Automatic Detection of Signs with Affine Transformation

Xilin Chen[1], Jie Yang[1], Jing Zhang[2], and Alex Waibel[1]

[1]*Interactive Systems Lab, School of Computer Science, Carnegie Mellon University*
[2]*Mobile Technologies, LLC*

xlchen@cs.cmu.edu, yang+@cs.cmu.edu, jzhang@mobytrans.com, ahw@cs.cmu.edu

Abstract

In this paper, we propose an approach for detecting signs from natural scenes. The approach efficiently embeds multi-resolution, adaptive search, and affine rectification algorithms in a hierarchical framework, with different emphases at each layer. We combine multi-resolution and multi-scale edge detection techniques to effectively detect text in different sizes. By using the cues from text inside the image, we introduce affine rectification transformation to recover deformation of the text region caused by an inappropriate camera view angle. This procedure can significantly improve text detection rate and OCR (Optical Character Recognition) accuracy. Experimental results have demonstrated feasibility of the proposed algorithms. We have applied the proposed approach to a Chinese sign translation system, which can automatically detect Chinese text input from a camera, recognize the text, and translate the recognized text into English or voice stream.

1. Introduction

Signs are good examples of objects in natural environments that have high information content. A sign is an object that suggests the presence of a fact. It can be a displayed structure bearing letters or symbols, used to identify or advertise a place of business. It can also be a posted notice bearing a designation, direction, safety advisory, or command. Signs are everywhere in our lives. They make our lives easier when we are familiar with them, but they pose problems or even danger when we are not. For example, a tourist might not be able to understand a sign in a foreign country that specifies warnings or hazards. Automatic sign translation, in conjunction with spoken language translation, can help international tourists to overcome these barriers. The objective of this research is to develop a robust system that can automatically detect text signs from natural scenes for sign translation tasks.

Automatic detection of text from natural scenes is a prerequisite for automatic sign recognition and translation. An initial challenge of the task comes from deformation of sign regions caused by an undesired viewing angle of the camera. In this paper, we propose an approach for detection and rectification of text from natural scenes. Compared with the existing text detection algorithms, this framework can better handle the dynamics of text detection in natural scenes. We have applied this approach to a Chinese sign translation system, which can automatically detect Chinese signs input

from a camera, recognize, and translate the recognized text into English or voice stream.

The rest of this paper is organized as follows: In section 2, we describe problems and framework of our proposed approach. In section 3, we discuss multi-resolution method for text detection. In section 4, we introduce affine rectification technique based appearance of detected text. In section 5 we present some experiments on the proposed approach and evaluation results. Finally, we conclude the paper.
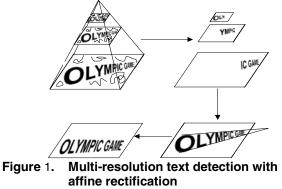
2. Problems and Framework

Automatic detection of text from natural scenes is a very difficult task. The primary challenge lies in variations of text: it can vary in font, size, orientation, and position of text, be blurred from motion, and be occluded by other objects. Originating in 3-D space, text as signs in scene images can be distorted by slant, tilt, and shape of objects on which they are found [11]. In addition to the horizontal left-to-right orientation, other orientations include vertical, circularly wrapped around another object, slanted, sometimes with the characters tapering (as in a distinct angle away from the camera), and even mixed orientations within the same text area, such as text on a T-shirt or wrinkled sign.

The work is related to the existing research in text detection from general backgrounds [4, 11, 14], video OCR [12], and recognition of text on special objects such as license plates and containers [1, 2, 5, 6, 10].

"Video OCR" was motivated by digital library and visual information retrieval tasks. In such a text detection and recognition task, an image sequence provides a lot of useful information that can be used to detect text and enhance the image's resolution [7, 8, 12, 13]. Compared with video OCR tasks, text detection under natural scene for recognition and translation purpose faces more challenges. The user's movement can cause unstable input images. Non-professional equipment can make the video input poorer than that of other video OCR tasks.

In order to address challenges of text detection from natural scenes, we use a three-layer hierarchical framework, with different emphases at each layer. In this three-layer structure framework, the first layer detects possible text regions. Once having the initial candidates, the approach uses the local information, such as color and shape, provided by the first layer to adaptively search the neighborhood of candidates in the second layer. The advantage of this strategy is that we can use the locality and color information provided

by the first layer to refine the detection results. The third layer performs layout analysis. Elaborate layout analysis algorithms are essential to achieve a high detection rate. The layout of a sign is usually language dependent. The major contribution of this framework lies in its flexibility and ability to refine the detection results by providing context information. We can focus on different emphases at the different levels and incrementally incorporate more information into the framework to improve detection accuracy.

Our objective is to automatically detect text from natural scenes for a recognition and translation task. For clearly segmented printed materials, state-of-the-art techniques offer virtually error-free OCR for several important alphabetic systems, such as Latin, Greek, etc. However, when the number of character set in the language is large, such as in the Chinese or Korean writing systems, or the characters are not separated from one another, as in Arabic or Devanagari print, the error rates of OCR systems are still far from that of human readers, and the gap between the two is exacerbated when the quality of the image is compromised, e.g., input using a video or image camera. In a sign recognition and translation task, an image of text can be deformed because of an inappropriate camera view angle. Such deformation can be, in fact, recovered by affine rectification. The basic idea is as follows: we perform a coarse layout analysis and estimate basic geometry parameters of sign regions after the coarse detection. We then do affine rectification for each sign region in the image. Finally, we perform text detection again in the rectified regions of the image to obtain refined detection results. This schema is shown as Figure 1.

Figure 1. **Multi-resolution text detection with affine rectification**

In next two sections we will discuss multi-resolution and affine rectification algorithms in more detail.

3. Multi-resolution Text Detection

The objective of the detection in the first layer is to avoid missing any candidate characters. Since lighting and contrast vary dramatically in a natural environment, detection algorithms need to perform reliably under different circumstances. The size of a character is dependent on many factors such as text font, distance from the camera, and camera view angle. Large variation in these effects can cause the failure of automatic detection: large characters can be mistakenly considered as the background and small ones can

be missed. In order to avoid the problem, we perform detection at various resolutions of the image by building a pyramid from the image. The multi-resolution approach can address the problem of changing scales. The small characters can be detected at the detailed level while the large ones may be considered to be background, and the large characters can be found at a coarse level while the small ones will be ignored. By combining the detection results from different resolutions of the image, the algorithm can correctly detect all characters of different sizes.

Two different methods have been successfully used for detecting text in an image. One is based on analyzing certain features in an area, e.g., texture and color analysis [4, 14]. DCT (Discrete Cosine Transformation) and wavelet transformation are widely used for area analysis [7, 9]. A major advantage of the DCT area analysis method is that DCT coefficients can be obtained directly from the JPEG or MPEG image, while the wavelet transformation can provide more stable features compared with DCT method. A disadvantage of area-based methods is that they are sensitive to lighting and character-scale changes. They often fail to detect text if the size is too large or too small. The other method is based on edges [17], which can provide more stable features and is more suitable for text detection from natural scenes. However, we have to pay special attention to filtering noises, because noises can add extra edges.

We currently use four different features, edge, position, size and texture to determine candidates. Within each feature, we also incorporate measures for enhancing robustness. We utilize a multi-resolution approach to compensate for variations and noise in the edge detection algorithm. We apply the edge detection algorithm with different scale parameters and then fuse the results from different resolutions. We use DOG (Difference of Gaussian) operator to get edges. There are different kinds of edges, such as step edge and roof edge. A character stroke can be considered as a roof edge at a coarse scale and also can be considered as a serial of positive step edges followed by a serial of negative ones at fine scale. Therefore, we omit the continuously positive or negative step edges to get the stroke intensity.

The text detection algorithm for initial candidates at each layer is as following:

1. Get edges using DOG operator;
2. Get the surround rectangle of each continuous edge R_i, and calculate the texture intensity, color distribution, and contrast;
3. Merge the surround rectangles, and update the attributes, if $R_i \cap R_j \neq \Phi$ or $SR_i \cap R_j \neq \Phi$, and they have the similar intensity and color distribution and contrast, where SR_i is the extended areas of R_i;
4. Calculate the edge projections on both horizontal and vertical directions for each candidate area, and add them to the attributes.
5. Fuse these areas using all the above attributes.

A more detail description on the candidate area selection can be found in [3].

4. Affine Text Restoration

The objective of layout analysis is to align characters in an optimal way, so that characters belong to the same context will be aligned together. Usually, the text layout has some cluster features. Each character in the same context will have almost same background, front color, contrast, and the similar color distribution. These characters usually align to form a row-based or column-based structure. Even if geometry of a sign is deformed, text on the sign can still keep in line. However, the line may be tilted in a certain degree, which can bring problems to layout analysis and further recognition. In order to recover the text from deformation, we can use some cues provided by the tilted lines.
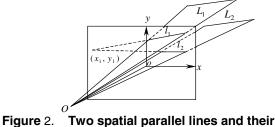
In our approach, every candidate character has 4 attributes attached: 1) geometry size, 2) intensity and color distribution, 3) edge intensity, 4) location. A cluster-based algorithm is applied to get a coarse detection based on these attributes, and then the Hough transformation is use to approach the text direction, and finally to recover the normal direction of the text plane. As shown in Figure 2, suppose that we have the following two lines l_1, l_2 associated with text in the image plane, which are mapped from space parallel lines L_1 and L_2.

$$l_i : a_i x + b_i y + c_i = 0 \quad i = 1,2 . \tag{1}$$

We can get the normalized space direction of L_1 and L_2 as:

$$\begin{pmatrix} r_1 \\ s_1 \\ t_1 \end{pmatrix} = \begin{cases} \dfrac{1}{\sqrt{a_1^2 + b_1^2}} (b_1 \quad -a_1 \quad 0)^T, & \text{if } l_1 /\!/ l_2 \\ \dfrac{1}{\sqrt{x_1^2 + y_1^2 + f^2}} (x_1 \quad y_1 \quad f)^T, & \text{otherwise} \end{cases}, \tag{2}$$

where f is the focal length of the camera, and can be obtained from calibration, and we also assume that the focal length is much smaller than the distance from the object to the camera. (x_1, y_1) is the intersection point of l_1 and l_2 when they are not parallel in the image. We must keep $r_1 > 0$, otherwise, the direction of this vector needs to be inverted.

Figure 2. **Two spatial parallel lines and their images in the image plane**

If we can only find one pair space parallel lines, we can only partly recover the normal direction of this text plane. If we can find another pair space parallel lines on the text plane, which are not parallel to the first pair in 3-D space, we can recover the normal of this text plane. Usually, we can get the second pair parallel line from following three cues:

1. Most of signs have a rectangle frame, which can provide two pair parallel lines.
2. For a sign with text in more than one rows, we can also get the second pair space parallel lines from two aligned left and right text sides.
3. If only one row's text can be used, we can also estimate the second pair parallel lines by measuring the differences between the left-most and right-most characters in the same text row for the top and bottom lines of the text. Then the second pair space parallel lines can be obtained from the following equation:

$$l_i' : a_i' x + b_i' y + c_i' = 0 \quad i = 1,2 . \tag{3}$$

Similar to Equation (1), we can get the normalized space direction as Equation (4).

$$\begin{pmatrix} r_2' \\ s_2' \\ t_2' \end{pmatrix} = \begin{cases} \dfrac{1}{\sqrt{a_1'^2 + b_1'^2}} (b_1' \quad -a_1' \quad 0)^T, & \text{if } l_1' /\!/ l_2' \\ \dfrac{1}{\sqrt{x_2^2 + y_2^2 + f^2}} (x_2 \quad y_2 \quad f)^T, & \text{otherwise} \end{cases}, \tag{4}$$

where (x_2, y_2) is the intersection point of l_1' and l_2' when they are not parallel in the image. We also must keep $s_2' > 0$, otherwise, invert the vector. Then, we can have the normal of the text plane as:

$$(r_3 \quad s_3 \quad t_3)^T = (r_1 \quad s_1 \quad t_1)^T \times (r_2' \quad s_2' \quad t_2')^T . \tag{5}$$

To guarantee $(r_2' \quad s_2' \quad t_2')^T$ orthotropic with the other two vectors, we need to perform:

$$(r_2 \quad s_2 \quad t_2)^T = (r_3 \quad s_3 \quad t_3)^T \times (r_1 \quad s_1 \quad t_1)^T . \tag{6}$$
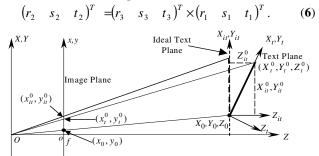

Figure 3. **Illustration of four coordinate systems**

It is possible to reconstruct a front view of the sign if we know the normal of the sign plane under the camera coordinate system. Figure 3 depicts an image in four coordinate systems:

1. The camera coordinate system, $OXYZ$, is the basic coordinate system.
2. The text plane coordinate system, $O_t X_t Y_t Z_t$, applies the text plane as $O_t X_t Y_t$ plane, and uses $(r_1 \quad s_1 \quad t_1)^T$, $(r_2 \quad s_2 \quad t_2)^T$ and $(r_3 \quad s_3 \quad t_3)^T$ as its axes. The origin of the system can be selected randomly at the $O_t X_t Y_t$ plane, which is located at (X_0, Y_0, Z_0) under the $OXYZ$ coordinate system. However, it is desirable that we select a point at a

character that is as close to the origin of *OXYZ* as possible.

3. The ideal text plane, $O_{it}X_{it}Y_{it}Z_{it}$, locates at the same origin as $O_t X_t Y_t Z_t$ but uses the vectors (1 0 0), (0 1 0) and (0 0 1) as its axes.

4. The image coordinate system, *oxy*, is a 2-D coordinate system while all other three are 3-D coordinate systems.

The mapping that maps a point $\left(X_t^0, Y_t^0, Z_t^0\right)$ from text plane $O_t X_t Y_t Z_t$ onto a point (x_t^0, y_t^0) in the image coordinate system can be written as:

$$\begin{pmatrix} x_t^0 \\ y_t^0 \end{pmatrix} = \begin{pmatrix} f & 0 \\ 0 & f \end{pmatrix} \begin{pmatrix} \left(X_{it}^0 + X_0\right)/\left(Z_{it}^0 + Z_0\right) \\ \left(Y_{it}^0 + Y_0\right)/\left(Z_{it}^0 + Z_0\right) \end{pmatrix}, \quad (7)$$

$$\begin{pmatrix} X_{it}^0 \\ Y_{it}^0 \\ Z_{it}^0 \end{pmatrix} = \begin{pmatrix} r_1 & r_2 & r_3 \\ s_1 & s_2 & s_3 \\ t_1 & t_2 & t_3 \end{pmatrix} \begin{pmatrix} X_t^0 \\ Y_t^0 \\ Z_t^0 \end{pmatrix}. \quad (8)$$

In order to reconstruct a front view of the text plain, we can use an affine rectification:

$$\begin{pmatrix} x_{it}^0 \\ y_{it}^0 \end{pmatrix} = \begin{pmatrix} f & 0 \\ 0 & f \end{pmatrix} \begin{pmatrix} \left(X_t^0 + X_0\right)/Z_0 \\ \left(Y_t^0 + Y_0\right)/Z_0 \end{pmatrix}. \quad (9)$$

Considering the text plane is on the $O_t X_t Y_t$ plane, we have $Z_t^0 = 0$. Since the origin $\left(X_0, Y_0, Z_0\right)$ of both $O_t X_t Y_t Z_t$ and $O_{ti} X_{it} Y_{it} Z_{it}$ maps onto the image plane as $\left(x_0, y_0\right)$, we have:

$$x_0 = f \frac{X_0}{Z_0}, \, y_0 = f \frac{Y_0}{Z_0} \, . \quad (10)$$

From Equation (7) to (10), we can obtain:

$$x_{it}^0 = x_0 + f \frac{\begin{vmatrix} x_0 & -x_t^0 & r_2 f - t_2 x_t^0 \\ y_0 & -y_t^0 & s_2 f - t_2 y_t^0 \end{vmatrix}}{\begin{vmatrix} r_1 f & -t_1 x_t^0 & r_2 f - t_2 x_t^0 \\ s_1 f & -t_1 y_t^0 & s_2 f - t_2 y_t^0 \end{vmatrix}}, \quad (11)$$

where

$$y_{it}^0 = y_0 + f \frac{\begin{vmatrix} r_1 f & -t_1 x_t^0 & x_0 - x_t^0 \\ s_1 f & -t_1 y_t^0 & y_0 - y_t^0 \end{vmatrix}}{\begin{vmatrix} r_1 f & -t_1 x_t^0 & r_2 f - t_2 x_t^0 \\ s_1 f & -t_1 y_t^0 & s_2 f - t_2 y_t^0 \end{vmatrix}}. \quad (12)$$

The equation (11) and (12) can be used to restore the front view of the text from an affined text image. A proper interpolation should be applied because it is not a one-to-one mapping for the digitalized image. We use a B-spline interpolation in our current implementation. To avoid additional blur in the interpolation, all edge pixels are interpolated only along the edge direction while the other points are used for surface interpolations. Figure 4 is an example of a restoration mapping from a deformed image. Figure 4(a) is an affined bulletin board, Figure 4 (b) is the

recovered one without interpolation and Figure 4(c) and (d) are the recovered ones with B-spline interpolations using different reference origins in the text plane.

(a) The deformed image   (b) Affine rectified image without interpolation

(c) Bottom-right point as origin (d) Bottom-left point as origin

Figure 4.   **An example of sign restoration from an affined image**

5. Experiments and Discussion

We have performed experiments to evaluate the proposed approach. The experimental results indicate that rectification of the affine signs can significantly improve text detection rate and recognition accuracy. We first tested affine rectification ability of the system using outdoor images taken from different viewpoints. Figure 5 is two examples. The first row is the original images, and the second row is rectified images from the original ones.
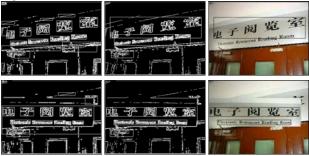

Figure 5.   **Experiments of sign affine rectification**

(a) Character candidates   (b) Coarse layouts   (c) Final results

Figure 6.   **An example of text detection with/ without affine rectification**

IEEE
COMPUTER
SOCIETY

We then tested performance of text detection using affine rectification. Figure 6 and Figure 7 are examples of text detection without and with affine rectification. It can be observed that only part of text is detected in the first row in Figure 6 and Figure 7. For the rectified signs in the second row, the text has similar size and is almost in horizontal direction. The algorithm has found the text successfully, even for a sign that contains different languages, such as what is in Figure 7.

(a) Character candidates   (b) Coarse layouts   (c) Final results

Figure 7.   **Another example of text detection with / without affine rectification**

We further tested improvement of OCR accuracy with affine rectification. Experiments were performed on Chinese characters, and a OCR system was used for the test. Fifty images were randomly selected from our Chinese sign database (more than 2000 sign images), and twenty of them have more or less deformations. These images include total 251 characters. Results are listed in Table 1, from which we can observe that improvement to OCR accuracy is significant. Without affine rectification, the recognition rate decreases rapidly as the angle between normal of text plane and Z-axis of camera coordinate system increases. The restoration can improve the results significantly, especially when the angle from 30° to 50°.

Table 1**. Comparison of recognition results**

	Total Characters	Recognized Characters	
		Before affine rectification	After affine rectification
$\gamma<20°$	24	20	21
$30°>\gamma\geq20°$	48	41	43
$40°>\gamma\geq30°$	46	37	42
$50°>\gamma\geq40°$	88	36	73
$\gamma\geq50°$	45	23	31

Note: γ is the angle between normal of text plane and Z-axis of camera

6. Conclusion

In this paper, we have presented a novel approach for detecting text from natural scenes. The approach efficiently embeds multi-resolution, adaptive search, and affine rectification algorithms in a hierarchical framework, with different emphases at each layer. The affine rectification algorithm has bridged a certain gap between 2D text detection algorithms and 3D real world. We have demonstrated that the rectification procedure can significantly improve text detection rate and OCR accuracy.

We have successfully applied the proposed approach to automatic sign translation systems. The prototype system can automatically translate Chinese sign into English on different platforms including palm-size PDAs [15, 16].

References

[1] Barnes, E., Image recognition for shipping container tracking and I.D, Advanced Imaging, Vol. 10, No.1 pp. 61-62, 1995.

[2] Cui, Y. and Huang, Q., Character Extraction of License Plates from Video. Proc. of CVPR, pp. 502-507, 1997.

[3] Gao, J., Yang, J., Zhang, Y., and Waibel, A., Text Detection and Translation from Natural Scenes, CMU-CS-01-139, 2001.

[4] Jain, A.K. and Yu, B., Automatic text location in images and video frames. Pattern Recognition, Vol. 31, No. 12, pp. 2055-2076, 1998.

[5] Kumano, S., Miyamoto, K., Tamagawa, M., Ikeda, H., and Kan, K., Development of container identification mark recognition system, Transactions of IEICE D-II, Vol. J84D-II, No.6 pp. 1073-1083, 2001

[6] Lee, C. M., and Kankanhalli, A., Automatic extraction of characters in complex scene images, International Journal of Pattern Recognition and Artificial Intelligence, Vol. 9, No. 1, pp. 67-82, 1995.

[7] Li, H., Doermann, D., and Omid Kia, Automatic Text Detection and Tracking in Digital Video, IEEE Trans. on IP, Vol. 9, No. 1, pp. 147-156, 2000

[8] Lienhart, R., Automatic Text Recognition for Video Indexing, Proc. of ACM Multimedia, pp. 11-20, 1996.

[9] Lim, Y., Choi, S., Lee, S., Text extraction in MPEG compressed video for content-based indexing, Proc. of ICPR, Vol. 4, pp. 409-412, 2000.

[10] Mullot, R., Olivier, C., Bourdon, J.L., Courtellemont, P., Labiche, J., and Lecourtier, Y., Automatic extraction methods of container identity number and registration plates of cars, Proc. of Int. Conf. on Industrial Electrlnics, Control and Instrumentation, pp. 1739-1744, 1991.

[11] Ohya, J., Shio, A., and Akamatsu, A., Recognition of characters in scene images, IEEE Trans. on PAMI, Vol. 16, No. 2, pp. 214-220, 1994.

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

[13] Wong, E. K. and Chen, M., A Robust Algorithm for Text Extraction in Color Video, Proc. of IEEE ICME, Vol. 2, pp. 797-800, 2000.

[14] Wu, V., Manmatha, R., and Riseman, E. M., TextFinder: An Automatic System to Detect, IEEE Trans. on PAMI, Vol. 21, No. 11, pp. 1224-1229, 1999.

[15] Yang, J., Chen, X., Zhang, J., Zhang, Y., Waibel, A., Automatic detection and translation of text from natural scenes, Proc. of ICASSP, Vol.2, pp. 2101-2104, 2002.

[16] Zhang, J., Chen, X., Yang, J., and Waibel, A., A PDA-based Sign Translator, Proc. of 4th Int. Conf. on Multimodal Interface, 2002.

[17] Zhong Y., Karu, K., and Jain, A.K., Locating Text in Complex Color Images, Pattern Recognition, Vol. 28, No. 10, pp. 1523-1536, 1995.