2016 IEEE Intelligent Vehicles Symposium (IV)
Gothenburg, Sweden, June 19-22, 2016

Robust Road Marking Detection Using Convex Grouping Method in Around-View Monitoring System

Daejin Hyeon, Soomok Lee, Soonhong Jung, Seong-Woo Kim and Seung-Woo Seo

Abstract— **As the around-view monitoring (AVM) system becomes one of the essential components for advanced driver assistance systems (ADAS), many applications using AVM such as parking guidance system are actively being developed. As a key step for such applications, detecting road markings robustly is a very important issue to be solved. However, compared to the lane marking detection methods, detection of non-lane markings, such as text marks painted on the road, has been less studied so far. While some of methods for detecting non-lane markings exist, many of them are restricted to roadways only, or work poorly on AVM images. In this paper, we propose an algorithm which can robustly detect non-lane road markings on AVM images. We first propose a difference-of-Gaussian based method for extracting a connected component set, followed by a novel grouping method for grouping connected components based on convexity condition. For a classification task, we exploit the Random Forest classifier. We demonstrate the robustness and detection accuracy of our methods through various experiments by using the dataset collected from various environments.**

I. Introduction

An around-view monitoring system, also known as a surround-view monitoring system, is one that provides a view of surrounding environments of the ego-vehicle. It is based on the top-view image synthesized from multiple images concurrently captured from four or six different cameras mounted on the left, right, front and rear sides of the vehicle. The AVM system now becomes one of essential components for ADAS, and many applications using AVM such as parking guidance or vehicle localization have been proposed recently [1, 2]. As a cornerstone for such applications, detecting markings painted on the road robustly is very important because they not only give the most reliable and unchanged clue for localization, but also contain local traffic rules such as 'left only', 'one way' and 'stop'.

Such road markings, in general, can be categorized into two classes; lane marking and non-lane marking. Furthermore, non-lane markings can be divided into text and symbols like direction arrow or disable markings. While lane marking detection has been studied by many researchers so far [3], detection of non-lane markings has been less focused and just studied as an extension of lane marking detection [4], or only a very small set of non-lane markings such as arrows has been considered for a detection target [4–6]. Only recently, road marking detection including texts has started to be investigated [7–9].

Kheyrollahi and Breckon [7] proposed an algorithm that can recognize both texts and symbols in real time. In this method, multi threshold binarization is first applied to the inverse perspective mapping (IPM) in order to handle various lighting conditions on the road. Then a set of connected contours is extracted as candidate marking regions in the next steps. Each candidate region is classified using a neural network classifier with 23 classes into one of predefined dictionary words, and the components classified as an alphabet are then grouped together by finding a line aligning them. However, the actual usage of this method for more general cases is very limited because it requires a much larger number of classes to detect various words, markings and other non-alphabet texts.

Wu and Ranganathan [8] used a template matching based method for road marking detection and recognition. Maximally stable external regions [10] is used for extraction of regions of interest (ROI), and then FAST corner and HOG are selected as a keypoint and a descriptor. For each text word and symbol, one template is trained via a training image set. To detect every possible word that can be encountered on the road, this method needs to be trained with a training set containing every possible word and marking template. For this reason, it is only capable of detecting a relatively small set of road markings.

Greenhalgh and Mirmehdi [9] proposed a method that can detect and recognize symbols and text markings covering any arbitrary word. MSER is also applied to a bird-eye view image for connected component (CC) extraction. Any two CCs are grouped together if they are close to each other in horizontal direction and their rotated minimum area rectangle is similar. Afterward, the grouped CCs are treated as a text candidate; otherwise as symbol marking candidate. A HOG feature is used with the support vector machine (SVM) classifier for symbol recognition. For text recognition, the open source optical character recognition (OCR) engine, Tesseract, is applied. The limitation of this method is that the text marking encountered on the road should be aligned in horizontal direction. In cases when a vehicle is crossing a parking lot, however, each letter in a word is not always aligned with each other horizontally because the word can be seen in a distorted way.

In this paper, we propose an algorithm that can detect non-lane markings on around-view images regardless of scaling,

*D. Hyeon, S. Lee, S. Kim and S. Seo are with the Department of Electrical Engineering and Computer Science, Seoul National University and S. Jung is with LG Electronics, daejin.hyeon@snu.ac.kr, soomoklee@snu.ac.kr, sinabrlo@snu.ac.kr, sseo@snu.ac.kr and soonhong.jung@lge.com. This research was supported by LG electronics (No.0626-20150002), the National Research Foundation of Korea (NRF) grant funded by the Ministry of Science, ICT& Future Planning (MSIP) (No. 2009-0083495) and in part by the Institute of New Media and Communications and the Automation and Systems Research Institute, Seoul National University.

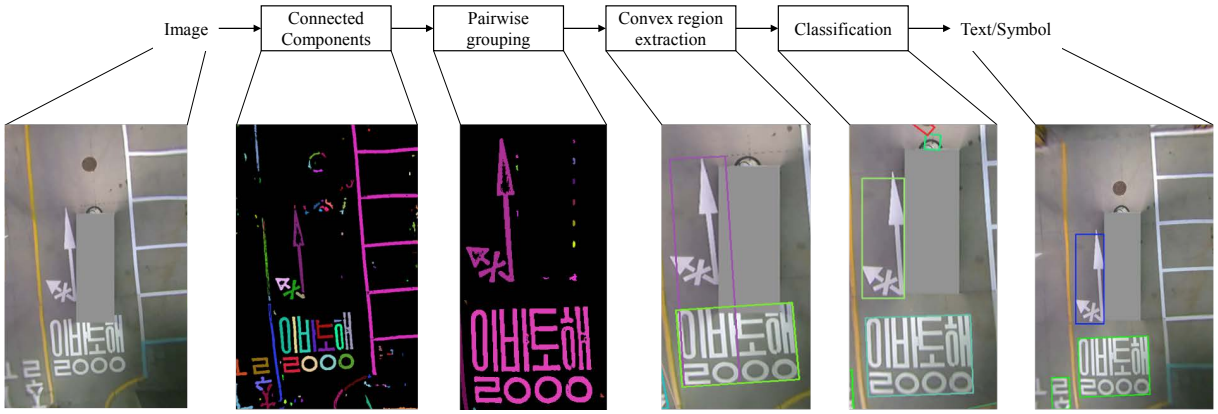

Fig. 1: The overall flowchart of the proposed algorithm.

rotation and even language type. Our algorithm consists of connected component extraction from around-view images, grouping of a connected component set and classification of each grouped connected components. The contributions of this paper can be summarized as the following. First, we propose a robust method for extracting CCs from AVM images which usually has a blurry characteristic. Second, the CC grouping method based on convex region is proposed. Unlike the existing grouping methods based on text line, our grouping method can perform well in more diverse scenarios including detection of rotated and non-alphabet texts.

The outline of this paper is as follows: In Section II, the overall flowchart of our algorithm is explained. In Section III, CC extraction method is explained in detail. In Section IV, convex region grouping method is introduced. In Section V, classification method including feature set and classifier is summarized. In Section VI, experimental results on our dataset are summarized, followed by the conclusion in Section VII. Since the focus of this paper is the robust detection of non-lane road markings, in rest of this paper, road markings refer to only non-lane markings for convenience.

II. Overview of the Algorithm

Most of the previous algorithms first transform an input image to a bird-eye view by IPM transform as a pre-process step. Since our algorithm takes a surround view image as an input, the first image in Fig. 1, the detailed step on IPM is skipped in this paper. However, our algorithm can also be applied to a front-view image if IPM transform is processed before hand. Fig. 1 shows the main flow of our algorithm. The proposed method first extracts a set of CCs followed by grouping of similar CCs. By splitting each initial group into a set of convex regions, the candidate regions for possible markings are determined. Then Random Forest classifier [11] decides the class each of candidate region. When it comes to text markings, while the previous methods try to recognize words and symbols in extracted ROI, the proposed algorithm only focuses on the accurate detection of text markings to separate the works of OCR because recognizing the meaning of text is solely the work of OCR field once the locations of text markings are decided. As the final output of the algorithm, locations of markings including text and symbols are decided using bounding boxes.

III. Candidate Region Extraction

A. Connected component extraction

For scene text localization task in Computer Vision, most of the existing algorithms choose MSER for extracting initial set of CCs. However, if an image is blurry or its contrast from background is low, the performance of MSER decreases. In case of a surround view image, regions near image boundary are happened to be blurred after IPM transform. To extract the initial connected component set robustly even with low contrast and blurry image, difference of Gaussians (DoG) based method is applied instead of MSER, which can be formulated as follows.

$$\begin{aligned} D_{\sigma,K\sigma}(x,y) &= I(x,y) * (G_{\sigma}(x,y) - G_{K\sigma}(x,y)), \\ \text{where } G_{K\sigma}(x,y) &= \frac{1}{2\pi K^2\sigma^2} e^{-(x^2+y^2)/(2K^2\sigma^2)}. \end{aligned} \tag{1}$$

I_b indicates a binary image obtained by threshold value, t_1, after applying the difference-of-gaussian, $D_{\sigma,K\sigma}$, on input image I.

$$I_b(x,y) := \begin{cases} 1, & \text{if } D_{\sigma,K\sigma}(x,y) \geq t_1 \\ 0, & \text{otherwise} \end{cases}. \tag{2}$$

A set of initial connected components, $\mathcal{U}$, is extracted by connecting neighboring pixels whose values are 1 in I_b. This set contains almost every connected component on painted road markings compared to MSER, however, it contains a lot of noises either. To remove such noises, top-hat filter [12] is applied afterward.

B. Properties on each connected component

An undirected graph $G = (\mathcal{V}, \mathcal{E})$ can be generated on sampled points in each component using Delaunay triangulation [13] in order to know spatial relationship between components and to access neighboring components efficiently. If there exists an edge $e_{p^i,q^j} \in \mathcal{E}$ between two vertices $v_i^p, v_j^q \in$

$\mathcal{V}$, where v_i^p indicates i-th vertex point in the component p, then its length, l_{p^i,q^j}, is updated in matrix $\mathcal{M} \in \mathbb{R}^{n \times n}$ by condition (3):

$$\mathcal{M}_{a,b} \leftarrow l_{p^i,q^j} \text{ if } l_{p^i,q^j} \leq \mathcal{M}_{a,b}, \tag{3}$$

where $n = |\mathcal{U}|, a = \max(p,q), b = \min(p,q)$ and each element in $\mathcal{M}$ is initialized by ∞ before hand. By generating graph on sampled points instead of mean or median point of each component, the number of neighbors is reduced to those who are only spatially neighbors and minimum distance between them can be calculated accurately regardless of shape of CC. A list of neighboring components is then stored in *u.neighbors*, where $u \in \mathcal{U}$. With matrix $\mathcal{M}$, this neighborhood list makes it easy to access and confirm neighbor component and its minimum distance efficiently.

The other properties of each connected component consist of color, rotated minimum area rectangle (RMAR) and mean strength value of edges in component calculated from (1). Since markings on the road are painted with prime colors, mostly white and yellow, reduced color is used instead of 24-bit color for efficiency. 3-bit RGB color, 1-bit for each channel, is assigned to each component according to (4):

$$\hat{C}_i := \begin{cases} 0, & \text{if } \frac{C_i}{C_R+C_G+C_B} \leq t_2 \\ 1, & \text{otherwise} \end{cases}, \tag{4}$$

where C_i and $\hat{C}_i$ indicate values of original and 3-bit color space on color channel i $\in \{R,G,B\}$, respectively. For each $u \in \mathcal{U}$, those values plus its index are stored in $u.color, u.rect$, $u.edge$ and $u.index$ respectively. The properties are then used as criteria for grouping.

IV. Convex region grouping

The purpose of grouping step is to group CCs, which come from a same marking, into one marking candidate. This step is often neglected when total marking set to be found consists of only symbols with one CC, but is very necessary if the goal is to find markings with multiple CCs like text word. For example, a word 'YIELD' is composed of five CCs. However, it is difficult to distinguish 'I' or 'L' from other lane markings because their properties such as colors and shapes are quite similar. Instead, it is easier to distinguish a word from noises and non-text markings if an exact text region is given.

Shi et al. [14] apply a graph cut method to find an exact text region, and other existing methods try to group components with a similar shape if their centroids lie on the same line [7, 9, 15–17]. Nonetheless, merely grouping CCs on the same text line often fails if the painted text is a non-alphabet word such as Chinese or Korean, in which each character is also composed of different disconnected strokes.

Our convex region grouping method is designed to work on even non-alphabet text under the simple assumption that text markings in a same word form a rectangular convex region. Similar to a merge and split manner, our method is divided into two stages: initial pairwise grouping and convex region extraction.

A. Initial pairwise grouping

First of all, at the initial pairwise grouping stage, any two neighboring components are grouped together if they lie on a same convex region and have similar properties. To inspect convexity of two components, binary image $\mathcal{B}$ with same size of input image I is first created by morphology closing [12] of an image whose pixels inside RMAR on each CC are filled with 1, with $k \times k$ structing element. Given $\mathcal{M}, \mathcal{B}$ and total CC set $\mathcal{U}$ initialized from Section III-B as inputs, $\mathcal{U}$ and $\mathcal{M}$ are updated by Algorithm 1.

Algorithm 1 Pairwise grouping

```
procedure PAIRWISE GROUPING(U, M, B)
    for all u ∈ U do
        list ← ∅
        for all v ∈ u.neighbors do
            i ← max(u.index, v.index)
            j ← min(u.index, v.index)
            if M_{j,i} is ∞ then
                M_{j,i} ← Convexity(u, v, B)
                if M_{j,i} > 0 and Similarity(u, v, M_{i,j}) ≤ t3 then
                    M_{j,i} ← 2
            if M_{j,i} is 2 then
                list ← list ∪ v
        u.neighbors ← list
function CONVEXITY(u, v, B)
    a ← u.rect.center
    b ← v.rect.center
    for all point (x, y) between a line segment ā b̄ do
        if B(x, y) < 1 then
            return 0
    return 1
function SIMILARITY(u, v, distance)
    w1 ← min(u.rect.width, u.rect.height)
    w2 ← min(v.rect.width, v.rect.height)
    similarity ← ∞
    if distance > γ × max(w1, w2) then
        similarity ← α‖u.color − v.color‖ + β‖u.edge − v.edge‖
    return similarity
```

Tolerance of convexity constraint can be controlled depending on convexity function. In this paper, centroid of each RMAR is used to check convexity for explanation, but four vertices of each RMAR can also be used if less tolerance is preferred. In a same way, similarity tolerance is also controlled by changing parameter values.

B. Convex region extraction

Each initial group is then further split into possible convex groups of maximum size. This problem is equivalent to find possible inscribed sub-convex regions of maximum size in a concave region, which can be reduced to Clique problem by assuming that each CC is a node and an edge exists if

convexity condition is preserved between two nodes. Since this is one of NP-Complete problems, it cannot be guaranteed to find a exact solution within polynomial time complexity. Nonetheless, it is possible to reduce the number of nodes to be checked for clique by finding pairs of nodes which share a common neighbor but disobey convexity with each others. By finding a set of maximum cliques on such seed nodes, possible convex region can be expanded starting from a clique to other nodes obeying convexity with the clique.

Algorithm 2 Convex region extraction

function CONVEX REGION EXTRACTION($\mathcal{U}, \mathcal{M}, \mathcal{B}$)
  $\mathcal{S} \leftarrow FindSeeds(\mathcal{U}, \mathcal{M}, \mathcal{B})$
  $\mathcal{G} \leftarrow \varnothing$
  for all $s \in \mathcal{S}$ **do**
    $\mathcal{G} \leftarrow \mathcal{G} \cup \{ExpandRegion(s, \mathcal{M}, \mathcal{B})\}$
  return $\mathcal{G}$
function FIND SEEDS($\mathcal{U}, \mathcal{M}, \mathcal{B}$)
  $seeds \leftarrow \varnothing$
  for all $u \in \mathcal{U}$ **do**
    for all $p, q \in u.neighbors$ **do**
      $i \leftarrow \max(p.index, q.index)$
      $j \leftarrow \min(p.index, q.index)$
      if $\mathcal{M}_{j,i}$ is ∞ **then**
        $\mathcal{M}_{j,i} \leftarrow Convexity(p, q, \mathcal{B})$
      if $\mathcal{M}_{j,i}$ is 0 **then**
        $seeds \leftarrow seeds \cup p \cup q$
        $\mathcal{M}_{j,i} \leftarrow -1$
  if *seeds* is empty **then**
    $seeds \leftarrow u \in \mathcal{U}$
  return *seeds*
function EXPAND REGION($s, \mathcal{M}, \mathcal{B}$)
  $\mathcal{C} \leftarrow \{s\}$
  $\mathcal{V} \leftarrow \{s\}$
  $queue \leftarrow queue \cup s.neighbors$
  while *queue* is empty **do**
    $v \leftarrow queue.dequeue$
    $i \leftarrow \max(s.index, v.index)$
    $j \leftarrow \min(s.index, v.index)$
    if $\mathcal{M}_{j,i}$ is ∞ **then**
      $\mathcal{M}_{j,i} \leftarrow Convexity(s, v, \mathcal{B})$
    if $\mathcal{M}_{j,i} > 0$ **then**
      $queue \leftarrow (queue \cup v.neighbors) - \mathcal{V}$
      $\mathcal{C} \leftarrow \mathcal{C} \cup v$
    $\mathcal{V} \leftarrow \{v\}$
  return $\mathcal{C}$

However, finding a set of maximum cliques step can be skipped for computation complexity if each node is located sparsely in a relatively simple concave region. Because distance between nodes is long and each node contains a number of points, it is less likely to be grouped as a convex set for two distanced nodes if they are not in a same convex region. Instead, as a heuristic way, expanding from a single seed node can be an alternative way treating each seed node as a clique set with one element. The heuristic method is explained in Algorithm 2.

As a final step for grouping, the probability that a group $\mathcal{X}$ is rectangular is approximated as :

$$Pr(\mathcal{X}) = \frac{\#\ of\ all\ points\ in\ \mathcal{X}}{\sum_{x \in \mathcal{X}} area\ of\ x.rect}. \tag{5}$$

Then one rectangle region can be extracted by choosing the group having highest probability.

$$\underset{g \in \mathcal{G}}{\operatorname{argmax}}(Pr(g)). \tag{6}$$

Once the extracted group is removed from $\mathcal{M}, \mathcal{B}$ and $\mathcal{G}$, other groups can be further detected by repeating the procedures.

V. CLASSIFICATION

A. Feature selection

After candidate marking regions are extracted as a binary image by grouping in previous section, the remaining issue is to detect good features for classification. In this problem, one of the conditions for good feature set is that they should be invariant on rotation. Also, they should be adequate for describing the general shape of markings since there is no fixed shape on text marking. Taking these conditions into account, we selected the following features for classification which are mostly introduced in [14, 18, 19].

- **Area** of the total number of pixels in CCs.
- **Perimeter ratio** between RMAR and CCs.
- **Euler characteristic** which is topological invariant value describing shape.
- **Horizontal crossings** which is the average number of transitions from 0 to 1 or 1 to 0 in horizontal direction.
- **Vertical crossings** which is same as horizontal crossing but in vertical direction.
- **Area hole ratio** which is the ratio between the area of pixels and that of region holes
- **Convex hull ratio** which is the ratio between the area of pixels and that of convex hull.
- **Regularity** which is 2-dimensional vector describing degree of regularity defined as below:

$$\begin{aligned} Regularity(1) &= \frac{\#\ of\ pixels\ in\ skeleton}{\#\ of\ pixels\ in\ contour}, \\ Regularity(2) &= \frac{\#\ of\ pixels}{(\#\ of\ pixels\ in\ contour)^2}. \end{aligned} \tag{7}$$

- **Uniformity** which is 2-dimensional vector describing uniformity of stroke width defined as below.

$$\begin{aligned} Uniformity(1) &= \frac{\#\ of\ elements\ in\ \mathcal{N}_p}{\#\ of\ pixels\ in\ contour}, \\ Uniformity(2) &= \frac{average\ stroke\ width}{\min(w, h)}, \end{aligned} \tag{8}$$

  where w, h are width and height of RMAR on candidate marking, and $\mathcal{N}_p$ is a set of pairwise edges whose gradient direction are opposite while connected by a line.
- **Occupation** calculated from (5).

TABLE I: Comparison of classifier.

Accuracy / Classifier	Text		Symbol	
	Precision	Recall	Precision	Recall
Random Forest	0.964	0.827	0.964	0.7
k-NN	0.856	0.828	0.739	0.755
MLP	0.933	0.82	0.83	0.712
Logistic regression	0.882	0.745	0.89	0.627

- **Hu moments** which is 7-dimensional vector describing shape invariant on rotation.

B. Classifier selection

A number of classifies were compared to choose the one best suit for our feature set. 9028 samples were used to train and test each classifier using a 10-fold cross-validation method. The portions of text, symbol and non-marking in sample data are set to reflect actual portions of them among candidate marking sets found in various scenes with markings. Table I shows top 4 classifiers whose precision/recall rates on both text and symbol markings are greater than 0.7. Random forest is selected for classifier in our problem since it outperformed others especially on precision rates of both text and symbol markings according to Table I.

VI. Experiment results

To test the performance of our algorithm under various scenarios, we collected the video outputs of AVM captured on various locations under different weather and road conditions with two different vehicles. From each collected video, afterward, we extracted image frames containing markings. For each parameter value used in our experiment, once its value is fixed, the same value was kept for entire test phase. Experimentally, setting threshold values to $t_1 = 1$, $t_2 = 70$, $t_3 = 10$, similarity parameter values to $\alpha = 10$, $\beta = 1$, $\gamma = 1$ and structing element size $k = 15$ produced the best result on 470×720 size image. Our algorithm is implemented using C++/OpenCV library and runs on a single thread in a PC with Intel 2.2GHz CPU and 8GB RAM.

Total computation time on the entire steps varied depending on the scenarios, but it took about 80ms in most of the cases: 50ms for binarization and filtering, 12ms for CC extraction, 6ms for CC grouping, 12ms for feature extraction and classification. Most of the running time is spent on binarization and filtering because of heavy morphology operations. Nevertheless, our method can run in real-time.

The images at the first row in Fig 2 show some of challenging scenarios which can be summarized as followings: apartment gate (disarranged directions), indoor parking lot (flare), road after snow (snow), outdoor parking lot (skid mark), bus stop near campus (multi language type), road at night (light). In our experiment under those difficult cases, our algorithm was able to detect most of the markings. It indicates that our algorithm works robustly in various conditions. However, it also fails to detect exact text region when a character from a word is disjointed into too many pieces, or text is overlapped by reflection light. In such cases, one-to-one correspondence relationship between detected marking set and groundtruth marking set is not established, but one-to-many or many-to-one.

TABLE II: Overall accuracy of the proposed algorithm.

	Precision	Recall	hmean
Score by [20]	0.6601	0.5517	0.6011
ICDAR 2003	0.6674	0.5073	0.5764

The evaluation method proposed by Wolf and Jolion [20], which is used ICDAR 2015 Robust Reading Competition, was applied in our experiment result to measure the overall performance of our algorithm correctly. Since this method is designed to consider one-to-many and many-to-one correspondence cases, the output of our algorithm is directly given as an input for evaluation. The evaluation result is shown in Table II with the result from ICDAR 2003 evaluation method for comparison.

VII. Conclusion

In this paper, a novel road marking detection algorithm for around-view image is introduced. To handle the issues regarding synthesized AVM image and non-alphabet text markings, a new method for CC detection and grouping is developed. Our algorithm is suitable for ADAS system since it is robust and runs in real-time, which are crucial requirements for in-vehicle applications. This method can also be applied to find text regions in natural images if appropriate parameter setting is followed. As a next step for this algorithm, applying OCR on each detected text region can be possible if the goal is to extracting the meaning of it. For localization purpose, on the other hand, classifying into two or three classes such as text/symbols/non-markings or markings/non-markings can give enough information for calculating ego motion of vehicle. we plan to further develop our algorithm in various applications including text and vehicle localization.

References

[1] T. Wu and A. Ranganathan, "Vehicle localization using road markings," in *Intelligent Vehicles Symposium (IV), 2013 IEEE*. IEEE, 2013, pp. 1185–1190.

[2] M. Yu and G. Ma, "360 surround view system with parking guidance," *SAE International Journal of Commercial Vehicles*, vol. 7, no. 2014-01-0157, pp. 19–24, 2014.

[3] T. Veit, J.-P. Tarel, P. Nicolle, and P. Charbonnier, "Evaluation of road marking feature extraction," in *Intelligent Transportation Systems, 2008. ITSC 2008. 11th International IEEE Conference on*. IEEE, 2008, pp. 174–181.

[4] S. Vacek, C. Schimmel, and R. Dillmann, "Road-marking analysis for autonomous vehicle guidance." in *EMCR*, 2007.

[5] J. Rebut, A. Bensrhair, and G. Toulminet, "Image segmentation and pattern recognition for road marking

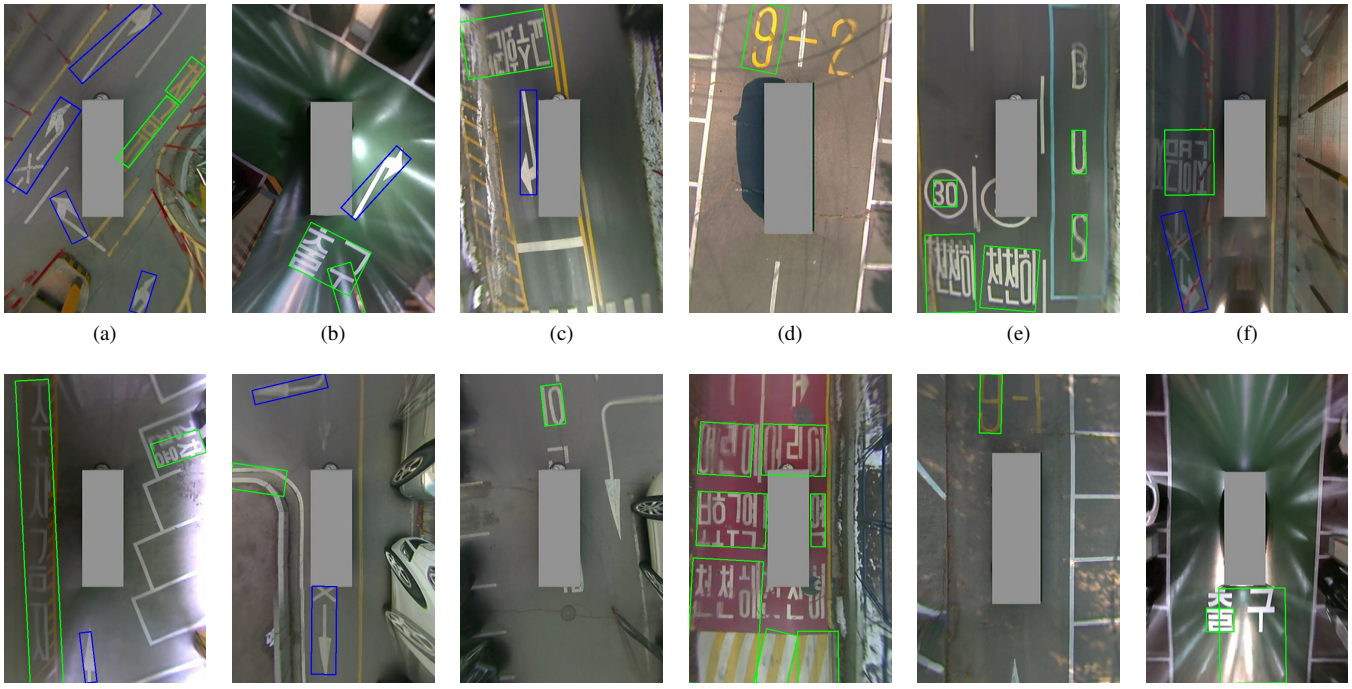

Fig. 2: Classification results on AVM images in our dataset. The green rectangles indicate regions classified as text marking while blue ones indicate regions classified as arrow marking.

analysis," in *Industrial Electronics, 2004 IEEE International Symposium on*, vol. 1. IEEE, 2004, pp. 727–732.

[6] Y. Li, K. He, and P. Jia, "Road markers recognition based on shape information," in *Intelligent Vehicles Symposium, 2007 IEEE*. IEEE, 2007, pp. 117–122.

[7] A. Kheyrollahi and T. P. Breckon, "Automatic real-time road marking recognition using a feature driven approach," *Machine Vision and Applications*, vol. 23, no. 1, pp. 123–133, 2012.

[8] T. Wu and A. Ranganathan, "A practical system for road marking detection and recognition," in *Intelligent Vehicles Symposium (IV), 2012 IEEE*. IEEE, 2012, pp. 25–30.

[9] J. Greenhalgh and M. Mirmehdi, "Detection and recognition of painted road surface markings."

[10] J. Matas, O. Chum, M. Urban, and T. Pajdla, "Robust wide-baseline stereo from maximally stable extremal regions," *Image and vision computing*, vol. 22, no. 10, pp. 761–767, 2004.

[11] L. Breiman, "Random forests," *Machine learning*, vol. 45, no. 1, pp. 5–32, 2001.

[12] E. R. Dougherty, "An introduction to morphological image processing," *Tutorial texts in optical engineering*, 1992.

[13] D.-T. Lee and B. J. Schachter, "Two algorithms for constructing a delaunay triangulation," *International Journal of Computer & Information Sciences*, vol. 9, no. 3, pp. 219–242, 1980.

[14] C. Shi, C. Wang, B. Xiao, Y. Zhang, and S. Gao, "Scene text detection using graph model built upon maximally stable extremal regions," *Pattern Recognition Letters*, vol. 34, no. 2, pp. 107–116, 2013.

[15] B. Epshtein, E. Ofek, and Y. Wexler, "Detecting text in natural scenes with stroke width transform," in *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*. IEEE, 2010, pp. 2963–2970.

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

[17] H. I. Koo and D. H. Kim, "Scene text detection via connected component clustering and nontext filtering," *Image Processing, IEEE Transactions on*, vol. 22, no. 6, pp. 2296–2305, 2013.

[18] M.-K. Hu, "Visual pattern recognition by moment invariants," *Information Theory, IRE Transactions on*, vol. 8, no. 2, pp. 179–187, 1962.

[19] L. Neumann and J. Matas, "Real-time scene text localization and recognition," in *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*. IEEE, 2012, pp. 3538–3545.

[20] C. Wolf and J.-M. Jolion, "Object count/area graphs for the evaluation of object detection and segmentation algorithms," *International Journal on Document Analysis and Recognition*, vol. 8, no. 4, pp. 280–296, 2006.