Incremental Detection of Text on Road Signs from Video with Application to a Driving Assistant System

Wen Wu[1] Xilin Chen[2] Jie Yang[1,2]

[1]Language Technologies Institute, School of Computer Science, Carnegie Mellon University, Pittsburgh, U.S.A.
[2]Human Computer Interaction Institute, School of Computer Science, Carnegie Mellon University, Pittsburgh, U.S.A.
{wenwu, xlchen, yang+}@cs.cmu.edu

ABSTRACT

This paper proposes a fast and robust framework for incrementally detecting text on road signs from natural scene video. The new framework makes two main contributions. First, the framework applies a Divide-and-Conquer strategy to decompose the original task into two sub-tasks, that is, localization of road signs and detection of text. The algorithms for the two sub-tasks are smoothly incorporated into a unified framework through a real time tracking algorithm. Second, the framework provides a novel way for text detection from video by integrating 2D features in each video frame (e.g., color, edges, texture) with 3D information available in a video sequence (e.g., object structure). The feasibility of the proposed framework has been evaluated on the video sequences captured from a moving vehicle. The new framework can be applied to a driving assistant system and other tasks of text detection from video.

Categories and Subject Descriptors

I.2.10 [**Vision and Scene Understanding**]: Video analysis; I.4.8 [**Scene Analysis**]: Color, Motion, Object recognition, Tracking.

General Terms

Algorithms, Design, Experimentation, Performance.

Keywords

Incremental text detection, road sign, natural scene video, driving assistant system.

1. INTRODUCTION

Automatic detection of text from video is an essential task for many multimedia applications such as video indexing, video understanding, and content-based video retrieval. Extensive research efforts have been directed to the detection, segmentation, and recognition of text from still images and video [1, 2, 7, 9, 10, 12, 18, 20]. In this paper, we focus on the task of automatically detecting text on road signs with application to a driving assistant system. Text on road signs carries much useful information necessary for driving – it provides information for navigation, describes the current traffic situation, defines right-of-way warnings about potential risks, and permits or prohibits certain directions. Automatic detection of text on road signs can help to keep a driver aware of the traffic situation and surrounding environments by highlighting and recalling signs that are ahead and/or have been passed [8]. The system can also read out text on road signs with a synthesized voice, which is especially useful for elderly drivers with weak visual acuity. Such a multimedia system can reduce driver's cognitive load and enhance safety in driving. Furthermore, it can be combined with other driving navigation and protection devices, e.g., an electric map tool.

There are two essential requirements of the proposed framework to improve the safety and efficiency of driving: 1) detecting text on road signs in real-time and 2) achieving high detection accuracy with a low false hit rate. The application scenario is that a video camera is mounted on a moving vehicle to capture the scene in the front of the vehicle. The system attempts to detect text on road signs from video input and assist the driver to maneuver in traffic. Correctly detecting text on road signs imposes many challenges. First, video images are relatively low resolution and noisy. Both background and foreground of a road sign can be very complex and frequently change in video. Lighting conditions are uncontrollable due to time and weather variations. Second, appearance of text can vary due to many different factors, e.g., font, size and color. Also, text can move fast in video and be blurred from motion or occluded by other objects. Third, text can be distorted by the slant, tilt, and shape of signs. In addition to the horizontal left-to-right orientation, other orientations include vertical, circularly wrapped around another object, and even mixed orientations within the same text area.

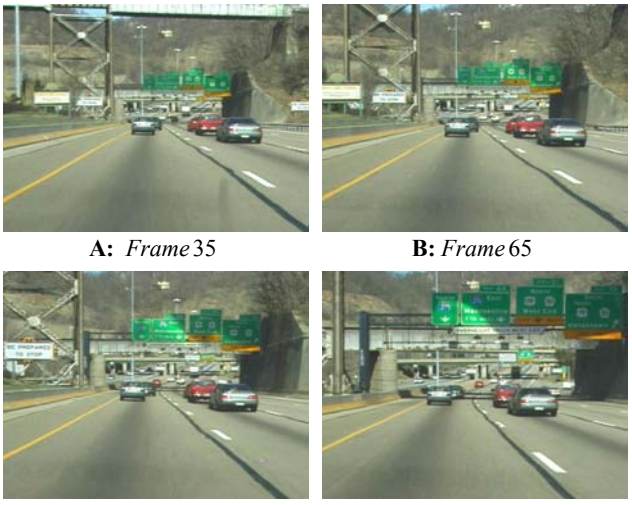
In order to address the above difficulties, we propose a novel framework that can incrementally detect text on road signs from video. The proposed framework takes full advantage of spatio-temporal information in video and fuses partial information for detecting text from frame to frame. The framework employs a two step strategy: 1) locate road signs before detecting text via a plane classification model by using features like discriminative points and color; and 2) detect text within the candidate road sign areas and then fuse the detection results with the help of a feature-based tracker. The concrete steps of the framework are as follows. A set of discriminative points are found for each video frame. Then these selected points are clustered based on local region analysis. Next, a vertical plane criterion is applied to verify road sign areas in video by recovering the orientations of possible planes. Through the sign localization step, thereby, the number of the false positives caused by "text-like" areas is reduced. A multi-scale text detection algorithm is further used to locate text lines within candidate road sign areas. If a text line is detected, a minimum-bounding rectangle (MBR) is fitted to cover it and previously selected points inside MBR are tracked. The

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.
MM'04, October 10–16, 2004, New York, New York, USA.
Copyright 2004 ACM 1-58113-893-8/04/0010…$5.00.

framework iterates the process for every video frame, locates road signs, detects, and tracks text on signs over the time.

Next we review some related work to this research. Based on its origin, text in video can be classified into two classes: *graphic text* (a.k.a., superimposed/ overlay text) and *scene text* [11]. Graphic text is added to the video after the video is captured, e.g., captions in news videos. Scene text exists as part of objects in a natural environment when it is directly captured by a camera, e.g., billboards, and street names on road signs. Obviously, this task is a scene text detection problem. A common assumption used by the previous research in graphic text detection from video is that text plane is perpendicular to optical axis of the camera [7, 12, 16]. Although the assumption holds in some domain data, e.g. broadcast news video, it does not necessarily hold in the task. More recently, some researchers attempted to detect scene text from still images [2, 3, 6]. Inspired by their work, we use the edge based features for text detection in this study, since they reported that edge features can better handle lighting and scale variations in natural scene images than texture features [2], which are often used for detecting text in news video [10, 12]. Myers et. al. described a full perspective transformation model to detect 3D deformed text from still images [15], however, they did not show that their model was feasible for the same task in the video domain. Some researchers introduced model-based methods for detecting and/or recognizing road signs in video, however, examples of road sign symbols, not text, were used in detection experiments, e.g., "stop", "do not enter" and "curve" signs [4].

The rest of the paper is organized as the following. Section 2 introduces the overview of the proposed framework. Section 3 presents a road sign localization algorithm using a vertical plane criterion. Section 4 discusses the detection of text by an edge-based algorithm. Section 5 describes system implementation and shows experimental results. Section 6 concludes the paper.



A: *Frame* 35 **B:** *Frame* 65

C: *Frame* 95 **D:** *Frame* 155

Figure 1. A video sequence of road signs.

2. INCREMENTAL SPATIO-TEMPORAL TEXT DETECTION

Figure 1 shows a typical example in which road signs appear in video captured from a video camera mounted on a moving minivan. We can see that a road sign normally first appears as a very small rectangle which progressively increases in size as the vehicle approaches it. Meantime, text on the sign starts to become visible. From the technical point of view, the size of the road sign is so small at the beginning of the process that it is impossible to run text recognition and even the detection on video frames, e.g., A & B in Figure 1. This is quite understandable since even a human driver cannot distinguish the text at the beginning. In contrast, it is realizable for some algorithm to detect the road sign instead of text at first. Moreover, the expensive computation prohibits running the text detection on the whole video image in a real-time application. Meanwhile, a high false hit rate is major consequence when most existing text detection algorithms are applied as many "text-like" areas can exist in every frame of natural scene video.

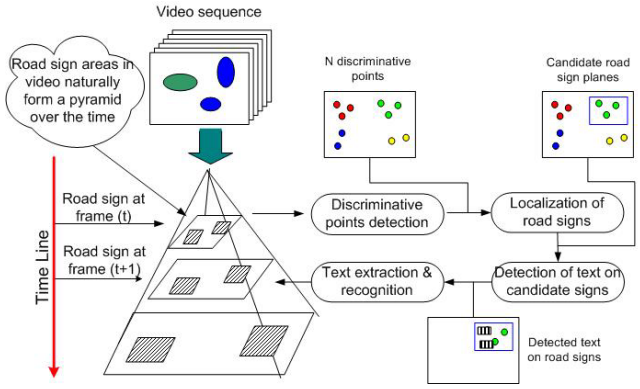


Figure 2. An illustration of the proposed framework.

To improve the efficiency of detection process while maintaining a low false hit rate in this task, we naturally employ a Divide-and-Conquer strategy to decompose the original task into two sub-tasks, that is, localizing road signs first and then detecting text. The key idea to realizing such an incremental framework is to exploit the temporal information available in video. This idea has been proved to be effective in other text detection tasks, e.g., caption detection in broadcast video [1, 10 11]. Moreover, because of government requirements on the design and placement of road signs [14], this task also has some good properties when conceiving the new framework: 1) text on the road sign is designed with high contrast to their background color; 2) text on the same road sign always has similar foreground and background patterns; 3) most road signs exist on vertical planes; and 4) certain number of color is used to design the road signs.

The proposed framework considers the whole period of appearance of a road sign in video as a pyramid of sign image patches along the time line. Figure 2 shows the architecture of the framework of which four main steps are summarized as follows.

Step 1. Discriminative points detection & clustering – detect discriminative feature points in every video frame and partite them into clusters.

Step 2. Road sign localization – select candidate road sign regions corresponding to clusters of feature points using a vertical plane criterion.

Step 3. Text detection – detect text on candidate road sign areas and track them.

Step 4. *Text extraction & recognition – extract text in candidate sign plane for recognition given a satisfactory size.*

In step 1, a number of discriminative points are selected in the current frame and they are clustered in terms of local region analysis. By computing the similarity of the points between adjacent frames, the framework finds the correspondence for each point in the next frame. Then the *Road Sign Localization* step detects candidate road sign areas from the point clusters. We use a vertical plane criterion because of the observation that most real-world road signs appear on vertical-plane objects. The correspondence information of every pair of points is the key to realize the verification using a plane classification model. Further a multi-scale text detection algorithm is performed on those candidate road sign areas, which is the *Text Detection* step. After that, parts of text on the sign are detected and MBRs are fitted to cover them. A feature-based tracking algorithm is used to track all detected areas over the time and they are merged with other newly detected text in the sequence. Finally, all detected text lines are extracted for recognition given a satisfactory resolution in *Text Extraction & Recognition* step. The recognition can be done by integrating an OCR system. Since the main contribution of this work is not about OCR, we skip the recognition part in this paper.

There are some interesting properties of the new framework. First, the number of selected points in step 1, N, balances the sign localization speed and system process rate because the more feature points the more likely the sign is located early while a large number means intensive computation. Second, spatio-temporal information is extracted and used by the framework to recover the orientation of potential planes in the 3D space. Once a point cluster is classified as a vertical plane, the text detection algorithm will be run within it. Third, the framework applies a feature-based tracker which can track a feature point in a sub-pixel level [13]. The corners of detected road sign areas and MBRs are tracked to the next frame by averaging the motions of the nearest points of each corner. There are two reasons for tracking discriminative points instead of the boundary corners directly: 1) boundary corners may not be "good" feature points to track compared to those selected points; and 2) tracking the selected points on the road sign area can relieve the problem of partial occlusion when it happens in video. This property is illustrated and discussed more detailed in Section 5. The new framework possesses two unique merits:

- By applying the two-step strategy, the first two steps can significantly narrow down the search space for the later text detection step and thus reduce the majority of false hits which occur in the case of the whole-image text detection.;
- It takes advantage of both temporal and spatial information in video for detecting text on road signs over the time.

We describe steps 1 & 2 in next section and step 3 in Section 4.

3. FINDING ROAD SIGNS

In order to differentiate road signs from other objects, we have to use properties of road signs such as color distribution and geometric constrains. The following sub-sections show how to detect discriminative points and use a vertical plane criterion for finding road signs from video.

3.1 Discriminative Points Detection

In order to recover the orientations of existent rigid planes in video, the system finds a number of N^t discriminative points at time t using the algorithm described in [17]. Such selected points are good for both discriminating objects and tracking objects in video. Since points on road signs share common color properties, we model the local region of each detected point by a Gaussian Mixture Model (GMM):

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

where G_f, G_b are the color distributions of the foreground and background respectively. Therefore, each point can be represented as a vector such as $(\beta, \mu_f, \mu_b, \theta_f, \theta_b)$. The GMM coefficients are used as features for the clustering of selected points. Here the H component of the HSI color space is used for color modeling and the *K*-means algorithm is applied for clustering.

In this stage we obtain M^t number of clusters of points, i.e., $P^t = \{c_1^t, c_2^t, ..., c_{M_t}^t\}$, where $c_i^t = [p_j^t, ..., p_k^t]$, $i = 1, ..., M^t$ is the *ith* cluster containing from the *jth* point to the *kth* point at time t. Points in the same cluster share the similar color patterns in their local regions, thus a cluster can be naturally considered as a candidate object plane for the later verification. The visual illustration of the outcome after this step is shown in Figure 2, where different-color points show that they belong to different clusters, e.g., all red points are in one cluster, the same for blue, green and yellow ones.

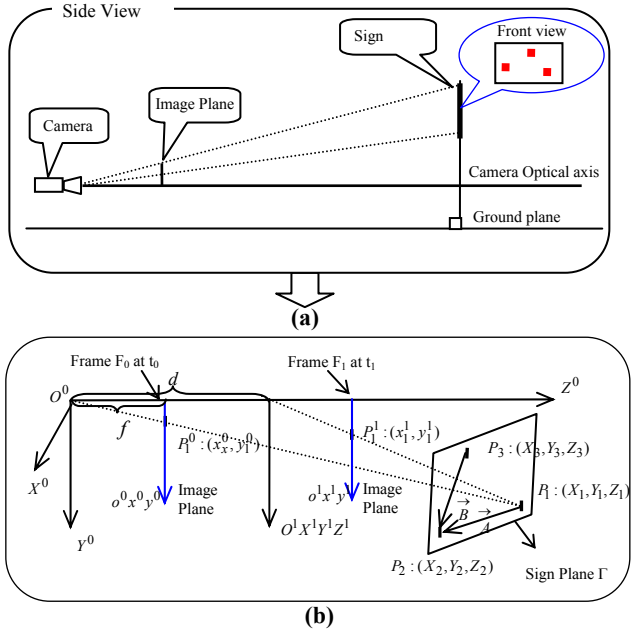


Figure 3. Spatial constrains between two successive frames.

3.2 A Plane Classification Model

We are estimating the orientation of the candidate planes (signs) given three or more points in two successive frames. Here we make two common assumptions as in other related work [5]: 1) the optical axis of the camera is roughly horizontal and the motion of the camera is going along its optical axis; and 2) text lies on planar surfaces. These two assumptions are often true in the real world setting. Particularly, in this task a camera is mounted on the vehicle and its optical axis is calibrated to be parallel to the horizontal plane of the vehicle. Figure 3 (a) shows the side view of such a scenario and (b) illustrate the spatial constraints among road sign planes, image planes and the camera between two

successive frames. We take the camera coordinate system at time t_0 as the basic coordinate system, $OXYZ$, and $o^t x^t y^t$ as the imaging coordinate system at *t*. The Z axis is the camera optical axis and the X axis is parallel to the vehicle's horizon plane. $P_1(X_1,Y_1,Z_1)$, $P_2(X_2,Y_2,Z_2)$ and $P_3(X_3,Y_3,Z_3)$ are assumed to be 3 no collinear points on a road sign plane Γ. The camera's focal length is f, and the camera moves forward the distance, d, from t_0 to t_1. $(t_1 - t_0)$ is usually small for a real-time video. The projections of P_i on $o^t x^t y^t$ at t_0 and t_1 are $p_i^{t0} : (x_i^{t0}, y_i^{t0})$ and $p_i^{t1} : (x_i^{t1}, y_i^{t1})$ $(i = 1,2,3)$, respectively. Here a feature-based tracker is used to find the correspondence of points between t_0 and t_1 [13].

Eq. (2) defines the projection between two coordinate systems. The left sides of the equations are points' coordinates in $o^t x^t y^t$ and the right sides are coordinates in $OXYZ$.

$$\begin{pmatrix} x_i^{t0} \\ y_i^{t0} \end{pmatrix} = \frac{f}{Z_i}\begin{pmatrix} X_i \\ Y_i \end{pmatrix}; \quad \begin{pmatrix} x_i^{t1} \\ y_i^{t1} \end{pmatrix} = \frac{f}{Z_i - d}\begin{pmatrix} X_i \\ Y_i \end{pmatrix} \quad i = 1,2,3. \tag{2}$$

$$P_i : \begin{pmatrix} X_i \\ Y_i \\ Z_i \end{pmatrix} = \frac{d}{f}\begin{pmatrix} x_i^{t0} \cdot x_i^{t1} / (x_i^{t1} - x_i^{t0}) \\ y_i^{t0} \cdot y_i^{t1} / (y_i^{t1} - y_i^{t0}) \\ f x_i^{t1} / (x_i^{t1} - x_i^{t0}) \end{pmatrix} \quad i = 1,2,3. \tag{3}$$

We further write down the expressions for P_i in Eq. (3). Although f and d are unknown, we will soon see that their values are not necessary to be specified in the later algorithm. We can find that

$$x_k^{t1} / \left(x_k^{t1} - x_k^{t0}\right) = y_k^{t1} / \left(y_k^{t1} - y_k^{t0}\right), \quad k = 1,2,3.$$

For the simplification in the following derivation, we define the following ratios M_k as:

$$M_k = x_k^{t1} / \left(x_k^{t1} - x_k^{t0}\right) = y_k^{t1} / \left(y_k^{t1} - y_k^{t0}\right), \; k = 1,2,3 \tag{4}$$

Figure 3 depicts that $\vec{A}$ is the vector from P_1 to P_2, and $\vec{B}$ is the vector from P_3 to P_2. Using the estimated coordinates of P_i in Eq. (3), we further obtain the estimations of $\vec{A}$ and $\vec{B}$ as:

$$\vec{A} = \frac{d}{f}\begin{pmatrix} x_1^{t0} M_1 - x_2^{t0} M_2 \\ y_1^{t0} M_1 - y_2^{t0} M_2 \\ f(M_1 - M_2) \end{pmatrix}, \quad \vec{B} = \frac{d}{f}\begin{pmatrix} x_3^{t0} M_3 - x_2^{t0} M_2 \\ y_3^{t0} M_3 - y_2^{t0} M_2 \\ f(M_3 - M_2) \end{pmatrix}. \tag{5}$$

In order to recover the orientation of the sign plane Γ, we need further know the normal vector of Γ, noted as N, which can be obtained by the cross product of $\vec{A}$ and $\vec{B}$.

$$N = \vec{A} \otimes \vec{B} = \begin{pmatrix} X_\Gamma & Y_\Gamma & Z_\Gamma \end{pmatrix}^T, \tag{6}$$

where each component of N, i.e., $X_\Gamma, Y_\Gamma, Z_\Gamma$, can be derived from Eqs.(4)&(5). Eq.(6) gives a nice way to estimate the orientation of the candidate plane by using three points' image coordinates given the spatial constrains. By taking advantage of the approximations, we can further define a model to classify planes into positive and negative categories using different criteria, e.g., vertical vs. non-vertical planes, rigid vs. non-rigid planes. Here we are interested in a vertical plane criterion.

3.3 Algorithm Description

Base on the property that most road signs are on vertical planes, the ratio of the X component to the length of N is supposed to be smaller than certain threshold. Thus, we can estimate the ratio and use it to locate vertical planes. As there are very likely more than three discriminative points in a cluster, a Maximum Likelihood Estimation (MLE) is used to estimate the mean and variance of the ratio in each cluster, shown in Eqs.(7) & (8).

$$\bar{\varphi} = \frac{1}{J}\sum_{j=1}^{J} \frac{|X_j|}{\|N_j\|}, \tag{7}$$

$$Var(\bar{\varphi}) = E_\varphi(\varphi_j - \bar{\varphi})^2, \tag{8}$$

where J is the number of recovered orientation vectors from the cluster. A cluster will be classified as a rigid plane if $Var(\bar{\varphi})$ is small than a threshold. Moreover, it will be classified as a vertical plane if $\bar{\varphi}$ is smaller than a threshold.

A special case is when the sign plane is perpendicular to optical axis *Z*; then we can use simplified criteria to verify candidate planes. In this situation, points on the road sign have almost the same distance to the camera, which can be expressed as:

$$Z_1 \approx Z_2 \approx Z_3. \tag{9}$$

Given the above condition we can obtain the equality among the ratios M_k from Eqs. (3) & (4):

$$M_1 \approx M_2 \approx M_3. \tag{10}$$

From Eq. (6) we see that the ratios of both components X and Y to the length of N should be smaller than a threshold if Γ is a vertical plane. This is a quite intuitive observation since we can image that N is almost parallel to the axis Z given the Eq.(9) holds. Thus, simplified verification criteria are shown as follows,

$$\bar{\gamma} = \frac{1}{J}\sum_{j=1}^{J} \frac{\sqrt{|X_j|^2 + |Y_j|^2}}{\|N_j\|}, \tag{11}$$

$$Var(\bar{\gamma}) = E_\gamma(\gamma_j - \bar{\gamma})^2. \tag{12}$$

The criteria indicate the triangle of P_1, P_2, P_3 is about to maintain the consistent spatial structure from t_0 to t_1. More generally, such consistency of spatial structure should hold whenever the condition in Eq.(9) is satisfied. Similarly, we use MLE to estimate the mean and variance of γ. Table 1 summarizes the algorithm of sign plane classification. In the algorithm the perpendicular plane is named as P, the vertical plane as V, the non-vertical plane as NV, the rigid plane as R and the non-rigid plane as NR. The relationship among them is: P $\subset$ V $\subset$ R and R $\cap$ NR = $\varnothing$.

Once we have candidate road signs, we can focus on text detection within these candidates only, which substantially reduces the text detection search space. Figure 4 shows a comparison experiment. (a) is the video image with the located road sign area bounded by the white MBR and yellow points are selected points on the sign; and (b) shows the results of text detection only run inside the red MBR; (c) displays the whole image without sign localization and (d) depicts the results of text

detection run on it. We clearly see that results in (b) and (d) have the same number of correct hits while (d) has one false hit (the white MBR on the left of sign). Thus, the two-step strategy is shown to effectively avoid complex background and "text-like" areas, thus can reduce the false hit rate, and even improve the hit rate by a focus-of-attention schema to reduce the noises. More details of the merit of this approach are given in Section 5.

Table 1. The algorithm for the road sign localization.

Input: Feature point clusters $P^t = \{c_1^t, c_2^t, ..., c_{M_t}^t\}$

Output: $L^t = \{l_1^t, l_2^t, ..., l_{M_t}^t\}$, $l_i^t \in \{P, V, R, NR\}$ label for c_i^t.

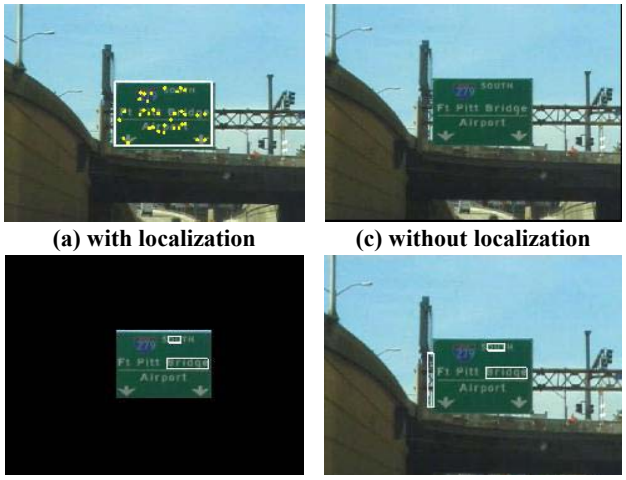
Algorithm:

1. R-plane identification step: Get c_i^t, compute the mean and variance of $\bar{\varphi}$. If $Var(\bar{\varphi}) < \varepsilon$, go to Step 2; otherwise, label $l_i^t = NR$. Get next cluster. If all clusters been examined, exit.

2. V-plane identification step: If $\bar{\varphi} < \delta$, go to Step 3; otherwise, label $l_i^t = R$. Go to Step 1.

3. P-plane identification step: Compute $\bar{\gamma}$ & $Var(\bar{\gamma})$. If $\bar{\gamma} < \xi$ & $Var(\bar{\gamma}) < \psi$, label $l_i^t = P$; otherwise label $l_i^t = V$. Go Step 1.

Note: $\varepsilon, \delta, \xi, \psi$ are thresholds in the algorithm.



(a) with localization **(c) without localization**

(b) 2 correct hits + 0 false hit **(d) 2 correct hits + 1 false hit**

Figure 4. Text detection with/out the road sign localization.

4. DETECTING TEXT

Nevertheless, correctly detecting text on road signs is still not easy because of deformations, highlights, shadows and other factors. We have to deal with these variations. To work around these changes in an image, we use an edge-based text detection method that integrates edge detection, adaptive searching, color analysis, and affine rectification. Figure 5 displays the basic flow of this schema.

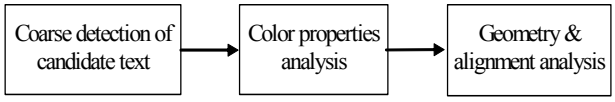


Figure 5. A text detection method.

4.1 Coarse Detection of Candidate Text

The intensity of an image is a major information source for text detection; however, it is well known that the intensity is sensitive to lighting variations. In contrast, the gradient of the intensity (edge) is less sensitive to lighting changes. Therefore, we use edge-based features in the first coarse detection phase. The main idea of the detection algorithm for coarse detection is as follows:

A multi-scale Laplacian of Gaussian (LOG) edge detector is used to obtain an edge set for each candidate text area. The properties of the edge set associated with each edge patch, such as size, intensity, mean, and variance, are then calculated. Some edge patches will be excluded from further consideration based on certain criteria applied to the properties, and the rest will be passed to a recursive procedure. The procedure attempts to merge adjoining edge patches with similar properties and re-calculate the properties recursively until no update can be made.

With LOG, we can obtain enhanced correspondences on different edge scales by using a suitable deviation σ. Since English letters in the same context share some common patterns, we can use them to analyze the alignment and rectification parameters, and refine detection results. Color distribution of the foreground and background is one such important property.

4.2 Color Analysis

Text on signs is designed for drivers to view at a distance, so they have highly distinguishable colors on their foregrounds and backgrounds, and also a high intensity contrast in their gray scale images. The property helps to make it easy to segment text and to describe letters using marginal distributions in a color space. However, it is almost impossible to obtain uniform color distributions of the foreground and background because of lighting sources, shadows, dust, etc. Here again we use GMM to characterize color distributions of the foreground and background of road signs, and more specifically, for each of the letters on road signs. Recall Eq. (1), β provides a cue on the complexity of the letter, $\|\mu_f - \mu_b\|$ indicates the contrast for a color space invariant to the lighting condition, and θ_b, θ_f yields the font style [2].

Since there could exist multiple lighting sources and shadows in natural scene video, contrasts of foreground and background might change significantly across the entire sign. Therefore, we model the distribution of each letter separately rather than the entire sign as a whole. We normalize *HSI* within the range of [0, 255] in computation.

These GMM parameters are used for text alignment analysis and estimation of affine parameters. An expectation maximization (EM) algorithm is applied to estimate the parameters. To differentiate background and foreground, we enlarge the boundary of the letter by 2 pixels on each side and calculate the color distribution of the region between the original boundary and the enlarged boundary. This distribution should be the same or similar to the background distribution. We then can determine the background distribution in the GMM by comparing distributions in the GMM to this distribution.

4.3 Text Alignment Analysis

The objective of text alignment analysis is to align characters in an optimal way, so that letters that belong to the same context will be grouped together. A text alignment has two cluster features:

intrinsic and extrinsic features. The intrinsic features are those which do not change with the camera position and the extrinsic features are those ones which change with the camera position. The intrinsic features include font style, color, etc; the extrinsic features include letter size, text orientation, etc. Both the intrinsic and extrinsic features can provide cues for the alignment analysis. The algorithm is shown in Table 2. The threshold T was selected empirically. For the English road sign database, the value was set as 0.7. Figure 6 depicts two images of the output from this stage. The left image shows the detected text lines within black MBRs and the right one contains the corresponding edge patches.

Table 2. The algorithm for text alignment analysis.

Input: Candidates of text regions $R = \{r_i = surround(e_i)\}$ where e_i is the edge of the candidate letters, $i = 0,1,\cdots,n-1$ and attribute sets $A = \{a_{e_i} = (x, y, h, w, \mu_f, \theta_f, \mu_b, \theta_b) \mid i = 0,1,\cdots,n-1\}$ where (x, y), h, and w are the center, height, and width of the surrounding rectangle. The rest are GMM parameters.

Output: $L = \{(x_i^1, y_i^1, x_i^2, y_i^2, x_i^3, y_i^3, x_i^4, y_i^4) \mid i = 0,1,\cdots,m-1\}$, the aligned text regions.

Algorithm:

1. Cluster r_i according to its attributes and obtain the candidate layout set $L = \{LR_j = \{r_i^j \mid r_i^j \in R\},\ j = 0,1,\cdots,m\}$;

2. If $\exists r_i, r_k \in LR_j$, $r = r_i \cup r_k$, satisfy $\max(h(r), w(r)) \le hw_{max}$, $hw_{max} = \max(h(r_i), w(r_i)), r_i \in LR_j$ & $r_i = r, LR_j = LR_j \setminus \{r_k\}$;

3. If $|LR_j| > 2$, use Hough transform to find all possible line segments $l_j^k\ (k = 0,1,\cdots,K-1)$, which are fitted by the centers of r_j^i. These line segments form several compatible sets $CL_j^m = \{l_j^k \mid k \in \{0,1,\cdots,K-1\}, \text{and } \forall l_j^{k_1}, l_j^{k_2}, \text{s.t.} l_j^{k_1} \cap l_j^{k_2} = \varnothing\}, m = 0,1,2,\cdots,M$

Only one winner of CL_j^m is selected by the following criteria:

a. $|CL_j^{win_1}| = \min\{|CL_j^m|, m = 0,1,\cdots,M-1\}$, and $|CL_j^{win_2}| = \min\{|CL_j^m|, m = 0,1,\cdots,M-1, m \ne win_1\}$;

b. Let μ_j^{win1} and μ_j^{win2} be the mean length of all line segments exclusive the shortest in $CL_j^{win_1}$ & $CL_j^{win_2}$. If $\mu_j^{win1} / \mu_j^{win2} < T$, $CL_j^{win_2}$ will be the winner. If $\mu_j^{win1} / \mu_j^{win2} > 1/T$, $CL_j^{win_1}$ will. Or, $CL_j^{win} = \arg\min_i \{\sum |l_j^k - \mu_j^i|, l_j^k \in CL_j^{win_i} \setminus \{\text{The shortest } l \text{ in } CL_j^{win_i}\}\}$

4. Remove all small candidate regions;

5. Find the corners of MBR for each candidate region.

Figure 6. The output of the text detection.

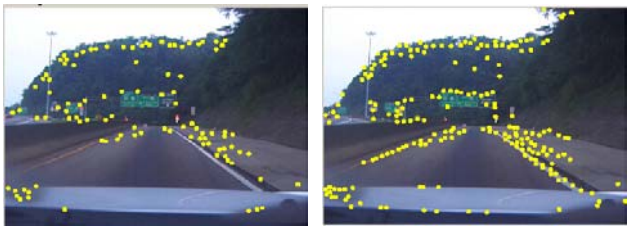
5. EVALUATION AND DISCUSSION

The proposed framework has been evaluated through experiments with a large and diverse set of road sign video sequences.

Figure 7. Incremental text detection process. (a)–(c) show 3 examples. Row 1: frames at the beginning of video sequences; Row 2: selected points; Row 3: located candidate road sign areas in white MBRs; Row 4: partial text detection results in yellow MBRs; Row 5: text detection results before the signs disappear; Row 6: images of road sign segments.

From a video database of 3-hour natural scene videos captured by a DV camera mounted on a moving minivan, we selected 22 short sequences with different driving situations, e.g., different routes (straight, curve), various vehicle speeds (low, high), different weather conditions (sunny, cloudy) and different times (day, dusk). For example, the speed of vehicle when videos were captured varies from 20 to 55 MPH. The objective of the selection is to be as diverse as possible and cover difficulty as well as generality of the task. Thus we did not include the extreme cases, e.g., crooked signs. Each video sequence is about 30 seconds and has a frame size of 640×480. The testing videos totally contain 92 road signs and 359 words (including numbers, e.g., speed limit). The system was implemented in C++ and evaluated on a 1.8 GHz Pentium IV PC. The number of selected points is set

as $N = 150$. The parameters in the sign localization algorithm are set as $\varepsilon = 0.1, \delta = 0.15$, $\xi = 0.2$, $\psi = 0.1$.



(a) $N = 100, 9th\ frame$ **(b)** $N = 200, 3th\ frame$

Figure 8. The number of selected feature points.

Figure 7 (a)–(c) show three examples of the flow of incremental detection of text on road signs from video. The video sequence in (a) contains three adjacent green road signs and other small road signs. The video was captured in cloudy weather conditions. We can see that the selected points (yellow ones) mainly appear in the texture-rich areas, e.g., the hill in background, road signs and side roads. Very few features appear in the less textured areas, such as the sky and the front ground. Images in Row 3 show results of road sign localization, the labeled candidate road signs are shown within white MBRs. This step refines the selected point sets by the plane classification algorithm, thus avoids feeding texture-rich areas into the text detection module. In the next a few frames, part of the text is detected on the road signs (Row 4), and they were merged and tracked over the time until all texts are detected or the road sign disappears from the video (Row 5). Finally, the images of road sign segments and their corresponding timing information are stored for later retrieval (Row 6). Indexing and retrieval are not discussed in this paper due to the page limit. Figure 7 (b) shows the process of detecting two green and one dark orange signs in dusk. Video sequences in both (a) and (b) were captured when the vehicle was moving forward straightly. Video in (c) contains two road signs which appear from the upper-left corner of the frame when the vehicle turns left on the ramp.

(a) **(b)**

Figure 9. False positive examples of road sign localization.

The number of selected points balances the computation speed and how fast the system can locate the road signs in video, see Figure 8. In case (a), the system selects 100 discriminative points every frame; and in case (b), the system selects 200 points. We can see that the system locates the road signs in the 9*th* frame when 5 points are found within the sign areas, shown in (a), while the system locates the signs in the 3*th* frame when more than 10 points are found, in (b).

Like other existing text detection algorithms, the framework has the problem of false positives. Figure 9 shows the false positive examples during the road sign localization step. Two falsely labeled candidate road signs are shown in (b), i.e., the vehicle's back in the front and the back of a road sign on the left. Obviously, we can use some constrains to remove such false positives, e.g., using the unique color distributions of road signs or differentiating the motions of moving objects and static objects. Recall Figure 4 which also shows an example of false positive in the text detection step.

(a) **(b)**

Figure 10. A case of occlusion. (a) selected discriminative points for the frame, (b) results of road sign localization

Figure 10 shows a video sequence in which occlusion happens twice. Three groups of road signs, each group in a line, appear in the sequence. We name them as A, B, and C shown in the figure. The first occlusion happens when a red car changes from the right lane across the road to the left lane in front of the camera. Results in rows 1-2 (b) show that the movement of the car does not affect the correct localization of B much, although the car does move in the neighborhood areas of B. The second occlusion happens when a white truck occludes the right part of B in row 3(b) and then

partially occludes B in row 4(b). Using the feature-based tracking, the system successfully tracks the un-occluded parts of B until it completely appears in row 5(b). The experiment shows that the proposed framework is robust for detecting and tracking road signs even when occlusion situation happens. Other algorithms can also be applied for tracking with occlusions, e.g. [19], however, we do not have space for the comparison and analysis.

Table 3. Results of road sign localization.

Total # of road signs	Hit rate	False hits
92	92.4%	17.9%

Table 4. Results of text detection.

Text Detection	Hit rate	False hits	Speed rate
Baseline	80.2%	85.6%	2~6 fps
Incremental	88.9%	9.2%	8~16 fps

Table 3 summarizes the performance of the road sign localization. We use the measures of "hit rate", "false hits" as in [12]. The system correctly detected 85 signs. Table 4 summarizes the overall text detection performance. Here, we consider a word as detected only if part of or the complete word is detected in the video by the system. We compared two methods, i.e., the baseline text detector (which runs the detection on the whole image), and the proposed approach. It is shown that the new framework significantly reduces the false hit rate and achieves a higher hit rate than the baseline. The low false hit rate is mainly because of the two step strategy of the new framework and a higher hit rate is mostly due to the focus-of-attention schema in the text detection stage. Table 4 the 4*th* column shows the processing rates of both methods. The new framework works much faster than the other one, which makes it possible to meet the run-time requirements of potential future applications.

6. CONCLUSIONS

Large amounts of information are embedded in the natural scene. Road signs are good examples of objects in natural environments which have rich information content. This paper presents a new framework for incrementally detecting text on road signs from video. The proposed framework efficiently embeds road sign localization and text detection into one framework using a Divide-and-Conquer strategy. Experiments have shown that the framework significantly improves robustness and efficiency of the existing text detection algorithm. The new framework has also provided a novel way for text detection from video by integrating features in 2D image with the 3D structure. The proposed framework can potentially be applied to other text detection tasks.

7. ACKNOWLEDGEMENTS

This research is partially supported by General Motors Satellite Research Lab at Carnegie Mellon University, and the Advanced Research and Development Activity (ARDA) under contract number H98230-04-C-0406. The authors would like to thank Dr. Datong Chen, Dr. Niall Winters, and anonymous reviewers for their helpful comments and suggestions.

8. REFERENCES

[1] Chen, D., Odobez, J.M., and Bourlard, H. Text detection and recognition in images and video frames. *Pattern Recognition, 37,* 3 (Mar. 2004), 595-608.

[2] Chen, X., Yang, J., Zhang, J., and Waibel, A. Automatic detection and recognition of signs from natural scenes. IEEE Trans. on IP, 13, 1 (Jan. 2004), 87-99.

[3] Clark, P., and Mirmehdi, M. Estimating the orientation and recovery of text planes in a single image. In Proc. of the 12th British Machine Vision Conference, 2001, 421-430.

[4] Fang, C.-Y., Fuh, C-.S., Chen, S.-W., and Yen, P.-S. A road sign recognition system based on dynamic visual model. In Proc. of the CVPR, 2003, I: 750-755.

[5] Gandhi, T., Kasturi, R., and Antani, S. Application of planar motion segmentation for scene text extraction. In Proc. of the ICPR, 2000, I: 445-449.

[6] Haritaoglu, E.D., and Haritaoglu, I. Real time image enhancement and segmentation for sign/text detection. In Proc. of the ICIP, 2003, III: 993-996.

[7] Jain, A.K., and Yu, B. Automatic text location in images and video frames. Pattern Recognition, 31, 12 (Dec. 1998), 2055-2076.

[8] Kastrinaki, V., Zervakis, M., and Kalaitzakis, K. A survey of video processing techniques for traffic applications. Image and Vision Computing, 21, 4 (Apr. 2003), 359-381.

[9] Lee, C.W., Jung, K., and Kim, H.J. Automatic text detection and removal in video sequences. Pattern Recognition Letters, 24, 15 (Nov. 2003), 2607-2623.

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

[11] Lienhart, R. Automatic text recognition for video indexing. In Proc. of ACM Multimedia (Nov. 1996), 11-20..

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

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

[14] http://www.fhwa.dot.gov/, Manual on Uniform Traffic Control Devices.

[15] Myers, G. Bolles, R., Luong, Q.-T., and Herson, J. Recognition of text in 3-D scenes. In Proc. of the 4th Symp. on Document Image Understanding Technology(2001), pp. 23-25.

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

[17] Shi, J., and Tomasi, C. Good features to track. In Proc. of the CVPR (1994), I:593-600.

[18] Wu, V., Manmatha, R., and Riseman, E,M. TextFinder: an automatic system to detect and recognize text in images, IEEE Trans. on PAMI, 21, 11 (Nov. 1999), 1224-1229.

[19] Wu, Y., Yu, T., and Hua, G. Tracking Appearances with occlusions. In Proc. of the CVPR (2003), II: 789-795.

[20] Zhang, D., and Chang, S. A Bayesian framework for fusing multiple word knowledge models in videotext recognition. In *Proc. of the CVPR* (2003), II: 528-533.