

# Kernel-based Tracking for Improving Sign Detection Performance

JongHo Lee<sup>1</sup>, Young-Woo Seo<sup>2</sup>, and David Wettergreen<sup>2</sup>

**Abstract**—To be deployed in the real-world, automatic and semi-automatic systems should understand traffic rules by recognizing and comprehending contents of traffic signs, because traffic signs inform what driving behaviors should be. In this paper, we present the successful application of methods to improve the traffic sign localization performance. Given a potential sign region, our algorithm represents both the detected sign as a target and candidates in the subsequent frame as probability density functions. Then, our algorithm maximizes the similarity between a target and candidates to localize the sign. Finally, the maximum similarity among candidates is assigned as a tracked sign. The experimental results verify that our algorithm can robustly localize traffic signs in images under various weather conditions and driving scenarios.

## I. INTRODUCTION

Traffic signs instruct drivers what their driving behaviors should be. For example, a stop sign tells a driver to stop at that location. Similarly, a workzone sign warns drivers of road work happening ahead. For safe and reliable driving, being able to understand traffic signs is critical. One way researchers are trying to ensure that daily driving is safer and more reliable by offering automatic and semi-automatic transportation, that is, by developing advanced driver assistance systems (ADAS) and autonomous vehicles. These automated systems, to be deployed in the real-world, should be capable of 1) recognizing and understanding traffic rules and 2) executing them accordingly. Thus, to successfully deploy ADAS and autonomous vehicles in the real-world, one of their essential components is an automatic traffic sign recognition system.

Since traffic signs are designed for human drivers, computer vision techniques are appropriate for traffic sign recognition system. Indeed, to understand the semantics of a traffic sign, its color, symbol, and text should be analyzed. Even though the locations and appearances of traffic signs are highly constrained by governmental regulations [15], it is still challenging to successfully locate signs from images; after all the same sign appears differently on assorted images due to image acquisition conditions. For example, the line of sight between signs and a vision sensor relative to the position of the sun can change the shape and the color of the same sign. Furthermore, uncontrolled illumination conditions, including those caused by weather, directly affect the digital image formation, giving the same sign a different appearance.

<sup>1</sup>JongHo Lee is with Mechanical Engineering, Carnegie Mellon University, 5000 Forbes Ave., Pittsburgh, PA, 15213, USA. [jonghole@andrew.cmu.edu](mailto:jonghole@andrew.cmu.edu)

<sup>2</sup>Young-Woo Seo and David Wettergreen are with the Robotics Institute, Carnegie Mellon University, 5000 Forbes Ave., Pittsburgh, PA 15213, USA. [{ywseo,dsw}@ri.cmu.edu](mailto:{ywseo,dsw}@ri.cmu.edu)

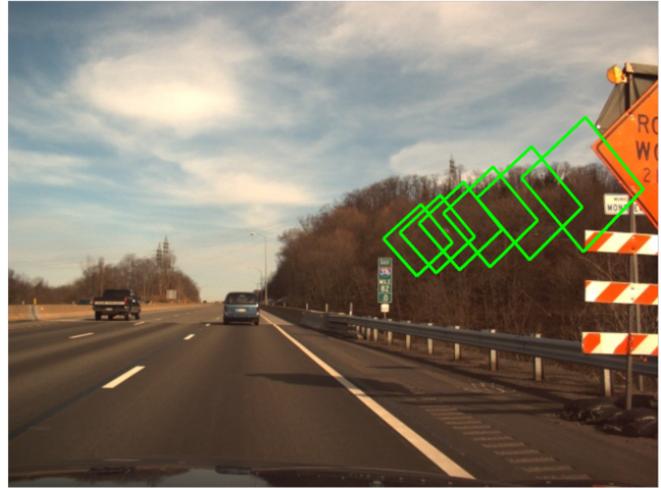


Fig. 1: Projection of the sequence of sign locations onto one image. Our approach exploits two facts of a sign's appearance in a video stream. The sign image sub-regions from two consecutive image frames overlap each other. There are small variations of their appearances and locations.

To effectively tackle variations in appearance of traffic sign, we developed a color-based sign detection system to recognize workzones on highways [13]. In particular, our system learns variations of a color in sign images so as to perform a pixel-wise binary color classification. Then, our algorithm identifies blobs to localize sign image regions. Finally, our algorithm represents a cropped sign image in a homogeneous feature space in order to reduce variation of geometric shapes. To handle potential sign recognition errors, we exploit temporal redundancy of sign appearances in such a way that sign-recognition confidence is forwarded to smooth out the effect of any signs missed. Because we propagate only the confidence, not the appearance information about the detected signs, the performance of our detection method was degraded.

To improve the performance of our color-based sign detection method, we apply an appearance-based, sign tracking algorithm. In particular, our approach exploits two features of a sign's appearance in a video stream, in which a sign appears multiple times before it disappears from the view point. First, in the two consecutive image frames, there are overlapping regions, particularly in the images of the sign. Second, there are small variations of the appearances and locations. Fig. 1 illustrates this observation. By considering these features, the detected sign and its rectangular region are first given as an input to our tracking system. Then, the detected sign image

is modeled with a probability density distribution as a target. Next, the rectangular region from the input is projected onto the subsequent frame as a candidate, which is also modeled with a probability density distribution. Lastly, our approach iteratively shifts the rectangular region until the distance between the target and the candidate becomes minimum.

In this paper, we describe a successful application of color-classification techniques and kernel-based tracking techniques to localize different signs from perspective images and a complete system for traffic sign recognition.

## II. RELATED WORK

Many researchers have proposed traffic sign recognition systems. The essential step is detection, and currently two dominant approaches exist: based on the color or on shape. The color-based sign detections, including ours, choose one color space and find the appropriate ranges of target color. Several researchers in the past elected to manually find the target color ranges [5], [9], which is simple but prone to error, while the others use machine learning techniques to obtain the optimal ranges of target color [10]. The shape-based sign detections usually utilize either the gradient of gray scale image [11] or trained model from sign shape data sets [1].

Some of the traffic sign recognition systems combine a detection process with a tracking process in order to improve performance. Among those systems, we divide them into two different approaches: one deals with the dynamics of the tracked signs, and the other copes with the appearance of the tracked signs. The dynamics approach for tracking is to utilize the discrete-time dynamics of vehicles, such as with Bayesian Filters [6], [8], [10], [11], [14] and information fusion [1] to predict the position of a sign in subsequent frames based on image frames received previously. This can reduce the computational cost, while still only relying on the detection performances, if the update model is defined by only the detection algorithm. The vehicle dynamics can also be calculated in predicting the state model with additional sensors, such as inertial measurement unit (IMU). Our tracking system can track signs without a motion prediction model.

The appearance approach for tracking is to utilize the visual appearance [9], such as the well-established Lucas-Kanade Tracker (LKT) [2]. Nonetheless, the computational cost of LKT sometimes shows its limitation for real-time operation, especially when it is tracking a large object. To overcome this limitation, Liu *et al.* [9] chose several interesting points and focused on those points. These improvements allow their algorithm to run in real-time, though there are problems when images are blurred due to vehicle motion.

We have created a sign tracking system using kernel-based object tracking [3], which minimizes the distance between the probability density functions of a target and a candidate, to improve a sign detection performance. Our system not only can track signs without regard to the complex vehicle dynamics, but also incorporates target representation based

on the appearance from an image to effectively handle the color variation through a sequence of images.

## III. COLOR-BASED SIGN DETECTION AND KERNEL-BASED SIGN TRACKING

The goal of this work is to improve the sign detection performance to reliably localize different types of signs in image sequences. In what follows, we briefly describe our sign detection algorithm and give a detailed description of our sign tracking system. We use similar notations to [3], with modifications appropriately in section III-B.

### A. Color-Based Sign Detection

The appearances of signs are strictly regulated by local and national governments. Even with regulations, there are issues commonly arising in color variations through images, which cause to misidentification of pixel color. We formulate learning of specific color variation, which is the target color we aim to detect, as a binary color classification using the Bayesian inference framework. AdaBoost [7] is used to learn the likelihood function of a given pixel as part of the target sign. The training data for each color is comprised of a set of images, some from the web, while the others from our video data. The data collected is utilized to train a set of weak-learners and their weights.

$$P(\mathbf{X}|\text{sign}) = \text{mode}(\cup_j g(f(\mathbf{x}_j|\text{sign})))$$

$$f(\mathbf{x}_j|\text{sign}) = \sum_H^{i=1} \alpha_i h_i(\mathbf{x}_j)$$

where  $\mathbf{X}$  is a set of 2-dimensional color vectors comprised of hue and saturation from pixels in an image,  $\mathbf{x}_j \in \mathbf{X}$ . More details are explained in [13].

### B. Kernel-Based Sign Tracking

The output of our sign detector, which contains the center position,  $\mathbf{z}$ , size,  $\mathbf{s}$ , and a *mask* of shape of the detected sign, is given as input for our sign tracking system. Our task is to localize the detected sign in the subsequent frame based on the appearance of the target. In order to localize it, we need to characterize the appearance of the target as probability density function (PDF). We choose hue-saturation-value (HSV) color space as feature space because it is less affected by illumination when compared to red-green-blue (RGB) color space. In particular, we choose only hue and saturation values from HSV to represent a specific color, which helps to decrease the dimension of color vector. The normalized hue and saturation values are quantized into  $n_h \times n_s$  bins, where  $n_h$  and  $n_s$  are the bin numbers of hue and saturation, respectively:  $\mathbb{H}$  and  $\mathbb{V}$  are the histograms of hue and saturation values, where  $\mathbb{H} = \{h_j\}_{j=1 \dots n_h}$  and  $\mathbb{V} = \{v_k\}_{k=1 \dots n_s}$ .  $n_h$  and  $n_s$  are directly related to the performance of tracking. If the numbers of bins are small or large, the target color will be under- or over-fitted to track. To represent the PDF consistently with various sizes of signs, the pixel coordinates should be normalized. Furthermore, it is obvious that the target color of the detected sign should be

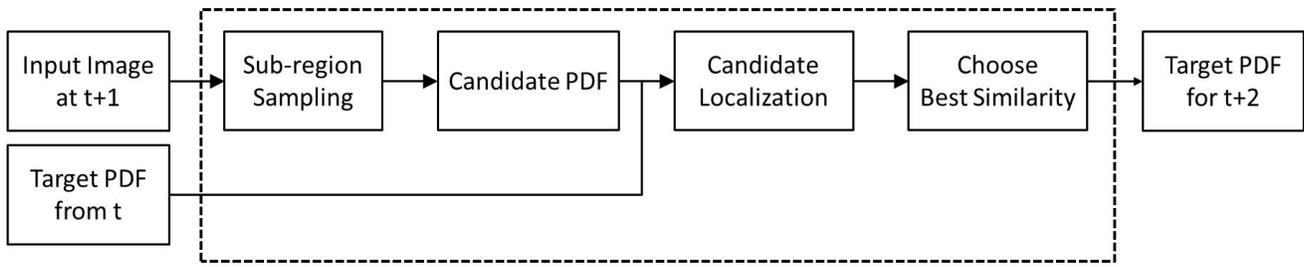


Fig. 2: A schematic overview of the sign tracking system. The image at time  $t+1$  and a target PDF from the detection result at time  $t$  are given as inputs. Different sizes of sub-regions of a subsequent frame are represented as candidate PDFs and localized by the mean-shift algorithm. A candidate with the maximum similarity to a target is an output as a new target for the subsequent frame.

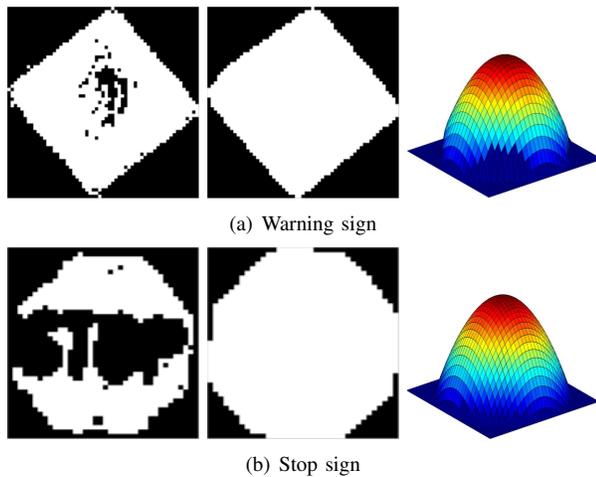


Fig. 3: Kernel functions. The left image is the output of the detected sign, and the middle image is a mask of shape. In order to reduce the background effect and concentrate on the target color, Epanechnikov kernel and the *mask* of shape from the detection output are combined. The right image illustrates this combined kernel.

more involved to the histogram than the background because it can dramatically be changed in a sequence due to image projection. In order to sample densely on the target color, Epanechnikov kernel,  $K$ , giving larger weights to pixel coordinates closer to the center, is combined to a *mask* of shape of the detected sign:  $K(\mathbf{p}) = c_k (1 - \mathbf{p}) \cdot \text{mask}$ . Fig. 3 illustrates this kernel. Finally, we can calculate the target PDF,  $\mathbb{T} = \{t_b\}_{b=1 \dots n_h \times n_s}$ , on normalized pixel coordinates,  $\mathbf{p}$ , as

$$\begin{aligned}
 t_b &= t_{n_s \cdot (j-1) + k} \\
 &= C \sum_{i=1}^n K \left( \|\mathbf{p}_i\|^2 \right) H(\mathbf{p}_i, h_j, v_k)
 \end{aligned}$$

where

$$H(\mathbf{p}_i, h_j, v_k) = \begin{cases} 1 & \text{hue}(\mathbf{p}_i) \in h_j \cap \text{sat}(\mathbf{p}_i) \in v_k \\ 0 & \text{otherwise} \end{cases}$$

and  $C$  is a normalization constant.

Once the target PDF is created, we calculate candidate PDFs,  $\mathbf{c}(\mathbf{z})$ , in the subsequent frame to localize, where  $\mathbf{z}$  is the new center. Calculating candidates is the same as the target except since the location of the sign in the subsequent frame shifts, the new normalized pixel coordinates,  $\mathbf{p}_i^{\text{new}}$ , based on  $\mathbf{z}$  should be calculated. Also, the size of the bounding box increases as the vehicle gets closer to the sign. We can estimate the size of sign in the subsequent frame, but it is unnecessary to exactly estimate size. To increase our performance, we apply five different sizes of kernels (0%, 2%, 5%, 7%, and 10% increased kernels) and choose one as the tracking result.  $\mathbf{c}(\mathbf{z})$  is calculated by

$$\begin{aligned}
 c_b &= c_{n_s \cdot (j-1) + k} \\
 &= C_s \sum_{i=1}^n K \left( \left\| \frac{\mathbf{p}_i^{\text{new}} - \mathbf{z}}{\mathbf{s}} \right\|^2 \right) H(\mathbf{p}_i^{\text{new}}, h_j, v_k)
 \end{aligned}$$

where  $C_s$  is a normalization constant.

To localize candidate, we need to maximize the similarity between these two discrete PDFs to track the target in the subsequent frame. We utilize Bhattacharyya coefficient,  $B(\mathbf{t}, \mathbf{c}(\mathbf{z}))$ , to measure the similarity, where  $B(\mathbf{t}, \mathbf{c}) = \sum_{b=1}^{n_h \times n_s} \sqrt{t_b \cdot c_b}$ . Using linear approximation around  $\mathbf{z}$ , we are now able to use the mean-shift algorithm to find the mode of

$$\frac{C_s}{2} \sum_{i=1}^n w_i K \left( \left\| \frac{\mathbf{z} - \mathbf{p}_i^{\text{new}}}{\mathbf{s}} \right\|^2 \right) \quad (1)$$

where  $w_i$  is

$$w_i = \sum_{b=1}^{n_h \times n_s} \sqrt{\frac{t_b}{c_b(\mathbf{z})}} H(\mathbf{p}_i^{\text{new}}, h_j, v_k) \quad (2)$$

We can find the maximum value of eq (1) by the gradient. If Epanechnikov kernel is used, the gradient of eq (1) will be a weighted summation as the standard mean-shift algorithm. In our case, though, the mean-shift algorithm is represented as

$$\mathbf{z}^{\text{new}} = \frac{\sum_{i=1}^n \mathbf{p}_i^{\text{new}} w_i \cdot \text{mask}}{\sum_{i=1}^n w_i \cdot \text{mask}} \quad (3)$$

which is still a dot product that has low computational cost. This is iterated until  $\epsilon < t_\epsilon$ , where  $\epsilon = \|\mathbf{z}^{\text{new}} - \mathbf{z}\|$ . If  $\epsilon$  is not

converged, eq. (2) and (3) are repeated with the new center;  $\mathbf{z} = \mathbf{z}^{\text{new}}$ . Also, we set the maximum number of iteration,  $N_{max}$ , to avoid an infinite loop. However, the number of iterations is usually less than it. Once all of the candidates converge, we choose the final output which has the highest Bhattacharyya coefficient.

#### IV. EXPERIMENTATION

This section details the experimentation in measuring the performance of our sign detection and tracking system using images acquired under various weather conditions and driving scenarios.

##### A. Experimental Setup

For our experimentation, we choose three different US traffic signs to detect and track: “stop”, “pedestrian crossing” in school area, and “workzone” signs due to their unique characteristics. All of these signs have distinctive color with standardized shapes as shown in table I. There are no signs which have the same characteristics as these, and we don’t need to individually classify them. Also, these signs perform an important role on improving the relationship between drivers, pedestrians, and road construction workers, particularly in suburban areas. Thus, three different target color models are trained accordingly for the detection algorithm, and applied appropriately based on the test data.

Video sequences were collected at 480 by 640 resolutions at 15 fps under various environmental conditions. Each of these videos was decomposed into sequences of images. Seven different image streaming of each sign (“stop”, “pedestrian crossing”, and “workzone”) were prepared for the test data, and the remaining images were used to train the color classifier. For each of the test data, sequential images were given as an input.

For comparison, two different types of localization algorithms were executed: ‘detection only’ and ‘detection and tracking’. ‘Detection only’ executed the algorithm described in section III-A in every input image. ‘Detection and tracking’ type executed the detection algorithm at first, and once the output from the detection satisfied the predefined conditions, the tracking algorithm was executed in the remaining frames. We empirically found that the tracking algorithm performed best when the size of the bounding box was bigger than 32 pixels, and the aspect ratio of bounding box was between two predefined thresholds. To represent the target and candidates, we set both  $n_h$  and  $n_s$  as 20, which resulted in a 400-bin histogram.

To evaluate the performance, we used the metrics used for PASCAL object detection challenges [12]. An output

TABLE I: Three US traffic signs are chosen as target signs for our experimentation. Each sign has its unique appearance [15].

type	shape	sign plate color
stop	octagon	red
pedestrian crossing	diamond	fluorescent yellow-green
workzone	diamond and rectangle	fluorescent orange

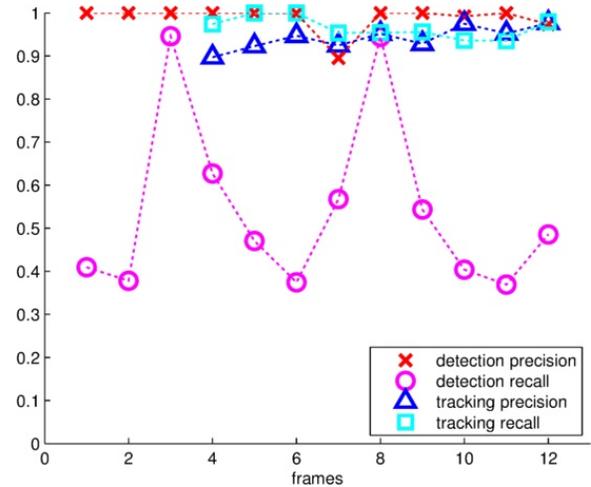


Fig. 5: Localization performance result of a “stop” sign sequence in Fig. 4.

bounding box,  $o_i$ , was considered a potential match to the ground truth bounding box,  $g_i$ , in a given image frame,  $i$ , if their area of overlap was greater than a predefined value,  $\tau < \frac{Area(o_i \cap g_i)}{Area(o_i \cup g_i)}$ . When a potential match was found in a given image, sign localization performance could be further analyzed by measuring the following:  $precision = \frac{Area(o_i \cap g_i)}{Area(o_i)}$  and  $recall = \frac{Area(o_i \cap g_i)}{Area(g_i)}$ .

##### B. Experimental Results

To depict our experimental results, we detail one of them qualitatively as well as quantitatively, and summarize the whole performance. Fig. 4 illustrates one of the experimental results. The detection process is executed until the size and aspect ratio of the bounding box satisfies the predefined thresholds ((a) to (c)). Then, the target model is calculated within the bounding box for the subsequent frame. The candidate which has the maximum similarity to the target is chosen as in III-B and it updates the target model for the subsequent frame. Once a successful detection occurs, the tracking system provides consistent bounding boxes through the whole sequence even when our detector only partially detects the sign. This is because our sign detector applied a pixel-wise binary classification to every pixel in ROI and identifies blobs using connected-component grouping. While doing this, our sign detector sometimes treats these blobs as two different parts (top and bottom individually) and picks the bigger part as a potential sign. Still our tracker provides a fitted bounding box due to the kernel. Once the sign is detected, the kernel contains a mask of the sign and finds the most similar appearance of the target in the subsequent frame. This provides stable tracking.

We detail the performance measurement of this sequence in Fig. 5, where the  $x$ -axis represents the number of image frames organized by the time, and  $y$ -axis represents the values of precision and recall; a red cross is the precision, and a magenta circle is the recall. A cyan rectangle and blue triangle represent a precision and recall of tracking respectively. As shown, the precision of detection are equal



Fig. 4: Localization of detection and tracking in a sequence. The yellow dotted rectangle is groundtruth bounding box labeled manually. The blue solid rectangle represents the output bounding box from tracking (starting at (d)), and the red dashed rectangle from detection alone. The inset at the bottom left in each frame is magnified from each respective frame.

TABLE II: Localization performance of 3 different signs from 7 image sequences of each sign. Each cell in the table shows the mean and standard deviation.

(a) workzone sign		
	Precision	Recall
Detection and tracking	<b>0.979</b> / 0.013	<b>0.910</b> / 0.033
Detection only	0.950 / 0.034	0.775 / 0.105
(b) Pedestrian crossing sign		
	Precision	Recall
Detection and tracking	<b>0.959</b> / 0.037	<b>0.916</b> / 0.051
Detection only	0.911 / 0.069	0.896 / 0.070
(c) Stop sign		
	Precision	Recall
Detection and tracking	<b>0.954</b> / 0.018	<b>0.963</b> / 0.013
Detection only	0.947 / 0.055	0.774 / 0.221

or higher than tracking in the whole sequence except frame 7. Our sign detector, as mentioned above, sometimes provides the bounding box from either the top half or the bottom half of the stop sign. Since the bounding boxes locate the inner-portion of the “stop” sign, they have high precision. However, they result in low recall because they only cover a portion of the sign. In contrast, our sign tracking system consistently covers the sign similarly in every frame.

We calculate the performance of each test data separately and average individual measurements over testing sequences to summarize the overall performance of each traffic sign in table II. The performances after tracking starts are included to evaluate. The second row of each sub table represents the precision and recall of ‘tracking’ while the third row represents those of ‘detection’. As shown, our tracking system improves both the precision and recall of three different traffic signs. Especially, the recall rate of ‘stop’ sign is higher than those of ‘workzone’ and ‘pedestrian crossing’ signs due to their shapes and kernel. We use a combination of a *mask* of shape of the detected sign and Epanechnikov as our kernel. A *mask*, however, sometimes doesn’t perfectly match to its actual sign. Also, the kernel helps to create candidate PDFs by weighting more from the center. This results in a smaller bounding box than its actual size. Since diamond shape signs have more background than octagon shape signs, recall values of diamond shape signs are lower than those of octagon shape signs. Not only do the precision and recall show better performance when tracking is included, but also the standard deviations show less variation. Thus, we can infer that including kernel-based tracking system provides more robust and stable sign localization performance. We can conclude that our sign tracking system covers at least 91% of an actual sign when at least 95% of the bounding of

box of our sign tracking system contains it.

## V. CONCLUSIONS AND FUTURE WORK

We presented a new road sign tracking method to improve the performance of our sign detection system for three-traffic signs. The selection of these signs is based on high priority due to safety reasons. The experimental results demonstrated that our tracking method can improve performance in sign localization. The contribution of this paper included the successful application of kernel-based sign tracking to accurately locate the difference between several different signs.

We showed promising results, but there is still room for improvement. Our color-based detector can be improved by including the shape information. We also had applied Histogram of Oriented Gradients (HOG) [4] method to utilize the shape information, but dropped it due to its expensive computational cost. The image pyramid should be generated for the scale invariant detection and each shape needs its own individual model. Since we already showed that the tracking system can localize robustly, applying HOG method without the image pyramid on ROI from our color-based sign detector for the detection and tracking will drastically reduce the computation cost. This work is underway along with continuing testing under changing road conditions.

## ACKNOWLEDGMENT

This work was supported by the Carnegie Mellon University Autonomous Driving Collaborative Research Laboratory (AD-CRL).

## REFERENCES

- [1] Claus Bahlmann, Ying Zhu, Visvanathan Ramesh, Martin Pellkofer, and Thorsten Koehler, A system for traffic sign detection, tracking, and recognition using color, shape, motion information, In *Proceedings of IEEE Symposiums on Intelligent Vehicles (IV-05)*, pp. 250-260, 2005.
- [2] Simon Baker and Iain Matthews, Lucas-Kanade 20 years on: a unifying framework, *International Journal of Computer Vision (IJCV-04)*, 56(3): 221-255, 2004.
- [3] Dorin Comaniciu, Visvanathan Ramesh, and Peter Meer, Kernel-based object tracking, *IEEE Transactions on Pattern Recognition and Machine Intelligence (PAMI-03)*, 25(5): 564-577, 2003.
- [4] Navneet Dalal and Bill Triggs, Histograms of oriented gradients for human detection, In *Proceedings of IEEE Computer Vision and Pattern Recognition*, pp. 886-893, 2005.
- [5] Marcin L. Eichner and Toby P. Breckon, Integrated speed limit detection and recognition from real-time video, In *Proceedings of IEEE Symposiums on Intelligent Vehicles (IV-08)*, pp. 626-631, 2008.
- [6] Chiung-Yao Fang, Sei-Wang Chen, and Chiou-Shann Fuh, Road-sign detection and tracking, *IEEE Transactions on Vehicular Technology*, 52(4): 1329-1341, 2003.
- [7] Yoav Freund and Robert E. Schapire, Experiments with a new boosting algorithm, In *Proceedings of International Conference on Machine Learning (ICML-96)*, pp. 148-156, 1996.
- [8] Miguel Ángel García-Garrido, Miguel Ángel Sotelo, and Ernesto Martín-Gorostiza, Fast traffic sign detection and recognition under changing lighting conditions, In *Proceedings of IEEE Intelligent Transportation Systems (ITS-06)*, pp. 811-816, 2006.
- [9] Wei Liu, Xue Chen, Bobo Duan, Hui Dong, Pengyu Fu, Huai Yuan, and Hong Zhao, A system for road sign detection, recognition and tracking based on multi-cues hybrid, In *Proceedings of IEEE Symposiums on Intelligent Vehicles (IV-09)*, pp. 562-567, 2009.
- [10] Luis David Lopez and Olac Fuentes, Color-based road sign detection and tracking, In *Proceedings of International Conference on Image Analysis and Recognition (ICIAR-07)*, pp. 1138-1147, 2007.
- [11] Mirko Meuter, Anton Kummert and Stefan Müller-Schneiders, 3D traffic sign tracking using a particle filter, In *Proceedings of IEEE Conference on Intelligent Transportation Systems (ITSC-08)*, pp. 168-173, 2008.
- [12] J. Ponce, T. Berg, M. Everingham, D. Forsyth, M. Hebert, S. Lazebnik, M. Marszalek, C. Schmid, C. Russell, A. Torralba, C. Williams, J. Zhang, and A. Zisserman, Dataset issues on object recognition, In *Towards Category-Level Object Recognition*, pp. 29-48, 2006.
- [13] Young-Woo Seo, David Wettergreen, and Wende Zhang, Recognizing temporary changes on highways for reliable autonomous driving, In *Proceedings of IEEE International Conference on Systems, Man, and Cybernetics (SMC-12)*, pp. 3021-3026, 2012.
- [14] George Siogkas and Evangelos Dermatas, Detection, tracking and classification of road signs in adverse conditions, In *Proceedings of IEEE Mediterranean Electrotechnical Conference (MELECON-06)*, pp. 537-540, 2006.
- [15] U.S. Department of Transportation, Federal highway administration, *Manual on uniform traffic control devices for streets and highways*, <http://mutcd.fhwa.dot.gov/>, 2009 Edition.