Preprocessing Challenges in Document Image Analysis

Keshao D. Kalaskar
Department of Computer Science,
Dr. Ambedkar College, Chandrapur-442 401(India)

Mahendra P. Dhore
Department of Computer Science,
SSESA's Science College, Nagpur (India)

ABSTRACT

Document Image Analysis (DIA) is the subfield of digital image processing that aims at converting document images to symbolic form for modification, storages, retrieval, reuse and transmission. It helps the transition from bookshelves and filing cabinets to the paperless and perhaps even wireless world. Preprocessing is the first stage in document image analysis. In Document Image Analysis, Preprocessing activity involves Representation, Noise reduction, Binarization, Skew estimation/detection, Zoning, Character segmentation. This paper focuses on the major challenges that are to be faced in preprocessing of document images for document image analysis.

Keywords

Document Image Analysis, Information Retrieval, Binarization, Skew detection, Character segmentation

1. INTRODUCTION

Document Image Analysis is the theory and practice of recovering the symbol structure of digital scanned from paper or produced by computer. It is essential to be able to display the results of processing in suitable form for human judgment. Accurate rendering of digitized picture at various scales requires some care. Images are classified in two categories according to their content.

Natural: Portraits, fingerprints, aerial photographs, satellite images and X-rays depict natural scenes or objects.

Symbolic: postal addresses, printed articles, bureaucratic forms, sheet music, engineering drawing and topographic maps.

TABLE 1. Schema for Document Image Analysis

LEVEL OF PROCESSING	Document Type	
	Mostly-text	**Mostly-graphics**
Pixels	**Preprocessing** Representation Noise reduction Binarization Skew detection Zoning Character segmentation Script, language & font recognition Character scaling	**Preprocessing** Representation Noise reduction Binarization Thinning Vectorization

2. PREPROCESSING

Preprocessing consists of a series of image-to-image transformations. It does not increase the knowledge of the contents of the documents, but may help to extract it. Here the identification of script, language and font which regard as metadata that assists information recovery. Some of the common operations performed prior to recognition are: thresholding, the task of converting a grey-scale image into a binary black-white image, noise removal, the extraction of the foreground textual matter by removing textured background, salt and pepper noise and interfering strokes; line segmentation, the separation of individual lines of text; word segmentation, the isolation of textual words, and character segmentation, the isolation of individual characters , typically those that are written discretely rather than cursively.

3. COMPRESSED REPRESENTATION

Until the advent of massive random access memories, there was considerable interest in character-level data compression method simply to avoid disk access during page analysis. Run-length coding and Freeman chain codes were used early on. Method that came along later include reduced terminal sequences of context-free grammars[1], coding on hexagonal meshes[2], produced rules for subblocks[3], and filtered contours[4]. For lossless, bilevel page compression, JBIG is gradually replacing CCITT-G3 and G4.

4. NOISE REDUCTION

Much research has been done on indexing and retrieval methods for text that is clean and free of errors, and substantial progress has been made in this area (Salton and McGill 1983, Salton 1989). Most Information retrieval systems represent any given text-an article (or a portion of an article), a query, etc. as a list of terms or keywords. In the past, experts on various topics were given the task of assigning an appropriate set of keywords to an article. This was called manual indexing. For even moderately large document databases, this approach is not feasible. Most contemporary Information retrieval systems, therefore, use automatic methods for indexing documents. The set of index term for a given document is usually obtained using the following general scheme outlined by Salton(1981).

- Individual words in the document are identified,
- Words occurring in a stop-list of common function words (and, of, or, for etc) are ignored, since they do not relate to the document's content,
- The remaining words are stemmed and reduced to their root form,
- Optionally, multi-word phrases are identified as additional indexing terms,
- Optionally, uncommon words are enhanced by a set of synonymous terms using a the-saurus,
- The resulting set of words stems and phrases are assigned to the document.

Given a user query (which is also a set of keywords), the documents containing these keywords are returned to the user. Keywords can also be combined using Boolean operators to construct more refined queries – thus, the user may retrieved documents to contain all of a set of keywords (AND), or the occurrence of only one out of several alternative terms may be sufficient (OR). Boolean methods are efficient and easy to implement, and yield good performance in certain situations

Indexing and retrieval of text that is free of errors a more realistic problem is the handing of text that is generated by an OCR device and contains errors. It is expected that the presence of recognition errors-misspelled words, garbled text, etc.- will result in the reduced performance if information retrieval systems. While some research efforts have focused on actually quantitatively measuring this effect, other groups have proposed ways to handle OCR errors during indexing and retrieval of noisy text.

5. BINARIZATION

Most early document scanners had hardware reflectance thresholds, but current scanners typically produce 8-bit gray-scale (or color) output. Researchers from the University of Oslo and Michigan State University conducted a sustained, thorough comparison and evaluation of published adaptive binarization methods (including their own) on hydrographic charts [5], [6], [7], [8]. Niblack's method, based on a threshold set below the mean gray-level of 15X15 window by a fixed fraction (0.2) of the standard deviation of the gray-levels, gave the best results on their maps. (A small modification is necessary when it is evident that the entire window is covered by a large foreground blot). They recommended post processing with the method of Yanowitz and Bruckstein, which iteratively creates a threshold surface that is essentially a low-pass-filtered version of the reflectance map. They also reported that character segmentation and recognition did not necessarily benefit from direct gray-scale processing as opposed to adaptive binarization [8]. Textured backgrounds are particularly difficult to handle. Liu and Srihari [9] provide a solution for postal address readers. It requires: 1) preliminary binarization based on a multimodal mixture distribution, 2) texture analysis with run-length histograms, and 3) selection of the threshold (using a small decision tree) that yields the stroke widths and lengths expected in a printed address.

In general, the appropriate binarization threshold is a sensitive function of *the* local reflectance map, but for high-contrast printed matter, it is difficult to improve on a fixed threshold centered between the extreme observed values. In adopting published binarization algorithms for applications, in which the gray-level distribution is not clear1y bimodal, it is important to take into consideration the amplitude transfer function of the specific scanner, as well as the spatial and gray-scale characteristics of the image.

Motivated by the observation that voting OCR labels obtained from multiple scans of the same page resulted in a substantial decrease in error rate, Sarkar et al. [10] examined the interplay between sampling and thresholding and counted the number of different bitmaps that can be produced for the same pattern by random displacement of the sampling grid. The resulting uncertainty limits the precision with which small patterns can be located [11], [12] but the effect is *less* significant with gray-scale scans. It also affects electronically-produced documents converted to bitmaps for display, and impedes character recognition in GIPF pictures.

6. SKEW DETECTION/ESTIMATION

Almost as many algorithms have been developed for skew detection as for binarization. All of them are accurate on full pages of uniformly aligned printed text. The better algorithms are less affected by the presence of graphics, paragraphs with different skew, curvilinear distortion arising from photocopying books, large areas of dark pixels near the margin, and few, short text lines.

A novel method is the Subspaced-Based Line Detection, based on an analogy between text-lines on a page and a linear antenna array emitting a planar propagating wavefront[1]. The distance from a reference line of each foreground pixel is converted into the phase of a complex sine wave. The detection algorithm, Based on radar signal processing, determines the spatial coherence between the contributions from different rows in the image.

One method was designed specifically for Indian scripts like Devanagari and Bangla [13] that have a head line *(shirarekha* or matra*)*. It was developed as part of a complete Bangla OCR system. The result of skew detection proved comparable to those from the Hough transform, but require less computation.

After skew detection, the page image is often rotated to a reference direction to facilitate further format analysis and OCR. On binarized pages the required resampling tends to distort the character patterns. Instead, it may be possible to modify the processing algorithms to take into account the skew [15]. Alternatively, either the document can be rotated before binarization, or the rotation can be approximated by small, distortion-free translations of entire word blocks.

7. CHARACTER SEGMENTATION

In 1996, Casey and Lecolinet [l6] surveyed the many approaches that have been proposed since 1959 to segment touching or fragmented character patterns. MIS- segmentation of characters is responsible for many OCR errors *(e.g.,* r n -> m or m -> r n). It *is* the consensus that *light* patterns are more difficult than heavy patterns, perhaps because of the greater import of missing and already scarce foreground pixels. The degree of difficulty depends on the typeface and print-source (smudged italics are difficult to segment) as well as on the ratio of font size to scanner resolution (point-spread function and spatial sampling rate).

Cascy and Lecolinet [16] defined dissection as the attempt to divide the image into classifiable units, whereas recognition-based segmentation either classified a multicharacter block at once, or segments the image according to features extracted from the entire block. Hybrid classification is a kind of soft segmentation, where the choice between multiple segmentation candidates is based on recognition. Here the winners among the jagged boundaries, imposed on the gray-scale patterns by pie-segmentation, are determined by the optimal path through a word-scale lattice (cf. [17]).

This comprehensive and scholarly survey concentrated on the underlying principles and did not attempt to evaluate the effectiveness of the various approaches. If there is a general conclusion, it is this: where dissection does not cut the muslard, he required gestalt techniques need to be so thoroughly integrated with recognition and context that character-level segmentation will soon disappear *as* a distinct area of research.

8. CHARACTER SCALING

In OCR, very small and very large word and character images are often scaled to a standard size, even though the outlines of characters of different sizes in the same typeface are not congruent. Resampling on gray-level arrays is relatively easy, but bilinear or bicubic interpolation distorts bilevel characters. The standard alternative is a two-stage process. The original smooth contour of the sampled character is first approximated using a weighted convolution filter and bilevel amplitude quantization. This stage is followed by resampling.

Scaling for OCR is not the purpose of the following methods, but they might find application in simulating or modeling OCR. Ulichriey and Troxel [19], writing at a time when hardware cycles were more scarce, developed "telescoping templates" for high-fidelity scaling with only logical operations. Namane and Sid-Ahmed [20] designed their algorithm for characters captured by a camera. After detecting the borders of the character, the contour is scaled, then interpolated (for magnification) with cubic splines. A 5x5 template is used to construct a bilevel image. Their results appear smoother than those obtained by replication or by telescoping templates. Also applicable here is the sophisticated contour construction of [4].

9. SCRIPT, LANGUAGE AND FONT RECOGNITION

The Script recognition reduces the number of different symbol classes that must be considered during classification. Language recognition is necessary for the use of appropriate context models. Font classification reduces the number of alternative shapes for each class, leading to essentially single font character recognition. Script language and font classification are also desirable for document indexing and interpretation.

Spitz[22], a pioneer in foreign language processing classified both script and language. He first differentiated between Latin and Han scripts according to the standard deviation of the vertical location of upward concavities with respect to the base line. In Latin print, these are mostly at the baseline or the x-line, whereas in complex Chinese, Japanese and Korean characters are uniformly distributed. The three oriental scripts are the recognized according to histograms of foreground pixel densities. The languages in Latin scripts are identified by the frequency of characteristics word shape, Tokens such as high-high-low for "the" in English, high-low for "le" and "la" in French, and high-low-low for "der" and "das" in German. Spitz uses run-length coding, bounding boxes and the pass codes of CCITT-G4 to speed up processing. Some European languages are recognized with only 90 percent accuracy.

Tan[23], classified 128x128 pixel samples into six script classes with 97 percent accuracy using rotation-invariant Gabor function coefficients. The presence of new typefaces affected, surprisingly, only the recognition of Chinese samples.

In practice, font recognition is likely to be faced with more classes than script or even language recognition. Fonts are classified according to typeface, weight, slope, width, and size. ApOFIS (A priori Optical Font Identification System) has a second-generation font model base for 280 fonts (10 typefaces, seven sizes, and four styles). Each model has statistics for six features estimated from 100 short text lines scanned at 300 dpi. Using this database and a Bayesian classifier, Zramdini and Ingold [21] classified fonts with 97 percent accuracy, and typeface, size, weight and slope with 97.5-99.9 percent. The accuracy increases rapidly with the size of the test sample, which may mean that short inserts of italics or boldface may be missed, and technical manuals with a variety of typefaces will be a challenge. Neverthless, this is a fine instance of computer classification that will outperform all but the most skilled typographer.

10. CONCLUSION

In this paper we focused on the major challenges that are to be faced in preprocessing of document images for document image analysis. Preprocessing being the first stage in document image analysis, involves Representation, Noise reduction, Binarization, Skew estimation/detection, Zoning, Character segmentation. All these preprocessing activities facilitate the proper document image analysis and understanding system. These preprocessing activities form the basis for analyzing the document images. The Script recognition reduces the number of different symbol classes that must be considered during classification. Language recognition is necessary for the use of appropriate context models. Font classification reduces the number of alternative shapes for each class, leading to essentially single font character recognition.

11. REFERENCES:

[1] E. T. Endo, "On a Methods of Bianry-Picture representation and its application to data compression," IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 2, no. 1 pp 27-35, January 1980.

[2] S. Yajima, J. L. Goodsell, T. Ichida, and H. Hirasishi, "Data Compression of Kanji Character Patterns Digitized on a Hexagonal Mesh", IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 3, no. 2 pp 121-229, February 1981.

[3] H. Nagahashi and M. Nakatsuyama, "A Pattern Description and Generation Method of Structural Characters", IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 8, no. 1 pp 112-117, January 1986.

[4] C. A. Cabrelli and U. M. Molter, "Automatic Representation of Binary image", IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 12, no. 12 pp 1190-1195, December 1990.

[5] T Taxt, PJ. Plynn, and A.K. Jain , "Segmentation of Document Images," IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 11, no. 12 pp 1322-1329, December 1989.

[6] O.D. Trier and T. Taxt, "Evaluation of Binarization Melhods for Document Images," IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 17, no. 3 pp 312-314, March 1995.

[7] O. D. Trier and A.K. Jain, "Goal-Directed Evaluation of Binarization Methods," IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 17, no. 12 pp 1191-1201, December 1995.

[8] O.D. Trier, T. Taxt, and G.K. Jain, "Recognition of Digits in Hydrographic Maps: Binary Versus Topographic Analysis," IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 19, no. 4 pp 399-404, April 1997.

[9] Y. Liu and S. Srihari, "Documcnt Image Binarization Based on Texture Features," IEEE Trans. Pattern

Analysis and Machine Intelligence, vol. 19, no. 5 pp 540-544, May 1997.

[10] P. Sarkar, G. Nagy, J. Zhou, and D. Lopresti, "Spatial Sampling of Printed Patterns," IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 20, no. 3 pp 344-350, March 1998.

[11] D.I. Havelock, "Geometric Precision in Noise-Free Digital Images," IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 11, no. 10 pp 1065-1075, Oct 1989.

[12] D.I. Havelock, " the Topology of locales and Its Effect on position Uncertainty," IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 13, no. 4 pp 380-385, April 1991.

[13] H.K. Aghajnn and T. Kailatli, "SLIDE: Subspace-Based Line Detection," IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 16, no. 11 pp 1057-1073, Nov 1994.

[14] B.B. Chaudhuri and U. Pal, "Skew Angle Detection of Digitized Script Documents" IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 19, no. 2 pp 182-186, Feb 1997.

[15] A.K. Jain and B. Yu "Document Representation and Its Application to Image Decomposition," IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 20, no. 3 pp 294-308, March 1998.

[16] R. G. Casey and E. Lccolinet, "A Survey of Methods and Strategies in Character Segmentation," IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 18, no. 7 pp 690-706, July 1996.

[17] J. Rocha and T. Pavlidis, "Character Recognition without Segmentation," IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 17, no. 9 pp 903-909, Sept 1995.

[18] Hoover et al., "An Experimental Comparison of Range Image Segmentation Algorithms" IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 18, no. 7 pp 673-689, July 1996.

[19] R. J. Ulichney and D.T. Troxel, "Scaling Binary Images with a Telescoping Template" IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 4, no. 3 pp 331-335, March 1982.

[20] Namane and M. A. Sid-Ahmad, " Character scaling by contour method," IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 12, no. 6 pp 600-606, June 1990.

[21] Zramdini and R. Ingold, "Optical Font Identification Using Typographic Features," IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 20, no. 8 pp 877-882, August 1998.

[22] A.L.Spitz, "Determination of the Script and Language Content of Document Images," IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 19, no. 3 pp 235-245, March 1997.

[23] T. N. Tan, "Rotation Invariant Texture Features and Their use in Automatic Script Identification," IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 20, no. 7 pp 751-756, July 1998.

[24] M. Cheriet and C.Y.SUEN, "Extraction of Key Letters Script Recognition," Pattern Recognition Letters, vol 14, pp. 1009-1017, 1993