A Robust Method for Tracking Scene Text in Video Imagery

Gregory K. Myers and Brian Burns
SRI International
Menlo Park, CA 94025 USA

Abstract

Text on planar surfaces in 3-D scenes in video imagery can undergo complex apparent motion and distortion as the surfaces move relative to the camera. Tracking such text and its motion through a contiguous sequence of video frames in which it is visible is desirable primarily for two reasons. First, reliable tracking of text enables the images of text persisting across multiple frames to be grouped, processed, and understood as a single unit. Second, text tracking aids the mapping of corresponding text and background pixels across multiple frames to enhance image quality and resolution before character recognition. Existing text tracking approaches, however, are limited to approximate pixel-based correspondences of adjacent frames without any explicit, rigorous modeling of 3-D scene geometry. To this end, we describe an approach that tracks planar regions of scene text that can undergo arbitrary 3-D rigid motion and scale changes. Our approach computes homographies on blocks of contiguous frames simultaneously using a combination of factorization and robust statistical methods. In spite of low resolution and noisy imagery, this approach produces a more accurate and stable motion estimate than existing methods using only two adjacent frames. In addition, our method is robust enough to tolerate imperfections in the spatial localization of text. Our results demonstrate that the mean offset pixel error of our tracker is as small as 1.1 pixels.

1. Introduction

Recognition of text that appears in real-world scenes, such as protest signs and name tags, is of utility for automated characterization and annotation of video imagery because of its valuable contribution to the video content. Such a capability enables information retrieval systems to index videos in a convenient and meaningful way for later reference. Text in video can take the form of artificially generated text that is overlaid on the imagery (such as superimposed captions in broadcast news programs and other commercially produced videos), or text that is part of the video scene itself (such as a sign outside a place of business or placards in front of conference participants). In this work we focus on scene text.

The recognition of scene text in video imagery involves several major processing steps, including text detection, text tracking, and OCR. This paper focuses on tracking of scene text. Since the same text can be visible on multiple consecutive frames in video, tracking of text is desirable so that all the images of text in the multiple frames can be grouped, processed, and understood as a single unit. In real-time applications with live video, such as portable road sign translators for tourists and soldiers, recognized text can be treated as an event that immediately triggers additional automated processes, such as machine translation and speech synthesis. An automated surveillance system may trigger database lookup of a recognized vehicle license plate number. Therefore, reliable means of text tracking is important to ensure a single response for each distinct text event.

There are two aspects to text tracking: (1) frame-to-frame association of text regions, and (2) frame-to-frame motion estimation of each region. The former involves determining the temporal continuity of regions and assigning an ID to each tracked text region. The latter involves computing a pixel-to-pixel mapping to establish localized frame-to-frame geometrical correspondence. Frame-to-frame association enables a single OCR result to be produced and reported for multiple contiguous appearances of the same text object. Furthermore, frame-to-frame geometrical corresponddence is required for the video OCR process to take advantage of temporal redundancy of text that appears in multiple frames to create an enhanced image before subsequent OCR processing. Such multiframe integration includes multiple image averaging [1,2] and superresolution [3,4].

Scene text has a number of characteristics that make it difficult to detect and track. Scene text exists in 3-D space and can be slanted, tilted, or modulated by the surface of the object on which the text is printed. Although scene text often lies in a plane, several types of distortion can be introduced when the plane containing the text is at an angle relative to the image plane. In the most general case, the distortion is described as a projective transformation between the plane containing the text and the image plane [5]. The tracking of scene text is complicated by the fact that its appearance can change drastically during its presence in the video. In addition to camera pan, tilt, rotation, and zoom affecting the text, the size and viewing angle of text on moving objects can vary significantly.

Figure 1. Subarea in consecutive frames of video shot from a moving vehicle.

In addition, video sequences that are generated with a nonstationary camera (e.g., a handheld camcorder or vehicle-mounted video camera) may contain a significant amount of random, jerky camera motion, which can blur the image of the text, cause interlace shearing, and/or make the text hard to track from frame to frame. This is especially true of imagery collected at high-magnification lens settings. For example, Figure 1 illustrates some of these artifacts on a subarea extracted from seven consecutive frames of a video captured from a moving vehicle. Furthermore, a camera's automatic focus mechanism may take time to adjust while the camera zooms and pans or the contents of the scene change or move, resulting in some intermittently out-of-focus frames. Scene text may be partially obscured temporarily by objects moving in the foreground (e.g., a person walking in front of a sign).

2. Prior work

Due to the difficulties in tracking scene text that were outlined in Section 1, many previous text tracking approaches were designed for overlay text [6,2,7], assumed the text was horizontally oriented, and used a translational motion model. Crandall et al. [8] used motion vectors in MPEG compressed video and a least-square-error search of a small neighborhood for tracking of text regions; for text captions that rotated or changed scale, features extracted from the connected components within the text region were matched in consecutive frames. Li and Doermann [1] and Li et al. [9] aligned blocks of scene text in adjacent frames using a translational motion model and correlation within a search window, and when the detected motion did not fit the translational model, the contour of the text region was determined by tracking a blob created by horizontal smearing of edges found in the text region. Tracking failures were detected [1] by using criteria such as straightness of the motion trail of the text region center.

In all the previous work cited above, motion was computed from adjacent frames only; therefore, these tracking methods may fail if the video is intermittently degraded, or if portions of the text regions are temporarily occluded. In addition, previous methods that relied on trajectory-based motion prediction would have difficulty tracking text in video with random, jerky camera motion. Furthermore, previous methods assumed that the text region boundaries are accurately defined. However, text detectors may not always find text region boundaries reliably. For example, non-text image patterns that have characteristics similar to text and are adjacent to text in the scene might incorrectly be included as part of the text region in some frames. Finally, previous methods estimated motion only in the 2-D image plane and did not attempt to explicitly model the 3-D motion of text in the scene.

In addition to the above text-specific tracking approaches, previous work on tracking regions in general include systems based on the brightness constraint equation [10–12], and motion estimation from two-frame point correspondences [13]. In the former method, the pixel intensity difference between two frames is expressed as a function of the motion between the frames and is minimized using nonlinear optimization techniques. This method, for the case of planar motion, has been extended to multiple frames in [14] using factorization. The expression assumes that all changes in intensity at each pixel are due to motion, instead of brightness change, occlusion, and other effects. This makes the method problematic in situations where there is clutter and occluding surfaces in the region that are not moving the same way. In addition, for the motion to be solvable by successive linearization, it has to either be small, or the region has to be large enough relative to the motion to be suitable for multiscale methods. This is not always the case for text regions of short height and viewed through a handheld camera. The other approach [13], tracking individual points followed by robust estimation of the whole region motion, is more suitable when there is clutter, occlusion, and large motion. In the method of Hartley and Zisserman [13], not all of the points have to be successfully tracked or consistent with the motion, as long as there is a large enough inlier set that is. However, the method proposed by Hartley and Zisserman [13] estimates tracking descriptors only on two frames at a time. Since text is often visible on 30 or more video frames (at 30 fps), estimation based on just two frames is limiting and is clearly not optimal.

3. Proposed approach

One of the primary contributions of the current work is its generalization of the work described in Hartley and Zisserman [13] to simultaneous multiple frame analysis. In our work we track points of interest across all the frames being considered, and then, within each detected text region, estimate the planar transformation simultaneously and robustly over blocks of multiple frames. We assume that the region to be tracked is planar in the scene and that there are a sufficient

number of points in the region with enough texture to be tracked over the frames that are compared. For six parameter affine motion, we require a minimum of three points tracked over all frames in a block. These assumptions are typically valid for text regions, since text regions are typically highly textured and bimodal in intensity. Using the process described below, the region of interest is tracked a block of images at a time. The motion in the previous block is used to locate the region in the first frame of the next block, and so on, until the end of the video or a point is reached where the minimum set of inliers goes below a threshold (four tracked points), indicating that the region cannot be tracked further. Ideally, the block size should be selected automatically to maintain a sufficient point correspondence count. In the experiments discussed here, the block sizes were manually fixed to five or ten frames, depending on the magnitude of image change and the size of the region.

3.1. Point location and tracking

Tracking a point has three steps: (1) initially selecting the point in the image, (2) localizing it in subsequent frames, and (3) determining when it is no longer trackable (termination). In our system, different points are selected and terminated in different frames, so that any given pair of relatively close frames will have points in common. Points are selected using two criteria: *texture* and *coverage*. The presence of *texture* aids reliable localization of the point across frames, which is true if the local intensity variation in the image in different directions is high. The peaks of the Laplacian-of-Gaussian image and/or the peaks of the Shi and Tomasi texture operator [11] can be used as indicators of high texture regions. We have found that these two methods have similar and reasonable results. When there are points detected and tracked over all parts of the region, the region is said to have high *coverage*. To improve coverage, we iteratively select detected points, greatest texture first, until the maximum distance from each pixel to the nearest detected point is below a preselected minimum threshold. Selected points are tracked and localized in subsequent frames using normalized correlation of a small image patch centered at the current position of the point. The new position of the point is located to sub-pixel precision by quadratic interpolation of the correlation surface. The tracking of a point is terminated when the correlation drops below a threshold, which is typically 0.65 out of a range of [−1, 1], where 1 is perfect. The image patch size is a function of the magnitude of the image transformation, the size of the scene surfaces, the magnitude of visual texture, and the video quality. Small patches tend to be tolerant of large image transformations and complex scene geometry, but less tolerant of poor image texture and poor video quality. Image patches of 15×15 pixels (relatively small) were used for the experiments discussed here since text tends to support good texture and the video quality was minimally adequate for this size.

3.2. Point selection and motion estimation

A large number of the points in the automatically extracted region of interest may be unusable for the estimation of the region motion for various reasons: They may be on a different surface, their trajectories may be misestimated due to noise and low resolution, or the points may be optical artifacts such as specularities and moving shadows. To make text tracking reliable, the unusable points must be detected as outliers. This outlier detection is challenged by the fact that there are many degrees of freedom (six or eight) in the projected motion of the scene surface.

Once points are tracked, we estimate the transformation of the whole region in blocks of multiple frames. Our method combines two approaches: (1) robust parameter estimation, such as RANSAC [15], which simultaneously estimates the parameters and determines the inlier data set; and (2) simultaneous, multiframe reconstruction of the projecting points and the projecting transformations for all the frames, which further reduces the transformation error and reduces outliers.

3.2.1. Application of RANSAC. Applying RANSAC, our algorithm randomly selects minimal subsets (of 3 points each) of point tracks in the frame block, estimates the projective transformation for each frame given the set (using the factorization approach discussed below), and then counts the number of point tracks that are consistent with the transformations (inliers). The largest inlier set is then used for the multiframe reconstruction, again using the factorization approach. Point track consistency is measured as the root-mean-squared projection error across all the frames after reconstructing the scene plane position of this point and reprojecting it in all frames. A projection error of two pixels was used as a consistency cutoff.

3.2.2. Multiframe reconstruction and motion estimation. The multiframe reconstruction of the scene plane and the projective transformations is done using a 2-D version of the 3-D factorization technique developed in [16]. Since the scene structure we are recovering is planar, we can force the factorized matrices to be of rank 2, which further constrains the reconstruction (beyond the 3-D case) and leads to a more accurate solution for our purposes. As in the original 3-D version, we construct a $2m \times n$ data matrix W, where each column is a point track $[x_1, y_1, x_2, y_2, ..., x_m, y_m]^T$, (x_i, y_i) is the point position in the ith frame minus the point centroid of that frame, and there are m frames and n points. Assuming an affine camera model,

$W = M \times S$, where M is a $2m \times 2$ motion matrix, with each pair of rows i representing the nontranslational components of the affine projection for frame i, and where S is a $2 \times n$ matrix with each column j representing point j's two component position on the scene plane. Using SVD, W can be factorized into $U \times D \times V^T$, where $U \times \sqrt{D}$ and $\sqrt{D} \times V^T$ are M and S up to a 2-D affine transformation. Since we are only interested in the 2-D motion between frames, this ambiguity can be ignored. The motion between the first frame 0 in the block and any other frame t is $H(0,t) = Q_t \times Q_0^{-1}$, where Q_t and Q_0 are the affine transformations between the scene plane and frames t and 0 respectively. Q_t is constructed by using the point centroid of the frame as the translation component and rows $2t$ and $2t - 1$ of M for the other four parameters. Q_0 is constructed analogously.

4. Preliminary Results

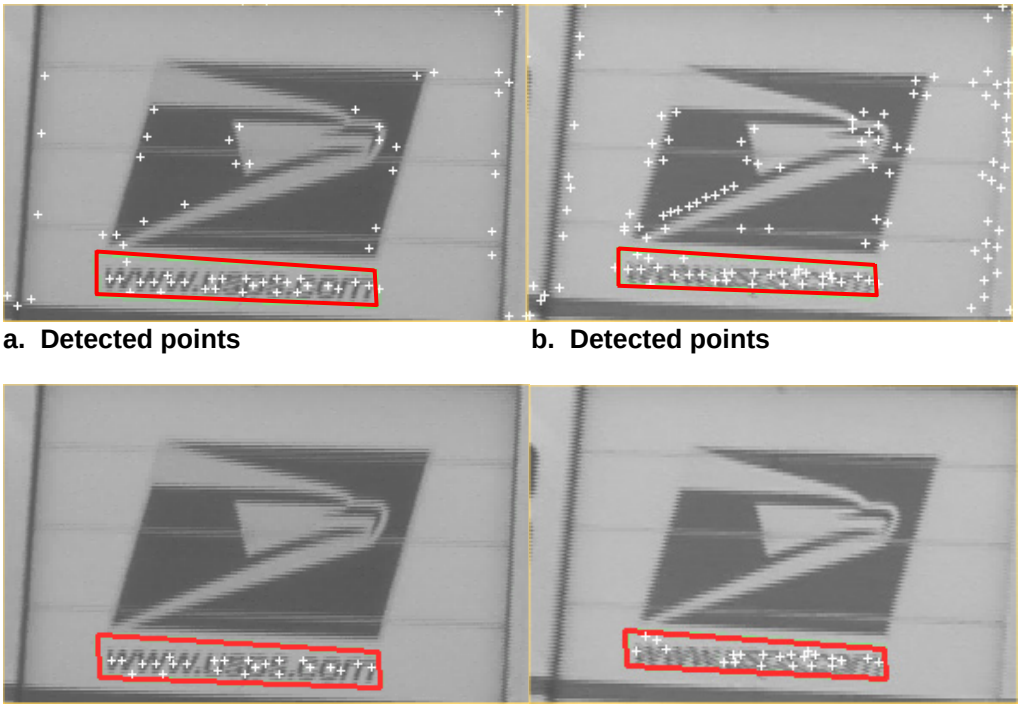
Figures 2 through 4 show some examples of the tracking performance. For each of these video sequences, the tracker was initialized by manually specifying a bounding box to simulate the results of a text detection process. Figure 2 shows the results of tracking the text region shown in Figure 1. Figures 3 and 4 show three frames extracted from two other sequences. The imagery in Figures 2 and 3 was taken with a handheld video camera from a moving vehicle. The middle frame in Figure 3 shows one of three consecutive frames in which the text region (the "Hallmark Cards" sign) was occluded by a pole in the foreground. The text region in Figure 4 was successfully tracked throughout the zoom out by a factor of 6, even to the point of the text being unreadable. To assess performance quantitatively, we calculated the mean offset pixel error by comparing the relative positions of 4 points in the first and last frames of 9 tracked text regions, each in a different video sequence. The mean offset pixel error of the tracker was 1.1 pixels.

Figure 5 shows some of the details of the point detection and selection. Figures 5a and 5b show all of the points that were detected on a portion of the back of a moving postal truck in video frames taken about 4 seconds apart; Figures 5c and 5d show the inlier points used for tracking the text region in frames 5a and 5b, respectively. Notice that only a subset of the detected points is used for tracking, and the set of points detected and tracked at the two video frame times is not identical.

Figure 2. Example with jerky camera motion.

Figure 3. Example with jerky camera motion and occlusion.

Figure 4. Example with large change of scale.



a. Detected points b. Detected points

c. Inlier tracked points d. Inlier tracked points

Figure 5. Details of point detection and selection on moving postal truck.

5. Discussion

Our method ascertains that, in a block of frames of nontrivial size, the trajectory of every point is consistent with the region motion over several frames. The likelihood of the trajectory of an outlier point being spuriously consistent over many frames is much lower than the chance of it being consistent over a single frame pair. Therefore, the multiframe nature of the proposed reconstruction approach assists the detection of outliers during the motion estimation process compared with the methods considering only two frames at a time [e.g., 13]. The multiframe reconstruction approach being used here, factorization in combination with robust statistical search methods, also reduces the effects of noise or low magnitude tracking errors in the inlier point set. Using factorization, a least squares reconstruction of the original scene points is generated from all the projections in the frame block, simultaneously with the projecting homographies from this point reconstruction to the individual frames. Thus, the estimated transformations are with respect to a reconstructed point set, not between two noisy point sets as in the two-frame approach.

As in 3-D factorization, our approach can be extended to handle the projective case using an iterative approximation [13]. However, the full eight parameter projective model should be used with caution on small text regions in noisy video, and it is typically unnecessary for tracking the text.

Unlike tracking methods such as Kalman filtering that rely on a particular motion model, this method does not require any knowledge about how the motion in different frames is related, and therefore can track text in video with random, jerky camera motion.

Future designs will include enhancements of both the point tracking and the geometric analysis. Currently, the point tracker can fail when there are temporary interruptions in the point visibility or quality of the video since the tracking stops once the correlation drops below a threshold. Instead, the tracker should continue to search for the point in subsequent frames and report only the parts of the track with high enough correlation. The geometric analysis can be improved by not forcing the factorization step to use exactly every frame in the block. The quality could improve by selecting only frames of high enough quality, essentially an outlier detection process for frames analoguous to the outlier detection already performed over the points.

6. Summary

Text on planar surfaces in 3-D scenes in video imagery can undergo complex apparent motion and distortion as the surfaces move relative to the camera. Tracking such text and its motion through a contiguous sequence of video frames in which it is visible is desirable primarily for two reasons. First, reliable tracking of text enables the images of text persisting across multiple frames to be grouped, processed, and understood as a single unit. Second, text tracking aids

the mapping of corresponding text and background pixels across multiple frames to enhance image quality and resolution before character recognition. Existing text tracking approaches, however, are limited to approximate pixel-based correspondences of adjacent frames without any explicit, rigorous modeling of 3-D scene geometry. To this end, we describe an approach that tracks planar regions of scene text that can undergo arbitrary 3-D rigid motion and scale changes. Our approach computes homographies on blocks of contiguous frames simultaneously using a combination of factorization and robust statistical methods. In spite of low resolution and noisy imagery, this approach produces a more accurate and stable motion estimate than existing methods using only two adjacent frames. In addition, our method is robust enough to tolerate imperfections in the spatial localization of text. Our results demonstrate that the mean offset pixel error of our tracker is as small as 1.1 pixels.

7. Acknowledgment

This material is based on work supported in whole by the U.S. Government. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the U.S. Government.

8. References

[1] H. Li and D.S. Doermann, "Text Enhancement in Digital Video Using Multiple Frame Integration", *Proc. ACM Multimedia '99*, Orlando, Florida, 1999, pp. 19–22.

[2] C. Wolf, J.-M. Jolion, and F. Chassaing, "Text Localization, an Enhancement, and Binarization in Multimedia Documents", *Proc. of International Conference on Pattern Recognition (ICPR),* Vol. 4, August 2002, pp. 1037–1040,.

[3] Huiping Li and David Doermann, "Superresolution-Based Enhancement of Text in Digital Video", International Conference on Pattern Recognition (ICPR'00), Barcelona, Spain, 2000.

[4] Katherine Donaldson and Gregory K. Myers, "Bayesian Super-Resolution of Text in Video with a Text-Specific Bimodal Prior", *International Journal on Document Analysis and Recognition*, Volume 7, Numbers 2-3, July 2005, pp. 159 – 167.

[5] G.K. Myers, R.C. Bolles, Q.-T. Luong, and J.A. Herson, "Recognition of 3-D Scene Text," Fourth Symposium on Document Image Understanding Technology (SDIUT01), Columbia, Maryland, April 2001, pp. 85–99 (http://www.esd.sri.com/projects/vace/docs/SDIUTMyers2.pdf).

[6] R. Lienhart, "Indexing and Retrieval of Digital Video Sequences based on Automatic Text Recognition", in *4th ACM International Multimedia Conference*, Boston November 1996.

[7] R. Lienhart and A. Wernicke, "Localizing and Segmenting Text in Images and Video", *IEEE Trans on Circuits and Systems for Video Technology*, Vol. 12, No. 4 April 2002.

[8] David Crandall, Sameer Antani, Rangachar Kasturi, "Extraction of Special Effects Caption Text Events from Digital Video", IJDAR 5(2-3): 138–157, 2003.

[9] H. Li, D. Doermann, and O. Kia, "Automatic Text Detection and Tracking in Digital Video", *IEEE Trans. Image Processing—Special Issue on Image and Video Processing for Digital Libraries*, Vol. 9, No. 1, 2000, pp. 147–55.

[10] B. Lucas and T. Kanade, "An Iterative Image Registration Technique with an Application to Stereo Vision", *IJCAI*, 1981.

[11] J. Shi and C. Tomasi, "Good Features to Track", *IEEE Conf on CVPR*, June 1994.

[12] M. Irani and P. Anandan, "About Direct Methods", in *Vision Algorithms: Theory and Practice*, B. Triggs, A. Zisserman, and R. Szeliski (eds), Springer, June 2000.

[13] Richard Hartley and Andrew Zisserman, *Multiple View Geometry in Computer Vision*, Cambridge University *Press*, 2000.

[14] L. Zelnik-Manor and M. Irani, "Multi-frame Estimation of Planar Motion", IEEE PAMI, 22(10):1–12, October 2000.

[15] M.A. Fischler and R.C. Bolles, "Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography", *Comm ACM*, 24 (6):381–395, 1981.

[16] C. Tomasi and T. Kanade, "Shape and Motion from Image Streams Under Orthography", *IJCV*, 9(2):137–154, November 1992.