Incremental Detection of Text on Road Signs

Wen Wu Xilin Chen Jie Yang

March 9, 2004

CMU-CS-04-116

School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213

Abstract

This paper presents a framework for incremental detection of text from road signs. The approach efficiently incorporates tracking and detection mechanisms into the same framework. The proposed approach first finds a set of discriminative feature points and clusters them into different regions. We then select candidate sign planes by a combination of color and vertical plane models. Within detected road sign planes, the framework selects candidate text regions again base on feature points. The feature points serve a dual purpose: correspondence for tracking if text has been detected in the region and cues of candidate regions for text detection. The framework further verifies candidate text regions using more sophisticated features. Once a text region is confirmed, the tracking algorithm will continuously track the region. The text region grows as more text around it is detected from frame to frame. Experimental results have demonstrated the feasibility of the proposed framework in incrementally detecting text on road signs over the time from video sequences captured from a moving vehicle.

This research was partially supported by the CMU/CM-CRL.

Keywords: incremental detection, text detection, traffic sign, vertical plane model, sign tracking, text tracking

1. Introduction

Automatic understanding of road signs is an essential task for autonomous and intelligent vehicles. It could help to keep a driver aware of the traffic situation by highlighting and recording signs that have been passed. The system could also read out the text on road signs with a synthesized voice, which is especially useful for drivers with weak visual acuity.

The previous research on road sign detection and recognition is limited to symbol recognition [7, 12]. Researchers developed systems for detecting and recognizing symbols, such as "stop" and "curve," etc. These systems were based on two different approaches: (1) segmentation through color thresholding, region detection and shape analysis; (2) segmentation through the border detection in a black and white image and their analysis. In shape-based recognition, it seems that Gavrila's methods [10, 11] are superior to other approaches for the implementation of a real-time application. Other methods of road sign detection include color detection [17], color then shape [16, 19], simulated annealing [1] and neural networks [21]. In this paper, we are interested in automatically locating road signs from video input and detecting text on road signs. Unlike the previous research, we are interested in not only recognizing shapes of road signs but also understanding text on road signs. Figure 1 shows four examples of road signs. Obviously, we face many challenges in detecting text from these road signs as in other object recognition tasks:

- Lighting conditions are uncontrollable and changeable because of time and weather variations.
- Background and foreground are very complex.
- Text on road signs varies in font, size and color.
- Video images are low resolution and noisy.

Figure 1 Examples of road signs

In order to address these challenges, we propose a robust and reliable framework that can automatically and incrementally detect text on the road signs in video. Video sequence contains a large amount of temporal redundant information of motion of the camera and objects in the scene. We can take advantage of the redundant information and incrementally detect the text from frame to frame. Different from most existing sign detection approaches, by employing the cues from the tracking scheme, the proposed framework integrates detection into tracking mechanism. First, the framework finds a set of discriminative features for current video frame. These points will serve as the input for

the second step as well as tracking features. Next, the framework uses two criteria to detect the possible candidate road sign planes from the set of feature points. The two criteria are color information and vertical plane model. The first criterion was widely used in text detection algorithms, particularly in the road sign text detection. The vertical plane model is based on the fact that 3D geometric relationship can be recovered from 2D data. After this step, the system has a set of possible candidate road sign areas. This framework applies an edge-based text detection algorithm to these candidate sign areas. The selected features in these areas provide cues of candidate text regions for further detection. The framework further verifies candidate text regions using multi-scale edges. Once a text region is confirmed, the tracking algorithm will continuously track the region. The framework repeats this process, and locates road signs and detects text on these signs over the time. We have performed extensive experiments. Experimental results indicate that the proposed approach can incrementally detect text on road signs from video sequences captured from a video camera mounted on a moving vehicle.

The rest of this paper is organized as follows: Section 2 describes the new framework in detail. Section 3 discusses the vertical plane model. Section 4 introduces system implementation and experimental results. Section 5 gives the conclusion and future work.

2. Incremental Text Detection

Information retrieval has activated research in automatic detection and recognition of text from video (video OCR) [3, 4, 8, 13, 20, 22]. Text in video can be classified into two categories: *graphic text* and *scene text*. Graphic text is added to the video after the video is captured by visual recording device. Scene text exists in a natural environment, and is directly captured by a video camera. That is, scene text is part of objects on which it appears. Examples of scene text include road signs, advertisement board, direction signs, text on costume and consumables. Some early research problems include text detection from general backgrounds, and recognition of text on particular objects like containers and license plates. In recent years, growing attention has been focused on sign text detection and translation from photography and video. The early work mainly focused on the prototype ideas and required human interaction to select the sign area in the image. Recent research attempts have moved toward automatic sign detection and recognition [4, 8].

In this research, we are interested in automatic detection of text on road signs from live videos. This is a scene text detection task. Area-based and edge-based methods have been widely used for detecting text in an image. Area based method aims to analyze certain features in an area, such as texture and color [6]. Some transforms, such as Discrete Cosine Transform (DCT), Gabor filtering, and Gaussian filtering, are used for area analysis. Although these area-based methods have different advantages, they share a common problem, i.e., the sensitivity to lighting and scale variations. Edge based method mainly relies on edge features that are relatively more stable to the above problem. Some noise filtering schemes should be applied to avoid adding extra edges. This method has been found more suitable for text detection from natural scenes [4, 8, 13, 14].

Li et. al [13] presented an approach detecting text from video by detecting texts periodically while tracking in the rest of the period. The approach used a hybrid wavelet/neural network classifier to segment text regions in the video frames. Once the text is detected, a multi-resolution sum of squared differences based tracking method is

applied to track the detected text. However, tracking and detection were separated modules in the system, and they were not integrated to facilitate each other over the time.

In order to efficiently detect text on road signs, we propose to combine tracking and detection mechanisms into a same framework. The attractiveness of this framework is that it can fully utilize the temporal information, and incrementally detects text on road sign planes from frame to frame.

In order to catch human attentions, road signs are designed with the following properties:

- Text on road signs is designed with high contrast to its background.
- Most road traffic signs appear on vertical planes.
- Foreground/background colors of a road sign are not randomly distributed. They are distinguishable from the surrounding environment.

The basic idea of the proposed approach is to effectively use these constraints and efficiently integrates the detection mechanism into the tracking process. As illustrated in Figure 2, the proposed framework consists of five steps.

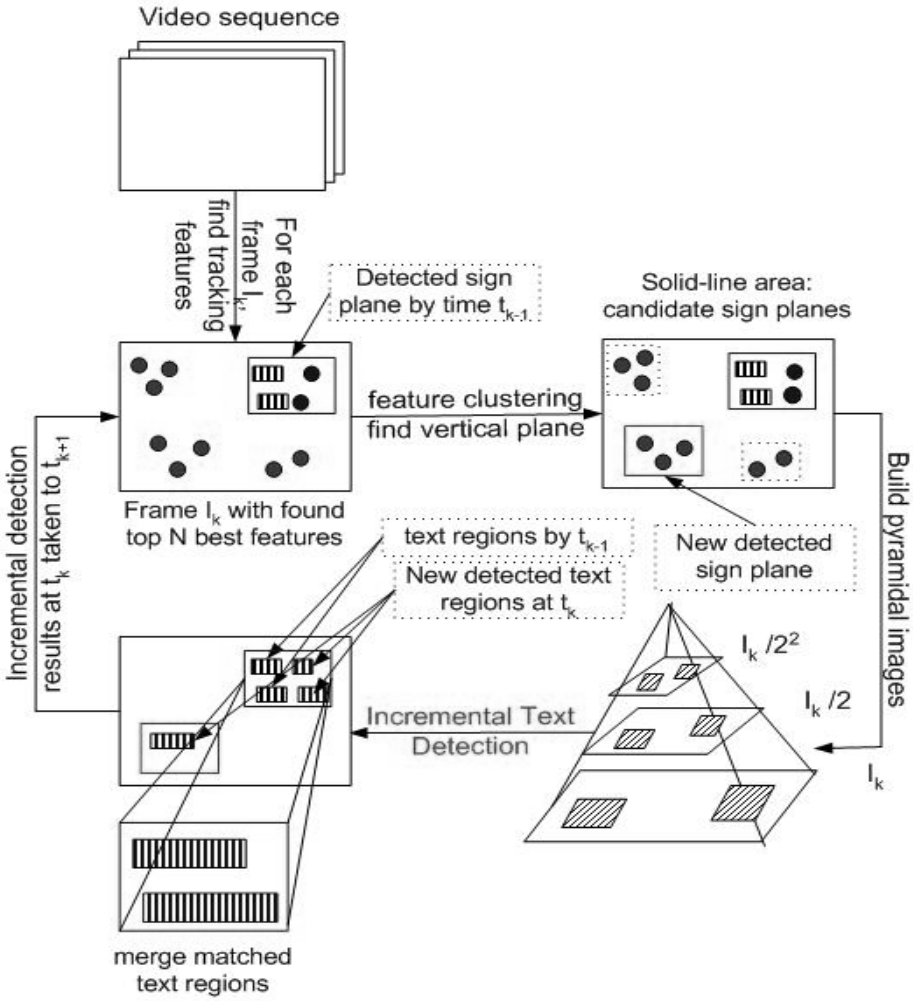


Figure 2 Incremental detection flow at time t_k

Algorithm

1. Select features. For the current video frame, good features in non-text areas are selected using some discriminative criteria (the whole image if non-text area is empty). These features will be used in the following steps and some of them will also be tracked;
2. Find road sign plane candidates. Use two criteria to detect possible candidates of road sign planes from the cues of features found in the current frame and tracked features from previous frames. The two criteria are color information and vertical plane property of road signs;
3. Build pyramidal multi-scale images. The constructed multi-scale images of the current frame will serve as input for the following incremental text detection;
4. Incrementally detect text. Both new candidates and previously detected sign planes will be examined by an edge-based text detection method on the pyramidal multi-resolution images. Detection results from different levels will be combined to detected text regions of original scale. We can obtain the detection results by merging matched text regions.
5. Feedback detection results. The detected text regions and features within them will be tracked. Their information will be taken into account in the analysis of the next iteration. Read next frame, and go back to Step 1.

The new algorithm works in an iterative manner that enables the algorithm to detect text incrementally over the time from a video sequence.

In the proposed framework, Step 1 & 2 attempt to locate road signs from a video frame. Step 3 & 4 detect new text regions in candidates of road signs and combine them with previous detected text regions. Step 5 feedbacks current most complete partial detection results to the analysis of next frame. It also tracks the detected text regions, sign planes and features within them when the next frame comes. Over the time, text on road sign planes will be detected and tracked incrementally until the road sign planes disappear from the scene. We describe the framework in more detail below, except that the vertical plane model will be introduced in Section 3.

Step 1: The key idea of this framework is to embed text detection into a tracking process. Robust tracking requires good feature points. Accurate detection of text also requires good features. The number of features is decided empirically to balance the detection rate and computation efficiency. An example of feature selection will be shown in Section 4.

Step 2: Next, obtained feature points are clustered by using their coordinates as features. Color segmentation in certain color spaces is then performed to get initial clusters of feature points. The vertical plane model will be used to extract possible road sign planes from the initial feature clusters.

Step 3: Pyramidal construction of an image is widely used in many tracking and text detection algorithms. It is a bottom-up process that build (L) level image from (L-1) level image. The original frame image is considered as 0 level image. This multi-scale approach attempts to solve the problem of different sizes of text. The small text can be detected at the lower levels while the large one is deemed to be background. On the other hand, the large texts can be found at higher levels while the small ones will be overlooked. By this strategy, the algorithm can detect text with different sizes by combining the detection results from different pyramidal levels of the image.

Step 4: The algorithm uses edge-based text detection module that consists of coarse detection and structure analysis.

We use a difference of Gaussian edge detector to obtain the edge set. We then compute size, intensity, mean, and variance of the edge set within the surrounding rectangle. Using some feature criteria, we can remove some edge patches from the set, and the rest remains for further consideration. Next, merge adjoining edge patches with similar properties and update properties of combined edge patches. Since texts in the same context share common color properties, we can use them to analyze the structure of the text, and further refine detection results. In this work, a Gaussian Mixture Model (GMM) is used to model color distributions of the foreground and background of each region.

$$g(c) = \beta G_f(\mu_f, \theta_f) + (1-\beta) G_b(\mu_b, \theta_b), 0 \le \beta \le 1 \quad (2.1)$$

where G_f, G_b are the color distributions of the foreground and background respectively. β indicates the complexity of the text, $\|\mu_f - \mu_b\|$ shows the contrast for a color space invariant to the lighting condition, and θ_f, θ_b provides the information of the text font style. So, each text can be represented with $(\beta, \mu_f, \mu_b, \theta_f, \theta_b)$. After new text regions are detected in the current frame, they will be merged along the string direction with detected text regions from the previous frames. Further, the new obtained text region will be tracked. Old text regions are removed from the tracking list if they have been merged.

Step 5: The corners of all detected text regions and sign planes are tracked by the feature tracker over the video sequence. However, tracking performance is affected by the problem that the corresponding features may drift. Thus, some constrains are used to reject outlier matched points.

a) The velocity (optical flow) of each features attempts to be consistent with those of its neighbor features;
b) Neighboring features should stay close to maintain the spatial cohesion, but collision should be avoided.

Distance and brightness change criteria can also be considered to reject outlier matches. After applying above constrains, we find that the tracking module works very well on real videos except when there are sudden lightness variations, severe obstruction before road signs or disappearance of the signs. Alternatively, we can apply the epipolar constraint to reject outlier matches. The epipolar constraint states that if P_1, P_2 are the coordinates of a same spatial point in the real world in the two frames, they must satisfy the following equation $P_2^T \cdot F \cdot P_1 = 0$, where F is the fundamental matrix that represents the epipolar geometry between two images. This equation means that point P_2 must pass through the epipolar line defined by $F \cdot P_1$ in the second frame image and vice versa.

3. A Vertical Plane Model

3.1.Model formulation

In Step 1 of the framework, we obtain clusters of feature points using color models. In this section, we will select candidate road sign planes by verifying whether those feature-point clusters satisfy the vertical plane model. The goal of this step is to provide candidate sign regions for incremental text detection. Another benefit of this strategy is

that it further narrows down the search space of text detection thus greatly improves the efficiency of the framework.

The basic idea is that most road sign planes exist as vertical planes in the real world. Since we can obtain spatial correspondence of feature points in two adjacent frames from the tracking algorithm, we can recover the normal of the candidate planes. Using the vertical property of sign planes, we can filter out non-sign planes from the feature- point clusters.

In our approach, we need, at least, 3 feature points to verify a candidate. Tracking algorithm provides the spatial information of every feature over the time. We then choose three feature points that are not in one line to check if their constructing plane is a vertical plane in 3D world or not. We use the following example to illustrate the idea, Figure 3 shows two adjacent video frames F_0, F_1 and their associated two camera coordinate systems at times t_0, t_1. Camera focal length is f, and camera moves d during the intermediate period. Camera coordinate system at t_0, $O^0X^0Y^0Z^0$, is the basic coordinate system. It uses the vectors (1 0 0), (0 1 0) and (0 0 1) as its axis. Feature points P_1, P_2, P_3 are represented in the basic coordinate system, $P_1:(x_1, y_1, z_1)$, $P_2:(x_2, y_2, z_2)$ and $P_3:(x_3, y_3, z_3)$ $o^0x^0y^0, o^1x^1y^1$ are image coordinate systems at times t_0, t_1. As $(t_1 - t_0)$ is very small for a real-time video sequence, we assume that the vehicle moves along the camera optical axis O^0Z^0 at t_0, t_1.

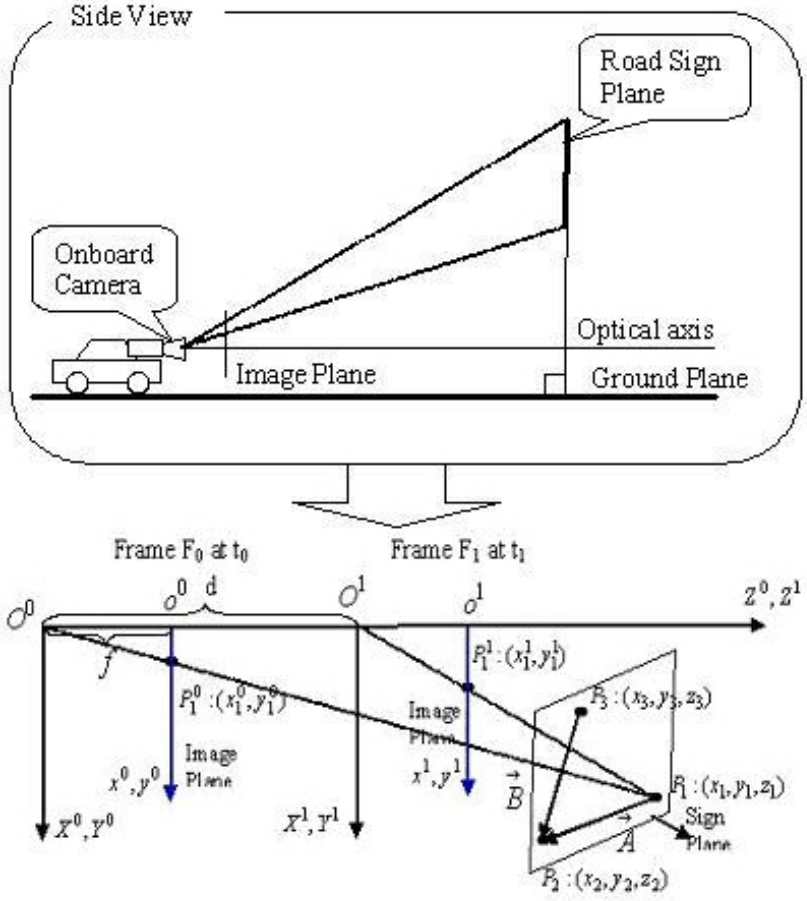


Figure 3. The basic geometry between two snaps.

The projection that maps a point $P_1 : (x_1, y_1, z_1)$ from camera coordinate system onto two points $P_1^0 : (x_1^0, y_1^0)$, and $P_1^1 : (x_1^1, y_1^1)$ in the two image coordinate systems at times t_0, t_1 is as follows

$$\begin{pmatrix} x_1^0 \\ y_1^0 \end{pmatrix} = \begin{pmatrix} f & 0 \\ 0 & f \end{pmatrix} \begin{pmatrix} x_1 / z_1 \\ y_1 / z_1 \end{pmatrix} \tag{3.1}$$

$$\begin{pmatrix} x_1^1 \\ y_1^1 \end{pmatrix} = \begin{pmatrix} f & 0 \\ 0 & f \end{pmatrix} \begin{pmatrix} x_1 /(z_1 - d) \\ y_1 /(z_1 - d) \end{pmatrix} \tag{3.2}$$

Based on above equations (3.1) and (3.2), we can obtain the following estimation of P_1 coordinates.

$$\begin{pmatrix} x_1 \\ y_1 \\ z_1 \end{pmatrix} = \begin{pmatrix} d \cdot x_1^0 \cdot x_1^1 / f \cdot (x_1^1 - x_1^0) \\ d \cdot y_1^0 \cdot y_1^1 / f \cdot (y_1^1 - y_1^0) \\ d \cdot x_1^1 /(x_1^1 - x_1^0) \end{pmatrix} \tag{3.3}$$

Similarly, we can get estimation of P_2 and P_3 coordinates.

Let that

$$M_{kx} = \frac{x_k^1}{x_k^1 - x_k^0} \tag{3.4}$$

$$M_{ky} = \frac{y_k^1}{y_k^1 - y_k^0} \tag{3.5}$$

where $k = 1,2,3,\ldots$.

In order to obtain the normal of a candidate plane, we can find the representations of vectors $\vec{A}$ and $\vec{B}$ as:

$$\vec{A} : \begin{pmatrix} x_1 - x_2 \\ y_1 - y_2 \\ z_1 - z_2 \end{pmatrix} = \begin{pmatrix} d \cdot (\frac{x_1^0}{f} \cdot M_{1x} - \frac{x_2^0}{f} \cdot M_{2x}) \\ d \cdot (\frac{y_1^0}{f} \cdot M_{1y} - \frac{y_2^0}{f} \cdot M_{2y}) \\ d \cdot (M_{1x} - M_{2x}) \end{pmatrix}, \tag{3.6}$$

$$\vec{B} : \begin{pmatrix} x_3 - x_2 \\ y_3 - y_2 \\ z_3 - z_2 \end{pmatrix} = \begin{pmatrix} d \cdot (\frac{x_3^0}{f} \cdot M_{3x} - \cdot \frac{x_2^0}{f} \cdot M_{2x}) \\ d \cdot (\frac{y_3^0}{f} \cdot M_{3y} - \frac{y_2^0}{f} \cdot M_{2y}) \\ d \cdot (M_{3x} - M_{2x}) \end{pmatrix}. \tag{3.7}$$

Then, we can obtain the normal of the candidate plane as

$$\vec{N} = \vec{A} \otimes \vec{B} = \begin{pmatrix} X \\ Y \\ Z \end{pmatrix} \tag{3.8}$$

where

$$X = \frac{d^2}{f} \cdot \begin{vmatrix} y_1^0 \cdot M_{1y} - y_2^0 \cdot M_{2y} & M_1 - M_2 \\ y_3^0 \cdot M_{3y} - y_2^0 \cdot M_{2y} & M_3 - M_2 \end{vmatrix} \tag{3.9}$$

$$Y = \frac{d^2}{f} \cdot \begin{vmatrix} x_1^0 \cdot M_{1x} - x_2^0 \cdot M_{2x} & M_1 - M_2 \\ x_3^0 \cdot M_{3x} - x_2^0 \cdot M_{2x} & M_3 - M_2 \end{vmatrix} \tag{3.10}$$

$$Z = \frac{d^2}{f^2} \cdot \begin{vmatrix} x_1^0 \cdot M_{1x} - x_2^0 \cdot M_{2x} & y_1^0 \cdot M_{1y} - y_2^0 \cdot M_{2y} \\ x_3^0 \cdot M_{3x} - x_2^0 \cdot M_{2x} & y_3^0 \cdot M_{3y} - y_2^0 \cdot M_{2y} \end{vmatrix} \tag{3.11}$$

In order to verify whether the plane of P_1, P_2, P_3 is on a vertical plane in the 3D world, the equations (3.9) – (3.11) can be used to recover the normal of the constructing plane of any three points P_1, P_2, P_3 from a feature-point cluster. We then measure the ratio of the X component to the length of vector *N*.

$$P_j = \left| X_j \right| / \left\| N_j \right\|, \; j = 1, 2, ,...J \tag{3.12}$$

where J is the number of recovered normal vectors from one feature-point cluster. A proper averaging scheme can be applied to all achieved normal vectors to minimize individual errors as shown in equation (3.13).

$$P = \sqrt{\sum_{j=1}^{J} P_j^2} \; . \tag{3.13}$$

3.2.Model Sensitivity Analysis

Equations (3.8)-(3.12) indicate that the accuracy of the normal of the verified plane highly depends on the accuracy of calibrated focal length f. From the equations (3.9) – (3.11), we can derive

$$P_j = \frac{\left| X_j \right|}{\left\| N_j \right\|} = \frac{| C_{jX} |}{\sqrt{C_{jX}^2 + C_{jY}^2 + \frac{1}{f^2} \cdot C_{jZ}^2}}, \tag{3.14}$$

where C_{jX}, C_{jY}, C_{jZ} are second components of (3.9) - (3.11) respectively. From the above equation, we know that P_j is the function of the camera focal length f, and the perturbation in P_j produced by perturbations in f is :

$$\frac{\partial P_j}{\partial f} = C_{jX} \cdot C_{jZ}^2 \cdot \left(C_{jX}^2 + C_{jY}^2 + \frac{1}{f^2} \cdot C_{jZ}^2 \right)^{-1.5} \cdot f^{-3} \tag{3.15}$$

then we can obtain

$$\frac{\Delta P_j}{P_j} = \frac{C_{jZ}^2}{\left(C_{jX}^2 + C_{jY}^2\right)\cdot f^2 + C_{jZ}^2} \cdot \frac{\Delta f}{f} \,. \tag{3.16}$$

From equation (3.16) we can observe that, the accuracy of normal of the verified plane linear depends on the accuracy of the calibrated focal length.

4. System and Experiments

We have conducted extensive experiments to validate our framework on real video streams of natural scenes. This section provides the details of the system implementation, conducted experiments, and results.

4.1.Technical description of the prototype system

The prototype system is implemented and evaluated on a PC with Intel Pentium 4 CPU @1.8 GHz and 1G memory running Windows XP. The evaluation video was captured from a SONY digital video camera mounted on a minivan. Intrinsic parameters of the camera were calibrated using the method proposed by Zhang [23]. The video frame size is 640*480.

Signs are designed for human to see easily at a distance. In most of cases, road signs have following properties: 1. Text is designed with high contrast to its background color. 2. Text on the same road sign has almost the same foreground and background patterns. These properties enable some points (corners) on sign planes could also be good tracking features. Thus, the feature selection is implemented using the algorithm in [18]. Better and more suitable text detection on road signs could be studied and evaluated. Shi-Tomasi algorithm shows good performance in this work.

The more number of selected features, the more accurate is the later detection algorithm, while the more computation power is needed. We tried different numbers of features in our system, and 50 were found to fairly balance the detection rate and computation efficiency.

To enable the system to detect new appeared road sign planes over the time, new features are selected in non-text regions and added to the system. Moreover, combined with rejecting outlier matches strategy mentioned in section 2, updating good features frame by frame alleviates the feature tracker drifting problem and improves the robustness and accuracy of the tracker.

Faster feature selection and tracking optimization [9] can be applied further to improve the running frequency of the system.

In order to extract road sign planes, obtained feature points are clustered by using their coordinates in the image as features. Color segmentation in certain color spaces is then performed to get initial clusters of feature points. In this research, we convert the camera RGB color space to the HSI color space, and normalize HSI within the range of [0, 255]. The vertical plane model will be then used to extract possible road sign planes from the initial feature clusters.

Details of the incremental detection of text have been introduced in section 2. Here we will mention a little more about text structure analysis. Figure 4 shows three situations in which partial text has been detected. Since English words are in horizontal direction in most cases, the text structure is analyzed horizontally. In the first situation, if two text

regions with similar height have the similar roof vertical position, they will be merged to one text region. Second case shows that two text regions with different height, while the same roof position, they will also be merged and the new height is decided by the previous big size region. However, for the third case, no merging action is performed because the left region can be better focused if it is separated with the right two regions. In the Step 3 of the algorithm, the pyramid depth is set to be 3. That means, the highest level image is one fourth of the original image size.

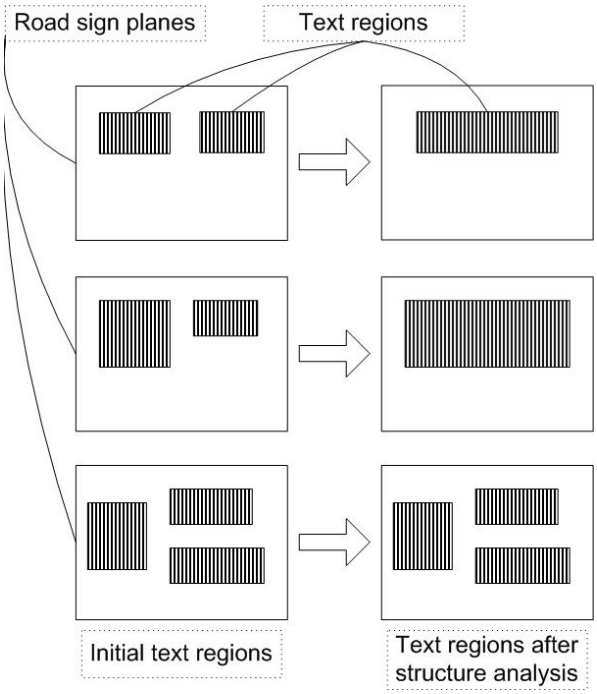


Figure 4. Text Structure Analysis

A pyramidal implementation of the Lucas Kanade Feature Tracker [2] is used to track the detected text areas. The pyramidal images come from Step 3. The search window of the optical flow is 10*10. Other trackers can also be plugged into the proposed framework easily.

4.2. Experimental Results

We have evaluated the proposed framework through experiments on our traffic sign video database. The database consists of 3 hours of various signs' videos, including highway signs, roadway signs, and other types of signs. We use an example of a highway sign to show the whole detection process in Figure 6 and the detection results of some other signs in Figure 7. Tables 1 & 2 will summarize the detection results of different types of road signs under different conditions.

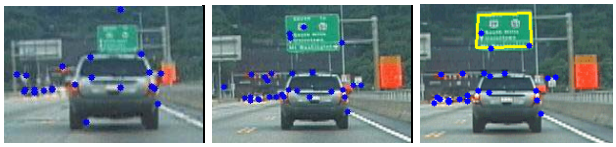
Feature selection in the first step was based on a discriminative criterion [18]. Most corner features can be extracted from the frame, including ones on the road sign planes. We cluster feature points into regions and use color models to filter out some non-sign regions. In order to reduce risk of removing real sign regions from this step, we set a very low threshold. This step can filter out majority of non-sign regions. We further verify the remaining regions using vertical plane models. A combination of color and geometric

information can do a very good job to reduce false detection. Figure 5 shows an example of text detection with/without preprocess. With the filters, the system could detect text regions correctly (Figure 5(a)). The system, however, falsely detected the frames as text in the case of no preprocessing (Figure 5(b)).

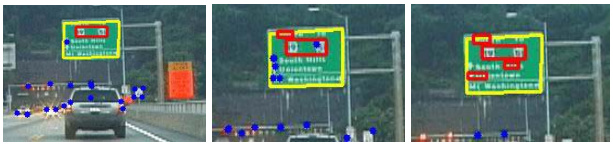
(a) **(b)**

Figure 5. An example of text detection with/without preprocessing

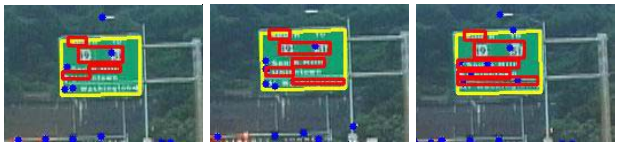
Figure 6 illustrates the process of incremental detection of text on a road sign. During the initial few frames of the video, no features points are found on the road sign planes (Figure 6(a)). On the frame of Figure 6(b), some feature points appeared on the road sign. Next, the system classified the region as a possible road sign plane and marked with yellow boundary on the image (Figure 6(c)). In the next few frames (Figure 6(d)), partial texts were detected on the road sign plane frame by frame. Some partial detected text regions were merged based on structure analysis, as shown in Figure 6(e). Figure 6(f) shows that all detected text regions are tracked over the time. Finally, all texts on the road sign are correctly detected (Figure 6(g)). A demo video sequence of Figure 6 has been attached to this submission.



(a) Feature selection (b) Features on sign (c) Find sign plane



(d) Initial text detection



(e) Incremental text detection

(f) Track detected text (g) Incrementally detect all texts

Figure 6. An illustration of incremental text detection

In the evaluation process, we noticed that, as shown in Figure 6 (a)-(d), there was a detour sign in the right side of the image. This detour sign had not been detected by our system. The reason was that the color of text on the detour sign was closed to the background of the sign. Since no feature points appeared on the sign, system simply ignored it. This shows the dependency property of the proposed framework. Each step of the algorithm contributes to the final detection result. Increasing the number of feature points can potentially make the system find this detour sign region. But text detection may still have problems because even a human had difficulty detecting the text on the sign from the video, when we evaluated it. To solve this problem, the system needs a better video camera.

Figure 7 shows more detection results from our road sign video database. Good recall and precision can be observed from these results. We evaluated the prototype system using several video sequences from our road sign database captured from a moving vehicle. The sequences were sampled at 15 frames per second. The videos are categorized based on different lightness conditions, e.g., sunny, cloudy and dusk. Table 1 shows the road sign detection performance. It is shown that performance is good in sunny and cloudy while performance is poor in the dusk. Table 2 shows the overall text detection performance.

Figure 7 More detection results.

Table 1. Road sign detection performance

	Sunny	Cloudy	Dusk
Total # of road signs in video	76	46	23
Detected	72	41	11

Table 2 Text detection performance

	Sunny	Cloudy	Dusk
Total # of text regions	315	197	93
Fully detected	227	115	31
Partially detected	53	39	7

Automatic detection of text on road signs is a challenging real problem. Many issues are associated with the problem, such as the performance impact of the different parts of the system, poor performance in low light conditions, sensitivity analysis of the algorithm performance, and comparison with other detection algorithms, etc. Our views on the above questions are as follows. First, good text detection rate relies on the quality of

features selected in the first step and the robustness of the vertical plane model applied in the second step of the framework. An undetected example was shown in Figure 6 to illustrate the dependency relationship. Second, we are working on a challenging problem and could not solve all the issues in one paper. We will develop new algorithms to address this problem in the future. The new algorithms, again, can be easily plugged into the proposed framework. Third, the system we built didn't require any manual setting of thresholds from video to video. All the thresholds are preset from the training data. The performance is relatively stable to different video streams under a reasonable resolution. Lastly, with regard to the comparison with other published text detection algorithms, we are presenting a new framework to solve a different problem, so we are not aware of any other algorithms that can solve the exactly same problem.

5. Conclusions

Large amounts of information are embedded in natural scenes. Signs are good examples of objects in natural environments that have rich information content. Detection of text on road signs has many applications in human computer interaction and robotics. Yet it poses new challenges to computer vision community. In this paper, we have proposed a new framework for incrementally detecting text on road signs. The proposed framework makes two major contributions. First, the framework efficiently embeds tracking and detection mechanisms into the same framework. Different feature selection methods, tracking mechanism, text detection approaches can be easily plugged into the framework. We have developed a prototype system to demonstrate the concept. Second, the framework has provided a novel way to integrate text detection from color, 3D vertical plane detection, and texture cues into tracking scheme. The novelty of the proposed work lies in the concept of incremental detection framework to detect text from video stream. In fact, we can plug in different technologies into this framework. In this paper, in order to demonstrate the feasibility of the new framework, we have embedded some efficient and effective existing technologies into it. We have no intention to claim contributions in extension of these algorithms or combinations of them. Experiments and evaluations have indicated the feasibility and reliability of the framework. The images used in the experiments consist of most highway images. However, in some situations sign detection is relatively more difficult as the signs are not clearly visible due to various reasons. For example, in more complex situations, the vertical plane constraint does not hold (e.g., twisted sign planes) or occlusion makes the sign detection difficult. We aim to solve these interesting problems in the future work.

6. References

[1] M. Betke and N.C. Makris, Fast object recognition in noisy images using simulated annealing, Proc. of ICCV, pp.523-530, 1995.

[2] D. Chen, H. Bourlard, J.-P. Thiran, Text identification in complex background using SVM, Proc. of CVPR, Vol. 2, pp. 621-626, 2001.

[3] X. Chen, J. Yang, J. Zhang, A. Waibel, Automatic detection of signs with affine transformation, Proc. of WACV 2002, pp. 32-36.

[4] Y. Cheng, Mean shift, Mode Seeking, and Clustering, IEEE Trans. on PAMI, 17(8), pp.790-799, 1995

[5] D. Comaniciu and P. Meer, Mean Shift: A Robust Approach Toward Feature Space Analysis, IEEE Trans. on PAMI, 24(5), pp.603-619, 2002

[6] C.-Y. Fang, C-.S. Fuh, S.-W. Chen, P.-S. Yen, A road sign recognition system based on dynamic visual model, Proc. of CVPR, pp. 750-755, 2003.

[7] J. Gao and J. Yang, An adaptive algorithm for text detection from natural scenes, Proceedings of CVPR 2001, Vol. 2, pp. 84-89.

[8] S. Ghiasi, K. Nguyen and M. Sarrafzadeh, Profiling Accuracy-Latency Characteristics of Collaborative Object Tracking Applications, Proceedings of International Conference on Parallel and Distributed Computing and Systems, November 2003.

[9] D. M. Gavrila, "Multi-feature hierarchical template matching using distance transforms," In Proc. of the ICPR, pp. 439–444, 1998.

[10] D. M. Gavrila and V. Philomin, Real-time Object Detection for Smart Vehicles, Proc. of ICCV, pp. 87-93, Kerkyra, Greece, 1999.

[11] D. M. Gavrila, U. Franke, S. Görzig and C. Wöhler, Real-time Vision for Intelligent Vehicles, IEEE Instrumentation and Measurement Magazine, 4(2), pp.22-27, 2001.

[12] H. Li, D. Doermann, O. Kia, Automatic text detection and tracking in digital video, IEEE Trans. on IP, 9(1), pp. 147-156, 2000.

[13] R. Lienhart and A. Wernicke, Localizing and segmenting text in images and videos, IEEE Trans. on CSVT, 12(4), pp.256-268, 2002.

[14] B. D. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision, Proc. of IJCAI, pp. 674–679, 1981.

[15] J. Miura, T. Kanda, Y. Shirai, An active vision system for real-time traffic sign recognition, Proc. of IEEE Intelligent Transportation Systems, pp. 52-57, 2000.

[16] G. Piccioli, E. De Micheli, P. Parodi, M. Campani, Robust method for road sign detection and recognition, Image and Vision Computing, vol. 14, pp. 109–223, 1996.

[17] J. Shi and C. Tomasi, Good Features to Track, Proc. of CVPR, pp.593-600, 1994

[18] G. Salgian and D.H. Ballard, Visual routines for autonomous driving, Proc. of ICCV, pp. 876-882, 1998.

[19] T. Sato, T. Kanade, E.K. Hughes, and M.A. Smith. Video OCR for digital news archives. IEEE Int. Workshop on Content-Based Access of Image and Video Database, pp. 52-60, 1998.

[20] S. Vitabile, A. Gentile, and F. Sorbello, A neural network based automatic road signs recognizer, Proc. of IJCNN'02, Vol. 3, pp. 2315-2320, 2002.

[21] D. Zhang and S. Chang, A Bayesian framework for fusing multiple word knowledge models in videotext recognition, Proceedings of CVPR, Vol. 2, pp.528-533, 2003.

[22] Z. Zhang. A Flexible new technique for camera calibration. IEEE Trans. on PAMI, 22(11):1330-1334, 2000.