Projective Transformation Estimation Using Straight Line Segments Detected and Estimated via Neighborhood Clustering [*]

Richard J. Radke Peter J. Ramadge Sanjeev R. Kulkarni
Department of Electrical Engineering
Princeton University
Princeton, NJ 08540
{rjradke,ramadge,kulkarni}@ee.princeton.edu

Tomio Echigo
IBM Tokyo Research Laboratory
IBM Japan, Ltd.
Kanagawa-ken 242-8502, Japan
echigo@trl.ibm.co.jp

Abstract

We describe a process for extracting straight line segments from a digital image with high spatial accuracy and confidence. The lines are then used to estimate projective transformations between natural, outdoor images with sparse features. The line estimates are more accurate than those obtained using a conventional scheme, and improve the estimation of projective transformations when used in conjunction with other features.

1 Introduction

We want to estimate the projective transformation relating two images taken by a rotating camera. This problem is straightforward in settings with good lighting and a feature-rich scene. However, the problem is difficult for scenes with moving objects and sparse features, e.g. sports video. In such scenes, correspondences between lines in the images can sometimes be used to aid in the estimation of the projective transformation. However, to be useful, the line parameters should be estimated with high spatial accuracy and confidence. Accurate line estimates may also be incorporated into problems of projective reconstruction [8], structure from motion [18, 26], and line matching [21].

Detecting the presence and estimating the parameters of straight line segments in digital images is a well-studied (e.g. [2, 4, 6, 10, 15, 16, 17, 24]) yet still challenging task. Many existing algorithms suffer from spurious, profuse, fragmented, or multiply detected lines.

This paper presents a method for projective transformation estimation based on line segment correspondences. This in turn necessitates the development of a new line segment detection scheme which ensures that the positions and orientations of detected lines are estimated with high accuracy. The proposed scheme has no prelimited spatial resolution, and the incidence of falsely or multiply detected line segments is much less frequent than the results associated with other line detection schemes.

The input to the line estimation procedure is a noisy binary image in which pixels marked as "1" nominally correspond to pixels on image lines. The extraction of this binary image from a real gray-scale image is a separate problem that is not the focus of this paper. However, two methods by which this map can be obtained are mentioned in section 4. Our line estimation scheme is robust to the noise that will inevitably be present in the binary image.

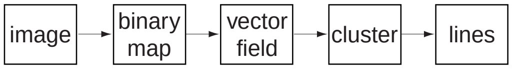


Figure 1: Steps of line estimation algorithm.

The estimation of lines from the binary image proceeds in two steps. First, a vector is associated to each marked pixel. This vector is an estimate of the normal to the best line passing through a neighborhood of the pixel. The vectors are estimated using principal component analysis with outlier rejection based on the median of the residuals.

Second, the vector field, the marked pixels, and their associated neighborhoods are used to cluster marked pixels into line support regions, and the line equation corresponding to each cluster is computed. Clusters can be ranked based on the number of members and their spatial extent. The output of the estimation procedure is set of rank-ordered putative image line segments.

The key insight of the proposed line estimation

[*] The work of the first three authors was partially supported by a grant from the Tokyo Research Laboratory of IBM-Japan, and by the New Jersey Center for Multimedia Research.

scheme is the use of arbitrarily shaped neighborhoods of marked pixels to compute the associated vector field and to cluster subsets of pixels into line support regions. Using these neighborhoods avoids some of the traditional problems in line detection, such as accurately estimating gradients at pixels in an image based on very local information, and using these noisy estimates as the basis for an edge-linking algorithm.

The estimation problems are formally posed and prior work briefly reviewed in section 2. Section 3 describes the details of the proposed line estimation algorithm. Detection accuracy results are reported in section 4, and the application to projective transformation estimation is discussed in section 5.

2 Problem formulation and prior work

It is well known that if two images $\mathcal{I}$ and $\mathcal{I}'$ are taken by a strictly rotating camera, the corresponding image coordinates $w = (x, y)^T$ and $w' = (x', y')^T$ are related by a projective transformation of the form:

$$w' = \frac{Aw + b}{cw + d}, \quad A \in R^{2\times 2}, b \in R^{2\times 1}, c \in R^{1\times 2}, d \in R.$$

The parameters of the transformation can be estimated by determining point correspondences between the image pair and solving a least-squares problem [9, 12, 23]. Alternate methods involve estimation of the optical flow at each pixel in the image [13] or direct minimization of intensity difference [20, 22]. However, it is often difficult to use these schemes in images with large, uniform regions. See, for example, figure 6a.

Correspondences between lines in the image pair can also be used together with point features to estimate the projective transformation parameters. However, because there are usually only a small number of lines (3-10) in each image, the equations of the detected lines must be estimated with high accuracy, and falsely detected and matched lines are problematic. However, every line in an image need not be detected, as long as the detected lines are well-estimated. Furthermore, a long line which is detected as two or three broken line segments is not harmful, provided the segments are well-estimated and the correct correspondences are made.

This leads naturally to a line detection and estimation problem. In some contexts, only thin straight line segments may be sought, whereas in other applications the detection of edges or other linear features may be desired. Our algorithm can handle either situation.

Many line detection algorithms operate on a binary image in which marked pixels correspond to pixels on image lines. Regardless of the method of extraction, the binary image will always be corrupted by noise and by the non-ideal character of natural image lines. A line detection scheme should be robust to these corruptions.

An extensive family of line detection algorithms is based on the Hough transform [10]. A line can be expressed by the angle its normal makes with the positive x-axis, θ, and its radial distance from the origin, ρ. Each pixel in the binary map is transformed to a curve in (ρ, θ) space which corresponds to all possible lines which pass through the point. The (ρ, θ) plane is quantized to form an accumulator array and the cells corresponding to each map pixel's curve are incremented. A search for the local maxima of the accumulator array produces estimates of the lines detected in the image. The principal deficiencies of such algorithms are the prelimited spatial accuracy due to quantization of the (ρ, θ) space, and the diffuse peaks in the accumulator array. As a result, lines are often broken or replicated in the Hough detection process (see [7]). Many extensions of the Hough transform have been proposed, e.g. modification to estimate line segment endpoints and width (see [3, 25]), but all such extensions suffer from these basic limitations.

Another approach to line detection is based on the edge linking techniques introduced by Nevatia and Babu [17]. Paths of pixels in the binary map are generated by selecting predecessors and successors for each marked pixel. Venkateswar and Chellappa [24] describe a method for linking pixels in the map which explicitly enforces the requirements that linkages occur along straight lines. The linking is performed on the basis of roughly quantized gradient estimates. Generally, edge linking algorithms require special steps to connect collinear fragments, bridge gaps in lines, and eliminate detected "lines" which have no physical significance. Even after heuristic steps to reduce the number of small and fragmented lines, many hundreds of lines are often detected.

Many other techniques do not fall into either of these two categories. In particular, we mention the Burns edge detector [4], in which gradient orientations are estimated at each pixel and quantized into a small number of bins. A connected components scheme is then used to group pixels into line support regions, and a plane fitted to the intensities of the points in each line support region is used to produce a line equation estimate. Kahn et al. [11] made several modifications to make Burns' scheme more efficient, including the use of principal component analysis to estimate line equations in each line support region. McLean et al. [14] replaced Burns' connected components method of generating line support regions with a region growing technique.

The scheme proposed in this paper is similar to the Burns detector, but attempts to avoid some of its problems by ensuring that the vectors associated with real image line pixels are accurate estimates of the normals to the corresponding lines, and applying a clustering algorithm to more widely supported neighborhoods to avoid fragmentation.

3 Line detection and estimation

We begin with a binary map. A vector is associated with each marked pixel in the binary map by applying principal component analysis to the marked pixels in an associated neighborhood of the marked pixel. Each neighborhood can generally be any shape but the neighborhood size should be on the same order as the largest line width it is desired to detect, and small relative to the size of the image.

For a specific marked pixel, the marked pixels in its associated neighborhood, denoted by the set $\{(x_i, y_i), i = 1 \ldots K\}$, are normalized to have zero mean and collected into a matrix R and a crosscorrelation matrix S:

$$R = \begin{pmatrix} x_1 & x_2 & \cdots & x_K \\ y_1 & y_2 & \cdots & y_K \end{pmatrix}$$

$$S = RR^T = \begin{pmatrix} \sum_{i=1}^K x_i^2 & \sum_{i=1}^K x_i y_i \\ \sum_{i=1}^K x_i y_i & \sum_{i=1}^K y_i^2 \end{pmatrix}$$

We compute the eigenvector associated with the eigenvalue of largest magnitude of S, and use the orthogonal vector as the estimate of the normal to the line passing through the pixel.

When the marked pixels deviate from an actual line by zero-mean Gaussian noise, the principal component is the maximum likelihood estimator of the line's direction. However, Gaussian deviation is unlikely to be true in practice. Like any estimator which minimizes a least-squares residual, principal component analysis is subject to corruption by outlying samples. Therefore, samples for which the residual exceeds the median residual by a number of standard deviations from the median are rejected [19], and the principal component analysis recomputed. This process can be iterated, but in practice we use one step of outlier rejection with a threshold of 1.5 median standard deviations. That is, for a sample point with residual r_i, reject the point if:

$$\frac{|r_i - r_{med}|}{\sigma_{med}} < 1.5, \quad \text{where}$$
$$r_{med} = \text{median}\{r_1, \ldots r_K\} \quad \text{and}$$
$$\sigma_{med} = \frac{1}{K}\sum_{i=1}^K (r_i - r_{med})^2$$

To avoid computational overhead in the following stage, vectors need not be computed for each marked pixel, but only for a selected subset, e.g. those pixels falling on a certain lattice.

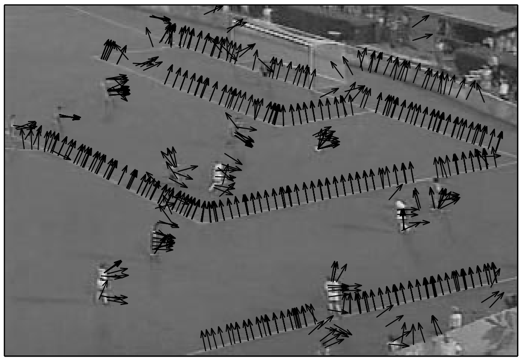


Figure 2: Result of vector field computation.

Figure 2 shows a typical result of the vector field computation, applied to the binary map of figure 6b. Vectors are computed for marked pixels falling on a rectangular lattice 5 pixels wide. For those marked pixels which lie on image lines, the corresponding vector well-approximates the normal to the line.

At this stage we possess a binary image for which each marked pixel is associated with a neighborhood and a vector. Next, a clustering scheme similar to that used for vector quantization [1] is applied which groups neighborhoods of marked pixels into line support regions. The proposed scheme circumvents many of the problems traditionally associated with edge linking, such as gap bridging and thinning. However, the results of the technique depend on the order in which neighborhoods are clustered.

In pseudocode, the clustering procedure is as follows:

```
1. List the marked pixels from 1 to $N$ in any reasonable order.
2. Create a cluster containing neighborhood 1, and set $c = 1$.
3. For j = 2 to $N$, do:
    (a) Compute the distance $q_k, k = 1, \ldots, c$, defined below, of neighborhood $j$ to each of the each of the $c$ clusters.
    (b) If $q_{k^*} = \min q_k$ is below a threshold $\epsilon$, add neighborhood $j$ to cluster $k^*$, and re-estimate the line equation corresponding to the cluster.
    (c) Otherwise, create a new cluster containing neighborhood $j$, and set $c = c + 1$.
```

The line equation for each cluster is estimated in step 3b above using principal component analysis (with outlier rejection) on all marked pixels in all the neighborhoods of the cluster.

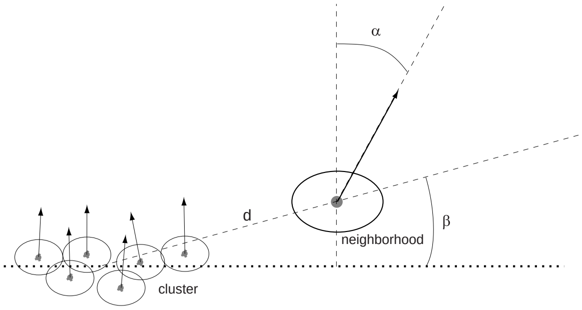


Figure 3: Proposed clustering metric.

The distance function from neighborhood j to cluster k used in step 3a above is defined as:

$$q_k = |\alpha_k| + |\beta_k| + M * 1_{\{d_k > \tau\}}$$

where α, β, and d are sketched in figure 3. This choice of metric ensures that the direction associated with the candidate neighborhood is similar to the average direction associated with the neighborhoods in the cluster, the candidate neighborhood lies near the line defined by the neighborhoods in the cluster, and the candidate neighborhood is spatially close to the neighborhoods in the cluster. The constants M and τ are chosen such that a neighborhood sufficiently far from existing clusters becomes the center of a new cluster. This metric is not unique and several alternatives are clearly possible.

At the end of the clustering step, clusters are rank-ordered, and clusters whose total line support region is of insufficient spatial extent or is comprised of too few neighborhoods may be rejected if desired.

A typical result of the clustering algorithm is shown in figure 4. Marked pixels which lie on non-linear features such as players do not corrupt the line estimates, and players occluding the back and side lines do not fragment the detected lines.

4 Experimental results

Due to space limitations, experimental results are only reported for two pairs of images. One image from each pair is pictured in figures 6a and 7a. The neighborhoods used in the algorithm are 20 by 20 blocks of pixels for set 1 and 10 by 10 blocks of pixels for set 2. Detected lines are displayed in black at a fixed width for visibility.

The first step is to obtain the binary map. Standard edge operators such as the Canny operator [5] can be used. However, these often produce binary maps that are complicated in highly detailed regions of the original image and thus not well-suited to extracting straight lines (see figure 5a). Here, we use an algorithm based on a series of simple filtering and thresholding operations. Space limitations prevent a detailed description of the method. However, an example is shown in figure 5b.

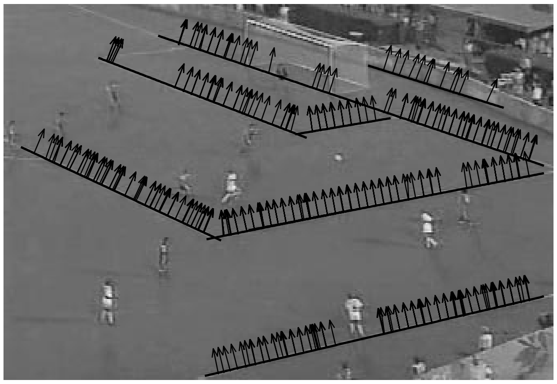


Figure 4: Result of neighborhood clustering.

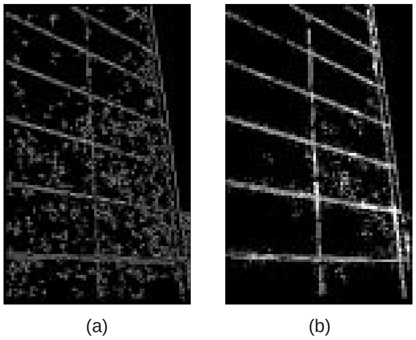


Figure 5: (a) Canny edge map, (b) binary image from proposed scheme.

Figures 6 and 7 show the original image (a), the input binary map (b), line segments detected using the proposed algorithm (hereafter PA) (c), and the lines corresponding to the top local maxima of a low-pass filtered Hough transform (d). Local maxima of the Hough transform were extracted until a match to a PA detected segment was found, or 50 local maxima were computed. Even with initial smoothing to dispel the effects of closely spaced local maxima, the Hough transform suffers from multiple responses to a single line, and fits spurious lines to collinear image features. Since ground truth is unavailable for these outdoor images, the "correct" line segment equations in each image were measured by hand and used to determine the errors in the estimates. The average errors between the PA line segments and the correct line segments in the radial and angular parameters (ρ, θ) are reported in table 1. The average errors of the best corresponding set of lines obtained from the Hough transform are also reported for comparison. Results for a synthetic image for which ground truth is known are reported in the fifth row of table 1.

5 Transformation estimation

In this section, we demonstrate an application of line estimation to the problem of estimating the projective transformation relating two images $\mathcal{I}$ and $\mathcal{I}'$ taken by a rotating camera.

If a line $\ell_i : n_{i1}x + n_{i2}y + k_i = 0$ in the first image and a line $\ell'_i : m_{i1}x+m_{i2}y+l_i = 0$ in the second image correspond to the same line in the 3-D world, the two lines are related to each other via the equation:

$$[m_{i1} \quad m_{i2} \quad l_i] \begin{bmatrix} A & b \\ c & d \end{bmatrix} = [n_{i1} \quad n_{i2} \quad k_i]$$

The parameter d can be normalized to 1, so that each line correspondence produces a linear system $M_i p = z_i$ of two equations in the eight remaining projective transformation parameters $p = [a_{11}\ a_{12}\ a_{21}\ a_{22}\ b_1\ b_2\ c_1\ c_2]^T$:

$$\begin{bmatrix} m_1 & 0 & m_2 & 0 & \frac{-n_1 m_1}{k} & \frac{-n_1 m_2}{k} & l & 0 \\ 0 & m_1 & 0 & m_2 & \frac{-n_2 m_1}{k} & \frac{-n_2 m_2}{k} & 0 & l \end{bmatrix} p = \begin{bmatrix} \frac{n_1 l}{k} \\ \frac{n_2 l}{k} \end{bmatrix}$$

When q line correspondences have been obtained, an estimate of the projective transformation parameters can be obtained by solving the weighted linear least squares problem given by $MWp = z$, where:

$$M = \begin{bmatrix} M_1 \\ \vdots \\ M_q \end{bmatrix}, W = \begin{bmatrix} W_1 & & \\ & \ddots & \\ & & W_q \end{bmatrix}, z = \begin{bmatrix} z_1 \\ \vdots \\ z_q \end{bmatrix} \qquad (1)$$

The weighting of the equations contributed by each line pair is based on the lengths of the detected segments, as well as a measure of confidence in their matching.

To illustrate the technique, projective transformation parameter estimation was applied to the images Soccer 1 and Soccer 2, which are 100 frames apart in an MPEG-1 compressed video sequence. Each of the images contains few robust point features on which to base the estimation. However, each image contains several strong lines which, if extracted accurately, can be used to estimate the projective transformation.

Binary maps were extracted from the two images and processed with the algorithm discussed in section 3 to yield a set of line segments for each image. To test the line estimation algorithm, line segments were matched between the image pair by hand, and the least squares problem (1) solved. Five line correspondences were used to estimate the projective transformation parameters. The resulting estimated projective transformation was used to render both images onto the same plane. The registered image pair is displayed in figure 8a. The same images registered using a point feature matching technique are displayed in figure 8b. White lines delineate the borders of the original images. In the regions where the registered images overlap, the intensities from both images are averaged, producing "ghosts" in areas where players have moved.

The point matching technique fails because of the sizable zoom between the images and the sparseness of robust point features. However, the line matching technique produces an accurate registration in spite of the small number of lines that were used to compute the transformation parameters. Since all five of the matched lines lie in the plane of the soccer field, objects which lie outside of this plane (e.g. the soccer goal) are not precisely matched. Using point correspondences in addition to the line correspondences should correct this problem.

6 Conclusions

We have shown that a small number of accurately estimated line segments can greatly aid in the estimation of the projective transformation relating two images. The proposed line detection and estimation scheme is suitable for other line detection problems where the tolerance for false detection is very low and the need for accuracy in the reported line segments is very high. The algorithm is robust to both noise in the binary input image and poorly estimated neighborhood vectors in the clustering stage. Furthermore, the scheme delivers clear, easily interpreted results, without the limited resolution and spurious lines from which other schemes suffer.

References

[1] H. Abut, editor. Vector Quantization. IEEE Press, 1990.

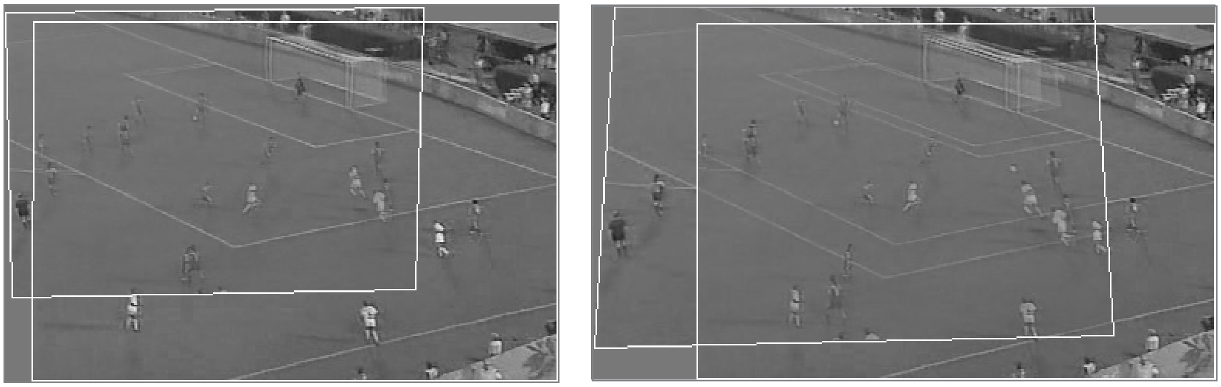
[2] H.K. Aghajan and T. Kailath. SLIDE: Subspace-Based Line Detection. *IEEE PAMI*, Vol. 16, No. 11, pp. 1057-1073, 1994.

[3] M. Atiquzzaman and M.W. Akhtar. A Robust Hough Transform Technique for Complete Line Segment Description. *Real-Time Imaging*, Vol. 1, pp. 419-426, 1995.

[4] J.B. Burns, A.R. Hanson, and E.M. Riseman. Extracting Straight Lines. *IEEE PAMI*, Vol. 8, No. 4, pp. 425-455, 1986.

Image	Detected PA Lines	Avg. PA ρ err. (pix.)	Avg. Hough ρ err. (pix.)	Avg. PA θ err. (rad.)	Avg. Hough θ err. (rad.)
Soccer 1	8	0.1836	3.8410	0.0010	-0.0124
Soccer 2	5	0.0619	4.6502	0.0007	-0.0138
Wall 1	18	1.1417	1.7587	-0.0054	-0.0046
Wall 2	15	1.2473	1.5849	-0.0023	-0.0047
Synthetic	7	-0.3500	2.8959	0.0023	-0.0085

Table 1: Typical experimental results.



(a) result using 5 line features

(b) result using 57 point features

Figure 8: Soccer pair registration.

[5] J.F. Canny. Finding Edges and Lines in Images. MIT AI Laboratory Technical Report No. 720, June 1983.

[6] A.C. Copeland, G. Ravichandran, and M.M. Trivedi. Localized Radon Transform-Based Detection of Linear Features in Noisy Images. *Proc. CVPR '94*, pp. 664-667, 1994.

[7] L. da Fontoura Costa. Small Camera Movements as a Means of Reducing the Amount of Broken and False Detected Lines in Hough Transform. *Real-Time Imaging*, Vol. 2, pp. 181-185, 1996.

[8] R.I. Hartley. Projective Reconstruction from Line Correspondences. *Proc. CVPR '94*, pp. 903-907, 1994.

[9] T.S. Huang and A.N. Netravali. Motion and Structure from Feature Correspondences: A Review. *Proc. IEEE*, Vol. 82, No. 2, pp. 251-268, February 1994.

[10] A.K. Jain. Fundamentals of Image Processing. Prentice Hall, 1989.

[11] P. Kahn, L. Kitchen, and E.M. Riseman. A Fast Line Finder for Vision-Guided Robot Navigation. *IEEE PAMI*, Vol. 12, No. 11, 1990.

[12] K. Kanatani. Geometric Computation for Machine Vision. Clarendon Press, 1993.

[13] S. Mann and R. W. Picard. Video orbits of the projective group; a simple approach to featureless estimation of parameters. In *IEEE Trans. Image Proc.*, Vol. 6, No. 9., September 1997.

[14] G.F. McLean, B. Prescott, and D. Kotturi. Hierarchical Clustering for Automated Line Detection. *IEEE Pac. Rim '93*, pp. 244-247, 1993.

[15] S.K. Nayar, S. Baker, and H. Murase. Parametric Feature Detection. *Proc. CVPR '96*, pp. 471-477, 1996.

[16] R.C. Nelson. Finding Line Segments by Stick Growing. *IEEE PAMI*, Vol. 16, No. 5, pp. 519-523, 1994.

[17] R. Nevatia and K.R. Babu. Linear Feature Extraction and Description. *CGIP*, Vol. 13, pp. 257-269, 1980.

[18] L. Quan. Uncalibrated 1-D Projective Camera and 3-D Affine Reconstruction of Lines. *Proc. CVPR '97*, pp. 60-65, 1997.

[19] P.J. Rousseeuw and A.M. Leroy. Robust Regression and Outlier Detection. John Wiley and Sons, 1987.

[20] H.S. Sawhney, S. Ayer, and M. Gorkani. Model-based 2D and 3D Dominant Motion Estimation for Mosaicing and Video Representation. *Proc. ICCV '95*, pp. 583-590, 1995.

[21] C. Schmid and A. Zisserman. Automatic Line Matching across Views. *Proc. CVPR '97*, pp. 666-671, 1997.

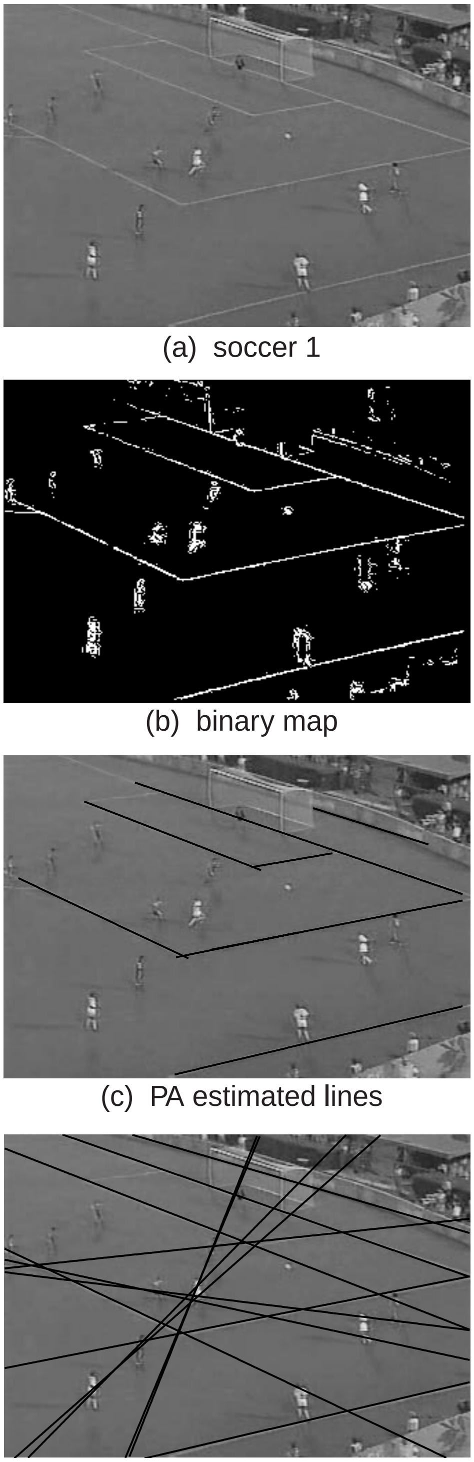
[22] R. Szeliski. Image Mosaicing for Tele-Reality Applications. Digital Equipment Corporation, Technical Report CRL 94/2, May 1994.

[23] Y.P. Tan, S. Kulkarni, and P. Ramadge. A New Method for Camera Motion Parameter Estimation. *Proc. ICIP '95*, vol. 1, pp. 406-409, 1995.

[24] V. Venkateswar and R. Chellappa. Extraction of Straight Lines in Aerial Images. *IEEE PAMI*, Vol. 14, No. 11, pp. 1111-1114, 1992.

[25] M.C.K. Yang, J-S. Lee, C-C. Lien, and C-L. Huang. Hough Transform Modified by Line Connectivity and Line Thickness. *IEEE PAMI*, Vol. 19, No. 8, 1997.

[26] Z. Zhang. Estimating Motion and Structure from Correspondences of Line Segments Between Two Perspective Images. *Proc. CVPR '95*, pp. 257-262, 1995.



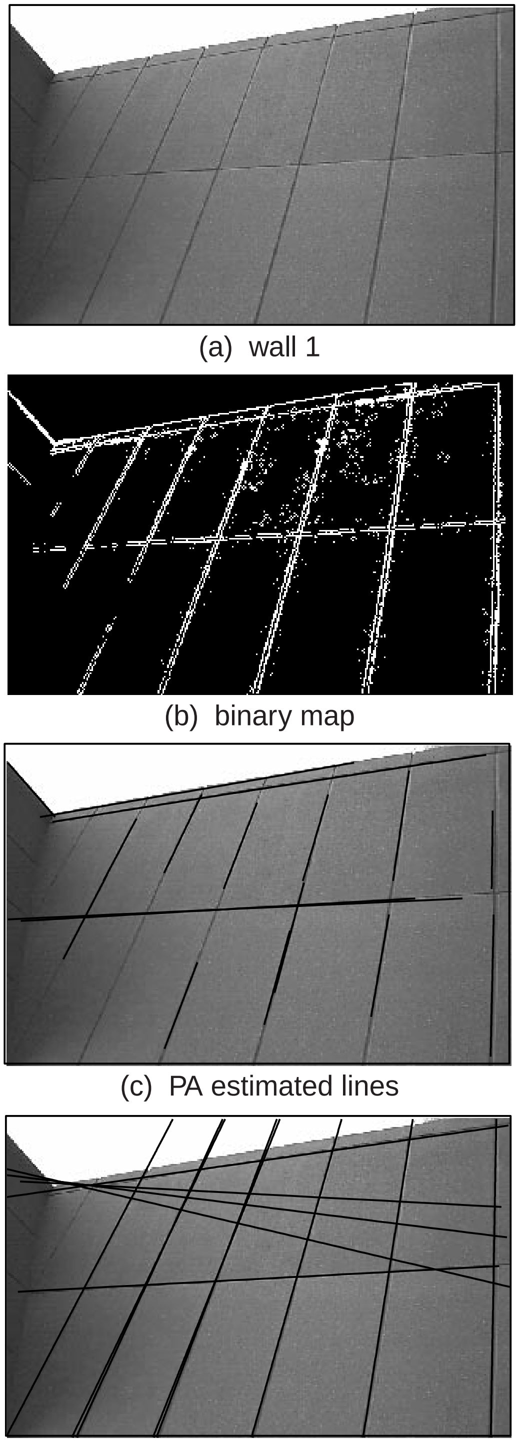
(a) soccer 1

(b) binary map

(c) PA estimated lines

(d) Hough estimated lines

Figure 6: Soccer image 1 (8 lines detected).



(a) wall 1

(b) binary map

(c) PA estimated lines

(d) Hough estimated lines

Figure 7: Wall image 1 (18 lines detected).