A Practical System for Road Marking Detection and Recognition

Tao Wu and Ananth Ranganathan

Abstract— **We present a system for detecting and recognizing road markings from video input obtained from an in-car camera. Our system learns feature-based templates of road markings from training data and matches these templates to detected features in the test images during runtime. We use MSER features and perform the template matching in an efficient manner so that our system can detect multiple road markings in a single image. Our system also scales well with the number of categories of road markings to be detected. For evaluating our system, we present an extensive dataset (available from `www.ananth.in/RoadMarkingDetection.html`) of road markings with ground truth labels, which we hope will be useful as a benchmark dataset for future researchers in this area.**

I. Introduction

Road markings refer to the signs drawn on the surface of the road. These differ from traffic signs erected by the side or on top of roads. Road markings are as important to detect as traffic signs for navigation systems and driver assistive devices. While many road markings provide information that is redundant to traffic signs (eg. speed limits), other information, such as arrows for turn-only lanes, are exclusively provided by road markings.

However, while there is extensive work in computer vision focused on detection of traffic signs [8], [7], [20], [9], the corresponding literature for road marking detection is rather sparse. Indeed, even large databases of road markings for evaluation are hard to come by, with a single exception [22], which however does not contain labels for the types of road markings.

In this paper, we present an efficient system for detecting road markings. We also present a large database of road markings, captured using an in-car camera, on which we test our system. Our road marking detection system is based on learning feature-based templates of the markings using training images. In particular, our choice of features for defining the templates are FAST corner features [19]. During runtime, these templates are matched using a two-step process of first selecting promising feature matches and subsequently performing a structural matching to account for the shape of the markings. This detection method seamlessly handles complex road markings with multiple disconnected components such as the railroad marking shown in Figure 3. Further it is efficient enough to run in real time without the use of special hardware such as GPUs, and scales well with the number of types of road markings to be detected.

Our road marking dataset contains almost all the commonly found markings on US roads. Further, the dataset consists of videos captured from an in-car camera, rather than stand alone images as in the ROMA dataset [22]. This addresses the common usage scenario on vehicles, where we usually deal with video, and allows a stricter evaluation of the detection algorithms. We envision our database and detection system forming a benchmark for future research on road marking detection.

In the rest of the paper, we first discuss related work followed by a description of our road marking detection algorithm. Subsequently, we provide details on our dataset and experiments for validating our detection system.

II. Related Work

Since lane marking detection has been pursued for a number of years, the same techniques have been attempted for road marking detection. An overview of lane detection techniques - such as the use of edges, regions, and tracking for continuity - can be found in [16]. For instance, Vacek et al. [21] use detection of lane markers to guide road marking detection. However, their method, which uses the horizontal extent of markings at various distances to classify them, requires a lot of hand-coding for each type of road marking and does not scale well to a large number of marking types. Another work that uses lane markers to detect markings such as arrows is [5]. However, this method is too slow to use in real-time. Methods which rely on lane marking detections cannot work when such markings are absent. Hence, we detect road markings directly without relying on lane markings.

The above methods depend on the detection of road markings before classifying them, and to this purpose, Veit et al. [22] provide a comparison of different methods for detecting road markings. They also provide an extensive dataset that contains road marking groundtruth. However, this dataset is not suitable for our purposes since it does not contain the type of road marking in the ground truth. Rebut et al. [18] use Fourier descriptors [10] and a k-NN classifier to label road markings. A threshold-based segmentation approach is used by Burrow et al. [4] to detect the presence of road markings automatically but not their type. Li et al. [14] use a binarized bitmap to represent the shape of the road marking and use shape matching to detect and classify road markings. The above methods only work for single component markings such as arrows and linear markings. Detecting road markings with multiple disconnected components, especially when they are surrounded by other markings on the road, is a more challenging task which we address in this work.

Tao Wu is with the Department of Electrical & Computer Engineering, University of Maryland, College Park, MD, 20742, USA `taowu@umiacs.umd.edu`

Ananth Ranganathan is with Honda Research Institute, 425 National Ave, Mountain View, CA, 94043, USA `aranganathan@honda-ri.com`

Noda et al. [17] compute an eigenspace on a template by perturbing an ideal template under different illumination, scale, blur, and motion blur conditions. Classification is performed using eigenspace analysis. However, this method is reliant on the proper cropping and centering of the road marking. In our system, we obtain some invariance to illuminance, motion blur etc. through the use of stable features for detecting candidate regions for road markings. In particular, we use MSERs [15], which have been shown to have a reasonable degree of invariance to these perturbations.

A method based on shape matching [13] is closest to our system in terms of applicability as it operates in real-time and can detect and classify a large number of complex road markings. However, we claim that our system is much simpler than the one presented therein, which uses a two-stage neural network classifier for which the number of hidden nodes and their connectivity has been carefully chosen by hand for the experiments.

III. Method

We now provide details on our road marking recognition system. As mentioned before, since a road marking could consist of several disconnected parts, the "segment and recognize" strategy does not work robustly in this scenario. Hence, we propose a method that can detect and recognize the road signs simultaneously.

Our system consists of training and testing phases. The input to the training phase is a set of images with groundtruth masks and labels of the road markings. We will call these training images as template images henceforth since these are used as templates for the road marking detection.

For each template image, we first perform rectification to compensate the lens and perspective distortions. Within the regions of interest containing the road markings, a set of points of interest (POI) are detected, for which we use FAST corner detectors [19]. A Histogram of Oriented Gradients (HOG) [6] feature vector is extracted for each POI and the template set is built using the locations and the feature vectors of the POIs extracted from all template images for the particular type of road marking.

During runtime, the same steps are repeated for each frame of the testing video and a set of POIs and their feature vectors are extracted, except that the regions of interest are detected automatically. Subsequently, we find multiple matching candidates for each POI in each template image. Lastly, a structural matching algorithm is employed to test if a subset of the matched POI pairs forms a road sign the same as the ones in the template images.

Details of each of these steps is provided below. A flow-diagram of our method is shown in Figure 1.

A. Image Rectification

The camera we use is mounted on a roof rack and points to the front of the vehicle. Due to this low viewpoint, there is a significant perspective distortion with distance. We rectify the image using an inverse perspective transform that significantly reduces this distortion. Inverse perspective transforms have also been used extensively in past work on road marking detection [16], [4].

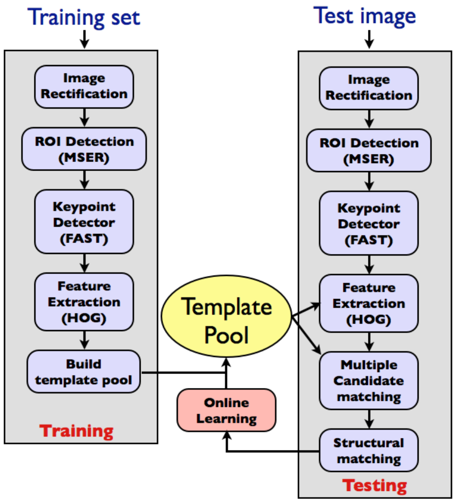


Fig. 1. Workflow of our road marking detection system.

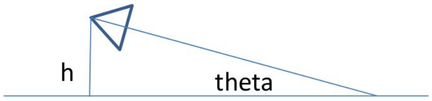


Fig. 2. The setting of the camera.

The inverse perspective transform is a matrix which only depends on the camera calibration, the height h of the camera above the ground, and the viewing angle of the camera θ with respect to the ground (Figure 2) [2]. Applying the matrix transforms an input image to a birds-eye view. An example of such a birds-eye view is shown in Figure 3. Note that the transform parameters are calibrated before hand assuming the ground is flat, rather than calibrated in realtime. Hence the lanes may be not perfect parallel in the birds-eye view when the vehicle is on hills or has pitch and roll movements. Obtaining the birds-eye view allows us to directly compute the gradient on this image to obtain the HOG descriptors. In the absence of the birds-eye view, we would have been forced to use an anisotropic gradient operator to account for the perspective distortion.

B. Region of Interest Detection

Our region of interest (ROI) detector helps prune the image to portions which are good candidates for being road markings. Since a traffic scene image, especially in urban areas, will contain many objects such as vehicles, people, trees etc., which are not of interest to us, removing most of these areas where road signs are not likely to occur improves the efficiency of our system considerably.

We base our detection of interesting regions on the observation that road signs are always brighter than their immediate surroundings since they are designed to be seen easily. To this end, we employ Maximally Stable Extremal Regions (MSERs) [15] to detect regions of interest, which are putative road markings in our case. An MSER is connected region that is extremal in the sense that the pixels inside the

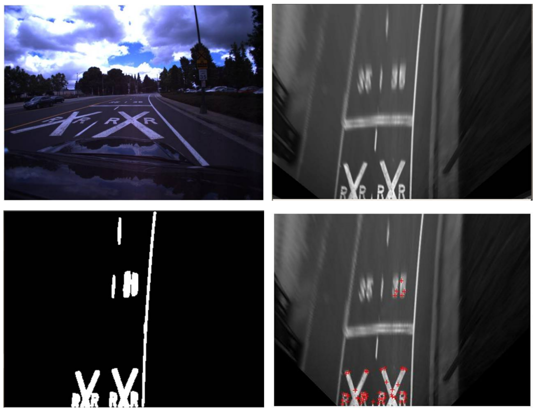


Fig. 3. Top left: the original image; top right: the rectified image (birds-eye view); bottom left: Detected regions of interest (MSER features); bottom right: POIs detected by the FAST corner detector (in red).

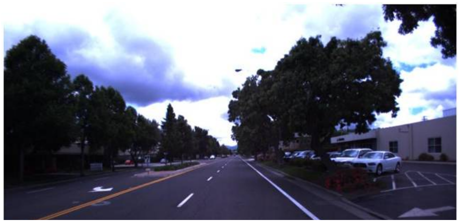


Fig. 4. An example of a rotated road sign appearing in the opposite lane. These markings should not be detected without orientation information.

regions are always brighter or darker than the pixels on the boundary. Efficient algorithms for computing MSERs exist.

We detect MSERs on the rectified road scene images. Almost all regions containing road markings are detected, in addition to some regions in other parts of the scene. However, these spurious detections do not pose a problem to our algorithm as they are pruned by the later steps. Examples of the MSER results are shown in Figure 3(bottom).

MSERs are stable across illumination and viewpoint changes. In the literature, viewpoint invariant matching is achieved by fitting ellipses to the MSER regions, transforming the ellipse regions to circular regions, and extracting local feature vectors [15]. However, in our case, the resulting rotation invariance is undesirable as it would lead to detection of road markings in the opposite side of the road also. An example is shown in Figure 4, where there is a left turn sign in the opposite lane. Hence, we do not transform the MSERs into elliptical regions.

C. Point of Interest Detection

We extract a set of feature points from the regions of interest computed as explained above. To enable real-time computation without the use of GPUs, we use the FAST corner detector proposed by Rosten et al. [19]. FAST has been reported to be about 23 times faster than the Harris corner detector, and 39 times faster than SIFT. Repeatability of feature detection is reported better than, and at worst comparable to, the SIFT key points detector. We apply the FAST corner detector on the regions of interest detected on the rectified images. An example is shown at the bottom of Figure 3.

D. Feature Extraction and Template Pool

A feature descriptor is extracted from the neighborhood of each detected POI. We compute the histogram of oriented gradients (HOG) descriptor [6] for each POI. The HOG descriptor consists of a 128 dimensional feature vector computed using certain "optimal" scales and orientations of the image patch around a POI. Determining these optimal scales and orientations is a time-consuming computational task. Considering the speed requirement, we extract the HOG feature vectors at 3 fixed scales and 1 fixed orientation. For each scale, a 128 dimensional feature is extracted for each POI. By concatenating the features extracted at different scales, we obtain the final 384-dimensional feature vector for each POI.

For all the template images, the ROI containing the road signs is obtained from the labeled groundtruth. After rectification, POIs are detected within the labeled ROIs of all template images. Then feature vectors are extracted for all the POIs. All of the feature vectors and the coordinates of the corresponding POIs are stored as the template pool.

IV. Runtime Template Matching

We now proceed to the runtime matching algorithms that recognize the road markings. The steps mentioned in the previous sections viz. rectification, MSER detection, FAST corner detection, and HOG descriptor computation are performed on each test image. The signs in the testing images are then detected and identified based on the POIs.

In a road scene image, there may be multiple different road signs. Moreover, a single type of road marking may appear at multiple locations in an image. Thus, each template may have multiple matched patches in a test image. Additionally, since the POI detector does not have 100% repeatability, some of the POIs detected in the test image may not be detected in the template image and vice versa. Hence, there is usually a subset of POIs in the test image that match a subset of the POIs in the template.

Our algorithm to compute this matching consists of two steps. First, we find putative matching pairs of POIs based on their feature vectors, and second, we refine the result through a structural matching algorithm that matches the 2D geometry of the POIs within the road marking. These steps are explained in further detail below.

A. Multiple Candidates Feature Matching

Let N and M denote the number of POIs in two images, a test image and a template image respectively. The Euclidean distance matrix $D_{N\times M}$ between the two sets of POIs using their feature vectors is calculated. The entry at the ith row and the jth column of D, $d_{i,j}$, is the distance between the features of the ith POI in the testing image and the jth POI in the template.

As is commonly done in the literature, we use the ratio between the smallest distance and the second smallest distance to choose possible matching points. We use this criterion to divide the POIs in the template to a possibly matched subset S_{match} and a possibly non-matched subset

$S_{nonmatch}$. The distances to the possibly matched POIs are usually smaller than the distances to the non-matched POIs. Due to efficiency considerations, we use a simple criterion to determine S_{match} and assume that:

$$\min_{j \in S_{nonmatch}} d_{i,j} / \max_{j \in S_{match}} d_{i,j} > \alpha \tag{1}$$

where α is a user-defined threshold.

Let $D'_i = \left[d_{i,J_1}, \cdots, d_{i,J_j}, \cdots, d_{i,J_M}\right]$ denote the vector of the ith row of D sorted in ascending order. The ratios $R = [r_1, \cdots, r_j, \cdots, r_{M-1}]$ between the adjacent entries in D'_i are calculated as $r_j = d'_{i,J_{j+1}} / d'_{i,J_j}$. If r_{J_K} is the first entry in r_j that satisfy Eqn.1, the J_1th to J_Kth POIs in the first image are the matching candidates to the ith POI in the second image.

Due to the use of a threshold, our matching algorithm is not symmetric. Hence, for greater robustness we first match the test image against the template image, and subsequently match the template image against the test image similarly. A pair of features (A, B), where A is a POI in the test image and B is a POI in the template, is counted as a matching edge if A is a matching candidate for B and B is a matching candidate for A. All the matching edges and the coordinates of the candidate POIs are sent to the structural matching algorithm to determine if any subset of the POIs in the test image has the same geometric layout (shape) as in the template. We now proceed to explaining this next step.

B. Structural Matching

Given the matching edges between two set of POIs, the next goal is to determine which subset of the matching edges is actually the road marking in the template image. Our structural matching algorithm solves this problem recursively. The idea is that for a subset of the matching edges, the POIs involved in the test image and the template form two shapes respectively. We calculate a matching cost between the two shapes to decide if they have the same geometric structure. The diagram of our method is shown in Figure 5.

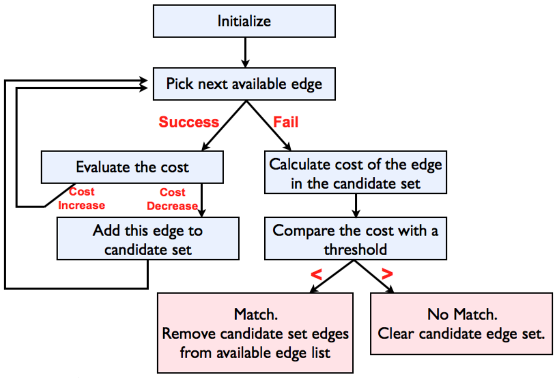

Initialize
Pick next available edge
Success
Fail
Evaluate the cost
Cost Increase
Cost Decrease
Add this edge to candidate set
Calculate cost of the edge in the candidate set
Compare the cost with a threshold
<
>
Match. Remove candidate set edges from available edge list
No Match. Clear candidate edge set.

Fig. 5. Flowchart for the structural matching algorithm.

The algorithm is initialized with a list of all available edges and an empty set of candidate edges. An available edge is added to the candidate edge set if the matching cost of the new candidate edge set (with the available edge added) is lower than the current candidate set. Regardless of the outcome of this test, the edge is marked unavailable so that it is not picked up again.

When no more edges are available, the matching cost of the candidate set is calculated. If the cost is smaller than a threshold β, the edges in the candidate set form a matched road sign with the same label as the template. These edges and the corresponding POIs are then removed from further consideration in the algorithm. If the cost is larger than the β, the candidate edge set is cleared and only the first edge in the candidate set is removed from further consideration.

The above procedure is repeated for each template until no edges remain in consideration. This matching algorithm is performed for each template, the result being the positions of all the detected regions of each template in the test image.

The only thing missing in the above description is a description of the cost function we use to compute the costs. We now address this issue.

C. Cost Function

Given the coordinates of the POIs, we adopt the Procrustes analysis [12] to measure the distance between the two shapes formed by sets of POIs. Procrustes analysis consists of centeroid alignment, scaling and rotational normalization. The position of the center of the shape is the mean of the x and y coordinates of all the POIs:

$$\bar{x} = \frac{1}{K}\sum_{k=1}^{K} x_k, \ \bar{y} = \frac{1}{K}\sum_{k=1}^{K} y_k \tag{2}$$

Thus, the centers of both shapes can be aligned by simply subtracting the means from the coordinates:

$$S' = S - \mathbf{1}\begin{bmatrix}\bar{x} & \bar{y}\end{bmatrix} \tag{3}$$

where $\mathbf{1}$ is a K dimensional column vector with each element equals to 1. After the center of the shapes are aligned and moved to the origin, the scale factor s of a shape is defined by the square root of the central moment:

$$s = \sqrt{\sum_{k=1}^{K} x_k^2 + \sum_{k=1}^{K} y_k^2} \tag{4}$$

Then the size of each shape can be normalized by dividing the coordinates of all the contained points by the scale factor $x'_k = x_k/s, \ y'_k = y_k/s$. Consider a pair of shapes. Assume one shape S_1 is fixed, then for the other shape S_2, the difference angle θ between the orientation of the two shapes can be solved by minimizing the distance between the two shapes after S_2 is rotated by θ. The coordinates of the ith landmark in S_2 after rotated by angle θ are $\begin{bmatrix}x'_{2,i} & y'_{2,i}\end{bmatrix} = \begin{bmatrix}x_{2,i} & y_{2,i}\end{bmatrix} A$ where $A = \begin{bmatrix}\cos\theta & \sin\theta \\ -\sin\theta & \cos\theta\end{bmatrix}$ is a rotation matrix. The solution of this minimization is given by the singular value decomposition (SVD) [11]. Let the SVD result of $S_1^T S_2$ be $U\Lambda V^T = S_1^T S_2$. Then the rotation matrix A is given by VU^T , and the rotated shape of S_2 is obtained by $S'_2 = S_2 V U^T$. The cost function used in our experiment is defined by $dist = 1 - \sum_i \lambda_i^2$ where λ_i is the ith entry in the diagonal of Λ. Hence, the more similar the two shapes are, the lower the value of the cost function.

D. Computational Complexity

The feature matching component of our algorithm is quadratic in the number of features. However, since there are typically only a few tens of features per template, the runtime is almost completely determined by the number of features detected on the test image, which is typically in the hundreds. If the feature vectors are clustered, using say K-means, it is possible to have sub-linear feature matching complexity at the cost of some accuracy.

The structural matching component of our algorithm has linear complexity with the number of the template images. As we show in our experiments, the system runs in realtime ($>$10 frames per second) with a few tens of template images.

In practical scenarios, when GPS and map data are available, the types of road markings to be expected can be determined given the current location of the vehicle. Since this would usually be a small subset of all possible types of road markings, there is usually only a limited number of templates that need to be used for the matching. Hence, we are confident of the ability of our algorithm to meet the requirements of real applications.

V. Dataset

To the best of our knowledge, currently there is no dataset that is designed for evaluating the performance of road marking detection and recognition. Hence we collected our data using a PointGrey Grasshopper camera, with a 3.6mm focal length lens, mounted on a roof rack of a vehicle and facing forwards. The camera was calibrated using the Camera Calibration Toolbox [3] and the calibration information is available as part of the dataset. The vehicle was driven on urban and suburban roads in California, USA under various weather, lighting, and road conditions, including extremely challenging ones such as twilight and night time. The videos of the road scenes were recorded at 800x600 resolution. We collected 29 videos, which contain a total of 28614 frames. We manually annotated a subset of the road markings appearing in the videos, which still amount to a huge dataset of 1208 annotated road markings. Since video annotation is a time-consuming task, we only annotated one in every 5-10 frames. We hope that this extensive dataset will provide a fruitful benchmark for other researchers working on this problem. An example of an annotated template image is shown in Figure 6. The dataset can be downloaded from `www.ananth.in/RoadMarkingDataset.html`.

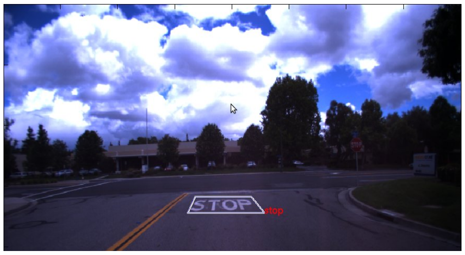


Fig. 6. An example of an annotated template image.

VI. Experiments

We tested our algorithm on the dataset described above. Our algorithm is implemented in C++ and uses OpenMP for parallelization in the feature extraction stage. A value of 1.3 was used for the α threshold from (1), while the value for the β parameter from Section IV-B was set to 0.01.

In our experiment, we select 22 images as the templates, which contain ten different types of road signs including speed limit signs ”35” and ”40”, ”left turn”, ”right turn” and ”forward” arrows, and ”bike”, ”stop”, ”ped”, ”xing” and ”rail” signs. Our algorithm achieves a true positive rate of 90.1% and a false positive rate of 0.9%, indicating that false positive detections occur only very rarely. We found that our algorithm could also detect and recognize road signs with partial occlusion, motion blur or dirt on them. While some robustness to blur is obtained through the use of MSER and HOG for ROI and POI descriptor computation, our algorithm can also handle occlusion because the template is feature based and can be matched to a test image even if a few features are hidden due to occlusion. A few example results are shown in Figure 7.

Since the dataset consists of video, commonly used image-based performance statistics are not appropriate, especially since not every frame of the video is annotated. Hence, we consider it to be a true detection of a road marking if the marking is detected during the time slot that it appears in the video. We define the true positive rate to be the number of true positives divided by the number of annotated distinct road signs. A detected sign is counted as a false positive if no road sign of that label occurs in the previous 5 frames, which corresponds to a time period of 0.3 seconds before the detected frame. The false positive rate is correspondingly defined as the number of false positives divided by the number of annotated frames.

The confusion matrix of all the detected road sign results computed as above is summarized in Table 8. The entry at the ith row and the jth column gives the percentage that the ith label is recognized as being the jth. The label ”other” means that the detected region is a false positive. It shows that the performance is high except for the case of the forward arrow. Since the shape is very simple in this cases, the structural matching algorithm produces a number of spurious results. Note that since there is currently no common, openly available dataset on which previous researchers have tested their algorithms, it is not possible to compare our results to theirs. In fact, by making our dataset public, we intend to address this very issue.

VII. Discussion

We have described a system for detecting and recognizing road markings that can effectively deal with complex road signs with disconnected parts. Our method uses a feature-based template to represent the road markings. We use FAST corners for the template features and perform structural matching in an efficient manner to detect multiple road signs per image in realtime. The system also has a good ability to deal with limited occlusions and generalizes well to novel

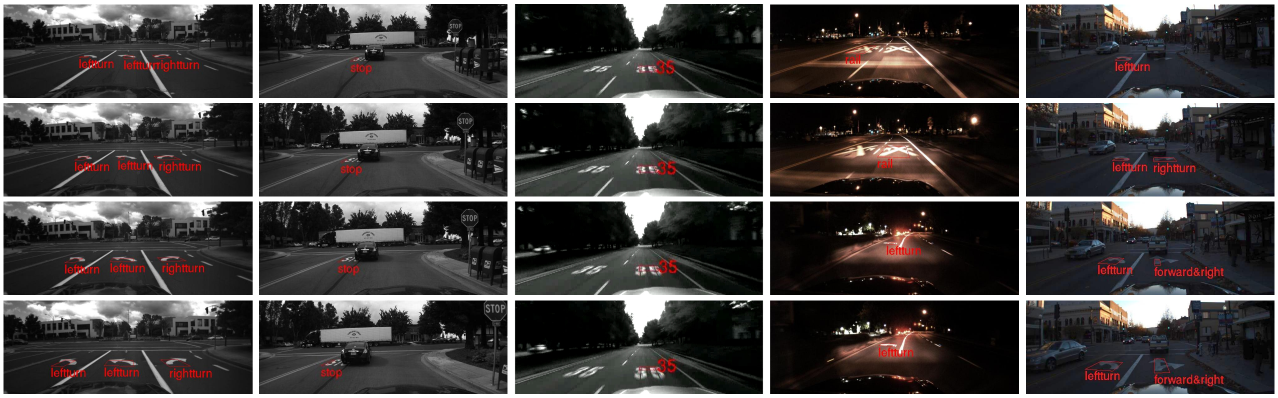
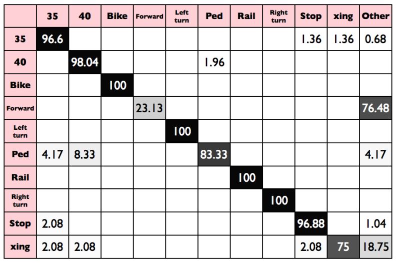


Fig. 7. Examples of road sign detection and recognition results (labeled in red) under different condition: multiple signs in cloudy weather(1st column), with occlusion in cloudy weather (2nd column), with motion blur in sunny weather (3rd column), during night (4th column), and in twilight (5th column). Night and twilight results are shown in color just to illustrate the lighting conditions better.



	35	40	Bike	Forward	Left turn	Ped	Rail	Right turn	Stop	xing	Other
35	96.6								1.36	1.36	0.68
40		98.04				1.96					
Bike			100								
Forward				23.13							76.48
Left turn					100						
Ped	4.17	8.33				83.33					4.17
Rail							100				
Right turn								100			
Stop	2.08								96.88		1.04
xing	2.08	2.08							2.08	75	18.75

Fig. 8. The recognition confusion matrix of all the detected road signs.

road and lighting conditions as we have demonstrated in our experiments. We have also presented an extensive dataset containing groundtruth locations and labels of road markings. The dataset consists of video obtained from a calibrated camera and is publicly available.

Our system currently depends on two manually selected threshold values, the α parameter of (1) and the β parameter of section IV-B. It is future work to automatically select these parameter values and adapt them online for different test conditions. We currently observe that the values used in our experiments presented above are applicable in a wide range of scenarios including large variations in lighting or road conditions, such as highly overcast days or muddy roads, such as shown in Figure 7.

In very bright conditions, our algorithm is sensitive to shadows which cause large differences in contrast. Shadow removal techniques [1] may also be useful in this regard. Our system works robustly for complex road markings but the false positive rate is higher for simpler signs such as forward arrows. Finding a tighter cost function that alleviates this problem is part of future work.

References

[1] J. M. Alvarez and A. M. Lopez. Road detection based on illuminant invariance. *IEEE Trans. Intelligent Transportation Systems*, August 2010.

[2] M. Bertozzi, A. Broggi, and A. Fascioli. Stereo inverse perspective mapping: theory and applications. *Image and Vision Computing*, 8(16):585–590, 1998.

[3] J.-Y. Bouguet. Camera calibration toolbox for matlab.

[4] M. P. N. Burrow, H. T. Evdorides, and M. S. Snaith. Segmentation algorithms for road marking digital image analysis. *Proceedings of the Institution of Civil Engineers, Transport*, 156(1):17–28, 2003.

[5] P. Charbonnier, F. Diebolt, Y. Guillard, and F. Peyret. Road marking recognition using image processing. In *IEEE Conference on Intelligent Transportation Systems*, pages 912–917, 1997.

[6] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 886–893, 2005.

[7] A. de la Escalera, J. M. Armingol, and M. Mata. Traffic sign recognition and analysis for intelligent vehicles. *Image and Vision Computing*, 21(3):247–258, 2003.

[8] S. Estable, J. Schick, F. Stein, R. Janssen, R. Ott, W. Ritter, and Y.-J. Zheng. A real-time traffic sign recognition system. In *Proceedings of the Intelligent Vehicles '94 Symposium*, pages 213–218, 1994.

[9] H. Fleyeh, M. Shi, and H. Wu. Support vector machines for traffic signs recognition. In *International Joint Conference on Neural Networks*, 2008.

[10] H. Ishiguro, K. C. Ng, R. Capella, and M. M. Trivedi. Omnidirectional image-based modeling: three approaches to approximated plenoptic representations. *Machine Vision and Applications*, 14(2):94–102, 2003.

[11] D. G. Kendall. Shape manifolds, procrustean metrics, and complex projective spaces. *Bull. London Math. Soc.*, 16(2):81–121, March 1984.

[12] D. G. Kendall. A survey of the statistical theory of shape. *Statistical Science*, 4(2):87–99, 1989.

[13] A. Kheyrollahi and T. P. Breckon. Automatic real-time road marking recognition using a feature driven approach. *Machine Vision and Applications*, pages 1–11, 2010.

[14] Y. Li, K. He, and P. Jia. Road markers recognition based on shape information. In *IEEE Intelligent Vehicles Symposium*, pages 117–122, 2007.

[15] J. Matas, O. Chum, M. Urban, and T. Pajdla. Robust wide baseline stereo from maximally stable extremal regions. In *Proceedings of the British Machine Vision Conference*, pages 414–431, 2002.

[16] J. C. McCall and M. M. Trivedi. Video based lane estimation and tracking for driver assistance: Survey, system, and evaluation. *IEEE Transactions on Intelligent Transportation Systems*, 7(1):20–37, 2006.

[17] M. Noda, T. Takahashi, D. Deguchi, I. Ide, H. Murase, Y. Kojima, and T. Naito. Recognition of road markings from in-vehicle camera images by a generative learning method. In *IAPR Conference on Machine Vision Applications*, 2009.

[18] J. Rebut, A. Benshair, and G. Toulminet. Image segmentation and pattern recognition for road marking analysis. In *IEEE International Symposium on Industrial Electronics*, pages 727–732, 2004.

[19] E. Rosten, R. Porter, and T. Drummond. Faster and better: A machine learning approach to corner detection. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 32:105–119, 2010.

[20] A. Ruta, Y. M. Li, and X. H. Liu. Towards real-time traffic sign recognition by class-specific discriminative features. In *Proceedings of the British Machine Vision Conference*, 2007.

[21] S. Vacek, C. Schimmel, and R. Dillmann. Road-marking analysis for autonomous vehicle guidance. In *Proceedings of the 3rd European Conference on Mobile Robots*, 2007.

[22] T. Veit, J.-P. Tarel, P. Nicolle, and P. Charbonnier. Evaluation of road marking feature extraction. In *Proceedings of the International IEEE Conference on Intelligent Transportation Systems*, 2008.