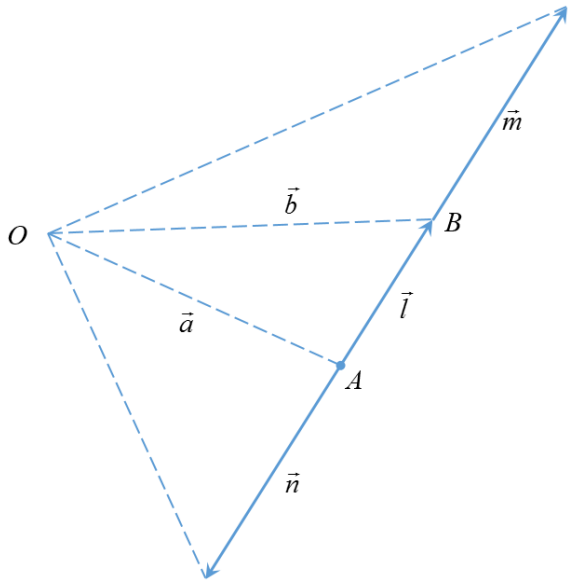


Fig. 4: Member axis-vector sliding.

2) Box Sliding along Members Axis: For each member, the key step for complete localization it is how to find the rest of the members by the box. With the axis vector of the members, the box can easily be moved to the right position in which the points of the remaining part of the members are.

First, the axis vector of member $\vec{l} = \vec{b} - \vec{a}$ was calculated by using two endpoints a and b of the member; then, two new axis vectors $\vec{m}$ and $\vec{n}$ were picked by Formula (2) in Figure 4, prepared for the next step of box sliding,

$$\begin{cases} \vec{m} = \lambda\vec{b} - \vec{a} \\ \vec{n} = \lambda\vec{a} - \vec{b} \end{cases} \tag{2}$$

where $\lambda \geq 2$. After two axis vectors $\vec{m}$ and $\vec{n}$ were calculated, each point of the two boxes $\boldsymbol{\Theta}$ was moved by two axis vectors, which is where the box-sliding algorithm was started for the movement of the box; then, two new boxes $\boldsymbol{\Theta}^{new}$ were generated (Figure 3b).

Due to the connectivity of the incompletely segmented members when new boxes $\boldsymbol{\Theta}^{new}$ were moved and then duplicated along both ends of axis vector $\vec{l}$ of the member, points within new boxes $\boldsymbol{\Theta}^{new}$ were likely to belong to the same member to some extent (Figure 3b).

Once the poses of new box $\boldsymbol{\Theta}^{new}$ were calculated, a crop-box algorithm from the Point Cloud Library (PCL) [13] was applied to extract points in it. In this way, the rest parts of the whole member no longer had to be localized by a segmentation model with a number of parameters to tune. $\boldsymbol{\Theta}^{new}$ can slide without repetition with the same width and height, but λ times (from our experience, λ is about 2.5) of the original-member axis length to completely localize the reset parts of the whole member.

B. Member Merging

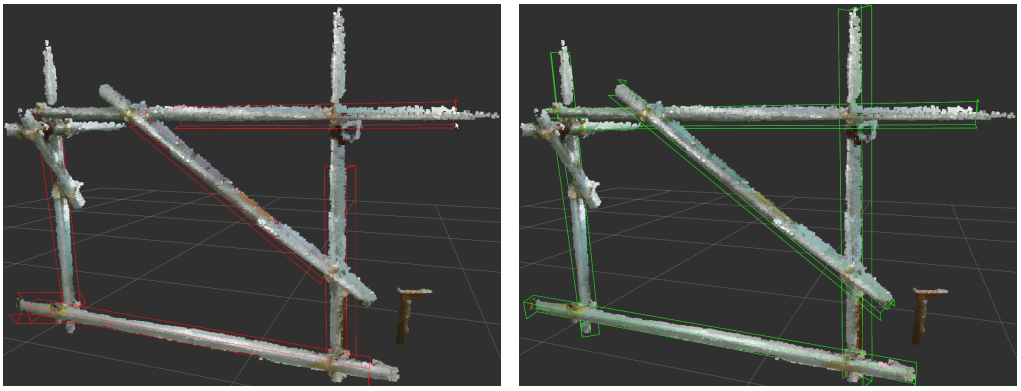
1) Points Projected to Member Axis: Based on prior information, candidate points $\boldsymbol{Q}$ in new bounding boxes $\boldsymbol{\Theta}^{new}$ can be recognized as parts of the members without a modelling algorithm such as RANSAC, while the endpoints of the member can be detected in a less complex way by Euclidean distance $\boldsymbol{D_j}$ of points $\boldsymbol{P}_j$ projected onto the axis of the members from $\boldsymbol{\Theta}^{new}$ by Formula (3)

$$\boldsymbol{P}_j = e + \frac{\vec{p} \cdot \vec{l}}{\vec{l} \cdot \vec{l}} \times \vec{l} \tag{3}$$

where $e = A$ or $e = B$, dependent on the axis vector of the member by which $\boldsymbol{\Theta}^{new}$ is moved. q is one of the total number $\boldsymbol{M}$ points in $\boldsymbol{\Theta}^{new}$, and $\vec{p} = q - e$ represents the vector pointing from e to q.

2) Member-Endpoint Detection: To obtain the pose of the members, an OBB algorithm was first carried out for the pointcloud. After that, the two endpoints of the member were determined by the axis of the member's intersection points with the bounding box. This method is obviously an ideal alternative in the case where there is only one member waiting for being segmented in the pointcloud. Nevertheless, when a member is split into several parts after processing by RANSAC or Hough transform, it is necessary to merge those sections into one member. At this time, if we are using the OBB method to obtain the endpoints of the member, several criteria of merging members were unavoidably introduced in the method, such as the Euclidean distance between every different endpoint of the different sections, and the axis vector of the different sections. Most of the time, the rate of successfully merging different sections of the same member is not high. It is worse even when they can be merged successfully, the pose of a merged member is still inaccurate.

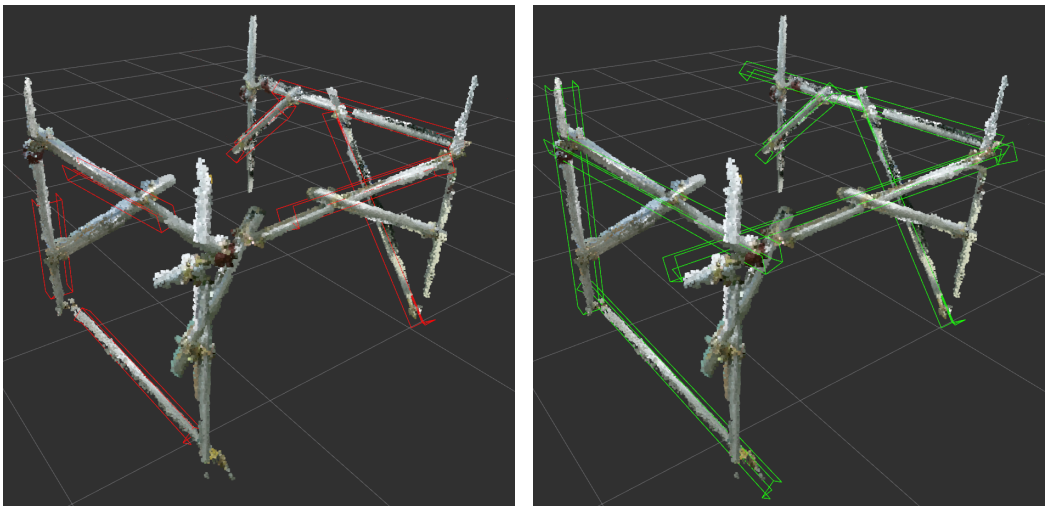
Distinct from the usual way of obtaining the length of members by the OBB algorithm, which could lead to another



(a) Bounding boxes in red are initial modelling result of previous method.

(b) Bounding boxes in green are final modelling result of improved method.

Fig. 5: Bounding boxes are truss-environment modelling results of Experiment 1.



(a) Bounding boxes in red are initial modelling result of previous method.

(b) Bounding boxes in green are final modelling result of improved method.

Fig. 6: Bounding boxes are truss-environment modelling results of Experiment 2.

pose of the members, our endpoint-detection method has two advantages. First, it is based on more accurate initial poses with only one pose of the member from the RANSAC algorithm that can remove outliners by voting. In addition, by projecting points onto the axis that can simplify the merging of members with only one Euclidean distance criterion of the merging members, it obtains the endpoints and, at the same time, expands the length of segmented incomplete members that are based on more accurate initial poses. To sum up, it diminishes the uncertainty and complexity of merging the multiple poses of one member.

The endpoint-detection method in one member was mainly designed as follows:

step 1 Point $\boldsymbol{P_j}$ is every point in $\boldsymbol{Q}$ projected onto the axis of the members.

step 2 Calculate Euclidean distance $\boldsymbol{D_j} =\mid \boldsymbol{P}_j, e \mid$ between $\boldsymbol{P}_j$ and the endpoint of the corresponding axis vector v.

step 3 Find $\boldsymbol{D}_{max}$ and $\boldsymbol{D}_{min}$ in $\boldsymbol{D}$.

step 4 Sort index of $\boldsymbol{P}_j$ by value of $\boldsymbol{D_j}$ from small to large.

step 5 Calculate gap $\boldsymbol{G}_j =\mid \boldsymbol{P}_j, \boldsymbol{P}_{j+1} \mid$ by the Euclidean distance between every two points $\boldsymbol{P}_j$ and $\boldsymbol{P}_{j+1}$ according to sorted index.

step 6 Find $\boldsymbol{G}_{max}$ and $\boldsymbol{G}_{min}$ in $\boldsymbol{G}$.

step 7 The new endpoints of the member along the axis are marked as e if $\boldsymbol{G}_j \geq \frac{\boldsymbol{G}_{max}}{\boldsymbol{G}_{min}} \cdot \varepsilon$, where ε is the threshold set by us.

step 8 Repeat Steps 1 to 7 in the opposite direction.

After two new endpoints of the members are found, the points in the two boxes of a member are merged, as shown in Figure 3d.

The implementation of this sliding bounding box algorithm can refer to the pseudocode of Algorithm 1 in the appendix.

IV. EXPERIMENTS

We carried out two experiments in the truss environment in Figure 7 for Climbot (Figure 8a) [1] , which was developed with a modular approach and implements autonomous climbing according to our previous work [14]. In the evaluation of our method's segmentation, we executed our improved method with the initial truss-environment modelling result that was applied to the dense pointcloud collected by RTABmap with Kinect 2 (Figure 8b). Both the initial model and the final model are shown in Figures 5 and 6 from two experiment datasets whose member numbers were 5 and 7, respectively.

Fig. 7: Our truss enviroment.

(a) Climbot (b) Kinect 2

Fig. 8: Our robot and RGB-D sensor.

Precision analysis of the two modelling methods is illustrated in Figures 9, 10 and 11 calculated by comparisons with Ground Truth.

In our two experiments, the average absolute error of the length of the members was reduced from 520.74 to 63.06 mm, and from 645.5 to 57.81 mm, respectively; average relative error was reduced by our method from 28.49% to 3.46%, and from 38% to 3.7%, respectively. Average absolute error of the axis midpoint distance was reduced by our method from 219.64 to 37 mm, and from 154.89 to 73.84 mm, respectively.Since both modelling methods shared the same axis direction and radius, average axis angle was 1.13°and 2.67°. In addition, there was no need to calculate the average relative radius error.

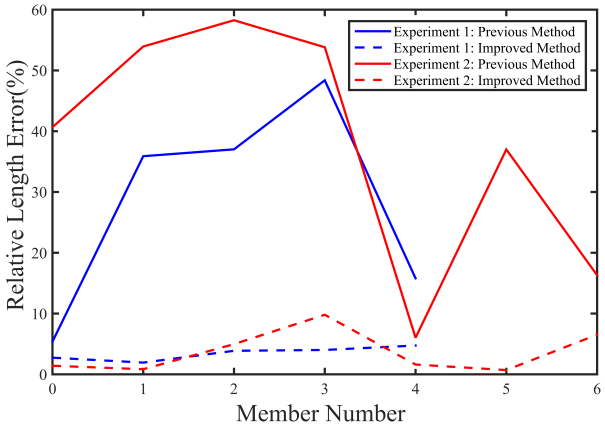


Fig. 9: Length of member estimated with two experiments and two methods.

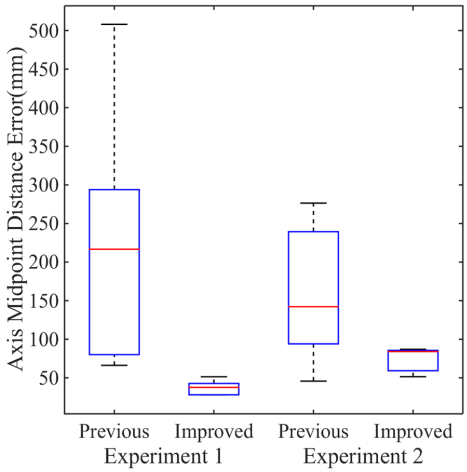


Fig. 10: Midpoint distance of member estimated with two experiments and two methods.

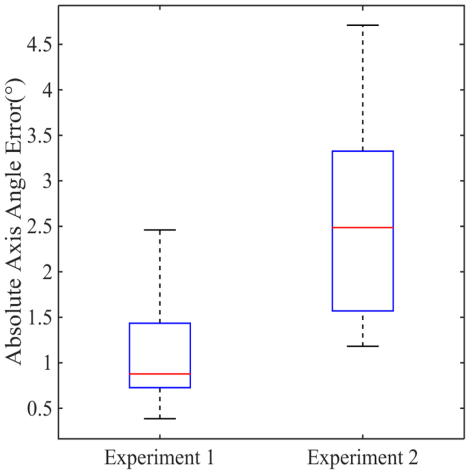


Fig. 11: Orientation of member estimated with two experiments and two methods. Maximal error was 4.7°.

Our method can be testified and it meets our requirements for the Climbot in terms of the three kinds of average errors above on in the experiments. In addition, the results of the experiments illustrate the performance of our TEM method, which was greatly improved and is steadier with smaller average errors than the previous method. All the errors not only just resulted from our two methods, but also from the drift while mapping.

V. CONCLUSIONS

For the purpose of obtaining more accurate poses of members in a truss environment for the biped climbing robots, a modelling method (TEM) was introduced in this paper, which was verified by modelling evaluations with better performance than the initial parametric-expression-representing system whose performance was testified by being applied to autonomous climbing with Climbot. Our work showed that, even under the challenging condition of members being incompletely segmented, using the sliding-box algorithm could solve the problem so that members could be completely recognized. When it comes to member-endpoint detection, the endpoint-detection method we proposed diminished the uncertainty and complexity of merging the multiple poses of one member.

To some extent, there were still some unrecognized members in the truss in the experiments. In the future, a more complex experimental environment can be built to test the effectiveness of the proposed method. Besides, there will be more possible grasping position for our biped climbing robots in the autonomous path planning with the accurate poses of members which can provided to Climbot for scene recognition and relocallizion. Also, we will focus on how to recognize all members in the truss environment and, while Climbot is autonomous climbing in the truss, how to correct the offline model with online sensor data.

REFERENCES

[1] Yisheng Guan, Li Jiang, Haifei Zhu, Wenqiang Wu, Xuefeng Zhou, Hong Zhang, and Xiangmin Zhang. Climbot: a bio-inspired modular biped climbing robotsystem development, climbing gaits, and experiments. *Journal of Mechanisms and Robotics*, 8(2):021026, 2016.

[2] Cesar Cadena, Luca Carlone, Henry Carrillo, Yasir Latif, Davide Scaramuzza, Jos'e Neira, Ian Reid, and John J Leonard. Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age. *IEEE Transactions on robotics*, 32(6):1309–1332, 2016.

[3] Ming Liu. Robotic online path planning on point cloud. *IEEE transactions on cybernetics*, 46(5):1217–1228, 2015.

[4] Andreas Breitenmoser and Roland Siegwart. Surface reconstruction and path planning for industrial inspection with a climbing robot. In *2012 2nd International Conference on Applied Robotics for the Power Industry (CARPI)*, pages 22–27. IEEE, 2012.

[5] Martin A Fischler and Robert C Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, 1981.

[6] Paul VC Hough. Method and means for recognizing complex patterns, December 18 1962. US Patent 3,069,654.

[7] Ruwen Schnabel, Roland Wahl, and Reinhard Klein. Efficient ransac for point-cloud shape detection. In *Computer graphics forum*, volume 26, pages 214–226. Wiley Online Library, 2007.

[8] Yun-Ting Su and James Bethel. Detection and robust estimation of cylinder features in point clouds. In *ASPRS Conference*, 2010.

[9] Mahmoud Fouad Ahmed, Carl T Haas, and Ralph Haas. Automatic detection of cylindrical objects in built facilities. *Journal of Computing in Civil Engineering*, 28(3):04014009, 2014.

[10] Viorica Pătrăucean, Iro Armeni, Mohammad Nahangi, Jamie Yeung, Ioannis Brilakis, and Carl Haas. State of research in automatic as-built modelling. *Advanced Engineering Informatics*, 29(2):162–171, 2015.

[11] Ashok Kumar Patil, Pavitra Holi, Sang Keun Lee, and Young Ho Chai. An adaptive approach for the reconstruction and modeling of as-built 3d pipelines from point clouds. *Automation in construction*, 75:65–78, 2017.

[12] Weinan Chen, Shichao Gu, Lei Zhu, Hong Zhang, Haifei Zhu, and Yisheng Guan. Representation of truss-style structures for autonomous climbing of biped pole-climbing robots. *Robotics and Autonomous Systems*, 101:126–137, 2018.

[13] Radu Bogdan Rusu and Steve Cousins. 3d is here: Point cloud library (pcl). In *2011 IEEE international conference on robotics and automation*, pages 1–4. IEEE, 2011.

[14] Zhiguang Xiao, Wenqiang Wu, Junjun Wu, Haifei Zhu, Manjia Su, Huaizhu Li, and Yisheng Guan. Gripper self-alignment for autonomous pole-grasping with a biped climbing robot. In *2012 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, pages 181–186. IEEE, 2012.

APPENDIX

Algorithm 1 Box-sliding algorithm.

Require: $\boldsymbol{\Theta}, A, B, \boldsymbol{N}$
for $i = 0$ to $\boldsymbol{N}$ **do**
  $\vec{l} = \vec{b} - \vec{a}$
  $\vec{m} = \lambda\vec{b} - \vec{a}$
  $\vec{n} = \lambda\vec{a} - \vec{b}$
  for $\{\vec{v} = \vec{m}, e_1 = A\}$ to $\{\vec{v} = \vec{n}, e_2 = B\}$ **do**
    get $\boldsymbol{\Theta}^{new}$ from $\boldsymbol{\Theta}$ by moving $\boldsymbol{\Theta}$ to $\vec{v}$
    get $\boldsymbol{Q}$ from $\boldsymbol{\Theta}^{new}$
    $\vec{p} = q - e$
    for $j = 0$ to $\boldsymbol{M}$ **do**
      $\boldsymbol{P}_j = e + \frac{\vec{p}\cdot\vec{l}}{\vec{l}\cdot\vec{l}} \times \vec{l}$
      $\boldsymbol{D}_j = \mid \boldsymbol{P}_j, e \mid$
      find the $\boldsymbol{D}_{min}, \boldsymbol{D}_{max}$
      Sort the index of $\boldsymbol{P}_j$
      $\boldsymbol{G}_j = \mid \boldsymbol{P}_j, \boldsymbol{P}_{j+1} \mid$
      find the $\boldsymbol{G}_{min}, \boldsymbol{G}_{max}$
      if $\boldsymbol{G}_j \geq \frac{\boldsymbol{G}_{max}}{\boldsymbol{G}_{min}} \cdot \varepsilon$ **then**
        $e = \boldsymbol{G}_j$
      end if
    end for
  end for
end for
return e_1, e_2