

Detection of Text on Road Signs From Video

Wen Wu, *Member, IEEE*, Xilin Chen, *Member, IEEE*, and Jie Yang, *Member, IEEE*

Abstract—A fast and robust framework for incrementally detecting text on road signs from video is presented in this paper. This new framework makes two main contributions. 1) The framework applies a divide-and-conquer strategy to decompose the original task into two subtasks, that is, the localization of road signs and the detection of text on the signs. The algorithms for the two subtasks are naturally incorporated into a unified framework through a feature-based tracking algorithm. 2) The framework provides a novel way to detect text from video by integrating two-dimensional (2-D) image features in each video frame (e.g., color, edges, texture) with the three-dimensional (3-D) geometric structure information of objects extracted from video sequence (such as the vertical plane property of road signs). The feasibility of the proposed framework has been evaluated using 22 video sequences captured from a moving vehicle. This new framework gives an overall text detection rate of 88.9% and a false hit rate of 9.2%. It can easily be applied to other tasks of text detection from video and potentially be embedded in a driver assistance system.

Index Terms—Object detection from video, road sign detection, text detection, vehicle navigation.

I. INTRODUCTION

AUTOMATIC detection of text from video is an essential task for autonomous or intelligent transportation systems [14]. There have been extensive research efforts in the detection, segmentation, and recognition of text from still images and video [2], [5], [6], [9], [10], [13], [15]–[19], [21], [23], [25], [27], [29], [32], [34]–[36]. In addition, research on road sign detection and recognition has recently become an active topic [3], [4], [6]–[8], [12], [24], [26], [31]. Related research on license plate recognition and vision-based navigation can be found in [1], [10], [30], and [33]. In this paper, we focus on the task of automatically detecting text on road signs from video and using that information in a driver assistance system.

Text on road signs carries much useful information for driving; it describes the current traffic situation, defines right-of-way, provides warnings about potential risks, and permits or prohibits roadway access. Automatic detection of text on road signs can help to keep a driver aware of the traffic situation and surrounding environments by seeing and highlighting signs that are ahead and/or have been passed. The system can also

Manuscript received July 20, 2004; revised August 5, 2005. This work was supported in part by the General Motors Satellite Research Laboratory, Carnegie Mellon University, and the Advanced Research and Development Activity (ARDA) under Contract H98230-04-C-0406. The work of X. Chen was supported in part by the Natural Science Foundation of China under Grant 60332010 and the “100 Talents Program” of the Chinese Academy of Sciences. The Associate Editor for this paper was C. Stiller.

W. Wu and J. Yang are with the School of Computer Science, Carnegie Mellon University, Pittsburgh, PA 15213 USA (e-mail: wenwu@cs.cmu.edu; jie.yang@cs.cmu.edu).

X. Chen is with the Institute of Computing Technology, Chinese Academy of Sciences, Beijing 100080, China (e-mail: xlchen@ieee.org).

Digital Object Identifier 10.1109/TITS.2005.858619



Fig. 1. Examples of road signs in different situations including different lighting conditions, weather, and highlights.

read out text on road signs with a synthesized voice. Such a multimedia system can reduce driver’s cognitive load and enhance safety in driving. Furthermore, it can be integrated with other navigation systems such as global positioning system (GPS)-based navigation systems.

The application scenario begins with a video camera mounted on a moving vehicle capturing the scene in front of the vehicle. The system attempts to detect text on road signs from video input and help the driver maneuver in traffic. Fig. 1 shows four examples of road signs. We can see that correctly detecting text on road signs poses many challenges. First, video images have relatively low resolution and are noisy. Both the background and foreground of a road sign can be very complex and can change frequently in video. Lighting conditions are uncontrollable due to the time of day and the current weather. A sign-reading system must be able to read signs in a variety of conditions such as broad daylight, shaded areas, cloudy days, dusk, rain, and snow. Second, the typography of the sign text can be rendered in a multitude of fonts, sizes, and colors. Third, text moving quickly in video can be blurred by motion or occluded by other objects. Finally, 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. We will only address the horizontal case in this paper and leave other situations to future research.

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 advantage of spatiotemporal 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. 2) Detect text within the candidate road sign areas and then fuse the detection results with the help of a feature-based tracker. More concrete ideas will be presented in Section II.

Next, we review some related work. Based on its origin, text in video can be classified into two classes, namely 1) graphic text; and 2) scene text [18]. Graphic text is text that is added to the video after the video is captured, such as captions added to news videos. Scene text exists as part of objects in a natural environment when it is directly captured by a camera, which includes billboards and street names on road signs. A common assumption used by previous research in graphic text detection from video is that the text plane is perpendicular to the optical axis of the camera [13], [17]. This is suitable for some domains such as broadcast video where the camera is fixed or has relatively little motion. However, the assumption does not necessarily hold in scene text detection task, since road sign planes are often encountered at a nonperpendicular angle with respect to the camera optical axis.

More general techniques for detecting scene text from still images have been developed in pattern recognition and computer vision fields. Recently, some researchers were able to detect scene text from still images [3], [5] and reported that edge features can better handle lighting and scale variations in scene images than texture features [4], which are often used for detecting text in news video [13], [15]. Inspired by their work, we chose to use the edge-based features for text detection in this study. Myers *et al.* described a full perspective transformation model to detect three-dimensional (3-D) deformed text from still images [23].

Research on extracting scene text from video has informed our work. Fang *et al.* introduced a dynamic visual model for recognizing road signs in video but was limited to road sign symbols, such as those for “stop” and “do not enter” instead of text [8]. Haritaoglu and Haritaoglu used a combination of symmetric neighborhood filtering and hierarchical connected component analysis to extract written information on road signs in scene images [12]. Piccioli *et al.* used *a priori* knowledge on scene and color clues to search suitable regions of road signs in images [26]. This approach works for images of cluttered urban streets as well as country roads and highways. Miura *et al.* designed a two-camera system consisting of a wide-angle-lens camera and a telephoto lens. The wide-angle-lens camera is used to detect candidates for road signs using color, intensity, and shape features, and the telephoto lens is directed to the road sign to capture a high-resolution image of the candidate [24]. Gandhi *et al.* applied a plane motion model to correct the perspective distortion of the text planes for robust detection [9]. Vitabile *et al.* proposed a method focusing on detecting road sign symbols instead of text by using multilayer perception neural network, and image data were used to evaluate the system [31].

The rest of the paper is organized as follows. Section II describes the overall architecture of the proposed incremental framework for text detection. Section III presents a road sign localization algorithm using a vertical plane criterion. Section IV discusses the detection of text by an edge-based algorithm.

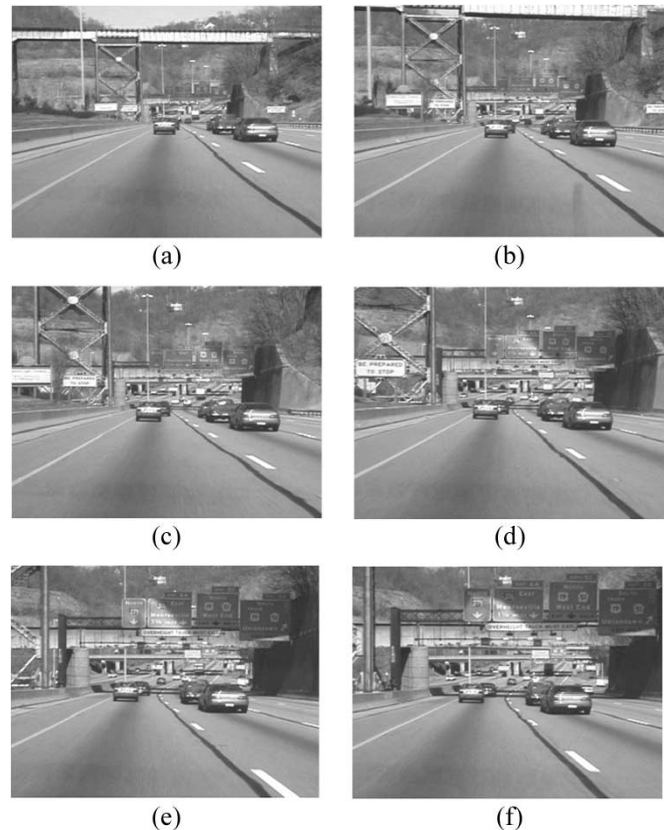


Fig. 2. Examples of road signs extracted from a single video sequence. (a) Frame 5. (b) Frame 32. (c) Frame 64. (d) Frame 96. (e) Frame 125. (f) Frame 155.

Section V describes the system implementation and presents experimental results. Section VI concludes with discussion of future work.

II. INCREMENTAL SPATIO TEMPORAL TEXT DETECTION

Some previous research work has paid particular attention to detecting and recognizing symbols on road signs, particularly warning signs such as “STOP,” “YIELD,” and “DO NOT ENTER.” Since only a finite number of shapes and colors can be applied on these warning signs, color and edge-based shape features are normally used to train the detector [8]. In this work, however, we are interested in detecting not only symbols, but also text, on road signs. Text appearing on road signs can have a variety of appearances. Color and shape features are not enough to train a robust detector. Without knowing text on the signs, drivers cannot obtain correct information about current traffic situation and appropriate driving instructions.

Accurate real-time sign detection with few false positives is an essential requirement for the proposed framework to improve the safety and efficiency of driving. Fig. 2 shows a video sequence of road signs captured by a video camera mounted on a moving minivan. At the beginning, there are some green information signs far ahead. A road sign first appears as a very small rectangle that progressively increases in size. As the vehicle approaches it, texts on the sign become visible. From a technical point of view, the size of the road sign is

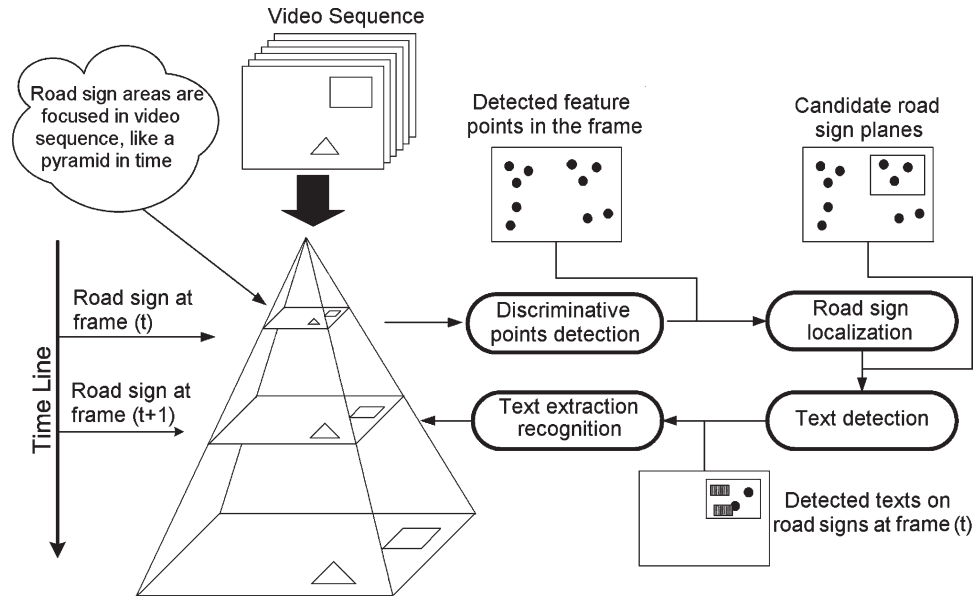


Fig. 3. Architecture of the proposed framework.

so small at the beginning of the process that it is impossible to perform text recognition or even detection, e.g., Fig. 2(a)–(c). This is unsurprising, since a human driver cannot distinguish the text at these ranges either. In contrast, it is realizable for some algorithm to detect the road sign instead of the text at first. Moreover, high computation requirements preclude the execution of text recognition algorithms on the real-time video sequences. Meanwhile, a high false hit rate is a major consequence when most existing text detection algorithms are applied, because many “text-like” areas can exist in every frame of natural scene video.

To improve the efficiency of the 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 the two subtasks, namely: 1) localizing road signs; and 2) detecting text. The key idea for realizing such an incremental framework is to exploit the temporal information available in video. This idea has been shown to be effective in other text detection tasks such as caption detection in broadcast video [17], [19]. Moreover, because of government requirements on the design and placement of road signs [22], this task also has some auspicious properties for the new framework, which are listed as follows: 1) text on road signs has higher contrast compared to most sign background colors; 2) text on the same road sign always has similar foreground and background patterns; 3) most road signs exist on vertical planes; and 4) there are only a limited number of colors used as background colors.

The proposed framework considers the whole period of appearance of a road sign in a video as a pyramid of sign image patches along the time line. Fig. 3 shows the architecture of the framework from which four main steps are summarized as follows.

- 1) Discriminative points detection and clustering—detect discriminative feature points in every video frame using the algorithm proposed in [28] and partition them into clusters.

- 2) Road sign localization—select candidate road sign regions corresponding to clusters of feature points using a vertical plane criterion.
- 3) Text detection—detect text on candidate road sign areas and track them.
- 4) Text extraction and 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 are clustered using 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 for the sign localization 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 verify if feature points appear on the same vertical plane. The number of the false positives caused by “text-like” areas is reduced through the sign localization step. Further, a multiscale text detection algorithm is performed on those candidate road sign areas, which is the text detection step. A minimum-bounding rectangle (MBR) is fitted to cover every detected text line. A feature-based tracking algorithm [20] is then used to track all detected areas and feature points in the MBRs over the timeline as they are merged with other newly detected texts in the sequence. Finally, all detected text lines are extracted for recognition, given a satisfactory resolution in the text extraction and recognition step. The recognition can be done by integrating an optical character recognition (OCR) system. Since recognizing the text on 640×480 pixel images might be a challenge for standard OCR, our strategy is to apply recognition step only when the detected text lines are at an adequate size such as 20 pixels or more. By doing this, our system can achieve a very good recognition rate.

There are several ways that we augment the reliability of existing OCR techniques including preprocessing,

postprocessing, and a trainable OCR system. In the preprocessing step, we can make use of the redundancy provided by a sequence of video images, since a sign appears in a sequence of video frames. Multiframe techniques such as interpolation, image mosaicing, and superresolution can be applied. The superresolution method has proven effective for improving accuracy of OCR from a video input [17], [27]. Postprocessing of OCR can potentially improve the accuracy of an OCR system, because there has been much work on robust parsing of imperfect speech recognition output. Similarly, a trainable OCR can be trained in low quality of images to improve recognition. Chen *et al.* demonstrated an intensity-based OCR system that achieved significant improvements on sign recognition tasks [4]. 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, N , which is the number of selected points in step 1), balances the sign localization speed and system process rate because of more feature points' likelihood that the sign is located early. A large number of feature points also mean intensive computation. Second, spatiotemporal information is extracted and used by the framework to recover the orientation of potential planes in the 3-D space. Once a point cluster is classified as a vertical plane, the text detection algorithm will be run on it. Third, the framework applies a feature-based tracker that can track a feature point in a subpixel level. 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 a good feature to track compared to those selected points. 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 V.

The new framework possesses two unique merits.

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

We describe steps 1) and 2) next and step 3) in Section IV.

III. 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 constraints. In the following subsections, we show how to detect discriminative points and use the vertical plane criterion for finding road signs from video.

A. Discriminative Point Detection

To recover the orientations of rigid planes in videos, the system finds a number of discriminative feature points in the

current video frame at any given frame. Features are found using the detector of Shi and Tomasi [28]. This method finds features that are good and easy to track. Compute the Laplacian matrix for each pixel in the image and also its minimum eigenvalue λ_m . Select λ_{\max} , which is the maximum value of λ_m over the whole image. Retain the image pixels that have a λ_m value larger than 10% of λ_{\max} . From these selected pixels, retain the local maximum pixels whose value is larger than that of any other pixel in its 3×3 neighborhood. In addition, keep a subset of those pixels so that the minimum distance between any pair of pixels is larger than a given threshold distance. After the feature selection step, we model the neighborhood of each detected feature point using a Gaussian Mixture Model (GMM), since points on road signs share common color properties

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

where G_f and G_b are the color distributions of the foreground and background, respectively. Therefore, each feature point can be represented as a vector such as $(\beta, \mu_f, \mu_b, \theta_f, \theta_b)$. The GMM parameters are used as features for the clustering of selected points. Each color space has its own characteristics. Previous research shows that the hue saturation intensity (HSI) color space can better handle the lighting variations than others when saturation is not too low [3], [4], which often happens in the natural scene environment. We use the H component in color analysis here.

In this stage, we then use the K -means algorithm to obtain a set of feature point clusters using color analysis, $\{C_1^t, C_2^t, \dots, C_K^t\}$ at time t . K is the number of clusters and is set at 10 in our experiments. $C_i^t = [p_j^t, \dots, p_k^t]$ is a cluster including from the j th feature point to the k th point. 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. MBRs are computed for each cluster. The visual illustration of the outcome after this step is shown in Fig. 3, where discriminative feature points are grouped into different clusters and, next, the localization algorithm is applied to detect candidate road sign area(s).

B. Vertical Plane Assumption

We are estimating the orientations of the candidate planes (signs) given three or more points in two successive frames. Here, we make two assumptions. 1) The optical axis of the camera is roughly horizontal and the motion of camera is also going along its optical axis. 2) Scene text lies on planar surfaces. These two assumptions are often true in the real-world setting. Particularly, the camera is mounted on the vehicle in our task and its optical axis is calibrated parallel to the horizontal plane of the vehicle. The upper half of Fig. 4 shows the side view of the scenario, and the lower half shows the spatial constraints among the road sign plane, the image plane, and the camera between two successive frames. Here, we have several coordinate systems.

- 1) The camera coordinate system $O^t X^t Y^t Z^t$ and the imaging coordinate system $o^t x^t y^t$ at time t . The optical axis

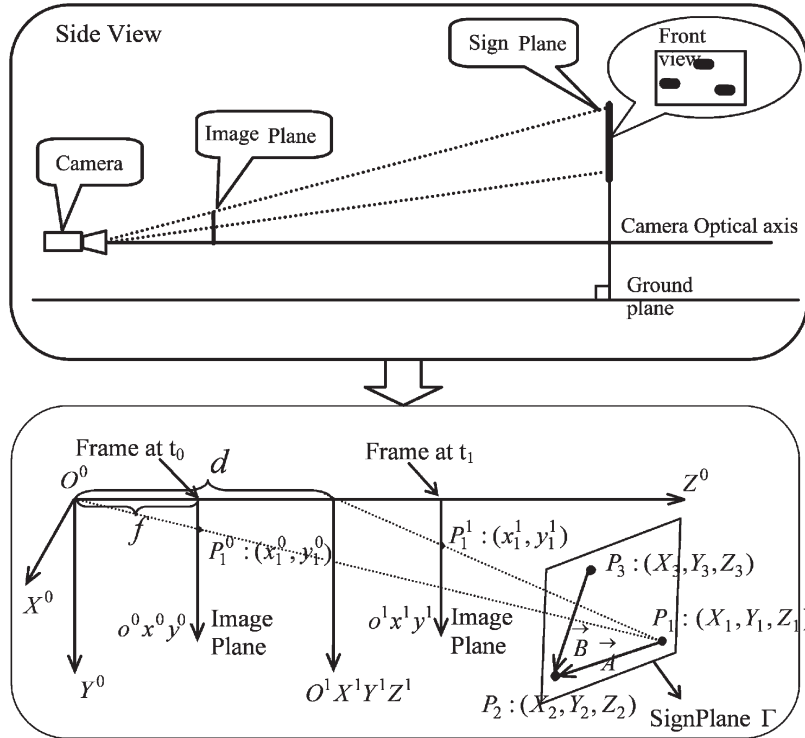


Fig. 4. Basic spatial relationship between two frames at t_0 and t_1 .

is the Z -axis of $O^t X^t Y^t Z^t$, and the X -axis is parallel to the horizon.

- 2) We take the camera coordinate system at time t_0 as the basic world coordinate system $OXYZ$.

$P_1(X_1, Y_1, Z_1)$, $P_2(X_2, Y_2, Z_2)$, and $P_3(X_3, Y_3, Z_3)$ are assumed to be three noncollinear points on a road sign plane Γ in Fig. 4. Here, a pinhole camera model is used and we let the camera's focal length be f , and the camera moves forward a distance d from time t_0 to t_1 . As $(t_1 - t_0)$ is normally very small for a real-time video stream, the assumption that the motion of vehicle is always small enough from t_0 to t_1 often holds. The projections of points P_i ($i = 1, 2, 3$) onto two image planes are $p_i^{t_0} : (x_i^{t_0}, y_i^{t_0})$ and $p_i^{t_1} : (x_i^{t_1}, y_i^{t_1})$ ($i = 1, 2, 3$), respectively.

Here, a feature-based tracker is used to find the correspondence of points between t_0 and t_1 [19]. Equation (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 $OXYZ$

$$\begin{pmatrix} x_i^{t_0} \\ y_i^{t_0} \end{pmatrix} = \frac{f}{Z_i} \begin{pmatrix} X_i \\ Y_i \end{pmatrix}, \quad i = 1, 2, 3 \quad (2)$$

$$\begin{pmatrix} x_i^{t_1} \\ y_i^{t_1} \end{pmatrix} = \frac{f}{Z_i - d} \begin{pmatrix} X_i \\ Y_i \end{pmatrix}, \quad i = 1, 2, 3. \quad (3)$$

We further write down the expressions for P_i in (4)

$$P_i : \begin{pmatrix} X_i \\ Y_i \\ Z_i \end{pmatrix} = \frac{d}{f} \begin{pmatrix} x_i^{t_0} x_i^{t_1} & y_i^{t_0} y_i^{t_1} & f x_i^{t_1} \\ x_i^{t_1} - x_i^{t_0} & y_i^{t_1} - y_i^{t_0} & x_i^{t_1} - x_i^{t_0} \end{pmatrix}^T \quad i = 1, 2, 3. \quad (4)$$

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

$$\frac{x_k^{t_1}}{(x_k^{t_1} - x_k^{t_0})} = \frac{y_k^{t_1}}{(y_k^{t_1} - y_k^{t_0})}, \quad k = 1, 2, 3. \quad (5)$$

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

$$M_k = \frac{x_k^{t_1}}{x_k^{t_1} - x_k^{t_0}} = \frac{y_k^{t_1}}{y_k^{t_1} - y_k^{t_0}}, \quad k = 1, 2, 3. \quad (6)$$

The lower half of Fig. 4 depicts that \vec{A} is a vector from P_1 to P_2 , and \vec{B} is a vector from P_3 to P_2 . Using the estimated coordinates of P_i in (4), we further obtain the estimations of \vec{A} and \vec{B} as

$$\vec{A} : \begin{pmatrix} X_1 - X_2 \\ Y_1 - Y_2 \\ Z_1 - Z_2 \end{pmatrix} = \begin{pmatrix} d \frac{x_1^{t_0}}{f} M_1 - d \frac{x_2^{t_0}}{f} M_2 \\ d \frac{y_1^{t_0}}{f} M_1 - d \frac{y_2^{t_0}}{f} M_2 \\ d(M_1 - M_2) \end{pmatrix} \quad (7)$$

$$\vec{B} : \begin{pmatrix} X_3 - X_2 \\ Y_3 - Y_2 \\ Z_3 - Z_2 \end{pmatrix} = \begin{pmatrix} d \frac{x_3^{t_0}}{f} M_3 - d \frac{x_2^{t_0}}{f} M_2 \\ d \frac{y_3^{t_0}}{f} M_3 - d \frac{y_2^{t_0}}{f} M_2 \\ d(M_3 - M_2) \end{pmatrix}. \quad (8)$$

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

$$N = \vec{A} \times \vec{B} = (X_\Gamma \quad Y_\Gamma \quad Z_\Gamma)^T \quad (9)$$

where the expression for each component of N , i.e., X_Γ , Y_Γ , Z_Γ , is written as

$$\begin{aligned} X_\Gamma &= \frac{d^2}{f} [(y_1^{t_0} M_1 - y_2^{t_0} M_2) (M_3 - M_2) \\ &\quad - (y_3^{t_0} M_3 - y_2^{t_0} M_2) (M_1 - M_2)] \\ &= \frac{d^2}{f} C_X \\ Y_\Gamma &= \frac{d^2}{f} [(x_3^{t_0} M_3 - x_2^{t_0} M_2) (M_1 - M_2) \\ &\quad - (x_1^{t_0} M_1 - x_2^{t_0} M_2) (M_3 - M_2)] \\ &= \frac{d^2}{f} C_Y \\ Z_\Gamma &= \frac{d^2}{f^2} [(x_1^{t_0} M_1 - x_2^{t_0} M_2) (y_3^{t_0} M_3 - y_2^{t_0} M_2) \\ &\quad - (x_3^{t_0} M_3 - x_2^{t_0} M_2) (y_1^{t_0} M_1 - y_2^{t_0} M_2)] \\ &= \frac{d^2}{f^2} C_Z \end{aligned}$$

where C_X , C_Y , and C_Z are used to represent the long terms in equations and to simplify the expressions. Equation (9) gives a nice way to estimate the orientation of the candidate plane by using the three points' image coordinates given the spatial constraints. By taking advantage of the approximations, we can further define a model to classify planes into positive and negative categories using different criteria such as vertical versus nonvertical planes or rigid versus nonrigid planes. Here, we are interested in the vertical plane criterion.

Based on the property that most road signs are on vertical planes, the ratio of the Y component to the length of N is supposed to be smaller than a certain threshold. Thus, we can estimate the ratio and use it to locate vertical planes. The ratio is defined as

$$R = \frac{|Y|}{\|N\|} = \frac{|C_X|}{\sqrt{C_X^2 + C_Y^2 + \frac{1}{f^2} C_Z^2}}. \quad (10)$$

C. Sign Localization Algorithm Description

We have obtained a number of clusters of feature points from the feature detection step. Now, we can apply the vertical plane assumption to verify if a cluster represents a candidate road sign plane. In order to reduce the error caused by outliers, we use the median as the normal vector of candidate plane and also calculate its variance

$$R_i = \text{median}(R_{ij}), \quad j = 1, 2, \dots, m_i \quad (11)$$

$$\text{Var}(R_i) = E(R_{ij} - R_i)^2, \quad j = 1, 2, \dots, m_i \quad (12)$$

where R_i is the median estimation for the i th cluster, m_i is the number of recovered orientation vectors from one cluster, and R_{ij} is the j th vector in the i th cluster. We use the median to approximate the orientation of the candidate plane associated with each cluster C_i^t . A cluster will be classified as a rigid

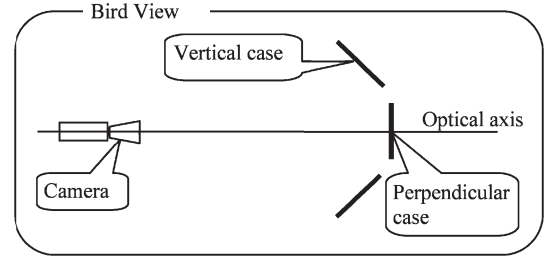


Fig. 5. Bird view of the perpendicular case.

plane if $\text{Var}(R_i)$ is smaller than a threshold. Moreover, it will be classified as a vertical plane if R_i is smaller than a threshold.

Specifically, when the sign plane is perpendicular to the optical axis of the camera, we can use a simplified criterion to verify candidate planes. In this case, points on the sign plane have almost the same distance to the camera along the Z -axis in $OXYZ$. The constraint for P_1 , P_2 , and P_3 can be written as

$$Z_1 \approx Z_2 \approx Z_3. \quad (13)$$

Given the above condition, we obtain the equality among the ratios M_k from (5) and (6)

$$M_1 \approx M_2 \approx M_3. \quad (14)$$

From (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 quite an intuitive observation, since we can imagine that N is almost parallel to the optical axis Z given that (13) holds. Fig. 5 depicts the perpendicular situation as a special case of the vertical case.

Thus, the simplified verification criterion is given as

$$\bar{\gamma} = \frac{1}{J} \sum_{j=1}^J \frac{\sqrt{|X_j|^2 + |Y_j|^2}}{\|N_j\|} \quad (15)$$

$$\text{Var}(\bar{\gamma}) = E_\gamma(\gamma_j - \bar{\gamma})^2. \quad (16)$$

The criterion indicates that the triangle of P_1 , P_2 , and P_3 is about to maintain the consistent spatial structure from t_0 to t_1 . More generally, such consistency of the spatial structure should hold whenever the condition in (13) is satisfied. Algorithm I summarizes the process of sign plane classification. In this algorithm, the perpendicular plane is named as P , the vertical plane as V , the nonvertical plane as NV , the rigid plane as R , and the nonrigid plane as NR . The relationship among them is stated as follows: $P \subset V \subset R$, $V \cap NV = \emptyset$, and $R \cap NR = \emptyset$.

Algorithm I: Road Sign Localization Using a Vertical Plane Criterion

Input: Feature point clusters $\{C_1^t, C_2^t, \dots, C_K^t\}$.

Output: $L^t = \{l_1^t, l_2^t, \dots, l_K^t\}$, $l_i^t \in \{P, V, R, NR\}$ label for C_i^t . P is for perpendicular planes, V is for vertical planes, R is for rigid planes, and NR is for nonrigid planes.

Algorithm:

- 1) R -plane identification step: Compute R_i and $\text{Var}(R_i)$ using (14) and (15). If $\text{Var}(R_i) < \varepsilon$, then go to step 2);

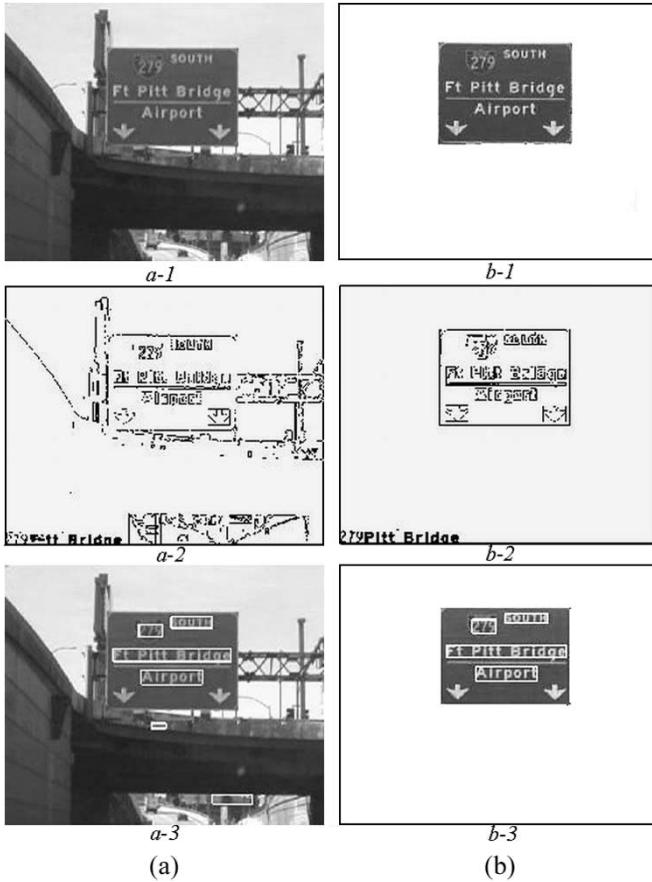


Fig. 6. Comparison of text detection performance of the proposed incremental detection algorithm and the baseline algorithm. Column (a) shows some results of the baseline algorithm and column (b) shows those of the proposed algorithm. Row (1) shows the input images to two algorithms; row (2) shows the intermediary edge detection results; and row (3) shows the text detection results. Two false hits are found in the baseline algorithm's result (a-3).

otherwise, label $l_j^t = NR$. Get the next cluster. If all clusters have been examined, exit.

- 2) *V*-plane identification step: If $R_i < \delta$, which means that it meets the vertical plane criterion, then go to step 3 to further verify that it is a perpendicular plane; otherwise, label $l_i^t = R$; it is a rigid but nonvertical plane. Go back to step 1).
- 3) *P*-plane identification step: Compute $\bar{\gamma}$ and $\text{Var}(\bar{\gamma})$ using (18) and (19). If $\bar{\gamma} < \xi$ and $\text{Var}(\bar{\gamma}) < \psi$, label $l_i^t = P$; otherwise, label $l_i^t = V$. Go back to step 1).

Note: ε , δ , ξ , and ψ are adjustable thresholds in the algorithm.

Once we identify candidate road signs from feature point clusters, we can focus on text detection within these candidates only, which can substantially reduce the search space for text detection. Fig. 6 shows a comparison experiment of two cases. Column (a) depicts the first case in which the text detection algorithm runs on every whole video image, and we call it the baseline algorithm. Column (b) illustrates the second case in which text detection runs only on the localized road sign areas, and it is the proposed algorithm. Row (1) shows input images for both algorithms; row (2) depicts results of edge detection; and row (3) shows the text detection results as bounded by white

MBRs. We clearly see that in both cases, the two algorithms achieve the same number of correct hits while the baseline algorithm has two false hits (two white MBRs outside the sign area in a-3). Thus, the new framework has been shown to effectively avoid complex backgrounds and “text-like” areas. This can reduce the rate of false hits and even improve the hit rate by constraining the search space to minimize noise. More details of the merits of the proposed algorithm will be given in the experiment section.

D. More General Case

Previously, we assume that the camera motion will only be traversal along the optical axis as in Fig. 4. However, this assumption is likely violated under some conditions such as when the vehicle makes a turn. Calibrating the camera parallel to the horizontal plane of vehicle does not help obviate the issue. From (10), we can see that variations of the ratio may come from two sources, namely: 1) a change in the focal length; and 2) violation of the assumption that motion occurs only along the optical axis. The first issue can be negated after f is calibrated prior to the experiment and fixed during experiments. The second issue is relatively complicated, because it could cause the variations of more than one term in (10). Next, we will devote the analysis for the turning case in particular.

The key question to ask here is, can we recover the normal vector N of Γ when the vehicle makes a turn? Our conclusion is that we can recover N if only if we know the turning angle, such as θ . Assume that the relative movement from t_0 to t_1 is $(\Delta X, 0, \Delta Z)$, caused by turning, and the turning angle relative to the *Y*-axis is θ . Fig. 7 depicts such a turning scenario.

The normal vector to the plane $O^0P_1P_2$ in the $O^0X^0Y^0Z^0$ coordinate system is

$$N_0 = \begin{pmatrix} x_1^0 \\ y_1^0 \\ f \end{pmatrix} \times \begin{pmatrix} x_2^0 \\ y_2^0 \\ f \end{pmatrix} = \begin{pmatrix} (y_1^0 - y_2^0) f \\ (x_2^0 - x_1^0) f \\ x_1^0 y_2^0 - x_2^0 y_1^0 \end{pmatrix}. \quad (17)$$

Similarly, the normal vector to the plane $O^1P_1P_2$ in the $O^1X^1Y^1Z^1$ coordinate system is

$$N_1 = \begin{pmatrix} x_1^1 \\ y_1^1 \\ f \end{pmatrix} \times \begin{pmatrix} x_2^1 \\ y_2^1 \\ f \end{pmatrix} = \begin{pmatrix} (y_1^1 - y_2^1) f \\ (x_2^1 - x_1^1) f \\ x_1^1 y_2^1 - x_2^1 y_1^1 \end{pmatrix}. \quad (18)$$

Therefore, the normal vector to the plane $O^1P_1P_2$ in the $O^0X^0Y^0Z^0$ coordinate system can be computed as

$$N'_1 = \begin{pmatrix} \cos \theta & 0 & \sin \theta \\ 0 & 1 & 0 \\ -\sin \theta & 0 & \cos \theta \end{pmatrix} \begin{pmatrix} (y_1^1 - y_2^1) f \\ (x_2^1 - x_1^1) f \\ x_1^1 y_2^1 - x_2^1 y_1^1 \end{pmatrix} \\ = \begin{pmatrix} (y_1^1 - y_2^1) f \cos \theta + (x_1^1 y_2^1 - x_2^1 y_1^1) \sin \theta \\ (x_2^1 - x_1^1) f \\ (y_2^1 - y_1^1) f \sin \theta + (x_1^1 y_2^1 - x_2^1 y_1^1) \cos \theta \end{pmatrix}. \quad (19)$$

Thus, the normal vector of $\overline{P_1P_2}$ is

$$N_{12} = N_0 \times N'_1. \quad (20)$$

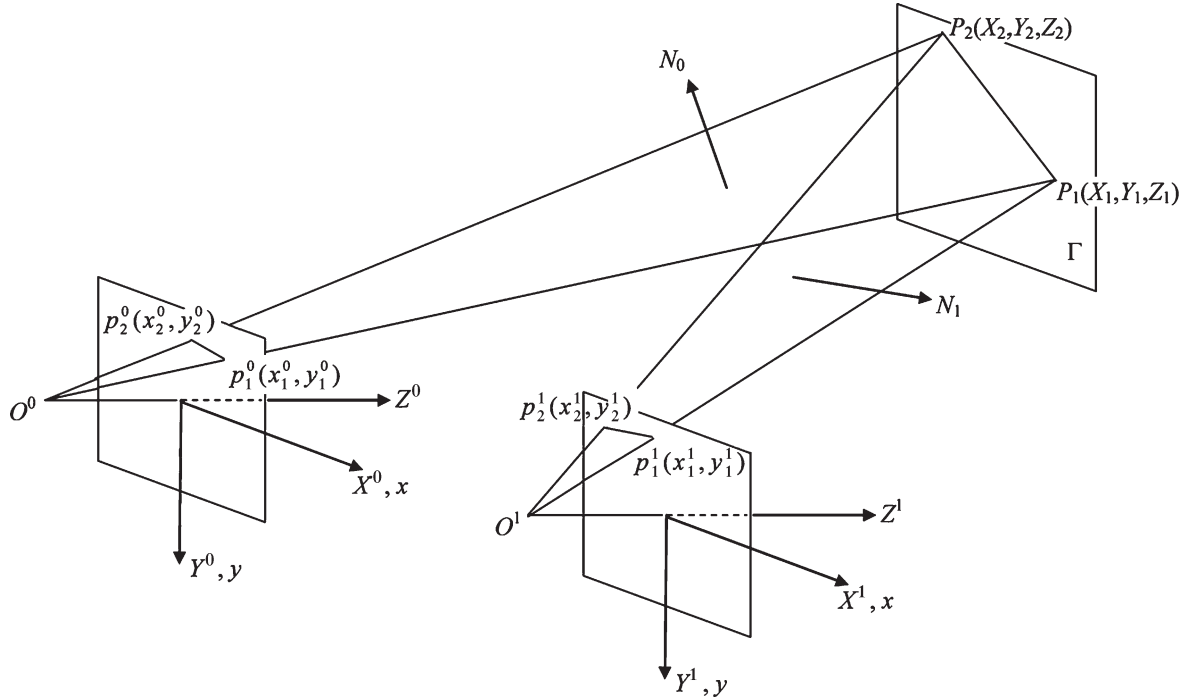


Fig. 7. Model sensitivity analysis for the vehicle turning case, when the assumption of the translation along the optical axis is violated.

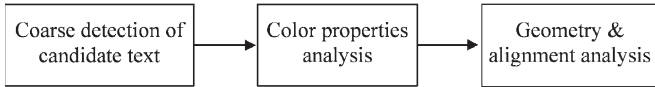


Fig. 8. Edge-based method for the detection of text from image.

Similarly, given a third point P_3 , we obtain N_{23} and N_{13} . Therefore, the normal vector of Γ will be

$$N = N_{12} \times N_{13}. \quad (21)$$

As we can see from the above equation, the answer to the question at the beginning of this section is that we can recover the normal vector of the plane Γ if we know or can approximate the turning angle θ .

IV. TEXT DETECTION

Even when sign locations are known in images, correctly detecting text on road signs is still not easy because of deformations, highlights, shadows, and other factors. To work around these changes in an image, we use an edge-based cascade text detection method that integrates edge detection, adaptive searching, color analysis, and geometry alignment analysis. This method was first proposed in [4]. Fig. 8 shows the basic flow of this schema.

A. Coarse Detection of Candidate Text

The intensity of an image is a major source of information 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 multiscale 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 recalculate 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. 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.

B. 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 they also have a high intensity contrast in their grayscale images. This property helps make it easy to segment text and 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 the GMM to characterize color distributions of the foreground and background of road signs, and more specifically, for each words on signs. Recall that 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.

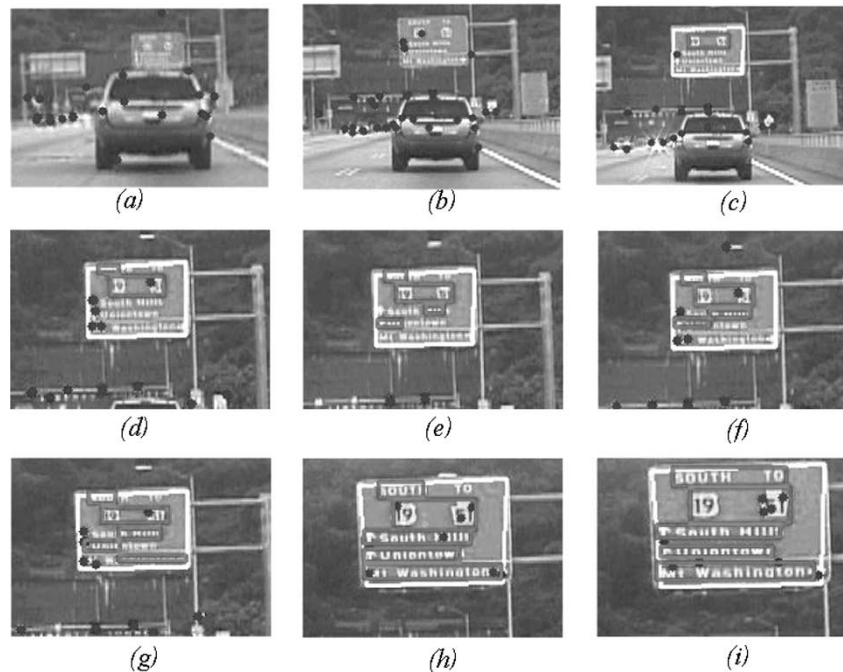


Fig. 9. Illustration of incremental text detection from a video sequence. The points are selected feature points, the biggest box bounds the localized road sign area and small boxes indicate the detected text lines.

Since there could be 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 components 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 the 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.

C. 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. Text alignment analysis includes two cluster features, namely 1) intrinsic features; and 2) extrinsic features. The intrinsic features are those that do not change with the camera position, and the extrinsic features are those that 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 first clusters text regions using intrinsic and extrinsic features, including the center, height, width, and GMM parameters of candidate text regions. Then, it uses the Hough transform to find all possible line segments. These line segments form several compatible sets. The two smallest sets are selected as

candidates. One winner from the compatible sets is selected by picking the line segment that has a larger mean length than all line segments excluding the shortest. Then, it removes all small candidate regions, and finally, it outputs the corner positions of MBR for each candidate region.

V. EVALUATION AND DISCUSSION

The proposed framework has been evaluated through experiments with a large and diverse set of road sign video sequences. From a video database of 3-h natural scene videos captured by a digital video (DV) camera mounted in a moving minivan, we selected 22 video sequences with different driving situations including different road conditions (straight, curve), vehicle speed (low, high), weather conditions (sunny, cloudy), and daylight variations. The objective of the selection was to be as diverse as possible and cover the range of difficulty as well as the generality of the task. Thus, we did not include the extreme cases, such as crooked lateral signs. Each video sequence is about 30 s, contains an average of 92 road signs, and 359 words (including numbers such as a speed limit), and has a frame size of 640×480 . Our system was implemented in C++ and tested on a 1.8-GHz Pentium IV personal computer (PC). The number of selected points was set at $N = 150$, and the number of clusters was set at $K = 10$. The parameters in Algorithm I were $\varepsilon = 0.1$, $\delta = 0.15$, $\xi = 0.2$, and $\psi = 0.1$.

A. Incremental Text Detection Process

Fig. 9 illustrates the process of incremental text detection. During a few initial video frames, no discriminative points are found on the road sign plane as shown in (a). In frame (b), some feature points are detected in the frame. The system

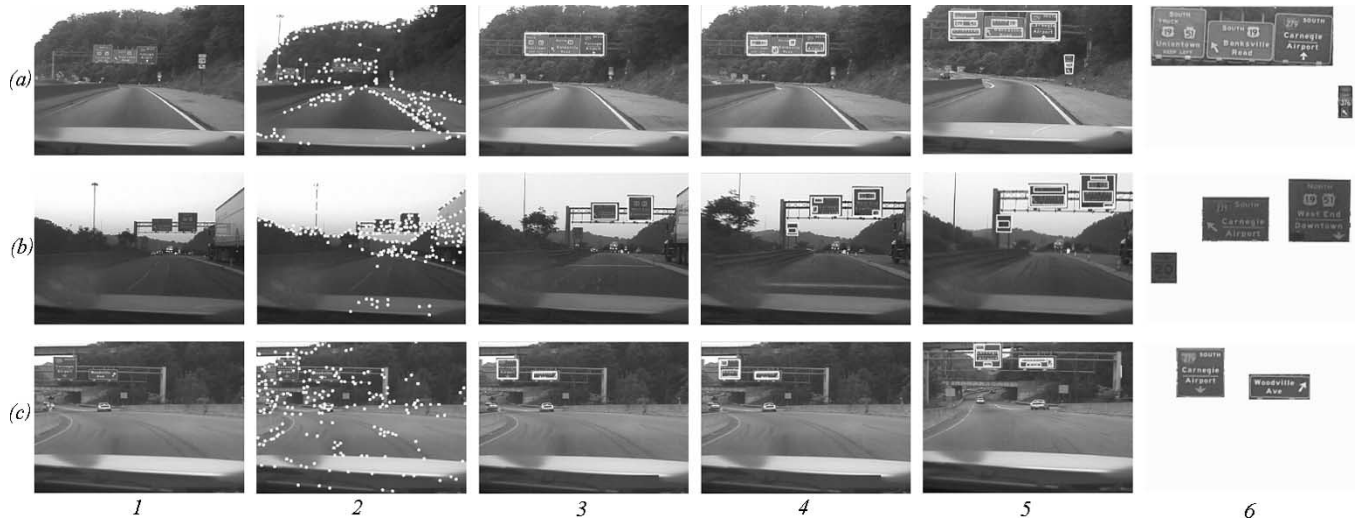


Fig. 10. (a)–(c) Three examples of incremental text detection process. Column 1: Frame of the beginning of the video sequence. Column 2: Selected discriminative points. Column 3: Located candidate road sign areas in white rectangles. Column 4: Partial text detection results in yellow rectangles. Column 5: Detection results before road sign disappears. Column 6: Extracted road sign segments from video.

then verifies the sign area by using the plane classification algorithm. Once the area is confirmed as a vertical plane it will be bounded with an MBR as in (c). The following frames, such as (d)–(h), show that more and more texts are detected on the road sign over the time. Newly detected text regions are merged with previously partial detection results. In the meantime, all detected text regions are tracked by averaging the optical flows of the feature points within the detected areas. Finally, all texts on the road sign are correctly detected as shown in (i). Note that there was a sign on the right at the beginning of the sequence in (a)–(c); however, our system did not detect it. The reason is that no feature points were found in that sign area.

Fig. 10 illustrates another three examples, one in each row. The video sequence in (a) contains three adjacent green road signs and other small road signs. The video was captured during daylight under cloudy weather. Selected feature points mainly appear in the texture-rich areas, such as the edge of forest in background, road signs, and side roads. Very few feature points appear in less texture areas, such as sky and front ground. Obviously, the feature selection step not only provides candidate clues for road sign localization but also reduces the search space for later text detection. Images in column 3 show results of road sign localization, i.e., the labeled candidate road signs within white rectangle. This step further refines the selected features by sign plane classification algorithm and avoids feeding some texture-rich areas, such as forests and side roads, into the text detection module, which could cause false hits. In the next few frames, partial texts are detected on the road signs from frame to frame. Detected text regions on road signs are merged and tracked over the time until all texts are detected or the road signs disappear from the frame. Finally, the image segments of extracted road sign are stored in the database and their appearance times and durations are recorded for other services such as indexing or retrieval. Row (b) shows the process of detecting another example of road signs from another sequence under dusk environment. Sequences in both (a) and (b) were captured when the vehicle was driving straight.

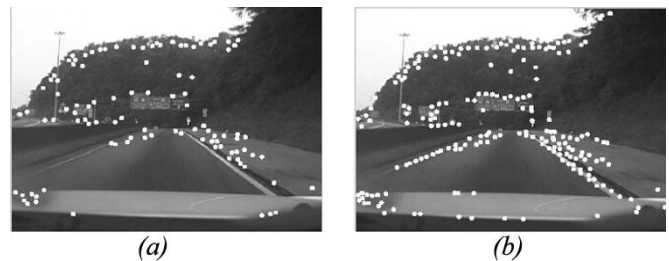


Fig. 11. Number of detected discriminative feature points. (a) $N = 100$. (b) $N = 200$.

However, row (c) shows the case when the vehicle makes a turn. As we analyze in Section III-D, when we assume a very small angle for the turning case, our algorithm can still handle the sign localization step well.

B. Algorithm Robustness Analysis

The number of selected features balances the computation speed and how fast the system can locate road signs in video. Fig. 11 depicts a comparison experiment. In case (a), the system selects 100 discriminative points; and in case (b), the system selects 200 points. At the same position of the sequence, the system can find 5 points on the sign in the first case, while more than 10 points are found on the sign in (b). The probability of early location of a road sign is directly proportional to the number of feature points on it. On the other hand, increasing the number of feature points can slow down the localization and detection steps and, further, the overall processing speed of the system.

Like other existing text detection algorithms, the framework has to deal with false positives. Fig. 12 shows examples of false positives in the road sign localization step. Two erroneously labeled candidate road signs are shown in (b), where one is the back of a road sign on the left (noted as A in the figure) and the other is the back of another vehicle (noted as B). We can use some constraints to remove such false positives. The

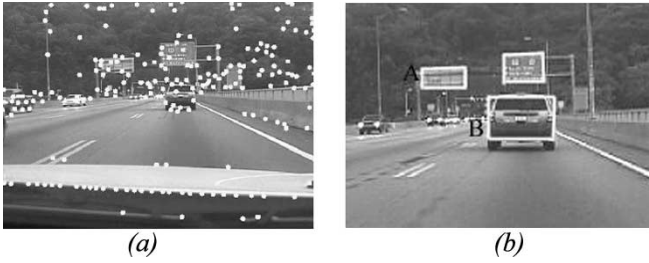


Fig. 12. False positives from road sign plane localization.

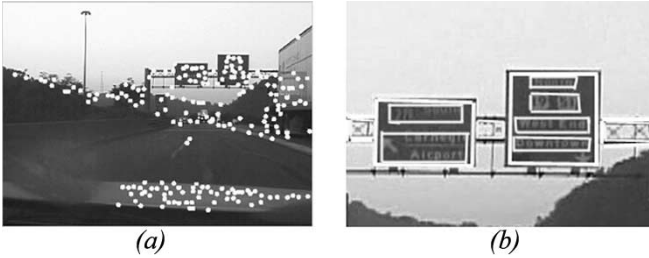


Fig. 13. False positives from the text detection step.

system has a postprocessing step to refine the sign localization results. Since very few discriminative points can be found on the back of a sign as it appears in the video, it is easy for the system to remove this kind of false positives by thresholding the number of detected feature points. For the type B case, the current system uses a color filter to filter them out. Other methods, such as motion from tracking, can also help to further reduce the false hit rate.

Fig. 13 shows an example of false positives in the text detection step. Fig. 13(a) displays the selected points for the frame, and Fig. 13(b) shows the detection results without color and text alignment analysis. Since the road sign support exists in the same vertical plane as the signs and also has a regular edge pattern that is similar to character strokes, the system falsely labels three areas between and beside road signs as “text.”

Fig. 14 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. The movement of the car does not affect the correct localization of B much, even though the car moves in the neighborhood areas of B. The second occlusion happens when a white truck occludes the right part of B in (b)-3 and then partially occludes B in (b)-4. Using the feature-based tracking, the system successfully tracks the unoccluded parts of B until it completely appears in (b)-5. The experiment shows that the proposed framework is robust for detecting and tracking road signs even when occlusion occurs.

C. Overall Evaluations

Table I summarizes the sign localization performance. The system correctly detected 85 of 92 signs. For each frame, the ground truth for text bounding boxes was initially created by

our system and further verified and adjusted manually. We apply the definitions of “hit rate” and “false hits” as defined in [18] and shown in (22).

We used a slight variation of the evaluation methodology. We directly counted the number of false hits instead of computing the ratio to avoid a greater-than-one value. The text box-based hit rate, false hit rate, and miss rate refer to the number of detected boxes that match with the ground truth, as defined in (22). A system-assigned text bounding box A was regarded as matching a ground truth text bounding box G if and only if these two boxes overlapped with each other by at least 80%

$$\begin{aligned} \text{hit rate}_{\text{box-based}} &= \frac{1}{M} \sum_{g \in G, a \in A} \delta(a, g) \\ \text{miss rate}_{\text{box-based}} &= 1 - \text{hit rate}_{\text{box-based}} \\ \text{false hits}_{\text{box-based}} &= N - \sum_{g \in G, a \in A} \delta(a, g) \end{aligned} \quad (22)$$

where $A = \{a_1, \dots, a_N\}$ and $G = \{g_1, \dots, g_M\}$ are the sets of box representing the system-assigned road sign boxes and the ground truth sign boxes. $N = |A|$ and $M = |G|$. In addition, $\delta(a, g)$ is defined as

$$\delta(a, g) = \begin{cases} 1, & \text{if } \min\left(\frac{|a \cap g|}{|a|}, \frac{|a \cap g|}{|g|}\right) \geq 0.8 \\ 0, & \text{otherwise.} \end{cases}$$

Table II and Fig. 15 summarize the overall text detection performance. We compare two methods in the experiment, namely: 1) the baseline algorithm (which analyzes the whole image for every video frame); and 2) the proposed algorithm. There are a total of 359 words in the 30 testing videos, including numbers and symbols such as arrows on all road signs. The new framework significantly reduces the false hit rate and achieves a higher hit rate than the baseline algorithm. The low false hit rate is mainly due to the two-step strategy of the new framework and a higher hit rate is due to the sign localization step before detection. The total false hit rate of text detection is 9.2% as shown in Table II. On average, there are about two false positives per minute in areas without traffic signs. Another merit of the new framework to mention is the improvement of processing bandwidth of the working system. Currently, the working system can process the video at the rate of 8–16 fps, which is about two to three times faster than the baseline algorithm. The detection rate of 88.9% is still not good enough for real-world applications. Some signs are missed in the sign plane localization step as shown in Table I. The reason for missing these signs is quite similar to the case of the orange sign in Fig. 9, where no feature points are found on the sign area. Therefore, later sign classification and text detection steps skip unlocalized areas in the video frame. On the other hand, sign tracking improves the text detection performance by reducing the number of false positives. At the beginning of the stage, signs are localized but the characters on them are too small to be detected, because the signs are far away from the vehicle. The feature-based tracker in our system tracks the localized signs in video, so it is possible to merge the newly detected text with previously detected text over the time.

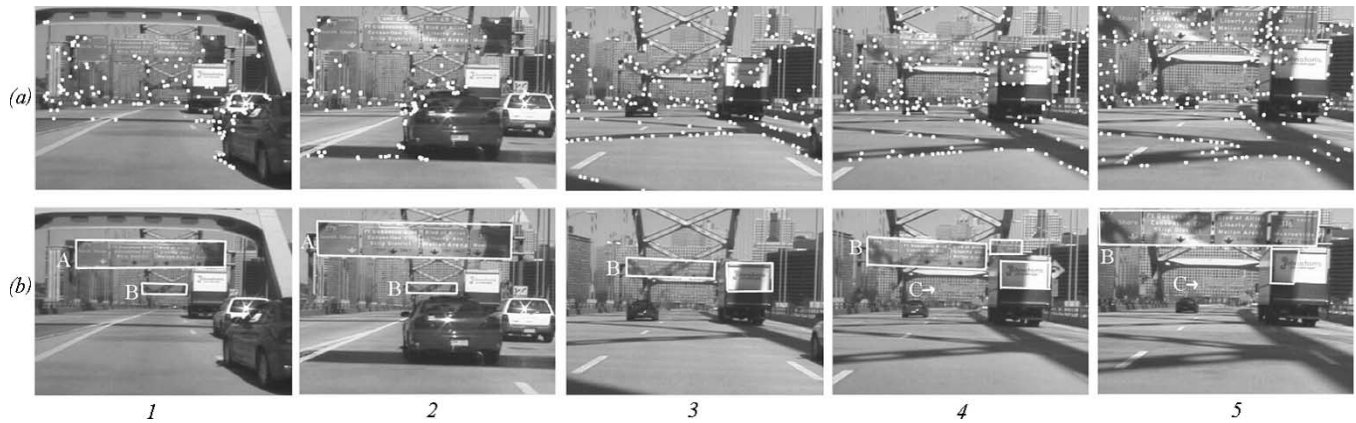


Fig. 14. Case of occlusion. (a) Selected feature points for each frame. (b) Localization of road sign.

TABLE I
RESULTS OF ROAD SIGN LOCALIZATION

Total # of road signs	Box-based	
	Hit rate	False hits
92	92.4%	17 (17.9%)

TABLE II
RESULTS OF TEXT DETECTION

Text Detection	Box-based		
	hit rate	false hits	miss rate
Refined baseline algorithm with text tracking	80.2%	307 (85.6%)	19.8%
Incremental algorithm with vertical-plane model	88.9%	33 (9.2%)	11.1%

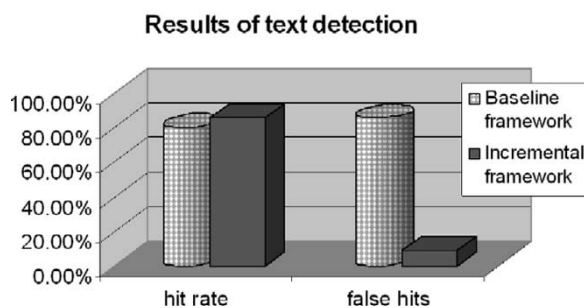


Fig. 15. Performance of two text detection frameworks.

VI. CONCLUSION AND FUTURE WORK

Large amounts of information are embedded in natural scenes. Road signs are good examples of objects in natural environments that 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 plane localization and text detection mechanisms with feature-based tracking into an incremental detection framework using a divide-and-conquer strategy. This strategy can significantly improve the robustness and efficiency of text detection. The new framework has also provided a novel way to detect road sign text from video by integrating image features and the vertical plane assumptions of road signs. Extensive

experiments have been conducted to demonstrate the feasibility of the new incremental detection framework under real-world settings. The basic ideas and techniques of this research can be potentially applied to other similar tasks of text detection from video. Interesting future work may include detecting variable message signs from video and exploring other robust image and video features for text detection.

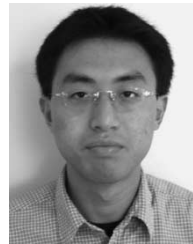
ACKNOWLEDGMENT

The authors thank J. Zhang, Y. Zhang, D. Chen, anonymous reviewers, and the Associate Editor for their helpful comments and suggestions. The authors also gratefully acknowledge the assistance of J. Wong for his helpful comments and for editing the English of the paper.

REFERENCES

- [1] S.-L. Chang, L.-S. Chen, Y.-C. Chung, and S.-W. Chen, "Automatic license plate recognition," *IEEE Trans. Intell. Transp. Syst.*, vol. 5, no. 1, pp. 42–53, Mar. 2004.
- [2] D. Chen, J. M. Odobez, and H. Bourlard, "Text detection and recognition in images and video frames," *Pattern Recogn.*, vol. 37, no. 3, pp. 595–608, Mar. 2004.
- [3] X. Chen, J. Yang, J. Zhang, and A. Waibel, "Automatic detection of signs with affine transformation," in *Proc. Workshop Application Computer Vision (WACV)*, Orlando, FL, 2002, pp. 32–36.
- [4] —, "Automatic detection and recognition of signs from natural scenes," *IEEE Trans. Image Process.*, vol. 13, no. 1, pp. 87–99, Jan. 2004.
- [5] P. Clark and M. Mirmehdi, "Estimating the orientation and recovery of text planes in a single image," in *Proc. 12th British Machine Vision Conf.*, Manchester, U.K., Guildford, U.K.: BMVA, Sep. 2001, pp. 421–430.
- [6] A. de la Escalera, J. M. Armingol, and M. Mata, "Traffic sign recognition and analysis for intelligent vehicles," *Image Vis. Comput.*, vol. 21, no. 3, pp. 247–258, Mar. 2003.
- [7] A. de la Escalera, J. M. Armingol, J. M. Pastor, and F. J. Rodriguez, "Visual sign information extraction and identification by deformable models for intelligent vehicles," *IEEE Trans. Intell. Transp. Syst.*, vol. 5, no. 2, pp. 57–68, Jun. 2004.
- [8] C.-Y. Fang, S.-W. Chen, and C.-S. Fuh, "Road-sign detection and tracking," *IEEE Trans. Veh. Technol.*, vol. 52, no. 5, pp. 1329–1341, Sep. 2003.
- [9] T. Gandhi, R. Kasturi, and S. Antani, "Application of planar motion segmentation for scene text extraction," in *Proc. Int. Conf. Pattern Recognition (ICPR)*, Barcelona, Spain, 2000, vol. I, pp. 445–449.
- [10] D. M. Gavrilu, U. Franke, S. Gorzig, and C. Wohler, "Real-time vision for intelligent vehicles," *IEEE Instrum. Meas. Mag.*, vol. 4, no. 2, pp. 22–27, Jun. 2001.
- [11] R. C. Gonzalez and R. E. Woods, *Digital Image Processing*. Reading, MA: Addison-Wesley, 1993.

- [12] E. D. Haritaoglu and I. Haritaoglu, "Real time image enhancement and segmentation for sign/text detection," in *Proc. Int. Conf. Image Processing (ICIP)*, Barcelona, Spain, 2003, vol. III, pp. 993–996.
- [13] A. K. Jain and B. Yu, "Automatic text location in images and video frames," *Pattern Recogn.*, vol. 31, no. 12, pp. 2055–2076, Dec. 1998.
- [14] V. Kastinaki, M. Zervakis, and K. Kalaitzakis, "A survey of video processing techniques for traffic applications," *Image Vis. Comput.*, vol. 21, no. 4, pp. 359–381, Apr. 2003.
- [15] K. I. Kim, K. Jung, and J. H. Kim, "Texture-based approach for text detection in images using support vector machines and continuously adaptive mean shift algorithm," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 25, no. 12, pp. 1631–1639, Dec. 2003.
- [16] C. W. Lee, K. Jung, and H. J. Kim, "Automatic text detection and removal in video sequences," *Pattern Recogn. Lett.*, vol. 24, no. 15, pp. 2607–2623, Nov. 2003.
- [17] H. Li, D. Doermann, and O. Kia, "Automatic text detection and tracking in digital video," *IEEE Trans. Image Process.*, vol. 9, no. 1, pp. 147–156, Jan. 2000.
- [18] R. Lienhart, "Automatic text recognition for video indexing," in *Proc. ACM Multimedia*, Boston, MA, Nov 1996, vol. 96, pp. 11–20.
- [19] R. Lienhart and A. Wernicke, "Localizing and segmenting text in images and videos," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 12, no. 4, pp. 256–268, Apr. 2002.
- [20] B. D. Lucas and T. Kanade, "An iterative image registration technique with an application to stereo vision," in *Proc. Int. Joint Conf. Artificial Intelligence (IJCAI)*, Vancouver, BC, Canada, 1981, pp. 674–679.
- [21] B. Luo, X. Tang, J. Liu, and H. Zhang, "Video caption detection and extraction using temporal information," in *Proc. Int. Conf. Image Processing (ICIP)*, Barcelona, Spain, 2003, vol. I, pp. 297–300.
- [22] *Manual on Uniform Traffic Control Devices (MUTCD) for Streets and Highways*, 2003 Edition, The Federal Highway Administration (FHWA), U.S. Dept. Transp. [Online]. Available: <http://mutcd.fhwa.dot.gov/>
- [23] G. Myers, R. Bolles, Q.-T. Luong, and J. Herson, "Recognition of text in 3-D scenes," in *Proc. 4th Symp. Document Image Understanding Technology*, Columbia, MD, Apr. 2001, pp. 23–25.
- [24] J. Miura, T. Kanda, and Y. Shirai, "An active vision system for real-time traffic sign recognition," in *Proc. IEEE Intelligent Transportation Systems*, Dearborn, MI, 2000, pp. 52–57.
- [25] J. Ohya, A. Shio, and S. Akamatsu, "Recognizing characters in scene images," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 16, no. 2, pp. 214–220, Feb. 1994.
- [26] G. Piccioli, E. De Micheli, P. Parodi, and M. Campani, "Robust method for road sign detection and recognition," *Image Vis. Comput.*, vol. 14, no. 3, pp. 109–223, Apr. 1996.
- [27] T. Sato, T. Kanade, E. K. Hughes, and M. A. Smith, "Video OCR for digital news archives," in *IEEE Int. Workshop Content-Based Access Image and Video Database*, Bombay, India, 1998, pp. 52–60.
- [28] J. Shi and C. Tomasi, "Good features to track," in *Proc. Computer Vision and Pattern Recognition (CVPR)*, Seattle, WA, 1994, pp. 593–600.
- [29] J. C. Shim, C. Dorai, and R. M. Bolle, "Automatic text extraction from video for content-based annotation and retrieval," in *Proc. Int. Conf. Pattern Recognition (ICPR)*, Brisbane, Australia, 1998, pp. 618–620. (SA22).
- [30] H. Veeraraghavan, O. Masoud, and N. P. Papanikolopoulos, "Computer vision algorithms for intersection monitoring," *IEEE Trans. Intell. Transp. Syst.*, vol. 4, no. 2, pp. 78–89, Jun. 2003.
- [31] S. Vitabile, A. Gentile, and F. Sorbello, "A neural network based automatic road signs recognizer," in *Proc. Int. Joint Conf. Neural Networks (IJCNN)*, Honolulu, HI, 2002, vol. 3, pp. 2315–2320.
- [32] V. Wu, R. Manmatha, and E. M. Riseman, "TextFinder: An automatic system to detect and recognize text in images," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 21, no. 11, pp. 1224–1229, Nov. 1999.
- [33] Y. U. Yim and S.-Y. Oh, "Three-feature based automatic lane detection algorithm (TFALDA) for autonomous driving," *IEEE Trans. Intell. Transp. Syst.*, vol. 4, no. 4, pp. 219–225, Dec. 2003.
- [34] D. Zhang and S. Chang, "A Bayesian framework for fusing multiple word knowledge models in videotext recognition," in *Proc. Computer Vision and Pattern Recognition (CVPR)*, Madison, WI, 2003, vol. 2, pp. 528–533.
- [35] Y. Zhong, H. Zhang, and A. K. Jain, "Automatic caption localization in compressed video," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 4, pp. 385–392, Apr. 2000.
- [36] X. Chen and A. L. Yuille, "Detecting and reading text in natural scenes," in *Proc. Computer Vision and Pattern Recognition (CVPR)*, Washington, DC, 2004, pp. 366–376.



Wen Wu (M'03) was born in Jiangsu Province, China. He received the B.S. degree in computer science from Tsinghua University, Beijing, China, in 2001, and the M.Sc. degree in computer science from the National University of Singapore, Singapore, in 2003. He is currently pursuing the Ph.D. degree at the School of Computer Science, Carnegie Mellon University, Pittsburgh, PA.

His research interests include statistical learning, multimedia, multimodal, and automotive safety applications.



Xilin Chen (M'00) received the B.S., M.S., and Ph.D. degrees from the Harbin Institute of Technology, Harbin, China, in 1988, 1991, and 1994, respectively, all in computer science.

He has been a Professor at the Harbin Institute of Technology since 1999. He was a Visiting Scholar at Carnegie Mellon University from 2001 to 2004. He joined the Institute of Computing Technology, Chinese Academy of Sciences, in August 2004. His research interests are image processing, pattern recognition, computer vision, and multimodal interface.

interface.

Dr. Chen is a member of the IEEE Computer Society. He has received several awards, including the National Scientific and Technological Progress Award in 2000 and 2003, respectively, for his research work. He has served as a Program Committee member for several international and national conferences.



Jie Yang (S'93–M'93) received the Ph.D. degree in electrical engineering from the University of Akron, Akron, OH, in 1994.

He is currently a Senior Systems Scientist at the School of Computer Science, Carnegie Mellon University, Pittsburgh, PA. He pioneered the hidden Markov model for human performance modeling in his Ph.D. dissertation research. He joined the Interactive Systems Laboratories in 1994, where he has been leading research efforts to develop visual tracking and recognition systems for multimodal human-computer interaction. He developed adaptive skin color modeling techniques and demonstrated software-based real-time face tracking system in 1995. He has been involved in the development of many multimodal systems in both intelligent working spaces and mobile platforms. He has been working on automatic detection of text from natural scenes over last six years, with applications to automatic sign translation and intelligent driving assistant systems. His current research interests include multimodal interfaces, computer vision, and pattern recognition.