The 2010 IEEE/RSJ International Conference on
Intelligent Robots and Systems
October 18-22, 2010, Taipei, Taiwan

Intelligent Road Sign Detection Using 3D Scene Geometry

Jeffrey Schlosser, Michael Montemerlo, and Kenneth Salisbury

Abstract— **This paper proposes a new framework for fast and reliable traffic sign detection using images obtained from a single front-facing road vehicle camera. Our focus is on a methodology for reducing the computational requirements and increasing the performance of existing detection methods by refining the image space search using 3D scene geometry. Information concerning physical traffic sign dimensions and vehicle camera parameters is integrated into a model that predicts the image scales and locations at which traffic signs are likely to appear. We apply our framework to a Haar-feature-based detection method trained on a collection of stop signs. Experimental results show that the refined image search space results in much less computation time while retaining the same true positive detection performance as existing methods that search all image scales and locations. In addition, false positives at physically implausible traffic sign locations are eliminated.**

I. Introduction

The broad problem addressed in this paper is the autonomous detection of road signs using data acquired from a single camera mounted on a road vehicle. Fast and reliable road sign detection could enable autonomous vehicle navigation in uncharted urban environments by providing information about traffic patterns, road hazards, speed limits, locations to stop and yield, and more. In addition, traffic sign detection systems can be implemented in human-operated vehicles as a warning system for weary drivers.

Given a sequence of images or video taken from the perspective of a vehicle, an ideal traffic sign detector should be able to perform the following tasks: (1) store invariant visual models of commonly encountered road signs; (2) search the image space and extract the pixel coordinates and size of any visible traffic signs; (3) identify the type of traffic sign (stop, yield, speed limit, warning, etc); (4) estimate the location of the sign relative to the vehicle; (5) perform reliable detection in real time.

Existing road sign detection approaches use a variety of models to represent traffic signs (task 1), and a wide range of detection methods to locate the signs within the image (task 2). Each method exploits one or more unique design aspects of traffic signs such as color, perimeter shape, and face pattern.

In many systems using color information, a pre-segmentation operation is first performed using thresholding in RGB, HSV, or customized color spaces. Shape-based models for traffic signs are subsequently used to detect signs in the pre-segmented image regions. For instance, Escalera et al. use corner features to model signs [1], Zadeh et al. use edge features [2], and Torrensen et al. use image templates [3]. Several authors examine a joint treatment of color and shape, such as Fang et al. [4] and Bahlmann et al. [5]. Lopez et al. use only color information to identify and track image regions likely to contain signs [6].

Other authors have implemented traffic sign detection algorithms on grey scale images. Gavrila et al. model traffic signs using templates, and employ a hierarchical detection method using distance transforms [7]. Shen et al. use polygon geometry to identify signs in cluttered images [8], and Barnes et al. use radian transforms and cross correlation for detecting traffic sign templates [9]. Baro et al. use a set of rectangular Haar features for modeling and detection of traffic signs [10].

However, none of the approaches listed above use *a-priori* information about the physical dimensions and mounting locations of road signs in the 3D environment. Such information is standardized for particular road environments by government agencies, and is readily available (for example, see [11]). Existing detection methods waste valuable computation time searching for signs in physically improbable locations, and furthermore, register false positives in image locations at which signs cannot exist. Hoiem et al. present a probabilistic approach to object detection using 3D geometry [12], but their method relies on object interdependence, and focuses on detection with a static camera. Picciolo et al. use a segmentation technique in traffic sign detection to eliminate uniform sections of sky and road [13], and limit their search to the right side of images, but do not present a quantitative model for incorporating physical traffic sign geometry.

Referring back to the task list, the primary contribution of this paper is the introduction of an add-on methodology to existing traffic sign detecors for intelligently refining the image-space sign search in task 2. Utilizing information about the vehicle's camera and physical dimensions of signs, we can predict the locations and sizes of the signs within the images captured by the camera. These predictions allow us to eliminate large regions of search space that cannot physically contain traffic signs. The reduced search space aids task 5 by significantly lowering the computation time required to search each image, as well as eliminating false traffic sign detections at unrealistic locations. As a result of our assumed scene geometry, we are also able to address task 4 by estimating the distance between the traffic sign and the approaching vehicle.

We address task (1) in our experiments by modeling stop signs using a learned cascade of simple Haar-features, as

J. Schlosser is with the Mechanical Engineering Department, Stanford University, Stanford, CA 94305 USA jschloser@stanford.edu
M. Montemerlo is with the Computer Science Department, Stanford University, Stanford, CA 94305 USA mmde@stanford.edu
J.K. Salisbury is with the Departments of Computer Science and Surgery, Stanford University, Stanford, CA 94305 USA jks@robotics.stanford.edu

introduced by Viola and Jones [14]. Stop signs were chosen because of their high incidence in urban environments, and their importance in preventing vehicular accidents. Noting that our search space reduction framework is applicable to almost any existing detection algorithm, we picked Haar-based detection for its robustness to lighting and scale, its computational efficiency, and its applicability to greyscale images (to avoid introducing the color thresholding problem). Traffic sign classification (task 3) is outside this paper's scope.

The core of the paper is section 2, which describes our novel image search-space reduction approach. Sections 3 and 4 describe validation results of the approach via data collection and computational experiments. We conclude with a discussion of the results and future research directions in section 5.

II. Approach

A. Forming a road sign model

Our approach is valid for any road sign model and corresponding detection method that searches image space at particular locations and scales. For the experiments in this paper, we employ a combination of simple rectangular features (called Haar features) to model the traffic sign in image space (Fig. 1). These features can be quickly computed using a concept called the "integral image" [13]. Using the Haar features and a boosting scheme called Adaboost, an efficient cascaded classifier can be trained that contains information only about the critical visual features of the sign [14]. Signs are detected within an image by evaluating the trained features at a particular image location and scale (detection window).

B. 3D scene assumptions

Our 3D scene model makes the following key assumptions: (1) traffic signs of a particular type have similar mounting heights relative to the ground; (2) the vehicle's pitch and the road's incline are limited; (3) signs of interest are facing the driver and camera; (4) the calibration matrix and height of the vehicle camera are known.

As mentioned in the introduction, precise traffic sign dimensions and mounting requirements are primarily government regulated, justifying our assumption of standardized sign geometry. For visibility reasons, traffic signs are generally designed to face the driver so that they are easily spotted. Furthermore, the incline of most urban roads is small for driver comfort and safety reasons, allowing us to limit the maximum road steepness and vehicle pitch in our model. Finally, since the vision system for autonomous vehicles remains the same throughout the course of driving, the camera calibration information will remain constant. A list of parameters necessary for the subsequent analysis is shown in Table I.

Our assumptions place restrictions on the locations where a traffic sign may exist in 3D space (X_0, Y_0, Z_0). As Fig. 2 illustrates for the case of flat ground, the sign may be posted anywhere in ground plane defined by Z_0 and Y_0, but its height and physical dimensions are specified.

C. Projection of scene geometry into image coordinates

Plausible 3D traffic sign locations can be projected into pixel coordinates (x', y') within the image using the standard pinhole camera model,

$$Z\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} fs_x & fs_\theta & o_x \\ 0 & fs_y & o_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} R & T \\ 0 & 1 \end{bmatrix} \begin{bmatrix} X_0 \\ Y_0 \\ Z_0 \\ 1 \end{bmatrix} \quad (1)$$

or more concisely,

$$Z\mathbf{x}' = K\Pi_0 g X_0, \quad (2)$$

where $\mathbf{x}'$ is the vector of pixel coordinates, Z is the 3D Z-coordinate in the camera frame, K is the camera calibration matrix, Π_0 is the canonical projection matrix, and g is a Euclidean transformation between 3D camera ($\mathbf{X}$) and world ($\mathbf{X}_0$) coordinate frames. The camera frame is assumed to be rigidly attached to the vehicle.

Equation (2) can now be used to project any 3D road sign location to pixel locations of the sign's bounding box in the image. Fig. 3 shows the transformation for a grid of stop signs evenly spaced in 3D, similar to the grid illustrated in Fig. 2.

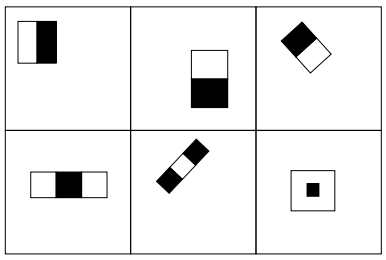

Fig. 1. Typical 2 and 3- rectangle Haar features used to model traffic signs. The value of each feature is the difference between the sum of pixels in the light and dark rectangles.

TABLE I
3D Scene and Camera Knowledge

Symbol	Interpretation
h_s	Average height of the middle of a sign measured along the X_0-axis in world coordinates
d_x	Width of a sign measured along the Y_0-axis in world coordinates
d_y	Height of a sign measured along the X_0-axis in world coordinates
f	Focal length of the vehicle camera
s_x, s_y, s_θ	Scaling factors relating pixels to metric distance units
o_x, o_y	Pixel coordinates of the location where the camera z-axis intersects the image plane
h_c	Height of the vehicle camera with respect to the ground
$\theta_{incline,max}$	Maximum expected road incline
$\theta_{pitch,max}$	Maximum expected vehicle pitch

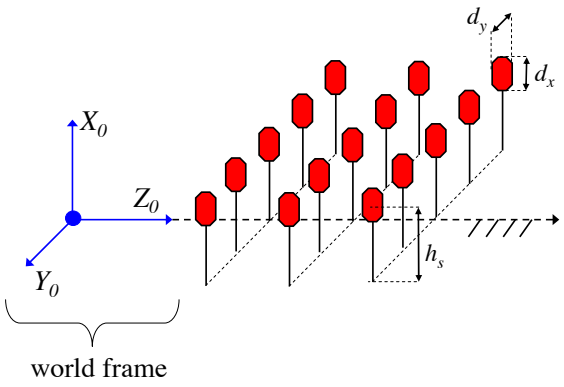

Fig. 2. Assumed 3D road scene context, showing possible locations of traffic signs on flat ground.

Fig. 3. Projection of various possible 3D traffic sign locations into the 2D image plane using (2).

D. Traffic Sign Detection Framework

In order to detect signs in an image, a reference point of the detection window must be specified in pixels (we will use the upper left corner), along with the scale of the detection window. Denote the reference point of the detection window in the image coordinates as (x'_w, y'_w). In many existing detectors, the search is executed according to the pseudocode shown below:

```
for(all possible window sizes s)
  for(all possible x'_w image coordinates)
    for(all possible y'_w image coordinates)
      run detector at (x'_w, y'_w) for scale s
    end
  end
end
```

According to this algorithm, locations in 3D space where a traffic sign could not physically exist will still be searched. This wastes valuable computation time, and increases the likelihood of false positives being detected, since the search space is larger than what is necessary.

Instead, it is proposed that the detection should proceed according to the following pseudocode:

```
for(all possible window sizes s)
  calculate Z-coordinate of sign
  for(all possible x'_w image coordinates)
    for(valid y'_w image coordinates)
      run detector at (x'_w, y'_w)  for scale s
    end
  end
end
```

We now detail each step in the pseudocode shown above.

1) Iterate through detection window sizes: The outer loop of our framework cycles through all detection window sizes in pixels, $s = (x'_{size}, y'_{size})$, for which the user would like to watch for traffic signs. Note that the relationship between x'_{size} and y'_{size} is fixed, according to the sign's aspect ratio. The lower limit on window size is dictated by the chosen deteciton algoithm, and the upper limit should be set for the maximum size a sign could appear in the image (depends on camera parameters). Iteration step size is determined by the sensitivity of the detection algorithm to window size.

2) Calculate appropriate Z-coordinate of sign: The next step is to calculate the expected distance of the traffic sign from the camera (Z), according to the current detection window size s. Given that 3D world-frame points on the top and bottom edges of the sign are $\mathbf{X}_0^{top,bottom} = (h_s \pm d_x/2, Y_0, Z_0, 1)^T$, (2) is used to compute the Z-coordinate of the sign as follows:

$$Z = (f s_y d_x)/y'_{size} \tag{3}$$

where y'_{size} is the current y' window size in pixels.

3) Iterate through x'_w coordinates: After calculating the sign's distance from the camera, the algorithm iterates through all possible x'_w coordinates in the image. In reality, not all the x'_w coordinates are valid positions, since traffic signs cannot be located in the middle of lanes. However, to account for curving roads and signs mounted in the median, the entire range of x'_w coordinates are checked across the image.

4) Iterate through valid y'_w coordinates: Next, only the valid range of y'_w coordinates are scanned, reducing the image space search for a given Z-distance of the sign from the camera. If the road is perfectly flat, and our 3D scene model is perfect, we can determine y'_w with certainty for a particular distance Z. That location is given by

$$y'_{nom} = [f s_y (h_c - h_s - d_x/2) + o_y Z]/Z \tag{4}$$

However, several sources of uncertainty will affect the actual y'-location of a traffic sign in the image, including: (1) vehicle pitch; (2) incline of the road; (3) camera calibration parameters; (4) height of the camera relative to the ground; (5) physical dimensions of the traffic sign. To mitigate these issues, a pixel uncertainty $\Delta y'$ is added and subtracted from y'_{nom} to form a search window. The first three uncertainty sources listed above are the most significant in our problem; we examine those here.

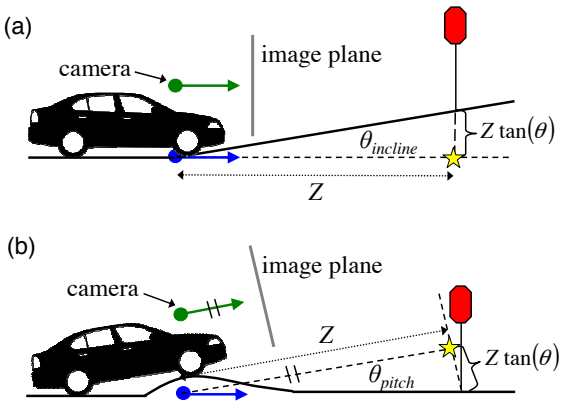

Fig. 4. Scenarios in which the nominal pixel coordinate of the sign in the y' direction could change due to (a) road incline and (b) vehicle pitch. The star represents the nominal location of the base of the sign.

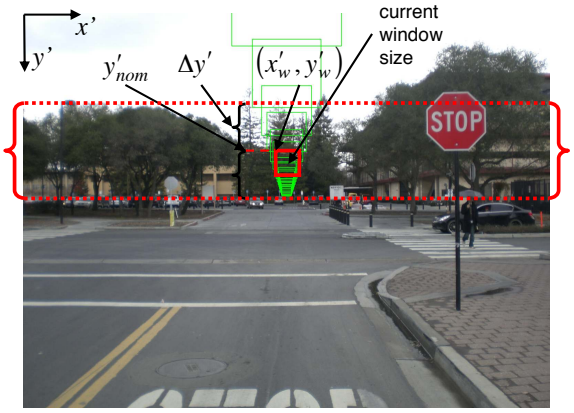

Fig. 5. Search region for a particular window size using 3D scene geometry. The search region is bracketed (best viewed in color).

Road incline and vehicle pitch: Fig. 4 (a) depicts a vehicle encountering a road incline, and Fig. 4 (b) depicts a vehicle that is pitched upwards (for instance, due to a speed bump). In each case, the effective location of the road sign relative to the ground of the world frame is changed by a distance $\pm Ztan(\theta)$. If we incorporate this term into (4), the corresponding change in pixel coordinate due to pitch or road incline is

$$\Delta y'_\theta = f s_y tan(\theta_{max}) \tag{5}$$

where $\theta_{max} = f(\theta_{pitch,max}, \theta_{incline,max})$ is a user-selected parameter based on the road environment. When selecting θ_{max}, note that in many commonly encountered scenarios, $\theta_{pitch} = \theta_{incline}$ as the sign is approached, and the nominal sign location remains unchanged. Also notice that the pixel uncertainty $\Delta y'_\theta$ is independent of the distance Z of the sign from the camera.

Traffic sign height: Considering an uncertainty of Δh_s pixels in traffic sign height, the corresponding uncertainty in pixel location of the sign is given by

$$\Delta y'_h = (\Delta h_s f s_y)/Z \tag{6}$$

Total pixel uncertainty is the combination of the two uncertainties outlined above,

$$\Delta y' = \Delta y'_\theta + y'_h = f s_y (tan(\theta_{max}) + \Delta h_s / Z) \tag{7}$$

The search window in the vertical image direction for y'_w is $[y'_{nom} - \Delta y', y'_{nom} - \Delta y']$. Combined pixel uncertainty is greater for closer Z distances, meaning the search window size decreases as the algorithm scans for traffic signs further away from the vehicle.

5) Run detector at x'_w, y'_w for size s: As demonstrated by Fig. 5, the search region for a particular window size is reduced from the whole image to a small strip when 3D scene context is incorporated into the detection framework. The traffic sign detector is now run at each location within the new search region. If a traffic sign is detected, we may estimate the distance of the sign from the vehicle as the coordinate Z, computed in step 2. The accuracy of the distance estimate will depend on the window step-size iteration selected in step 1.

III. Experimental Setup and Data Collection

To verify our approach, we first collected a series of 4000 images using a tripod-mounted camera in a moving vehicle (Fig. 6 (a)). The images were obtained at 5 frames/second in bursts of 1-15 seconds while approaching intersections with stop signs, and at random intervals throughout a residential-area drive. A Nikon D700 camera with 24mm lens focused at infinity captured the images at 2128x1416 resolution. Table 2 shows the physical parameters of the road scene and the results of camera calibration.

Dimensions d_x and d_y are standardized for stop signs in urban and residential streets in the USA [11]. Modified values are also available for freeways and specialized areas, had the vehicle entered such zones. Stop sign height h_s is also standardized, with small variations allowed in certain situations (such as multiple traffic sign mountings on a single post), so Δh_s was chosen accordingly. Angle θ_{max} was picked to account for small vehicle pitch and incline mismatches, noting that the data collection was performed in a relatively flat area.

Next, the collected images were converted to grayscale and used to train a cascaded Haar classifier. A total of 700 stop signs were manually cropped from the images, scaled to 25x25 resolution, and used as positive samples (Fig. 6 (b)). Negative samples were obtained at random from 1500 images that did not contain stop signs.

The trained stop-sign classifier and the parameters in Table II were used to implement our traffic sign detection framework. Code was written in C++ using the OpenCV computer vision library on a Intel Core i7 2.67GHz machine running Windows. Our test set consisted of 300 images that were not used in the classifier training process, containing 21 unique stop signs. We use (3), (4), and (7) to compute Z-coordinate, nominal y' search coordinate, and pixel uncertainty, respectively. Computational results comparing our detection process to a comprehensive search over all image locations and scales are presented in the following section.

TABLE II

PARAMETERS FOR STOP SIGN DETECTION

Symbol	Value	Symbol	Value
h_s	2.1 meters	fs_θ	0
d_x	75 cm	o_x	1055
d_y	75 cm	o_y	698
fs_x	1427	h_c	1.1 m
fs_y	1427	θ_{max}	5 degrees
Δh_s	0.2 meters		

(a) (b)

Fig. 6. Experimental setup and sample training images.

IV. RESULTS

Table III summarizes the outcomes of our experiments, and Fig. 7 shows typical stop sign detection results using our 3D scene geometry framework. Detection percentage refers to the total number of signs (not necessarily unique) detected in the image set relative to the total number of signs present. Signs with sizes less than the training template (25x25) are not counted as present, since they cannot be detected using the trained Haar features.

A. Computation time

As Table III demonstrates, our add-on framework significantly decreased the computation time searching for traffic signs within each image. The computation time scales with the size of the window used for traffic sign detection, and resulted in an average reduction factor of 2.44 compared to a full search.

TABLE III

EXPERIMENTAL RESULTS

	Comprehensive Image Search	Reduced Search Using 3D Scene Context
Average Computing Time	1014 ms/image	415 ms/image
# Unique Signs Detected	21/21	21/21
Total # Signs Detected	283/380	283/380
Detection Percentage	75%	75%
Total False Positives	19	12

Fig. 7. Stop sign detection results using 3D scene context framework.

B. Traffic sign detection

Equally significant, our reduced image space search detected all 21 of the traffic signs in the testing images, and achieved the same detection percentage compared to the comprehensive image search. This means that our reduction in computation time did not come at the cost of reduced sign detection performance.

C. False positives

Fig. 8 shows the same image analyzed by the comprehensive search algorithm, and by our 3D scene context software. Notice that a false positive is registered in the comprehensive search, but is not detected using 3D scene context. Since our framework only searches physically plausible traffic sign locations, it does not accumulate false positives such as the one in Fig. 8 (a), explaining the reduced number false positives presented in Table 3 for the 3D scene context search.

D. Effect of pixel uncertainty

Without data about vehicle pitch or terrain elevation, we must pick relatively large values for $\Delta y'$ to detect all traffic signs in a given environment. Reducing $\Delta y'$ decreases computation time, but also risks missed traffic sign detections due to the uncertainties mentioned in the Approach section. Fig. 9 quantifies this effect by varying the θ_{max} parameter in our 3D scene context implementation (with Δh_s =0.2m), using a 50-image subset of testing data. The number of correct detections levels out at 51, meaning that further increasing θ_{max} will not likely yield better results. Since computation time continues to increase with increasing θ_{max}, the ideal parameter to use is θ_{max} = 5 degrees.

(a) (b)

Fig. 8. Physically implausible false positive registered by comprehensive search algorithm (a), but skipped by 3D scene geometry search (b).

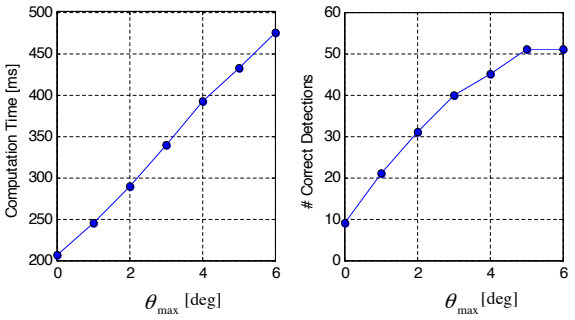

Fig. 9. Effect of pixel uncertainty on computation time and road sign detections in 50-image sample set.

V. Discussion

The number of correct detections, false positive rate, and computation time for Haar-based object detection methods are highly sensitive to factors such as image resolution, pixel step size during searches, and training parameters like number of classifier stages, minimum hit rates, number of training images, and many others. Therefore it was ensured that all of these factors remained constant during the comparison between the comprehensive search algorithm and the 3D context algorithm. Had more training examples or higher minimum hit rate been used in the training process, the percentage of correct detections would have been higher. However, the relevant information for the evaluation of our add-on methodology is the *relative* false positive rate and correct detections, and it is clear from the previous section that our 3D context algorithm performs better relative to the comprehensive algorithm.

Although the search space reduction framework was discussed in the context of grey-scale detection for a single type of traffic sign, it is applicable to a much wider range of traffic sign detection methods. For instance, the framework can be used with color-based segmentation methods to limit the region of the image that needs to be color-thresholded, allowing more computationally-expensive and robust color mapping techniques to be used. Furthermore, our framework can be used to detect multiple types of traffic signs by searching the image multiple times, each for a different type of sign, and each with a unique search window defined by its physical parameters and our 3D scene model. Alternatively, bounds for the physical sign parameters could be implemented to account for multiple sign types, and the image could be searched in a single run for all types of signs with a reduced search space based on the bounds.

A number of areas remain unexplored for future research and testing. First, our method could be further tested in areas with extreme hills or rough terrain, in order to evaluate the algorithm's performance in a wider range of 3D scenes. In these cases, it is likely that the uncertainty parameter θ_{max} would need to be increased to account for wider variations in vehicle pitch and road incline discrepancies. An analysis of the effect of pixel uncertainty on detection success in such environments could be enlightening. Second, a search space reduction method in the horizontal image direction would be a valuable add-on to our method. For example, road boundaries or lanes could be detected via edge chains [15], probabilistic Markov-style processes [16], or multi-model methods [17], and the projection of the localized road coordinates into the image space could be excluded from our search.

VI. Acknowledgements

The authors thank Jana Kosecka for several helpful discussions, and Blaise Hamel for assistance in capturing the experimental images. Jeffrey Schlosser is supported by an NSF fellowship.

References

[1] 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.

[2] Mahmoud M. Zadeh, T. Kasvand, and Ching Y. Suen. Localization and recognition of traffic signs for automated vehicle control systems. volume 3207, pages 272–282. SPIE, 1998.

[3] J. Torresen, J.W. Bakke, and L. Sekanina. Efficient recognition of speed limit signs. In *Intelligent Transportation Systems, 2004. Proceedings. The 7th International IEEE Conference on*, pages 652 – 656, 3-6 2004.

[4] Chiung-Yao Fang, Sei-Wang Chen, and Chiou-Shann Fuh. Road-sign detection and tracking. *Vehicular Technology, IEEE Transactions on*, 52(5):1329 – 1341, sept. 2003.

[5] C. Bahlmann, Y. Zhu, Visvanathan Ramesh, M. Pellkofer, and T. Koehler. A system for traffic sign detection, tracking, and recognition using color, shape, and motion information. In *Intelligent Vehicles Symposium, 2005. Proceedings. IEEE*, pages 255 – 260, 6-8 2005.

[6] Luis David Lopez and Olac Fuentes. Color-based road sign detection and tracking. *Lecture Notes in Computer Science*, pages 1138–1147, 2007.

[7] Dariu Gavrila. Traffic sign recognition revisited. In *Mustererkennung 1999, 21. DAGM-Symposium*, pages 86–93, London, UK, 1999. Springer-Verlag.

[8] Hua Shen and Xiaoou Tang. Generic sign board detection in images. In *MIR '03: Proceedings of the 5th ACM SIGMM international workshop on Multimedia information retrieval*, pages 144–149, New York, NY, USA, 2003. ACM.

[9] N. Barnes and A. Zelinsky. Real-time radial symmetry for speed sign detection. In *Intelligent Vehicles Symposium, 2004 IEEE*, pages 566 – 571, 14-17 2004.

[10] J Vitri X Bar. Traffic sign detection on greyscale images. In *CCAI*, pages 209–216, 2004.

[11] Federal Highway Administration. Manual on uniform traffic control devices, 2009.

[12] Derek Hoiem, Alexei A. Efros, and Martial Hebert. Putting objects in perspective. *Computer Vision and Pattern Recognition, IEEE Computer Society Conference on*, 2:2137–2144, 2006.

[13] G. Piccioli, E. De Micheli, P. Parodi, and M. Campani. Robust method for road sign detection and recognition. *Image and Vision Computing*, 14(3):209 – 223, 1996.

[14] Paul Viola and Michael Jones. Robust real-time object detection. 2001.

[15] M. Campani, M. Cappello, G. Piccioli, E. Reggi, M. Straforini, and V. Torre. Visual routines for outdoor navigation. In *Intelligent Vehicles '93 Symposium*, pages 107 –112, 14-16 1993.

[16] ZuWhan Kim. Robust lane detection and tracking in challenging scenarios. *Intelligent Transportation Systems, IEEE Transactions on*, 9(1):16 –26, march 2008.

[17] R. Labayrade, J. Douret, and D. Aubert. A multi-model lane detector that handles road singularities. In *Intelligent Transportation Systems Conference, 2006. ITSC '06. IEEE*, pages 1143 –1148, 17-20 2006.