An Adaptive Algorithm for Text Detection from Natural Scenes

Jiang Gao
Robotics Institute
Carnegie Mellon University
Pittsburgh, PA, 15213
E-mail: jgao@cs.cmu.edu

Jie Yang
Interactive Systems Laboratory
Carnegie Mellon University
Pittsburgh, PA, 15213
E-mail: yang+@cs.cmu.edu

Abstract

We present a new adaptive algorithm for automatic detection of text from a natural scene. The initial cues of text regions are first detected from the captured image/video. An adaptive color modeling and searching algorithm is then utilized near the initial text cues, to discriminate text/non-text regions. EM optimization algorithm is used for color modeling, under the constraint of text layout relations for a specific language. The proposed algorithm combines the advantages of several previous approaches for text detection, and utilizes a focus-of-attention approach for text finding. The whole algorithm is applied in a prototype system that can automatically detect and recognize sign input from a video camera, and translate the signs into English text or voice streams. We present evaluation results of our algorithm on this system.

1. Introduction

Text detection from a natural scene is useful in many applications. A well-known example is vehicle license detection and recognition. In this paper, we describe a more general application: detection of signs containing text from a natural scene.

A sign is something that suggests the presence of a fact, condition, or quality. In this research, we are interested in signs that have direct influence upon a tourist from a different country or culture. These signs include, at least, the following categories:

- Names: street, building, company, etc.
- Information: designation, direction, notice, etc.
- Commercial: announcement, advertisement, etc.
- Traffic: warning, limitation, etc.

Fully automatic extraction of signs is a challenging problem. Although text with a limited scope can be successfully detected using existing technologies, such as in high quality printed documents, it is difficult to detect signs with varying size, embedded in real world, and captured using an unconstrained video camera. Compared with other object detection tasks, many information sources are unavailable for a sign detection task, such as:

- No motion information: signs move together with background;
- No approximate shape information: text areas assume different shapes;
- No color reflectance information: signs assume different colors.

Languages impose another level of variation in text. For example, based on pictographic characters, the layout of Chinese characters in signs differs from the layout in European (phonological) languages. Handling Chinese characters layout requires an elaborate method of modeling. Figure 1 shows an example of a Chinese sign with both vertical and horizontal layouts as well as distortion because of warping.

There have been several research efforts toward automatic detection of text areas from general backgrounds[2-11]. Most of the previous researches focus on extracting text from pictures or video, though a few study are on character extraction from a natural scene, such as vehicle license plates detection (Cui and Huang 1997).

The existing approaches for text detection lie in the following categories:

- Edge filtering (Zhong and Jain 1995);
- Texture segmentation (Wu 1999);
- Color quantization (Jain and Yu 1998);
- Neural networks and bootstrapping (Lienhart 1996, Li and Doermann 1998).

Each of these approaches has its advantage/drawbacks concerning reliability, accuracy, difficulty in improvement and implementation.

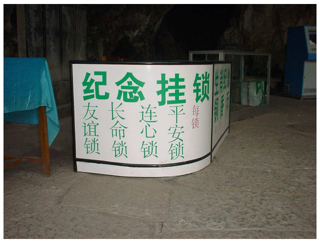


Fig. 1. A sign with deformation and complex layouts.

In this paper, we propose a new adaptive algorithm for text detection from a natural scene. This algorithm differs from the previous text detection approaches by utilizing a hierarchical algorithm structure, with different emphasis at each layer. At the first layer, we detect initial text cues based on edge filtering. The initial text cues provide the a priori information (location, size, and color) for the adaptive color segmentation algorithm at the second layer, which utilizes a Gaussian mixtures color modeling algorithm to adapt to varying conditions in a natural environment. The a priori information reduces the computational complexity of the color segmentation algorithm. Finally, we utilize language-specific layout relations as constraints to use with the text extraction algorithm based on color modeling. This focus-of-attention algorithm structure combines the advantages of several previous text detection approaches, and is more effective and robust to handle the dynamics of text detection from a natural scene.

We developed a prototype system that can recognize Chinese sign inputs from a video camera that is a common gadget for a tourist, and translate the signs into English text or voice stream. The text detection algorithm is utilized to detect signs in a natural scene automatically.

The organization of this paper is as follows: Section 1 describes challenges in text detection from a natural scene. In section 2 we propose the adaptive algorithm for sign detection. Section 3 gives layout analysis algorithm and criteria, which can be extended and utilized for multilingual text detection. Section 4 addresses issues in system development, and gives experimental results. Section 5 concludes the paper.

2. An adaptive approach for text detection

2.1. A hierarchical algorithm structure

We utilize a focus-of-attention approach for text detection by embedding an adaptive algorithm in a hierarchical algorithm structure, with different emphases at each layer. The sign detection algorithm consists of:

- A multi-scale edge detection algorithm;
- Adaptive searching and color modeling in the neighborhood of initial candidate text regions;
- Layout analysis of the detected text regions.

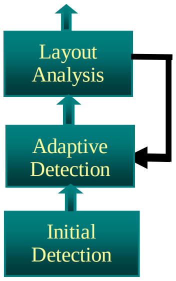


Fig. 2. The text detection algorithm.

The formulation of the algorithm is given below, using the following abbreviations: {*c*: Color, *p*: Position, *s*: Size, *t*: Text, t_0: The initially detected text}.

STEP1: Detect the initial text cues, i.e., determine regions such that $TextCue(c, p, s, t_0) = True$. For each region, let $SR0(c, p, s, t_0) = TextCue(c, p, s, t_0)$;

STEP2: Determine the search region $SR(c, p, s, t_0)$ in the neighborhood of $SR0(c, p, s, t_0)$; If for each (c, p, s, t_0), there is no $SR(c, p, s, t_0)$, such as $SR(c, p, s, t_0) \supset SR0(c, p, s, t_0)$, let $SR(c, p, s, t)$ = $SR0(c, p, s, t_0)$, go to STEP5; otherwise, continue;

STEP3: Perform color modeling in $SR(c, p, s, t_0)$, under the layout constraints, to extract characters or text regions in $SR(c, p, s, t_0)$;

STEP4: Update the $SR(c, p, s, t_0)$ to $SR(c, p, s, t)$, t is the character/text in $SR(c, p, s, t_0)$ extracted in STEP3. Delete the search regions that have no characters or text in it, and let $SR0(c, p, s, t_0)$ = $SR(c, p, s, t)$, go to STEP2;

STEP5: Perform layout analysis on $SR(c, p, s, t)$, output the detected text regions.

The proposed algorithm framework is depicted in Figure 2. We explain the details in the following sections.

2.2. Multi-resolution edge detector

In the first layer of the framework, an edge detection algorithm is utilized to obtain the initial candidates of text regions under varying lighting conditions. We use a multi-resolution approach to compensate other variations, such as noise and contrast. The multi-scale edge detector first computes edge filtering:

$$g(x, y) = D[Gauss(x, y)] * f(x, y), \qquad (1)$$

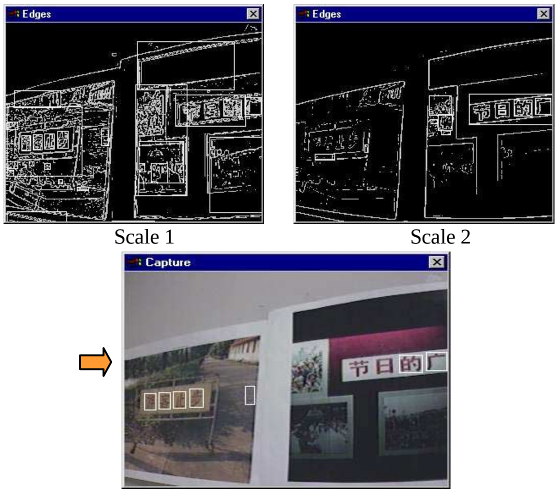
where $D[\cdot]$ is derivative function, $f(x, y)$ is the input pixel intensity at position (x, y). Eq. (1) combines image smoothing with edge filtering.

After edge filtering, threshold detection is conducted on $g(x, y)$ to find the edges. We control the sensitivity (scale) of the edge detector by the width of the Gaussian function $Gauss(x, y)$ used in (1) and the threshold for edge detection from $g(x, y)$.

After edge detection, we perform edge clustering, and use several criteria, such as aspect ratio, existence of pairs of rising and falling edges, etc., to find the text cue regions with property $(Color, Position, Size)$ such that

$$TextCue(c, p, s) = True. \qquad (2)$$

Figure 3 illustrates this algorithm. The two signs in the left and right have different contrast and lighting conditions, but can be optimally detected using the edge detector in different scales. The integration result is obtained by combining the detection results from different scales.



Scale 1

Scale 2

Integration

Fig. 3. Initial detection using edge detectors.

2.3. Adaptive searching and color modeling

The second layer of the algorithm performs adaptive search and color modeling near the initially detected text cues to discriminate text/non-text regions and find all the text regions in the image.

2.3.1. Adaptive searching based on layout syntax. The adaptive searching strategy determines the search region by utilizing information of the initial candidates detected by the first layer, and layout of the texts. We discuss these issues in some more details in the following.

Since Chinese language is rather character-based than word-based, Chinese texts in both horizontal and vertical directions are in common use. Some special signs are designed in special shapes for aesthetics reasons, but we will ignore these layouts at this stage. The direction control is formulated as follows: Given $SR0(c, p, s, t_0)$, perform layout analysis on t_0 . Then, the search region $SR(c, p, s, t_0)$ is determined as:

$$SR(c, p, s, t_0) = SR0(c, p, s, t_0) \cup ER(c, p, s, t_0) . \quad (3)$$

In (3) $ER(c, p, s, t_0)$ represents all the possible text extension regions based on $SR0(c, p, s, t_0)$ and the layout direction of t_0 in $SR0(c, p, s, t_0)$.

In addition to layout directions, we use shape and color criteria to discriminate different signs. The characters in the same sign tend to have similar appearance, such as using similar font, color, and size. Considering this, the $ER(c, p, s, t_0)$ in the refined adaptive searching algorithm represents all the possible text extension regions based on the layout direction of *each text regions* in $SR0(c, p, s, t_0)$.

Using these heuristics, we designed the searching strategy and criteria. The text layout constraints plays the similar role as syntax in language understanding when parsing sentences, except that it is used to discriminate different layouts to assist the adaptive searching of text regions.

2.3.2. Gaussian mixtures based color modeling. Now we have the searching region and direction control, and also have criteria to discriminate different signs. The next step is to extract the texts from the search regions. Texts extraction can be a simple case or a complex case.

In a simple case, the color or intensity of text in a sign does not change significantly. In this case, we can use color information of the initially detected text to extract characters with similar attributes, under constraints of the layout syntax. In some situations, however, colors within a sign region change dramatically. We use an adaptive color modeling algorithm to solve the problem.

We use Gaussian mixtures for adaptive color modeling. In this way we can model both the text region and background using arbitrary number of basis functions (clusters). The Gaussian mixtures model utilized is as follows: The probability distribution of a *D*-dimensional color vector *x* is represented as a weighted mixture of *K* basis functions (components) as:

$$p(x) = \sum_{k=1}^{K} w_k \cdot \alpha_k(x) , \quad (4)$$

w_k is the mixture weight. The basis functions $\alpha_k(x)$ are chosen to be Gaussians of the form:

$$\alpha(x) = \frac{1}{(2\pi)^{D/2} |C|^{1/2}} \exp\{-\frac{1}{2}(x - m_x)^T C^{-1} (x - m_x)\} . \quad (5)$$

m_x and C are the expectation and covariance matrix *x*. $|C|$ is the determinant of *C*.

In Gaussian mixtures color modeling, we use the basis functions to represent regions of different color property. Given the model parameters, the probability that the region *k* being responsible for generating pixel with color vector *x* can be computed as:

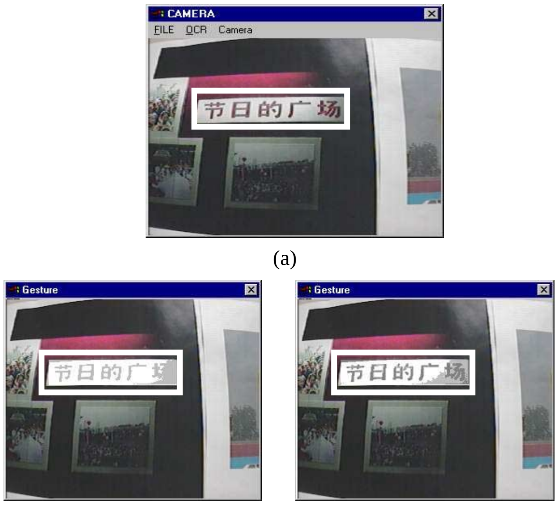
$$\gamma_k(x) = \frac{w_k \cdot \alpha_k(x)}{p(x)} . \quad (6)$$

An EM algorithm is utilized to determine the optimal parameters C_k and w_k . Given the number *K* of Gaussian mixtures in the region, the EM algorithm maximizes the likelihood:

$$L = \prod_{n=1}^{N} p(x) . \quad (7)$$

N is the number of pixels in search region. The model parameters are initialized using fast clustering algorithm.

A crucial problem of this method is how to adaptively determine the number of Gaussian mixtures *K*, since color composition differs in different locations.



(a)

(b) (c)

Fig. 4. Adaptive character/text extraction using color modeling: (a) Original sign, (b) Color space modeled by two Gaussian mixtures, (c) Color space modeled by three Gaussian mixtures.

We determine the number of Gaussian mixtures adaptively by considering if the characters/text can be extracted from the background, and the characters/text should satisfy the layout syntax as described in section 3. Specifically, the algorithm is given below in Figure 5.

A result of the color modeling algorithm is given in Figure 4. In Figure 4, (b) and (c) are color segmentation results using two and three Gaussian mixtures, respectively. We are interested in extracting texts from the rectangle area. The rightmost character in the rectangle area is confused with the background if using only two Gaussin mixtures, but can be extracted using three Gaussian mixtures.

```
Let $K = 2$;
Do {
        Extract character/text regions based on $K$
        Gaussian mixtures color modeling, until there is
        no $SR_K(c, p, s, t)$, such that:
        $SR_K(c, p, s, t) \supset SR0_K(c, p, s, t_0)$;
        If $SR_K(c, p, s, t) \supset SR_{K-1}(c, p, s, t)$
            FLAG =1;
        If ( $K < K_{\max}$ ) AND (FLAG =1)
            $K = K + 1$;
        Else
            $K_{final} = K$, break;
}
```

Fig. 5. Algorithm for determine number of Gaussian mixtures *K*.

3. Layout relation analysis

Layout relation is the way characters align with each other in written languages. Figure 6 is an example of Chinese sign. Each character in the sign is composed of several connected sub-components. Sometimes the sub-components align in the same way as characters in a sign. However, such a sign poses a problem to automatic text layout analysis: how can a system know if a text region is a character or only segment of a character, without recognition of the whole text area?

We use layout analysis techniques, which utilize various criteria, to deal with this problem. The criteria used include (in which C1-C5 are constants determined empirically, and other symbols are defined in Figure 7):

1. The components should be near to each other:

$$Dist(i, j) / \max(Width(i), Width(j)) < C1. \qquad (8)$$

2. The aspect ratios (AP) of the components fall in a certain range:

$$C2 < AP_i < C3. \qquad (9)$$

3. After assigning character labels to individual components, The components should align with each other:

$$Align(i, j) > C5, \qquad (10)$$

$$Align(i, j) = C4 \cdot \frac{Ol(i, j)}{\max(Height_i, Height_j)}, \qquad (11)$$

and utilize most of the components. In (11) $Ol(i, j)$ is the length of overlaid part of the component *i* and component *j*.

In our system, the criteria are designed to allow tolerance to considerable slanting of sign images, non-uniform character sizes, and images captured from unfavorable angles.

Layout relation is not only used to extract different text regions in the final stage, but also used in the adaptive algorithm for color modeling and character/text extraction: In each adaptive searching and color modeling step of the adaptive text extraction algorithm, layout analysis is performed after character/text extraction, and layout analysis results are used to determine the search region $SR(c, p, s, t_0)$ in the next iteration.

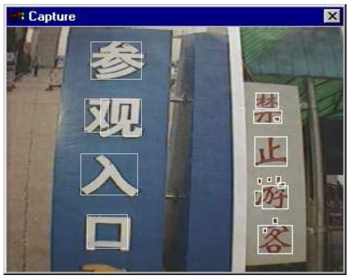


Fig. 6. Multi-segment Chinese characters.

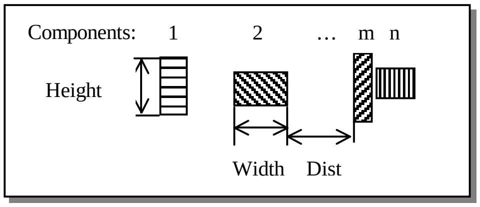


Fig. 7. Layout relation: definitions.

Layout relations are language-specific, especially for languages with significantly different features, such as Chinese and English. By changing the layout relations, the detection algorithm can be applied to multilingual texts, as demonstrated in section 4.

In most cases, applying these criteria can find the true sign regions correctly, though there are some situations where several sub-components of the characters in a sign will also be considered as a sign area. Since there do exist some cases where a big sign contains a smaller one in it, we decide that this type of errors is to be solved by human-computer interaction.

4. Experimental results

4.1. System architecture

We developed a system for sign detection, recognition and translation. The system can help international tourists to overcome language barriers. The system structure is shown in Figure 8. A video camera is used to capture the images. The video stream or picture is then input to the sign detection module to find the text regions. The sign detection results are displayed to allow user interaction with the system.

These text regions are further processed and fed into the OCR engine, which recognizes the contents of the sign areas in the original language. Then, the recognition results are sent to the translation module to obtain an interpretation in target language. Presently we use the EBMT software (Brown 1996) for translation.

Figure 9 is the current user interface of the system, and shows an example of automatic sign detection and translation. Figure 10 is an intermediate result of image preprocessing/binarization for OCR module, and translation. For a more detailed description of system implementation, including recognition and translation, please refer to (Gao, 2001).

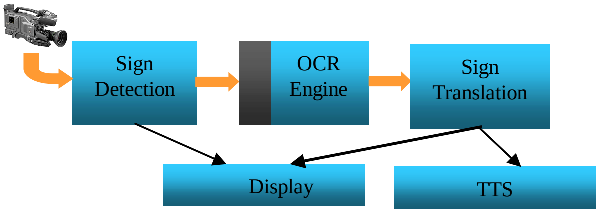


Fig. 8. System architecture.

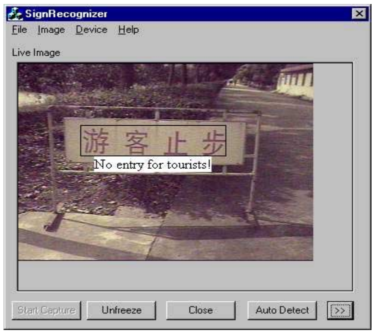
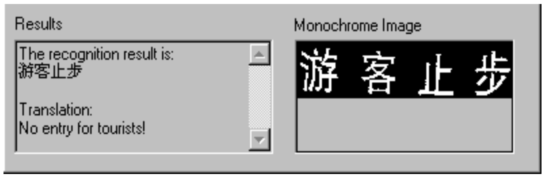


Fig. 9. Screen shot of the user interface.



The recognition result is:
游客止步

Translation:
No entry for tourists!

Fig. 10. Preprocessing for OCR, and translation.

4.2. System evaluation

We evaluated our text detection algorithm on the system. The overall performance of the algorithm on our sign database is given in Table 1.

We tested the robustness of the detection algorithms by changing conditions such as image resolution, camera view angle, and lighting conditions. The algorithm worked fairly well for low resolution images (e.g., from 320 x 240 to 80 x 60). Some detection results are given in Figure 11–Figure 16. The rectangles indicate the detected text regions. Both examples in Figure 15 and Figure 16 are in English. The algorithms can also effectively extract signs in other languages, by making some changes on the definition of layout relations. Figure 12 gives an example of false alarms.

By properly selecting parameters, we can control the ratio of miss detections and false alarms. Presently, such parameters are selected according to user's preferences, i.e. acceptability of different types of errors from users' point of view.

There are several ways to further improve the sign detection accuracy. For example, it is possible to eliminate false alarms by combining sign detection with OCR. The confidence of the text detection algorithm can be improved by incorporating the OCR engine in an early stage.

Table 1. Overall performance

Total text regions in database	Detected without missing characters	Detected with missing characters	False alarms	Not detected
823	93.3%	5.9%	10.1%	0.8%

5. Conclusions

We present an algorithm for automatic text detection from natural scenes. The algorithm is utilized in a sign detection and translation system to assist international tourists overcome language barriers. We propose a hierarchical algorithm structure, with different emphasis at each layer. The algorithm framework considers critical challenges in text detection and can detect signs robustly under different conditions. The algorithm can be extended to handle text detection in other languages by changing the layout constraints.

Acknowledgements

The authors would like to thank reviewers for their comments on this paper. This work was performed at Interactive Systems Laboratory, CMU, and was partially supported by DARPA under TIDES project.

References

[1] Brown, R.D., “Example-based machine translation in the pangloss system”, *Proceedings of the 16th International Conference on Computational Linguistics*, pp. 169-174, 1996.

[2] Cui, Y. and Q. Huang, “Character extraction of license plates from video”, *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 502-507, 1997.

[3] Jain, A.K. and B. Yu, “Automatic text location in images and video frames”, *Pattern Recognition*, vol. 31, no. 12, pp. 2055-2076, 1998.

[4] Gao, J., J. Yang, Y. Zhang, and A. Waibel, “Text detection and translation from natural scenes”, Tech. report CMU-CS-01-139, School of Computer Science, Carnegie Mellon University, June, 2001.

[5] Li, H. and D. Doermann, “Automatic identification of text in digital video key frames”, *Proceedings of IEEE International Conference of Pattern Recognition*, pp. 129-132, 1998.

[6] Lienhart, R., “Automatic text recognition for video indexing”, *Proceedings of ACM Multimedia 96*, pp. 11-20, 1996.

[7] Ohya, J., A. Shio and A. Akamatsu, “Recognition of characters in scene images”, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 16, no. 2, pp. 214-220, 1994.

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

[9] Wong, E. K. and M. Chen, “A robust algorithm for text extraction in color video”, *Proceedings of IEEE Int. Conference on Multimedia and Expo (ICME2000)*, 2000.

[10] Wu, V., R. Manmatha, and E.M. Riseman, “Textfinder: an automatic system to detect and recognize text in images”, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 21, no. 11, pp. 1224-1229, 1999.

[11] Zhong, Y., K. Karu, and A.K. Jain, “Locating text in complex color images”, *Pattern Recognition*, vol. 28, no. 10, pp. 1523-1536, 1995.

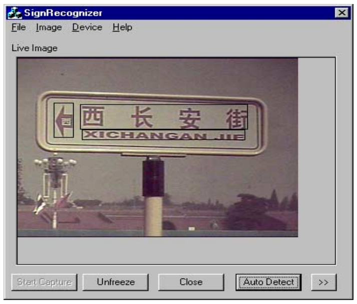


Fig. 11

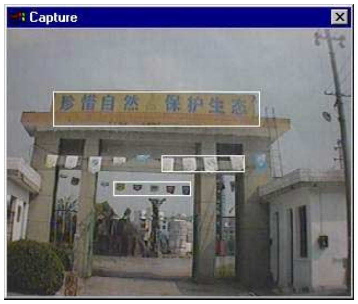


Fig. 12

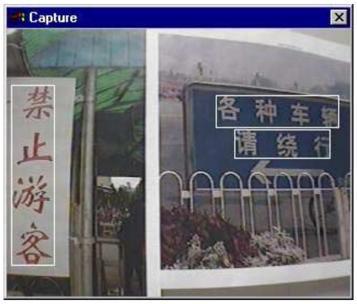


Fig. 13

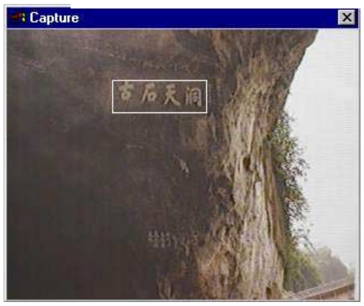


Fig. 14

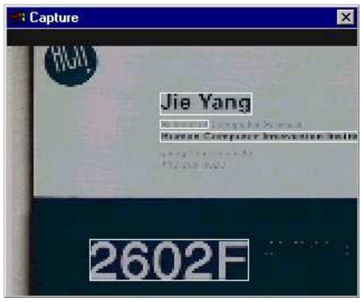


Fig. 15

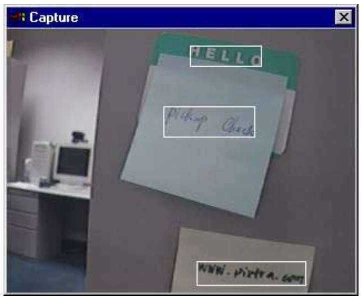


Fig. 16

Figure 11-Figure 16. Examples of sign detection.
11: layout analysis. 12: False alarms. 13: lighting and slant I. 14: lighting and slant II. 15: Low resolution. 16: Handwritting.