Efficient Traffic Sign Detection Using 3D Scene Geometry

Jeffrey Schlosser
Stanford University
Stanford, CA 94305
jschlosser@stanford.edu

Abstract

This paper proposes a new framework for fast and reliable traffic sign detection exploiting 3D scene geometry. A collection of images are obtained from front-facing cameras in vehicles. Haar classifiers are used to train representations of traffic signs, and 3D scene context information is used to refine the sign search within the images. The reduced image search space results in less computation time and lower false positive rates, while retaining the same true positive detection performance.

1. Introduction

The broad problem addressed in this paper is to quickly and reliably detect road signs using data acquired from a camera mounted on an autonomous vehicle. The specific focus of the work presented here is to utilize information about the 3D geometry of the road scene to improve the performance of previous detection methods. A new framework for traffic sign detection incorporating scene-specific 3D information is described, and is compared to a model that does not use 3D context information.

1.1. Summary of approach and results

Ideally, given a sequence of images or a video of the road and surrounding environment taken from the perspective of the vehicle, a traffic sign detector should be able to perform the following tasks:

1. Store invariant visual models of commonly encountered road signs
2. Extract the pixel coordinates and size of the traffic signs within the images
3. Identify the type of traffic sign (stop, yield, speed limit, warning, etc)
4. Perform reliable detection in real time

The second task is the primary focus of this paper. To simplify task (1), only a single type of traffic sign, the stop sign, is learned and stored in memory. Task (2) involves a search within the images to detect the traffic sign model learned in task (1). Traditionally, a computationally expensive search has been performed at all scales and locations within the image space. The contribution of this paper is to refine that search using the expected size and location of traffic signs in a 3D road scene. Task (3) is omitted in this paper's discussion, since detection of only one type of sign is described.

Task 4 is achieved with the proposed detection algorithm using 3D scene context. Frame rates of about 14 FPS are observed on a large image sequence, with much greater rates possible using more accurate camera calibration and scene information. In addition, it was found that the proposed algorithm is more reliable than comparable traditional algorithms—it yields a significantly lower false positive rate with an equal number of true positives compared with algorithms using comprehensive searches.

1.2. Motivation for traffic sign detection

Autonomous traffic sign detection is interesting and applicable for a couple of reasons. First, it could enable autonomous vehicles to explore urban environments for which they have little previous knowledge. For example, in the DARPA urban challenge, the autonomous vehicles had a large amount of prior information available about the traffic laws and roads before the contest began. Without this prior information, the vehicles would not have been able to assume fully autonomous operation. A traffic sign detector could greatly aid vehicles without prior information by indicating where the vehicle should stop, speed limits, environmental conditions it should be aware of (such as construction or wet pavement), traffic patterns, and more.

Second, traffic sign detection could be used to autonomously build road maps of uncharted urban areas. A vehicle with a human driver could be driven within a certain area, and data from a video camera mounted on the car could be collected. This data could be later analyzed by the traffic sign detector to identify where an autonomous vehicle should stop, yield, or perform other operations within the area. A road map could be built using the data, and vehicles such as those used in the DARPA urban challenge could then autonomously navigate the areas using the data collected.

2. Prior Work

The fast and accurate detection of objects within an image has been the focus of much research in the recent past. For example, Viola and Jones have implemented an object detector using the AdaBoost learning algorithm and a cascade of classifiers, capable of running in real time [1]. Tuzel et al. have used Reimannian manifolds to achieve high rates of human detection in cluttered scenes [2]. In video sequence data, Mikolajczyk et. al. propagate object detection probabilities over time to improve upon frame-based detection results [3]. Several detectors specific to road signs have been also developed, such as an edge-based detector proposed by Piccioli et. al. [4], a polygon geometry detector by Shen and Tang [5], and a detector using template hierarchies and distance transforms proposed by Gavrila and Philomin [6]. However, none of these detectors exploit information about the 3D scene which is being captured.

Hoiem et al. were among the first to model relationships between visual objects in the 3D world in the context of object detection [7]. They used probabilistic estimates to simultaneously refine 3D geometry and object detection hypotheses, showing improvements over traditional 2D image searches. The framework proposed in this paper goes one step further by using knowledge about the physical dimensions and locations of objects (traffic signs) in 3D space, as well as information about the camera location and calibration parameters, to further refine the search for signs within the image.

3. Approach

This section describes the novel framework for traffic sign detection that was implemented utilizing 3D scene context information. Traffic sign detection consists of two primary stages:

1. Establishing a traffic sign representation
2. Detecting the traffic signs using 3D scene geometry

The following subsections will describe each stage in detail.

3.1. Forming a traffic sign representation

In order to detect traffic signs in an image, a representation of the signs must first be formed. There are several different ways of forming object representations, including the use of trained Haar features [1], scale invariant feature transforms (SIFT) [8], template hierarchies [6], and polygon geometry [5]. The framework presented in the next section is independent of the specific object detection representation, so any of the methods listed above can be used. The main focus of this paper is on the efficient detection of signs, not on forming sign representations, so the easiest and most accessible method was chosen. Intel's OpenCV (Open Computer Vision) library, the programming library of choice in this paper due to speed and memory requirements, already contained code for training Haar classifiers, so Haar features were chosen to represent traffic signs. It was decided to focus exclusively on representation and detection of stop signs, without loss of generality for the proposed framework.

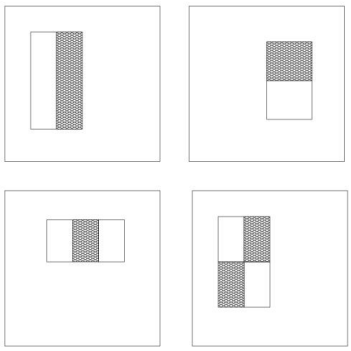


Figure 1: Typical rectangular Haar features used for traffic sign representation.

To give the reader a general understanding of representing objects with Haar features, a brief summary of the algorithm is presented here. Figure 1 shows a few examples of the simple rectangle features used for object classification.

The value of Haar features within a given detection window can be quickly computed using a concept known as the "integral image," which contains information about the sum of all pixels in the rectangle above and to the left of any given pixel location. A very large number of these features are contained within a detection window, so to avoid evaluating all of the features, a classification function can be learned that finds a relatively small set of features that adequately represent the object. Using a boosting scheme called Adaboost, an efficient cascaded classifier can be trained that contains information only about the critical visual features of the object. For more information, please refer to [1].

Training the cascaded Haar classifier requires a very large number of both positive and negative sample images. The training data used in the model developed for this paper consisted of several data sets. First, a sequence of about 2400 images captured from a front-facing camera mounted to Stanford's DARPA urban challenge vehicle was provided by Mike Montemerlo, a member of the DARPA urban challenge team. Second, a set of about 700 images collected from a front-facing camera mounted on a

Figure 2: Example training images for Haar classifier.

car were provided by Jana Kosecka. Third, a collection of random background images were obtained from the internet as negative samples. Finally, an array of digital photographs taken by the author while standing in the road, with the camera at eye level, were taken around campus at Stanford University. Examples of the images are shown in Figure 2.

The most robust way to train the Haar classifier is to manually input the locations of all signs in the training images. Since the number of images required for training is so large, this would require a large time investment, and again since object representation is not the focus of this paper, a less time consuming method was used. OpenCV has a built-in function that randomly pastes a template into background images and automatically identifies where in the image the template is pasted. In this fashion, the classifier can be trained without manually identifying traffic signs, with a moderate performance sacrifice.

The Haar training functions in OpenCV are only able to train and detect objects in images using a single image channel. There were several options in choosing the single channel when training a representation of the stop sign, including

- Black and white single-channel versions of the color images
- Only the red channel of the color images
- The saturation channel of transformed HSV images

Each of these options was explored by the author. Using OpenCV's default object detection function (cvHaarDetectObjects) and the trained Haar classifiers

Figure 3: Training results on different image channels (top: saturation; left: red; right: black and white)

using each of the channels listed above, the results shown in Figure 3.

As evidenced by these images, it is clear that the black and white channel classifier performed the best. It was the only classifier that correctly identified the stop sign, and registered no false positives. The red channel classifier registered no false positives, but failed to find the stop sign. This is likely due to the fact that when extracting the red channel data from images, image sections that are purely white will have a full red component as well. This makes the letters on the stop sign very hard to detect with the Haar features, since the features will register very little difference between the letters and the rest of the stop sign when using the red channel. The saturation channel classifier registered two positives close to the stop sign (not directly on it), but a large number of false positives elsewhere in the image. It is hard to speculate why the saturation classifier registered so many false positives, but it is probably due to the large saturation channel variation of various background objects in the training images. Due to time constraints, other options such as RGB thresholding were not attempted.

In the following section, the Haar feature representation described above will be used along with 3D scene information to locate signs within new images.

3.2. Detecting road signs using 3D scene geometry

Once a representation of the traffic sign is formed, the signs may be detected in new images by searching the image space for the representation. In this case, the new images will be searched for the Haar classifier features trained in the previous section. However, it is important to note that the material presented in this section is not specific to Haar classifier features. In any case, the new image space must be searched for the stored representation of the sign, whatever it may be. The goal of this section is to describe a framework for restricting the image space search to specific areas and scales based on 3D scene and traffic sign geometry, thus improving the speed and reliability of current detection schemes.

Faster and more reliable traffic sign detection can be achieved by taking advantage of prior knowledge about the geometry of road scenes, as well as the specific type of images obtained by front-facing cameras mounted on vehicles. Specifically, the following assumptions are made about the 3D road scene:

- All stop signs have similar dimensions
- The signs' heights above the ground are similar
- The road is relatively flat
- The signs are directly facing the driver and camera
- The signs are not occluded
- The focal length and pixel scale factors of the vehicle camera are known

All of these assumptions are justified in the context of stop sign detection. It is very rare to find a stop sign that has dimensions that significantly differ from other stop signs, or is mounted at a height that radically differs from other signs. Most of the time roads are designed to be flat from driver comfort, but the framework does allow for small inclines and declines in the road. In addition, stop signs are designed to directly face the driver without occlusions so that they are easily visible. Finally, since the vision system for autonomous vehicles remains the same throughout the course of driving, the focal length and scale factors will also remain constant. Note that a similar list of assumptions can be made about any type of traffic sign, not just stop signs. A list of assumed parameters necessary for the subsequent analysis is shown in Table 1.

Table 1. Assumed 3D scene and camera knowledge.

Symbol	Interpretation
h	Average height of the middle of a sign measured along the x-axis in camera coordinates
d_x	Width of a sign measured along the y-axis
d_y	Height of a sign measured along the x-axis
f	Focal length of the 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 z-axis intersects the image plane

The assumptions listed above can be used to restrict the search for traffic signs within images. The following equation relates pixel coordinates in the image (x', y') with 3D coordinates of points (X_0, Y_0, Z_0), using the standard pinhole camera model:

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

or more concisely,

$$Z_0 \boldsymbol{x}' = K \Pi_0 g \boldsymbol{X}_0 \quad (2)$$

where $\boldsymbol{x}'$ is the vector of pixel coordinates, Z_0 is the 3D Z-coordinate in the camera frame, K is the camera calibration matrix, Π_0 is the standard projection matrix, g is a Euclidean transformation, and $\boldsymbol{X}_0$ is the vector of 3D scene coordinates in the camera frame. The geometry of this setup is illustrated in Figure 4.

Based on the geometry shown in Figure 4, we have the

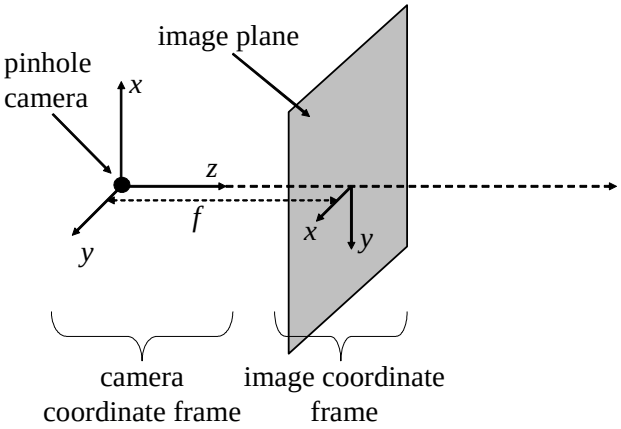


Figure 4: Pinhole camera geometry.

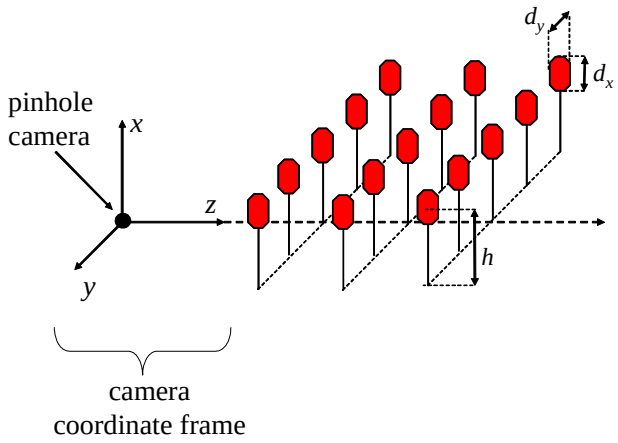


Figure 5: Assumed 3D road scene context, showing possible locations of traffic signs.

following transformations:

$$R = \begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \text{ and } T = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

The matrix K is obtained by calibrating the particular camera used to produce the images. Therefore the only variables we have in Equation (2) are the pixel coordinates $\boldsymbol{x}'$ and the 3D coordinates $\boldsymbol{X}_0$.

Our assumptions place restrictions on the locations in images where a traffic sign may exist in 3D. Specifically, the sign must be at a certain height relative to the camera (h), and it must possess particular dimensions in the x and y directions (d_x, d_y). The sign is free to translate in the z and y directions. Figure 5 illustrates this information.

Equation (2) can be used to transform any 3D road sign location to pixel locations of the sign's bounding box in the image. Figure 6 shows this transformation for a grid of stop signs evenly spaced in 3D, similar to the grid in Figure 5. Using this transformation, one only has to search the image at the scales and locations identified by the projection of the possible 3D traffic sign coordinates to the 2D image.

In order to detect objects in an image, a reference point of the detection window must be specified in pixels (in the case of Haar classifiers, it is 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 existing detectors, the search is executed according to the pseudocode shown below:

Figure 6: Projection of various possible 3D traffic sign locations into the 2D image plane using equation (2).

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

According to this algorithm, locations in 3D space where a traffic sign could not possibly 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 (z-coordinates in camera frame)
  calculate appropriate window size s
  for (all possible x'_w coordinates in
  the image)
    for (valid y'_w coordinates in the
    image)
      run detector at x'_w, y'_w for size s
    end
  end
end
```

The outer loop cycles through all 3D z-coordinates (Z) at which the user would like to watch for traffic signs. There is a practical limit to how large Z can grow, because as Z increases, the scale decreases, and the window size will eventually be too small to detect the sign. The next step is to calculate the appropriate window size based on the

current *z*-coordinate. The window size in the *y'*-direction can be found by projecting a 3D point on top of the sign and on bottom of the sign into 2D pixel coordinates, then taking the difference of the pixel coordinates. Since the window size is independent of the 3D *y*-coordinate in the camera frame (this can be seen in Figure 6—all rectangles at the same vertical location are the same size), and the *z*-coordinate and *x*-coordinate in the camera frame are specified, we have the following two 3D points to use:

$$\boldsymbol{X}_{top} = \begin{bmatrix} h + d_x/2 \\ 0 \\ Z \\ 1 \end{bmatrix} \text{ and } \boldsymbol{X}_{bottom} = \begin{bmatrix} h - d_x/2 \\ 0 \\ Z \\ 1 \end{bmatrix}$$

Notice that the *y*-coordinates were set to zero, since they are not relevant. These 3D points can be projected into the pixel coordinates $\boldsymbol{x}'_{top}$ and $\boldsymbol{x}'_{bottom}$ via equation (2). The window size in pixels can then be calculated as: $y'_{top} - y'_{bottom}$. If the size in the *x'*-direction and the size in the *y'*-direction are different, the *x'* window size can be calculated using the ratio $x_{size}/d_x = y_{size}/d_y$.

After calculating the scale, the algorithm cycles through all possible x'_w coordinates in the image. In reality, on a straight road, all of the x'_w coordinates are not valid positions, since traffic signs cannot be located in the middle of lanes. However, to account for curving roads, the entire range of x'_w coordinates are checked across the image. Next, *only the valid range of* y'_w *coordinates are scanned*. This is the most important step of the algorithm, in which the image search is greatly reduced for a given *Z* coordinate (or window scale). The valid search range of y'_w coordinates is simply given by $(y'_{top} - \Delta y'/2, y'_{top} + \Delta y'/2)$, where $\Delta y'$ is the pixel uncertainty in the y' direction. The pixel uncertainty is added in to mitigate several possible sources of uncertainty in the model, including

- Camera calibration parameters
- Height of traffic sign relative to camera
- Incline of the road

By expanding the search window in the y' direction by $\Delta y'$ pixels, the model becomes more robust to poor estimates of the calibration parameters and changes in traffic sign height and road incline. The size of the uncertainty $\Delta y'$ should be chosen based on the user's confidence level for each of the uncertain parameters listed above. Figure 7 illustrates the search window for a particular detection window size.

As demonstrated by Figure 7, 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. This reduction in search region size will yield a substantial reduction in computation time for sign detection, as demonstrated in the next section.

4. Results and Future Work

This section quantitatively assess the performance advantage of using 3D scene information to detect traffic signs. The end of the section describes several aspects of the framework that could be further refined in future research.

4.1. Performance Evaluation

To evaluate the performance of the proposed traffic sign detection framework using 3D scene context, the detection results were compared to results obtained with a traditional detection algorithm that scanned all scales and image locations. Since Haar features were used to form stop sign representations, the new algorithm was compared to OpenCV's function `cvHaarDetectObjects`. Typical detection results of the two algorithms run on one of the test images obtained by the author are shown in Figure 8.

The first aspect of Figure 8 to notice is that the comprehensive search registers two false positives, and the 3D context algorithm does not have any. Since the 3D context algorithm restricts the search only to physically plausible regions and scales on the image, it is not fooled

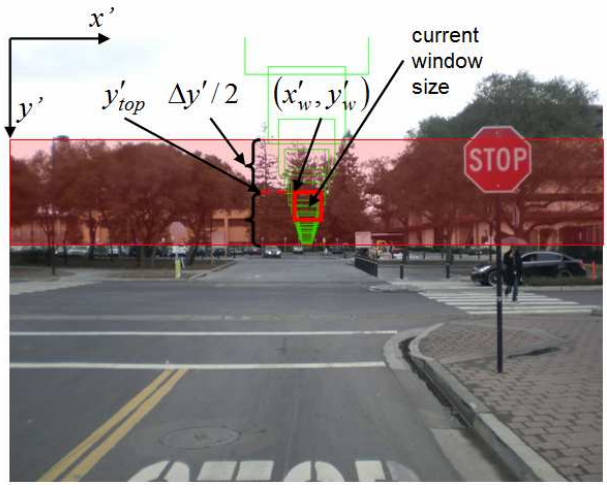


Figure 7: Search region for particular window size. The search region is shaded in red.

Traditional comprehensive search algorithm (`cvHaarDetectObjects`)

Proposed algorithm utilizing 3D scene context

Figure 8: Comparison of traditional algorithm and proposed algorithm on author's test image.

by the two false positives found in `cvHaarDetectObjects`, which are located at physically impossible scales and coordinates.

The second interesting aspect of the experiment is that the detection time using the comprehensive search was 1118 milliseconds, and the time using the 3D context algorithm was only 275 milliseconds. This is a direct result of the smaller search window depicted in Figure 7, since smaller search windows correspond to smaller computation times. It should be noted that the larger the test image, the better the computational speed improvement of the 3D context algorithm will be, since the search window constitutes a small percentage of the whole image (given a constant pixel uncertainty $\Delta y'$).

Next, the comprehensive search algorithm and the proposed algorithm were tested on the large sequence of images collected from a front-facing camera mounted to a vehicle, provided by Jana Kosecka. Typical images are shown in Figure 9, and a summary of the results are shown in Table 2.

Table 2. Image sequence testing results.

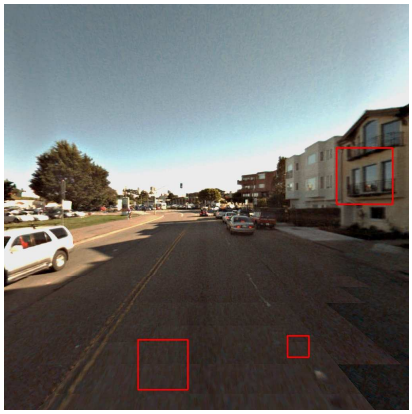
	Traditional Algorithm	Algorithm Using 3D Scene Context
Average Computing Time	262 milliseconds/image	79 milliseconds/image
FPS	3.8	12.7
False Positive Rate	1.53 FP/image	0.3 FP/image
# Correct Detections	5	5

The images in Figure 9 reveal again that false positives corresponding to unrealistic stop sign locations in the image are eliminated, but false positives that are at scales and locations at which a real stop sign could reside are not eliminated. Table 2 indicates that the FPS rate of the 3D context algorithm significantly outperformed the rate of the comprehensive search algorithm, and the false positive rate is decreased by more than 5 times. Notice that the number of correct detections made seems to be very low. This is mainly because the data set contained a small number stop signs, but also has to do with the quick and less-reliable random logo training method used (see the *Forming a traffic sign representation* section). However, the most important aspect of the detection data was that *the same number of correct detections were made in each case*, meaning that the much-improved false positive and FPS figures using the 3D scene context algorithm come at no sacrifice to true detection rates! This fact sums up the key benefit of implementing this paper's newly proposed traffic sign detection algorithm.

4.2. Discussion and future work

The false positive rate, number of correct detections, and FPS for each algorithm are highly subject to factors such as image resolution, the quality of the training method, relevance of the training images, and the detection sampling rate within a certain region in the image. Therefore it was ensured that all of these factors remained constant during the testing of the comprehensive search algorithm and the 3D context algorithm. Had a better training method been used in the experiments, or a better stop sign representation than Haar classifiers been used, the number of correct detections and the false positive rate could have been improved. However, the relevant information for this paper is the *relative* false positive rate and correct detections between the images, and it was clear from the previous section that the 3D context algorithm performed much better relative to the comprehensive algorithm.

Traditional comprehensive search algorithm (`cvHaarDetectObjects`)



Proposed 3D context algorithm

Figure 9: Typical results from testing on image sequence provided by Jana Kosecka.

Even better FPS results than those obtained in Table 2 for the 3D context algorithm could be obtained in several ways. First, the camera calibration parameters and camera height were imprecisely estimated by the author. Had precise calibration data and camera height data been available, the pixel uncertainty $\Delta y'$ (chosen as 120 in the experiments) could have been significantly reduced. Reducing $\Delta y'$ by a factor of ½ will yield a 2x faster computation time, since the search region is also cut in half. Second, the pixel uncertainty could have been further reduced if pitch data was available for the vehicle. By knowing the exact slope of the road, the height of stop signs relative to the camera could be estimated more precisely, thus decreasing $\Delta y'$. Finally, prior road maps or GPS data could have been used to estimate the road location. Knowing the exact location of the road, the search region for traffic signs could have been further reduced, since road signs cannot be located in the middle of the street.

Future research could focus on developing models for further reducing the image search region based on vehicle pitch data, road maps, and/or GPS. Prior information concerning the approximate location of traffic signs could also be incorporated in the detection algorithm. For example, the detector could turn on in the expected proximity of a sign, saving computation time elsewhere, and helping to precisely identify the actual location of the sign when detected. In addition, a revised algorithm could be developed that ran the sign detector more densely in image regions that are likely to contain traffic signs, thus increasing the reliability of the detector. Furthermore, temporal data could be used to boost FPS by switching to a simple tracking algorithm after a sign is detected in a particular frame.

5. Summary

A framework for fast identification of traffic signs in digital images was formulated by exploiting 3D scene information. Only regions within the image which could correspond to physical locations of traffic signs were searched, and faster computation times and lower false positive rates resulted. Correct stop sign identification rates remained the same as a comprehensive search algorithm, meaning that the proposed 3D context algorithm yielded faster FPS and lower false positive rates without sacrificing positive detection accuracy.

References

[1] P. Viola and M. J. Jones. **Robust real-time object detection**. Technical report, Compaq Cambridge Research Lab, 2001.

[2] O. Tuzel, F. Porikli, and P. Meer. **Human detection via classification on riemannian manifolds**. In Proc. CVPR, pages 1–8, 2007.

[3] Mikolajczyk, K., Choudhury, R., Schmid, C.: **Face detection in a video sequence - a temporal approach**. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. Volume 2, Kauai, Hawaii, USA, 2001.

[4] Piccioli, G., Micheli, E., and Campani, M. **A robust method for road sign detection and recognition**. Lecture Notes in Computer Science. Vol. 800, 1994.

[5] Shen, H., and Tang, X. **Generic sign board detection in images**. In: *MIR*. Berkeley, CA, November 7, 2003

[6] Gavrila, D. M., and Philomin, V. **Real-time object detection for "smart" vehicles**. In: Proc. of IEEE Internation Conference on Computer Vision, pp. 87-93, 1999.

[7] D. Hoiem, A. Efros, and M. Hebert. **Putting objects in perspective**. In CVPR, 2006.

[8] Lowe. D. **Distinctive image features from scale-invariant keypoints**. International Journal of Computer Vision, 2004.