Road marking recognition for map generation using sparse tensor voting

Hiroyuki Ishida, Kiyosumi Kidono, Yoshiko Kojima, Takashi Naito
Toyota Central R&D Labs., Inc., Japan
h-ishida@mosk.tytlabs.co.jp

Abstract

A recognition method of road markings for map generation is presented. For accurate position estimation and classification, two voting schemes are proposed and combined. The first is multi-frame sparse tensor voting for geometric feature extraction, and the second is contour localization using the resulting tensor field. Classification is based on the similarity between the aligned contour and the tensor field. The experimental results show that the proposed method outperforms conventional matching-based approaches.

1. Introduction

Driver assistance systems using road marking recognition have gained attention in recent years. Road markings can be used not only as indicators of traffic regulations but also as matching cues for the correction of GPS errors [1]. For this purpose, we need a road marking map [2] that contains accurate positions of symbolized road markings. However, construction and revision of the map requires a large-scale mobile mapping system and manual procedures. To solve this problem, this paper presents a road marking recognition method using an on-vehicle camera and GPS measurements [3].

Conventional recognition methods use the Radon transform [4], moment features [5], contour orientation [6], or a histogram of oriented gradient (HOG) features [7], after constructing an orthoimage by inverse perspective mapping. However, the recognition from an orthoimage tends to suffer from position estimation errors due to trade-offs between resolution and searching cost. Another problem is how to store the map. Neither storing the complete orthoimage nor storing only the center positions of road markings are suitable for map matching applications. An alternative is to store geometric features compatible to the contour-based map representation. This paper proposes a method to extract geometric features by tensor voting [8] and to store them as a form of a sparse tensor field for recognition. The orthoimage is substituted by the resulting tensor field, which can avoid the position estimation errors caused by image matching.

The flow of the proposed method is presented in Fig. 1, and the three main processes are described in Sections 2, 3, and 4. The results are presented in Section 5 and the conclusions in Section 6.

2. Inverse perspective mapping

Edge points are extracted from the captured image by the Canny edge detector and projected to real world coordinates. Let us denote intrinsic and extrinsic camera parameter matrices by $A$ and $[R|t]$, respectively, a vehicle’s yaw angle by $\phi$ [rad], and the camera height from the ground by $h$. The relation between the real world coordinates $(e, n)$ [m] and the image coordinates $(x, y)$ is given in Fig. 2 and the following equation:

$$
\begin{pmatrix}
  x \\
  y \\
  1
\end{pmatrix} \propto A[R|t]
\begin{pmatrix}
  \cos \phi & 0 & \sin \phi & 0 \\
  0 & 1 & 0 & 0 \\
  -\sin \phi & 0 & \cos \phi & 0 \\
  0 & 0 & 0 & 1
\end{pmatrix}
\begin{pmatrix}
e \\
h \\
n \\
1
\end{pmatrix}.
$$

(1)

3. Geometric feature extraction

The geometric features of road marking contour lines are extracted. The tensor voting framework is em-
Figure 2. Relation between the real world and image coordinates

Ball voting

Stick voting (voter’s orientation $\beta = 0$ [rad])

Figure 3. Voting fields. Tensors of casted votes are illustrated by line segments.

Figure 4. Enhancement of the tensor field using multiple frames

employed, since it is adaptable to local features and simultaneously robust against noise and degradation.

In tensor voting [8], edge points and voted values are represented as tensors, which are decomposed as

$$S = \begin{bmatrix} e_1 & e_2 \end{bmatrix} \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix} \begin{bmatrix} e_1^\top \\ e_2^\top \end{bmatrix},$$

(2)

where the first eigenvector $e_1$ represents the edge orientation, and the stickness value $\lambda_1 - \lambda_2$ represents the saliency of the edge.

The voting process is divided into ball voting and stick voting. Ball voting identifies the edge orientation, whereas stick voting reinforces the given orientation. The voting fields are illustrated in Fig. 3. A voter (vote-caster) and a votee (vote-receiver) are randomly chosen from the point set.

3.1 Multi-frame sparse tensor voting

In this work, multiple frame integration is incorporated into the tensor voting scheme. As illustrated in Fig. 4, the edge points stored in the tensor field continue to receive stick voting. Since the edge points are appended to the tensor field at every frame, the points with small stickness values should be discarded. As a result, noise and position errors in the tensor field are reduced, both of which contribute to accurate map construction. Unlike directional stick voting, the ball voting stage should be frame-independent, because mixing multiple frames together in ball voting can degrade the orientation estimation performance.

Another difference from the standard tensor voting framework is that only a sparse tensor field is constructed; namely, both voters and votees are limited to edge points. Road markings are recognized by the sparse tensor field instead of an orthoimage.

3.2 Calculus

Three-dimensional vector notation is used here instead of tensor notation. A tensor $S = [S_{xx} S_{xy}; S_{yx} S_{yy}]$ can be represented by a vector $S = [S_{xx} S_{xy} S_{yx}]^\top$ because $S_{xy} = S_{yx}$. Suppose $r$ is a distance between the voter and the votee, $\alpha$ is an angle of the votee viewed from the voter, and $\beta$ is an orientation of the voter. The votee is updated by the following equations.

Ball voting:

$$S \leftarrow S + \frac{1}{2} e^{-\frac{r^2}{\sigma^2}} \begin{bmatrix} 1 + \cos 2\alpha \\ \sin 2\alpha \end{bmatrix}$$

Stick voting (only if $|\alpha - \beta| > \pi/4$):

$$S \leftarrow S + \frac{1}{4} e^{-\frac{2 \sigma^2}{\pi^2} \frac{r^2}{\sigma^2}} \begin{bmatrix} 1 + b_c & -2b_s & 1 - b_c \\ b_s & 2b_c & -b_s \\ 1 - b_c & 2b_s & 1 + b_c \end{bmatrix} \begin{bmatrix} 1 + d_c \\ d_s \\ 1 - d_c \end{bmatrix}$$

The stickness $\lambda_1 - \lambda_2$ and the orientation $\psi$ are calculated by

$$\lambda_1 - \lambda_2 = \sqrt{4S_{xy}^2 + (S_{yy} - S_{xx})^2}$$

$$\psi = \frac{1}{2} \arctan \frac{2S_{xy}}{S_{xx} - S_{yy}}.$$
Figure 5. Construction of the search table

Table 1 shows the averaged position estimation errors from the ground truth. The proposed method out-

Figure 6. Position estimation. Hypothesis is given from the sparse tensor field via the search table.

4. Recognition

The road marking recognition is realized by voting of edge indices. Compared with the character detection method [9] and the generalized Hough transform [10], which require a 2D (or more) voting space, the position estimation error is smaller because the degrees of freedom are limited.

Figure 5 shows a table for searching fine segments of road marking contours. The query parameters including the angle difference $\theta$, the azimuth $\phi$, and the distance $r$ are calculated for each combination of two edge indices and stored in the table.

4.1 Position estimation

For the position estimation, a base point is chosen from the sparse field. The query parameters $(\theta, \phi, r)$ are calculated from a pair consisting of the base point and another point nearby. The edge index $i$ to be voted is referred by the query parameters and the search table. As a result of iterating this process using various nearby points, the hypothesis of the road marking position is obtained as shown in Fig. 6. Multiple hypotheses can be obtained by choosing several base points.

4.2 Classification

The hypothesis with the largest similarity is accepted as the classification result. First, local similarities are calculated by comparing the sparse tensor field and the projected hypothesis. For each edge index $k$, the angle difference between the edge orientation $\beta_k$ and the orientation $\psi$ of the nearest tensor is calculated and denoted by $\theta_k = |\beta_k - \psi|$. Suppose that the local similarity is defined as $\cos \theta_k$. The similarity of the hypothesis is the sum of the weighted local similarities calculated by

$$P = \frac{\sum_k \cos \theta_k \log[H(\beta_k) + 1]/H(\beta_k)}{\sum_i \log[H(\beta_i) + 1]/H(\beta_i)},$$

where the weight is determined from a saturation-like function $w(t) = \log[t + 1]/t$ of the edge direction histogram $H(\beta_k)$. The reason for using this function is to restrict the contribution of long edge segments to the classification.

5. Experimental results

Experiments were performed to test the classification and position estimation performance. The road marking map was generated using a rear-view camera (KENWOOD CMOD-200) and a high-accuracy GPS [11] in order to exclude GPS errors from the performance evaluation. The recognition targets were 63 road markings (e.g., directional arrows, stop signs, and speed limit signs) with various levels of degradation. The number of categories was 20. The proposed method was compared with orthonormal-based recognition methods using the HOG features [12] or template matching by the subspace method [13]. The resolution of the orthoimage was 20 [pixels/meter].

Figure 7 shows the recognition accuracy (detected and classified into the correct category) and the false detection rate at each detection threshold. The template matching method was not effective because some road markings were seriously degraded and even partially missing. Although the method using HOG exhibited robustness against degradation, the proposed method was more effective for reducing false detections. This result of the proposed method is because the sparse tensor field provided more reliable geometric features than did the orthoimage analysis. Figure 8 shows an example of the road marking recognized only by the proposed method, together with its tensor field. The road marking was identified by its characteristic parts, even though the other parts were seriously degraded.

Table 1 shows the averaged position estimation errors from the ground truth. The proposed method out-
performed the orthoimage-based methods, since the direct use of the sparse tensor field eliminated false hypotheses of the position estimation. Moreover, the sparse tensor field did not suffer from resolution limitations of the orthoimage. The localization result by the generalized Hough transform is also shown. Compared with the generalized Hough transform, the voting table of the proposed method was considerably small, which was effective for restricting position estimation errors. An example of the constructed map is shown in Fig. 9.

6. Conclusion

A road marking recognition method combining two voting schemes is presented. The performance of position estimation and recognition is experimentally demonstrated. In future work, undefined markings and 3D objects will be investigated.

References


Table 1. Position estimation errors

<table>
<thead>
<tr>
<th>Method</th>
<th>Error distance [m]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Proposed method</td>
<td>0.227</td>
</tr>
<tr>
<td>HOG</td>
<td>0.436</td>
</tr>
<tr>
<td>Template matching</td>
<td>0.402</td>
</tr>
<tr>
<td>Generalized Hough transform*</td>
<td>0.340</td>
</tr>
</tbody>
</table>

*The position estimation process of the proposed method is replaced by the generalized Hough transform.

Figure 7. Classification performance

Figure 8. Detected road marking (Speed limit “40”) and its tensor field

Figure 9. Example of generated map


