Discontinuous Character Recognition Method in Variable Message Signs for Intelligent Transportation Systems

Wahyono, Han-Sung Park, and Kang-Hyun Jo

***Abstract*—Variable Message Signs (VMSs) are a key element of Intelligent Transportation Systems. The VMSs generally displays dynamic text information that each character consists of matrix of a light-emitting diodes lamp, named LED character. Compared with general printed text, the LED character is quite difficult to be recognized due to its discontinuity. Hence, this paper addresses a novel method for recognizing the LED character. Firstly, color segmentation is performed in order to distinguish foreground (i.e. text) and background. The supporting points (SP) represented the center of a segment are then acquired. For generating character region, these points are grouped based on their properties. Lastly, feature extraction is applied by dividing component into several blocks and counting the number of SP in the block. Afterwards, character template matching approach is implemented for recognizing component as certain alphanumeric. Experimental results evince that the proposed method is a robust to recognize the LED text.**

***Index Terms*—Character template matching, LED text, intelligent transportation systems, variable message signs.**

I. Introduction

The VMSs are playing increasingly important roles in attempts to improve highway safety, operations, and use of existing facilities. The VMSs are electronic traffic sign devices used for traffic warning, regulation, routing and management, and are intended to affect the behavior of drivers by providing real-time traffic-related information. In urban areas, the VMSs are used within parking guidance and information systems to guide drivers to available car parking spaces. They may also ask vehicles to take alternative routes, limit travel speed, warn of duration and location of the incidents or just inform of the traffic conditions [1]. Fig. 1 shows several the variable message signs in urban areas.

Generally, the VMSs are displayed as a Light-Emitting Diodes matrix text (LED text). Each character of the LED text consists of a matrix of LEDs or matrix of segments with circle or rectangular shape. On the other hand, in recent years, text recognition plays a significant part in intelligent transportation systems (ITS) such as vehicle plate recognition, road navigation, and information retrieval from traffic sign. Many researchers have studied text detection and/or recognition with significant result [2]-[7], [10]-[11]. Unlike general text, LED text is relative difficult to be recognized because of discontinuous character. Even so, text recognition system as part of the ITS that is built should be able to handle this type of text. Unfortunately, based our survey, researchers rarely involve this type of text on their text recognition systems. Hence, in this work, we propose a method to recognize individual character on the LED text. This method will be described detail in Section II. The experimental results from our method will be presented in Section III. Section IV concludes the paper and discusses our future works.

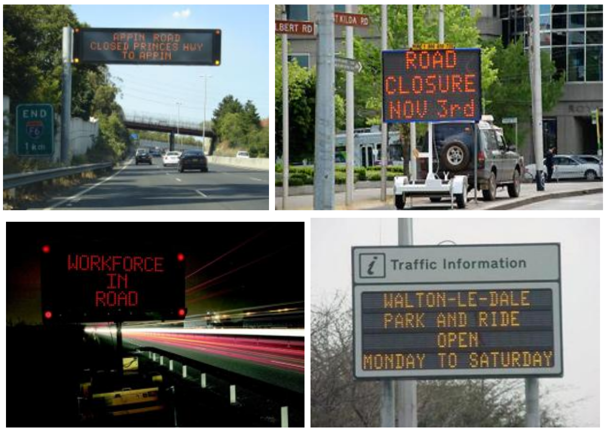


Fig. 1. Several sample of variable message sign in urban areas.

II. The Proposed System

A. Overview of the System

The proposed system consists of the following two main stages: detection and recognition. Prior to do these processes, the variable message signs need to be localized. To do such a task, color segmentation can be performed well as it is composed by a LED lamp as foreground (commonly red and orange color) and its background is always in black color. Afterwards, our method is performed on the detected variable message sign (VMS image). The proposed method consists of four processes: 1) color segmentation, 2) region generating, 3) feature extraction, and 4) character recognition. These processes, as described in detail in following section, are broadly illustrated in Fig. 2.

B. Segmentation Based on Color Model

Most of the VMS image consists of black background and LED character as foreground with either red or orange color. Therefore, we utilize color segmentation in order to localize character region. Firstly, VMS images are converted into grayscale value. Then the candidate regions are found by using gray scale color model on the basis of our training dataset. Region detection is based on its color properties, such as mean and standard deviation values of gray scale. To estimates these properties, 30 images of the VMS image are

Manuscript received September 16, 2013; revised November 15, 2013.

Wahyono and Han-Sung Park are with Graduate School of Electrical Engineering, University of Ulsan, Korea (e-mail: wahyono@islab.ulsan.ac.kr, hspark@islab.ulsan.ac.kr).

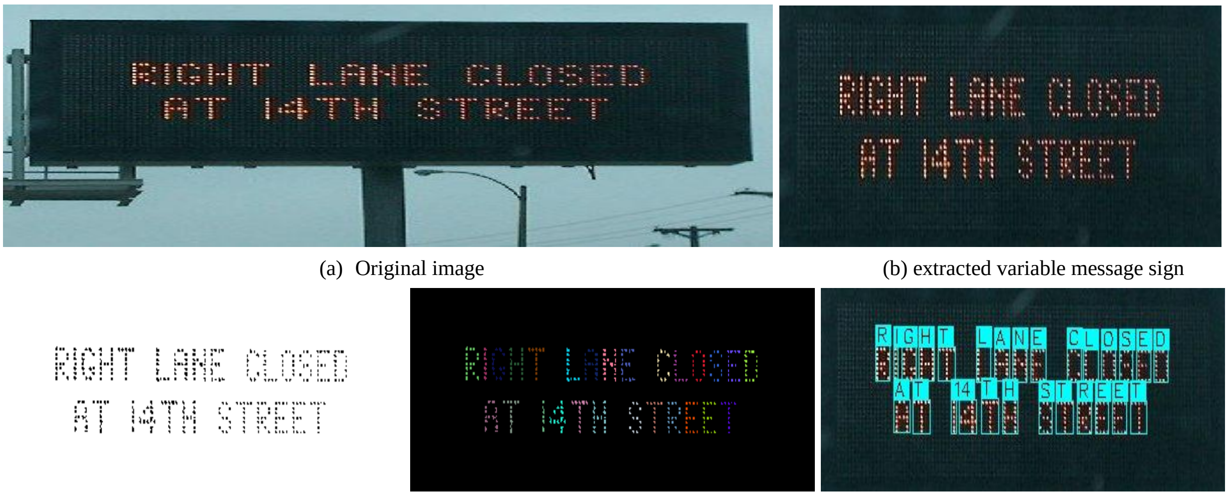
Kang-Hyun Jo is with School of Electrical Engineering, University of Ulsan, Korea (e-mail: acejo@ulsan.ac.kr).

DOI: 10.7763/IJMLC.2013.V3.362

taken under different lighting and weather conditions. By training from those sample data, the binarization is performed in (1)

$$B(x,y)=\begin{cases}1, & G(x,y)\le\mu_b\\ 0, & otherwise\end{cases} \tag{1}$$

where 1 and 0 are defined as background and foreground, . respectively, μ_b is average of gray value of foreground in sample data, while $G(x,y)$ is gray value of pixel. As shown in Fig. 2, after applying color segmentation on Fig. 2(b), we will obtain the binarization result as Fig. 2(c).



(a) Original image

(b) extracted variable message sign

(c) Supporting point extraction

(d) character region grouping

(e) detected and recognized character

Fig. 2. The illustration of the process of character recognition in variable message sign.

Algorithm 1 Supporting Point Extraction

```
Input
    - Set of foreground pixel points $P=\{p_1,p_2,\ldots,p_n\}$, where $p_i=\{x,y\}$ and $n$ is the number of points.
Output
    - Set of supporting points $SP=\{sp_1,sp_2,\ldots sp_m\}$, where $sp_i=\{x,y,\mu,AC\}$ and $m$ is the number of SP.
Process
For all foreground pixel $p_i(x,y)$:
1: Follow the ray $r$ in four directions alternately until background pixel $q_0$ is found,
       $Ray_{right} \rightarrow r^{(k+1)}(x,y)= r^{(k)}(x+1,y)$
       $Ray_{left} \rightarrow r^{(k+1)}(x,y)= r^{(k)}(x-1,y)$
       $Ray_{up} \rightarrow r^{(k+1)}(x,y)= r^{(k)}(x,y-1)$
       $Ray_{down} \rightarrow r^{(k+1)}(x,y)= r^{(k)}(x,y+1)$
2: Determine $q_i$ as rotated $q_0$ by $\alpha_i$ counterclockwise
       $\alpha_i=\frac{2\pi i}{b} \quad i=1..b-1$
3: Compute $d_i=\|p-q_i\| \quad i=0..b-1$
4: Calculate
       $N=\sum_{i=0}^{b-1} f(q_i)$
       $\mu=\frac{1}{N}\sum_{i=0}^{b-1} d_i\, f(q_i)$
       $\sigma^2=\frac{1}{N}\sum_{i=0}^{b-1}(d_i-\mu)^2 f(q_i)$
   where $f(q_i)=1$ if $q_i$ is edge pixel, $0$ otherwise
5: Point $p_i(x,y)$ is considered as supporting points if:
       $\frac{N}{b}\le T_1$ and $\sigma^2\le T_2$
6: Store properties of $p_i$ as properties of $sp_i=\{x,y,\mu,AC\}$, where AC is defined as average colors from point $p_i$ and its 8-neighboring pixels.
```

C. Region Generating

In LED text, each character consists of more than one small dot region, called blobs. Consequently, we need to combine them for generating character region. Rather than processing all the foreground pixels on the VMS image, we prefer to extract the supporting point (SP), representing the center of a blob. From this fact, the number of SPs is always less than the number of foreground pixels. In consequence, the processing time will be reduced.

To acquire SP, as shown in Fig. 3, we propose a new method as described details in Algorithm 1, instead of using circle detection [8] due to large number of blob, processing time and uncertainly blob shape.

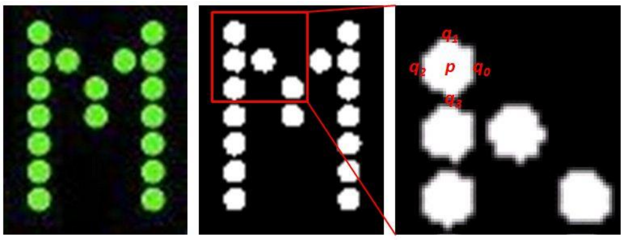


Fig. 3. The supporting point extraction. The center image is the binary image after applying color segmentation on the left image. In the right image, the point p is the supporting point candidate, while q_0, q_1, q_2 and q_3 are associated points of p (in this case m=4).

The output of the previous step is a set of supporting points sp_i and their properties such as the location (x_i,y_i), radius μ_i, and average color AC_i. Afterwards, these points are grouped into several character regions. Let denote sp_1 and sp_2 are the first and the second supporting point. Inspired from connected component labeling [9], point sp_1 and sp_2 may be grouped together as the same region if they have similar radius, similar average color, and the distance between them is close enough. The details of character regions generating process can be found at Algorithm 2. For preliminary filter, the region which has supporting point less than 10 is discarded, since we assume that the character in the LED text consists of at least ten blobs. As shown in Fig. 2(d) foreground pixels are labeled as several character regions. Different color represents different character region.

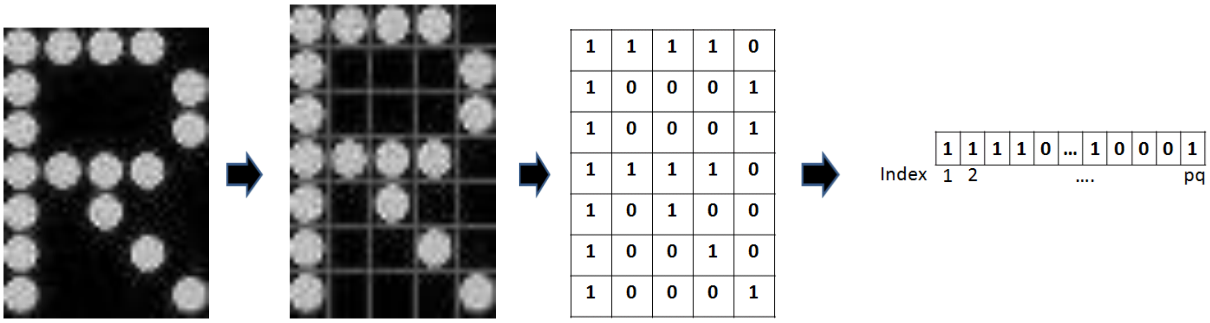
```
Algorithm 2 Character Region Generating
Input
    - Set of supporting points $SP=\{sp_1,sp_2,...sp_n\}$, where
      $sp_i=\{x,y,\mu,AC\}$ and $n$ is the number of SP.
Output
    - Set of character regions $CR=\{c_1,c_2,...c_m\}$, where
      $c_k=\{sp_1,sp_2,...,sp_{cn(k)}\}$, $m$ is the number of character
      regions, and $cn(k)$ is the number of supporting points
      in character region $k$.
Process
Set $sp_i\{label\}$ is not labeled for all supporting points.
for i: 1 to n do
for j: 1 to n do
if $(i=j)$ back to step 3;
compute $d_{ij}=dist(sp_i,sp_j)$
if $(d_{ij}>2*(\mu_i+\mu_j))$ back to step 3;
if $(\mu_i+\mu_j>T_3)$ back to step 3;
compute $colorDistance_{ij}=dist(AC_i,AC_j)$
if $(colorDistance_{ij}>T_4)$ back to step 3;
if $sp_i$ and $sp_j$ are not labeled
create new character region $k$
set $sp_i\{label\}$ and $sp_j\{label\}$ equal to $k$
else if $sp_i$ is labeled and $sp_j$ is not labeled
set $sp_j\{label\}= sp_i\{label\}$
else if $sp_i$ is not labeled and $sp_j$ is labeled
set $sp_i\{label\}= sp_j\{label\}$
else
$k1= sp_i\{label\}$;   $k2= sp_j\{label\}$; $minK=\mathbf{min}(k1,k2)$
set $sp_i\{label\}$ and $sp_j\{label\}$ equal to $mink$
merge character region $k1$ and $k2$
```

D. Feature Extraction

Based on our survey, most of character recognition methods employ character region and border to extract feature [2], [3], [11], [12]. In region-based, the feature is usually extracted from the statistical attributes, such as occupation ratio, aspect ratio, vertical and horizontal voting, . etc. Whilst in border-based, the feature is generally gained by exploiting the gradient of border. However, if we apply these methods, the extracted feature cannot distinguish the characteristic of character due to character component in the LED text only has a set of blobs. Though region can be formed using region growing from the blobs, but it will take high cost in processing time. Therefore, we consider exploiting the spatial distribution of the SPs from previous step.

```
Algorithm 3 Feature Extraction
Input
    - Character region $CR$
    - LED template size $pxq$
Output
    - Feature vector $F=\{f_1,f_2,...,f_{pq}\}$.
Process
Divide region $CR$ into $pxq$ cells.
for $i$: 1 to $p$ do
for $j$: 1 to $q$ do
set $b_{ij}$ as the number of SPs in cell [row $i$ , column $j$]
set $k=(pi+j)+1$
set $f_k=\begin{cases}1 & if\ b_{ij}\ge 3\\ 0 & otherwise\end{cases}$
return $F=\{f_1,f_2,...,f_{pq}\}$.
```

In order to extract feature, the character region is divided into pxq cells, where pxq is the size of LED character. Hereinafter, b_{ij}, denoted as the number of SPs in cell at row i and column j, is quantized. Ideally, its value should be equal to 1. However, through our process, it may more than threshold. According to our experiment, we set threshold equal to 3. To overcome this issue, lastly, the quantization result has to be converted as binary vector. As the result that pq elements vector $F=\{f_1,f_2,...f_{pq}\}$ is acquired that will be compared later with template feature (for illustration, see Fig. 4). Algorithm 3 expresses our feature extraction process.



1	1	1	1	0
1	0	0	0	1
1	0	0	0	1
1	1	1	1	0
1	0	1	0	0
1	0	0	1	0
1	0	0	0	1

Fig. 4. The illustration of feature extraction (see text for details).

Based on our database template, LED character size may be 3×5, 4×6, 5×7, 5×8, 6×7, and 6×8 dots. In contrast, we do not know exactly the character size in the test image. Consequently, we should determine the appropriate size of the LED for each character region, defined as minimum bounding box. Let denote k be the number of LED character size and p_ixq_i be the possible LED character size in the database template, where i=1...k. The appropriate LED character size for mxn character region is defined as (2)

$$\min_i \left|\frac{m}{p_i}-\frac{n}{q_i}\right| \qquad (2)$$

E. Character Recognition

To recognize LED character, the common text recognition method could not be performed directly since character candidate does not contain of adjacent points as general text. Hence, in the last process of our method, the filtering as well as recognizing schema are performed by featured-based template matching. First, we collected character (alpha-numeric) templates, as shown in Fig. 5, and then classify them into 62 classes ('a-z','A-Z','0-9'). We utilize Hamming distance [13] for comparing feature between region and template. The Hamming distance between two vectors of equal length is the number of positions at which the corresponding symbols are different. This distance function can be computed in one linear pass, O(n). Every candidate region is only compared to those templates that have the same size as the region to reduce the complexity of the problem. Let denote *d* (S_i,S_j) as hamming distance between region *i* and template *j* that is formulated as (3)

$$d(S_i,S_j)=\sum_{k=1}^{pq}\begin{cases}1 & if\ \ S_i[k]\neq S_j[k]\\ 0 & \text{otherwise}\end{cases} \qquad (3)$$

The character region *i* may be recognized as character *j*, if the *d*(S_i,S_j) between region *i* and template *j* is minimal compared with other templates.

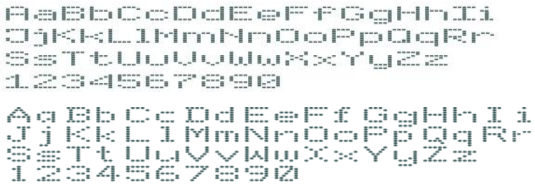


Fig. 5. Several samples of template that used in our system.

III. Experimental Results

The dataset contains 310 character templates and 124 images, which both taken using perpective camera and collected from Google image. The parameter values in experiment are set according to Table I. All parameters are empirically determined based on training template.

TABLE I: Parameter Setting

Parameter	*b*	T_1	T_2	T_3	T_4
Range value	[2,12]	[0,1]	[0,1]	[0,1]	[100,200]
Set	8	0.6	1	1	160

In order to evaluate our system, we divided evaluation category into two parts: evaluation for detection and evaluation for recognition. The following parameters are defined for each image. True positive (TP) and False positive (FP) are the number of true and false character region, respectively. True recognized character (TRC) and False recognize character (FRC) are denoted as the number of true character region recognized as true character and false character, respectively. The last, actual character region (ACR) is denoted as the number of true character region in ground truth. The performance measures are defined as follows:

- Detection Recall (DR) = TP/ACR
- Detection Precision (DP) = TP/(TP+FP)
- Detection F-measure (DF) = (2×DR×DP)/(DR+DP)
- Recognition True Rate (RTR) = TRC/ACR
- Recognition False Rate (RFR) = FRC/ACR

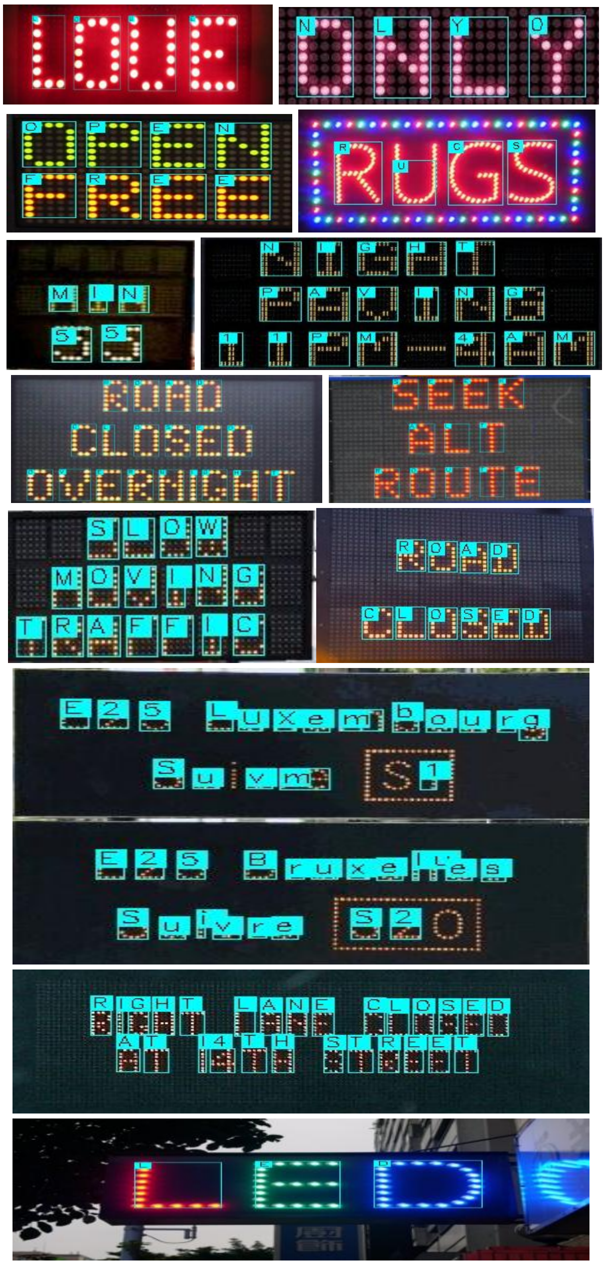


Fig. 6. Some example results of our proposed method. The results are marked with blue rectangle and recognized characters. Note that LED texts are detected in full image, but we only show cropped subimage due to space limitation.

Based on above measurements, the performance of our methods are shown in Tabel II. According to Table III, our method achieved 0.16 second per image or 7 fps rate. Since we have not found other works on LED dot-matrix detection and recognition in literature, we cannot compare our work with others. However, these results show that our proposed method to detect and recognized LED dot matrix is reliable and effective. Furthermore, Fig. 6 shows some detection and recognition results from our proposed method.

TABLE II: Performance of Our Method

Measurement	*DR*	*DP*	*DF*	*RTR*	*RFR*
Performance	87%	83%	84.9%	57%	18%

TABLE III: PROCESSING TIME EVALUATION

Process	*Average running time (ms)*
Color segmentation	9.5
Character region generating	97.8
Feature extraction	29.5
Recognition & bounding boxing	8.1
TOTAL TIME	144.9

IV. CONCLUSION

In this paper, a novel and fast method for detecting and recognizing the discontinuous characters often appeared in variable message sign board has been presented. In our method, feature-based template matching is utilized to filter as well as recognize character component as either character or non-character region. Assumed, the variable message sign region on the original image has been obtained. First, color segmentation is performed for obtaining binary image. From this binary image, the supporting points are extracted on it. These points then should be grouped as one component by the proposed heuristic rules. Experimental results demonstrate the effectiveness of our method.

Nevertheless, the method still has limitations, such as handling touching character and dealing with slant text. In the future work, these problems will be set and considered to recognition.

ACKNOWLEDGMENT

This research was supported by the MOTIE (The Ministry of Trade, Industry and Energy), Korea, under the Human Resources Development Program for Convergence Robot Specialists support program supervised by the NIPA(National IT Industry Promotion Agency) (H1502-13-1001).

REFERENCES

[1] W. F. Huang, “Designing a display unit to drive the 8x8 dot-matrix display,” in *Proc. IEEE 5th International Nanoelectronics Conferences (INEC*, 2013*)*, pp. 385-388.

[2] L. Neumann and J. Matas, “A method for text localization and recognition in real-world images,” in *Proc. 10th Asian Conference on Computer Vision (ACCV)*, 2010.

[3] K. Wang, B. Babenko, and S. Belongie, “End-to-End scene text recognition,” in *Proc. International Conference on Computer Vision (ICCV)*, 2011.

[4] J. Yi, Y. Peng, and J. Xiao, “Color-based clustering for text detection and extraction in image,” in *Proc. 15th International Conference on Multimedia*, 2007.

[5] C. Liu, C. Wang, and R. Dai, “Text detection in images based on color texture features,” in *Proc. ICIC*, 2005, vol. 1, pp. 40-48.

[6] J. Epshtein, E. Ofek, and Y. Wexler, “Detecting text in natural scenes with stroke width transform,” in *Proc. CVPR*, 2010.

[7] J. Zhang and R. Kasturi, “Text detection using edge gradient and graph spectrum,” in *Proc. International Conference on Pattern Recognition*, 2010.

[8] E. Cueevas *et al.*, “Fast algorithm for multiple-circle detection on images using learning automata,” *IET Image Processing*, 2012.

[9] L. He, Y. Chao, and K. Suzuki, “A run-based two-scan labeling algorithm,” *IEEE Transaction on Image Processing*, vol. 17, no. 5, pp. 749-756, 2008.

[10] Wahyono and K.-H. Jo, “A fast stroke width transform generating method for text detection,” in *Proc. QCAV*, 2013.

[11] S. Yi and Y. Tian, “Text string detection from natural scenes by structure-based partition and grouping,” *IEEE Transactions on Image Processing*, vol. 20, no. 9, pp. 2594–2605, 2011.

[12] J. J. Weinman, E. Learned-Miller, and A. R. Hanson, “Scene text recognition using similarity and a lexicon with sparse belief propagation,” *IEEE Transaction on Pattern Analysis and Machine Inteligence*, vol. 31, pp. 1733–1746, 2009.

[13] R. W. Hamming, “Error detecting and error correcting codes,” *Bell System Technical Journal*, vol.29, pp 147-160, April, 1950.

Wahyono received his Bachelor of Computer Science from Gadjah Mada University, Indonesia in 2010. He then joined Gamatechno Indonesia as a software engineering researcher until the end of 2011. Since 2012, he has been serving as an assistant lecturer in Department of Computer Science and Electronics, Faculty of Mathematics and Natural Science, Gadjah Mada University, Yogyakarta, Indonesia. He is currently a Ph.D. student, at the Graduate School of Electrical Engineering, University of Ulsan, Ulsan, Korea. He is actively participating as a member of the societies as IEEE, ICROS. His research interests include digital image processing, pattern recognition, machine learning, computer vision, and software engineering.

Han-Sung Park received his Bachelor of Electric Engineering from University of Ulsan, Korea in 2012. He is currently a master course student, at the Graduate School of Electrical Engineering, University of Ulsan, Ulsan, Korea. His research interests include Human Computer Interaction, augmented reality, computer vision, and digital image processing.

Kang-Hyun Jo received his Ph.D. degree from Osaka University, Japan in 1997. He then joined the School of Electrical Eng., University of Ulsan right after having one year experience at ETRI as a postdoc research fellow. Dr. Jo has been active to serve for the societies for many years as directors of ICROS (Institute of Control, Robotics and Systems) and SICE (Society of Instrumentation and Control Engineers, Japan) as well as IEEE IES TC-HF Chair. He is currently contributing himself as an editorial member for a few renowned international journals, such as IJCAS (International Journal of Control, Automation and Systems), TCCI (Transactions on Computational Collective Intelligence, Springer) and IteN (IES Technical News, online publication of IEEE). He had involved in organizing many international conferences such as ICCAS, FCV, IFOST, ICIC and IECON. He had visited for performing his research activity to Kyushu University, KIST and University of California Riverside. His research interest covers in a wide area where focuses on computer vision, robotics, and ambient intelligence.