REAL-TIME EMBEDDED SKEW DETECTION AND FRAME REMOVAL

S. Banerjee[1], S. Noushath[2], P. Parikh[3*], S. Ramachandrula[1], A. Kuchibhotla[1], A. Sharma[1]*

[1]HP Labs, Bangalore, India; [2]College of Applied Sciences, Oman; [3]Kritikal Securescan, Noida, India

ABSTRACT

It is common to observe document skew and frame artifacts while photocopying and scanning documents. The motivation of this work is to embed skew correction and frame removal in the copy pipeline of a device to achieve 'one touch' cleanup. The two challenges that this poses are the need for: (a) substantially reducing computation and memory requirements and (b) minimizing the false positives. Peripheral document features, such as, page/content edges are low-complexity document skew predictors, and content-based approaches are of relatively higher complexity skew predictors. But state-of-the-art page edge detection methods fail on low-contrast document images, or for similar scanbed/document background. To minimize false positives required in embedded implementations, we propose: (1) a robust page edge detection algorithm that is a multiplicative combination of gradients and line based page edge detectors, (2) a robust skew detection algorithm that is a linear combination of page/content edge and content based predictors, and (3) a pipeline for skew correction and frame removal that uses these algorithms and has near-100% accuracy over a wide range of document images.

Index Terms— real-time embedded skew detection, embedded frame removal, page edge detection

1. INTRODUCTION

Document cleanup is the first step towards effective processing of paper, so that document images are more amenable for subsequent downstream operations. While scanning/photocopying, predominantly two types of artifacts, namely skew and frame get introduced. The skew arises in the document due to incorrect placement or due to shifting of the object when the scanner lid is being closed. The frame artifacts arise while scanning thick documents or due to the inadvertent opening of the scanner lid. For any ensuing document image processing these two artifacts have to be removed beforehand. In our approach we carry out robust page edge detection, followed by skew detection and frame removal. After skew detection the document is progressively rotated by the detected angle in the device [3]. As this rotation approach uses swaths of the image it can rotate the image without loading the whole image in memory, and saves about 80% memory as compared to approaches that need the full image in memory. Also, the output quality is equivalent to that of rotating the image with the traditional three shear rotation [4].The block diagram of the whole pipeline is as shown in Fig. 1, and a fixed point version has been embedded in an All-in-One (AiO) copier.

Prior work in document cleanup has focused on removing document skew and frames during scanning, including having this functionality in scanning software [1-2]. Our research addresses efficient cleanup that can be embedded in a copier. The advantage of the embedded implementation is that subsequent downstream operations can be pipelined and speeded up. However, the two challenges this poses are (a) the algorithms need to be memory and computation efficient and (b) there is no scope for interactivity thus needing to avoid false positives.

Past work in low complexity skew estimation have focused on feature reduction from the document images, so that the reduced feature points could predict the skew [1]. Peripheral document features, such as the page/content edges are one of the fast ways of estimating skew. However, accuracies of state-of-the-art page edge detection algorithms (Table 1) suffer for cases such as noisy low contrast images.

To improve accuracies for page edge detection, we pose it as a multiplicative ensemble (logarithmic opinion pool) of statistics, namely gradients, and line based detectors. We then pose the skew estimation problem as a linear combination of page/content edges based predictors and relatively higher complexity content based predictors. For a non-embedded implementation with our approach it is possible to find optimal combination parameters to guarantee highly accurate skew and frame detection. However, our work also considers computational tradeoffs that are important for embedded implementations.

Table 1: Past work in page edge detection

Method	Assumptions	Limitations
Statistical analysis [2,5]	Contrasting scanbed and doc. background	* They are similar * CCD sensor noise
Line fitting [6]	Good points detected	* Needs partial edge
Bed template [7]	Empty scanbed image	* Lighting variations
Doc. layout [8]	Manhattan layout	* Restricted types

[*] This work was carried out while the authors were at HP Labs, Bangalore, India.

2. ALGORITHM FORMULATION

2.1. Page edge detection as a multiplicative ensemble

Page edges are straight lines lying on the outermost periphery, where the scanbed and the document intersect. On this line would also be a gradient, however small. To minimize false positives, both gradients and line based predictors are equally important in determining the page edge and they both have to agree. Let f_1 and f_2 be the gradient and line based predictors, respectively. We thus define a multiplicative combined predictor, F, that is a Logarithmic Opinion Pool (LOP) of the predictors [9], as,

$$F = \frac{1}{Z}\exp(\alpha_1 \log f_1 + \alpha_2 \log f_2) \quad \ldots \quad (1)$$

where Z is the normalization factor. As both f_1 and f_2 have equal importance and are of low implementation complexity, we choose $\alpha_1 = \alpha_2 = 0.5$, instead of having different values.

On a gradient image, we find the outermost straight line using our approach of adaptive thresholding. Let $I(x,y)$ be the intensity image, and $G(x,y)$ be the gradient image determined by the Sobel operator. We find threshold, t, s.t.,

$$G(x_i, y_i) > t, \text{ for } i = 1,\ldots n,$$
$$\text{s.t. } (x_1, y_1)\ldots(x_n, y_n) \text{ form a line} \ldots (2)$$

Under noisy conditions, to determine if a set of points form a line, we first sub-divide the set of points $\{(x_1,y_1) \ldots (x_n, y_n)\}$ into disjoint subsets so that points in each of the subsets form a connected component. Say, $\{(x_{cl}, y_{cl}) \ldots (x_{cm}, y_{cm})\}$ is the c^{th} subset where $1 < l < m < n$. c^{th} subset is linear if the standard deviation of the angle subtended by each of the points and the mean is small. Mathematically,

$$\frac{1}{(m-l)}\sum_{k=l}^{m}(\alpha_k - \bar{\alpha})^2 < \varepsilon \text{ where } \varepsilon \text{ is small, } \bar{\alpha} \text{ is mean } \alpha_k,$$
$$\alpha_k \text{ is the angle between } (x_{ck}, y_{ck}) \text{ and } (\bar{x_c}, \bar{y_c}),$$
$$\text{and } (\bar{x_c}, \bar{y_c}) \text{ is the mean of the subset } c \ldots (3)$$

Between subsets, the c^{th} and $(c+1)^{th}$ subsets are collinear if

$$\frac{1}{(m_c - l_c)}\sum_{kc=l_c}^{m_c}(\alpha_{kc} - \overline{\alpha_{c+1}})^2 < \varepsilon \text{ and}$$
$$\frac{1}{(m_{c+1} - l_{c+1})}\sum_{k(c+1)=l_{c+1}}^{m_{c+1}}(\alpha_{k(c+1)} - \bar{\alpha_c})^2 < \varepsilon$$
$$\text{where } \varepsilon \text{ is small, } \bar{\alpha_c} \text{ is mean } \alpha_{kc}, \overline{\alpha_{c+1}} \text{ is mean } \alpha_{k(c+1)},$$
$$\alpha_{kc} \text{ is the angle between } (x_{ck}, y_{ck}) \text{ and } (\overline{x_{c+1}}, \overline{y_{c+1}}),$$
$$\alpha_{k(c+1)} \text{ is the angle between } (x_{(c+1)k}, y_{(c+1)k}) \text{ and } (\bar{x_c}, \bar{y_c}),$$
$$(\bar{x_c}, \bar{y_c}) \text{ is mean of } c \text{ and } (\overline{x_{c+1}}, \overline{y_{c+1}}) \text{ is mean of } (c+1) \ldots (4)$$

A plot of the linearity measure of the top margin points vs. each threshold, t, is as in Fig. 2(b). Under thresholding retains a lot of scanbed noise (Fig. 2(d)) and over thresholding loses the page edge (Fig. 2(e)). The scanbed noise also leads to long and connected lines. So, initially the linearity measure vs. threshold plot shows a minimum that corresponds to detecting the scanbed noise. It subsequently increases, and reaches a local minimum that corresponds to the page edge. To further reduce computational complexity we can also restrict ourselves to finding only the top page edge instead of finding page edges for all the four sides.

The advantage of the combined LOP predictor is its relationship with the Kullback-Leibler divergence [9]. Let P_{f1} and P_{f2} be the probability distribution of finding the page edge using gradients and linearity estimates, respectively, and P_e the target probability of finding the page edge. Then, the error between the combined predictor and the target is,

$$E(e,F) = \sum_{i=1}^{2}\alpha_i E(e, f_i) - \sum_{i=1}^{2}\alpha_i E(F, f_i) = \langle E(e, f_i)\rangle - A(f) \ldots (5)$$

where $\langle E(e,f_i)\rangle$ is the weighted ensemble mean and $A(f)$ is the non-zero ambiguity, as can be shown by Jensen's inequality. Thus, the error of the combined predictor is always less than or equal to the weighted ensemble mean of the error of the individual predictors. Also, the ambiguity in (5) is independent of the target probability and can be estimated using unlabeled data. For scanners as the noise distribution is known, the ambiguity can be estimated without any data. Hence, with the combined predictor we are able to achieve near-100% accurate page edge detection, for various scanbed noise/document background combinations.

2.2 Skew detection as a linear ensemble

As we need a robust embedded implementation that is very accurate and of low implementation complexity, we pose skew detection as a linear combination of two predictors from orthogonal document features, namely page/content edge based predictors, f_1, and content based predictors, f_2. So, our combined skew predictor, F,

$$F = \frac{1}{Z}\exp(\alpha_1 f_1 + \alpha_2 f_2 + \alpha_3) \quad \ldots \quad (6)$$

Peripheral document features, such as the detected page/content edges as described in the previous section are a fast way of estimating document skew. So, f_1 is of much lower implementation complexity than f_2. Thus instead of finding statistically optimal values of the parameters, α_1, α_2, and α_3, we give preference to f_1. If the skew prediction from f_1 is not deemed confident based on the confidence measure, we predict skew from f_2. If both f_1 and f_2 are not confident, the output remains unaltered. As our low implementation complexity algorithms have to be scalable over various scanners and copiers, we dynamically determine α_1, α_2, and α_3, based on confidence measures. Fig. 1 shows our pipeline.

We now describe skew estimation with f_1 (the Adapted Quasi Hough Transform (AQHT) method). The AQHT module traces for the first black (foreground) pixel from all four sides of the document, providing four sets of page/content edge boundary points. On each set of these points, we apply QHT [11] and accumulate angles into a global histogram. The angle bin with maximum count in the histogram gives the skew angle. As the number of points

arising from the page/content edges is much smaller compared to points arising from the document content, the implementation complexity of QHT is reduced significantly.

The confidence measure is built by checking consistency of the four angles reported from all the four sides of the document. The four generated angles are weighted by the length of the page/content edges of each side. We then group these angles into clusters. If all the four sides agree, we would have one cluster. However, if they do not we would have up to four clusters. The confidence measure is calculated based on the strength of the top cluster divided by the cumulative strength of the top two clusters.

The time and memory complexity of AQHT is O(max(*MN*, $(A_{max}-A_{min})/A_{res}$)) and *3MN*, respectively, where $M \times N$ is the image size, A_{max} and A_{min} are the maximum and minimum skew angles and A_{res} is the angle resolution.

Dey *et al.* [12] detail the computation-accuracy tradeoffs for f_2 in their Enhanced Piecewise Covering of Parallelograms (ePCP) method. The traditional PCP method iterates over all angles and determines the skew angle to be the angle where the white sections in the document are maximized. The ePCP method reduces computation by adopting a coarse-to-fine angle refinement strategy, so that the same skew angle is obtained with fewer computations. Based on experimental results (Table 2), we observed that f_2 is invoked approximately 5% of the time. The difference between the top two predicted angles indicates confidence.

2.3 Frame removal

The frame boundary is the line that separates the scanbed and the document boundary so that the scanbed noise is removed but the document content is intact. The page edge is a set of points, say, $\{(x_1, y_1) \ldots (x_n, y_n)\}$ and the skew angle determines the slope of the line *m*, assuming a $y = mx+c$ representation. The goal then is to estimate the desired line so that small sized contents, such as, page numbers are retained but the noise is eliminated. As smaller content would be of larger size than noise, we separate them using two dimensional connected component analysis. For the remaining points, we need to determine the farthest point (x_b, y_b), through which the line passes. So,

$$(x_b, y_b) = (x_i, y_i) \text{ s.t.} (x_i - (\frac{y_i - (y_1 - mx_1)}{m})) \text{ is maximum } \forall i \ldots (7)$$

3. EVALUATION AND COMPARISON

We tested our approach on a few hundred document images, having substantial variations in text, graphics, thickness, folds, and size. Results for a limited 40 of them are tabulated in Table 2. Of them 11 were text only documents, thereby suitable for content based skew detection. Our results show that both page edge detection (using our multiplicative predictor) and skew detection (using our linear combinative predictor) outperforms the individual predictors. For our evaluation, page edge detection was determined to be successful if the page edges were properly detected from all the four sides of the document, based on subjective evaluation. Skew correction results were compared with respect to manually generated ground truth for the test images. Skew correction and frame removal with our method was 97.5% accurate.

The PC run-time was 0-1s/image on a 2GHz Intel Core Duo Processor T2500 machine with 1GB of RAM, and the memory complexity is (O(*MN*)). On the embedded AiO, skew detection and frame removal took less than 1 second/image. Our approach was unsuccessful on one side of one of the documents, where a shadow interfered with the detection. State-of-the-art 'flood-fill' based frame removal [2] failed on 18 of those test images, and almost all failed on a competitor's scanner, when scanbed and document background were similar. Fig. 3 visually compares our approach to other available methods for page edge detection.

Table 2: Confusion matrix for page edge and skew detection.
Color code: True positive, False positive, False negative, True negative

Page edge detection results								
105	8	113	58	0	58	149	0	149
45	2	47	92	10	102	1	10	11
150	10	160	150	10	160	150	10	160
Gradient based			Line based			**Combined LOP**		
Skew detection results								
36	0	36	11	0	11	38	0	38
3	1	4	28	1	29	1	1	2
39	1	40	39	1	40	39	1	40
Page/content edge			Content based			**Our combination**		

4. REFERENCES

[1] J. J. Hull, "Document image skew detection: Survey and annotated bibliography," *Document Analysis System II*, vol. 29, pp. 40–64, 1998.

[2] G. Mattos, R. D. Lins, A. d. A. Fomiga and F. M. J. Martins, "Bigbatch: A document processing for clusters and grids," in *Proc. ACM Symposium on Applied Computing*, pp. 434–441, 2008.

[3] S. Banerjee, A. Kuchibhotla, "Real-time optimal-memory image rot. for embedded systems," *Proc. IEEE Int. Conf. on Image Proc.*, Nov. 2009.

[4] A. W. Paeth, "A fast algorithm for general raster rotation", *Proc. Graphics Interface*, pp. 77-81, May 1986.

[5] B. Gatos, T. Konidaris, K. Ntzios, I. Pratikakis and S. J. Perantonis, "A segmentation-free approach for keyword search in historical typewritten documents," in *IEEE Int. Conf. on Doc. Analysis and Recog.*, vol. 1, 2005.

[6] J. Fan, "Enhancement of camera-captured document images with watershed segmentation," in *Proc. Int. Workshop on Camera-Based Document Analysis and Recognition*, pp. 87–93, Sep 2007.

[7] J. T. Newell, "Auto-width detection using backing image," *No US66215999B1, Xerox, Corp.*, Sep 2003.

[8] G. Sheng and F. Leou, "Apparatus and method for detecting book document along the longitudinal edge," *No US6124950A, Minolta*, 2000.

[9] J. V. Hansen and A. Krogh, "A general method for combining predictors tested on protein secondary structure prediction", *Artificial Neural Networks in Medicine and Biology*, pp. 259-264, 2000.

[10] H. Jin and Y. Lu, "The optimal linear combination of multiple predictors under the generalized linear models", *Elsevier Statistics. And Probability Letters*, pp. 2321-2327, 2009.

[11] D. Li and S. Simske, "Shape retrieval based on distance ratio distribution", *Technical Report, HPL, No. 251*, Sep. 2002.

[12] P. Dey and S. Noushath, "e-PCP: A robust skew detection method for scanned document images", *Pattern Recognition*, vol. 43, no. 3, 2010.

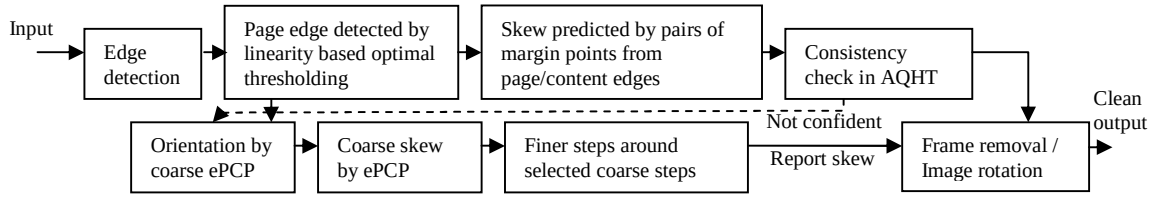

Fig. 1: Proposed skew correction and frame removal pipeline

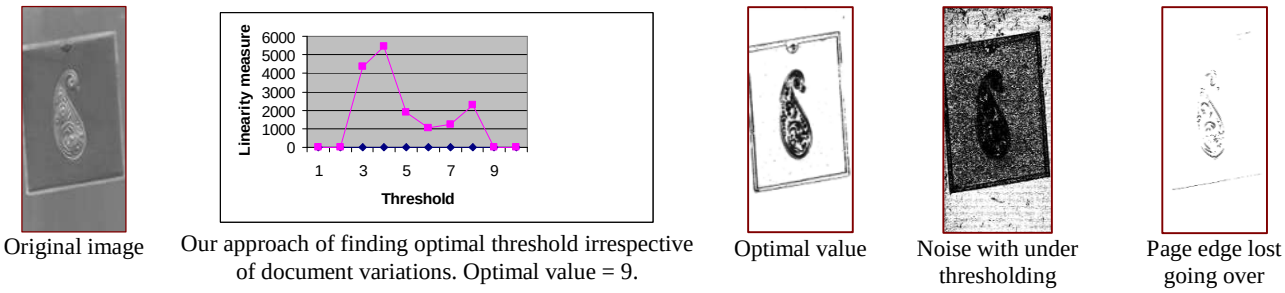

Fig. 2: Page edge can be detected in the presence of noise using the proposed linearity measure based adaptive thresholding of edge detected output and analysis for the top side of the document.

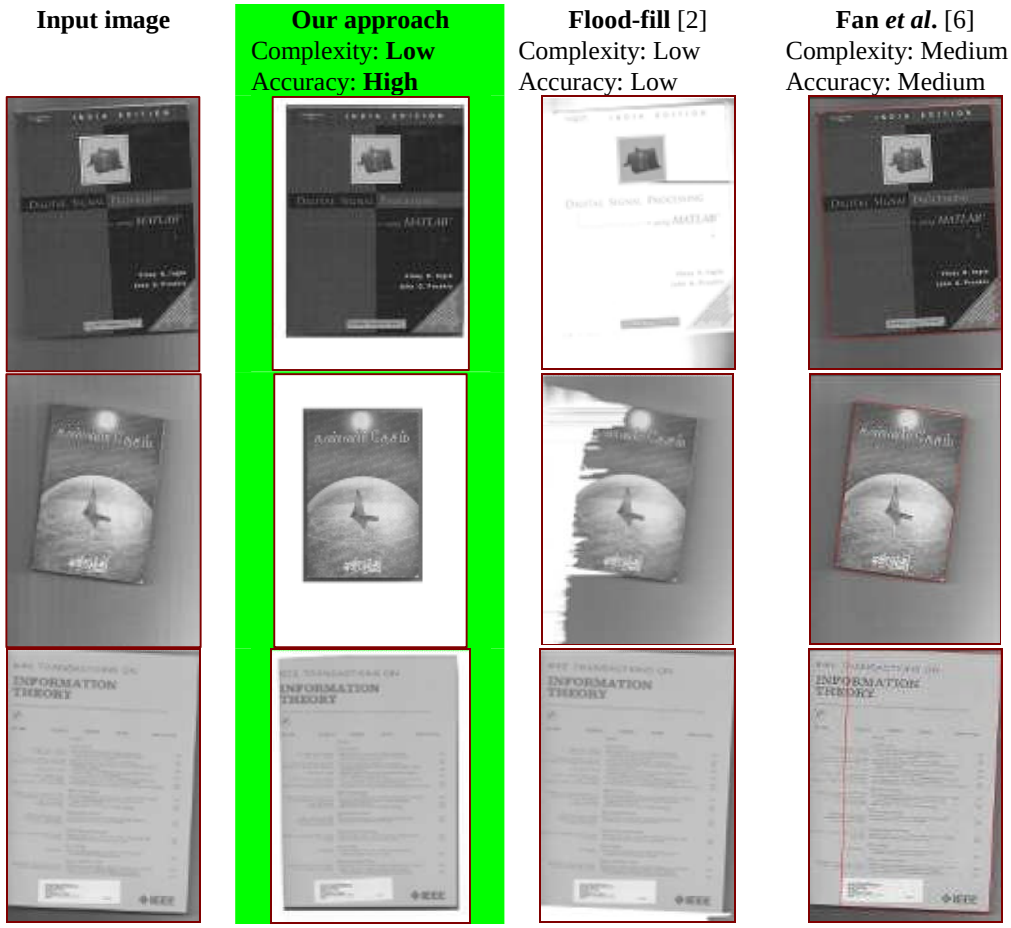

Fig. 3: Comparison of page edge detection outputs for different document types and backgrounds.