Object shape classification and scene shape representation for three-dimensional laser scanned outdoor data

Xiaojuan Ning
Xi’an University of Technology
Department of Computer Science Engineering
Xi’an, China
and
CAS Institute of Automation
National Laboratory of Pattern Recognition
Beijing, China
E-mail: ningxiaojuan@xaut.edu.cn

Yinghui Wang
Xi’an University of Technology
Department of Computer Science Engineering
Xi’an, China

Xiaopeng Zhang
CAS Institute of Automation
National Laboratory of Pattern Recognition
Beijing, China
E-mail: xpzhang@nlpr.ia.ac.cn

Abstract. Shape analysis of a three-dimensional (3-D) scene is an important issue and could be widely used for various applications: city planning, robot navigation, virtual tourism, etc. We introduce an approach for understanding the primitive shape of the scene to reveal the semantic scene shape structure and represent the scene using shape elements. The scene objects are labeled and recognized using the geometric and semantic features for each cluster, which is based on the knowledge of scene. Furthermore, the object in scene with a different primitive shape could also be classified and fitted using the Gaussian map of the segmented scene. We demonstrate the presented approach on several complex scenes from laser scanning. According to the experimental result, the proposed method can accurately represent the geometric structure of the 3-D scene.

© 2013 Society of Photo-Optical Instrumentation Engineers (SPIE)
[DOI: 10.1117/1.OE.52.2.024301]

Subject terms: terrestrial laser scanner; point cloud data; primitive shape; shape classification and recognition; scene representation; Gaussian map.

Paper 121045 received Jul. 18, 2012; revised manuscript received Dec. 19, 2012; accepted for publication Jan. 17, 2013; published online Feb. 5, 2013.

1 Introduction

Recently, laser scanner has become a practical way to acquire the three-dimensional (3-D) data of outdoor scene, which opens an unprecedented chance for object segmentation and scene modeling. 3-D objects with different shapes play an important role in constituting a complete scene (such as a city scene containing ground, architecture, trees, cars, etc). Therefore, object shape analysis has received much more attention lately, and it has more applications in such areas as object extraction, shape segmentation, shape modeling, shape morphing, etc.

In recent years, there has been increasing research in high-level analysis of 3-D object shapes. Some methods attempt to infer high-level knowledge of a shape from its geometry. Some methods focus on the shape analysis on single object. Approaches have been proposed for extracting objects from scenes, reconstructing objects (e.g., buildings, trees), and identifying different objects. However, the problem of analyzing the shape of a whole scene has received less attention.

Generally, we can say that an object or a scene could be decomposed into several different shapes, i.e., an object or a scene can be described by the basic shape. In this paper, we investigate a method to identify objects from 3-D data in real world environments, and represent the environment using different shapes. The overview of our algorithm is shown in Fig. 1, and the contribution of our paper could be described in three aspects:

1. A knowledge-based object classification method is proposed, which can effectively recognize different objects in a scene.
2. A Gaussian map-based method for shape recognition is presented to represent different objects.
3. A basic shape representation method is introduced for the whole scene.

A key distinguishing feature of our approach is that it could describe the structure of scene by recognizing the shape elements. We evaluate the presented approach on scenes taken from our laser scanner and demonstrate that the presented method effectively analyzes the structure of 3-D scene or objects.

2 Preliminaries

2.1 Basic Shape

At first, we make a declaration to clearly define the following concepts:

1. Object: a set of segmented points labeled (for instance, wall, tree, etc.)
2. Scene Object: an object instance in the current scene
3. Characterization: a set of low-level and high-level features
4. Feature: a criterion value supposed to be computed from a segmentation process and quantified.
5. Shape: a feature that could best describe an object; here it mainly contains planar, cylindrical, conical, and spherical shapes.
The representation and parameters of geometric shape are shown in Figs. 2 and 3.

### 2.1.1 Planar shape

Generally, a plane can be determined by three points \( p_1, p_2, p_3 \), which constitute the minimal data set when not considering the normals in the points. A more creditable way is to confirm the plane’s normal from normal \( n_1, n_2, n_3 \), and an arbitrary point in the plane. The plane is defined in Eq. (1):

\[
n_x \cdot (x - x_0) + n_y \cdot (y - y_0) + n_z \cdot (z - z_0) = 0, \quad (1)
\]

where \( n = \{n_x, n_y, n_z\} \) represents the normal to the plane, which is a unit vector, and \( p_0 = \{x_0, y_0, z_0\} \) is an arbitrary point in the plane.

The signed distance function for the plane is given by

\[
d(x) = n(x - p) = n_x \cdot (x - x_0) + n_y \cdot (y - y_0) + n_z \cdot (z - z_0) \quad (2)
\]

### 2.1.2 Cylindrical shape

The cylindrical surface is formed by the points at a fixed distance from a given line segment, i.e., the axis of the cylinder. Assume that a cylinder with an axis \( p + \lambda \cdot a, \lambda \in \mathbb{R} \), \( ||a|| = 1 \), \( p = (x_0, y_0, z_0) \) and radius \( r \), the cylindrical surface can be represented by

---

**Fig. 2** Simple geometric shape: (a) solid geometry shape, (b) shape point set.
A circular cone can be represented by

\[ (x - x_0)^2 + (y - y_0)^2 + (z - z_0)^2 - \left[n_x(x - x_0) + n_y(y - y_0) + n_z(z - z_0)\right]^2 - r^2 = 0 \]  

(3)

Then the signed distance to the cylinder can be computed as

\[ d(x) = \sqrt{\|x - p\|^2 - <a, x - p>^2 - r} \]  

(4)

2.1.3 \textbf{Spherical shape}

Assume that \( O \) is the center of a sphere with radius \( r \), thus it must satisfy \( \|x - o\| = r \), i.e.,

\[ (x - x_0)^2 + (y - y_0)^2 + (z - z_0)^2 - r^2 = 0, \]  

(5)

where \( r \) is the radius of the sphere. The signed distance function is given as

\[ d(x) = \|x - o\| - r, \]  

(6)

where \( o = (x_0, y_0, z_0)^T \) is the center of the sphere, and \( r \) is the radius. Generally, the complete sphere could not exist in the scanned scene. Most of the shapes in scene may be semi-sphere or ellipsoidal.

2.1.4 \textbf{Conical shape}

A circular cone can be represented by

\[ \left[\frac{(x - x_0)^2 + (y - y_0)^2 + (z - z_0)^2}{\cos^2(a)} - \left[n_x(x - x_0) + n_y(y - y_0) + n_z(z - z_0)\right]^2\right] = 0, \]  

(7)

where \( [x_0, y_0, z_0]^T \) is the apex of the cone, \( [n_x, n_y, n_z]^T \) is the unit vector defining the orientation of the cone axis, and \( \alpha \) is the semi-vertical angle. The signed distance function is given as

\[ d(x) = \|x - p\| \cos \alpha - n(x - p). \]  

(8)

where \( p = (x_0, y_0, z_0)^T \) is the center of the cone.

2.2 \textbf{Gaussian Image of Different Shapes}

Gaussian map, as defined in Ref. 14, could be illustrated as follows: let \( S = S(u, v) \subset R^3 \) be a regular surface with consistent normal vectors, the map \( G: S \rightarrow S^2 \) associates to each point \( p \in S \) a unit normal vector at \( p \) and takes its values in the unit sphere \( S^2 = \{(x, y, z) \in R^3; x^2 + y^2 + z^2 = 1\} \).

Then the unit sphere \( S^2 \) is the Gaussian sphere, and the map \( G \) is called the Gaussian map of \( S \), shown in Fig. 4. For the discrete point clouds, we assume \( P = \{p_i, i = 1,\ldots,n\} \) is a set of points sampled from a surface, where \( p_i \in R^3 \). The \( r \)-sphere neighborhood of \( p_i \) is defined by \( N_{pi} = \{q||q - p_i||_2 \leq r, q \in P\}, i = 1,2,\ldots,n \). A plane is fitted to the neighborhood of \( p_i \) by least squares fitting method. Then, the unit normal of \( n_{pi} \) to the plane is considered as the normal to the underlying surface at \( p_i \). After that, all directions of \( n_{pi} \) should be consistent by fixing the normal direction at a point, and propagate this information to the rest of points. Then, the Gaussian image of point sets \( P \) is \( G(P) = [n(q), q \in P] \).

For different geometric shapes, the gaussian images are different, which can help us to distinguish the shape of objects.

1. On a plane, the tangent plane never changes, so the Gaussian image of a plane is a point.
2. On a cylinder, the tangent plane is constant along the rulings, so the Gaussian image of a cylindrical surface is a part of a great circle (i.e., an intersection with a plane containing the center of a sphere) on \( S^2 \).
3. On a sphere centered at the origin, the Gaussian image of a sphere is a similarity.
4. On a cone, the Gaussian image is a circle that is perpendicular to the cone axis. The diameter of the circle will vary with the cone angle.

The Gaussian image of each regular surface can be displayed in Fig. 5. Figure 5(a) is the Gaussian image sketch of cylinder, Fig. 5(b) gives the half-cylinder data, and we obtain its Gaussian images (generally a half circle) in Fig. 5(c)–5(e), respectively, which display the regular sphere and its Gaussian image.
3 Primitive Object and Shape Classification

The scenes are generally segmented into clusters using our previous method.\(^{16}\) We adopted the surface-growing algorithm.\(^{16}\) A brief description about this algorithm appears here because of its strong relevance to our object shape classification and scene shape representation method.

The surface-growing algorithm starts by choosing seed points. The seed points are a group of nearby points that fit well to a plane (each point has a residual value which is used to measure its planarity). This method selects a point with smallest residual value, and the seed points tries to grow to their nearby points. Only the point within a certain distance to the seed point and has similar normal vector with seed point could be added to the seed point sets to make it grow.

In this section, we will proceed to classify the objects from the generated segments; the features generated here can distinguish object shape types from one another. For

![Gaussian image instances: (a) Gaussian image sketch of cylinder, half-cylinder points are displayed in (b), and its corresponding Gaussian image in (c), (d), and (e) are the sphere and corresponding Gaussian images.](image)

![Segmentation results on data with 1,224,422 points: (a) refinement results, (b) building part, (c) residual-based segmentation.](image)

**Table 1** Feature description and representation of objects in scene.

<table>
<thead>
<tr>
<th>Detail</th>
<th>Geometric features</th>
<th>Semantic features</th>
</tr>
</thead>
<tbody>
<tr>
<td>Size</td>
<td>Position</td>
<td>Shape</td>
</tr>
<tr>
<td>Ground</td>
<td>$H_g$</td>
<td>$N_g$, $C_g$</td>
</tr>
<tr>
<td>Wall</td>
<td>$H_{wa}$</td>
<td>$N_{wa}$, $C_{wa}$</td>
</tr>
<tr>
<td>Windows</td>
<td>$H_{wi}$</td>
<td>$N_{wi}$, $C_{wi}$</td>
</tr>
<tr>
<td>Door</td>
<td>$H_d$</td>
<td>$N_d$, $C_d$</td>
</tr>
<tr>
<td>Pillars</td>
<td>$H_p$</td>
<td>$N_p$, $C_p$</td>
</tr>
<tr>
<td>Tree trunk</td>
<td>$H_{tt}$</td>
<td>$N_{tt}$, $C_{tt}$</td>
</tr>
<tr>
<td>Lamp-post</td>
<td>$H_{lp}$</td>
<td>$N_{lp}$, $C_{lp}$</td>
</tr>
<tr>
<td>Noise</td>
<td>$H_n$</td>
<td>$N_n$, $C_n$</td>
</tr>
<tr>
<td>Tree leaves</td>
<td>$H_l$</td>
<td>$N_l$, $C_l$</td>
</tr>
</tbody>
</table>

Note: Convex hull of a segment: $H$; normal of a segment: $N$; barycenter of a segment: $C$; projection of a segment: $T$; residual of a segment: $R$; angle between two segments: $A(N_i, N_j)$; distances of two segments: $D(H_i, H_j)$
the extracted objects, they can be categorized into different shapes of objects by the following criterion: the number of points in the cluster, the minimum bounding box of each cluster, and the semantic relations among the clusters which will be considered as features to classify different objects. All these features could be summarized as the domain knowledge of the scene.

Algorithm 1  Scene Object Recognition and Classification.

1: Input: Scene segments \( l_1, l_2, \ldots, l_m \), convex hulls of all the segments \( C_{l_1}, C_{l_2}, \ldots, C_{l_m} \) and the size of the convex hull \( S_{l_1}, S_{l_2}, \ldots, S_{l_m} \).
2: Calculate the centroid of each convex hull;
3: for \( i = 1 : \text{AllHullList.size()} - 1 \) do
4: if it is vertical and its centroid point has minimum \( z \)
5: then it will be considered as ground scene, break.
6: else continue;
7: end if
8: end for
9: Find the largest cluster \( C_{l_i} \) in the remaining \( \text{AllHullList} \)
10: Compare the angle \( AI_i \) between the normal \( NI_i \) and that of ground
11: if \( AI_i \) intersects with ground object then
12: it is the architecture scene object.
13: end if
14: for the remaining clusters in \( \text{AllHullList} \) do
15: if the size of the cluster \( C_{l_k} \) is less than \( S_{l_n} \) then
16: it belongs to noisy points.
17: if \( C_{l_k} \) is perpendicular to the ground and the projection is a circle or half-circle then
18: if it is near to the architecture then
19: it is the pillar scene object
20: else there are much more noisy points in the vicinity of \( C_{l_k} \) then
21: \( C_{l_k} \) belong to tree trunk scene object
22: end if
23: end if
24: end if
25: end for

3.1 Feature Representation
The segmentation results will provide the geometry of each object including the whole architecture, ground, notice board, and pillars. At the same time the residual-based segmentation algorithm is adopted to achieve the detail components in the architecture, see in Fig. 6(b) and 6(c).

Based on the results, the characteristic feature of the objects in scanned scenes could be classified into two categories: one is low-level feature, and the other one belongs to high-level feature.

3.1.1 Low-level features
Low-level features (i.e., geometric features) including the size, shape, and position. Based on each segment (a cluster of points), their size is obtained by counting the number of points, the shape is obtained by estimating the convex hull of each segment, and the position is determined by the normal and the barycenter coordinates of the convex hull.

3.1.2 High-level features
High-level features (i.e., semantic features) are defined from contextual relations. They refer to semantic features which contain the relations between each segment. We create a feature indicating where objects are likely to be with respect to other objects. Here, we focus on the spatial relationship between objects, i.e., “intersection,” “parallel,” and “contain,” which can be represented by their orientation.

3.1.3 Symbolic representation of feature
Some variables are defined to represent the features which could be used to recognize different types of object. Variable symbols appeared in Table 1 are described in the following.

3.2 Object Classification
The object existed in the scene may contain architecture, tree, ground, pillars, lamp-post, etc. Each object has their own features and domain knowledge, e.g., the ground often has the minimum \( z \)-coordinate in the scene. Therefore, in the following, we describe the specific representation which is used to distinguish each object from the scene, and the classification process is displayed in Algorithm 1. The threshold \( S_{th} \) is set as 30 points.

- **Ground** is generally the segment with the lowest position which can be recognized easily.
- **Architecture** is perpendicular to the ground and always the largest object in the scene.
- **Tree trunk** is similar to the cylindrical shape. Sometimes it is difficult to distinguish the tree trunk from the lamp post because they have similar features. To further identify them, the top end close to the canopy and the tree leaves is another critical factor. Generally, there are more noisy points in close vicinity of tree trunk.
- **Pillars** appear more frequently in modern buildings. They cannot be distinguished by simply using the geometric feature and semantic feature. The distance between pillar and architecture is smaller than that of the tree trunk and building.
Lamp post is often a little bit far away from the building, and it is also similar to cylinder.

Noisy points are sparse, irregular and especially distributed scattered.

In addition, there are more noisy points in the vicinity of tree trunk, which are considered as tree leaves.

For the architecture scene object, there are also some other objects such as windows and doors. The difference between windows and doors is that the latter may intersect with the ground. Also, priori knowledge (e.g., width and length) could pay a good way to differentiate them.

Wall is an important part of the building, which is generally vertical and intersects with the ground. Most of the buildings are composed of several walls. Under this condition, if one wall is determined, other walls can be recognized by judging the size and
Fig. 9 Gaussian image of other objects in scene 1: (a) and (b) are the ground and its Gaussian image, respectively; (c) and (d) are the pillar and its Gaussian image, respectively.

Fig. 10 Gaussian image of other objects in scene 1: (a) and (b) are the billboard and its Gaussian image, respectively; (c) and (d) are the lamp and its Gaussian image, respectively.

Fig. 11 Results for Data 2.
the contextual relations with the ground and recognized wall.

- **Windows** are difficult to identify since the points of windows are sparse because of the occlusion by curtains or something else, and reflections by glass when scanning. Consider that windows are parallel with and inside the boundary of the wall.
- **Doors** are specialized from windows from their basis proximity to the ground. It is a low and vertical plane, and it is also inside a wall facade.

Furthermore, others contain the sparse Noise and scattered leaves. Usually, the scanned trees are incomplete due to the occlusion of leaves which will also be considered as noisy points. The criterion of identifying the shape is the residual distribution of the data, which indicated that the larger the residual is, the higher the noise probability is. Generally, the noisy points can be deleted according to residual parameter, i.e., those points with higher residuals are removed.

### 4 Shape Analysis and Fitting of Object

Overall, the primitive shapes (i.e., object type) existed in the scanned scenes can be divided into planar, cylindrical, conical, ellipsoid, spherical, and other unknown shape. To classify the object type, the object characterization should be specific; however, the obtained data is unilateral, which made the shape feature incomplete and difficult to represent. Therefore it is necessary to propose a method to fit the object with a given shape, and further classify and recognize them. In this paper, we proposed to adopt a Gaussian map for shape classification because the Gaussian images of different shapes are distinct. The representation and parameters of geometric shapes are shown in Figs. 2 and 3 in Sec. 2.

#### 4.1 Gaussian Map-Based Shape Classification

As the scene has been decomposed into different segments, and different segments are projected into a Gaussian sphere, then it is a good way to determine the shape of the object by dealing with clusters on the Gaussian sphere. However, the scanned data contains more noisy points, the number of which varies largely in different data. Hence, it often leads to lots of noise when estimating the normal of the data. We can see from Fig. 7 that the exited noisy points are scattered.

Under this condition, we propose to analyze the shape of the clusters on the Gaussian sphere. We know that the scene is segmented into different objects, and the object is segmented into detail shapes. Also, different objects are recognized by the domain knowledge, so the critical task is to extract clusters on the Gaussian sphere. On the Gaussian sphere, the points within given distance threshold $D_{th}$ are clustered.

Assume the clusters $C = \{c_i, i = 1, 2, \ldots, k\}$ are obtained, and if the number of points in one cluster $c_i$ are less than a threshold num then they would be discarded as noisy points. At the same time, we judge whether the number $k$ of a remaining cluster is one, and if so, the shape is planar. If the number $k$ of cluster $C$ is more than one, then further work to classify the shape is required.

The points on the Gaussian sphere are projected onto a plane perpendicular to $n_{p_i}$, and we detect circles on the plane using hough transform. The shape is determined by the radius size of projected cluster. If it is a great arc-form cluster (i.e., a great circle), then the shape is cylindrical; otherwise, if it is a small arc-form cluster (i.e., a small circle), then the shape is conical.

#### 4.2 Shape Fitting

The shape is recognized through the analysis on Gaussian images; we continue to approximate respective shapes by fitting those points. The critical work is to calculate the axis vector of each shape and determine the position of the axis.

##### 4.2.1 Axis vector of each shape

Based on the Gaussian image of each object, we analyze the data on the Gaussian sphere by clustering them; if most of them could be clustered, and the density is high, then it could be a planar surface, and the normal of the plane can be easily determined by the Gaussian sphere center and the clustered points on the Gaussian sphere. In addition, if a circle or half-circle is detected in the Gaussian sphere, the axis is the vector perpendicular to tangent plane of the circle (or half-circle).

##### 4.2.2 Position of axis

Generate a plane in the direction of axis vector, and then project those clustered points in the Gaussian sphere into the plane. Choose the centroid point as one point that the axis may go through. Also, the least square fitting method can be adopted to estimate the radius and axis vector for each geometric shape. The fitting error can be used to determine which shape the object may belong to. The smaller the fitting error is, the more credible that the shape is. The results will be displayed and analyzed in Sec. 5.

### 5 Experimental Results

In this paper, the experiments are measured on an Intel Core 2.5-GHz computer with 2-GB RAM. The data used in this paper are scanned from different scanners: Data 1 is obtained by Topcon, Data 2 and Data 3 are from Faro. These data are all processed (delete extra noisy data) when exporting from the scanner software.

Our proposed, semi-automatic method involves a two parameters: the size of noisy points and the distance threshold sometimes require user intervention for different data. The default values of these parameters typically give good results. However, for very noisy input, the threshold for that size of noisy points would be manually set. And, for data scanned from different scanners, the distance threshold should be reset.

#### 5.1 Object and Shape Recognition Results

Based on the results in Fig. 6(b) and 6(c), we implement the process of object recognition and classification by using the aforementioned domain knowledge including geometric and semantic features; see Fig. 8. According to the algorithm described in Sec. 3.2, based on the domain knowledge, we can distinguish the small objects in scanned scenes. The scene including the ground, windows, and wall can be determined easily. For the pillar and lamp post, we can project the points in each segment and estimate the shape in 2-D projection.

Furthermore, a Gaussian image-based shape understanding method is implemented for different objects, which can classify an object into different shapes including planar, cylindrical, conical, ellipsoid, spherical, and other unknown shape.
cylindrical, spherical, and conical. Making use of the Gaussian image for each different region, it is easier to understand the component shape of objects. In Fig. 7, the Gaussian images of different architecture parts are acquired, from which we noticed that most of the points are clustered together. Therefore we can conclude that these parts belong to planar surface. Figures 9 and 10 represent the Gaussian images of other objects in the scene; they indicate the point distribution for different shapes, and for cylinder in Fig. 10(c) and 10(d). The building in Data 2 and its Gaussian image are obtained in Figs. 11 and 12.

Table 2 Fitting parameters for Data 1.

<table>
<thead>
<tr>
<th>Objects</th>
<th>Axis vector</th>
<th>Point on shape</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ground(Planar)</td>
<td>$n = (0.0231, -0.0284, 0.9993)$</td>
<td>$p = (-0.0135, -0.0140, 0.9996)$</td>
</tr>
<tr>
<td>Wall1(Planar)</td>
<td>$n = (0.7302, 0.6551, -0.1939)$</td>
<td>$p = (-0.5859, 0.8094, 5.5172e-04)$</td>
</tr>
<tr>
<td>Wall2(Planar)</td>
<td>$n = (-0.8000, -0.5996, -0.0239)$</td>
<td>$p = (0.8052, 0.5909, -8.8702e-04)$</td>
</tr>
<tr>
<td>Wall3(Planar)</td>
<td>$n = (0.5577, 0.8272, -0.0687)$</td>
<td>$p = (0.8098, 0.5848, -6.9845e-04)$</td>
</tr>
<tr>
<td>Wall4(Planar)</td>
<td>$n = (-0.5676, 0.8208, 0.0639)$</td>
<td>$p = (-0.5821, 0.8113, 0.0011)$</td>
</tr>
<tr>
<td>Wall5(Planar)</td>
<td>$n = (0.5825, -0.8108, 0.0132)$</td>
<td>$p = (-0.5858, 0.8101, -9.7800e-04)$</td>
</tr>
<tr>
<td>Pillar(Cylindrical)</td>
<td>$n = (-0.1538, 0.6524, 0.7421)$, $r = 1.1225$</td>
<td>$p = (-0.0177, 0.0696, 0.6276)$, $h = 7.8086$</td>
</tr>
<tr>
<td>Lamp(Spherical)</td>
<td>$n1 = (0.9972, -0.0039, -0.0749)$</td>
<td>$p1 = (0.985, 0.0024, -0.0691)$, $r = 0.1483$</td>
</tr>
<tr>
<td>Lamp(Cylindrical)</td>
<td>$n2 = (-0.0210, -0.0267, 0.9994)$, $r = 1.0080$</td>
<td>$p2 = (-0.0081, 0.0047, 0.0034)$, $h = 1.4903$</td>
</tr>
</tbody>
</table>

5.2 Shape Fitting Results Analysis

In this section, we simply analyze the parameters for fitting each shape, which are respectively shown in Tables 2 and 3. Table 2 displays the results of parameters (i.e., the axis vector and a point on the shape) for each scene object in Data 1 when fitting respective shape. For the objects in Data 1, walls and ground are recognized as a planar shape, and we approximate a plane by deriving its normal and a point on the plane. Pillars are recognized as a cylindrical shape, and the normal, radius, height, and a point on the
cylinder are calculated to fit its shape. Table 3 shows the parameters when fitting different shapes of each object in Data 2. Those parameters are then used to organize the object and the scene.

In order to determine the accuracy, we estimate the fitting error by calculating the distance to the fitting shape model, which is not only useful to estimate the fitting results, but also helpful to determine the recognition rate. In Figs. 13 and 14, we list the error deviates of each shape in the two data. For the scene in Data 1, we classify it into wall, ground, pillar, and lamp, and fit them with suitable shape. For the scene in Data 2 (see Fig. 14), it is classified into wall and four trees. And, based on the Gaussian image, we could acquire the corresponding shape and approximate each object with its shape. The fitting error for each object in Data 2 is shown in Fig. 14. Wall is fitted into a cuboid, and the error between scanned data and fitting model is calculated as shown in Fig. 14(a). Trees are fitted into a cone,

<table>
<thead>
<tr>
<th>Objects</th>
<th>Axis vector</th>
<th>Point on shape</th>
</tr>
</thead>
<tbody>
<tr>
<td>Building1(Planar)</td>
<td>$n = (0.4057, -0.7511, -0.5208)$</td>
<td>$p = (0.7006, -0.1784, -0.6107)$</td>
</tr>
<tr>
<td>Tree1(Cylindrical)</td>
<td>$n_1 = (0.0116, 0.9999, 0.0068)$, $r = 1.0023$</td>
<td>$p_1 = (0.0122, -0.1377, 0.02273)$, $h = 0.4502$</td>
</tr>
<tr>
<td>Tree1(Conical)</td>
<td>$n_2 = (0.4777, -0.8785, -0.0037)$, $r = 0.8072$</td>
<td>$p_2 = (0.3129, 0.2280, 0.3148)$, $\alpha = 9.3867$</td>
</tr>
<tr>
<td>Tree2(Conical)</td>
<td>$n = (-0.0047, 0.7198, 0.6942)$, $r = 0.7752$</td>
<td>$p = (-0.0200, -0.0088, -0.0257)$, $\alpha = 9.3914$</td>
</tr>
<tr>
<td>Tree3(Conical)</td>
<td>$n = (0.1978, -0.2151, 0.9564)$, $r = 0.7475$</td>
<td>$p = (-0.0135, -0.0140, 0.9996)$, $\alpha = 9.5751$</td>
</tr>
<tr>
<td>Tree4(Conical)</td>
<td>$n = (0.8866, -0.1022, 0.4510)$, $r = 0.8780$</td>
<td>$p = (-0.0365, 0.0069, -0.0358)$, $\alpha = 9.3811$</td>
</tr>
</tbody>
</table>

![Fig. 13 Error analysis for different objects in Data 1.](image-url)
and the errors are shown in Fig. 14(b)–14(f). The scale unit of horizontal axis is point number of data, and the scale unit of vertical axis is cm (which depends on the scanner accuracy). The error deviation is larger when fitting the tree leaves as cone shape [see Fig. 14(c)–14(f)], which is affected by the noisy data in tree leaves. The scene shape fitting results on the two data are shown in Fig. 15.

To further evaluate the effectiveness of our approach, we implement our method on two other datasets (Data 3 contains 401,462 points and Data 4 has 412,730 points), and results are displayed in Figs. 16 and 17, respectively. Our approach successfully represents the shape of the scene. Note that our approach is also applicable to scenes with spherical trees. The original scanned Data 3 in Fig. 16(a) and Data 4 in Fig. 17(a) are decomposed into different objects, respectively, and the segmentation results [Figs. 16(b) and 17(b)] and corresponding fitting results [Figs. 16(c) and 17(c)] are acquired by our Gaussian map-based shape classification and fitting method.

Our approach cannot only be used for object shape recognition and understanding, but also could be further applied to scene representation and reconstruction. According to the fitting results, we can describe the scene using the geometric shape, which makes it easier to understand.

6 Conclusions
We presented a primitive shape classification and fitting method for a 3-D scene using geometric and semantic features (domain knowledge) and the Gaussian image. Our method is based on the previous hierarchical segmentation method. Geometric features and semantic features of each
Acknowledgments
The work is supported in part by National Natural Science Foundation of China under Grant Nos. 61072151, 61272284, 60970093, and 61271431; Shaanxi Science and Technology research under Grant No. 2011K06-35; Shaanxi Education Technology Plan under Grant No. 2010JK734; Xian Scientific Research Plan under Grant No. CX1252(3); Xian BeiLin Scientific Research Plan under Grant No. GX1212; in part by Doctoral Fund of Ministry of Education of China under Grant Nos. 61072151, 61072018; and by Doctor Start-Up Foundation of Xi’an University of Technology.

References

Fig. 17 Fitting results for Data 4: (a) the data, (b) scene segmentation, (c) shape fitting.