# Identification of degraded traffic sign symbols by a generative learning method

Hiroyuki Ishida, Tomokazu Takahashi, Ichiro Ide, Yoshito Mekada, and Hiroshi Murase
Graduate School of Information Science, Nagoya University
Furo-cho, Chikusa-ku, Nagoya, Aichi, 464-8601, Japan
{hishi, ttakahashi, ide, mekada, murase}@murase.m.is.nagoya-u.ac.jp

## Abstract

*We present a novel training method for recognizing traffic sign symbols undergoing image degradations. In order to cope with the degradations, it is desirable to use similarly degraded images as training data. Our method artificially generates these data from an original image in accordance with the actual degradations. We experimentally confirmed the usefulness of our method for the camera-based traffic sign recognition.*

## 1. Introduction

Technologies for supporting car drivers with car-mounted cameras have gained attention in recent years. Traffic sign recognition is one of the important tasks. The two main issues in traffic sign recognition are detection and classification. Various attempts have been carried out on the detection of traffic signs. They detect signs using: edge detection mask [1], hierarchical template [2], shape information [3], and color information [4]. On the other hand, relatively few studies have been conducted on the category classification of extracted signs. Furthermore, most of them are oriented mainly toward high-quality images [5]–[7]; few studies have focused on low-quality images.

Our work focuses on the classification of low-quality traffic sign symbols taken by a car-mounted camera. A difficult problem in classifying traffic sign symbols is image degradation. In order to recognize symbols undergoing the degradations, training data ought to be actually captured images in a similar condition. However, it is difficult and unrealistic to collect all the training data under various conditions.

In this paper, we propose a generative learning method in which the training data are generated automatically in accordance with the actual degradations. We define generation parameters corresponding to degradation factors; combining these parameters produces a large variety of degradations. Our method estimates parameter distribution from
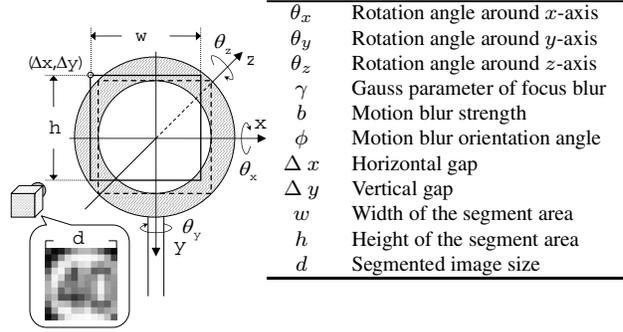


**Figure 1. Parameters for the generation models.**

| | |
|---|---|
| $\theta_x$ | Rotation angle around $x$-axis |
| $\theta_y$ | Rotation angle around $y$-axis |
| $\theta_z$ | Rotation angle around $z$-axis |
| $\gamma$ | Gauss parameter of focus blur |
| $b$ | Motion blur strength |
| $\phi$ | Motion blur orientation angle |
| $\Delta x$ | Horizontal gap |
| $\Delta y$ | Vertical gap |
| $w$ | Width of the segment area |
| $h$ | Height of the segment area |
| $d$ | Segmented image size |

actually taken images. For this purpose, we search for an optimal generation parameter set from each image by Genetic Algorithm (GA) [8].

This paper is organized as follows: Section 2 introduces the generation parameters. In Section 3, the generative learning method is introduced, and in Section 4, the Subspace method [9] for the recognition step is described. Results are presented in Section 5.

## 2. Degradation models

Training data are generated from an original image using three generation models : rotation, blur, and segmentation. These models are defined with the generation parameters listed in Figure 1. The generation process from the original image $P_0$ to a generated image $P_4$ is given below:

**(a)** Rotation

$$P_1(x,y) = P_0(x',y') \qquad (1)$$

$$[x', y', z']^t = \left( \boldsymbol{R_z}(\theta_z)\boldsymbol{R_y}(\theta_y)\boldsymbol{R_x}(\theta_x) \right)^{-1} [x, y, 0]^t$$

where $\boldsymbol{R_x}, \boldsymbol{R_y}$, and $\boldsymbol{R_z}$ denote the rotation matrices around each axis.

**(b)** Blur

$$P_2(x,y) = P_1(x,y) * \left[ \frac{1}{2\pi\gamma^2} \exp\left( -\frac{x^2 + y^2}{2\gamma^2} \right) \right] \quad (2)$$
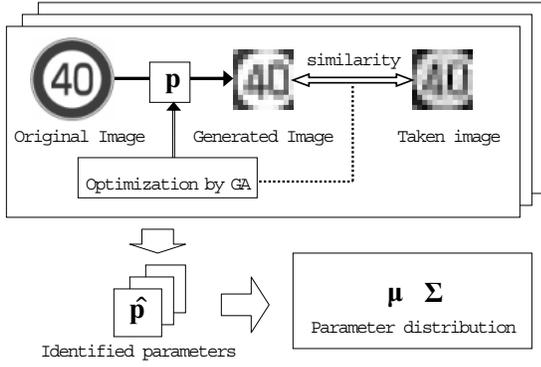
**Figure 2. Estimation step.**
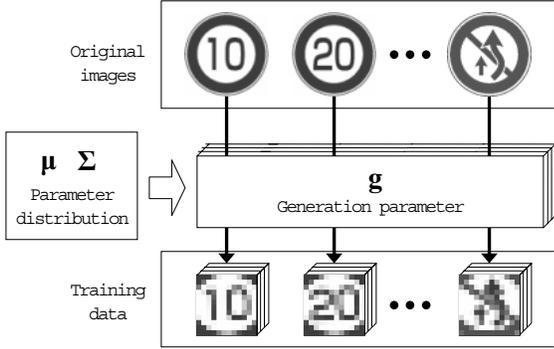


**Figure 3. Generation step.**

$$P_3(x,y) = \int_{-\frac{1}{2}}^{\frac{1}{2}} P_2(x - bt\cos\phi, y - bt\sin\phi)dt \quad (3)$$

**(c)** Segmentation

$$P_4(i,j) = \sum_{x,y \in D_{(i,j)}} P_3(x,y) \quad (4)$$

$$D_{(i,j)} = \left\{ (x,y) \ \middle| \ \frac{i}{d+1}w \leq x - \Delta x < \frac{i+1}{d+1}w, \right.$$
$$\left. \frac{j}{d+1}h \leq y - \Delta y < \frac{j+1}{d+1}h \right\}$$

## 3  Generative learning method

We propose a generative learning method which allows us to generate various training data without actually collecting them in various conditions. This method consists of an estimation step (Figure 2) and a generation step (Figure 3).

### 3.1  Parameter estimation

Distribution of generation parameters are estimated from the actual images, which is used to simulate degradations. In order to estimate the distribution, the parameters for each

**Table 1. Parameter identification algorithm.**

| | |
|---|---|
| $M$ : Population size, | $G$ : Number of generations |
| $P_c$ : Crossover rate, | $P_m$ : Mutation rate |
| $C_p$ : Parents set, | $C_c$ : Children set |
| $t$ : Vectorized taken image $T$ | |
| $q$ : Vectorized generated image $Q$ | |

1.   initialize set $C_p$ and its $M$ chromosomes $p_i$
2.   **do**
3.     **for** all $p_i \in C_p$
4.       generate $q_i$ from the original image of $t$ with $p_i$
5.       calculate fitness $s_i = q_i \cdot t$
6.     **do**
7.       select chromosomes $p_a, p_b$ by roulette selection
8.       reproduce $p_a \to p'_a, p_b \to p'_b$
9.       **if** $\text{Rand}[0,1) < P_c$ **then** cross $p'_a$ with $p'_b$
10.      add $p'_a, p'_b$ to $C_c$
11.     **until** $|C_p| = |C_c|$
12.    **for** each chromosome $p_i$ of $C_c$
13.      **if** $\text{Rand}[0,1) < P_m$ **then**
14.       initialize one of the element of $p_i$ randomly
15.    copy $C_c \to C_p$
16.    empty $C_c$
17.   **until** generation reaches $G$
18.   $\hat{p} := p_i$ with the largest fitness $s_i$
19.   **return** $\hat{p}$

image needs to be identified. A parameter vector $p$ consists of following generation parameters:

$$p = (\theta_x, \theta_y, \theta_z, \gamma, b, \phi, \Delta x, \Delta y, w, h). \quad (5)$$

Let $T$ be a taken image for the estimation, and $Q$ be an image generated from the original image of $T$ with $p$. We find the parameter set $\hat{p}$ which maximizes the similarity between $Q$ and $T$, and accept it as the generation parameters of $T$. Table 1 describes the parameter identification algorithm based on GA.

The parameter distribution is estimated from multiple $\hat{p}$ identified from taken images. We assume normal distribution, hence an average vector $\mu$ and a variance-covariance matrix $\Sigma$ are obtained from multiple $\hat{p}$ by

$$\mu = \mathcal{E}[\hat{p}], \quad (6)$$
$$\Sigma = \mathcal{E}\left[(\hat{p} - \mu)(\hat{p} - \mu)^t\right]. \quad (7)$$

### 3.2  Generation step

A parameter vector $g$ which follows the estimated distribution is generated by the following producing function:

$$g = \Sigma^{\frac{1}{2}}r + \mu \quad (8)$$

where $r$ denotes a vector consisting of standard normal random numbers [10], and $\Sigma^{\frac{1}{2}}$ denotes the Cholesky decomposition [11] of $\Sigma$. As illustrated in Figure 3, various parameter vectors from this equation, and correspondingly, various training data of all categories and sizes are generated.

# 4 Recognition by the subspace method

## 4.1 Construction of a subspace

We employ the subspace method [9] in the recognition step. Let $\boldsymbol{x}_{n,r}^{(c)}$ $(n = 1, \cdots, N)$ be normalized vector from category $c$'s training data whose size is $r$. The autocorrelation matrix $\boldsymbol{X}_r^{(c)}$ is computed by

$$\boldsymbol{X}_r^{(c)} = \left[\boldsymbol{x}_{1,r}^{(c)}, \cdots, \boldsymbol{x}_{N,r}^{(c)}\right]\left[\boldsymbol{x}_{1,r}^{(c)}, \cdots, \boldsymbol{x}_{N,r}^{(c)}\right]^t. \quad (9)$$

Eigenvectors are derived from $\boldsymbol{X}_r^{(c)}$, of which $\boldsymbol{e}_{l,r}^{(c)}$ ($l = 1, \cdots, L$) with the largest $L$ ($< N$) eigenvalues are used for the recognition.

## 4.2 Multiframe integration

The subspace method recognizes given images by comparing the similarities to subspaces. Yanadume et al. demonstrated that integrating information from multiple frames improves recognition accuracy [12]. We apply their method to traffic sign symbols. Given $M$ frames of the same target, the recognition result of the target images $\boldsymbol{y}_m$ ($m = 1, \cdots, M$) whose size is $r_m$ is obtained by,

$$\hat{c} = \arg\max \sum_{m=1}^{M} \sum_{l=1}^{L} (\boldsymbol{e}_{l,r_m}^{(c)} \cdot \boldsymbol{y}_m)^2, \quad (10)$$

## 4.3 Round sign detection

Here we briefly introduce an algorithm for detecting round signs. The red circumference of a traffic sign is found by the following discriminant function:

$$\text{red}(x,y) = \begin{cases} 1 & \left(\begin{array}{c} -\frac{\pi}{9} < H(x,y) < \frac{\pi}{9} \\ \text{and } 0.2 < S(x,y) \leq 1 \\ \text{and } 30 \leq V(x,y) \leq 255 \end{array}\right) \\ 0 & \text{otherwise} \end{cases}, (11)$$

where $H, S, V$ represent the intensities in HSV color space [13]. Round signs can be detected by matching a doughnut-shaped structure. The area to be segmented is the smallest square which includes the whole symbol region.

# 5 Experiment

We performed experiments using video data taken by a car-mounted camera (Table 2). The number of categories was twenty. Figure 4 illustrates the twenty traffic signs used in Japan. The video data contained sixteen traffic signs: two of no.2, six of no.4, three of no.5, three of no.12, and two of no.20. We used one of the video streams containing sign no.4 for parameter estimation, while the other fifteen

| Product model | Sony DCR-PC105 |
|---|---|
| Resolution | $720 \times 480$ |
| Frame rate | 30 fps |
| Focus length | 3.7 mm |
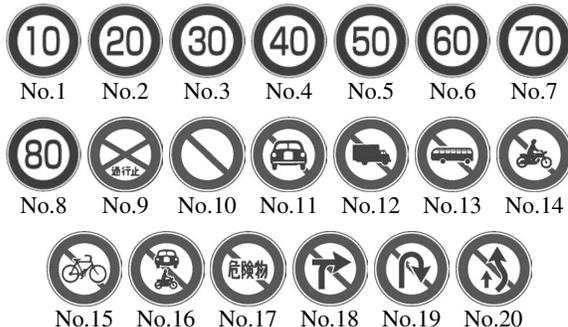


Figure 4. Traffic sign categories.
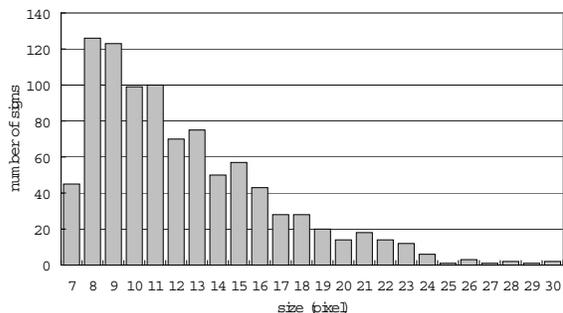


Figure 5. Distribution of the traffic sign size.

videostreams were used as test data. Frames in which signs were not detected were excluded from test data. Figure 5 shows the size distribution of segmented test data, and Figure 6 shows some examples of the test data.

Parameter distribution was estimated from 40 traffic sign images (size 8–24). We applied the algorithm in Table 1 with $M = 100$, $G = 100$, $P_c = 0.7$, and $P_m = 0.01$. Table 3 shows the average $\boldsymbol{\mu}$ and the standard deviation $\text{diag}(\boldsymbol{\Sigma}^{\frac{1}{2}})$ of the estimated distribution.

For this experiment, we used a parameter producing function in which $\boldsymbol{\Sigma}^{\frac{1}{2}}$ was weighed on as

$$\boldsymbol{g} = k\boldsymbol{\Sigma}^{\frac{1}{2}}\boldsymbol{r} + \boldsymbol{\mu}. \quad (12)$$

We compared five cases, where $k = \frac{1}{4}, \frac{1}{2}, 1, 2, 4$. When $k = 1$, it is identical to the proposed method. We obtained recognition results by equation (10) using ten successive frames.

The average recognition rate of the proposed method ($k = 1$) was 98.2%. This results demonstrates the effectiveness of generating training data using the defined models. The recognition rates according to the size of the segmented

| | No.4 | No.5 | No.12 | No.20 |
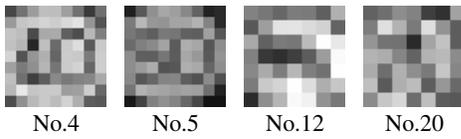
**Figure 6. Test data.**

**Table 3. Estimated distribution.**

| | $\mu$ | $\mathrm{diag}(\Sigma^{\frac{1}{2}})$ | | $\mu$ | $\mathrm{diag}(\Sigma^{\frac{1}{2}})$ |
|---|---|---|---|---|---|
| $\Delta x$ [cm] | 2.08 | 1.16 | $\theta_y$ [°] | $-1.47$ | 4.41 |
| $\Delta y$ [cm] | 2.28 | 1.44 | $\theta_z$ [°] | $-4.97$ | 1.77 |
| $w$ [cm] | 38.3 | 1.97 | $\gamma$ | 12.0 | 4.38 |
| $h$ [cm] | 42.9 | 1.84 | $\phi$ [°] | 11.5 | 49.2 |
| $\theta_x$ [°] | 0.56 | 4.72 | $b$ [cm] | 0.05 | 1.70 |



**Figure 7. Recognition results according to the maximum sign size using multiframes.**

signs are presented in Figure 7 along with the results from other $k$s. It is worthy of noting that it was the most appropriate case for recognizing signs taken in similar conditions when $k = 1$. This result indicates that the GA-based estimation successfully worked, and also exhibited the superiority of the proposed method for small signs. Since most of the available sign images are small as shown in Figure 5, the robustness to size reduction should be important for real-world applications.

## 6 Conclusion

In this paper, we proposed a generative learning method for recognizing traffic sign symbols. In order to simulate actual degradations, we defined degradation parameters, and estimated the parameter distribution from taken images. We experimentally proved the effectiveness of our method for car-mounted cameras.

The proposed method is applicable for any sign by combining it with other detection methods [1]–[7]. As to future research, the effectiveness of the method under various weather conditions should be evaluated.

## Acknowledgement

## References

[1] A. Escalera and M. Salichs, Road traffic sign detection and classification. *IEEE Trans. Industorial Electronics*, vol.44, no.848–859, December 1997.

[2] D. Gavrila, Multi-feature hierarchical template matching using distance transforms. *Proc. IEEE 14th International Conference on Pattern Recognition*, vol.1, no.439–444, August 1998.

[3] J. Miura, T. Kanda, and Y. Shirai, An active vision system for real-time traffic sign recognition. *Proc. IEEE Intelligent Transportation Systems*, no.52–57, June 2000.

[4] G. Mo and Y. Aoki, A recognition method for traffic sign in color image (In Japanese). *IEICE Trans.* vol.J87-D-II, no.12, pp.2124–2135, December 2004.

[5] S. Lafuente-Arroyo, P. Gil-Jimenez, R. Maldonado-Bascon, and F. Lopez-Ferreras, Traffic sign shape classification evaluation I: SVM using Distance to Borders. *Proc. IEEE Intelligent Vehicles Symposium*, pp.557–562, June 2005.

[6] C. Bahlmann, Y. Zhu, V. Ramesh, M. Pellkofer, and T. Koehler, A system for traffic sign recognition, tracking, and Recognition using color, shape, and motion information. *Proc. IEEE Intelligent Vehicles Symposium* : 255–260, June 2005.

[7] P. Gil-Jimenez, S. Lafuente-Arroyo, H. Gomez-Moreno, F. Lopez-Ferreras, and S. Maldonado-Bascon, Traffic sign shape classification evaluation II: FFT applied to the signature of blobs. *Proc. IEEE Intelligent Vehicles Symposium*, pp.607–612, June 2005.

[8] L. Daris, Handbook of genetic algorithms, Van Nostrand Reinhold, New York, 1991.

[9] E. Oja, Subspace methods of pattern recognition, Research studies, Hertfordshire, UK, 1983.

[10] J. von Neumann, Various techniques used in connection with random digits, National Bureau of Standards Series, no.12, pp.36–38, 1951.

[11] G. Golub and C. Loan, Matrix computations, 2nd ed., Cambridge Univ. Press, Cambridge, MA, 1992.

[12] S. Yanadume, Y. Mekada, I. Ide, and H. Murase, Recognition of Very Low-resolution Characters from Motion Images. *Proc. PCM2004*, Lecture Notes in Computer Science, Springer-Verlag, 3331, pp.247–254, December 2004.

[13] A. Smith, Color Gamut transform pairs, *Computer Graphics*, vol.12, no.3, pp.12–19, August 1978.