IJDAR
DOI 10.1007/s10032-016-0273-3

ORIGINAL PAPER

Conservative preprocessing of document images

Jin Chen[1] · **Daniel Lopresti**[2] · **George Nagy**[3]

Received: 14 October 2015 / Revised: 16 August 2016 / Accepted: 27 August 2016

Abstract Many preprocessing techniques intended to normalize artifacts and clean noise induce anomalies in part due to the discretized nature of the document image and in part due to inherent ambiguity in the input image relative to the desired transformation. The potentially deleterious effects of common preprocessing methods are illustrated through a series of dramatic albeit contrived examples and then shown to affect real applications of ongoing interest to the community through three writer identification experiments conducted on Arabic handwriting. Retaining ruling lines detected by multi-line linear regression instead of repairing strokes broken by deleting ruling lines reduced the error rate by 4.5 %. Exploiting word position relative to detected rulings instead of ignoring it decreased errors by 5.5 %. Counteracting page skew by rotating extracted contours during feature extraction instead of rectifying the page image reduced the error by 1.4 %. All of these accuracy gains are shown to be statistically significant. Analogous methods are advocated for other document processing tasks as topics for future research.

✉ Jin Chen
jin.chen@nuance.com

Daniel Lopresti
lopresti@cse.lehigh.edu

George Nagy
nagy@ecse.rpi.edu

[1] Nuance Communications, 675 Massachusetts Avenue, Cambridge, MA 02139, USA

[2] CSE Department, Lehigh University, 19 Memorial Drive West, Bethlehem, PA 18015, USA

[3] ECSE Department, Rensselaer Polytechnic Institute, 6020 Johnsson Engineering Center, Troy, NY 12180, USA

Keywords Preprocessing · Document image analysis · Writer identification

1 Introduction

Preprocessing images aim to facilitate later processing stages by providing inputs that satisfy certain simplifying assumptions. According to Gonzalez and Woods [21], "There is no general agreement among authors regarding where image processing stops and other related areas, such as image analysis and computer vision start." For the purpose at hand, we define preprocessing as an image-to-image (pixelmap-to-pixelmap) transformation designed to satisfy given input specifications for some subsequent operation on the image. Any subsequent operation may result either in an image array or in some other data structure derived from the image. If the original image already satisfies the specification for the next stage, then the output of preprocessing should be the same as its input.

Conservative preprocessing preserves all information in the original image either by (1) using only perfectly invertible transformations or (2) describing rather than altering regions of interest. Examples of (1) include rotations by multiples of 90° and affine transformations where each row of pixels is shifted horizontally by an amount corresponding to the vertical coordinate of the row (assuming no cropping). An example of (2) is recording the boundary pixels of a suspected noise blob (a coffee stain or a camera flash) instead of changing pixel values to try to erase the blob.

The theory of linear systems implies that the geometric transformations of translation, rotation, and scaling by arbitrary amounts are asymptotically invertible as the size, dpi, and color depth of the output image approach infinity. However, the removal of occluding artifacts cannot be reversed

Published online: 20 September 2016

unless the original image is kept. In real applications, all of these transformations alter the image in some way.

In the context of document images, *preprocessing* is broadly synonymous with image restoration and image enhancement. In Marinai's words [31], "Preprocessing operations in document image analysis transform the input image into an enhanced image more suitable for further analysis." Popular preprocessing methods include binarization, rescaling, cropping, page rectification (including rotation of scanned images and more general geometric transformations for camera capture), thinning and skeletonization, slant removal from handwriting or printed italics, artifact removal (ruling and guide lines, stamps, scribbles), and conventional noise filters (salt and pepper, Gaussian, edge, blob). Marinai's taxonomy of preprocessing also includes boundary detection and thinning.

Current implementations result in irreversible transformations and information loss because image data preservation is not part of the downstream specifications, or because the system designer is not fully aware of the damage such transformations can introduce. In general, it is impossible to guarantee that an irreversible transformation will not result in a significant loss of information. Preservation of the original image data allows further analysis that exploits the recorded observations for improved overall results. For example, if an error in estimating the skew angle of a very sparse page results in too many characters rejected by downstream OCR on the rectified image, the program can re-try character recognition with the runner-up skew angle estimate. Alternatively, OCR features tuned to the skew angle can be extracted from the original image.

Document image analysis (DIA) has gradually separated from image processing research and practice on applications for phototagging, surveillance, remote sensing, and bioimaging that address images obtained from cameras, satellite sensors, digital X-rays, CT, MRI, and ultrasound. Document images are generally obtained from high-contrast originals like printed and handwritten text, equations and formulas, tables and forms, engineering drawings, and schematic diagrams. When these originals have a monotone foreground of symbols, they are often represented by bi-level bitmaps in spite of the inevitable random-phase spatial sampling effect [43]. Furthermore, in earlier days limited computing speed and storage precluded grayscale and full color processing. Consequently, many of the document processing methods reported in the literature start either with black-and-white scans or with software binarization. Although finer hardware or software quantization is preferable, most of our recommendations target bi-level image input because the distortions introduced by subsequent preprocessing steps are greatly aggravated by bi-level amplitude quantization. Conservative preprocessing is likely to benefit bitonal document images far more than grayscale, RGB, or multi-spectral digital images and videos where binarization is avoided.

Because of their dominant orthogonal or near orthogonal layout, skew considerations play a more important role in DIA than in most other applications. The objectives, results, and potential negative effects of various kinds of preprocessing are summarized in Table 1.

We show several contrived examples of the effects of irreversible preprocessing for illustrative purposes, because it is difficult to visualize what actually happens in a real application of interest. We also present experiments on real document images that compare conventional preprocessing with conservative preprocessing followed by feature extraction using information recorded during earlier stages in the DIA pipeline. Ensuring that a given set of features are invariant to irrelevant image characteristics complicates feature extraction compared to extracting them from ideal images. An alternative is to formulate features intrinsically invariant to image characteristics that are irrelevant to the desired output, like rotation-invariant moment features. Our experiments show some unexpected benefits from retaining image contents like ruling lines that in prior work were considered "noise," and from rotating features instead of rotating page bitmaps.

We are particularly interested in documents constructed in several stages, such as printed forms that may have handwriting, signatures, guide lines, and stamps added, possibly after photocopying or faxing, but before digitization by a scanner or camera. Degradation arising from multi-pass document construction is discussed in the context of bank checks in [46]. Most forms designed for collecting information are now "born digital" and are filled out online. However, automated information extraction from archival documents is attracting increasing attention as more and more corpora of historical interest are digitized and posted on the Web (often for genealogical applications). Non-textual components may not only assist text interpretation, but may benefit scholarly studies as well. An example of a collection offering interesting DIA problems is that of *Jia Pu* (Chinese family histories) digitized by FamilySearch [13].

After touching on relevant prior work in Sect. 2, we give some examples in Sect. 3 of egregious distortions introduced by irreversible image transformations. The remainder of the article reports experimental results that support the purported advantage of conservative image processing on a set of related DIA applications of ongoing interest to the community. Section 4 shows the loss of accuracy in writer identification induced by attempts to repair strokes broken by the removal of guidelines and demonstrates that not only is writing quality preserved by retaining guidelines, but the location of words with respect to adjacent guidelines can be exploited for more accurate writer identification. The important notion of normalizing features instead of normalizing

Table 1 Preprocessing in document image analysis

DIA stage	Assumptions regarding input	Preprocessing applied to satisfy assumptions	Potential negative impacts of preprocessing	Possible alternatives
Layout analysis	Image contains only specified document components, e.g., text and tables [1]	Artifact removal, e.g., logos, seals, photographs, coffee stains	Missed/spurious lines or words or other components of interest	Tag rather than remove artifacts
Line- and word-level segmentation	Upright orientation with no page skew [12]	Trigonometric rotation	Resampling error that affects line and word spacing	Segment lines and words according to estimated tilt
Connected-component analysis for feature extraction	Bi-level 4- or 8-connected foreground components [22]	Global or local binarization	Missed/spurious foreground components	Grayscale morphology and features [22]; selective re-binarization
Column/line segmentation	Vertical margins and horizontal lines; X–Y tree layout [27]	Rotation, affine, or perspective transformation for camera images with resampling	Wrong skew angle for near-empty pages; Distorted character shapes	Retain transform parameters; use continuous coordinates; non-isothetic partition
Word, character, typeface, writer recognition	Uniform spatial sampling rate (dpi) [2]	Isotropic scaling with resampling to compensate for variable input dpi	Missed/spurious strokes and rulings, touching/broken glyphs	Retain exact pixel coordinates; rescale grayscale image; SIFT [30]
Handwriting and typeset italics recognition	Slant-free writing or print	Global or local slant removal with resampling	Distorted glyphs, missed word boundaries	Retain local slant angle; use slant-invariant features [50]
Graphics recognition	Single-pixel wide lines [38]	Thinning or skeletonization	Anomalous spurs, broken/merged components	Use complete distance transform; vectorization
Any DIA task	No interaction between recto and verso	Bleed-through removal [42,53]	Alter/remove text information on both verso and recto	Tag information on verso and recto; compensate during feature extraction

images is illustrated in Sect. 5 using contour-hinge features as a case study. The concluding section evaluates the scope of applicability of conservative preprocessing and proposes further critical experiments.

2 Prior work

In digital image processing, image resampling is the process of geometric transformation of a discrete image from one coordinate system to another [16]. Preprocessing document images by resampling is widely recommended for size, rotation, or skew normalization [22]. An authoritative guide for practitioners [12] states that: "After the skew angle of the page has been detected, the page must be rotated in order to correct this skew." Resampling is usually accomplished by interpolating the pixel values in some neighborhood of the source coordinates of the transformed pixel and then rounding or truncating the result. Common interpolants include nearest neighbor, bilinear and bi-cubic ones. Thorough analyses of resampling are available in a general picture processing context [2,40], but not specifically for document image analysis.

Nonlinear shape normalization in handwritten text recognition [10,29,36,41,50,54], mapping pen coordinates for extracting features like turning angles [52], and document defect models [5] are all implemented via resampling. The error in computing features from the normalized bitmaps is reduced, but not eliminated, by higher spatial sampling rates and by scaling or rotating about the centroid of the image rather than the origin.

It is generally agreed that features invariant to scale, gray level, and skew are more reliable than image normalization. Many geometric features invariant to translation, scale, and rotation have been proposed [19,20,36], including scale invariant feature transform [30], and graph-based features that are affine invariant, Hu, Zernike, Krawtchouk, and Fourier–Mellin moments [23,25,49,51,55], functions of the number of stroke crossings along transects [15] and, by definition, all "topological" features [33].

Nevertheless, resampling remains a common approach to normalization because it simplifies subsequent extraction of features lacking an invariant formulation [26,48]. Examples of non-invariant features include template matching and N-

tuples [24]. Furthermore, most of the above features are not invariant to rubber-sheet transformations.

The recently published Handbook of Document Image Processing and Recognition [17] edited by Doermann and Tombre does not include a chapter titled "Preprocessing," but most of Chapter 4 by Gatos is devoted to binarization, enhancement, and geometric page normalization methods. Barney Smith's Chapter 2 reviews the characteristics of various methods of document generation, typical degradations from handling and storage of documents, and distortions due to digitization. In the last chapter, Margner and El Abed explain several metrics for binarization and segmentation and list relevant test datasets, ground-truthing tools, and contests.

Mohamad and his colleagues do not state the idea of avoiding preprocessing, but they do make use of slant information in feature extraction [32]. Some of the examples below were presented at the SPIE DR&R conference in 2013 [35]. The experiments on writer identification were part of the first author's 2015 doctoral dissertation at Lehigh University [9] under the guidance of the second author.

3 Examples of irreversible distortion caused by scanning and preprocessing

Scanning a page converts reflectance values to a rectangular array of intensity values (a *pixel* map) by spatial and amplitude quantization of the sensor response to a swath of light that sweeps across the page. The light source is usually a linear LED or laser array, while the sensor is either a charge-coupled device (CCD) or complementary metal-oxide-semiconductor (CMOS) array. The analog output of the sensor is sampled and quantized. The essential aspects of the optical performance of a scanner or camera are captured by its optical transfer function or spatial response function [7] and its spatial sampling frequency (resolution). Photodensitometers are slow precision instruments for accurate reflection measurement. Flatbed scanners provide one dimension of the sweep electronically and the other mechanically by moving the illumination/sensor assembly down the page. High-speed scanners move the paper relative to the stationary light source and optoelectronic transducer. Cameras have 2-D sensor arrays and natural or targeted external illumination. Specialized devices are available for digitizing microfilm, checks, envelopes, bound books, and large-format drawings. The faithfulness of page digitization, and the amount of information retained about digitization parameters and transducer characteristics greatly influence all subsequent image preprocessing, processing, and postprocessing.

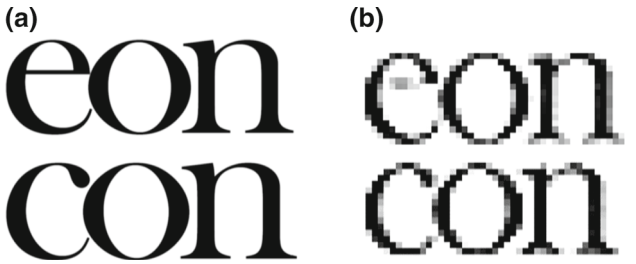

Fig. 1 "eon" and "con." **a** Original images. **b** Grayscale representation of scanned images. The broad range of gray-level values is due to the finite extent of the point spread function. Pixel values at edges are the weighted average of the foreground and background areas covered by the sampling spot [45]

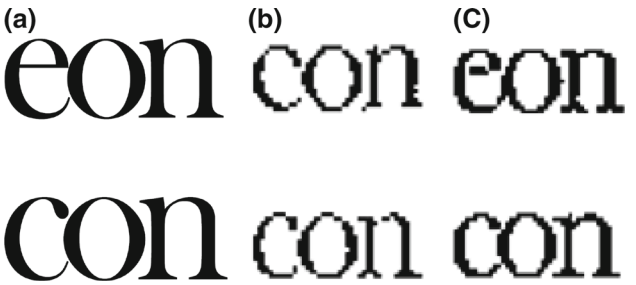

Fig. 2 Threshold effects. **a** Original patterns, highly magnified, on the same page. **b** The chosen binarization threshold preserves the gaps between e and o, and between c and o, but erases the bar of the e. **c** A lower threshold produces the opposite effect: it preserves the bar of the e, but connects e and o, and c and o

Spatial and amplitude quantization, especially binarization, is the most drastic type of irreversible image transformation. Some scanners map *optical reflectance* to gray value (typically 0-255), while others map *reflective density* (the logarithm of reflectance) [3,34]. The mapping can be modified by adjusting the scanner's brightness and contrast settings. These settings are seldom preserved in the output file. Even if calibration charts are scanned with each batch of documents (as they should be!), often the resulting information is eventually separated from the document stream. Then, there is no way of knowing the difference in reflectance between a gray value of 110 and a gray value of 150.

Amplitude quantization does not necessarily preserve contrast, as shown in Fig. 1. Furthermore, although the tails of the e and of the c appear to be the same in the original, they are different in the digitized version. This can happen even with a perfect ideal scanner because of the random placement of the glyphs with respect to the underlying sampling grid [43].

Otsu's binarization algorithm is a global thresholding technique [37]. Figure 2 shows an example where even a local binarization algorithm may be unable to find a threshold that preserves the horizontal bar of the e in *eon*, yet leaves a gap between c and o in *con*.

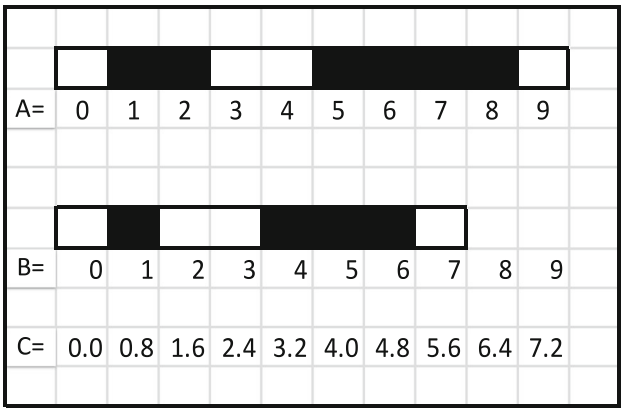

Fig. 3 Exact coordinate scaling versus resampling

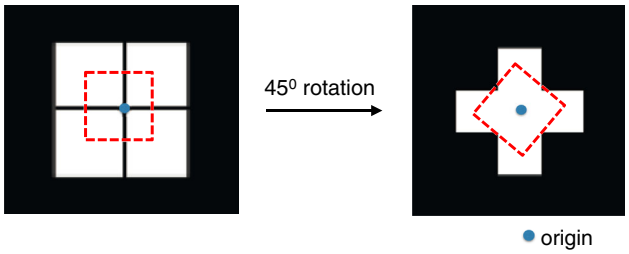

Fig. 4 Digital square and rotated digital square. Small patterns and large rotations are common in camera-based OCR. Resampling often changes the shape of commas and periods

Inaccuracies in feature extraction from resampled images are the intrinsic consequence of the irreversible mapping of real numbers into integers. The mapping can be avoided by retaining the floating point values of the transformed coordinates (ignoring rounding error). Normalizing the features can then take into account the size, skew, and gray scale of the original image without requiring the truncation of real numbers. Even though the difference between approximation and exact calculation may be small, feature normalization does not need experimental justification because it is a dominant alternative (i.e., it is always superior or equal) to resampling.

Feature normalization can be accomplished by computing the features from the exact values of the transformed coordinates. We illustrate this in Fig. 3 on a 1×10 pixel image. In this case, the second moment of an image is defined as following:

$$M_i = \sum_x x^i I(x) \tag{1}$$

where $I(x)$ is a 0-/1-valued amplitude function (1 for foreground) of the pixel at the coordinate $[x]$. Thus, M_2 of the original image A is computed as

$$\begin{aligned} M_2(A) &= 1^2 + 2^2 + 5^2 + 6^2 + 7^2 + 8^2 \\ &= 179 \end{aligned} \tag{2}$$

Likewise, M_2 of the resampled image B is $M_2(B) = 78$. The computation on exact coordinates scaled by 0.8:

$$\begin{aligned} M_2(C) &= 0.8^2 + 1.6^2 + 4.0^2 + 4.8^2 + 5.6^2 + 6.4^2 \\ &= 114.56 \end{aligned} \tag{3}$$

Which is correct? The continuous coordinate transformation yields the moment ratio of $\sqrt{(114.56/179)} = 0.80$ as expected, while resampling results in $\sqrt{(78/179)} = 0.66$. Therefore, any invariant moment computation based on the resampled image will be grossly in error!

Computing moments from resampled bitmaps that attempt to correct for skew (rotation or shear) is also error prone. A worst-case example is the 45° rotation of a 2×2 digital square (Fig. 4). Regardless of how the rotation is executed, there is simply no way to represent a small diamond on an isothetic grid. Some less striking effects of resampling on real documents are examined in Sect. 5.

The geometric moments can, however, be computed exactly in a continuous system of coordinates. Direct methods have also been derived for geometric moments under rotation (analogous to the one shown for 1-D scaling) and other linear transformations, but not for perspective or rubber-sheet transformations [39,44].

Resampling bitonal images can change connectivity of the image components. Thin line segments (e.g., rulings) in the original image may be broken, or adjacent parallel segments may be connected. As shown in Fig. 4, in the original image, each of the four foreground pixels is four-connected to two others, while in the rotated image the four contour pixels are four-connected only to the center pixel. Since connected-component algorithms depend on specified foreground and background connectivity (either four-connected foreground and eight-connected background, or vice versa), the topology of either could change.

All normalized features based on geometric distance (convex areas, perimeter lengths, stroke widths, moments, gradients, distance transforms, Hessians, directional derivatives, and projections) can be computed exactly using either invariant formulations or transformed coordinates instead of resampled bitmaps. The transformed coordinates of all the pixels of an $m \times n$ bitonal image can be stored as an $m \times n \times 2$ array of floating point x- and y-coordinates.

It is misleading to argue computing efficiency to avoid conservative methods. For example, MATLAB takes less than 0.1 s to compute 2000×3000 exact x and y rotation coordinates on a 2.5 GHz laptop. In contrast, rotating this

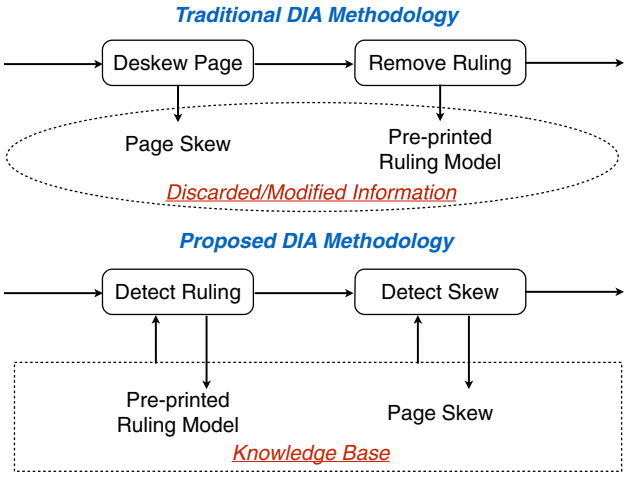

Fig. 5 A comparison of a typical DIA pipeline and our proposal. *Arrows* represent information flow between pipeline modules

bi-level image by resampling with nearest-neighbor interpolation takes MATLAB 1.7 s.

4 Conservative approaches to page artifacts

Conservative preprocessing preserves data in the image that could be valuable to later processing steps such as feature extraction.

This paradigm is depicted in Fig. 5. As described in the DIA literature, traditional preprocessing techniques attempt to clean up images by deskewing pages, removing ruling lines, eliminating scanning noise, etc. In these irreversible procedures, the bitmap is modified without keeping a record of the original or the processing that has taken place. Thus, information is lost that could be valuable to follow-up modules in the DIA pipeline.

In contrast, the proposed paradigm maintains the integrity of the input image by detecting and storing preprinted information, user added data, and digitization characteristics in a knowledge base accessible to later DIA stages. The knowledge base is a collection of extracted document attributes, structures, and artifacts, as well as any information derived from them, such as histograms of channel or grayscale intensity distributions, preprinted ruling lines and their attributes, tabular structures. No information from the original image is ever discarded, thus reserving the possibility of making use of it throughout the pipeline.

Page artifacts can be distracting during document primitive detection or feature extraction. We turn now to ruling lines as a specific case to illustrate our points. Preprinted ruling lines are designed to help people write neatly, but handwriting will often overlap the underlying ruling line. It is therefore customary to attempt to remove ruling lines before performing handwriting recognition or writer identification.

Arvind et al. [4] introduced a rule-based method that detects the ruling lines within segmented handwritten blocks by computing horizontal projection profiles. Abd-Almageed et al. [1] proposed a ruling line removal algorithm based on modeling rulings in linear subspaces. Kumar and Doermann [28] developed a fast ruling line removal algorithm that takes advantage of integral images to compute line features and uses a re-sampling scheme to reduce the sample size for training an SVM. Cao et al. [8] relied on local shape analysis to reconstruct handwritten strokes broken by removing ruling lines.

4.1 Negative effects of ruling line removal on writer identification

To demonstrate information loss, we implemented the traditional approach of detecting and removing ruling lines and then repairing possibly broken handwritten strokes. First, horizontal projection profiles (HPPs) are computed and the page skew is determined by finding the minimal entropy in the HPPs. Next, positions of the ruling lines are detected by finding the peak in the corresponding HPP. The bin with top vote in the histogram of vertical run lengths is selected as the line. Then, foreground pixels corresponding to this line are removed.

The next step is to recover handwritten strokes broken by ruling line removal. Following the strategy from Cao et al. [8], there are three sub-steps: broken stroke reconnection, thinned stroke recovery, and "U-shape" pattern detection and stroke regeneration. Broken strokes are recognized by computing the distances between segments above the ruling line and those below. Thinned strokes are preserved by inserting ink pixels column by column in the direction of the ruling line. "U-shape" segments are connected by drawing a straight line between two segments and partial ellipses at the ends to make the artificial stroke look natural.

Although care is taken to restore writing segments that cross ruling lines, this is at best an imperfect process (Fig. 6). Strokes parallel to and overlapping ruling lines, which are particularly common in Arabic handwriting, are vulnerable to the damaging effects of ruling line removal [8].

4.1.1 Experimental setup

The Arabic dataset we used for evaluation was provided by the Linguistic Data Consortium (LDC) [47]. Sixty native Arabic writers contributed samples of handwriting on paper sheets. To avoid sampling bias, we split each writer's handwritten lines into four disjoint subsets for cross-validation. Each fold in turn served as a test set with the remaining three subsets used as the training set. The results

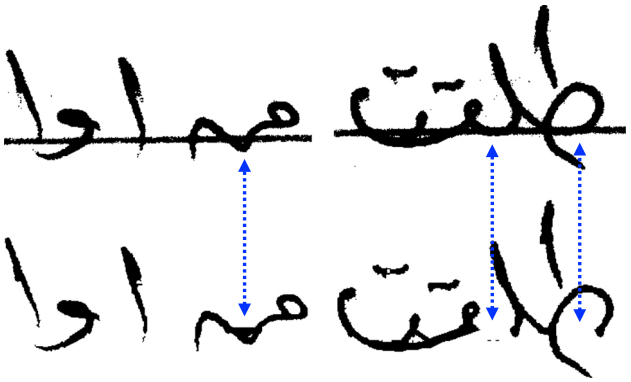

Fig. 6 Examples of potentially harmful stroke modifications introduced by ruling line removal

Table 2 Datasets used in our experiments

Dataset	Sample size (text lines)		
	Training	Testing	Total
Ruling-line-only (RLO)	2700	900	3600
Ruling-line-free (RLF)	20,700	6900	27,600
Mixed (M)	3600	1200	4800

are averaged across all fourfold. The sizes of the three datasets, as determined by the presence or absence of ruling lines on the sheets used by the writers, are shown in Table 2.

We use contour-hinge features, which have been shown to be useful for writer identification [6]. An illustration of this feature extraction procedure is shown in Fig. 7.

The feature computation is based on contours extracted using connected-component analysis. Specifically, we compute the angles ϕ_1 and ϕ_2 from the horizontal axis of each pair of adjacent segments (each five pixels long) along the contours (Fig. 7a). We treat pairs of angles as jointly distributed random variables. Quantizing the angle plane ($[0, 2\pi)$) into $n = 24$ bins, we accumulate the counts in each bin as we traverse every contour. Because of the assumed symmetry in the histogram, only half of the bins ($\phi_2 \geq \phi_1$) are used to compute a probability distribution function (PDF). Thus, the final feature vector is 300-dimensional $\left(\binom{24}{2} + 24 = 300\right)$.

We classified the ruling-line-only (RLO), ruling-line-free (RLF), and mixed (M) datasets using support vector machines (SVMs). Given that each page was scribed by one person, the writer identification accuracies reported in this paper were computed at the page level based on majority voting across the text lines on the page in question.

4.1.2 Experimental results

Table 3 shows that removing ruling lines actually decreases the accuracy of writer identification. The significance of the differences was validated by McNemar's test as discussed in "Appendix" section. The RLF result shows that the ruling line detection does not generate false positives. These observations motivated us to investigate whether there might be a better approach for handling such artifacts.

Table 3 Writer identification accuracy after applying a ruling line removal algorithm

	Training/testing subsets (%)		
	RLO	RLF	M
Before removal	62.5	74.7	62.0
After removal	58.0	74.7	61.0

The figures below are the means of fourfold cross-validation

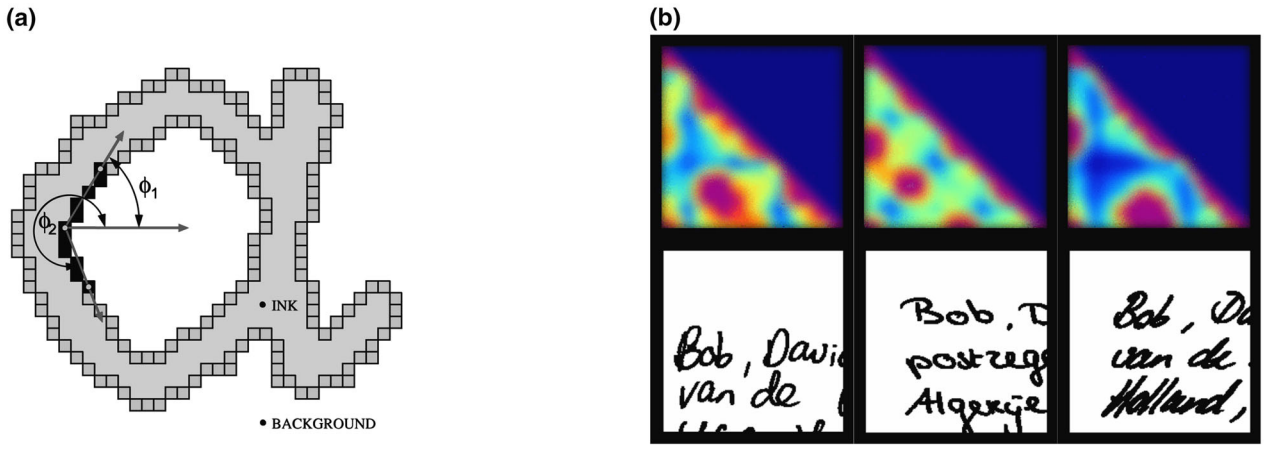

Fig. 7 An illustration of computing contour-hinge features. Diagrams of contour-hinge features are courtesy of Dr. Lambert Schomaker and Dr. Marius Bulacu. **a** Is a copy of Fig. 3a in the their paper [6]. *Red color* in **b** means high values and *blue* means low values (color figure online)

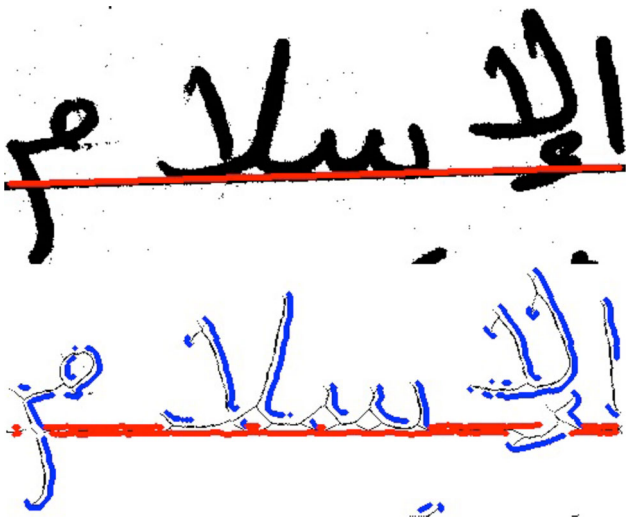

Fig. 8 Accounting for rulings during feature extraction. In the lower half, *blue pixels* are valid contour points that contribute to the contour-hinge features, while *red pixels* are contour points that overlap ruling lines (color figure online)

4.2 Positive contributions of ruling lines to writer identification

To eliminate ruling effects in feature extraction, we skip two consecutive contour segments if they lie on a ruling. We compensate for the page skew by subtracting it from the local angles. We detect "salt-and-pepper" noise and ignore it since its contour length is usually small. In this way, we can effectively handle scanning noise, page skew, and the presence of rulings during feature extraction, rather than in a conventional preprocessing stage. Figure 8 shows invalid contour points overlapping rulings in red and valid contour points in blue. Some contour segments are not colored because using only half of the probability distribution function (PDF) matrix is recommended [6].

Keeping ruling lines does more than preserve the integrity of the input handwriting by obviating the need to repair broken strokes. As the following experiment shows, the position of the script relative to the ruling lines can be a discriminating characteristic to boost the accuracy of writer ID. Like automobile drivers, it appears that some writers meticulously keep to the middle of the lane, while others are line huggers or even line straddlers.

We evaluated three strategies of handling preprinted ruling lines in our experiments. *Remove-Ruling* is the baseline paradigm of using ruling removal followed by broken stroke recovery. The other two methods detect and exploit ruling lines without removing them.

> *Remove-Ruling* remove ruling lines and try to recover broken strokes by local shape analysis [8].

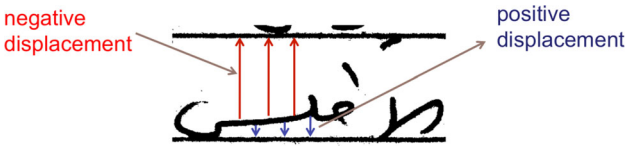

Fig. 9 An illustration of computing displacement features that exploit preprinted ruling lines

> *Offset-Ruling* detect ruling lines using a model-based method and account for them during feature extraction. *Exploit-Ruling* add displacement features to *Exploit-Ruling*.

4.2.1 Experimental setup

The Arabic dataset we used for this evaluation was also provided by the Linguistic Data Consortium (LDC [47]). 61 writers contributed, each providing 10 handwritten pages.[1] Each page was scanned at 600 DPI using a bitonal setting; a typical size for a page image is $5100w \times 6600h$. We divided the dataset into fivefold, each containing two pages by each writer. Each page was then annotated with non-isothetic bounding polygons for handwritten words and text lines, along with the corresponding text transcription. Using fivefold cross-validation, each text line was tested once. We classified 4890 text lines using an SVM with a radial basis function (RBF) kernel. As consistent with our stated philosophy, conventional preprocessing steps such as median filtering or deskewing were not used (Fig. 9).

4.2.2 Experimental results

Figure 10 presents the top-1 accuracy of the different approaches. As can be seen, *Offset-Ruling* outperformed the baseline *Remove-Ruling* system. This result shows that it is feasible to handle ruling lines during feature extraction without error-prone recovery of broken strokes. On the image samples where *Offset-Ruling* performed better, we observed mutilated strokes in the *Remove-Ruling* output which led to classification errors.

Moreover, the presence of preprinted rulings helped boost writer ID. Adding the displacement features in *Exploit-Ruling* increased the accuracy to 67.7 %, compared to 65.8 % for *Offset-Ruling*, and 61.2 % for *Remove-Ruling*. These accuracy gains are statistically significant with a confidence level of 95 % (see "Appendix" section).

[1] This differs from the previous 60-writer setup because of new releases of datasets from LDC.

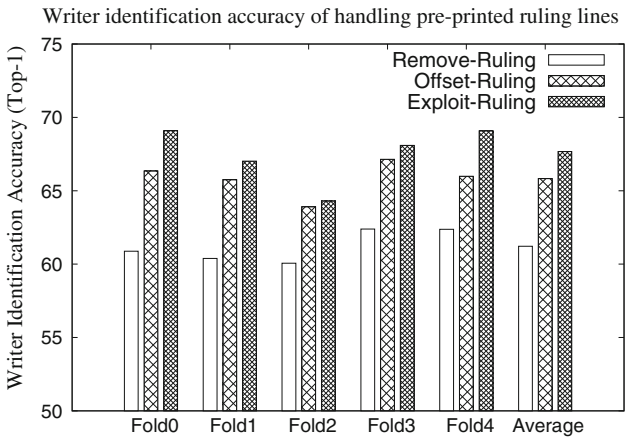

Fig. 10 Writer identification accuracy on different approaches of handling preprinted ruling lines. All the numbers are top-1 accuracy in the output n-best lists

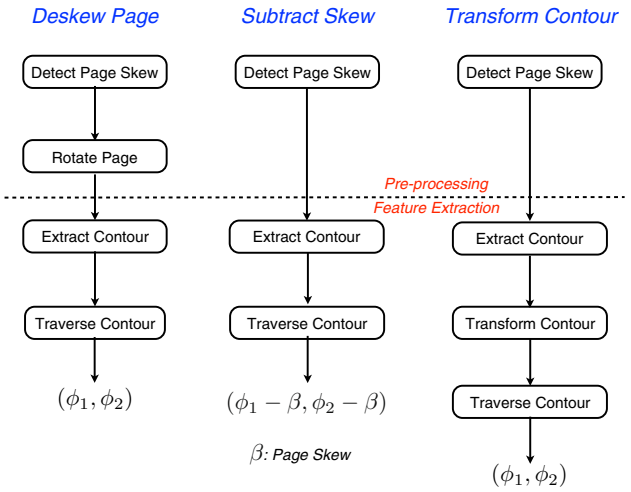

Fig. 11 The three different feature extraction methods under study in the presence of page skew. $(\cdot, \cdot)$ means the actual angles used to index in the PDF matrix

5 Conservative accommodation of page skew

Rotating a bitmap on a 2-D grid can cause distortion, as shown in Fig. 4. To extract local slope features critical for identifying writers, we present a conservative approach that rotates the features instead of the bitmap. We also describe an image processing anomaly that is not due to preprocessing, but rather to insufficient attention paid to offsets that affect intrinsic symmetries.

5.1 Experimental setup

We examine two alternatives to page rotation when computing hinge contour features: (1) explicitly subtract the page skew from the hinge segment angles, and (2) after contour traversal transform the coordinates of extracted contour pixels by rotating them through the negative of the estimated skew angle before extracting the hinge features. Hence, including the traditional resampling approach to deskewing, we have three methods to evaluate, as shown in Fig. 11.

These methods differ in the order that page skew is handled in the processing pipeline. *Deskew Page* normalizes page skew during preprocessing, by rotating the bitmap directly. *Subtract Skew* ignores the presence of page skew until computing the indices of bins in the PDF matrix. *Transform Contour* first extracts the contours and then rotates them in a continuous coordinate system before computing the contour-hinge angles. Note that this obviates the problematic coordinate transform in *Deskew Page* because here we can use real values to represent point coordinates.

Deskew Page baseline system which rotates the image to counter the detected page skew, as is common in traditional preprocessing.

Subtract Skew subtracts the page skew from the angles of hinge segments and then computes the indices in the PDF matrix.

Transform Contour counters page skew by transforming the coordinates of extracted contours in a continuous coordinate system.

In a world without quantization effects, these methods would generate the same feature vectors. Due to the discrete 2-D grid, however, this will often not be the case. This is illustrated in Fig. 4. A way to quantify a more subtle version of this effect is by extracting features for a compute-generated ellipse under rotations in the interval $[-1.0°, 1.0°]$.

In the work that proposed the use of contour-hinge features [6], the authors only computed half of the PDF matrix as a feature vector ($\phi_2 \geq \phi_1$), regarding the other half as being redundant due to symmetry. We examine the benefits of using the entire PDF matrix as a feature vector.

Half_PDF the baseline method from the original literature which uses half of the PDF matrix as a 300-dimensional feature vector, as in [6].

Full_PDF uses the full PDF matrix for $n^2 = 576$-dimensional feature vectors.

5.2 Experimental results

The experimental results comparing *Transform Contour* and *Transform Contour* with *Subtract Skew* are summarized in Fig. 12. While the baseline seems to outperform *Subtract Skew*, McNemar's test indicates that the difference of 0.4 % is not statistically significant: the two systems perform similarly. With *Transform Contour*, however, we obtain an

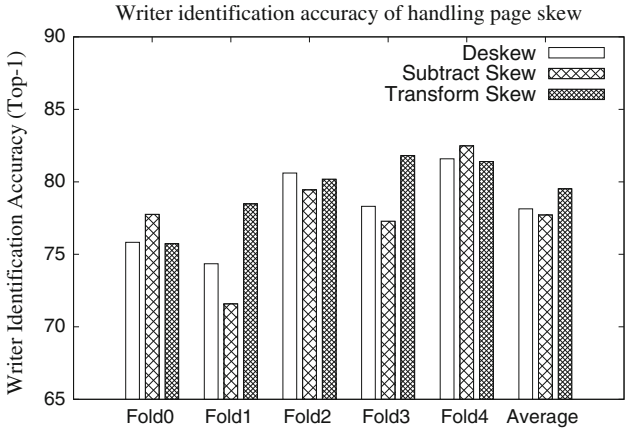

Fig. 12 Writer identification accuracy using different ways of handling page skew

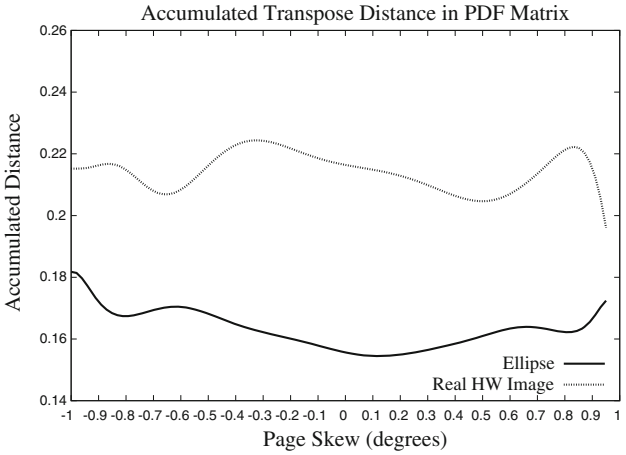

Fig. 13 Transposed PDF matrix distances of different objects

Table 4 Asymmetric PDF matrix in feature extraction

	Half PDF (%)	Full PDF (%)
Deskew	73.5	78.1
Transform contour	74.9	79.5

accuracy gain (1.4 %) which is statistically significant. This result validates our hypothesis that it is possible to avoid degradations caused by rotating bitmaps during traditional preprocessing and instead account for page skew during feature extraction (Table 4).

For our implementation of feature extraction, we found that the full PDF matrix is not symmetric as originally assumed. Bulacu and Schomaker used only half of the matrix, because they considered the other half redundant. They assumed that the contours are symmetric with respect to the horizontal axis, so that the PDF matrix will be symmetric. We investigated this assumption by rotating a standard ellipse by different angles to test feature extraction. For each skew angle in $[-1.0°, 1.0°]$, we computed the distance D between the PDF matrix M and its transpose:

$$D = \sum_{i=1}^{n} \sum_{j=i+1}^{n} \|M[i][j] - M[j][i]\| \tag{4}$$

where $n = 24$. Then, we computed the average distance for each bin in the skew range. We also computed this metric using all the text line images.

The asymmetry is illustrated by both the standard ellipse with zero skew that produces a nonzero transpose distance and, to a lesser extent, by the handwriting primitives. Figure 13 shows the discrepancy of the accumulated transpose distances for a standard ellipse and for a typical handwritten word.

The lack of symmetry is due in part to partitioning contours of arbitrary length into fixed-length segments when extracting local hinge features and in part to the setting the 15° bin boundaries for histogramming the hinge angles. These details are also likely to explain the lack of improvement by *Subtract Skew*. Unexpected, unwanted asymmetry in the implementation of image processing algorithms is far from rare. This can be readily detected, for example, by checking whether 90° rotations of the input image produce 90° rotations of the output. Because of the observed lack of symmetry, we choose to use the full PDF matrix for feature extraction.

6 Conclusion

Conservative preprocessing preserves data in an image that could be valuable to later steps, such as feature extraction. We advocate preserving all information in the original image by either (1) using only invertible transformations or (2) describing rather than altering regions of interest. This is certainly not restricted to the DIA task of writer identification, which we have used as our illustrative application.

We are aware that deep learning, the most popular current method for many classification and regression tasks, bypasses most of the DIA pipeline of Fig. 5, including preprocessing. However, it requires significantly more training data than previous approaches; data that might not be readily available in certain scenarios. Although data augmentation is possible, it requires precise parameter calibration to preserve discriminative aspects of handwriting [11]. Second, even if there is enough data, it is still far from obvious what architecture to use: the number, size and type of layers of hidden units, connectivity, choice of activation functions, training regimen, etc. In other words, it is trading "feature engineering" for "architecture engineering." It is tempting to believe

that deep learning is capable of doing anything any other method can do, but the implementation of all pattern recognition applications involves a series of trade-offs. Depending on those trade-offs, deep learning may or may not be a good choice for a given application.

We showed examples where resampling-based preprocessing introduces distortions and demonstrated that maintaining the integrity of the image (by not removing ruling lines and repairing broken strokes) makes possible higher levels of writer identification accuracy. Our experiments also show that page skew can be compensated by rotating features instead of altering the bitmap.

This work is both promising and suggestive, a single instance of what could become a comprehensive approach to building DIA pipelines. In fact, each preprocessing technique listed in Table 1 could be replaced by some type of conservative alternative, depending on a specific DIA task setting. This is a topic for future research.

Acknowledgments We thank the anonymous reviewers for their valuable comments.

Appendix

For binary classification errors [18], we define:

- Type I (*false positive*): detecting a class that is not present.
- Type II (*false negative*): failing to detect a class that is present.

One often needs to compare the accuracy of one classification algorithm with that of another. According to Dietterich's study of five statistical significance tests, McNemar's [14] has a low probability of incorrectly detecting a difference when no difference exists.

Suppose there are two algorithms, baseline $\mathcal{A}$ and proposed $\mathcal{B}$. The available n samples are classified by both algorithms. It is observed that n_{10} of the samples are misclassified by classifier $\mathcal{A}$ but not by $\mathcal{B}$, n_{01} samples are misclassified only by $\mathcal{B}$, and n_{11} samples are misclassified by both algorithms. The accuracy of $\mathcal{A}$ is $\mathcal{A} = (n-n_{10}-n_{11})/n$, and the accuracy of $\mathcal{B}$ is $\mathcal{B} = (n-n_{01}-n_{11})/n$. McNemar's test is formulated as:

$$Z^2 = \frac{(|n_{10} - n_{01}| - 1)^2}{n_{10} + n_{01}}. \tag{5}$$

where n_{01}, number of samples misclassified by the proposed algorithm $\mathcal{B}$, but not by the baseline $\mathcal{A}$; n_{10}, number of samples misclassified by the baseline $\mathcal{A}$, but not by the proposed algorithm $\mathcal{B}$; null hypothesis $\mathcal{H}_0$, $\mathcal{A} = \mathcal{B}$; alternative hypothesis $\mathcal{H}_1$, $\mathcal{A} < \mathcal{B}$.

The test statistic Z^2 *approximately* follows the Chi-square distribution with one degree of freedom. As a rule of thumb, we say one algorithm outperforms another significantly with a confidence level of 95 %. The test value Z^2 corresponding to this 95 % confidence is 3.84. Although this statistic test is approximate, it is effective in detecting accuracy differences between algorithms [14].

As an example of the application of McNemar's hypothesis test, consider two cases:

- Case 1: $n_{01} = 10$, $n_{10} = 20$, therefore $Z^2 = 2.70$. We cannot conclude that $\mathcal{B}$ is significantly better than $\mathcal{A}$.
- Case 2: $n_{01} = 5$, $n_{10} = 15$, $Z^2 = 4.05$. Therefore, $\mathcal{B}$ is more accurate than $\mathcal{A}$ at a confidence level of 95 %.

Note that if $n_{11} = 20$ and $n = 100$ in both cases, then in Case 1 $\mathcal{A} = 60\,\%$ and $\mathcal{B} = 70\,\%$. In Case 2, $\mathcal{A} = 65\,\%$ and $\mathcal{B} = 75\,\%$. Different values of n_{11}, the number of samples misclassified by both algorithms, would give different accuracies for $\mathcal{A}$ and $\mathcal{B}$, but that would not change our conclusion with respect to their comparative accuracy. At low error rates and large sample sizes, small differences in accuracy can be statistically significant. It is, of course, essential to keep track of the specific errors. The commonly used *recall* and *precision* measure does not provide sufficient information for this test.

All of the differences in error rate reported as significant in Sects. 4 and 5 yielded confidence greater than 95 % with McNemar's test.

References

1. Abd-Almageed, W., Kumar, J., Doermann, D.: Page rule-line removal using linear subspaces in monochromatic handwritten Arabic documents. In: Proceedings of the 12th International Conference on Document Analysis and Recognition, pp. 768–772 (2009)
2. Abdou, I., Wong, K.: Analysis of linear interpolation schemes for bi-level image applications. IBM J. Res. Dev. **26**(2), 667–680 (1982)
3. Agfa: An Introduction to Digital Scanning. Agfa-Gevaert (1994)
4. Arvind, K., Kumar, J., Ramakrishnan, A.: Line removal and restoration of handwritten strokes. In: Proceedings of the 7th International Conference on Computational Intelligence and Multimedia Application, pp. 208–214 (2007)
5. Baird, H.: Document image defect models. In: Baird, H., Bunke, H., Yamamoto, K. (eds.) Structured Document Image Analysis. Springer, Berlin (1995)
6. Bulacu, M., Schomaker, L.: Text-independent writer identification and verification using textural and allographic features. IEEE Trans. Pattern Anal. Mach. Intell. **29**, 701–717 (2007)
7. Burns, P.: Slanted-edge MTF for digital camera and scanner analysis. In: Proceedings of the IS&T 2000 PICS Conference, pp. 135–138 (2000)
8. Cao, H., Prasad, R., Natarajan, P.: A stroke regeneration method for cleaning rule-lines in handwritten document images. In: Procedings

of the MOCR Workshop at the 10th international Conference on Document Analysis and Recognition (2009)
9. Chen, J.: Information preserving processing of noisy handwritten document images. Ph.D. thesis, Lehigh University, Bethlehem, PA (2015)
10. Chen, J., Cao, H., Prasad, R., Bhadwaj, A., Natarajan, P.: Gabor features for offline Arabic handwriting recognition. In: Proceedings of the 9th IAPR International Workshop on Document Analysis Systems, pp. 53–58. Boston (2010)
11. Chen, J., Cheng, W., Lopresti, D.: Using perturbed handwriting to support writer identification in the presence of severe data constraints. In: Proceedings of the Document Recognition and Retrieval XVIII (IS&T/SPIE International Symposium on Electronic Imaging) (2011)
12. Cheriet, M., Kharma, N., Liu, C., Suen, C.: Character Recognition Systems. Wiley, Hoboken (2007)
13. Citing Feng Ping Shan Library, H.K.U.: China, Collection of Genealogies, 1239–2014. http://FamilySearch.org (2015)
14. Dietterich, T.: Approximate statistical tests for comparing supervised classification learning algorithms. Neural Comput. **10**, 1895–1923 (1998)
15. Ding, X.: Machine printed Chinese character recognition. In: Bunke, H., Wang, P. (eds.) Handbook of Character Recognition and Document Image Analysis, 305–329. World Scientific, Singapore (1997)
16. Dodgson, N.: Image resampling. Technical Report. University of Cambridge (1992)
17. Doermann, D., Tombre, K.: Handbook of Document Image and Recognition. Springer, Berlin (2014)
18. Duda, R., Hart, P., Stork, D.: Pattern Classification. Wiley, Hoboken (2000)
19. Favata, J., Srikantan, G.: A multiple feature/resolution approach to handprinted digit and character recognition. Int. J. Image Syst. Technol. **7**(4), 304–311 (1998)
20. Fischer, A., Riesen, K., Bunke, H.: Graph similarity features for HMM-based handwriting recognition in historical documents. In: Proceedings of the International Conference on Frontiers in Handwriting Recognition, pp. 253–258 (2010)
21. Gonzalez, R., Woods, R.: Digital Image Processing, 3rd edn. Pearson, New Jersey (2008)
22. Ha, T., Bunke, H.: Image processing methods for document image analysis. In: Bunke, H., Wang, P. (eds.) Handbook of Character Recognition and Document Image Analysis. World Scientific, Singapore (1997)
23. Hu, M.: Visual pattern recognition by moment invariants. IRE Trans. Inf. Theory **8**(2), 179–187 (1962)
24. Jung, D., Krishnamoorthy, M., Nagy, G., Shapira, A.: N-tuple features for OCR revisited. IEEE Trans. Pattern Anal. Mach. Intell. **18**(7), 734–745 (1996)
25. Khotanzad, A., Homg, Y.: Invariant image recognition by Zernike moments. IEEE Trans. Pattern Anal. Mach. Intell. **12**(5), 489–497 (1990)
26. Kmiec, M.: New optimal character recognition method based on Hu invariant moments and weighted voting. J. Appl. Comput. Sci. **19**(1), 33–50 (2011)
27. Krishnamoorthy, M., Nagy, G., Seth, S., Viswanathan, M.: Syntactic segmentation and labeling of digitized pages from technical journals. IEEE Trans. Pattern Anal. Mach. Intell. **15**(7), 737–747 (1993)
28. Kumar, J., Doermann, D.: Fast rule-line removal using integral images and support vector machines. In: Proceedings of the 11th International Conference on Document Analysis and Recognition, pp. 584–588 (2011)
29. Liu, C., Sako, H., Fujisawa, H.: Handwritten Chinese character recognition: alternatives to nonlinear normalization. In: Proceedings of the International Conference on Document Analysis and Recognition, pp. 524–528 (2003)
30. Lowe, D.: Object recognition from local scale-invariant features. In: Proceedings of the International Conference on Computer Vision, pp. 1150–1157 (1999)
31. Marinai, S.: Introduction to document analysis and recognition. In: Marinai, S., Fujisawa, H. (eds.) Machine Learning in Document Analysis and Recognition, pp. 1–20. Springer, Berlin (2008)
32. Mohamad, R.A.H., Likforman-Sulem, L., Mokbel, C.: Combining slanted-frame classifiers for improved HMM-based Arabic handwriting recognition. IEEE Trans. Pattern Anal. Mach. Intell. **31**(7), 1165–1177 (2009)
33. Nadler, M., Smith, E.: Pattern Recognition Engineering. Wiley, Hoboken (1993)
34. Nagy, G.: Optical scanning digitizers. IEEE Comput. **16**(5), 13–24 (1983)
35. Nagy, G.: Preprocessing document images by resampling is error prone and unnecessary. In: Proceedings of the SPIE Conference on Document Recognition and Retrieval (2013)
36. Natarajan, P., Lu, Z., Bazzi, I., Schwartz, R., Makhoul, J.: Multilingual machine printed OCR. Int. J. Pattern Recognit. Artif. Intell. **15**(1), 43–63 (2001)
37. Otsu, N.: A threshold selection method from gray-level histograms. IEEE Trans. Syst. Man Cybern. **9**(1), 62–66 (1979)
38. Ouyang, T., Davis, R.: Recognition of hand drawn chemical diagrams. In: Proceedings of the Association for the Advancement of Artificial Intelligence (2007)
39. Pan, P., Zhu, Y., Sun, J., Naoi, S.: Recognizing characters with severe perspective distortion using hash tables and perspective invariants. In: Proceedings of the International Conference on Document Analysis and Recognition, pp. 548–552 (2011)
40. Parker, J., Kenyon, R., Troxel, D.: Comparison of interpolating methods for image resampling. IEEE Trans. Med. Imaging **2**(1), 1983 (1983)
41. Rocha, J., Pavlidis, T.: Character recognition without segmentation. IEEE Trans. Pattern Anal. Mach. Intell. **17**(9), 903–909 (1995)
42. Rowley-Brooke, R., Pitié, F., Kokaram, A.: A non-parametric framework for document bleed-through removal. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2954–2960 (2013)
43. Sarkar, P., Lopresti, D., Zhou, J., Nagy, G.: Spatial sampling of printed patterns. IEEE Trans. Pattern Anal. Mach. Intell. **20**(3), 344–351 (1998)
44. Sivaramakrishna, R., Shashidharf, N.: Hu's moment invariants: How invariant are they under skew and perspective transformations? In: Proceedings of the WESCANEX 97: Communications, Power and Computing, pp. 292–295 (1997)
45. Smith, B.: Characterization of image degradation caused by scanning. Pattern Recognit. Lett. **19**(13), 1191–1197 (1998)
46. Sridhar, M., Houle, G., Bakker, R., Kimura, F.: Comprehensive check image reader. In: Chaudhuri, B., Parui, S. (eds.) Advances in Digital Document Processing and Retrieval, pp. 123–156. World Scientific, Singapore (2014)
47. The Linguistic Data Consortium. http://www.ldc.upenn.edu/ (2013)
48. Tatele, S., Khare, A.: Character recognition and transmission of characters using network security. Int. J. Adv. Eng. Technol. **11**, 351–360 (2011)
49. Teague, M.: Image analysis via the general theory of moments. J. Opt. Soc. Am. **70**(8), 920–930 (1980)
50. Uchida, S., Sakeo, H.: A survey of elastic matching techniques for handwritten character recognition. Trans. Inst. Electron. Inf. Commun. Eng. **88**(D8), 1781–1790 (2005)
51. Wang, X., Yiao, B., Ma, J.F.: Scaling and rotation invariant analysis approach to object recognition based on Radon and Fourier–Mellin transforms. Pattern Recogn. **40**(12), 3503–3508 (2007)

52. Watt, S., Dragan, L.: Recognition for large sets of handwritten mathematical symbols. In: Proceedings of the International Conference on Document Analysis and Recognition, pp. 740–744 (2005)
53. Wolf, C.: Document ink bleed-through removal with two Hidden Markov Random Fields and a single observation field. IEEE Trans. Pattern Anal. Mach. Intell. **32**(3), 431–447 (2010)
54. Yamada, H., Yamamoto, K., Saito, T.: A nonlinear normalization method for Kanji character recognition-line density equalization. Pattern Recognit. **23**(9), 1023–1029 (1990)
55. Yap, P., Paramesran, R., Seng-Huat, O.: Image analysis by Krawtchouk moments. IEEE Trans. Image Process. **12**(11), 1367–1377 (2003)