Natural Landmark Extraction in Cluttered Forested Environments

Meng Song , Fengchi Sun, Karl Iagnemma

***Abstract*— In this paper, a new systematical method for extracting tree trunks landmarks from 3D point clouds of cluttered forested environments is proposed. This purely geometric method is established on scene understanding and automatic analysis of trees. The pipeline of our method includes three steps. First, the raw point clouds are segmented by utilizing the circular shape of trees, and segments are grouped into tree sections based on the principle of spatial proximity. Second, circles and axis are extracted from tree sections which are subject to loss of shape information. Third, by clustering and integrating the tree sections resulted from various space inconsistencies, straight tree trunks landmarks are finally formed for future localization. The experimental results from real forested environments are presented.**

I. Introduction

Accurate self localization is a fundamental ability of mobile robot to fulfill tasks in any environment. However, challenges arising from the forested environments leave localization in these specific environments an open problem. In forested areas, GPS cannot be reliably used since its signals are often occluded by dense foliage. The rough, even highly sloped terrain breaks the flat world assumption in 2D navigation, and calls for the accurate full 6D pose estimation. Nevertheless, this problem cannot be solved by 3D dead reckoning alone since the odometry and inertia sensors suffer from large accumulating errors.

In this situation, laser range finders become the most promising sensors for pose correcting purpose since they provide distance measurements at high precision and large field of view, and are not subject to illumination variation. Therefore, we employ a 3D laser sensor to perceive the forest environments, and plan to address the 6D localization problem based on scan matching.

In these years, a lot of work has been done on point-level 3D scan matching. For example, [1] presented a solution to the 6D SLAM problem by using the popular ICP method. In [2], ICP based registration of range scans was used to refine the initial pose estimation for outdoor mapping. However, these point wise matching approaches always require large time cost and expensive computation. [3], [4] adopted a branch-and-bound strategy for the translation part of the transformation, but they couldn't give the solution for all rest freedoms by registration. Therefore, high level features are expected to be reliably extracted from the forested environments to facilitate scan matching.

Driven by this need, we select the tree trunks as landmarks and try to extract theirs features since they are the most regular and distinguishable objects in this kind of highly unstructured environment. In literatures, many algorithms have been proposed to extract features from environments. In the HAYAI algorithm [5], extreme values corresponding to corners are extracted from the polar coordinates of each 2D scan and matched to each other for localization. Nevertheless, it worked best in a planar indoor or semi structured environment. [6], [7] performed 2D SLAM algorithm by using tree trunks as circular landmarks. [8] extracted environment models from range images, and utilized them in matching successive views for localization. However, the proposed scene modeling method was an intrinsic one since it represented nature objects as ellipsoids and superquadrics. The system in [9] presented a 2D SLAM algorithm for semi-structured outdoor environments by extracting obstacle points as landmarks.

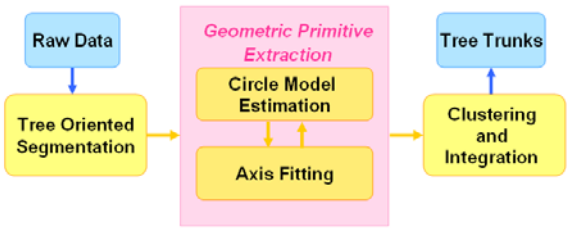


Fig. 1. The framework of feature extraction process

Different from these algorithms, we extract tree trunks from 3D scans of the forested scenes. The trunks are abstracted in a compact and informative way by estimating circles and axes from the partial and corrupted range data. These geometric primitives can then be directly used for feature association and scan transformation.

Our method proceeds in there steps, as outlined in Fig. 1. First, tree sections are distinguished from the rest of raw scene range data by segmentation. Then, a group of circles and one axis are fitted from each tree section. Finally, tree sections are selected and integrated into straight tree trunks. The main contribution of this method is that it captures the nature characteristics of trees, and allows us to successfully perform the high-level feature extraction and modeling on the range data lack of shape information.

The remainder of the paper is organized as follows. In Section II, 3D laser range scans used in this work are described with the acquisition manner. Section III-V present the three steps of our approach in detail. The results of experiments conducted in real forested environments are

M. Song, Fengchi Sun are with the College of Software, Nankai University, Tianjin, 300071, China, songmeng86@hotmail.com, fengchisun@nankai.edu.cn

Karl Iagnemma is with the Department of Mechanical Engineering at Massachusetts Institute of Technology, Cambridge, MA, 02139, USA, kdi@mit.edu

This work is supported by National Science Foundation of China (61175083, 61175085) and Robot Mobility Group in Massachusetts Institute of Technology at Boston, USA.

presented in Section VI. Finally, the work is concluded in Section VII.

II. 3D Laser Range Data Description

The 3D range sensor we use is a nodding device constructed from 2D laser range finder Hokuyo UTM-30LX/LN. A motor driven platform rotates the laser about the Y axis of device coordinate system U, thus yields 180 2D scans per full revolution. The 2D laser itself returns range measurements over 270 degree with angular resolution of 0.25 degrees (Fig. 2). Therefore, a 3D scan consists of 180×1081 or 194580 measured points.

Because of the sampling manner inherent from nodding sensor, the 2D scan planes within 3D scan are not parallel, nor evenly spaced to each other, thus produce scan patterns totally different from range images. In vertical direction, nodding angle ϕ is employed to describe the angle between current 2D scan plane and the horizontal one. By arranging the points of 3D scan S in the form of 2D matrix, each row R represents a 2D scan composed in S with descending ϕ. Therefore, the $N \times M$ 3D point clouds can be defined as $S = \{R_1, R_2, \cdots, R_N\}$, where $\phi_1 > \phi_2 > \cdots > \phi_N$, and $R_i = \{P_{i1}, P_{i2}, \cdots, P_{iM}\}$.

The local coordinate system L is attached to scan S. Its origin is fixed at the sensor's current position. The direction of X axis is set to be the robot's current heading. Besides, the direction of Z axis is set to be the opposite of gravity.

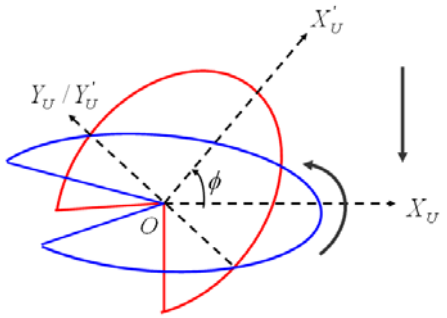


Fig. 2. The nodding 3D scanner. To acquire a 3D scan, 2D laser is tilted from 90 degrees (upwards looking) to -90 degrees (downwards looking). In 2D scan frame U, the X axis is always along the robot's heading direction. Data on 2D scan plane is acquired in a counter-clockwise direction from -135 degrees to 135 degrees.

III. Tree Oriented Segmentation

The tree oriented segmentation of raw 3D point clouds aims to find the 3D point groups related to certain trees from the raw point clouds. To achieve this goal, each 2D scan is first separated into segments. Then the neighboring segments are assembled into tree sections in the vertical direction. The sections hitting trunks or large branches are preserved as the end results of this step. Therefore, the algorithm of tree oriented segmentation is presented as following two parts.

A. Segmentation of 2D Scans

To partition the 2D scan into segments, the discontinuity of scan data is first considered. If one or more laser beams existing between two points don't hit any object, these two points certainly belong to different object surfaces, thus segments are separated.

For consecutive points P_{ij} and P_{ij+1} on the ith 2D scan R_i, we separate them when they are acquired from two trees, of which one tree is occluded by the other. This situation is illustrated in Fig. 3.

Due to the fact that trees can be approximately seen as cylinders, segments are viewed as intersections of cylinders. Noting that any cross section of a cylinder corresponds to the same circular base, we project the segments to the X-Y plane of frame L. After projection, segments are distributed on circle arcs no matter they are tilted or horizontal (Fig. 4).

We denote the projected points of P_{ij} and P_{ij+1} as $P_{ij}^{'}$ and $P_{ij+1}^{'}$. If P_{ij} and P_{ij+1} are acquired from the same tree, $P_{ij}^{'}$ and $P_{ij+1}^{'}$ should lie within the distance threshold $D_t(P_{ij}^{'}, P_{ij+1}^{'})$. That is to say, $D_t(P_{ij}^{'}, P_{ij+1}^{'})$ is the largest possible distance of two points located on the same projected circle.

To determine when $D_t(P_{ij}^{'}, P_{ij+1}^{'})$ is reached, the circle radius and the locations of two points have to be considered. In forested scenes, the acceptable maximal trunk radius is denoted as $R_{\max}$ ($R_{\max} = 0.5m$ for our experiments). $D_t(P_{ij}^{'}, P_{ij+1}^{'})$ is equal to the distance between $P_{ij}^{'}$ and $P_{ij+1}^{'}$ when one of their respective projected laser beams is tangent to the circle with radius $R_{\max}$.

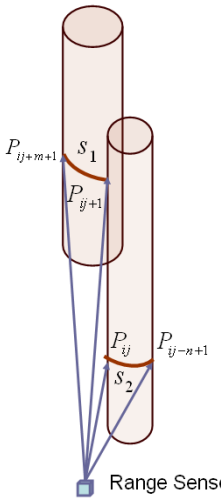


Fig. 3. Illustration of the case in which consecutive points P_{ij} and P_{ij+1} should be segmented. This happens when one tree locating between the sensor and the other tree causes occlusion. Points hitting two trees are separated into segments s_1 and s_2, where P_{ij} is the end point of s_2, and P_{ij+1} is the start point of s_1.

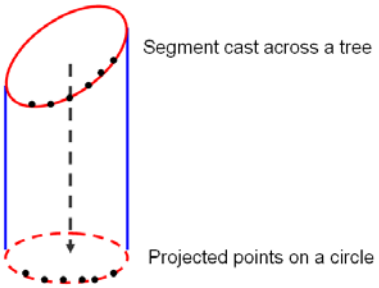


Fig. 4. Segment projection

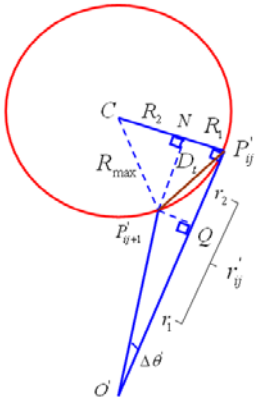


Fig. 5. Illustration of the geometric relationships for solving $D_t(P'_{ij}, P'_{ij+1})$. By setting $R_{\max} = R_1 + R_2$ and $r'_{ij} = r_1 + r_2$, we can derive that in $\Delta QP'_{ij}P'_{ij+1}$, $D_t^2 = R_1^2 + r_2^2$, in $\Delta CNP'_{ij+1}$, $R_{\max}^2 = R_2^2 + r_2^2$ and in $\Delta QO'P'_{ij+1}$, $R_1 = r_1 \cdot \tan\Delta\theta'$. $D_t(P'_{ij}, P'_{ij+1})$ is thus obtained by solving these equations.

We assume that P'_{ij} is the point of tangency. According to the geometric relationships illustrated in Fig. 5, we can obtain that

$$D_t = \sqrt{2\cdot[R\cdot r\cdot\tan\delta + \frac{R\cdot\tan\delta\cdot(\sqrt{R^2+\tan^2\delta+2\cdot R\cdot r\cdot\tan\delta - r^2\cdot\tan^2\delta} - r\cdot\tan^2\delta + R\cdot\tan\delta)}{\tan^2\delta+1}]} \quad (1)$$

In (1), D_t, r, δ and R are the simplified representations of $D_t(P'_{ij}, P'_{ij+1})$, r'_{ij}, $\Delta\theta'$ and $R_{\max}$ respectively. $\Delta\theta'$ is the angle between $O'P'_{ij}$ and $O'P'_{ij+1}$, where O' is the projected position of the sensor. r'_{ij} is the length of $O'P'_{ij}$.

B. Tree Section Assembling

In this part, we will group the segments together based on the principle of spatial proximity. Furthermore, tree sections representing trunks or salient branches will be identified by filtering technique.

To measure the spatial relationship of segments in 3D space, the position of segment has to be described at first. Here, s_{ij} is the *jth* segment on the *ith* 2D scan. In the direction of Z axis, segments on the same 2D scan plane can differ in height. Therefore, the index i is used to locate s_{ij} vertically. On horizontal plane X-Y of frame L, s_{ij} is projected to be a 2D point set characterized by width w_{ij} and mass center μ_{ij}. w_{ij} is the Euclidean distance between the start and end point of s_{ij} on X-Y plane. μ_{ij} is the mean X-Y location of all points assigned to s_{ij}.

The tree section can be seen as a sequence of spatially close segments. The segments in the same tree section should come from consecutive 2D scan planes. Therefore, the *kth* tree section is represented as $t_k = \{s^k_{i_1 j_1}, s^k_{i_2 j_2}, \cdots, s^k_{i_n j_n}\}$, $\phi_{i_1} < \phi_{i_2} < \cdots < \phi_{i_n}$, where $i_1 > i_2 > \cdots > i_n$ are n consecutive 2D scan plane indices.

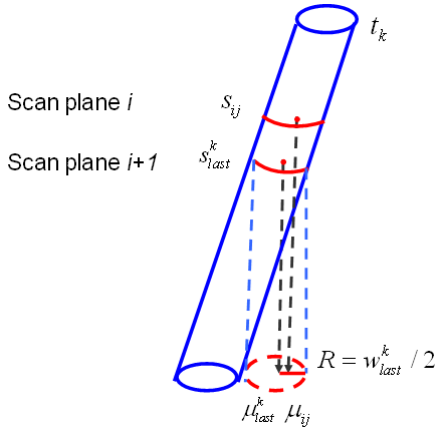


Fig. 6. s_{ij} is in the neighborhood of s^k_{last} on tree section t_k

To construct tree sections, a filtering strategy is adopted to remove the long segments that are acquired from ground, dense foliage, or other large solid objects. Segment s_{ij} is preserved if its width is within the diameter of the largest acceptable trunk, that is

$$w_{ij} < 2\cdot R_{\max} \quad (2)$$

Next, tree sections will be constructed by merging the nearby segments layer by layer. This assembling algorithm starts with an empty tree section set T, and performs the section searching operation for each segment s_{ij} on 2D scan planes in the order of ascending ϕ. To find the tree section s_{ij} belongs to, we will check every tree section t_k in current T. If there is no section meeting the proximity requirement, we add $t_{new} = \{s_{ij}\}$ to T as a new section. Otherwise, s_{ij} is appended to its tree section as the last segment.

In tree section t_k, the last segment s^k_{last} is the one having the largest nodding angle. Therefore, based on the searching direction of the algorithm, s^k_{last} is also the closest segment to s_{ij} among all segments of t_k vertically. Thus, if s_{ij} belongs to t_k, s_{ij} should lie within the neighborhood of s^k_{last}.

According to the segment's position descriptor defined above, s_{ij} is considered to be lie in the proximity of s^k_{last} only when the following constraints are satisfied.

(1) $i^k_{last} - i = 1$

(2) $D(\mu^k_{last}, \mu_{ij}) \le w^k_{last}/2$

In constraint (1), s^k_{last} and s_{ij} are required to lie on the neighboring scan planes. In constraint (2), the horizontal mass

center of s_{ij} need to fall into the circle region restricted by s_{last}^{k} on X-Y plane. The circle center is just the mass center of s_{last}^{k}, and the radius is half of s_{last}^{k}'s width (Fig. 6).

Once the tree section construction has been completed, the sections that are clearly not acquired from trunks or salient branches should be filtered out. These undesirable sections are usually composed of small segments formed by hitting slender branches or sparse leaves. Since these scattered segments are distributed randomly in the space, it is hardly possible for them to constitute a statically high section. Therefore, the sections not high enough are removed.

IV. GEOMETRIC PRIMITIVE EXTRACTION

In this step, our purpose is to extract the underlying geometric models from tree sections. Two iterative phases will be carried out. First, circle models are estimated from the segments of tree sections. Second, tree section axes are fitted using the circle centers.

A. Circle Model Estimation

As a result of the scanning manner, the segments on tree sections are not always normal to the tree axis. Therefore, points on segments are first projected to the tree section's projection plane. All of these planes should pass through the sensor's position, and be perpendicular to their tree sections' axes. Since we don't know the tree sections' orientation in advance, the normals of these planes are initialized as the direction of Z axis in frame L.

However, directly fitting the projected points to circles will produce circle radiuses that are much larger than they actually are. This is because the raw measured data cannot provide sufficient shape information for circle modeling. In the forested scene scan, it can be observed that the projected points on the circle only cover a partial circumference, and its distribution is uneven. The central points facing the sensor are much denser than the edge ones. However, these sparse points locating on edge are more important in determining the circle's curvature. Unfortunately, they are corrupted heavily by the factors, such as tree skin roughness and measurement noise. Hence, the segment's projection tends to be distorted as a linear one. Additionally, because of the radial scanning manner in nature, the number of points will decrease dramatically as the distance from sensor increase.

To solve this problem, we will first estimate circle radius from the angular span of the projected points. Then the point set is fitted to get the unknown center with estimated radius.

Assume that the projected segment has n points, and is given by $s^{'} = \{P_{ij+1}^{'}, P_{ij+2}^{'}, \cdots, P_{ij+n}^{'}\}$. R is the circle radius to be estimated which satisfies $R_{sub} \le R < R_{\sup}$, where R_{sub} and $R_{\sup}$ are lower and upper bounds of R. The values of these two bounds only depend on the start point $P_{ij+1}^{'}$ and end point $P_{ij+n}^{'}$. The cases where R_{sub} and $R_{\sup}$ are achieved will be illustrated in Fig. 7 and discussed as follows.

In Fig. 7, O denotes the sensor's position, and $C^{'}$ denotes the projected circle center. By fixing the origin at O, a 2D coordinate system is set on the projection plane. The polar coordinates of $P_{ij+1}^{'}$ and $P_{ij+n}^{'}$ are thus denoted as $(r_{ij+1}^{'}, \theta_{ij+1}^{'})$ and $(r_{ij+n}^{'}, \theta_{ij+n}^{'})$, which will be utilized to express unknowns R_{sub} and $R_{\sup}$ later.

As the minimal possible radius, R_{sub} is achieved when the projected scan lines of $P_{ij+1}^{'}$ and $P_{ij+n}^{'}$ are both tangent to the circle. Therefore, in Fig. 7 (a), we have $\alpha_{sub} = \dfrac{\theta_{ij+n}^{'} - \theta_{ij+1}^{'}}{2}$, and r_{sub} is the length of the tangent segment. Though in practice, $r_{ij+1}^{'}$ and $r_{ij+n}^{'}$ may not be equal as this case, the average of them can be used to estimate r_{sub}, namely $r_{sub} = \dfrac{r_{ij+1}^{'} + r_{ij+n}^{'}}{2}$. Finally, based on the property of tangent line, we obtain that

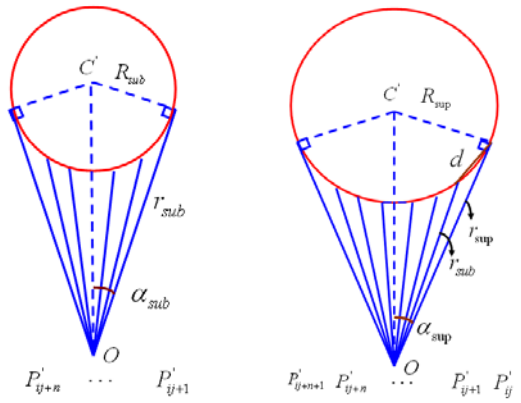
$$R_{sub} = r_{sub} \cdot \tan \alpha_{sub} . \tag{3}$$

Unlike R_{sub}, $R_{\sup}$ is the maximal radius of circle model which cannot be reached. $R_{\sup}$ presents when the projected scan lines of $P_{ij}^{'}$ and $P_{ij+n+1}^{'}$ are both tangent to the circle. This means that $s^{'}$ has to incorporate two more points which do not belong to it. Hence, in Fig. 7(b), we have $\alpha_{\sup} = \dfrac{\theta_{ij+n+1}^{'} - \theta_{ij}^{'}}{2}$, and $r_{\sup}$ is the length of the tangent segment. By formulating the geometric relationships in $\Delta P_{ij+1}^{'} C^{'} P_{ij}^{'}$, $\Delta O C^{'} P_{ij}^{'}$ and $\Delta P_{ij+1}^{'} O P_{ij}^{'}$ of Fig. 4(b), we have

$$\begin{cases} \angle P_{ij+1}^{'} P_{ij}^{'} O = \dfrac{1}{2} \angle P_{ij+1}^{'} C^{'} P_{ij}^{'} \\ \dfrac{d}{2} = R_{\sup} \cdot \sin \dfrac{\angle P_{ij+1}^{'} C^{'} P_{ij}^{'}}{2} \\ \dfrac{\sin \angle P_{ij+1}^{'} P_{ij}^{'} O}{r_{sub}} = \dfrac{\sin \Delta\theta^{'}}{d} \\ R_{\sup} = r_{\sup} \cdot \tan \alpha_{\sup} \\ d^2 = 2 R_{\sup} r_{sub} \sin \Delta\theta^{'} \\ d^2 = r_{\sup}^2 + r_{sub}^2 - 2 r_{\sup} r_{sub} \cos \Delta\theta^{'} \end{cases} \tag{4}$$

Solving (4) and rewriting r_{sub}, $\Delta\theta^{'}$, $\alpha_{\sup}$ as simple symbols r, δ, α, we obtain that

$$R_{\sup} = r \cdot \cos\delta \cdot \tan\alpha + r \cdot \tan^2\alpha \cdot \sin\delta \pm r \cdot \tan\alpha \cdot \sqrt{\cos^2\delta + 2 \cdot \cos\delta \cdot \tan\alpha \cdot \sin\delta + \tan^2\alpha \cdot \sin^2\delta - 1} \tag{5}$$



(a)Lower bound (b) Upper bound

Fig. 7. Lower and upper bounds of the circle radius

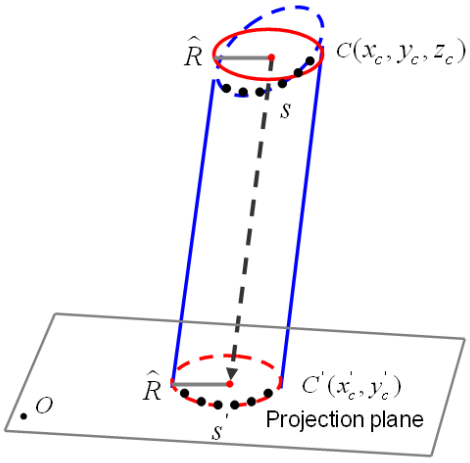


Fig. 8. Circle model projection

Due to the symmetry between the two pairs of neighbor points P'_{ij} , P'_{ij+1} and P'_{ij+n} , P'_{ij+n+1} , there are two value possibilities for $\Delta\theta'$ and r_{sub} . First, $\Delta\theta' = \theta'_{ij+1} - \theta'_{ij}$ and $r_{sub} = r'_{ij+1}$. Second, $\Delta\theta' = \theta'_{ij+n+1} - \theta'_{ij+n}$ and $r_{sub} = r'_{ij+n}$. Substitute $\Delta\theta'$ and r_{sub} of two cases into (5) respectively, we get $R_{\sup 1}$, $R_{\sup 2}$. Then, $R_{\sup}$ is calculated as $R_{\sup} = \dfrac{R_{\sup 1} + R_{\sup 2}}{2}$.

By assuming that P'_{ij+1} and P'_{ij+n} may locate at any point between their respective bounds with equal possibility, the estimation of R is shown as

$$\hat{R} = \frac{1}{2}(R_{\sup} + R_{sub}) \tag{6}$$

On the projection plane, $C'(x'_c, y'_c)$ can be calculated using the least square estimation (LSM). With the known circle radius $\hat{R}$, the error function is expressed as

$$J_{LSM} = \sum_{i=1}^{n} (\sqrt{(x'_i - x'_c)^2 + (y'_i - y'_c)^2} - \hat{R})^2 \tag{7}$$

where (x'_i, y'_i) is the coordinate of the *ith* point in s' with respect to the projection coordinate system.

The original 3D segment s has the circle model with center $C(x_c, y_c, z_c)$ and radius $\hat{R}$. The coordinates (x_c, y_c) can be retrieved by transforming $C'(x'_c, y'_c)$ from the projection coordinate system to local frame L . z_c is defined as the z coordinate of the segment's mass center (Fig. 8).

To test the validity of circle fitting, we notice the fact that from the view of robot, the visible parts of trees should always have a convex contour. Therefore, the circle center C and robot's sensor O should lie on the different sides of the segment.

B. *Tree Section Axis Fitting*

After extracting the circle models from segments, we fit all valid circle centers of tree section t_k to centerline l_k using LSE. The axis of t_k is denoted as line segment d_k . d_k is a part of line l_k bounded by endpoints P^k_{bottom} and P^k_{top} . These two points are equal in z coordinate to the lowest and highest points on t_k respectively. Moreover, R_k , the radius of t_k , is defined as the mean of valid circles' radiuses.

Once the axis has been estimated, the tree section orientation will become the new normal of t_k 's projection plane. For accuracy, the algorithm will iteratively perform the circle model fitting step and the axis fitting step until the orientation stabilizes.

V. CLUSTERING AND INTEGRATION OF TREE SECTIONS

In order to facilitate robot localization, each tree should provide only one trunk section as the natural feature to uniquely represent itself. Relying on the spatial relationships of axes, we will first cluster the tree sections, and then integrate the related sections into the target trunks.

A. *Clustering*

Because of the spatial discontinuity, some trees have been partitioned into more than one tree section in the previous process. To group them together, our algorithm will assign every section into the cluster corresponding to the tree it belongs to.

Before clustering, the distance measure should be introduced to describe the spatial relationships of tree sections. Based on the geometric abstraction, the distance between tree section t_i and t_j is defined as the distance between their axes, that is

$$D(t_i, t_j) = D(d_i, d_j) \tag{8}$$

Suppose that the segment perpendicular to both line l_i and line l_j intersects them at point P and Q respectively. If P is on d_i and Q is on d_j , the distance between axes is equal to the distance between corresponding centerlines, that is

$$D(d_i, d_j) = D(l_i, l_j) \tag{9}$$

Otherwise, assume that $t_i \le t_j$, then we have

$$D(d_i, d_j) = \min\{D(P^i_{top}, l_j), D(P^j_{bottom}, l_i)\} \tag{10}$$

Here $t_i \le t_j$ means that the lowest point on t_i is not higher than the lowest point on t_j. In this situation, the distance between axes is defined as the smaller one of the distance from d_i's upper endpoint to line l_j and the distance from d_j's lower endpoint to line l_i (Fig. 9).

Moreover, assume that the *kth* tree cluster contains n_k tree sections, $T_k = \{t_1^k, t_2^k, \cdots, t_{n_k}^k\}$, $t_1^k \le t_2^k \le \cdots \le t_{n_k}^k$. The distance from tree section t_i to cluster T_k is defined as the distance between t_i and its closest tree section t_j^k in T_k, that is

$$D(t_i, T_k) = \min_{1 \le j \le n_k} D(t_i, t_j^k) \tag{11}$$

Obviously, these distance definitions reflect our emphasis on tree sections' endpoints closely to the discontinuous part of the tree. This is because the sections tend to get closer to each other when they are approaching the discontinuity's location. Therefore, we can analyze the upper bound of the distance between tree sections by classifying trees into three categories according to the type of discontinuity (Fig. 10).

(1) Broken: Tree point clouds are broken into two sections due to occlusion or point absence. The distance between the sections is nearly zero.

(2) Bending: The abruptly change of tree orientation forces the tree to split into two straight sections. The distance between them will not exceed the larger one of their radiuses.

(3) Branched: Branches may grow from top or the side of the trunk. When the former occurs, the distance between branch and trunk is not larger than the radius of trunk. While in the later case, their distance is up to the sum of their radiuses.

From the analysis above, we can conclude that if section t_i belongs to cluster T_k, the distance from t_i to its closest section t_j^k in T_k will not beyond the sum of their largest circle radiuses, that is

$$D(t_i, T_k) \le R_{\max}^i + R_{\max}^j \tag{12}$$

As the first step of the clustering algorithm, all tree sections are sorted to ensure that the tree sections on the same tree will enter the cluster bottom up. The ordered tree section list is $T = \{t_i \mid i = 1, \cdots, N_T\}$, where $t_1 \le t_2 \le \cdots \le t_{N_T}$. The set of clusters C is initialized to be empty. Next, the algorithm traverses T in sequence and tries to allocate every tree section t_i to its cluster. If a cluster T_k in current T satisfies (12), t_i will be inserted to T_k. Otherwise, we add $T_{new} = \{t_i\}$ to C as a new cluster.

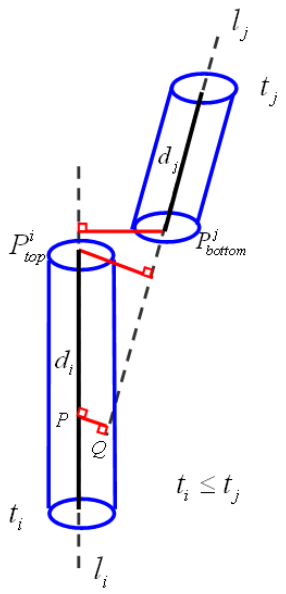


Fig. 9. The distance between axes d_i and d_j when P or Q is not on its corresponding axis

In this algorithm, the sections in cluster are required to be stored bottom up, thus inserting a tree section into the proper position of the cluster is closely related to the tree's structure. We associate every tree cluster T_k with a label F_k, such that F_k=*WHOLE* if T_k has only one section, F_k=NON-*BRANCHED* if T_k describes a non-branched tree which has more than one section, and F_k=*BRANCHED* if T_k represents a branched tree.

Specifically, the trunk sections of a branched tree are stored upwards until tree branches for the first time. The rest sections are appended to T_k in turn. The lowest trunk section with branchpoint is denoted as t_{branch}^k.

To insert section t_i into an ordered and nonempty cluster T_k, the following algorithm is proposed.

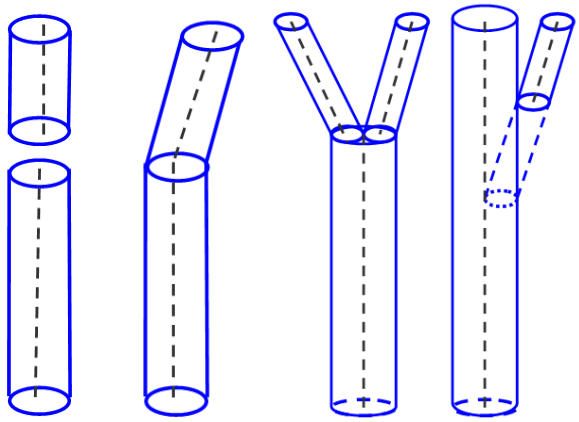
(1) If $F_k \ne BRANCHED$ and t_i is above the highest tree section $t_{n_k}^k$ in T_k, then append t_i to T_k, set F_k = NON-*BRANCHED*. Otherwise, enter the next step.

(2) If $F_k = BRANCHED$, append t_i to T_k. Otherwise, traverse T_k to find t_j^k first satisfying the requirement that the bottom segment on t_i is not lower than the top segment on t_j^k, and is not higher than the bottom segment on t_j^k. Append t_i to T_k, set $F_k = BRANCHED$ and $t_{branch}^k = t_j^k$.

B. Integration

Given an ordered cluster T_k, our purpose is to integrate the desired tree sections of T_k into a straight tree trunk. In the cluster with $F_k = BRANCHED$, all sections after t_{branch}^k are removed first since they are acquired from branches. Then, for each cluster with more than one section, a bottom up merging operation is performed with orientation change detector. Tree

sections are connected one by one until the orientation difference between the current section and the previous one is too large. After connection, we re-estimate the axis of the integrated trunk, and finally take this trunk feature as the end result.



(a)Broken (b)Bending (c)Branched

Fig. 10. Tree sections with space inconsistency

VI. EXPERIMENTAL RESULTS

The proposed feature extraction approach has been tested in a forest located in the greater Boston area. The 3D laser data is collected in a stop-and-scan manner. This forested environment provides a series of testing scenes varying in degree of roughness and clutter to a large extent. Fig. 11 shows a camera view of the testing environment.

As an example to illustrate the processing stages in our approach, tree trunks are extracted from a typical scan of the testing scene. Fig. 12 shows the detailed experimental results from a closer view.

To demonstrate the adaptability of our method to the diverse terrains, the feature extraction results of three different scenes are depicted in Fig. 13 by comparison with the original scans. The first scene represents the normal woodland area which can be used as a baseline for the other more complex scenes. The second scene is a highly cluttered forest containing dense shrubs and leaves. The last scene presents the trees on a large slope.

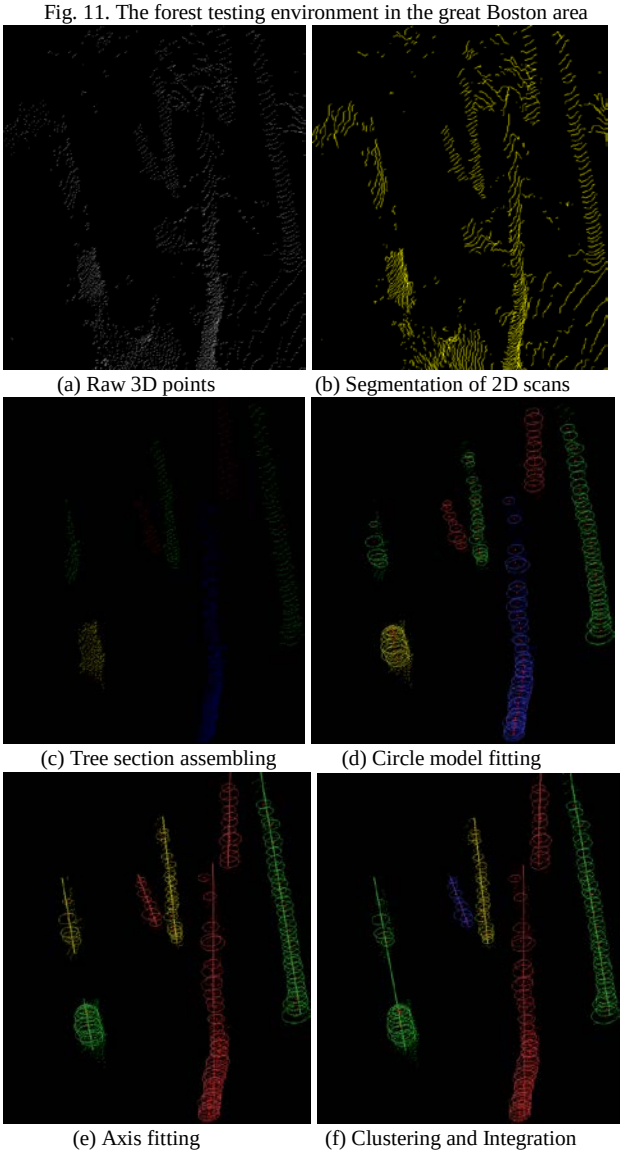
The accuracy of our algorithm is evaluated by computing the fitting errors between the raw points and the axis of the tree trunk as follows

$$e_k = \frac{\sum_{i=1}^{n} |\mathbf{a_k} \times (\mathbf{p_i} - \mathbf{p})|}{n} - R_k \qquad (13)$$

where $\mathbf{p}$ is a point on the axis of trunk t_k. $\mathbf{p_i}$ is the *ith* range point acquired from t_k. $\mathbf{a_k}$ is the direction vector of t_k. $|\mathbf{a_k} \times (\mathbf{p_i} - \mathbf{p})|$ indicates the distance from $\mathbf{p_i}$ to the axis. R_k corresponds to the radius of t_k. Therefore, feature fitting error e_k means the average distance from raw points to the trunk surface.

We randomly select 100 scans from the data set covering 12 scenes in the testing environment. The average e_k of all 2103 trunk features is 6.27 mm. $|e_k|$ of 88.96% features are below 5mm.

Fig. 11. The forest testing environment in the great Boston area



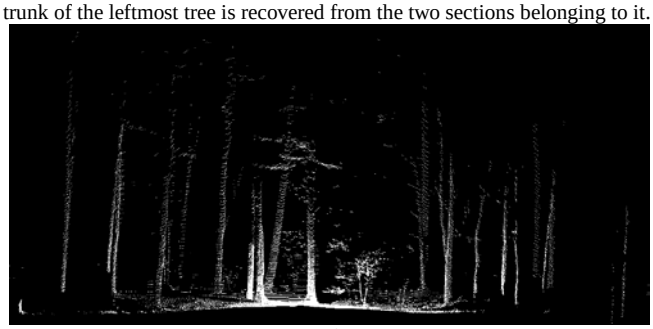
(a) Raw 3D points (b) Segmentation of 2D scans

(c) Tree section assembling (d) Circle model fitting

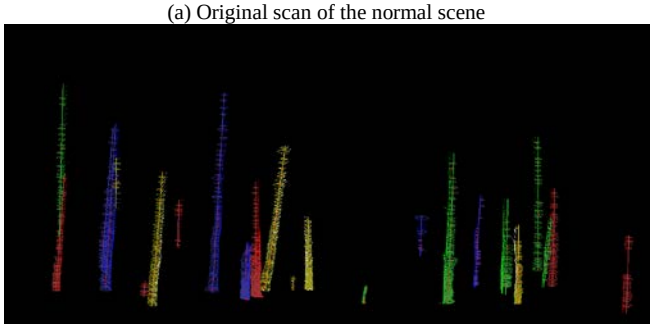
(e) Axis fitting (f) Clustering and Integration

Fig. 12. Experimental results after performing each stage of the feature extraction approach. (a) shows the raw 3D points taken from a part of the testing scan. Six trees can be recognized from (a), and the leftmost one is broken into two parts due to the lack of scan points. In (b), points on the same segment are connected together in yellow. (c) presents the seven tree sections constructed from segments. Different sections are indicated by yellow, red, blue and green in turn. In (d), valid circle models are estimated from segments on the trees. The circle centers are marked in red. In (e), the corresponding tree section axes are fitted. In (f), tree sections are integrated

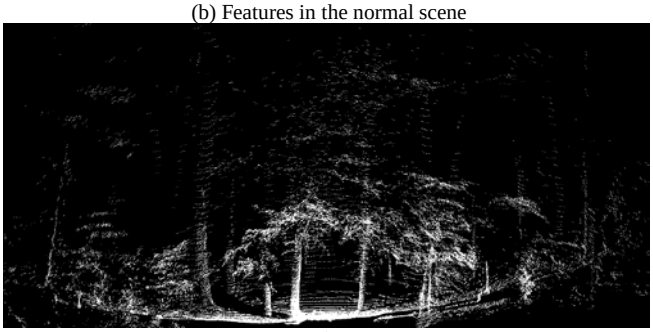
into trunk features, which are discriminated by colors. As a result, the whole trunk of the leftmost tree is recovered from the two sections belonging to it.



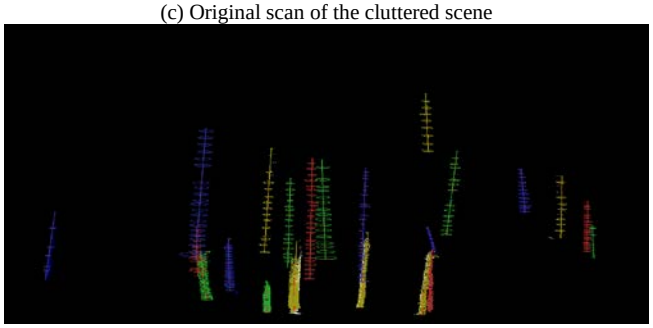
(a) Original scan of the normal scene



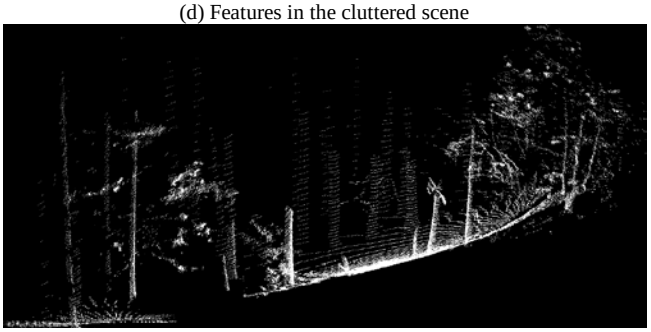
(b) Features in the normal scene



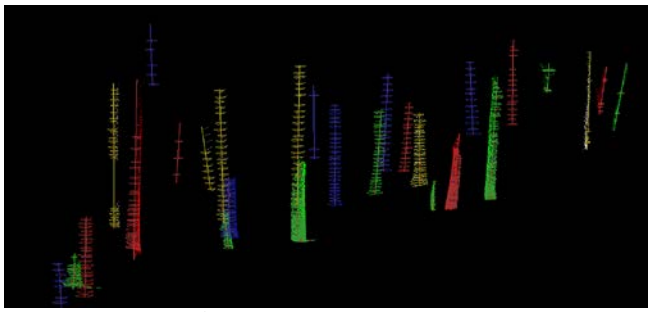
(c) Original scan of the cluttered scene



(d) Features in the cluttered scene



(e) Original scan of the sloped scene



(f) Features in the sloped scene

Fig. 13. Tests in three scenes: normal, cluttered, and sloped

VII. Conclusion

This paper has presented a geometric method to automatically extract tree trunks from cluttered forested environments. The underlying trunk models, circles and axes, are estimated from partial and corrupted 3D scan. Experimental results have shown the method's efficiency and robustness in the cluttered and complicated forest scenes.

Besides benefiting the application in forests, this feature extraction approach can also be extended for extracting column-like objects in indoor or urban environments. Moreover, the scene analysis involved in recognizing and representing these high-level objects is also a preliminary attempt to improve robot's intelligence.

In the future work, we are planning to align the successive 3D scans based on these high-level features. The full 6D transformation is then used to track the robot pose in forested environments.

References

[1] A. Nüchter, K. Lingemann, J. Hertzberg, and H. Surmann, "Heuristic-based laser scan matching for outdoor 6D SLAM," in *28th Annual German Conference on Advances in Artificial Intelligence*, Koblenz, Germany, 2005, pp. 304–319.

[2] D. F. Huber and M. Hebert, "A new approach to 3-D terrain mapping," in *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems*, Kyongju, Korea, 1999, pp. 1121–1127.

[3] C. F. Olson, "Probabilistic self-localization for mobile robots," *IEEE Transactions on Robotics and Automation*, vol. 16, no. 1, pp. 55–66, 2000.

[4] P. Forsman and A. Halme, "3-D mapping of natural environments with trees by means of mobile perception," *IEEE Transaction on Robotics*, vol. 21, no. 3, pp. 482–490, 2005.

[5] K. Lingemann, A. Nüchter, J. Hertzberg, and H. Surmann, "High-speed laser localization for mobile robots," *Journal of Robotics and Autonomous Systems*, vol. 51, no. 4, pp. 275–296, 2005.

[6] J. Guivant, F. Masson, and E. Nebot, "Simultaneous localization and map building using natural features and absolute information," *Journal of Robotics and Autonomous Systems*, vol. 40, no. 2–3, pp.79–90, 2002.

[7] T. Bailey, "Mobile robot localisation and mapping in extensive outdoor environments," Ph.D. dissertation, Australian Centre for Field Robotics, University of Sydney, 2002.

[8] S. Betge-Brezetz, R. Chatila, and M. Devy, "Object-based modeling and localization in natural environments," in *Proc. IEEE International Conference on Robotics and Automation*, Nagoya, Japan, 1995, pp. 2920-2927.

[9] C. Brenneke, O. Wulf, and B. Wagner, "Using 3D Laser Range Data for SLAM in Outdoor Environments," in *Proc. IEEE/RSJ International Conference on Intelligent Robots and System*, Las Vegas, USA, 2003, pp. 188-193.