Perspective Estimation for Document Images

Christopher R. Dance
Xerox Research Centre Europe, 61 Regent Street, Cambridge CB2 1AB, England

ABSTRACT

There has been increasing interest in document capture with digital cameras, since they are often more convenient to use than conventional devices such as flatbed scanners. Unlike flatbed scanners, cameras can acquire document images with arbitrary perspectives. Without correction, perspective distortions are unappealing to human readers. They also make subsequent image analysis slower, more complicated and less reliable.

The novel contribution of this paper is to view perspective estimation as a generalisation of the well-studied skew estimation problem. Rather than estimating one angle of rotation we must determine four angles describing the perspective. In our method, separate estimates are made for angles describing lines that are parallel and perpendicular to text lines. Each of these estimates is based on a twice-iterated projection profile computation. We give a probabilistic argument for the method and describe an efficient implementation. Our results illustrate its primary benefits: it is robust and accurate. The method is efficient compared with the time required to warp the image to correct for perspective.

1. INTRODUCTION

Our research has been motivated by the convenience of using cameras for document capture, compared to conventional scanners.[1] Digital still cameras are portable and offer face-up, near-instantaneous image acquisition, but suffer from image quality problems resulting from the wide range of conditions in which they may operate.[2] One of the most severe problems is that cameras acquire document images with arbitrary perspectives. The presence of perspective is distracting to human readers and makes image analysis operations, such as text recognition,[3] layout analysis[4] and compression,[5] slower or less reliable.

The perspective transform of a family of parallel lines is a *pencil*: that is, a family of lines through a common point, known as a *vanishing point*. Document images can be described in terms of two orthogonal families of lines: one parallel to the text lines and one parallel to the borders of formatted text columns. Under perspective, we refer to these as the *X-pencil* and *Y-pencil* respectively.

The novel contribution of this paper is to estimate these pencils by a generalisation of skew detection methods. We evaluate each pencil independently in a two-phase procedure justified by a simple probabilistic model. The inferred parameters are then used to warp the image to remove the effects of perspective, as illustrated in Figure 1, or to define a co-ordinate system for any subsequent computations that are sensitive to errors introduced in warping. Our estimator is designed to robustly produce accurate results for images with unknown resolution and variations in content such as the presence of halftones and multiple text columns. It should also degrade gracefully in the presence of noise, page curvature and clutter. Estimation is fast relative to operations such as text recognition.

We are aware of only one previously published paper on document perspective estimation, which has appeared very recently.[6] However, there is a voluminous literature on document skew estimation[7] and perspective estimation for natural images. Some authors have addressed the related problems of locating text on oriented surfaces in natural scenes,[8] and of non-uniform skews from curved pages.[9]

Most successful document skew estimators[10] and perspective estimators for natural scenes[11] rely on voting to cope with outliers and multiple lines. Scores are constructed from votes for a range of skew or perspective parameter values. The parameter value with maximum score is the estimate. Whereas techniques for natural scenes typically assume that continuous edges are available whose orientations can be accurately measured, skew estimators typically extract votes by counting the number of points of some type along lines across the entire image.[12,13] To extract a score, votes for different lines at the same skew angle must be combined, often by taking sums of some function (usually the square) of the votes.[12] A more accurate and robust score,[10,14]

closely related to that developed in this paper, is the variance in the difference in the number of points on adjacent lines.

The next section explains our method. Our results, in Section 3, study the robustness and accuracy of our method, upon which Section 4 concludes.

2. ESTIMATOR

Given a perspective image of a document, our method evaluates the X- and Y-pencils independently and then combines them into an estimated perspective, as shown in Figure 2. Independent evaluation of the pencils allows efficient computation and the application of different statistical models for text lines and borders of text columns. A convenient representation for a pencil is in terms of the slopes and intercepts of its lines with some other line. The key observation is that a pencil is a set of lines with a *linear relation between slope and intercept.* Our method therefore stores votes for lines in an accumulator with bins parameterised by slope and intercept. The score for a pencil is computed simply from the votes on one line through the accumulator.

A pair of pencils has four parameters, whereas a perspective transform has eight. The remaining parameters correspond to two translations and two scale factors. To perform perspective correction, we select the translation to centre the corrected image and the scale factors so that the average lengths of the sides of the observed and corrected images are the same. We interpolate by backwards mapping to prevent gaps: i.e. we iterate over each pixel in the corrected space extracting the corresponding point in the observed image from a formula for the perspective transform.

The most problematic part of this process is scoring the Y-pencil. This is largely because there are often few vertical lines on a page, hence the estimates are more prone to noise and outliers. As illustrated in Figure 3, the projection profile of vertically superposed columns also usually contains less alignment information than that for horizontally adjacent columns. We overcome these difficulties partly by selecting points from the image that form vertically oriented structures and partly by exploiting a simple probabilistic model.

2.1. Probabilistic Model

Our model is similar to considering the sequence of adjacent lines in a pencil as the "time-steps" in a 1D Hidden Markov Model with two states labelled T, B. State T describes text edges, such as text baselines in an X-pencil or column boundaries in a Y-pencil. State B describes background noise that contains no useful alignment information. However, our model differs slightly from a Hidden Markov Model, since we shall model transition probabilities between *consecutive* T states, rather than between all *adjacent* states. A sensible objective for pencil estimation would be to maximise the posterior of the pencil given the observed image I. However, for computational efficiency, we make the Viterbi approximation[15] that the posterior for the pencil can be replaced by the posterior for the pencil *plus state labels* for the lines.

As in document image decoding,[3,16] scores are defined by taking the logarithm of the posterior normalised by the probability of an all background image. To compute scores we factor this posterior into a product of conditionally independent observations. Each observation is a vote I_λ corresponding to a line λ of the pencil Λ. Let S_T, S_B denote a set of lines labelled T, B from a pencil Λ, so that $S_T \cup S_B = \Lambda$. The joint probability for this labelling is

$$\Pr(I, \Lambda, S_T) = \Pr(\Lambda) \Pr(S_T) \prod_{\lambda \in S_T} \Pr(I_\lambda | T) \prod_{\lambda \in S_B} \Pr(I_\lambda | B) \tag{1}$$

and the probability of the all background image is

$$\Pr(I, \Lambda, \emptyset) = \Pr(\Lambda) \Pr(S_T = \emptyset) \prod_{\lambda \in \Lambda} \Pr(I_\lambda | B). \tag{2}$$

Clearly the ratio of these is the same as the ratio of posteriors

$$\frac{\Pr(\Lambda, S_T | I)}{\Pr(\Lambda, \emptyset | I)} = \frac{\Pr(I, \Lambda, S_T)}{\Pr(I, \Lambda, \emptyset)} = \frac{\Pr(S_T)}{\Pr(\emptyset)} \prod_{\lambda \in S_T} \frac{\Pr(I_\lambda | T)}{\Pr(I_\lambda | B)}, \tag{3}$$

whose logarithm gives us the following score

$$\text{score}(\Lambda) = \max_{S_T \subseteq \Lambda} \left\{ \log \frac{\Pr(S_T)}{\Pr(\emptyset)} + \sum_{\lambda \in S_T} \log \frac{\Pr(I_\lambda|T)}{\Pr(I_\lambda|B)} \right\}. \tag{4}$$

It is worth considering three implicit issues relating to the use of this score. Firstly, maximising the given score is only equivalent to maximising the posterior if the likelihood of the all background image $\Pr(I, \emptyset|\Lambda)$ is independent of Λ. This can only be true for certain choices of $\Pr(I_\lambda|B)$. Secondly, for discrete 2D images, it may be hard to ensure that votes for different lines from a pencil are strictly independent: in an extreme case, where the common point of a pencil appears in the image, all lines will contain a contribution from that point. Finally, the notion that all pixels on a given line can be modelled as coming from one state T or B is questionable: for example, a given line through an image often contains a mixture of text lines, noise and graphics. In our practical experience, these approximations do not appear problematic.

We now consider models for the conditional densities $\Pr(I_\lambda|\cdot)$ and transition probabilities $\Pr(S_T)$ for each type of pencil.

2.2. X-Pencil Score

Our X-pencil score is a direct generalisation of the skew estimator proposed by Postl,[14] aside from a thresholding step. This estimator takes the sums along lines through the image for a set of slopes (or skew angles) m and for all y-intercepts y. We consider the vote I_λ for a line $\lambda = (m, y)$ to be the difference between of the sums on adjacent lines:

$$I_\lambda = \sum_x I(x, mx + y + 1) - I(x, mx + y). \tag{5}$$

The votes are modelled as normally distributed with zero mean for both states, but with a larger variance for state T than for state B: $\sigma_T^2 > \sigma_B^2$. Thus

$$\log \frac{\Pr(I_\lambda|T)}{\Pr(I_\lambda|B)} = \frac{I_\lambda^2}{2\sigma_B^2} - \frac{I_\lambda^2}{2\sigma_T^2} + \log \frac{\sigma_B}{\sigma_T}. \tag{6}$$

All lines in a pencil are independently of type T or B with some fixed prior probability, so that $\log \frac{\Pr(S_T)}{\Pr(\emptyset)}$ is just proportional to the cardinality of S_T.

This leads to the following score derived from (4)

$$\text{score}(\Lambda) = \max_{S_T \subseteq \Lambda} \sum_{\lambda \in S_T} I_\lambda^2 - \tau \tag{7}$$

for some threshold value τ. Our X-pencil esimator simply chooses the value of Λ which maximises this score.

Generally the threshold τ depends on the variances of votes and the prior probability of state T, which are unknown and vary from image to image. Rather than fixing τ, we choose it so that few background values of I_λ are likely to lie above it, exploiting analogy with results on minmax risk estimators.[17] In accordance with these results, we take a threshold $\tau = \sigma\sqrt{2 \log N}$ where N is the number of values at which I_λ has been evaluated and σ is a robust estimate of the "background noise" in I_λ

$$\sigma = \frac{\text{median}|I_\lambda|}{0.6745}. \tag{8}$$

The statistical benefit of this threshold can be seen intuitively by considering a page containing only a few text lines, but which has noise spread over large regions, which might originate from areas of desk around the document that are visible in the camera image. The votes I_λ corresponding to the text lines have large magnitudes, but without the threshold, the cumulative sum of many small noise terms in (7) terms could dominate, causing a spurious estimate. A corresponding threshold arises in the estimation of the Y-pencil, where this benefit is more significant since there are typically far fewer vertical lines than horizontal lines on a page. The second benefit of this threshold is that it typically zeros most of the terms in the score's sum, thus speeding up score computation by an order of magnitude.

2.3. Y-Pencil Score

In order to improve the signal-to-noise ratio of column boundaries, our Y-pencil estimator extracts votes by counting the number of black points on a line that are adjacent to relatively large white spaces. This is justified by the success of distance-based methods for page layout analysis.[4,18] As illustrated in Figure 4, we detect such points using a triangle of some height s. Since the image resolution and page layout are unknown a priori, we collect votes $I_{\lambda,s}$ on each line λ for several values of the scale parameter s. A triangle makes this computationally convenient since points detected at one scale are also detected at any smaller scale.

Point Density Models. We consider the simplest plausible point distribution model for these votes: a spatially uniform distribution. The number of such points n observed in some area A is given by the Poisson distribution with mean ρ

$$p(n;\rho) = e^{-A\rho}\frac{(A\rho)^n}{n!}. \tag{9}$$

In particular, we assume that the vote for a line with state T, such as a column border, is a Poisson variate with a mean which is constant for a given image. The background state B is considered as a mixture of two Poisson distributions modelling clutter and noise. The clutter term models strong outlier lines that do not form part of the true pencil and has a mean density equal that for text edges. Meanwhile, the noise term has a spatially variant mean density, to capture phenomena such as points arising from non-aligned text borders such as the sides of a block of centred text.

Formulas for the terms in (4) corresponding to these models may be stated in terms of the number $n = I_{\lambda,s}$ of points detected on line λ, the mean density of points on text edges ρ, the noise density $\mu_{\lambda,s}$ and the ratio of clutter-to-noise γ:

$$\Pr(I_{\lambda,s} = n|T) = e^{-\rho}\frac{\rho^n}{n!} \tag{10}$$

$$\Pr(I_{\lambda,s} = n|B) = (1+\gamma)^{-1}\left(e^{-\mu_{\lambda,s}}\frac{\mu_{\lambda,s}^n}{n!} + \gamma e^{-\rho}\frac{\rho^n}{n!}\right). \tag{11}$$

Spacing Model. Typically text edges are not immediately adjacent to each other. We control this by setting $\Pr(S_T)$ to zero if S_T contains two lines that are too close together. Otherwise each text edge or background line is considered to be equally probable. Given two lines λ_1, λ_2 we measure the distance between them on the horizontal line through the centre of the image. This distance is denoted by $d(\lambda_1, \lambda_2)$. For a scale s, we require that

$$d(\lambda_1, \lambda_2) \geq s\Delta. \tag{12}$$

In practice, the spacing between lines is an important part of our model. Figure 5 shows a map of point counts for a typical line in slope-intercept space. It has a characteristic X-shape. Without a constraint on spacing between lines, the score will be larger for lines that pass through the tails of this X-shape than for lines that pass through its centre. Hence the estimator will suffer from increased bias and variance.

Score. These models may be combined into a score of the form of (4):

$$\text{score}(\Lambda, s) = \max_{S_T \subseteq \Lambda} \sum_{\lambda \in S_T} f(I_{\lambda,s}, \mu_{\lambda,s}) \; - \; \kappa \tag{13}$$

where κ is a constant determined by the ratio of text edges to background, which acts like the threshold for the X-pencil estimator, and

$$f(n,\mu) = \log \frac{e^{\mu-\rho}\left(\frac{\rho}{\mu}\right)^n}{1+\gamma\, e^{\mu-\rho}\left(\frac{\rho}{\mu}\right)^n}. \tag{14}$$

Our Y-pencil estimator maximises this score over Λ *and* s subject to the spacing constraint (12).

Of course, in practice, it is not the exact form of this complex-looking score which is critical, but rather its global behaviour, which differs in two key ways from the X-pencil score. Firstly, it can adapt to the level of

background noise, without which it is hard to cope with unformatted text borders and noise variations induced by pictures. Secondly, the contribution to the score from any individual vote is upper bounded, because of the clutter term. Without such a bound, we find that our method would regularly fail in situations where multiple pages are present in a single image.

The evaluation of this score for a given pencil and scale Λ, s may be written as a dynamic programming recurrence. For this purpose denote the column intercept of a line λ from Λ by $c = 1, \ldots, w$, where w is the width of the image. Accordingly let

$$f_c(\Lambda) = f(I_{\lambda,s}, \mu_{\lambda_s}). \tag{15}$$

In terms of a *value function* $V_c(\Lambda)$, the recurrence is

$$\begin{aligned} V_c(\Lambda) &= f_c(\Lambda) \text{ for } c = 1, \ldots, s\Delta & (16)\\ V_c(\Lambda) &= \max\{f_c(\Lambda) + V_{c-s\Delta}(\Lambda), V_{c-1}(\Lambda)\} \text{ for } c > s\Delta & (17)\\ \text{score}(\Lambda, s) &= V_w(\Lambda). & (18) \end{aligned}$$

2.4. Implementation

At the top level, our estimator has three main steps: the X-pencil score (7) is evaluated by summing along lines through an accumulator; dynamic programming problems are solved for each Y-pencil score (13); in each case, these scores are evaluated for many pencils and the pencil giving maximum score is the estimate. To illustrate an efficient choice of implementation, we break these steps up as follows:

1. Count the number of black pixels along a set of nearly-horizontal lines through the input binary image J. This is efficiently accomplished by adding a line to an accumulator of pixel counts for each black pixel encountered in J. A similar counting scheme is employed in steps 2 and 4. The accumulator can itself be viewed as an image with rows corresponding to line intercepts and columns corresponding to line slopes. Take the difference between adjacent rows of this accumulator, resulting in array of X-pencil votes as in (5).

2. Compute the threshold of (8) and transform each vote to the corresponding term of (7), *i.e.* $\max\{I_\lambda^2 - \tau, 0\}$. Take sums along each line Λ through this array of transformed votes. Each such sum is the score of a particular X-pencil. The maximum sum corresponds to the X-pencil estimate.

3. Determine a scale s for the central left- and right-most points of each black connected component in J. This is accomplished by recursively growing a triangular region from such a point until the triangle hits a black pixel. The result can be viewed a *scale-image* in which each pixel stores the scale of the corresponding point in J, or zero if the scale of this point was not evaluated. Using the scale-image, choose a set of four threshold scales for evaluating the Y-pencil score.

4. Starting from the largest threshold scale s, count the number of pixels whose scale is at least s along each nearly-vertical line λ through the scale-image. This results in an array of Y-pencil votes $I_{\lambda,s}$, whose columns correspond to line intercepts and whose rows correspond to line slopes.

5. Compute a moving average along each row of these votes, to define the background mean $\mu_{\lambda,s}$. We find that our results are not particularly sensitive to the choice of moving average window size and typically use a value corresponding to 21 columns of J. Apply the function f of (14) to obtain an array of transformed votes, where we set $\rho = 10, \gamma = 0.01, \kappa = 0$.

6. Solve the dynamic programming recurrence (16-18) to compute the score (13) for each line Λ through the array of transformed votes. Fast iteration is possible by exploiting the sparsity of this array: for each non-zero entry, we update the value function for only those Λ for which $\lambda \in \Lambda$. Memory efficient iteration is also possible: for each Λ we maintain a list of only those $V_c(\Lambda)$ that could be relevant to future iterations.

7. Iterate steps 4-6 for each threshold scale, noting that it is possible to update the vote array at each stage, rather than restarting each time. The pencil with largest value over all scales is the Y-pencil estimate.

Various improvements to this basic scheme are desirable. Rather than working directly with a full resolution image, substantial speed improvements are possible by working with lower resolution versions of the input binary image and refining estimates with higher resolution information only for limited ranges of Λ. When working with connected components at step 4, it is best to pre-filter the image to remove small noise components. Finally, measures of confidence should be associated with the estimates, so difficult cases may be rejected. To measure confidence, we essentially threshold the posterior for the estimates, by comparing the sum of exponentiated scores for pencils in the vicinity of the estimate with a corresponding sum for exponentiated scores outside this neighbourhood. Scaling factors applied prior to exponentiation were empirically determined.

3. RESULTS

We acquired a database of document images with a handheld camera with a resolution of 2048×1536 pixels, consisting of ten images of each of 15 document types. The image resolutions varied from 150 to 300 dpi and the predominant font sizes varied from 8 to 12 pt. The angle between the optical axis of the camera and the normal to the page varied by up to 20 degrees. In most of the images, underlying documents and regions of desk were visible and some samples had significant page curvature.

The orientation of the images was detected by a grayscale version of the method described in.[10] The images were then upsampled by 2x and binarised by background surface thresholding.[2] Ground truth text lines and column borders were determined by manually clicking points on horizontal and vertical lines for each binary image. Pencils were determined from these lines by an iterative least squares fit. To quantify error in the ground truth due to the inaccuracy of manual clicking and page curvature, we evaluated the root mean squared (RMS) error in the fit to each pencil. These errors are measured in pixels for resolutions between 300 and 600 dpi.

Evaluating the accuracy of pencil estimates is less intuitive than evaluating skew estimates, since perspective is spatially variant. We chose to evaluate the error in an estimated pencil by selecting the lines from the pencil that minimise the RMS error to the clicked points defining each line of the ground truth. Then, we found the RMS error over all clicked points.

Of the 150 images in the database, 2 gave a low value for the confidence criterion and were rejected by the X-pencil estimator and 15 were rejected by the Y-pencil estimator. Rejections for the Y-pencil estimator were due to large amounts of curvature, or to the absence of at least 2 long vertical lines, or to the presence of clutter from underlying pages. In order to compensate for decreased accuracy of the estimator in the presence of page curvature, we consider the RMS error in the estimate as a function of the RMS error in the ground truth. Figure 6 shows the RMS error in the non-rejected X- and Y-pencil estimates for each of three ranges of RMS error in the clicked points. Typical run-times on an 800MHz Pentium III computer were 0.6 seconds for the X-pencil estimation and 1.5 seconds for Y-pencil estimation. This compares favourably with 2.7 seconds to perform a backwards mapping to correct a full resolution binary image for perspective.

We also conducted an analogue of the experiment in[10] on 168 synthetic images from the University of Washington Database, subsampled by 2x. The objective is to measure the intrinsic error in the technique in the absence of curvature or noise. Of the 168 images, 15 were rejected by the X-pencil estimator and 26 by the Y-pencil estimator. These rejections seem to be indicative of the difficulty of perspective estimation: an intuitive reason for rejection could be found in each case. For instance several of the images only contain a few text lines which are all located in one region of the page. Thus, while a skew could be determined, there is insufficient information to estimate an accurate perspective over the entire image.

We account for variations in the aliasing of the images by measuring the two pencils when each image is rotated through each of 11 angles uniformly spaced between $\pm 2^\circ$. For comparability with skew detection results, here we measure the X-pencil in terms of the estimated angle for the line through the centre of the image and the change in angle between lines through the top-centre and bottom-centre. Analogous parameters are used for the Y-pencil. Cumulative histograms of the measured errors in these parameters for non-rejected samples are shown in Figure 7. Generally our X-pencil estimates are only about twice as bad as with the best conventional

skew detectors. The least satisfactory aspect of these results is that around 5% of the Y-pencil estimates remain in error by about 0.5°.

The stepping appearance of the histograms for angle change is due to the discretisation of the pencil space employed. The results for the central angle are effectively smoothed out because this discretisation is incommensurate with the uniform rotation steps, but the ground-truth angle change is always zero so no such smoothing occurs. The angle change histograms also exhibit small intermediate jumps between the large steps, which have the following explanation. When selecting the pencil with the maximum score, occasionally several pencils have the same value. Our estimate in such cases was taken to be the average of all pencils with this maximal score.

Finally, Figure 8 illustrates the application of our method to a range of document types. The example of a curved page of a book demonstrates how the method typically corrects for the "commonest" perspective in the image.

4. DISCUSSION AND CONCLUSIONS

We have presented a method for estimating the perspective of document images acquired with handheld cameras. The method separately evaluates two pencils according to a probabilistic model: the X-pencil describes text lines and the Y-pencil describes formatted column boundaries. The X-pencil estimator is a direct generalisation of existing document skew estimators and shares their excellent accuracy and robustness properties. The proposed Y-pencil estimator is slower, and our results indicate that it is less accurate and more likely to reject samples. We believe that this is because the Y-pencil is intrinsically more difficult to estimate.

One merit of our probabilistic approach is that it could be extended in future work to take account of additional cues, such as centred text, scaling effects on character height and interline spacing. It would also be interesting to exploit additional prior information on page size or character shapes to determine the scale factor by which a perspective corrected image should be stretched.

REFERENCES

1. W. Newman, C. Dance, A. Taylor, S. Taylor, M. Taylor, and A. Aldhous, "CamWorks: A video-based tool for efficient capture from paper source documents," in *Proc. IEEE International Conference on Multimedia Computing and Systems*, 1999.
2. M. Seeger and C. Dance, "Binarising camera images for OCR," in *Proc. 6th International Conference on Document Analysis and Recognition*, 2001. To appear.
3. G. Kopec and P. Chou, "Document image decoding using Markov source models," *IEEE Trans. Pattern Analysis and Machine Intelligence* **16**(6), pp. 602–617, 1994.
4. T. Breuel, "Layout analysis by exploring the space of segmentation parameters," in *Proc. 4th IAPR Workshop on Document Analysis Systems (DAS 2000)*, December 2000.
5. S. Inglis, *Lossless Document Image Compression*. PhD thesis, University of Waikato, 1999.
6. P. Clark and M. Mirmehdi, "Estimating the orientation and recovery of text planes in a single image," in *Proc. 12th BMVC*, pp. 421–430, 2001.
7. R. Cattoni, T. Coianiz, F. Fignoni, S. Messelodi, and C. Modena, "Geometric and logical layout analysis techniques for document image understanding: a review," Tech. Rep. 9703-09, ITC-irst, 1997.
8. P. Clark and M. Mirmehdi, "Location and recovery of text on oriented surfaces," in *Proc. SPIE: Document Recognition and Retrieval VII*, pp. 267–277, The International Society for Optical Engineering, January 2000.
9. S. Han, M. Lee, and G. Medioni, "Non-uniform skew estimation by tensor voting," in *Proc. Workshop on Document Image Analysis*, pp. 1–4, 1997.
10. D. Bloomberg, G. Kopec, and L. Dasari, "Measuring document image skew and orientation," in *Proc. SPIE: Document Recognition II*, pp. 302–316, 1995.
11. J. Shufelt, "Performance evaluation and analysis of vanishing point detection techniques," *IEEE Trans. Pattern Analysis and Machine Intelligence* **21**(3), pp. 282–288, 1999.
12. H. Baird, "The skew angle of printed documents," in *Proc. SPIE Symp. on Hybrid Imaging Systems*, pp. 21–24, 1987.
13. A. Bagdanov, "Projection profile based skew estimation algorithm for JBIG compressed images," in *Proc. 4th International Conference on Document Analysis and Recognition*, pp. 401–405, 1997.
14. W. Postl, "Detection of linear oblique structures and skew scan in digitized documents," in *Proc. 8th International Conference on Pattern Recognition*, pp. 687–689, 1986.

15. F. Jelinek, *Statistical Methods for Speech Recognition*, MIT Press, Cambridge, Massachusetts, 1997.
16. K. Popat, "Decoding of text lines in grayscale document images," in *Proc. International Conference on Acoustics, Speech, and Signal Processing (ICASSP 2001)*, (Salt Lake City, Utah), May 2001. To appear.
17. D. Dohono and I. Johnstone, "Asymptotic minimaxity of wavelet estimators with sampled data," Tech. Rep. 1997-4, Stanford, 1997.
18. K. Kise, A. Sato, and K. Matsumoto, "Document image segmentation as selection of Voronoi edges," in *Proc. Workshop on Document Image Analysis*, pp. 1–4, 1997.

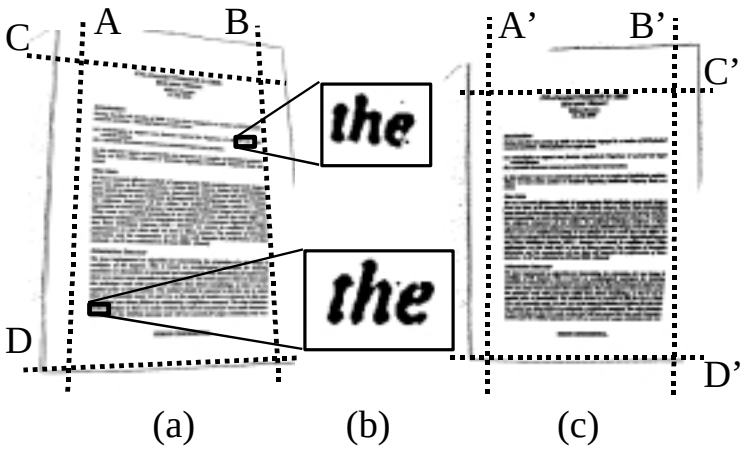

Figure 1. Effect and correction of perspective. (a) Observed document image. Lines A and B parallel to the column boundaries have differing orientations. Lines C and D parallel to the text lines have differing skews. (b) Subregions of the image are scaled and distorted under perspective. (c) Perspective correction makes A, B and C, D parallel.

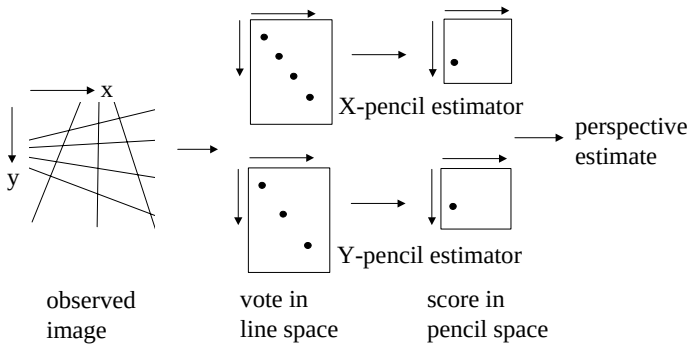

Figure 2. Overview of estimation method. The sum along a line λ through the observed image I is taken as the vote I_λ for that line. The vote accumulator is parameterised by the slope and intercept of lines. Only certain points (black dots) in this accumulator remain when a probabilistically motivated non-linear transform is applied. Sums are taken along lines Λ through this accumulator to compute scores in a second accumulator, which is parameterised by the slope and intercept of the lines Λ. The X- and Y-pencils with maximal scores (black dots) constitute the perspective estimate.

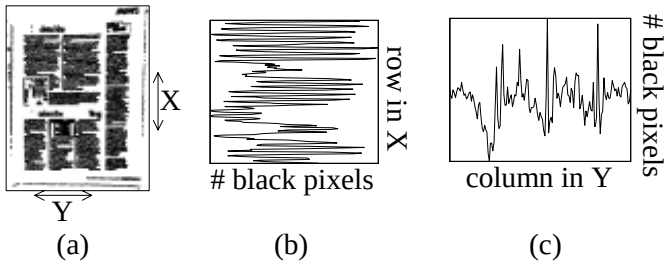

Figure 3. Signal-to-noise characteristics of projection profiles. (a) Observed image with no perspective or skew. (b) Horizontal profile of section X. (c) Vertical profile of section Y.

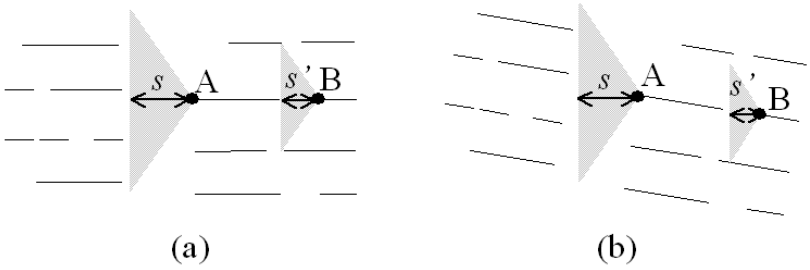

Figure 4. Selection of text edge points for (a) unrotated and (b) rotated image. In each case, the lines represent text words. Point A is detected at a large scale s in both cases since the triangle to its left is empty. Point B is only detected for small scales s' since the triangle to its left can be made no larger while remaining empty.

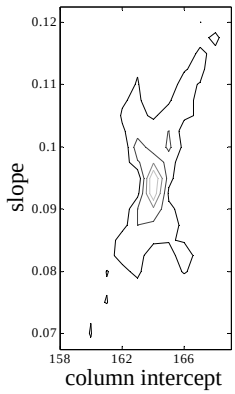

Figure 5. Contour map of votes in proximity to a column boundary. This is a plot of the number of selected points lying on every possible line from a small region of line space.

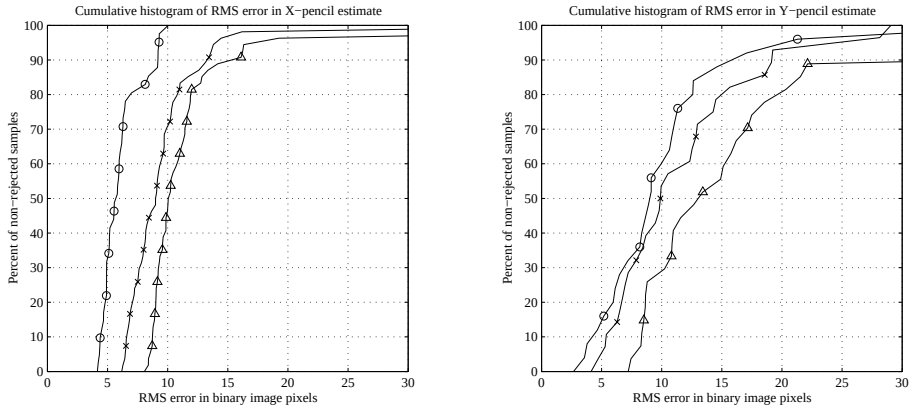

Figure 6. Cumulative histograms of the RMS error in X-pencil estimation (left) and Y-pencil estimation (right) as a function of RMS error in ground-truth. Each error is measured in pixels in the original binarised image. Circle: Low ground truth error (less than 4 pixels); Cross: Moderate ground truth error (up to 7 pixels); Triangle: High ground truth error (over 7 pixels).

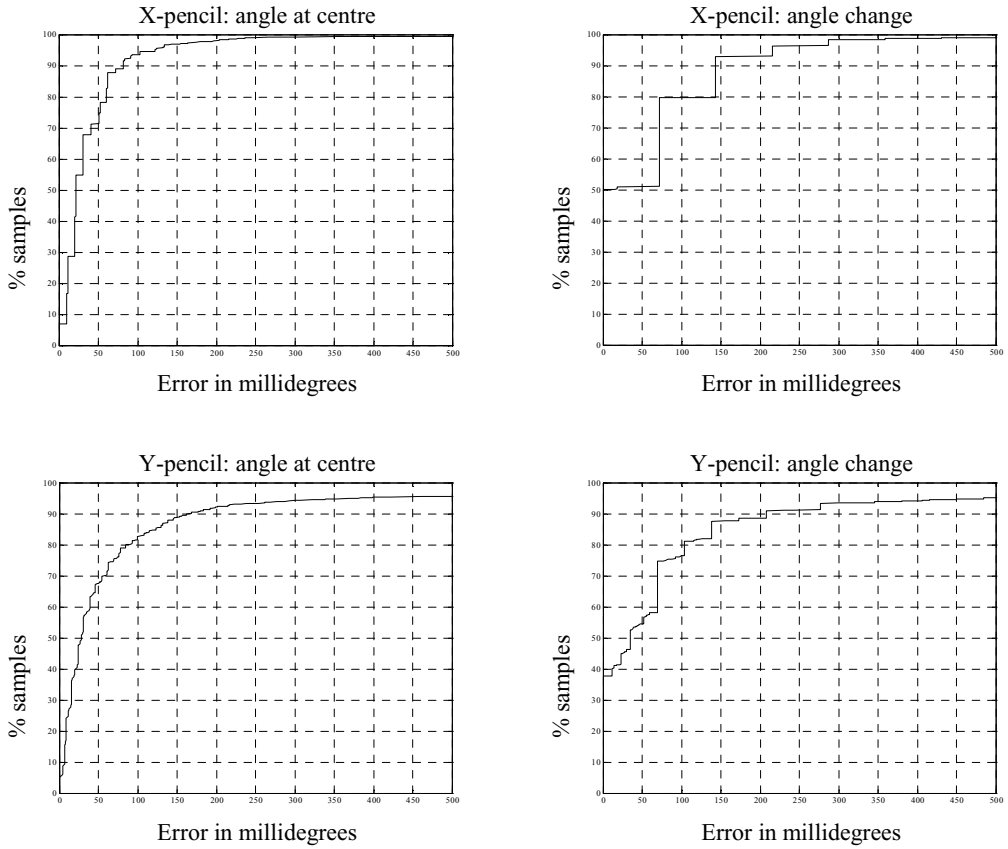

Figure 7. Cumulative histograms of the errors in pencil estimation for 168 images from the UW database, each rotated at 11 angles between $\pm 2^{\circ}$. The errors are for estimates of the angles of lines in the pencils through the centre of the image, and the changes in angle between these lines and lines from the top and left of the image.

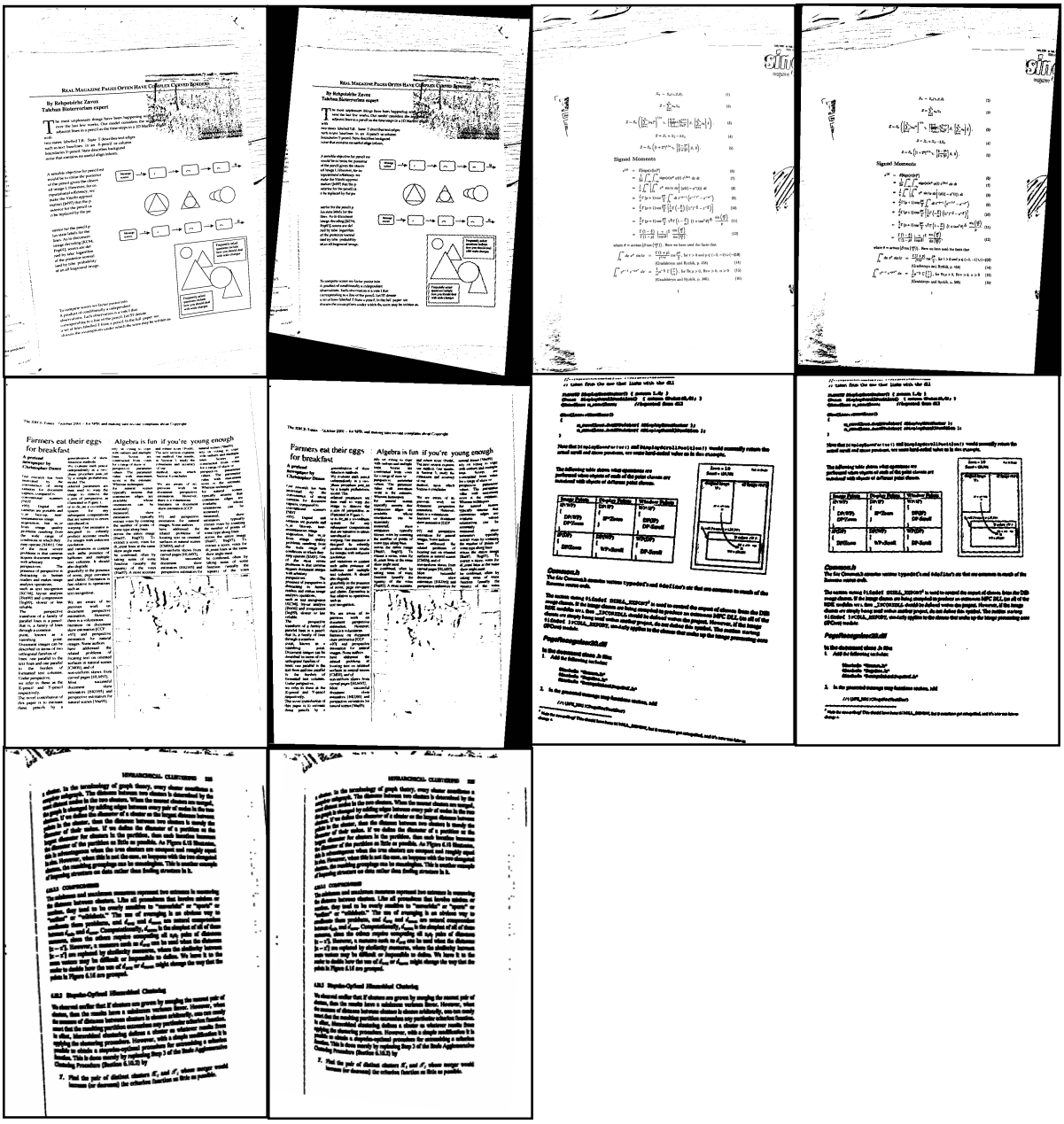

Figure 8. Illustrative examples of the application of our method to perspective correction. Each pair of images has the original input image on the left and the corrected image on the right.