

RECOGNITION OF TEXT IN 3-D SCENES

Gregory K. Myers, Program Director
Robert C. Bolles, Program Director
Quang-Tuan Luong, Computer Scientist
James A. Herson, Senior Computer Scientist
SRI International
333 Ravenswood Avenue
Menlo Park, CA 94025
myers@erg.sri.com

ABSTRACT

Video is an increasingly important source of information to the intelligence analyst. Recognizing text that appears in real-world scenery is potentially useful for characterizing the contents of video imagery. Previous research in text recognition for both printed documents and other sources of imagery has generally assumed that the text lies in a plane that is oriented roughly perpendicular to the optical axis of the camera. However, text such as street signs, name plates, and billboards appearing in captured video imagery often lies in a plane that is oriented at an oblique angle. SRI International (SRI) is developing an approach that takes advantage of 3-D scene geometry to detect the orientation of the plane on which text is printed. The text recognition process will then be able to transform the video image of the text to a normalized coordinate system before performing OCR, yielding more robust recognition performance. Our approach applies full-perspective projections and image-to-image homographies that capture the appearance of a plane viewed through perspective optics. We describe our approach and present some preliminary results.

PROBLEM STATEMENT

Video is an increasingly important source of information to the intelligence analyst, and the volume of collected multimedia data is expanding at a tremendous rate. A capability to automatically identify the contents of video imagery would enable videos to be indexed in a convenient and meaningful way for later reference, and would enable actions (such as automatic notification and dissemination) to be triggered in real time by the contents of streaming video. Methods of realizing this capability that rely on the automated recognition of objects and scenes directly in the imagery have had limited success because (1) scenes may be arbitrarily complex and may contain almost anything, and (2) the appearance of individual objects may vary greatly with lighting, point of view, etc. The recognition of text is easier than the recognition of objects in an arbitrarily complex scene, because text was designed to be readable and has a regular form that humans can easily interpret.

Our effort is focused on scene text, such as street signs, name plates, and billboards, that is part of the video scene itself. Most previous text recognition efforts in video and still imagery [Jain and Bhattacharjee 1992; Ohya, Shio, and Akamatsu 1994; Zhong, Karu, and Jain 1995; Smith and Kanade 1995; Lienhart 1996; Yeo and Liu 1996; Wu, Manmatha, and Riseman 1997; Jain and Yu 1998; Sato et al. 1998; Li and Doermann 1998a; Li and Doermann 1998b] have assumed that the text lies in a plane that is oriented roughly perpendicular to the optical axis of the camera. Of course, this assumption

is valid for scanned document images and imagery containing overlaid text captions, but is not generally true for scene text. Figure 1 shows an image, captured from a video camera, of a café scene in which the name of the café is viewed from an oblique angle. Such a configuration is quite common when the main subject of the scene is not the text itself, but such incidental text could be quite important (for example, it may be the only clue to the location of the captured imagery).

To address the problem of recognizing text that lies on a planar surface in 3-D space, we note that the orientation angle of such text relative to the camera can be modeled in terms of three angles, as shown in Figure 2:

- q , the rotation in the plane perpendicular to the camera's optical axis
- j and g , the horizontal (azimuth) and vertical (elevation) components, respectively, of the angles formed by the normal to the text plane and the optical axis.



Figure 1. Café Scene

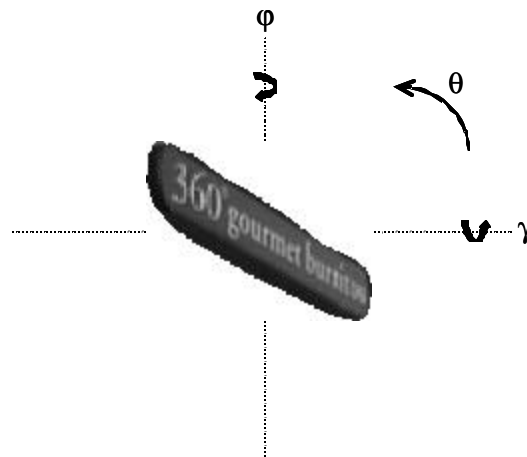


Figure 2. Orientation Angles of Text

The three angles represent the amount of rotation that the text plane must undergo relative to the camera in each of its three axes to yield a frontal, horizontal view of the plane in the camera's field of view. When q and g are zero and j is nonzero, the apparent width of the text is reduced, resulting in a change in aspect ratio and a loss of horizontal resolution. Similarly, when q and j are zero and g is nonzero, the text appears to be squashed vertically. The severity of perspective distortion is proportional to D/Z , where D is the extent of the text parallel to the optical axis (its "depth") and Z is the distance from the text to the camera. When the text is not centered at the optical axis or both j and g are nonzero, the text appears to be rotated in the image plane (see Figure 3). If the text were rotated to remove this apparent angle by a text recognition process that mistakenly assumed the text is fronto-parallel, the characters would become sheared (see Figure 4). When both j and g are nonzero and perspective distortion is significant, the shearing angle varies from left to right within the text region. OCR engines perform poorly if the shearing causes characters to touch or to be severely kerned (overlapped vertically).



Figure 3. Image Showing Apparent Rotation in the Image Plane



Figure 4. Image from Figure 3, After In-Plane Rotation

TECHNICAL APPROACH

In our approach we take advantage of 3-D scene geometry to detect the orientation of the plane on which text is printed. The text recognition process can then transform the video image of the text to a normalized coordinate system before performing OCR. There are two ways to estimate the parameters of a plane containing text. The first uses the shape and orientation of the text and the plane in a single image. The second examines the motion of the text and plane through a sequence of video image frames. We plan to develop techniques of both types and combine them to form a robust estimation procedure that takes into account the full perspective projection involved in the imaging process. In this paper we report some preliminary results from single-image analysis.

When the plane that contains the text is at an angle relative to the image plane, several types of distortions can be introduced that make it difficult to read the text. In the most general case, the distortion is described as a projective transformation (or homography) between the plane containing the text and the image plane. We can correct this distortion by applying the appropriate “corrective” projective transformation to the image. That is, we can rotate and stretch the original image to create a synthetic image, which we call a “rectified image,” in which the projective distortion has been removed.

In general, a two-dimensional projective transformation has eight degrees of freedom. Four correspond to a Euclidean 2-D transformation (translations along two axes, a rotation, and an isotropic scale factor); two correspond to an affine transformation (a shear and a nonisotropic scaling of one axis relative to the other); and the remaining two degrees of freedom represent a perspective foreshortening along the two axes.

From an OCR point of view, some of the eight parameters produce changes that are harder to handle than others. In particular, the two translations are not a problem, because they simply produce an image shift that is naturally handled by OCR systems. Similarly, the two scale factors are not a problem, because the OCR systems typically include mechanisms to work at multiple scales. The Euclidean rotation is important, but is easily computed from a line of text. Therefore, three critical parameters produce distortions that are difficult for OCR systems to handle: the two perspective foreshortening parameters and the shearing.

In our single-image analysis approach, estimates of the plane parameters are computed from the orientations of the lines of text in the image and the borders of planar patch, if they are visible. To remove a projective distortion, we need to compute the three critical degrees of freedom associated with the plane on which the text is written. In general, we can do this by identifying three geometric constraints associated with the plane. For example, we can compute necessary parameters, given two orthogonal pairs of parallel lines, such as the borders of a rectangular sign or two parallel lines of text and a set of vertical strokes within the text. The three constraints derivable from these sets of lines are two vanishing points (one from each set of parallel lines) and an orthogonality constraint between the sets of lines.



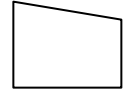

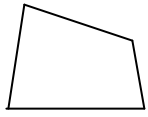
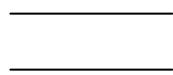
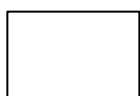
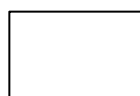






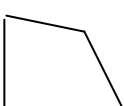
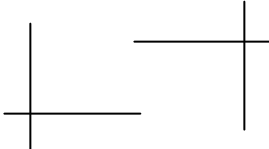




Sometimes, however, such linear properties are difficult to detect. In such cases, we can estimate the parameters by making assumptions about the camera-to-plane imaging geometry that are often true. For example, people normally take pictures so that the horizon is horizontal in the image. In other words, they seldom rotate the camera about its principal axis. In addition, they often keep the axis of the camera relatively horizontal. That is, they do not tilt the camera up or down very much. When these two

assumptions apply and the text lies on a vertical plane, such as a wall of a building or a billboard, the projective distortion is only along the X axis of the image. The perspective foreshortening in that direction can be computed from one constraint, such as a pair of horizontal parallel lines.

Another assumption that often holds is that the perspective effects are significantly smaller than the effects caused by the out-of-plane rotations. This is the case if the depth variation in the text is small compared with the distance from the camera to the plane. In this case, the perspective distortion is reduced to an affine shear and the projection is described as a weak perspective projection.

Table 1 summarizes the degrees of freedom that remain uncorrected after different sets of linear features are found and different assumptions are made about the plane-to-camera geometry.

Table 1. Degrees of Rectification

Geometric Relations Identified	Vertical Alignment		General Position	
	Weak Perspective	Full Perspective	Weak Perspective	Full Perspective
 Horizontal Line	 Fully Rectified	 Foreshortening in X	 Shear	 Foreshortening in X Foreshortening in Y Shear
 Parallel Horizontal Lines	 Fully Rectified	 Fully Rectified	 Shear	 Foreshortening in Y Shear
 Horizontal Line Vertical Line	 Fully Rectified	 Fully Rectified	 Fully Rectified	 Foreshortening in X Foreshortening in Y
 Parallel Horizontal Lines Parallel Vertical Lines	 Fully Rectified	 Fully Rectified	 Fully Rectified	 Fully Rectified

Given these relationships, our general strategy is to identify as many properties of a region of text as possible, and then compute a corrective transformation, using as few assumptions as possible. Initially, we use information derived independently from each individual line of text. Next, we combine information from multiple text lines after partitioning them into sets of lines that lie within a common

plane. We then further augment the process by detecting auxiliary lines that can provide horizontal and vertical cues. These can include lines in the same plane as the text (such as sign borders), and extraneous lines (e.g., building edges). Finally, depending upon our success in finding these features, we can either make assumptions to substitute for missing constraints (and then compute a transformation that corrects for a full perspective projection) or compute a transformation that does not completely remove all degrees of freedom. This approach is more general than the method described by Clark and Mirmehdi [2000], which requires the text to lie within a quadrilateral whose edges must be found; this quadrilateral is then transformed to a rectangle under a weak perspective assumption.

PRELIMINARY EXPERIMENTS

Thus far we have implemented only the first part of our general strategy—rectifying each text line in a single image independently. After possible lines of text are detected, various features of each text line are then estimated. These include the top and base lines, and the dominant vertical direction of the character strokes. The rectification parameters for each text line are computed from these characteristics. Each text line is then rectified independently and sent to an OCR engine.

The text detection and location process, somewhat similar to those described by Smith and Kanade [1995] and by Wu, Manmatha, and Riseman [1997], detects vertically oriented edge transitions in the gray-scale image, and links those that are compatible in size and relative position to form lines of text. A rectangle is then fitted to each line of detected text. Figure 5 shows a test image of a poster containing text that was captured at an azimuth angle of 70 degrees; the rectangles that have been fitted to each detected text line are shown in overlay. (Some of the rectangles do not look to the eye like true rectangles because of the perspective view of the image contents). Computing the best-fitting rectangle for each text line is an expedient way to approximate the location and extent of the text, but the top and bottom of the text are not accurately computed when significant perspective distortion is present.

A top line and base line for each line of text are estimated by rotating the text line at various angles and then computing a series of horizontal projections over the vertical edge transitions. (When the text consists of predominantly lower-case characters, the “top” line actually corresponds to the “midline” of the text that touches the tops of lower-case characters, excluding their ascenders.) The best estimate of the top line should correspond to the rotation angle that yields the steepest slope on the top side of the horizontal projection; the best estimate of the base line is similarly computed. Figure 6 shows an example of this procedure.

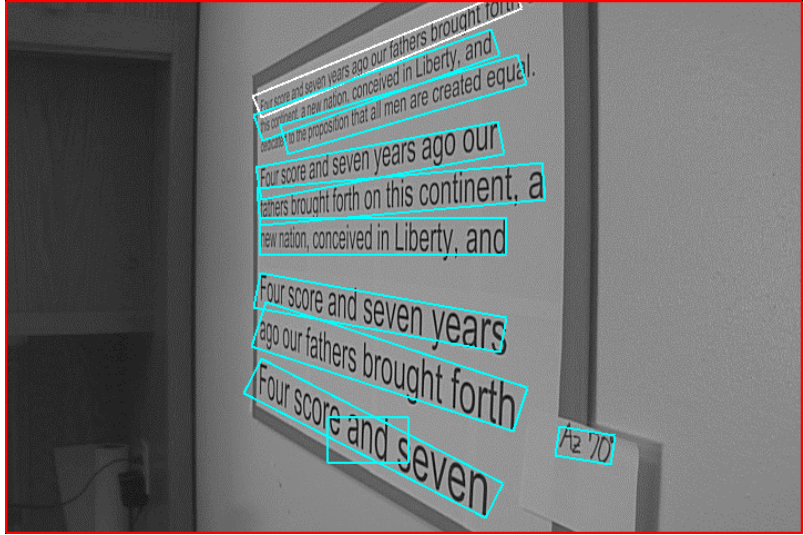


Figure 5. Rectangular Bounding Boxes Overlaid on a Test Image (Azimuth 70 Degrees)

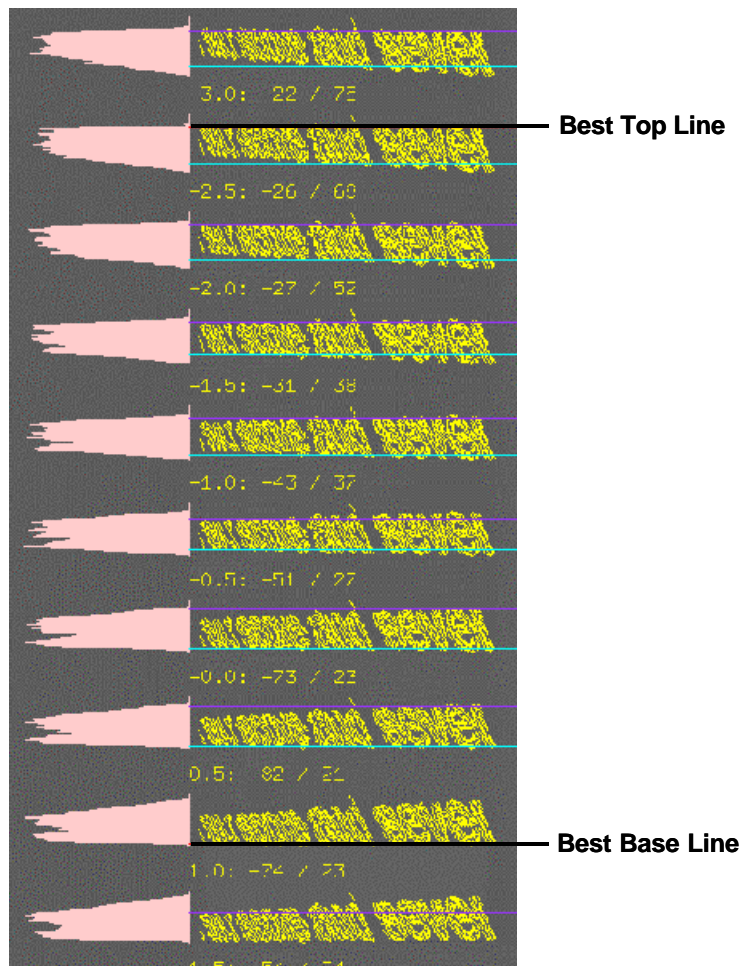


Figure 6. Estimation of Top and Base Lines

In addition to computing two horizontally oriented lines, we would like to find and measure the angles of two vertically oriented lines to use in the computation of the rectification parameters. Unfortunately, an individual line of text does not have much vertical extent, and it is difficult to determine which parts of the text could be used as vertical cues. However, the height of the text is not usually a significant fraction of the depth of the text in 3-D space, so that the perspective foreshortening in the Y dimension should be relatively small. Therefore, in the absence of any other reliable vertical cues, we compute the dominant vertical direction (shear) of the text by computing a series of vertical projections over the vertical edge transitions after rotating the text line in 2-degree increments. The best estimate of the dominant vertical direction should correspond to the angle at which the sum of squares of the vertical projection is a maximum (on the assumption that the projection of true vertical strokes is greatest when they are rotated to a vertical position). Figure 7 shows an example of shear computation. The deshearing process can be somewhat unreliable, because it assumes that a significant fraction of the characters contain vertical strokes. Figure 8 shows the refined bounding boxes based on the top and base lines and on the dominant vertical direction. Figure 9 shows the warped text lines (a) after the initial rectangular bounding box is deskewed; (b) after the baseline is refined (without including the top line in the dewarping computation) and then deskewed; and (c) after the lines are desheared.

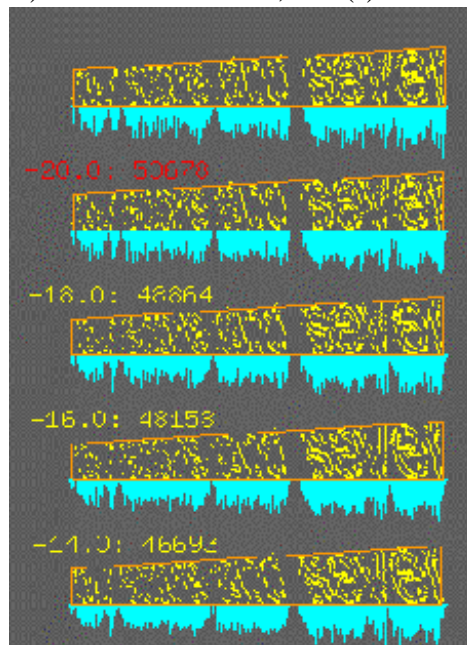


Figure 7. Estimation of Shear

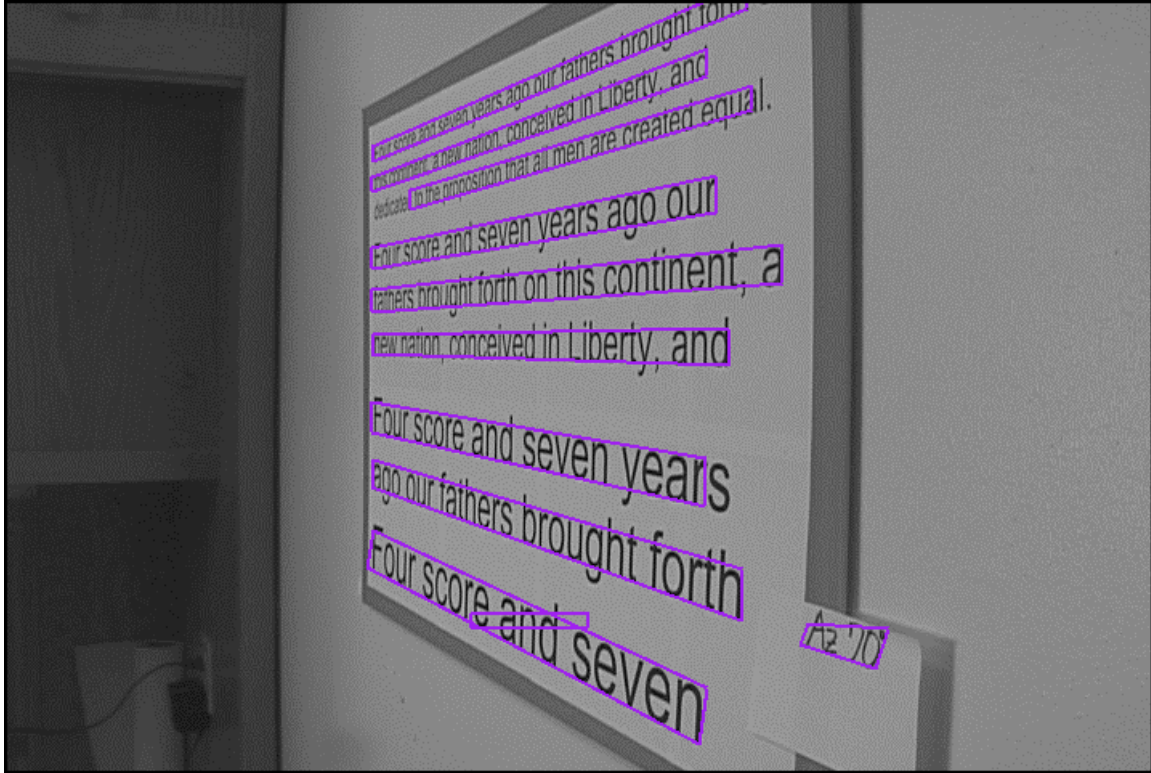


Figure 8. New Boxes from Top Lines, Base Lines, and Shear

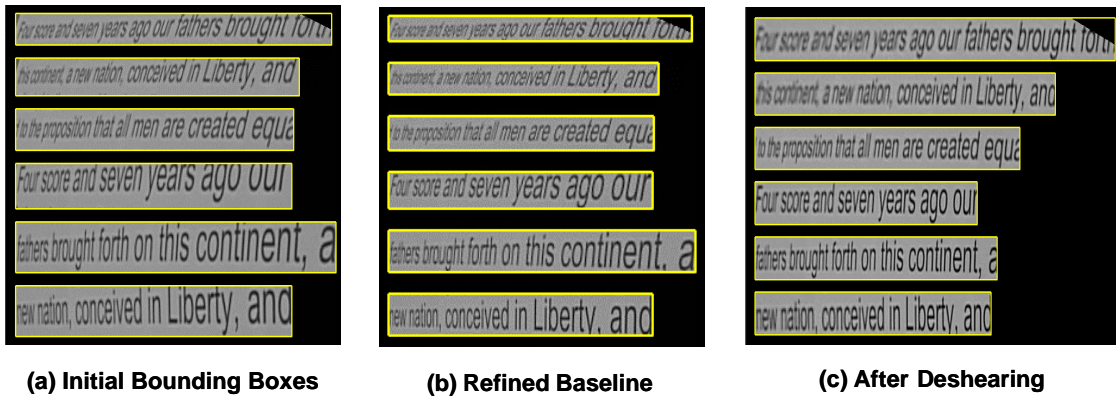


Figure 9. Warped Text Lines

The transformation used to rectify the image of each text line, L_j , occurring in an obliquely viewed image, O_i , is a projective transformation, T_{ij} , of the text plane. This transformation is described by

$$m' = Hm \quad ,$$

where H is a 3×3 matrix that maps the homogeneous coordinates $m = \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$ in O_i to the homogeneous

rectified coordinates $m' = \begin{bmatrix} sx' \\ sy' \\ s \end{bmatrix}$ in a normalized image N_i . The horizontal and vertical vanishing points

are mapped to the points at infinity in the horizontal $\begin{pmatrix} [1] \\ [0] \\ [0] \end{pmatrix}$ and vertical $\begin{pmatrix} [0] \\ [0] \\ [1] \end{pmatrix}$ directions. This process

takes care of the perspective foreshortening in both directions, as well as the skew and rotation. The remaining four degrees of freedom correspond to the origin and scale factors that place the line in the normalized image N_i . The image N_i , which contains all of the rectified lines from image O_i , is then sent through the OCR process. (We are currently using the Scansoft, Inc. DevKit2000 OCR package.) Figure 10 shows, for the 70 degree azimuth test image, the recognition results overlaid on the normalized image.

To measure the improvement in recognition performance due to the rectification process, we ran our process on a set of test images of a poster containing text viewed at various angles. The evaluation was performed semiautomatically by the process shown in Figure 11. We generated a ground truth data set (including the bounding boxes as well as the identities of the characters) by running the text detection and OCR process on a reference image R , and manually correcting any recognition errors. For our reference image we used a fronto-parallel view of the poster. In each of the test images O_i , the positions of the four corners of the poster were automatically detected and used to compute a transformation E_i that maps a pixel position in O_i into a corresponding position in R . By applying T^{-1}_{ij} and then E_i , we mapped the OCR results for line L_j from normalized coordinate space into the coordinate space of R . We expect that the lines and characters of text in correctly rectified images will coincide with those in a true fronto-parallel image. An automated process compares the recognized results to truth data on a line-by-line and character-by-character basis. Figure 12 shows the reference image overlaid with the ground truth data. Figure 13 shows the OCR results of Figure 10 after the inverse mapping back into the coordinate system of reference image R .

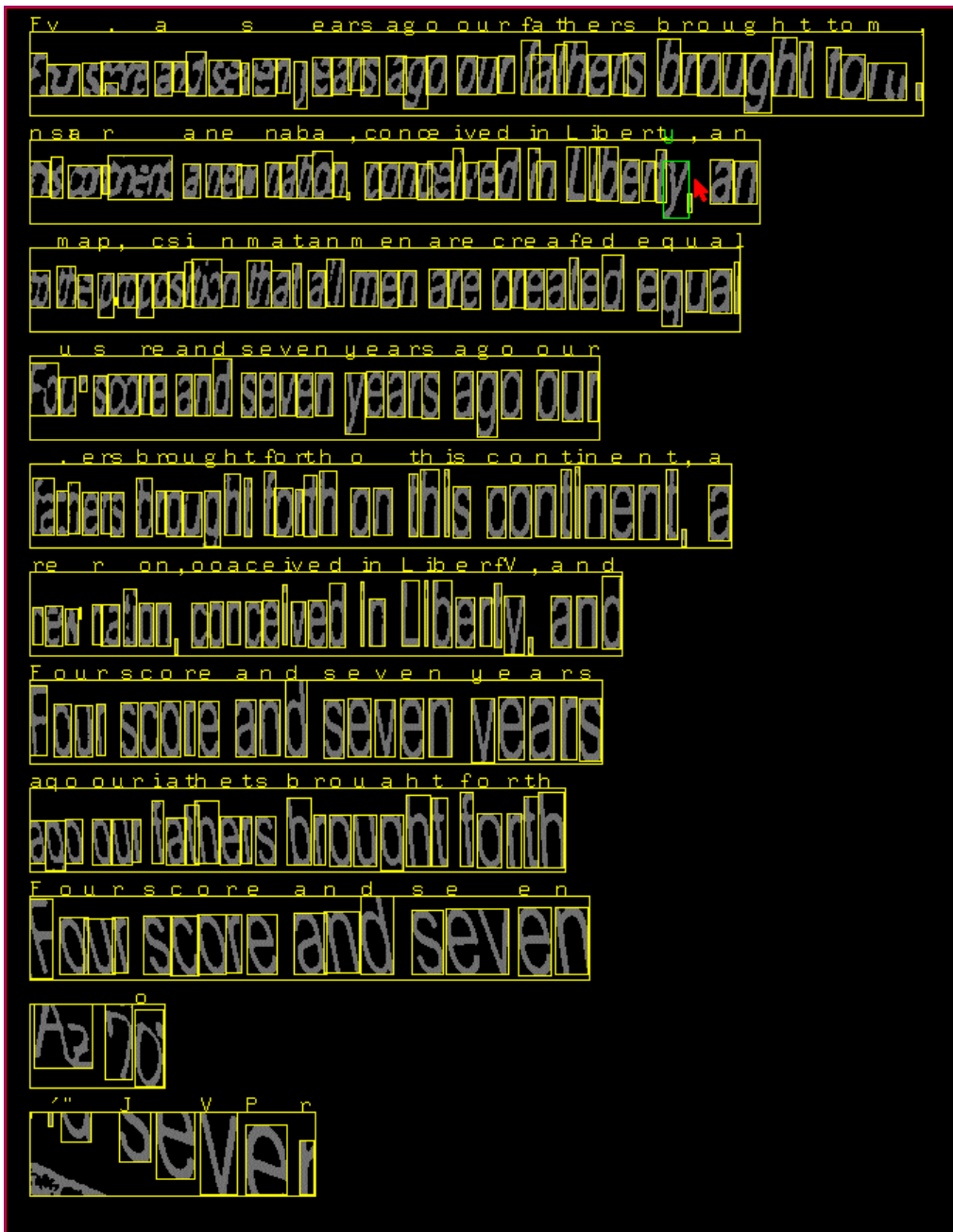


Figure 10. OCR Results Overlaid on the Normalized Image

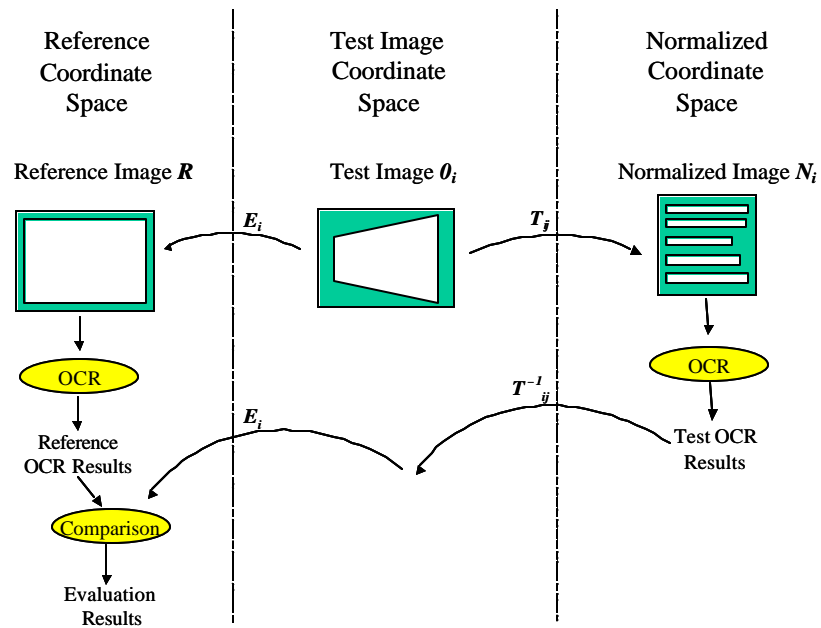


Figure 11. Evaluation Procedure

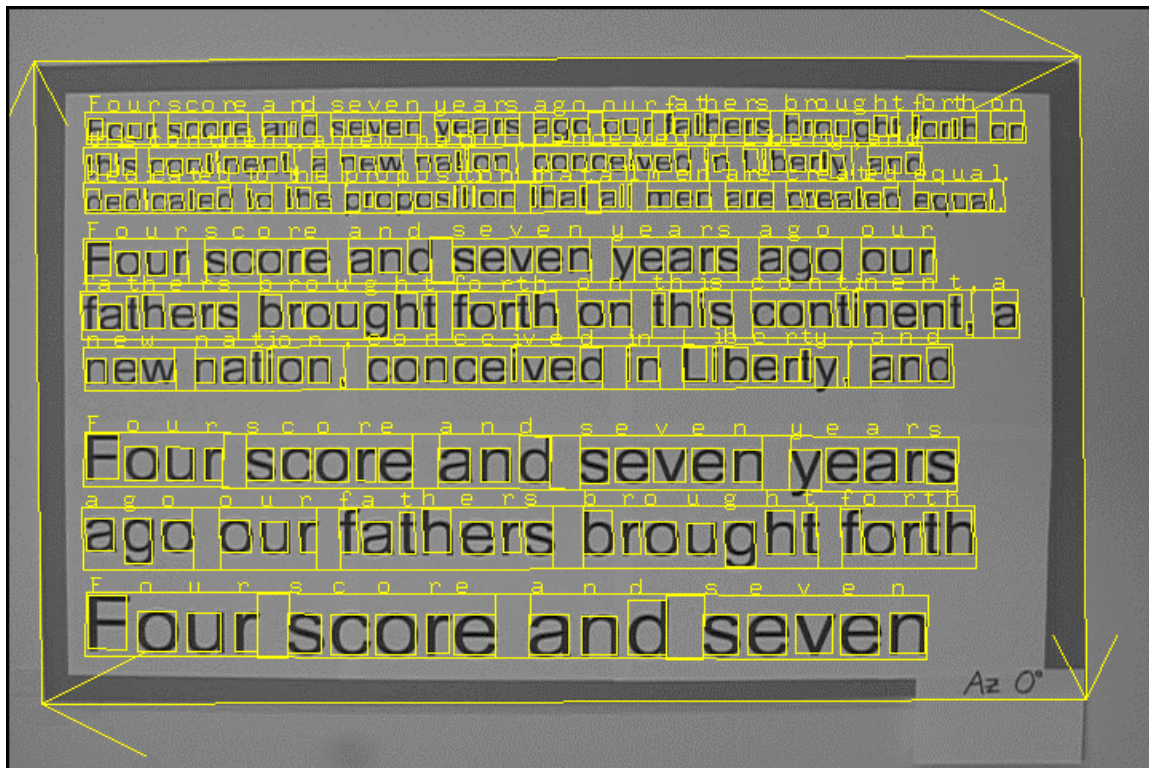


Figure 12. Reference Image and Ground Truth Data

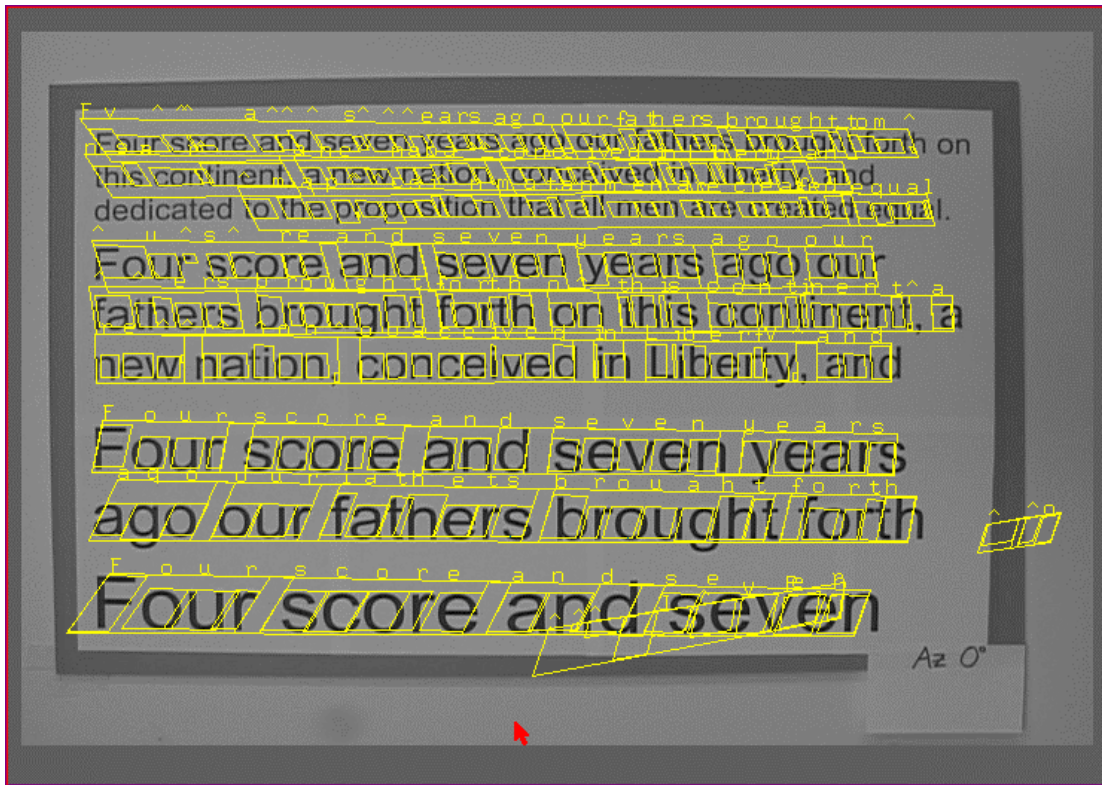


Figure 13. Test Results Overlaid on the Reference Image

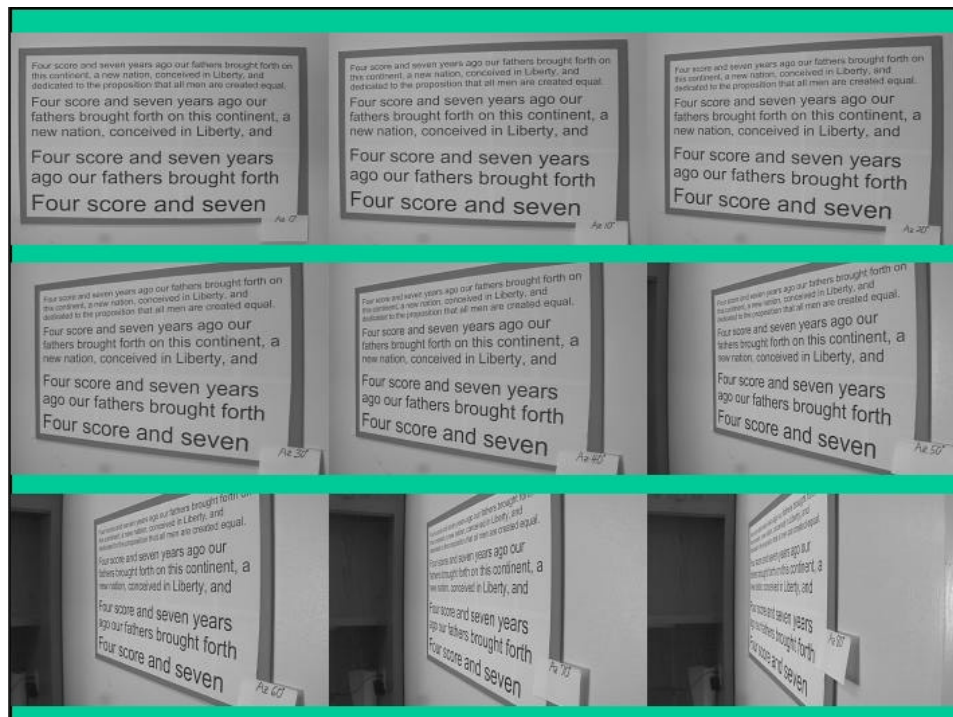


Figure 14. Azimuth Test Image Set

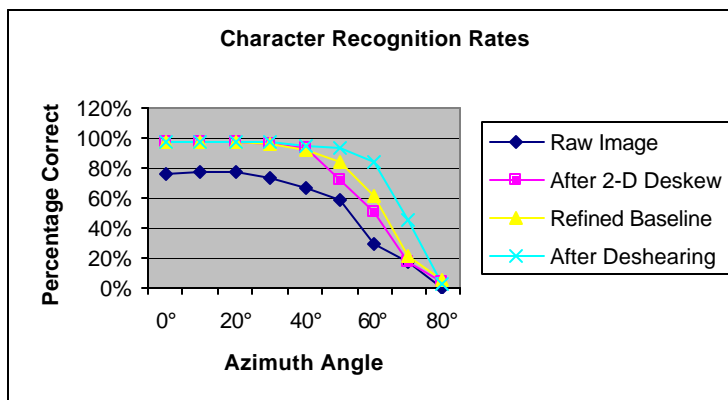


Figure 15. Test Results

Figure 14 shows one series of test images where the azimuth angle varies in increments of 10 degrees. Figure 15 shows the character recognition results as a function of azimuth angle for the various version of rectification. “Percentage Correct” means the number of characters recognized correctly, divided by the number of characters in the reference ground truth data set. As expected, the performance drops as the azimuth angle increases. At the most oblique angles, when the character stroke width and/or spacing between characters becomes one pixel or less in the original image, the resolution available for the interpolation process during rectification is not sufficient to adequately preserve the character features. The graph shows that each of the three processing steps contributes to the increase in performance. These improvements are greatest at the more oblique angles.

CURRENT AND FUTURE WORK

We are currently developing methods that will automatically compute the rectification parameters for all text lines in a single image simultaneously. This process is expected to yield more reliable estimates of the rectification parameters, especially for short lines of text. Part of this process will automatically determine which sets of lines (the top and base lines from lines of text, and other significant lines in the image) have a common vanishing point and are therefore truly parallel and lie within a single plane.

Because lens distortion can affect the accuracy of our estimation of straight lines, we are implementing a preprocessing step that characterizes and corrects for lens distortion in the imagery. We estimate the parameters of our lens distortion model by providing the sequence of pixels along four or more straight lines detected in the set of image frames in a video sequence captured with the same lens setting.

In the future we plan to use the information from multiple images in the video sequence to produce a more robust estimate of the rectification parameters. Methods we will consider include exploiting the consistency across frames of the rectification parameters themselves, tracking features of the text regions such as baselines and sign borders, and deducing the orientation of text planes by tracking the displacement of individual points through the image sequence.

ACKNOWLEDGEMENT

This work was supported in part by the Advanced Research and Development Agency (ARDA).

REFERENCES

- Clark, P. and M. Mirmehdi. 2000. "Location and Recovery of Text on Oriented Surfaces," *SPIE Conf. on Document Recognition and Retrieval VII*, pp. 267–277 (January).
- Jain, A., and S. Bhattacharjee. 1992. "Text Segmentation Using Gabor Filters for Automatic Document Processing," *Machine Vision and Applications*, Vol. 5, pp. 169–184.
- Jain, A., and B. Yu. 1998. "Automatic Text Location in Images and Video Frames," *Proc. ICPR*, pp. 1497–1599.
- Li, H., and D. Doermann. 1998a. "Automatic Identification of Text in Digital Video Key Frames," *Proc. Intl. Conf. on Pattern Recognition*, pp. 129–132.
- Li, H., and D. Doermann. 1998b. "Automatic Text Tracking In Digital Videos," *Proc. IEEE 1998 Workshop on Multimedia Signal Processing*, pp. 21–26.
- Li, H., D. Doermann, and O. Kia. 1998. "Text Extraction and Recognition in Digital Video," *Proc. Third IAPR Workshop on Document Analysis Systems*, pp. 119–128.
- Li, H., D. Doermann, and O. Kia. 1999. "Automatic Text Detection and Tracking in Digital Video," *IEEE Trans. Image Processing—Special Issue on Image and Video Processing for Digital Libraries*, pp. 147–155.
- Lienhart, R. 1996. "Indexing and Retrieval of Digital Video Sequences based on Automatic Text Recognition," in *4th ACM International Multimedia Conference*, Boston (November).
- Ohya, J., A. Shio, and S. Akamatsu. 1994. "Recognizing Characters in Scene Images," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Vol. 16, No. 2, pp. 214–220.
- Sato, T., K. Takeo, E. Hughes, and M. Smith. 1998. "Video OCR for Digital News Archive," *Proc. 1998 Intl. Workshop on Content-Based Access of Image and Video Databases (CAIVD '98)*, Bombay, India, IEEE Computer Society, ISBN 0-8186-8329-5 (3 January).
- Smith, M.A., and T. Kanade. 1995. *Video Skimming for Quick Browsing Based on Audio and Image Characterization*, Technical Report CMU-CS-95-186, Carnegie Mellon University (July).
- Wu, V., R. Manmatha, and E. Riseman. 1997. "Automatic Text Detection and Recognition," in *Proc. Image Understanding Workshop*, pp. 707–712.
- Yeo, B.-L., and B. Liu. 1996. "Visual Content Highlighting via Automatic Extraction of Embedded Captions on MPEG Compressed Video in Digital Video Compression: Algorithms and Technologies," in *Proc. SPIE 2668-07*.
- Zhong, Y., K. Karu, and A. Jain. 1995. "Locating Text in Complex Color Images," in *Proc. Third Intl. Conf. on Document Analysis and Recognition*, Montreal, Canada (14–16 August).