PERGAMON

Pattern Recognition 36 (2003) 2673–2686

PATTERN RECOGNITION
THE JOURNAL OF THE PATTERN RECOGNITION SOCIETY

www.elsevier.com/locate/patcog

# Rectifying perspective views of text in 3D scenes using vanishing points

Paul Clark, Majid Mirmehdi*

*Department of Computer Science, University of Bristol, Bristol BS8 1UB, UK*

Received 1 October 2002; accepted 2 April 2003

**Abstract**

Documents may be captured at any orientation when viewed with a hand-held camera. Here, a method of recovering fronto-parallel views of perspectively skewed text documents in single images is presented, useful for 'point-and-click' scanning or when generally seeking regions of text in a scene. We introduce a novel extension to the commonly used 2D projection profiles in document recognition to locate the horizontal vanishing point of the text plane. Following further analysis, we segment the lines of text to determine the style of justification of the paragraphs. The change in line spacings exhibited due to perspective is then used to locate the document's vertical vanishing point. No knowledge of the camera focal length is assumed. Using the vanishing points, a fronto-parallel view is recovered which is then suitable for OCR or other high-level recognition. We provide results demonstrating the algorithm's performance on documents over a wide range of orientations.


*Keywords:* Document perspective recovery; Paragraph format; Vanishing point detection; Document analysis and recognition

## 1. Introduction

With the use of cheap digital cameras becoming common-place in the home, the office, and in personal digital assistants (PDAs), the demand for simple, instantaneous scanning of documents is rising. Pointing a camera at a document, clicking a button to capture the image and then using some software to interpret the text, as in OCR, has many advantages. For example, it is fast and removes the need for a flatbed scanner, and allows for non-contact point-and-click capture of text documents. However, difficulties arise if the view of the document is perspectively skewed. Then, some form of rectification is necessary to obtain a fronto-parallel view of the document plane to allow off-the-shelf OCR software the best chance of interpretation.

There has been little research into the recognition of text in real scenes in which the text is perspectively oriented relative to the camera. As well as for general OCR document recognition, processing and compensating for perspective skew of text has applications in assisting the disabled and/or visually impaired, vehicle navigation (road signs) and monitoring (number plates), wearable computing tasks requiring knowledge of local text, and general automated tasks requiring the ability to read where it is not possible to use a scanner. In recent years, we have presented different methods for locating, segmenting and recovery of text in real scenes [1–4] (Fig. 1). As a result, we developed an approach for *estimating* the rectification of the perspective view of a document [2,4]. In this work, the issue is addressed in more depth and we present a *robust* approach which is also independent of the focal length of the camera and document font size. Additionally, we analyse and show the limits and range of orientations of the text documents to which our proposed method can be applied.

There are various works on correction of text documents, when rotated or skewed in the view plane only, such as [5–8]. Most such methods correctly use 2D assumptions such as parallel lines in the view plane to determine the

* Corresponding author.
*E-mail address:* majid@cs.bris.ac.uk (M. Mirmehdi).

0031-3203/03/$30.00 
doi:10.1016/S0031-3203(03)00132-8

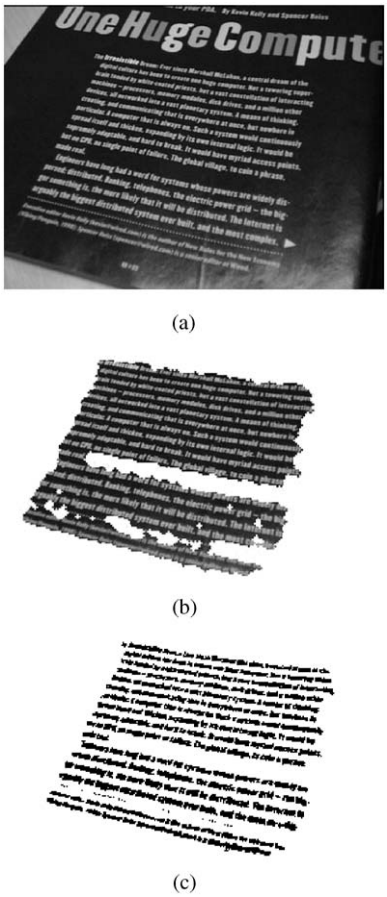


Fig. 1. Preparation of paragraph for planar recovery: (a) original image, (b) classified text region, (c) text region after adaptive thresholding.

parameters for rotational correction. However, other than our own previous work, the authors are aware of only two other groups involved in perspective skew recovery of documents [9,10]. In [9], a method is presented on extraction of linear illusory clues in skewed documents. Horizontal clues are extracted from a binarised input image where the characters, words (partial or full), and lines are transformed into blobs. Association networks are then built based on the relationships between neighbouring compact or elongated blobs. Using a pool of horizontal clues, a partial rectification of the document is performed. Vertical clues are then sought in this new intermediary image to help perform a full rectification. However, as explained in [9], the vertical rectification is dependent on the number of illusory clues obtained as well as on how reliable they are. In [10], sets of parallel lines corresponding to text lines and formatted column boundaries are grouped to estimate vanishing points which are then used for perspective correction. The author does not establish how the sets of lines are originally obtained. This approach provided only an estimation and was tested on orientation angles varying only 20° between the optical axis of the camera and the normal to the document. Furthermore, it only works on fully formatted (i.e. on both left and right side) paragraphs whereas the method proposed here will recognise different types of justifications in paragraphs of text and is applicable to a much wider range of orientations.

Outside of the document recognition area there are many works that estimate the orientation of planar surfaces in still images using repeating textures or specific object models based on some image features. For example, Ribeiro and Hancock [11] observed affine distortions in power spectra of an image to find lines of consistent texture indicating the vanishing points of a plane. Criminisi and Zisserman [12] also used texture to find the vanishing points of a plane, first by finding the orientation of the vanishing line with normalised auto-correlation, and then applying another similarity measure to find its position. Text too has repetitive elements (characters and lines) but these elements do not match each other, and also sometimes may cover only a small area of the image. Moreover, their spacing can be irregular.

Rother [13] found vanishing points corresponding to the three mutual orthogonal directions of a scene using the parallel lines commonly available in architectural environments. Murino and Foresti [14] used a 3D Hough transform to estimate the orientation of planar shapes with known rectilinear features. Gool et al. [15] employed invariants to recover scene geometry from image points with known planar homologies. All of these methods initially require selecting points in the image which are believed to have a certain relationship in the scene. In our work, we first detect probable paragraphs and lines of text in the image. We then attempt to minimise the error between the lines and a simple model constructed from a priori information about documents. The fitting equation itself detects the set of points which satisfy the constraints for a paragraph, and finds the parameters of the 3D model.

Knowledge of the principal vanishing points of the plane on which text lies is sufficient to recover a fronto-parallel view. The text lines of a paragraph on a plane oriented relative to the camera point towards the horizontal vanishing point of the text plane in the image. We make the reasonable assumption that a paragraph must display some sort of left, right, centred or full formatting, i.e. with straight margins on the left and/or right, or if the text is centred, a central vertical line around which the text is aligned. In such cases, these vertical lines point toward the vertical vanishing point of the text plane. We have therefore concentrated our work on the recovery of paragraphs using these principles to extract the horizontal and vertical vanishing points.

One approach to building a model of a paragraph is the bottom-up grouping of pixels into characters, words and lines. This can be a noisy and messy process. Here, we use all of the global information about the paragraph at one time. The principle of 2D projection profiles is extended to the

problem of locating the horizontal vanishing point by maximising the separation of the lines in the paragraph. From this vanishing point we are able to accurately segment paragraphs into separate lines of text. The segmented lines are then analysed to reveal the style of justification (or formatting or alignment) of the paragraph. Depending on the type of paragraph, either the margins or the projective spacings between the lines are used to find the vertical vanishing point. For fully formatted paragraphs the vertical vanishing point is found using the paragraph's straight margins. For all other paragraphs, i.e. left, right, or centrally formatted, line spacings are analysed to find the position of the vertical vanishing point accurately. The proposed method eliminates the need for knowledge of the focal length of the camera, hence the technique is applicable to images taken from cameras with unknown internal parameters.

In Section 2 we briefly review our previous work which provides the input to the work described here. Section 3 concentrates on locating the horizontal vanishing point. This information is applied in Section 4 to determine the formatting style of paragraphs. In Section 5 we demonstrate our method based on line spacings for precisely locating the vertical vanishing point. We evaluate the range of orientations our methods apply to for both vanishing points. Section 6 illustrates the process of rectification of a document into a fronto-parallel view and presents further results. We conclude the paper in Section 7.

## 2. Finding text regions

The source images input into the methods proposed in this paper can be either direct point-and-click images of text documents, e.g. from a PDA, or regions of text segmented in more cluttered scenes, e.g. using our earlier work in [4]. Hence, although our images come from different sources and are of various sizes, on average they are about $400 \times 300$ pixels in size. In the simpler point-and-click scenario, the whole image can be searched for clues and features knowing of the presence of a paragraph of text. To deal with cluttered scenes, we described two alternative algorithms in [4] to segment regions of text first, followed with a simple approach to their fronto-parallel recovery. In this work, we can take direct point-and-click images from a hand-held camera or the output of the segmentation system presented in [4] and analyse each recognised region individually to classify the shape of paragraphs, recover the 3D orientation of the text plane more robustly, and generate a fronto-parallel view of the text. However, the emphasis here is not on how the text region was originally obtained or segmented, but on what can be done next to deal with its accurate rectification.

In order to analyse the paragraph shape, we first require a classification of the text and background pixels to obtain a binary representation. Elaborate adaptive thresholding schemes could be used [16–18], however dealing with regions of text only, this classification is simplified since the region will contain easily separable background and foreground colours. The thresholding does not need to be too accurate and breakages in characters or words are not detrimental to later stages. Nevertheless, we prefer reasonably good results and compute the average intensity of the image neighbourhood as an *adaptive local threshold* for each pixel, in order to compensate for any variation in illumination across a text region. Partial sums [19] are employed to generate these local thresholds efficiently. To ensure the correct labelling of both dark-on-light and light-on-dark text, the proportion of pixels which fall above and below the adaptive thresholds is considered. Since in a block of text there is always a larger area of background than of text elements, the group of pixels with the lower proportion is labelled as text, and the other group as background. The example shown in Fig. 1 demonstrates the correct labelling of some light text on a dark background and is typical of the input into the work presented here.

## 3. Locating the horizontal vanishing point

In [8], Messelodi and Modena demonstrated a text location method on a database of images of book covers. They employed projection profiles to estimate the skew angle (in the view plane) of the text. A number of potential angles were found from pairs of components in the text, and a projection profile was generated for each angle. They observed that the projection profile with the minimum entropy corresponded to the correct skew angle. This is to be expected since the profile at the correct angle will have well-defined peaks and troughs corresponding to each line of text and the gaps between them. Projection profiles at other angles will cause lines to overlap, merging peaks and troughs, and increasing the entropy of the profile. This guided 1D search is not directly applicable to our problem where the orientation of the text is not parallel to the camera plane. Instead, the *vanishing point* in $\mathbb{R}^2$, with two degrees of freedom, must be found. In order to search this space, we will generate projection profiles from the point of view of vanishing points rather than from skew angles.

### 3.1. Search space

We need a circular search space $C$ as illustrated in Fig. 2(a). Each cell $c = (r, \theta)$, $r \in [0, 1)$ and $\theta \in [0, 2\pi)$, in the space $C$ corresponds to a hypothesised vanishing point $\mathbf{H} = (H_r, H_\theta)$ on the image plane $\mathbb{R}^2$, with scalar distance $H_r = r/(1-r)$ from the centre of the image, and angle $H_\theta = \theta$. This maps the infinite plane $\mathbb{R}^2$ exponentially into the finite search space $C$. A projection profile of the text is generated for every vanishing point in $C$, except those lying within the text region itself (the central hole in Fig. 2(b)).

A projection profile $B$ is a set of $N$ bins $\{B_i,\ i = 0, \ldots, N-1\}$ into which image pixels are accumulated. In the classical 2D case, to generate the projection profile of a binary image

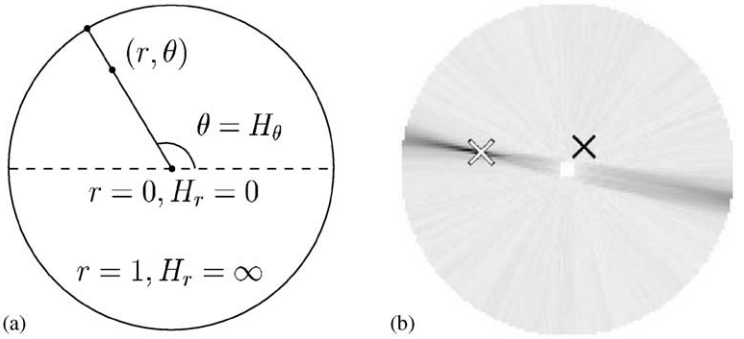



Fig. 2. Search space $C$. (a) Relationship between search space $C$ and $\mathbb{R}^2$. (b) Scores for all projection, profiles in $C$ generated from Fig. 1(c).

from a particular angle $\phi$, each positive pixel $\mathbf{p}$ is assigned to bin $B_i$, where $i$ is dependent on $\mathbf{p}$ and $\phi$ according to the following equation:

$$i(\mathbf{p}, \phi) = \frac{1}{2} N + N \frac{\mathbf{p} \cdot \mathbf{U}}{s}, \tag{1}$$

where $\mathbf{U} = (\sin\phi, \cos\phi)$ is a normal vector describing the angle of the projection profile, and $s$ is the diagonal distance of the image. In this equation, the dot product $\mathbf{p} \cdot \mathbf{U}$ is the position of the pixel along the axis of the projection profile in the image defined by $\phi$. The values of $N$ and $s$ in Eq. (1) then help map each pixel $\mathbf{p}$ to a bin index in the range 0 to $N - 1$.

For a perspectively skewed target, instead of an angle $\phi$, we have the point of projection $\mathbf{H}$ on the image plane, which has two degrees of freedom. The bins, rather than representing parallel slices of the image along a particular direction, must represent angular slices projecting from $\mathbf{H}$. Hence, we refine Eq. (1) to map from an image pixel $\mathbf{p}$ into a bin $B_i$ as follows:

$$i(\mathbf{p}, \mathbf{H}) = \frac{1}{2} N + N \frac{\angle(\mathbf{H}, \mathbf{H} - \mathbf{p})}{\Delta\theta}, \tag{2}$$

where $\angle(\mathbf{H}, \mathbf{H} - \mathbf{p})$ is the angle between pixel $\mathbf{p}$ and the centre of the image, relative to the vanishing point $\mathbf{H}$, and $\Delta\theta$ is the size of the angular range within which the text is contained, again relative to the vanishing point $\mathbf{H}$. $\Delta\theta$ is obtained from

$$\Delta\theta = \angle(\mathbf{T}_L, \mathbf{T}_R), \tag{3}$$

where $\mathbf{T}_L$ and $\mathbf{T}_R$ are the two points on the bounding circle whose tangents pass through $\mathbf{H}$ (shown in Fig. 3(a)). Unlike $s$ in Eq. (1), it can be seen that $\Delta\theta$ is dependent on the point of projection $\mathbf{H}$. In fact $\Delta\theta \to 0$ as $\mathbf{H}_r \to \infty$ since more distant vanishing points view the text region through a smaller angular range. The use of $\mathbf{T}_L$ and $\mathbf{T}_R$ to find $\Delta\theta$ ensures that the angular range over which the text region is being analysed is as closely focused on the text as possible, without allowing any of the text pixels to fall outside the range of the projection profile's bins. This is vital in order for the generated profiles to be comparable, and also beneficial computationally since no bins need to be generated for the angular range $2\pi - \Delta\theta$ which is absent of text.

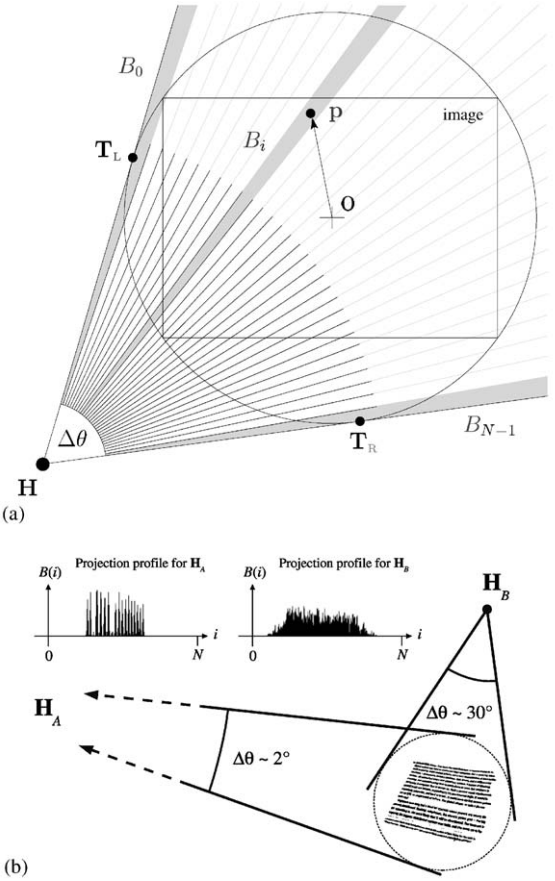



Fig. 3. Generating projection profiles from Fig. 1(c). (a) Generating the projection profile of the text from a vanishing point $\mathbf{H}$. (b) The winning projection profile from vanishing point $\mathbf{H}_A$ and a poor example from $\mathbf{H}_B$.

Having accumulated projection profiles for all the hypothesised vanishing points using Eq. (2), a simple measure of confidence is applied to each projection profile $B$ to mark its significance. The confidence measure must be chosen to respond favourably to projection profiles with distinct peaks and troughs. Since straight lines are most clearly distinguishable from the point where they intersect, this horizontal vanishing point and its neighbourhood should be favoured by the measure. We experimented with taking the entropy, squared-sum, and derivative-squared-sum of the projection profiles, where

$$\text{Entropy}: \quad E(B) = \sum_{i=1}^{N} B_i \log(B_i). \tag{4}$$

$$\text{Squared-sum}: \quad S(B) = \sum_{i=1}^{N} B_i^2. \tag{5}$$

$$\text{Derivative-squared-sum}: \nabla S(B) = \sum_{i=1}^{N-1} (B_{i+1} - B_i)^2. \tag{6}$$

The derivative measure $\nabla S(B)$ proved far more resilient to noise than the other measures, which were easily misled to view narrow paragraphs from the top or bottom, rather than the side. This made $\nabla S(B)$ the only viable measure for the hierarchical scan (see below) or noisy images.

To locate the vanishing point accurately, the resolution of the search space must be sufficient to hypothesise a large number of potential vanishing points. During experiments we found empirically that $10^4$ vanishing points was reasonable. Since each vanishing point examined requires the generation and analysis of a projection profile, a full search of the space, as shown in Fig. 2(b), is computationally expensive. However, due to the large scale features of the search space, we introduced an efficient hierarchical approach to the search. An initial scan of the search space at a low resolution is performed, requiring the generation of only a few hundred projection profiles. Adaptive thresholding is then applied to the confidence measures of these projection profiles, to extract the most interesting regions of the search space. (In our experiments, this was taken to be the top scoring 2% of the space.) A full resolution scan is then performed on these interesting regions in the search space, requiring the generation of a few further projection profiles close to the expected solution. Finally, the projection profile with the largest confidence is chosen as the horizontal vanishing point of the text plane. This hierarchical search reduced the processing time on a 440MHZ HP-PA Risc 8500 processor running HP-UX from over 40 s to about 3 s. In all our experiments, this adapted search found the same horizontal vanishing point as when we performed a full search. The winning projection profile and an example of a poor projection profile are shown in Fig. 3(b), and marked in Fig. 2(b) with a white cross and a black cross, respectively.

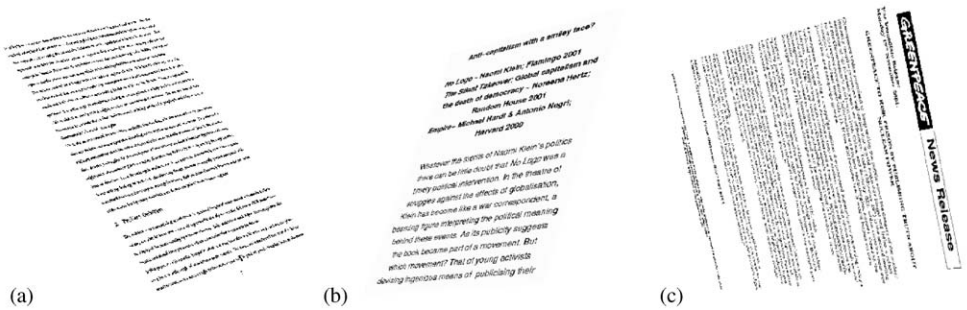
### 3.2. Assessing horizontal vanishing point accuracy

In order to assess the performance of the algorithm, simulated images such as those in Fig. 4 were generated at various orientations ranging from 0° to 90° in both yaw and pitch, resulting in 900 test images. Fig. 5(a) shows the accuracy of recovery of the horizontal vanishing point (HVP) for these images, calculated as the relative distance of the located vanishing point $\mathbf{H}$ from the ground truth $\mathbf{H}_{gt}$, given by

$$\textbf{HVP accuracy} = -\frac{|\mathbf{H} - \mathbf{H}_{gt}|}{|\mathbf{H}_{gt}|}. \tag{7}$$

As can be seen, the proposed method achieves excellent accuracy across a wide range of orientations. However, not unexpectedly, the performance begins to drop as the orientation of the plane approaches 90° in yaw or pitch. In these cases, the document has been rotated so as to be almost orthogonal to the view plane, and hence nearly invisible in the image, explaining the reduction in performance. The slope of the graph at low yaw may be attributed to the discretisation of the search space $C$. Since the vanishing points in these situations lie close to infinity, the distances of the located vanishing points cannot be precise. Nevertheless, the vanishing point chosen will be in the correct direction, and suitably large so as not to affect further processing.

To examine the influence of noise on the performance, we ran the same test repeatedly on images with increasing




(a) (b) (c)

Fig. 4. Examples of simulated images used for performance analysis: (a) fully-justified, (b) centrally-justified and various font sizes, (c) left-justified and extraneous elements.

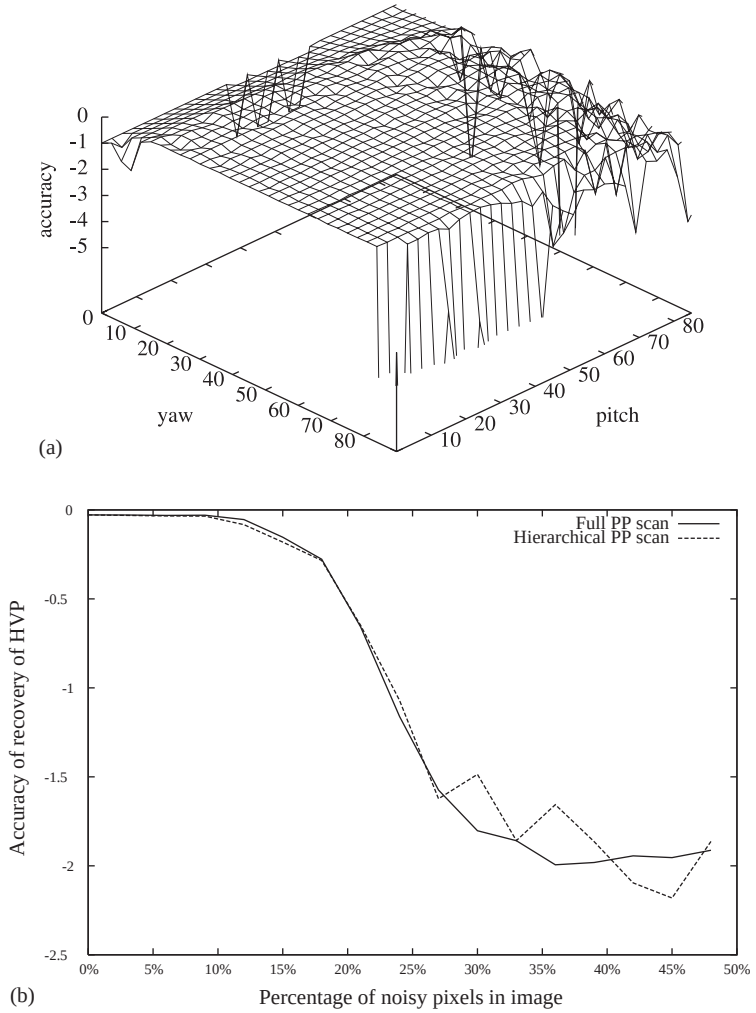



Fig. 5. Accuracy of recovery of the horizontal vanishing point. (a) Accuracy of recovery of a simulated paragraph at various orientations. (b) Accuracy averaged over all orientations plotted against increasing image noise.

proportions of speckled noise, and took the average of the results over all different orientations. Fig. 5(b) shows the average accuracy of recovery of the horizontal vanishing point against image noise. The results when using a full scan of the search space are also plotted, to demonstrate that there is only minimal loss of accuracy at high noise levels when using the hierarchical scan. A numerical analysis of the performance is given in Table 2 and discussed at the end of this paper.

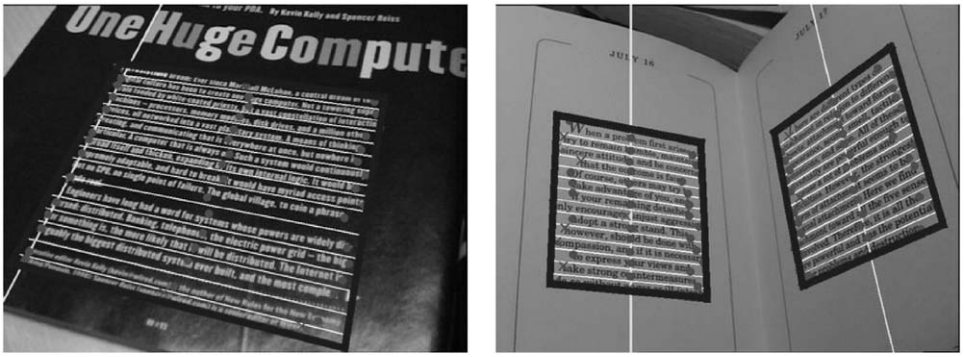
## 4. Determining the format of paragraphs

The location of the horizontal vanishing point and the projection profile of the text from that position make it possible to separate the individual lines of text. This will allow the style of formatting or justification of the paragraphs to be determined, and lead to the location of the vertical vanishing point as shall be described in Section 5.

We apply a simple algorithm to the winning projection profile to segment the lines. A *peak* is defined to be any range of angles over which all the projection profile's bins register more than $K$ pixels, with $K$ as the average height of the interesting part of the projection profile:

$$K = \frac{1}{y - x + 1} \sum_{i=x}^{y} B_i, \tag{8}$$

where $x$ and $y$ are the indices of the first and last non-empty bins, respectively. A *trough* is defined to be the range of angles between one peak and the next. The central angle of each trough is used to indicate the separating boundary of two adjacent lines in the paragraph. We project segmenting



(a) (b)

Fig. 6. Example of recognition: line segmentations are marked in white, spot points are used for margin fitting (green), crosses are rejected outliers (red), the baseline (yellow), and the final rectangular frame on the text plane (blue). (a) Typical example and (b) two centrally aligned paragraphs.

lines from the vanishing point through each of these central angles. All pixels in the binary image lying between two adjacent segmenting lines are collected together as one line of text. Example results of this segmentation are shown in Fig. 6. Noisy pixels, very short lines, and extraneous document elements may become attached to a true text line, or be segmented as a separate line. However, the processing which follows will compensate for this irrelevant data.

Depending on the formatting of the paragraph being recovered, there are now two possible ways to analyse the segmented lines to reveal the vertical vanishing point. If the paragraph is *fully-justified*, then the left and right margins of the text are straight, and intersecting these two margin lines will provide us with the vertical vanishing point, and the problem is fully resolved.

Alternatively, if the paragraph is *left-justified*, *right-justified*, or *centred*, a straight line will be visible either on the left margin, on the right margin, or through the centres of the lines, respectively. The vanishing point will lie somewhere along this *baseline* (see Fig. 6). However, its actual position will still be unknown, and must be estimated. Initially, the structure of the paragraph must be determined, i.e. its formatting style and its baseline. We collect the left end, the centroid, and the right end of each of the segmented paragraph lines, to form three sets of points $P_L, P_C, P_R$ respectively. Since some justification or formatting is anticipated in the text document, we will expect a straight line to fit well through at least one of these sets of points, representing the left or right margin, or the centre line of the text. To establish the line of best fit for each set of points, we use a RANSAC (random sampling consensus [20]) algorithm to reject outliers caused for example by short lines, equations or headings. Given a set of points $P$, the line of best fit through a potential fit $F = \{\mathbf{p}_i,\ i = 1,\ldots,M\} \subseteq P$ passes through $\mathbf{c}$, the average of the points, at an angle $\psi$ found by minimising the following error function:

$$E_F(\psi) = \frac{1}{M^5} \sum_{i=1}^{M} ((\mathbf{p}_i - \mathbf{c}) \cdot \mathbf{n}^2), \tag{9}$$

where $\mathbf{n} = (-\sin\psi,\ \cos\psi)$ is the normal to the line, $M^2$ normalises the sum, and a further $M^3$ rewards the fit for using a large number of points. Hence, for the three sets of points $P_L, P_C, P_R$ we obtain three lines of best fit $F_L, F_C, F_R$ with respective errors $E_L, E_C, E_R$. It is now possible to classify the formatting style of the paragraph using the rules in Table 1. Fig. 6(a) shows the line $F_L$ passing through the left margin of the paragraph. In this case $E_L < E_C$ and $E_L < E_R$, hence the second condition in Table 1 is satisfied and the paragraph is correctly identified as being left-justified. The examples in Fig. 6(b) show the detection of centrally-justified paragraphs in a book of quotations.

Table 1
Classifying the type of paragraph given the error of fit in Eq. (9)

| Condition | Type of paragraph |
|---|---|
| $E_L \simeq E_C \simeq E_R$ | Fully justified |
| $\min(E_L, E_C, E_R) = E_L$ | Left-justified |
| $\min(E_L, E_C, E_R) = E_R$ | Right-justified |
| $\min(E_L, E_C, E_R) = E_C$ | Centrally justified |

As mentioned earlier, for fully justified paragraphs, the recovery of the vertical vanishing point is trivial. This may be achieved by intersecting the left and right margins of the paragraphs, e.g. $F_L$ and $F_R$, the results of which are shown later in Table 2 and Fig. 9(a). However, for a left-justified, right-justified or centralised paragraph, we can retrieve only one baseline. The other two fitted lines will have significant errors due to the jagged margin(s). In these situations, a

Table 2
The average error over 10°–80° in yaw and pitch for the various methods described

| Method and paragraph (pgh) type HVP = horizontal VP, VVP = vertical VP | VP error | Angular error (deg) |
|---|---|---|
| HVP using projection profiles | 0.129 | 2.16 |
| VVP using margin intersection on fully justified pghs | 0.132 | 3.93 |
| VVP using margin intersection on centrally justified pghs | 0.655 | 17.35 |
| VVP using margin intersection on left-justified pghs | 0.589 | 19.12 |
| VVP using line spacings on fully justified pghs | 0.333 | 4.65 |
| VVP using line spacings on centrally justified pghs | 0.318 | 4.45 |
| VVP using line spacings on left-justified pghs | 0.383 | 4.66 |

different method must be used to determine the position on the baseline at which the vanishing point lies.

## 5. Locating the vertical vanishing point

In a perspective view of a document, the spacing between adjacent lines of text in the image will vary relative to their distance from the camera. This is the same effect that causes the railway sleepers beneath a train track to appear closer together as they approach the horizon. This change in spacing can be used to determine the angle at which the document is tilted, and hence the vertical vanishing point **V** of the text plane.

In [21], Schaffalitzky and Zisserman group repetitive elements in an image (such as railings in front of a building) and use them to detect vanishing points and lines. They fit a vanishing point model to a group of objects using maximum likelihood estimation, providing the image coordinates of all relevant pixels as data for the fitting. Our approach is similar, except that for this application we have already performed recognition of the paragraphs of text and may therefore extract more representative data for the fitting (in the form of line position and line spacing pairs). It is important to note that in [21], the objects used contain regularly spaced features in the real-world, whereas our text data are corrupted by irregular spacings between paragraph lines and by various font sizes.

### 5.1. Fitting using line spacing

Our overall method is as follows. After first transforming the image to simplify the problem, we extract observations of the altitude of each line in the image. A model of the geometry will allow us to express these observations as a function of the paragraph's orientation. By fitting the observations to the function, we will obtain the desired parameters of the orientation, and hence the position of the vanishing point. We show that a higher-order fitting function can assist in avoiding outliers. The whole process is now described in further detail.

An appropriate rotation of the image plane about the $z$-axis will position the baseline (found in the previous section—call it **B**) vertically (see Fig. 7). This rotation is equivalent to

$$\mathbf{B}_v = \begin{pmatrix} \cos\left(\frac{\pi}{2} - \theta\right) & -\sin\left(\frac{\pi}{2} - \theta\right) \\ \sin\left(\frac{\pi}{2} - \theta\right) & \cos\left(\frac{\pi}{2} - \theta\right) \end{pmatrix} \mathbf{B}, \tag{10}$$

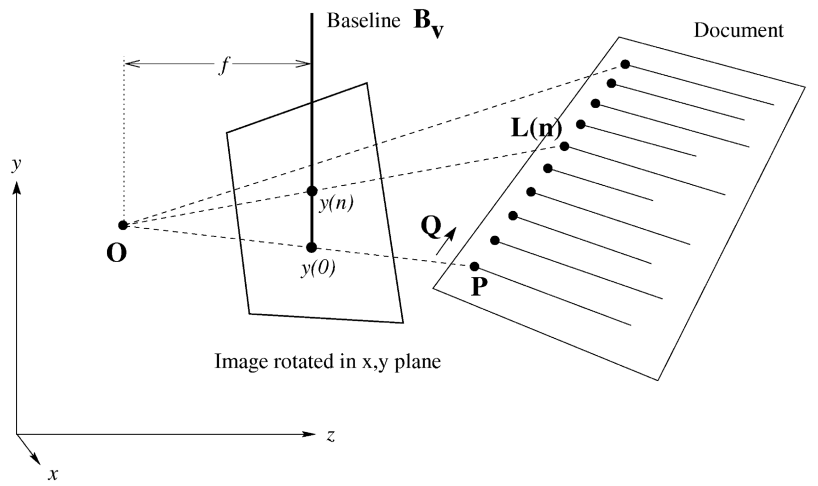



Fig. 7. Geometry involved in line spacings.

where $\theta$ is the angle between **B** and the horizontal. Henceforth, we can disregard the $x$-coordinates and deal solely in the $y,z$ plane. If we ignore spaces between paragraphs momentarily, we can describe an ideal geometrical model in which the bottom of a single paragraph is positioned at **P** with lines occurring at even spacings of distance **Q**. The $n$th line from the bottom of the paragraph will lie at

$$\mathbf{L}(n) = \mathbf{P} + n\mathbf{Q} \tag{11}$$

and will appear in the image at an observable altitude of $y(n)$. Using simple perspective projection ratios, we can relate the image points to the world position with

$$\frac{y(n)}{f} = \frac{\mathbf{L}(n)_y}{\mathbf{L}(n)_z}, \tag{12}$$

where $f$ is the focal length of the camera. Hence, the perspective projection of the $n$th line in the image plane, after substituting (11) into (12), is

$$y(n) = f\,\frac{\mathbf{P}_y + n\mathbf{Q}_y}{\mathbf{P}_z + n\mathbf{Q}_z} \tag{13}$$

with $y(\infty)$ giving the position of the vanishing line. Without losing the nature of the projection, we may scale the scene about the focal point in order to set $\mathbf{P}_z$ to $f$, hence modelling the paragraph as if it touched the image plane. In this case, $\mathbf{P}_y = y(0)$, and we may rewrite (13) as

$$y(n) = U\,\frac{1 + nV}{1 + nW} \tag{14}$$

with $U = y(0)$ and only two unknowns, $V = \mathbf{Q}_y/\mathbf{P}_y$ and $W = \mathbf{Q}_z/\mathbf{P}_z$. The cancelling of the focal length $f$ in this way means that the technique is applicable to images captured with any optical camera and the internal parameters of the original camera need not be known.

Eq. (14) now relates the parameters $\{V, W\}$, which describe the tilt of the document, to a set of points which may be extracted from the image. Observations for $y(n)$ are obtained by projecting the centroid of each detected line of text onto the baseline $\mathbf{B}_v$, to collect a set of points $O(n)$. A least-squares fit of $O(n)$ to the function $y(n)$ should now yield values for $V$ and $W$, and hence the vertical vanishing point.

However, this fitting is only applicable to the ideal document with equally spaced lines. In reality, it is common for documents to contain spaces between paragraphs, lines of text which are not part of a regularly spaced block, and unexpected extraneous elements. The presence of such outliers in the data can break the correspondence between the $n$th line in the ideal paragraph and the $n$th data point $O(n)$, which is implied in the fitting of $O(n)$ to $y(n)$. For example, in Fig. 8(a), data point (or line number) 9 is an undesirable outlier which has shifted all successive points one position along the horizontal $n$-axis, resulting in a poor fit. Due to its proximity, the point was not even rejected when RANSAC was used in the fitting.

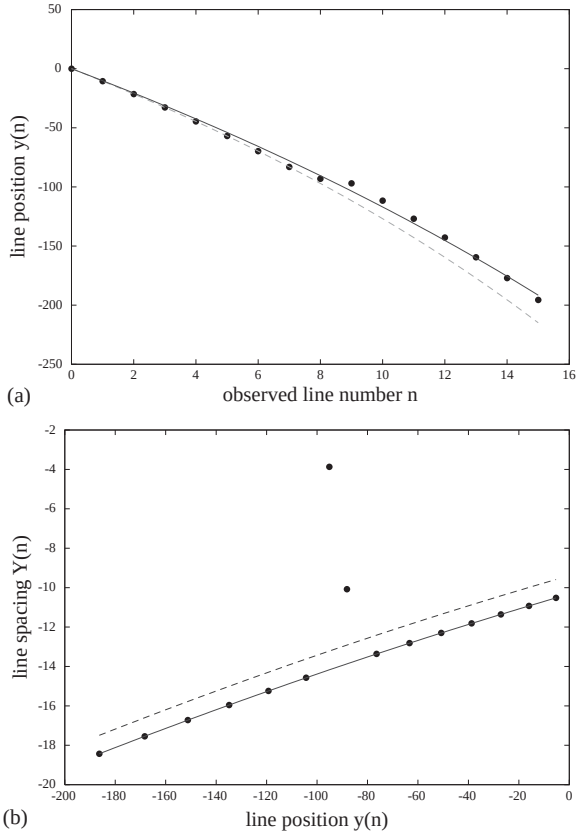



Fig. 8. Response of different fitting methods to outliers. (a) Line position gives a poor fitting due to the presence of an outlier. Ground truth is shown as a dashed line. (b) Using line spacing, the outlier does not affect the remaining points. Fitting before RANSAC is shown as a dashed line.

However, we can remove the direct influence of $n$, by instead fitting the curve of *line spacing* $Y_n$ against *position* $X_n$, defined by

$$Y_n = y(n+1) - y(n) \quad \text{line spacing,} \tag{15}$$

$$X_n = y(n) \quad \text{line position.} \tag{16}$$

With this approach, any irregularities in line spacing will appear as isolated outliers in the data, but will not propagate through the remaining points. By substituting Eq. (14) into the definition of line spacing (15), the curve of $Y$ in terms of $X$ may be written as

$$Y(X) = U\,\frac{1 + (n(X)+1)V}{1 + (n(X)+1)W} - U\,\frac{1 + n(X)V}{1 + n(X)W} \tag{17}$$

with $n$ in terms of $X$ derived by a similar substitution of Eq. (14) into the definition of line position (16) and

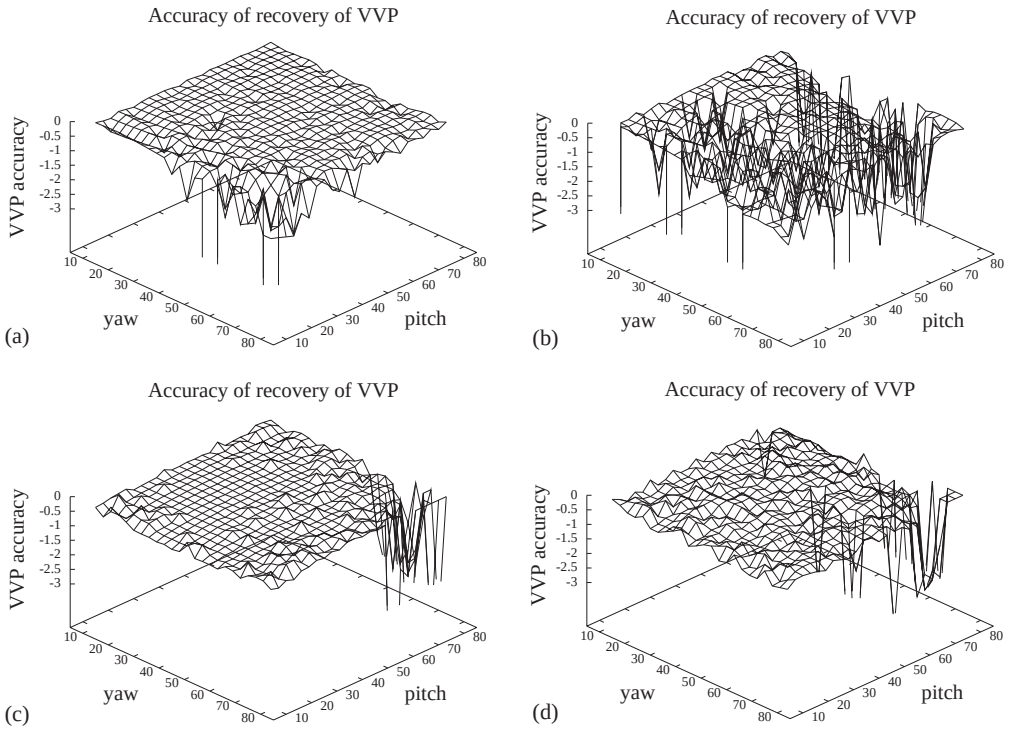



Fig. 9. Accuracy of recovery of vertical vanishing point (VVP) on simulated paragraphs at various orientations. (a) VVP recovery on fully justified paragraphs by intersecting margins. (b) Margin intersection fails on left-justified paragraphs. (c) VVP recovery on centrally justified paragraphs using line spacings. (d) VVP recovery on left-justified paragraphs by fitting line spacings.

rearrangement to

$$n(X) = \frac{X - U}{UV - XW}. \tag{18}$$

Initial values for the parameters $V = 2.0$ and $W = 0.1$ were empirically chosen as seeds for the fitting. To avoid the false local minima caused by the complexity of Eqs. (17) and (18), we ensure that parameters close to the global minimum are used as seeds, by making an initial fit with an approximation of Eq. (17):

$$Y(X) = \frac{UV}{1 + n(X)W}, \tag{19}$$

which refines the parameters before the final fitting with (17). The dashed line in Fig. 8(b) shows that the outlier from Fig. 8(a) has now affected two of the data points, but not the rest of the data. Since the line spacings exhibit more clearly the irregularities of the points observed in the image, RANSAC was able to easily reject these outliers, and produce a good fit (solid line in Fig. 8(b)). In fact, the fitting using line spacings is able to cope with situations when points are missing, presence of irrelevant points, or a space between paragraphs causes a change in the phase of the line positions.

Once optimised, $V$ and $W$ may be substituted into (14) to find the altitude of the horizon:

$$y(\infty) = \frac{UV}{W}. \tag{20}$$

After reversing the transformation made earlier in (10) to bring the baseline upright, this point will correspond to the location of the vertical vanishing point in the original image.

### 5.2. Assessing vertical vanishing point accuracy

Fig. 9 shows the accuracy of recovery of the vertical vanishing point for the whole range of 0–90° in yaw and pitch using the methods described. In Fig. 9(a) it can be seen that, as expected, intersecting the left and right margins $F_L$ and $F_R$ of a fully justified block of text gives a good estimate of the vertical vanishing point. Also, as expected and can be observed in Fig. 9(b), when such margins are used to estimate the vertical vanishing point of a non-fully formatted paragraph (in this example a left-justified one), performance is poor due to the paragraph's jagged edge. However, when line spacings are employed on, for example centrally-justified and left-justified paragraphs, as in Figs. 9(c) and (d) respectively, very good results with low error rates are obtained. This method provides good results for all of the simulated images except those documents oriented

beyond 80° in pitch, where the algorithm begins to fail. As with the horizontal vanishing point in Section 3, this may be explained by the orientation of the document becoming nearly perpendicular to the image plane. At such an extreme tilt, even if the lines of text are separated correctly, their proximity in the image means there is little accuracy in position and spacing for the curve fitting. In real world images, documents at such extreme angles cannot practically be read or used by OCR once recovered, hence this failure is not a great concern. The advantage of the line spacings method is that it provides consistent results for paragraphs which are not fully-justified.

The results for these experiments, and the location of the horizontal vanishing point in Section 3, can be compared more closely in Table 2. The vanishing point (VP) error is calculated as the relative distance of the vanishing point from its groundtruth, as described in Section 3 (see Eq. (7)). The angular error is the difference in angle between the determined orientation of the vertical vector of the text plane and the same vector from the groundtruth. It can be seen that the accuracy of location of the vertical vanishing point is good for both the margin intersection and the line spacings method. As rows 3 and 4 of Table 2 show, intersecting margins is not suitable for documents with jagged edges giving a large angular error. Results for right-justified paragraphs have been omitted for brevity, and are of course comparable to those for left-justified paragraphs.

Having found both of the vanishing points of the plane, we may project two lines from each to describe the left and right margins and the top and bottom limits of the paragraph(s) in the image. These lines are intersected to form a quadrilateral enclosing the text, as shown in Fig. 6. This quadrilateral is then used to recover a fronto-parallel viewpoint of the document.

## 6. Recovery of fronto-parallel view

In some applications we may not know the focal length of the camera used to capture the image. However, having a quadrilateral in the image which is known to map to a rectangle in the scene is sufficient to recover the focal length of the camera.

The vectors joining the focal point $\mathbf{O}$ to the two vanishing points $\mathbf{H}$ and $\mathbf{V}$ in the image plane are parallel to the horizontal and vertical vectors of the document. Since we expect these two vectors to be mutually perpendicular in the scene, their scalar product will be

$$(\mathbf{H}_x, \mathbf{H}_y, f) \cdot (\mathbf{V}_x, \mathbf{V}_y, f) = 0. \tag{21}$$

This constraint expands to

$$f = \sqrt{-\mathbf{H}_x\mathbf{V}_x - \mathbf{H}_y\mathbf{V}_y}. \tag{22}$$

It is worth noting that if $(\mathbf{H}_x, \mathbf{H}_y) \cdot (\mathbf{V}_x, \mathbf{V}_y) > 0$, then no solution exists for $f$. This situation means that the angle between the origin and the two vanishing points on the image plane is acute, and any corresponding quadrilateral cannot possibly be a rectangle in the scene. If such a quadrilateral is encountered during processing, we could hypothesise that the document is in fact slanted on the text plane, or that the quadrilateral does not actually correspond to a rectangular document in the scene, and should be ignored.

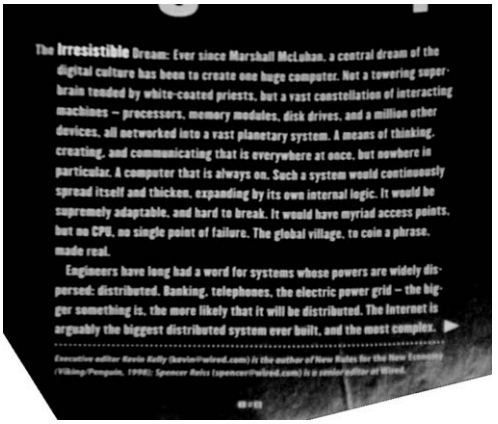
Having obtained the focal length of the camera, we may now recover a fronto-parallel view of the document. The mapping into the recovered image takes place in world space rather than image space. The grid of square pixels values in the original image project onto the document in the scene as quadrilaterals. However, bi-cubic interpolation between these points on the document plane would be overkill, unless data is being extracted for superresolution. A simple perspective mapping with interpolation in the image plane is more efficient, and will give rise to the same performance in the final stage of optical character recognition. Therefore, the pixel value of the document at position $X \in [0, 1],\ Y \in [0, 1]$ from the top-left is found by

$$doc(X, Y) = image\left(\frac{f\mathbf{S}_x}{\mathbf{S}_z}, \frac{f\mathbf{S}_y}{\mathbf{S}_z}\right), \tag{23}$$

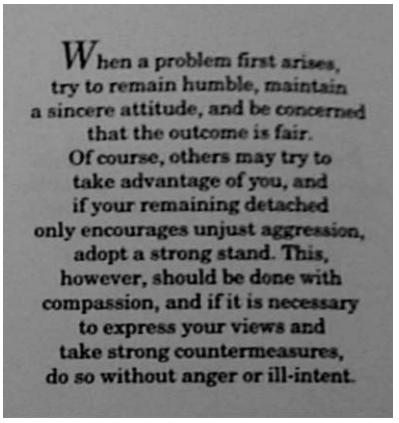
where $\mathbf{S} = \mathbf{A} + X(\mathbf{B} - \mathbf{A}) + Y(\mathbf{C} - \mathbf{A})$ is the corresponding scene point on the document plane derived from $\mathbf{A}, \mathbf{B}$ and $\mathbf{C}$, i.e. the top-left, top-right and bottom-left corners of the quadrilateral in the scene, respectively, and $image(x, y)$ is the interpolated pixel value at point $(x, y)$ in the image.

Fig. 10 shows the rectified images of the examples in Fig. 6. Further examples in Figs. 11 and 12 show the recovery of various paragraphs, sometimes multiple, sometimes single, and also multiple regions of text within a single image. For example, in Fig. 11(a) a centrally-justified document containing multiple paragraphs has been recovered at high resolution and is easily readable or Fig. 12(c) shows the recovery of a segmented region of a book cover which contains text of different sizes, and other image noise such as specularity.

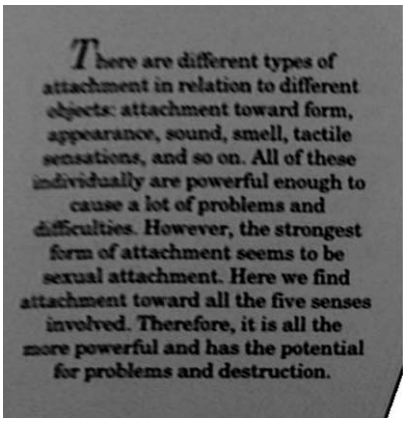
We put our rectified paragraphs through standard off-the-shelf OCR software, however, the final quality of character recognition is largely dependent on the quality of the camera resolution, the distance to the document, and the OCR software used. Since we make use of paragraph structure rather than individual printed characters, the proposed method is more dependent on the appearance of the paragraph in the image than on the font size of the text. Provided the lines of text are visible in the image (i.e. there is some spacing keeping the lines apart from each other), then the paragraph can be recovered. However, OCR will only be successful if the text in the original image is readable (i.e. characters are of a height within OCR-software limitations). Hence distant (small) paragraphs can only be recognised in images with a high resolution. This covers a wide range of situations, from close-ups of documents, to distant posters. We captured our images using a variety of cameras at various distances. In some recovered images,




(a)

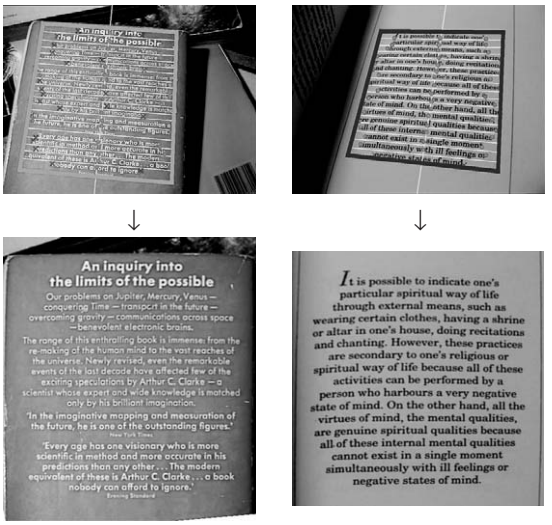



(b)




(c)

Fig. 10. Fronto-parallel recovery of example documents in Fig. 6: (a) recovery of earlier example; (b) recovery of paragraph from Fig. 6(b), suitable for OCR; (c) recovery of second paragraph from Fig. 6(b).




(a) (b) (c) (d)

Fig. 11. Further examples of fronto-parallel recovery of documents. In each case the original image is shown above the recovered output.

the OCR accuracy was below 50% while for others it was above 90%. For example, the image recovered in Fig. 11(b) achieved 100% accuracy. It would be easy to have a series of images captured under suitable conditions that result in close to 100% recognition rates. Here, we have refrained from that and concentrated on showing that irrespective of the final OCR result, and under general conditions of image capture, the proposed method can recover the text orientation from a wide range of perspective skews.

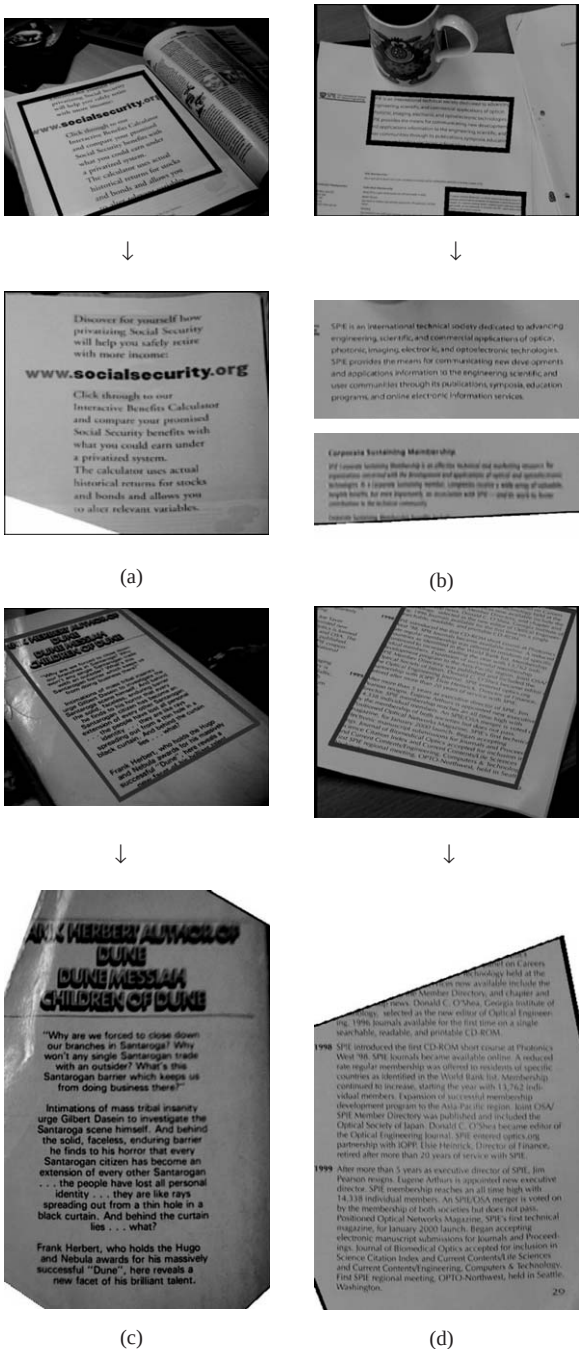



Fig. 12. Further examples of fronto-parallel recovery of documents. In each case the original image is shown above the recovered output.

## 7. Conclusion

We have presented a novel approach to the fronto-parallel recovery of a complete range of paragraph formats under perspective transformation in a single image, without knowledge of the camera focal length. Projection profiles from hypothesised vanishing points are used to recover the horizontal vanishing point of the text plane, and to segment the paragraph into its constituent lines. Line fitting on the margins and central line of the document is then applied to deduce the formatting style of the paragraphs. To estimate the vertical vanishing point, for fully justified paragraphs the margin lines are intersected. For other types of paragraphs, the observed difference in the spacings of the lines of text are used to retrieve the tilt of the text plane, and hence the vertical vanishing point. Using the two principal vanishing points we find the orientation of the document and recover a fronto-parallel view. The algorithm performs well for all types of paragraphs, provided there is some formatting. As a by-product of the proposed method we have demonstrated how to understand the format of paragraphs which is of many uses in the Document Recognition area. At present, the process takes around 20 s to recover a document, demonstrating its potential and applicability for normal scanning. We hope to report a much faster time in future through optimisation of the code and implementation on a faster processor.

In the future we intend to integrate the work described here and in [3] towards an automatic system for text recognition in the environment, suitable for a wearable computer system.

## References

[1] P. Clark, M. Mirmehdi, Combining statistical measures to find image text regions, in: Proceedings of the 15th International Conference on Pattern Recognition, IEEE Computer Society, Barcelona, Spain, 2000, pp. 450–453.

[2] P. Clark, M. Mirmehdi, Estimating the orientation and recovery of text planes in a single image, in: Proceedings of the 12th British Machine Vision Conference, Manchester, UK, 2001, pp. 421–430.

[3] M. Mirmehdi, P. Clark, J. Lam, Extracting low resolution text with an active camera for OCR, in: Proceedings of the Ninth Spanish Symposium on Pattern Recognition and Image Processing, Benicassim, Spain, 2001, pp. 43–48.

[4] P. Clark, M. Mirmehdi, Recognising text in real scenes, Int. J. Document Anal. Recognition 4 (4) (2002) 243–257.

[5] B. Yu, A. Jain, A robust and fast skew detection algorithm for generic documents, Pattern Recognition 29 (10) (1996) 1599–1629.

[6] A. Amin, S. Fischer, A. Parkinson, R. Shiu, Comparative-study of skew detection algorithms, J. Electron. Imaging 5 (4) (1996) 443–451.

[7] M.J. Taylor, A. Zappala, W.M. Newman, C.R. Dance, Documents through cameras, Image Vision Comput. 17 (1999) 831–844.

[8] S. Messelodi, C.M. Modena, Automatic identification and skew estimation of text lines in real scene images, Pattern Recognition 32 (5) (1999) 791–810.

[9] M. Pilu, Extraction of illusory linear clues in perspectively skewed documents, in: IEEE Conference on Computer Vision and Pattern Recognition, Hawaii, USA, 2001, pp. 363–368.

[10] C.R. Dance, Perspective estimation for document images, in: SPIE Conference on Document Recognition and Retrieval IX, San Jose, USA, Vol. 4670, 2002, pp. 244–254.
[11] E. Ribeiro, E. Hancock, Estimating the perspective pose of texture planes using spectral analysis on the unit sphere, Pattern Recognition 35 (10) (2002) 2141–2163.
[12] A. Criminisi, A. Zisserman, Shape from texture: homogeneity revisited, in: Proceedings of the 11th British Machine Vision Conference, Bristol, UK, 2000, pp. 82–91.
[13] C. Rother, A new approach for vanishing point detection in architectural environments, in: Proceedings of the 11th British Machine Vision Conference, Bristol, UK, 2000, pp. 382–391.
[14] V. Murino, G. Foresti, 2D into 3D Hough-space mapping for planar object pose estimation, Image Vision Comput. 15 (1997) 435–444.
[15] L.V. Gool, M. Proesmans, A. Zisserman, Planar homologies as a basis for grouping and recognition, Image Vision Comput. 18 (1998) 21–26.
[16] J. Kittler, J. Illingworth, Minimum error thresholding, Pattern Recognition 19 (1) (1986) 41–47.
[17] P. Sahoo, S. Soltani, A. Wong, Y. Chen, A survey of thresholding techniques, Comput. Vision, Graphics Image Process. 41 (2) (1988) 233–260.
[18] L. O'Gorman, Binarization and multithresholding of document images using connectivity, CVGIP: Graphical Models Image Process. 56 (6) (1994) 494–506.
[19] S. Hodges, R.J. Richards, Faster spatial image processing using partial summation, Technical Report TR.245, Cambridge University, 1996.
[20] R. Bolles, M. Fischler, A RANSAC-based approach to model fitting and its application to finding cylinders in range data, in: Proceedings of the Seventh International Conference on Artificial Intelligence, Vancouver, Canada, 1981, pp. 637–643.
[21] F. Schaffalitzky, A. Zisserman, Planar grouping for automatic detection of vanishing lines and points, Image Vision Comput. 18 (2000) 647–658.

**About the Author**— PAUL CLARK graduated from Bristol University in 1998 with a B.Sc. degree in Computer Science with Mathematics. His recent work since has been on the extraction of text from real world images. He is an advocate of GNU software and abstract programming languages.

**About the Author**—MAJID MIRMEHDI received the B.Sc. (Hons) and Ph.D. degrees in Computer Science in 1985 and 1991, respectively, from the City University, London. He has worked both in industry and in academia. He is currently a Senior Lecturer in the Department of Computer Science at Bristol University, UK. His research interests are in Image Processing, Computer Vision, Pattern Recognition, and Parallel Processing. He is a member of the IEEE, the IEE, and a member of the Executive Committee of the British Machine Vision Association as well as its Publicity Officer. He is also a member of the Technical Committee on Image Processing of the International Association of Science and Technology for Development (IASTED).