3D Road Boundary Detection Using Conformal Geometric Algebra

HIROYUKI ISHIDA$^{1,a)}$ JUN-ICHI MEGURO$^{1}$ YOSHIKO KOJIMA$^{1}$ TAKASHI NAITO$^{1}$

Received: March 29, 2013, Accepted: July 22, 2013, Released: October 18, 2013

Abstract: This paper presents a novel method for detecting 3D road boundaries, such as walls, guardrails, and curbs, using on-board stereo cameras. The proposed method uses conformal geometric algebra, which can describe different shapes in a common representation. 3D road boundaries on straight and curved roads are seamlessly detected by use of this representation, and this framework is also applied to curb detection by a subtle modification. Experimental results show that despite its algorithmic simplicity, the proposed method exhibited competitive detection performance compared with conventional model fitting and curb detection methods.

Keywords: road environment recognition, map generation, point cloud analysis, geometric algebra

1. Introduction

Advanced driver assistance systems for increased automobile safety have gained attention in recent years. One potential application is intelligent speed adaptation for the prevention of over-shooting on curves. For this application, it is necessary to construct a database in which the accurate position and shape of road boundaries are embedded in a map. The combination of this database with the position of the vehicle can then provide useful information to drivers. The construction and maintenance of this database, however, requires not only a large-scale mobile mapping system but also extensive processing. A framework for constructing this database automatically and by using low-cost sensors is thus necessary. To help meet this challenge, we present a technique for detecting 3D road boundaries such as walls, guardrails, and curbs.

There have been various attempts at 3D object recognition. The iterative closest-point algorithm [1] and its extensions are widely used for fitting models to point clouds. Surface simplification [2] and geometric feature extraction [3] are often applied to noisy and large-scale point clouds. However, some complications arise in the detection of 3D road boundaries. First, the point clouds may contain multiple outliers such as vehicles, pedestrians, and vegetation. Second, the point clouds should not be of a high density, since a wide range of points is processed at once. Third, the position of the points is only moderately accurate. Even with a high-accuracy GPS [4], estimates of the position suffer from variations in the pitch angle and from errors in stereo reconstruction. In addition to the problems mentioned above, a scheme for detecting boundaries needs to adapt easily to the geometry of the road. Among the various road boundaries, curved curbs are the most difficult to detect. Our proposed method copes with these problems by combining the use of conformal geometric algebra (CGA) [5] with an evolutionary approach. The high-dimensional geometric representation of CGA provides a common representation for various shapes of the road boundary. The proposed method also encodes particular road geometries and constraints into a compact representation that provides for effective detection of road boundaries even from sparse and noisy point clouds. By a small modification of the optimization algorithm, this framework is extended to the challenge of detecting curved curbs.

Our detection process consists of four steps: acquiring the image, calculating the point cloud, detecting the 3D road boundary, and updating the database. This paper is organized as follows: Section 2 discusses the calculation of point clouds, Section 3 describes the detection of road boundaries, Section 4 presents experimental results, and Section 5 presents our conclusions.

2. Point Cloud Calculation

The points that represent 3D road boundaries are obtained by stereo cameras. Disparities between the two stereo images are calculated by a stereo-matching algorithm [6] so that the density of points can be varied from sparse to dense. As described in Ref. [7] and elsewhere in the literature, the distance from the camera to a point is calculated by $z = \Delta x f / d$, given a disparity $d$, a focal length $f$, and a baseline distance $\Delta x$. Multiple frames are integrated using vehicle positional information such as latitude, longitude, and heading angle. The positional information is obtained with a high-accuracy GPS [4] or with the combination of a low-cost GPS and the vehicle trajectory [8]. A set of points is captured for every 100 meters, and these are integrated together to generate a local map. The full database is obtained by merging the local maps.

Examples of point clouds are shown in Fig. 1. Although dense point clouds can be provided, the use of too many points leads to an increase in the algorithmic complexity and may lead to false detections. Our aim is to detect 3D road boundaries using a moderate number of points.
3. Detecting Road Boundaries

Entities representing road surfaces, sidewalks, walls, and guardrails are extracted from point clouds. The main benefit of this method is that the entity representation and extraction process are universally applicable to different shapes and types of road boundaries.

It is difficult to detect curbs because they do not always have a distinctive appearance. Several specialized methods for curb detection have been proposed. Oniga et al. [9], [10] proposed calculating an elevation map in order to extract curb segments, while Siegemund et al. [11] proposed a polynomial curb model with surface parameter estimation. These need to be used in addition to other 3D detectors because an elevation map cannot be used to detect guardrails. Another challenging task is the treatment of a curved road. The use of different representations for different road geometries not only complicates the algorithm but also causes false detections. A solution for these problems is provided by CGA [5], which can use a common representation for different shapes and types of road. This section focuses on the entity extraction from point clouds in a local map. Throughout this section we adopt a right-hand coordinate system with axes pointing horizontally rightward (e1), vertically downward (e2), and forward (e3). We assume that the ground surface is nearly perpendicular to the axis e2. This axis e2 is left unchanged by the vehicle movement.

3.1 Brief Overview of CGA

The CGA for a 3D Euclidean base space employs a 5D space consisting of basis vectors \( \{e_1, e_2, e_3, e_\omega, e_o\} \). The first three correspond to those of a 3D Euclidean space, and the last two represent points at infinity and at the origin. In this CGA space, points, spheres, and planes in Fig. 2 share a common representation as

\[
P = s_1e_1 + s_2e_2 + s_3e_3 + s_\omega e_\omega + s_o e_o.
\]

For example, a point \([x, y, z]^T\) in a 3D space is represented as

\[
P = x + \frac{1}{2}x^2e_\omega + e_o
\]

\[
x = xe_1 + ye_2 + ze_3
\]

using the following coefficients.

\[
s_1 = x, \quad s_2 = y, \quad s_3 = z, \quad s_\omega = \frac{1}{2}(x^2 + y^2 + z^2), \quad s_o = 1 (4)
\]

The coefficients of a sphere with its center at point \([x, y, z]^T\) and a radius of \(r\) are

\[
s_1 = x, \quad s_2 = y, \quad s_3 = z, \quad s_\omega = \frac{1}{2}(x^2 + y^2 + z^2 - r^2), \quad s_o = 1.
\]

Also, generic geometric transformations are encoded by algebraic operations such as the inner product (\(\cdot\)), outer product (\(\wedge\)), and dualization (\(\dagger\)). By definition of the geometric algebra, the inner and outer products are expressed as

\[
A \cdot B = \sum_{p,q} \langle(A)_p(B)_q \rangle_{p+q}
\]

\[
A \wedge B = \sum_{p,q} \langle(A)_p(B)_q \rangle_{p+q}
\]

using grade operator \(\langle X \rangle_p\) which takes grade \(p\) part of a general multivector \(X\). In a specific case where \(\text{grade}(A) = \text{grade}(B) = 1\), the inner product between basis vectors of the CGA is given by

\[
e_i \cdot e_j = \begin{cases} 
1 & (i, j) = (1, 1), (2, 2), (3, 3) \\
-1 & (i, j) = (\omega, \omega), (o, o) \\
0 & \text{otherwise}
\end{cases}
\]

The outer product satisfies the following relationships for arbitrary CGA vectors \(A\) and \(B\):

\[
A \wedge B = -B \wedge A
\]

\[
A \wedge 1 = A
\]

The dual of \(A\) is calculated by

\[
A^* = AI^{-1} = A(e_\omega \wedge e_1 \wedge e_2 \wedge e_3 \wedge e_o).
\]

For more details, refer to Ref. [5]. C++ libraries [12] and [13] are available to facilitate the above-mentioned algebraic operations.

The representation of entities and their geometric transformations are listed in Table 1. For example, a plane \(Pl\) passing through three points \(P_1, P_2,\) and \(P_3\) is denoted by \(Pl =\)
Table 1 3D representation of entities using CGA.

<table>
<thead>
<tr>
<th>Entity</th>
<th>by bases</th>
<th>by points</th>
</tr>
</thead>
<tbody>
<tr>
<td>Point</td>
<td>( x + \frac{1}{2}(</td>
<td>e_n</td>
</tr>
<tr>
<td>Sphere</td>
<td>( e + \frac{1}{2}(e^2 - r^2)</td>
<td>e_n</td>
</tr>
<tr>
<td>Plane</td>
<td>( n + de_m )</td>
<td>normal: ( n ) distance: ( d )</td>
</tr>
<tr>
<td>Circle</td>
<td>( (P_1 \land P_2 \land P_3))</td>
<td>( P_1 \land P_2 \land P_3 \land e_o )</td>
</tr>
<tr>
<td>Line</td>
<td>( (P_1 \land P_2 \land e_o))</td>
<td>( P_1 \land P_2 \land e_o )</td>
</tr>
</tbody>
</table>

\((P_1 \land P_2 \land P_3 \land e_o)\) \(P\). The least distance between this plane \(P\) and a point \(P\) is calculated by \(P \cdot P\), provided that \(P\) is normalized such that \(|P| = 1\).

Conventional object-recognition methods [14], [15] based on CGA employ a least-squares approach for estimating the parameters in Eq. (1). In application to the road environment, however, the least-squares approach can be affected by outliers due to stereo matching errors and other on-road objects. The removal of the outliers is necessary prior to a parameter estimation. An alternative method [16] combines CGA and Hough transforms [17] to detect lines and planes. Although CGA has the potential to represent many different shapes, it is not directly applicable to 3D road boundaries because they are not simply composed of planes and spheres. We present a means for applying the CGA object representations to the road environment.

3.2 Detecting 3D Boundaries

This method uses the representation of entities that is shown in Table 1. The components of the entities are determined as a result of optimization. The objects detected at this stage are road surfaces, sidewalks, walls, and guardrails. The guardrails are detected by the same way as other planar objects; for example, a straight guardrail can be detected as a plane because its feature points are located approximately on the same plane.

The algorithm is described in Table 2 and proceeds as follows.

1. Transform
First of all, 3D points are transformed to CGA vectors \(P\).

2. Initialization
The initial entities are calculated from the components using

\[
S_j = (\overline{P}_{j1} \land \overline{P}_{j2} \land \overline{P}_{j3} \land \overline{P}_{j4})^\ast \tag{13}
\]

where we choose these four components \(\overline{P}_{j1}, \overline{P}_{j2}, \overline{P}_{j3}, \overline{P}_{j4}\) randomly and without repetition from a component set \(\overline{P} = \{P_{1\leq j \leq N}, e_m, e_2, 1\} \). For the purpose of distance calculation, \(S_j\) needs to be normalized by the following operation.

\[
S_j = S_j / |S_j| \quad \text{if } s_0 = 0 \text{ (plane)}
\]

\[
-\frac{S_j}{(e_m \cdot S_j)} \quad \text{if } s_0 > 0 \text{ (sphere)} \tag{14}
\]

Table 3 shows the relationship between the components and the resulting entities. Valid entities include the following: sphere, plane, plane*, circle, line, and line**. If other, invalid combinations are selected, the entity is reset. Note that the component \(e_2\) imposes a constraint that the entity is perpendicular to the ground, and the component '1' is treated

\[\ast\ast\] Although line\(^2\) can be used for the detection of poles, it is not focused on nor treated as a road boundary in this paper.

Table 2 Pseudo-code of our method using CGA and GA. \(N\) and \(M\) are the number of points and chromosomes, respectively. \(N\) decreases as the steps 2–4 are iterated, and finally converge to zero. In practice, we break the iteration when the number of remaining points becomes small enough.

\[
\text{(In) } x_i \leq i \leq N) \text{ points} \\
\text{end while} \quad \text{for } j = 1 \ldots N \\
P_i = x_j + \frac{1}{2}S_j^* e_m + e_o
\]

\[
j = 1 \ldots N \\
P_j = x_j + \frac{1}{2}S_j' e_m + e_o \\
P_j \leftarrow \text{select one from } \{P_1, \ldots, P_N, e_m, e_2, 1\}
\]

\[
\text{max } w_i \text{ becomes large enough}
\]

3. Optimization

A genetic algorithm (GA) is used to produce chromosomes \(\{\overline{P}_{j1}, \overline{P}_{j2}, \overline{P}_{j3}, \overline{P}_{j4}\}\) consisting of the components in Eq. (13). The number of points within a distance \(r_1\) from the entity \(S_j\) is used as the fitness (Fig. 3(a)), and denoted by \(w_j\). We set this threshold \(r_1\) to 0.1 [m], since most points on walls or guardrails are considered within ±0.1 [m] from their original entity. The distance between a CGA vector \(P_i\) and the entity \(S_j\) is given by

\[
d(P_i, S_j) = \begin{cases} 
 \frac{P_i \cdot S_j}{|P_i|^2} & \text{if } s_0 = 0 \text{ (plane)} \\
 \frac{1}{r - \sqrt{|r^2 - 2P_i \cdot S_j|^2}} & \text{if } s_0 > 0 \text{ (sphere)}
\end{cases}
\]

where \(r\) is the radius of the sphere obtained from Table 1, and \(s_0\) is the coefficient of the term \(e_o\). The sign of the distance represents which side \(P_i\) is located on, though only the unsigned distance \(d(P_i, S_j)\) is used in this section. Later in Section 3.3, the difference of the sign is used to distinguish

\[\ast\ast\] We substitute the following sphere \(S_j^*\) for a circle \(S_j\) in order to simplify the distance calculation.

\[
S_j^* = (e_m \cdot S_j) \cdot S_j.
\]

Since \(S_j^*\) is a sphere with a great circle \(S_j^*\), \(d(P_i, S_j^*)\) approximates the distance to a curved guardrail if the radius \(r\) is large enough. Note that this \(S_j^*\) also requires the normalization by Eq. (14) for the distance calculation. For the same reason, a line \(S_j\) can be substituted by the following plane \(S_j^*\) perpendicular to the ground.

\[
S_j^* = S_j \cdot e_2.
\]
third step in Table 2 involves ML calculations of $S_j$ and NML calculations of $d(P_i,S_j)$. Although this is much smaller than an exhaustive search for the optimal $S_j$ (more than $N^3$ in total), keeping N small is still necessary. We restrict the density of points such that $N < 1,000$.

3.3 Curb Detection

Curb detection is performed if the extracted entities contain multiple planes parallel to the ground surface. Such planes satisfy

$$s_1 = 0, s_2 \approx 1, s_3 = 0, s_0 = 0.$$  \hspace{1cm} (18)

We regard the curb as the boundary that best separates a sidewalk surface $Pl_s$ and a road surface $Pl_r$, as illustrated in Fig. 3(b). To find such a boundary $S_j$ from the 3D points $Pl_i$, we maximize

$$\sum_i l(P_i, Pl_s, Pl_r, S_j),$$  \hspace{1cm} (19)

where

$$l(P_i, Pl_s, Pl_r, S_j) = \begin{cases} 1 & \text{if } i \in \mathcal{P}_s \text{ and } d(P_i, Pl_s) > 0 \\ 1 & \text{if } i \in \mathcal{P}_r \text{ and } d(P_i, Pl_r) < 0 \\ -1 & \text{if } i \in \mathcal{P}_s \text{ and } d(P_i, Pl_r) < 0 \\ -1 & \text{if } i \in \mathcal{P}_r \text{ and } d(P_i, Pl_s) > 0 \end{cases}$$  \hspace{1cm} (20)

Equation (19) represents how well the boundary $S_j$ classifies points which are labeled as $\mathcal{P}_s$ (sidewalk points) or $\mathcal{P}_r$ (road surface points). It attains its maximum value when all signs of $l(P_i, Pl_s, Pl_r, S_j)$ are the same, namely the labeled points are separated by the boundary $S_j$. In the actual process, the following exponential function is used instead to accelerate the optimization.

$$w_j = \exp \left( \frac{\omega}{|\mathcal{P}_{slr}|} \sum_{P_i \in \mathcal{P}_{slr}} l(P_i, Pl_s, Pl_r, S_j) \right),$$  \hspace{1cm} (21)

where $\mathcal{P}_{slr}$ is a set of points on the plane $Pl_s$ or $Pl_r$, and $\omega(\approx 7)$ is a parameter which controls the range of the fitness such that $0 \leq w_j \leq 1,000 \approx \exp(\omega)$. This algorithm to detect a curb is the same as in Section 3.2 and Table 2 except that the fitness $w_j$ is replaced by Eq. (21).

An example of curb detection is illustrated in Fig. 4. The boundary that best separates the points is found as a result of the evolutionary operation.

3.4 Segmentation

The extracted entities are projected on the road map as a set of boundary segments. A segment is registered on the database if the number of points lying on it is larger than a threshold $\tau_z$. The coordinates are transformed from the camera-centered coordinate system to the north-east coordinate system.

4. Experiments

We conducted experiments to test the performance of this

Table 3 Relations between components and the resulting entities. The components • represent arbitrary points $P_i$ in the form of a CGA vector. The symbol $\perp$ indicates that the entity is perpendicular to the ground surface.

<table>
<thead>
<tr>
<th>Components</th>
<th>Resulting entity</th>
</tr>
</thead>
<tbody>
<tr>
<td>• • • • •</td>
<td>Sphere</td>
</tr>
<tr>
<td>• • • • e</td>
<td>Plane</td>
</tr>
<tr>
<td>• • • • e</td>
<td>Plane</td>
</tr>
<tr>
<td>• • • 1</td>
<td>Circle</td>
</tr>
<tr>
<td>• • • 1</td>
<td>Line</td>
</tr>
<tr>
<td>• • • 1</td>
<td>Line</td>
</tr>
<tr>
<td>• • • 1</td>
<td>Line</td>
</tr>
</tbody>
</table>

The number of points satisfying $|d(P_i, S_j)| < \tau_1$ and classification rates of points on two surfaces.
method. Test data were captured during a run of 7.9 km in a residential district using a pair of stereo cameras (SONY XCD-SX90, 1,280 × 960, 15 fps for each). The test data contained both straight (86%) and curved (14%) sections of road. The road boundary consisted of guardrails (1.0 km), curbs (1.9 km), wall-like objects (3.8 km), or a mixture of these. Results were obtained by merging local maps constructed from every 100-meter run. We evaluated the overall true positive rates of 3D road boundaries, including guardrails and curbs. The ground truth of the 3D road boundaries was constructed using a mobile mapping system (MMS) [19]. The MMS used a laser scanner and real-time kinematic-(RTK-)GPS [4]. Based on the observations, boundary lines of the ground truth were drawn manually by CAD designers. The overall positioning error of the ground truth was assumed to be ±0.10 [m]. This experiment also used RTK-GPS for the test sequences of the vehicle position in order to exclude GPS errors from the performance evaluation.

4.1 Tested Methods
We compared our method to the following:
(I) 2D Hough transform for lines and circles
(II) Method I in conjunction with curb detection [9]
(III) 3D point cloud analysis using planar and cylindrical models [20] and RANSAC [21]
(IV) Method III using dense stereo matching [6]
In Method I, 2D Hough transforms are applied to feature points projected on the horizontal plane, and then lines or circles are extracted in the order of voting. Lines are defined by two parameters (angle and distance from the center), whereas circles are defined by three parameters (center and radius). The resolution of the Hough space was set to 0.1 [m]. The maximum number of points per cubic meter was 12,500 for method IV, and 100 for the other methods. Basically, all the methods used the iterative extraction scheme presented in the fourth step of Table 2.

4.2 Detection Results
Rates of true positive and false positive detections are shown in Fig. 5. The true positive rate is the percentage of the actual boundary length that was detected, and the false positive rate is the average length of the boundaries that were detected but that did not actually exist. The receiver operating characteristic (ROC) curves were calculated by varying the threshold value τ2. The use of a small τ2 raises a true positive rate, but simultaneously yields many false detections. The individual true positive rates in straight and curved sections of a road are shown in Table 4.

Figure 6 shows examples of road boundaries that were detected by the proposed method. They contain a challenging situation, where the road and the sidewalk were divided only by a curved curb. In this situation, the road surface was initially extracted using the procedure in Section 3.2, followed by the extraction of the sidewalk surface and the electric pole. Since more than one plane satisfied the condition of Eq. (18), the procedure in Section 3.3 was performed, and the curb was thereby detected.

4.3 Positioning Accuracy and Computational Cost
The averaged error distance between the detected 3D road boundaries and the ground truth by MMS [19] was calculated. Since the positioning error estimates of the laser scanner and the RTK-GPS in the MMS were both within ±0.03 [m], the resulting positioning error was considered to be mostly due to the detection schemes. Table 5 shows the average error distance together with the total processing time per 100 meters of run. Methods I and II, which used the Hough transform, exhibited a large positioning error. Although method IV exhibited the smallest positioning error, it required the highest computational cost due to the dense.
point clouds.

4.4 Discussion

The proposed method exhibited a high detection performance despite its simple algorithmic implementation (Fig. 5). This is due to the method of representing entities, which allowed for the compact encoding of multiple road geometries and constraints (such as perpendicularity to the ground). The scheme presented in Section 3.3 raised the curb detection performance even though the point density was low. When the targets were 3D road boundaries, the detection performance was nearly equivalent to that of a 3D model fitting. Although 3D models have advantages in object recognition, they are sensitive to outlier objects in the road environment. The true positive rates in Table 4 indicate the usefulness of a CGA model in curved sections. Meanwhile, the detection of a curved boundary by the Hough transform (methods I and II) required a large-scale voting space spanned by three variables, which complicated the boundary identification. The effectiveness of the elevation map for detecting curbs was limited due to noise-inducing false-positive detections.

However, the proposed approach also has problems, as follows. If a small curb is located between two surfaces of the same height, it is difficult to detect. Also, the computational cost was higher than method III due to the inner-product operations. A solution for accelerating the computation is to use an approximation technique introduced in Section 14.5.2 (pp.419–420) of Ref. [5]. Improving the optimization framework to solve these problems is an area for future work.

5. Conclusion

We have presented a novel method for detecting 3D road boundaries. Using the framework of CGA, we can detect guardrails and curbs in challenging situations by a simple and routine procedure. Experimental results show that the performance of the proposed method is highly competitive compared to other methods that use 3D model fitting or curb detection. Future work will involve the integration of multiple local maps to improve the localization performance without the need for a high-accuracy GPS.
Hiroyuki Ishida was born in 1982. He received his B.E. degree in information engineering, and M.E. and D.E. degrees in information science from Nagoya University. His research interests include pattern recognition and its application to engineering. He is an Associate Researcher in Toyota Central R&D Labs.

Jun-ichi Meguro received his B.E. and M.E. degrees in mechanical engineering from Waseda University in 2003 and 2005, respectively, and D.E. degree from Waseda University in 2008. His research interests include navigation and positioning systems for automobile. He is an Associate Researcher in Toyota Central R&D Labs.

Yoshiko Kojima received her B.E. and M.E. degrees in electronics engineering from Nagoya University in 1993 and 1995, respectively. Her research interests include information fusion for road environment reconstruction and positioning system for automobile. She is a Principal Researcher in Toyota Central R&D Labs.

Takashi Naito received his B.E. and M.E. degrees in electronic-mechanical engineering from Nagoya University in 1987 and 1989, respectively. His research interests include image processing technology for automobile. He is a Principal Researcher in Toyota Central R&D Labs.

(Communicated by Brendan McCane)