IJDAR
DOI 10.1007/s10032-011-0176-2

ORIGINAL PAPER

Coupled snakelets for curled text-line segmentation from warped document images

Syed Saqib Bukhari · Faisal Shafait · Thomas M. Breuel

Received: 14 February 2011 / Revised: 26 September 2011 / Accepted: 29 September 2011

Abstract Camera-captured, warped document images usually contain curled text-lines because of distortions caused by camera perspective view and page curl. Warped document images can be transformed into planar document images for improving optical character recognition accuracy and human readability using monocular dewarping techniques. Curled text-lines segmentation is a crucial initial step for most of the monocular dewarping techniques. Existing curled text-line segmentation approaches are sensitive to geometric and perspective distortions. In this paper, we introduce a novel curled text-line segmentation algorithm by adapting active contour (snake). Our algorithm performs text-line segmentation by estimating pairs of x-line and baseline. It estimates a local pair of x-line and baseline on each connected component by jointly tracing top and bottom points of neighboring connected components, and finally each group of overlapping pairs is considered as a segmented text-line. Our algorithm has achieved curled text-line segmentation accuracy of above 95% on the DFKI-I (CBDAR 2007 dewarping contest) dataset, which is significantly better than previously reported results on this dataset.

S. S. Bukhari (✉) · T. M. Breuel
Image Understanding and Pattern Recognition Research,
Department of Computer Science Technical,
University of Kaiserslautern, 67663 Kaiserslautern, Germany
e-mail: bukhari@informatik.uni-kl.de

T. M. Breuel
e-mail: tmb@informatik.uni-kl.de

F. Shafait
Multimedia Analysis and Data Mining Competence Center,
German Research Center for Artificial Intelligence (DFKI),
67663 Kaiserslautern, Germany
e-mail: faisal.shafait@dfki.de

Keywords Curled text-line segmentation · Page segmentation · Camera-captured document image processing

1 Introduction

Text-line segmentation is one of the important layout analysis steps in document image understanding systems. It is usually applied before feeding text to an optical character recognition (OCR) system. Text-lines information can also be used for implementing most of the other document image processing tasks such as binarization [1], document cleanup [2], skew correction [3–5], zone segmentation [6], indexing/retrieval bas on word and character recognition [7], dewarping of camera-captured warped document images [8]. Dewarping is relatively a new document image pre-processing step when compared to others which are mentioned here. It is a process of rectifying camera-captured document images that suffer from perspective and geometric distortions. It can be done either by applying stereo vision techniques [9] or by using monocular dewarping techniques [8]—a dewarping technique that is developed for images which are captured by single camera is called a *monocular dewarping* technique. Most of the stat-of-the-art monocular dewarping methods are based on text-line segmentation.

Documents are traditionally digitized using scanners. When a page containing straight, horizontal text-lines is scanned, the resulting scanned image may have horizontal or skewed text-lines owing to the paper positioning distortions introduced by the scanning process, as shown in Fig. 1. These types of document images are referred to as planar document images. There is a large number of state-of-the-art techniques for planar document image segmentation [10], such as projection profile [11,12], Hough transform [13],

Published online: 20 October 2011

Fig. 1 Examples of scanned document images. **a** Straight, horizontal text-lines. **b** Skewed text-lines (8° skew angle)

run-length smearing [14], Docstrum [6], branch and bound method [15]. Most of the commercial and open-source OCR systems work on the assumption that input document images are planar in nature.

Nowadays cameras are available widely at low cost and offer fast, flexible, and non-contact document imaging. These advantages make cameras a potential substitute of scanners for document digitization. Liang et al. [16] presented a brief comparison between scanners and cameras and concluded that camera-based document analysis systems are more flexible than scanner-based systems. The camera-captured document image of a planar document surface is shown in Fig. 2a, where the captured image looks like a scanned image.

However, some image degradations come along with the flexibility of using digital cameras for document imaging. For a planar document surface, a digital camera can produce a distorted image due to perspective distortion that arises from the perspective viewpoint of the camera, as shown in Fig. 2b. Furthermore, for a thick book page, a digital camera can produce a distorted image because of geometric distortion that is caused by the curled document surface. In such a case, the distorted image is composed of curled text-lines with multiple skew angles as shown in Fig. 2c. Therefore, the quality of camera-captured document images generally declines due to perspective and/or geometric distortions.

Camera-captured document images that contain perspective and/or geometric distortions are usually called *warped document images*. The main problem with a warped document image is that it reduces not only human readability, but also causes problems for document image processing, like layout analysis and character recognition. Consequently, dewarping is a necessary step in camera-captured document image processing. Most of the monocular dewarping techniques are based on text-line information [8,17–22]. Therefore, text-line segmentation is an important step in camera-captured document image processing.

A text-line is composed of different typographic lines, i.e., ascender-line, x-line, baseline, and descender-line. For each connected component in a text-line, we defined its top point as the coordinate of its top most pixel and bottom point as the coordinate of its bottom most pixel. These terms, being frequently used in the rest of this paper, are illustrated in Fig. 3 for a sample text-line.

Curled text-line detection in warped, camera-captured document images (Fig. 2c) is a challenging problem. Planar document image segmentation techniques, like Docstrum [6], X–Y cut [12] cannot be robustly applied for curled text-line segmentation [23]. For example, Docstrum is one of the state-of-the-art and widely used planar/straight document image segmentation algorithms, but it performs poorly on warped document image segmentation as shown in Fig. 4.

In recent years, several curled text-line finding methods are proposed in the literature [17–20,22,24–28] mainly in the context of monocular dewarping approaches. Most of these methods use nearest-neighbor–based grouping of connected components for detecting text-lines, but these methods

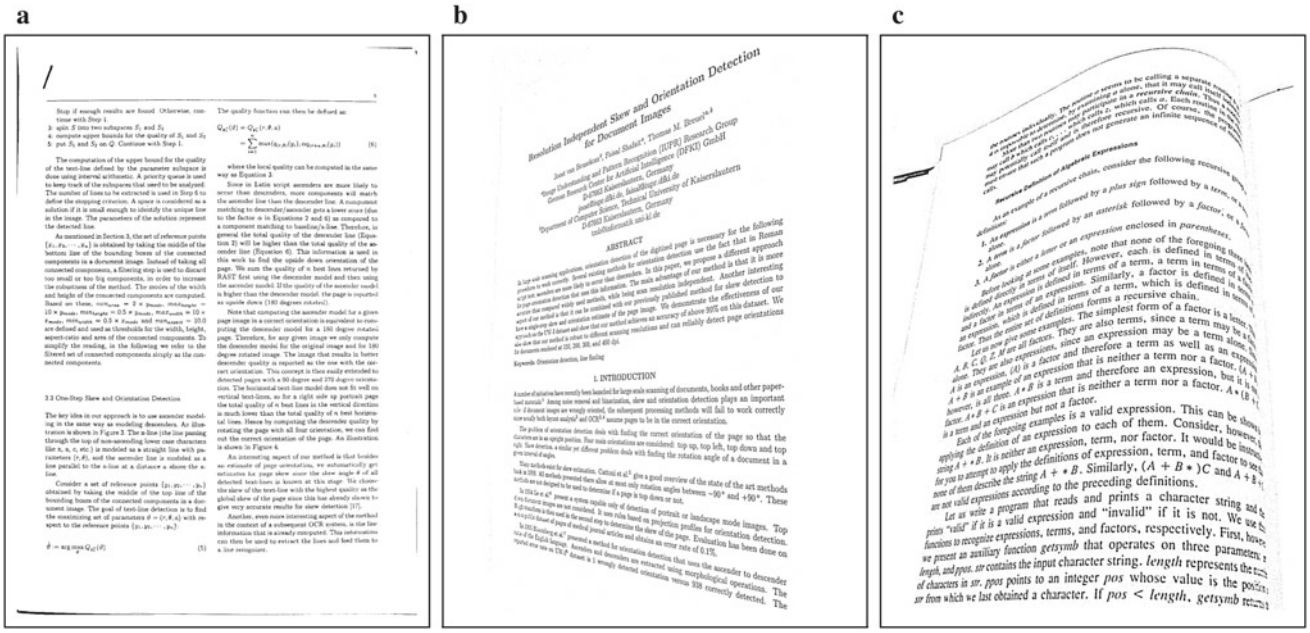

Fig. 2 Examples of camera-captured document images. **a** Straight, horizontal text-lines. **b** Skewed text-lines due to perspective distortion. **c** Curled text-lines due to perspective and geometric distortions

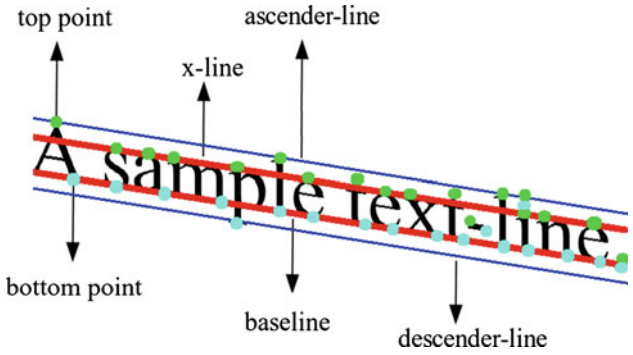

Fig. 3 The typographic lines and points of a sample text-line

usually produce undersegmentation failures in the presence of high degree of curl/skew in document images. Another general observation about the previous approaches is that they estimate x-line and baseline pairs after segmenting text-lines by using regression over top and bottom points of segmented text-lines, respectively, that may result in inaccurate estimation. A brief overview of these methods is given in Sect. 2.

In this paper, we present a curled text-line segmentation method applying active contours (snakes) [29]. We adapt snakes for estimating a local pair of x-line and baseline at each connected component in a document image, where each connected component may represent a character, a broken piece of a character, or a bunch of joined characters. Afterward, each group of overlapping pairs is considered as a segmented text-line that also provides x-line and baseline information of the segmented text-line. Our curled text-line segmentation method is less sensitive to high degree of curl and skew in document images and produces better segmentation results than previous curled text-line segmentation approaches as shown in the performance evaluation section (Sect. 5). Furthermore, unlike other approaches, our algorithm performs segmentation of text-lines and estimation of their x-lines and baselines together at the same time, which also gives more precise x-lines and baselines information than regression-based methods.

Our text-line detection algorithm is designed for hand-held camera-captured images of isolated or bound pages that contain straight text-lines of typed-text Latin script. As mentioned earlier, hand-held camera-captured images usually suffer from perspective distortion (due to camera view angle) and/or geometric distortion (due to curled document surface). Therefore, straight text-lines in documents (as shown in Fig. 2a) are transformed into skewed and/or curled text-lines in camera-captured images (as shown in Fig. 2b, c, respectively). Our algorithm can handle skew and/or curl angle up to $\pm 45°$. It can also deal with variable character sizes within a document image with a minimum (average) character size (length/height) of 10 pixels. Our algorithm can also work in the presence of figures, tables, equations, and noise.

Part of the work presented in this paper was published in [30] for timely dissemination of this work. This paper is a substantially extended version of the previous conference publications. Here, we have described the method in more detail. We have also done an extensive experimental eval-

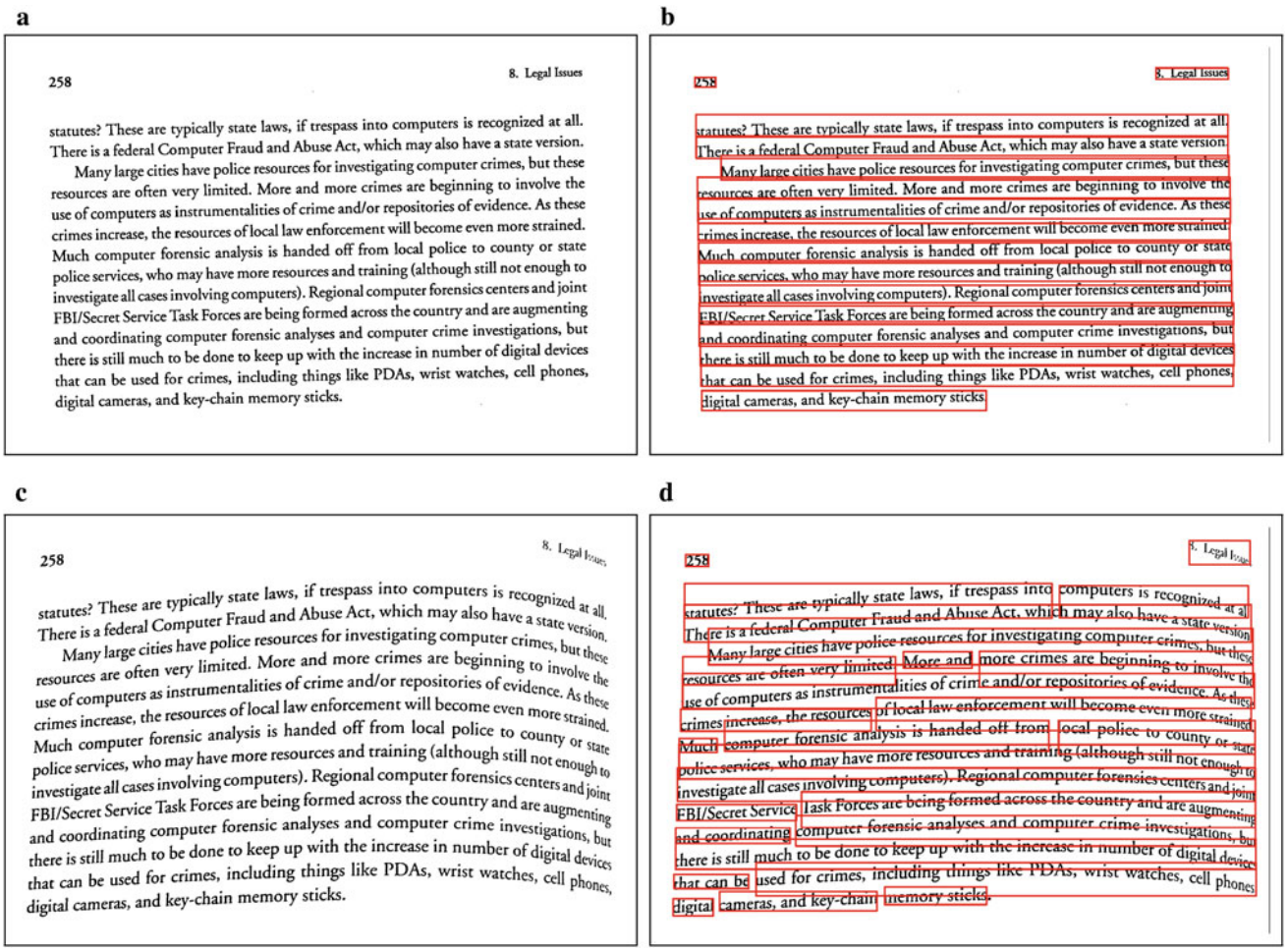

Fig. 4 Text-line extraction results of Docstrum [6]—a state-of-the-art planar document image segmentation technique—for a planar document image and a curled document image. For the curled document image, Docstrum produced a lot of segmentation errors and failed to extract text-lines. **a** Planar (scanned) document image. **b** Accurate text-line segmentation results of Docstrum algorithm. **c** Curled (camera-captured) document image. **d** Text-line segmentation failures of Docstrum algorithm

uation of our method and its comparison with other state-of-the-art techniques.

The rest of this paper is organized as follows. A brief description of previous curled text-line segmentation approaches is presented in Sect. 2. Our coupled snakelets model is described in Sect. 3. Implementation details of our curled text-line segmentation algorithm applying coupled snakelets model are presented in Sect. 4. Performance evaluation and experimental results are given in Sect. 5, followed by a conclusion in Sect. 6.

2 Related work

Several curled text-line segmentation approaches are proposed in the literature [17–20,22,24–28] for camera-captured warped/curled document images. Most of these curled text-lines extraction approaches are mainly proposed as a pre-processing step of monocular dewarping of camera-captured document images. Some of these approaches are briefly discussed here.

Goto and Aso [24] proposed a text-line segmentation method for a document image that may contain curved text-lines with arbitrary orientations. Their algorithm is based on linking of locally linear components. First, the *primitive rectangles* are estimated from the connected components of a document image. Then, these rectangles are grouped together on the basis of a predefined criteria to achieve segmented text-lines.

Zhang et al. [17] introduced a curled text-line finding algorithm using *box-hand* [31] approach. In this algorithm, connected components are first combined to form words using nearest-neighbor analysis. Then, a pair of left and right rectangular box-hands are attached with each word. Each chain of overlapping words is considered as a segmented text-line.

Loo and Tan [25] proposed a word and sentence extraction method for a document image that may contain a wide variety of text-line orientations and layouts. Their algorithm is based on the *irregular pyramid structure* that help in merging characters into words and then words into sentences.

Lu and Tan [18] proposed a curled text-line segmentation approach, where top and bottom points of connected components are first estimated by using morphological operations. Then, text-line detection is performed by tracking either top or bottom points. For a point, left and right nearest-neighbors are searched and this process is repeated for neighbors until no more neighbor is found. The same process is repeated for remaining points. Each group of connected components is considered as a segmented text-line.

Gatos et al. [19] proposed a smearing-based curled text-line detection algorithm. In this approach, horizontal run-length smearing is used to combine characters into words. The height corresponding to the maximum peak of connected components' height histogram (H) is used as a threshold for smearing. After smearing, left and right neighboring words are searched for each word within a limited distance (D) such that $D < 5H$, and the search is repeated until no more neighbors are found. The same process of grouping words together is repeated for the remaining words. Each group of words is referred to as a segmented text-line. We have observed that the algorithm works well on clean document images where the parameter H can be reliably estimated. However, in the presence of salt-and-pepper noise or a large number of broken characters, the estimated value of H is usually too small. This badly affects the performance of their algorithm. We have proposed a slight modification in this algorithm such that if H is less than a predefined threshold (T), all values less than T are removed from the height histogram and the height corresponding to the maximum peak of the remaining histogram is selected as H. The value of T can be set equal to the mean height of a character in a targeted dataset of document images.

Fu et al. [20] proposed a curled text-line segmentation technique using nearest-neighbor analysis over text-lines portions. In this approach, portions of text-lines are first estimated using wavelet-based enhancement technique [32]. These portions are then grouped together using nearest-neighbor approach, where each group is considered as a segmented text-line.

An active contour-based *baby-snakes* curled text-line segmentation algorithm is introduced by Bukhari et al. [26] that is different from coupled snakelets algorithm presented in this paper. Active contour (snake) [29] is one of the state-of-the-art photographic image segmentation techniques. Baby-snakes algorithm adapts active contour for curled text-line segmentation. In this algorithm, an input image is first smeared by using morphological operations. Then, open-curve slope-aligned snakes are initialized over the smeared connected components, which are called "baby-snakes". External energy using gradient vector flow (GVF) [33] is then calculated from smeared document image. This energy is used for baby-snakes deformation. After a few number of deformation steps, neighboring baby-snakes are joined together. Each group of joined snakes is considered as a segmented text-line.

Bukhari et al. [27,28] also introduced another curled text-line detection technique for grayscaling camera-captured document images, which can be equally applied on binary images as well. In this approach, Gaussian filter bank smoothing is first applied over a document image for enhancing/smoothing its text-lines structure. Then, ridge detection method is applied on the smoothed image. Each detected ridge represents a segmented text-line.

Oliveria et al. [22] proposed a rule-based method for warped text-line segmentation. In this algorithm, a *same-size* nearest-neighbor is found for each connected component. All pairs are added into a *priority-queue*. Then, for each pair, nearest-neighbors are iteratively searched in both right and left directions using *moving-window* analysis which holds the following conditions: *same-size*, *smaller than window*, in-between *parallel line with offset*, and distance is less than *maximum distance between letters*. Each group of connected components is referred to as a text-line. Afterward, detected text-lines are further improved by using the following steps. Each text-line is selected one by one in a decreasing text-line's length priority order, and *upper and lower text-lines* are searched for its each component. Two upper and/or lower text-lines are merged together if they satisfy some predefined thresholding criteria. The final step is the removal of those text-lines that contain connected components less than some predefined threshold or contain connected component on 10% of image border. Together with some predefined threshold, all of the above italicized terms are defined using some empirically selected values.

Most of the above curled text-lines segmentation methods (like [17–19,24,25]) are based on grouping of connected components using some predefined nearest-neighbor criteria. The main limitation of a nearest-neighbor-based curled text-line finding method is that it can only handle a moderate skew/curl angle, and it produces a number of over- and under-segmentation errors under a high degree of skew/curl. In contrast to nearest-neighbor-based text-line finding methods, our proposed method comparatively produces less number of under-segmentation errors. In baby-snakes method [26], the length of baby-snakes is a sensitive parameter; the method produces large number of over-segmentation errors for a small length and large number of under-segmentation errors for a comparatively big length. When compared to baby-snakes method, our presented method does not contain that problem and results in a small number of over-segmentation and under-segmentation errors. The curled text-line segmenta-

tion method of Oliveria et al. [22] performs well even in the presence of high degree of skew/curl, but it contains a large number of free parameters. Our proposed method contains around six free parameters, where most of them are non-sensitive. We have compared the performance (Sect. 5) of our coupled snakelets-based curled text-line segmentation method with: (i) nearest-neighbors (Gatos et al. [19]), (ii) baby-snakes (Bukhari et al. [26]), (iii) ridges detection (Bukhari et al. [27,28]), and (iv) rule-based (Oliveria et al. [22]), and (v) Docstrum [6]. The main reason of selecting these method for comparison is to show the performance of different categories of curled text-line detection techniques on a common dataset.

3 Coupled snakelets for curled text-line segmentation

Coupled snakelets model for curled text-line segmentation is based on active contour (snake) [29], which is one of the state-of-the-art image segmentation techniques in computer vision. First, a brief description of basic active contour (snake) model is presented in Sect. 3.1, and then salient features of our coupled snakelets model are explained in Sect. 3.2.

3.1 Review of active contours (snake) model

Active contour (snake) was introduced by Kass et al. [29] for image segmentation. A snake is a closed-curve of points $S(s) = [x(s), y(s)]$, where $s \in [0, 1]$, that moves through the spatial domain of an image to minimize the energy function (E):

$$E = \int_0^1 E_{\text{int}}\{S(s)\} + E_{\text{ext}}\{(S(s)\}\text{d}s \tag{1}$$

$$E = \int_0^1 \frac{1}{2}\left[\alpha\{S'(s)\} + \beta\{S''(s)\}\right] + E_{\text{ext}}\{S(s)\}\text{d}s \tag{2}$$

The snake slithers toward a targeted object under the influence of internal energy (E_{int}) and external energy (E_{ext}), where the internal energy is estimated from the snake points, and the external energy is computed from image contents. The internal energy tries to keep the snake's points close to each other and the external energy tries to move the snake toward the boundary of a targeted object. These internal and external energies are defined in such a way that the snake deforms iteratively toward a targeted object and finally wraps around the object's boundary. Internal energy is further decomposed into two factors: (i) $S'(s)$ (first-order derivative of $S(s)$) represents tension within snake's points, (ii) $S''(s)$ (second-order derivative of $S(s)$) represents rigidity within snake's points. The weighted parameters α and β are used for controlling snake's tension and rigidity, respectively. The snake remains more rigid for a big value of β than a small value.

The wight of the external energy can also be controlled by a free parameter that can take a value in between 1 to 0. In this paper, we have defined this parameter as γ. The Eq. 1 can be rewritten as:

$$E = \int_0^1 E_{\text{int}}\{S(s)\} + \gamma E_{\text{ext}}\{(S(s)\}\text{d}s \tag{3}$$

In general, external energy can be calculated from the edge map of an image by using gradient, Gaussian of gradient or gradient vector flow (GVF) [33]. The gradient vectors or Gaussian of gradient vectors have large magnitudes only in the immediate vicinity of the edges; but these vectors are zero in homogeneous regions where image data are nearly constant. Therefore, the range of gradient or Gaussian of gradient-based external energy is limited and it only exists near the edges. In such a case, manual assistance is required for initializing snake near a targeted object. In contrast to these types of external energies, GVF is calculated by using the computational diffusion of gradient vectors iteratively, where it maintains the gradient vectors near the edges and at the same time extends these vectors farther away from the edges into homogeneous regions. Therefore, GVF covers a large range of energies (gradient vectors) around edges that helps to diverge the snake toward the boundary of a targeted object even if it is initialized far away from the object. In such a case, manual assistance is not required for snake initialization.

A simple toy example to illustrate the basic concept of object's boundary detection using active contour (snake) is illustrated in Fig. 5. Traditional active contour mechanism of image segmentation, which is illustrated in Fig. 5, cannot be directly applied for text-lines segmentation in document images as shown in the Fig. 6. In this paper, we adapt active contour (snake) for text-line segmentation. For this purpose, we introduced a *coupled snakelets* model that is derived from active contour (snake) model. Detailed discussion about coupled snakelets model is given in the next section (Sect. 3.2).

3.2 Coupled snakelets model

We have introduced and added some relevant features in the basic active contour (snake) model [29] for making it applicable for text-line segmentation problem. We refer to our adapted active contour (snake) model as *coupled snakelets* model. Some salient features of coupled snakelets model are explained below.

– **Open-Curve Snake:** A text-line can be represented by a close-curve boundary around it or simply by typo-

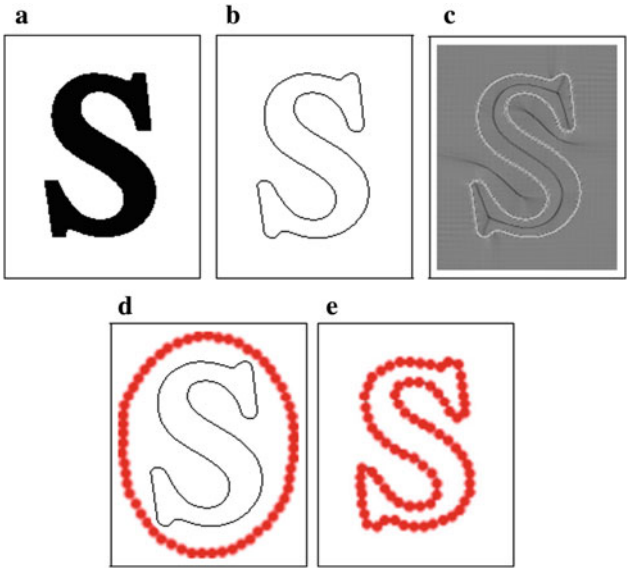

Fig. 5 An example of image boundary detection using active contour (snake) model. **a** Image of an alphabet. **b** Edge map of the alphabet. **c** GVF [33] vectors of the edge map. **d** Initial closecurve snake. **e** Deformed snake

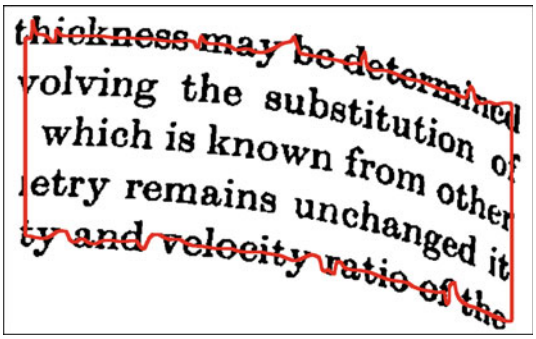

Fig. 6 A traditional process of active contour (snake)-based image segmentation, as shown in Fig. 5, cannot be applied directly for text-line segmentation in document images

graphic lines, for example x-line, baseline, ascender-line or descender-line. Traditional close-curve snakes, as show in Fig. 5d, cannot be used to find close-curve boundary of a text-line due to the close proximity of a text-line to its neighboring text-lines. We introduce the concept of open-curve snakes for text-line segmentation.

In contrast to a close-curve snake, an open-curve snake is a straight line snake. For example, a group of open-curve snakes are shown in Fig. 7b.

– **Multiple Snakes:** Each text-line of a document image consists of several connected components. Furthermore, there are many text-lines in a single document image. Coupled snakelets model uses multiple snakes to cope with this problem. These snakes are deformed independently with respect to one another.
– **Automatic Initialization of Pair of Snakes:** Coupled snakelets model uses automatic initialization of snakes over connected components in a document image. For each connected component, a pair of open-curve snakes is initialized over it such that one snake is initialized at its top point and another one at its bottom point.

Multiple open-curve pairs of snakes that are initialized automatically over connected components of Fig. 7a are shown in Fig. 7b.
– **External Energy Calculation from Discrete Points:** Instead of using an edge map of connected components, coupled snakelets model uses discrete top or bottom points of connected components for GVF (external energy) calculation.
– **Deformation of Snakes in Targeted Direction:** In Latin scripts, text-lines are usually horizontal in nature. Neighboring snakes can be joined together for segmenting text-lines by deforming them in vertical direction only. Coupled snakelets model deforms a snake only in vertical direction such that x-coordinates of the snake points are kept static and y-coordinates of the snake points are deformed with respect to the vertical components of GVF of discrete points.

GVF vectors that represent only vertical components and both vertical and horizontal components are shown in Fig. 7c, d, respectively, for comparison. In coupled snakelets, an open-curve snake is deformed using only vertical components of GVF vectors.
– **Evolving Snakes:** Coupled snakelets model introduces the concept of *evolving snake*. As mentioned earlier, a pair of snakes is initialized at each connected component. For a connected component, both of its top and bottom snakes are deformed independently with respect to the top and bottom points, respectively, in evolving fashion which is described as follows. First, a small rectangular region that centered around the connected component is selected. Then, the top (bottom) snake is deformed with respect to the vertical components of GVF that is calculated from the top (bottom) points of connected components inside the selected area. After the first cycle of deformation, a second cycle is started such that the top (bottom) snake's length and the selected area are increased before deformation. The same process is repeated for a few number of deformation cycles. The main motivation behind this approach is that there exist a number of left, right, top, and bottom neighboring connected components around a particular connected component in a document image. For a small rectangular region around the connected component, almost all of these points belong to the same text-lines to which the connected component belongs. For a big rectangular region, some of these points belong to the same text-line to which the connected component

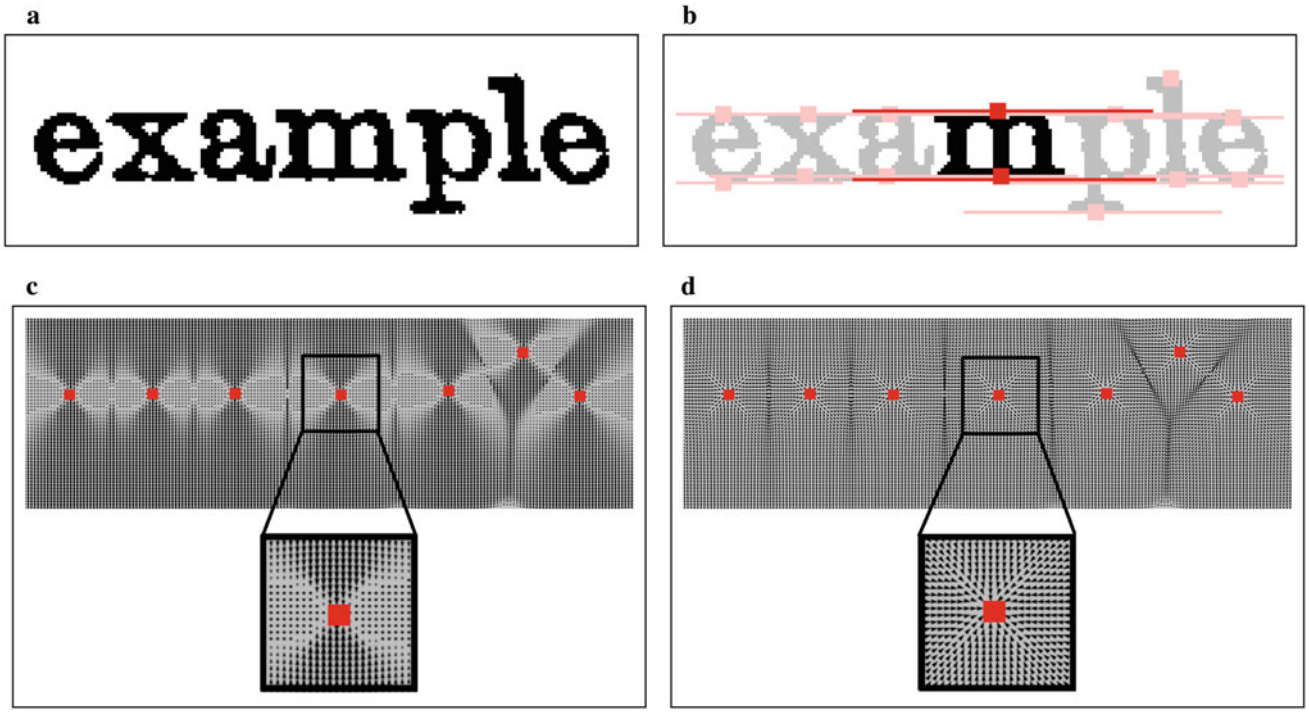

Fig. 7 Coupled snakelets features: **a** An example image, **b** Multiple open-curve snakes are initialized automatically over the *top and bottom points* of connected components, which are shown here by *square symbol* (*red color*), **c** Vertical components of GVF vectors that were calculated using top points (shown in *red color*, *square symbol*) of connected components; it is visible in the enlarged portion that each of theses vectors either points downwards or upwards with some amplitude. **d** Both vertical and horizontal components of GVF vectors for top points, it is visible in the enlarged portion that these vectors are pointing toward all directions. Both of these images (**c** and **d**) are shown here for illustration

belongs, and others belong to the neighboring top and/or bottom text-lines. If all of these top (bottom) points within a big rectangular region are used for external energy calculation, the top (bottom) snake may deform in a wrong upward or downward direction and may cause segmentation failures. In contrast to that, evolving snake criteria expands the top (bottom) snake in a corresponding text-line direction even in the presence of a high degree of skew and/or curl and prevents segmentation failures.

- **Weighted-Coupled Pair of Snakes:** Two or more snakes can also be simultaneously used for a image segmentation such that each of them is deformed independently and then all of them are adjusted before further deformation steps [34–36]. This type of snakes are referred to as coupled snakes. Our coupled snakelets model adapts this idea for curled text-line segmentation. Here, we exploit two general observations of Latin script document images for defining our coupled snakes idea:
  - **Observation #1:** for a text-line in Latin scripts, where ascenders are more frequent than descenders [5], majority of the bottom points of the connected components lie over its baseline when compared to the top points of connected components over its x-line.
  - **Observation #2:** within a text-line, the same distance exists between the pair of its x-line and baseline, as long as the complete text-line has the same font.

For a connected component, a pair of evolving snakes is first initialized over it. Then, on the basis of observation #1, the top snake is deformed using a small weighted percentage of the external energy of top points within a initial selected region, and the bottom snake is deformed using a comparatively large weighted percentage of bottom points. After each deformation step, the top and bottom snakes in the pair are coupled such that first the average distance is calculated from the distances between corresponding pairs of points in the top and bottom snakes and then each corresponding pair of points in these snakes is updated to make its distance equal to the average distance. This type of coupling between the top and bottom evolving snakes is done on the basis of observation #2. In this way, the top and bottom snakes estimate a local pair of x-line and baseline at the connected component after a few number of deformation cycles. The same process is repeated for other connected components as well, and each group of overlapping pairs of snakes represents a segmented text-line.

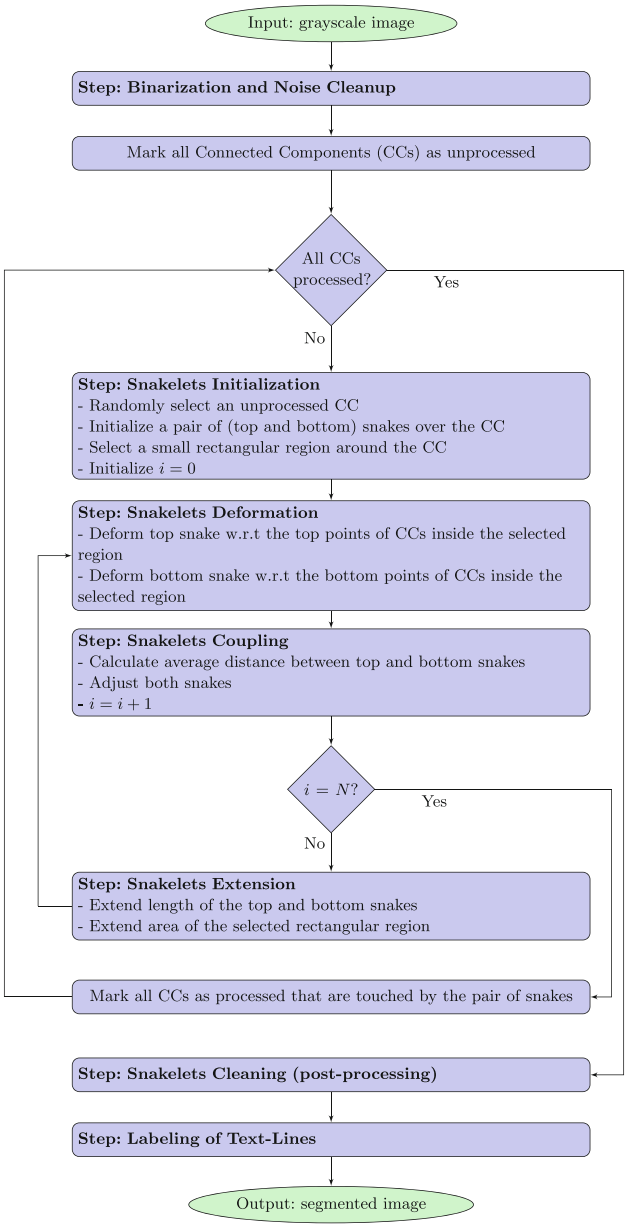

Fig. 8 Snakelets-based curled text-line segmentation algorithm

4 Curled text-line segmentation algorithm

The steps of our curled text-line segmentation algorithm on the basis of above described coupled snakelets model are shown in Fig. 8. Each of these steps is described here in detail.

Binarization and Noise Cleanup: An input grayscaling camera-captured document image is first binarized by using adaptive thresholding technique. The binarization method is defined as follows: "for each pixel, the background intensity $B(p)$ is defined as the 0.8-quantile in a window shaped surrounding; the pixel is then classified as background if its intensity is above this constant fraction of $B(p)$". It should be noted that this binarization scheme is the same as used in [37]. An example binarized image is shown in Fig. 9a. The binarized document image may contain marginal as well as salt-and-pepper noise. A heuristic-based noise cleanup process is applied as follows. Let H_{doc} and W_{doc} represent the height and width of the document image, respectively; H_{avg} and W_{avg} represent the mean height and width of connected components, respectively; σ_{H} and σ_{W} represent standard deviation of heights and widths of connected components, respectively. A connected component, whose height and width are represented by H_{cc} and W_{cc}, respectively, is removed as a large noisy component if any of the conditions specified in Eq. 4 is true or as a small noisy component if the condition specified in Eq. 5 is true:

$$\begin{aligned} H_{\text{cc}} &> (0.1 \times H_{\text{doc}}) \quad \text{or} \quad H_{\text{cc}} > (7 \times \sigma_{\text{H}}) \\ W_{\text{cc}} &> (0.1 \times W_{\text{doc}}) \quad \text{or} \quad W_{\text{cc}} > (7 \times \sigma_{\text{W}}) \end{aligned} \tag{4}$$

$$(H_{\text{cc}} \times W_{\text{cc}}) < \left(\frac{1}{3} \times H_{\text{avg}} \times W_{\text{avg}}\right) \tag{5}$$

After removing noisy connected components, let mean width and mean height of all the remaining connected components be represented by W and H, respectively.

Snakelets Initialization: All the connected components of the binarized document image are marked as unprocessed. Then, a connected component is selected randomly. A pair of horizontal open-curve snakes is initialized over the selected connected component, such that one snake is initialized at its top point and another one at its bottom point. By keeping the connected component at center, a small rectangular region is also selected around it. The initial length of the snakes (L) and the size of the rectangular region ($W_{\text{R}} \times H_{\text{R}}$) are selected in such a way that both of them cover a few neighboring connected components around the selected connected component. Let H_{cc} and W_{cc} represent the height and width of the connected component, respectively. L, W_{R}, and H_{R} are defined as:

$$\begin{aligned} L &:= W_{\text{cc}} + 2 \times W \\ W_{\text{R}} &:= W_{\text{cc}} + 4 \times W \\ H_{\text{R}} &:= H_{\text{cc}} + 2 \times H \end{aligned} \tag{6}$$

The main reason of selecting the width greater than the height of the rectangular region is that text-lines in a document are usually horizontal in nature. An example of an initial pair of snakes and a selected rectangular region for a connected component is shown in Fig. 9b.

Snakelets Deformation: Gradient vector flow (GVF) is calculated by using the top points of all connected components inside the selected region around the connected component. Then, the top snake is deformed by using the vertical components of the GVF with $\gamma/2$ (Eq. 3). Simi-

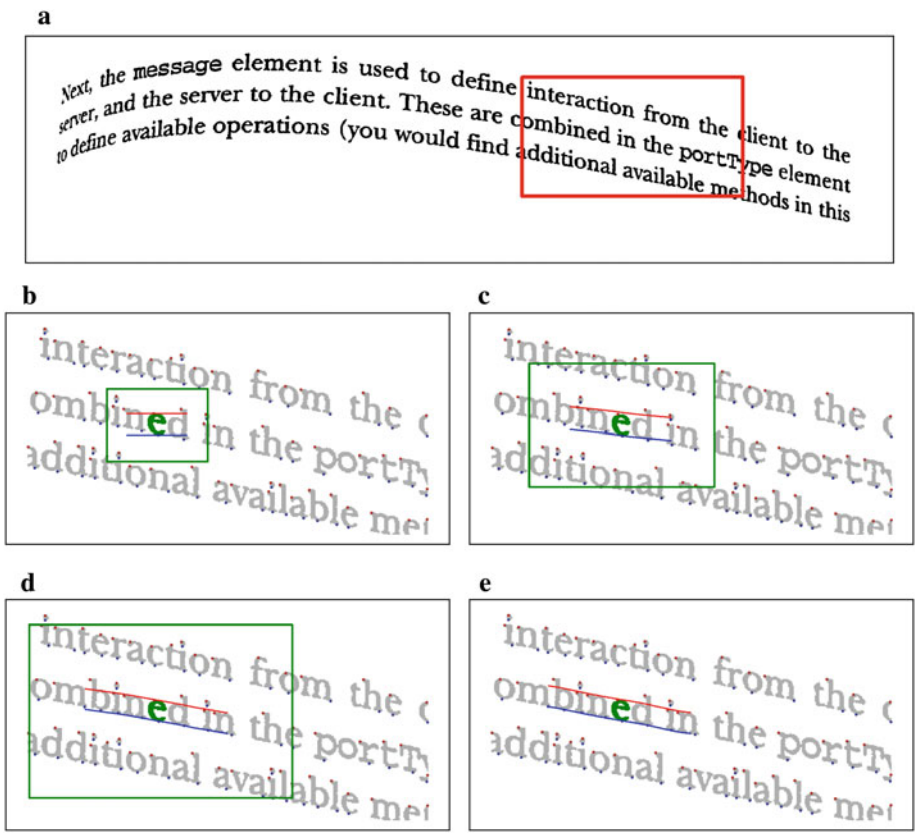

Fig. 9 Coupled snakelets features: illustration of evolving and weighted coupling nature of a pair of snakes ($N = 3$). Note that the snakelets were not mislead by the ascenders "b" and "d" due to their coupling. **a** An example image. **b** Initial pair of coupled snakelets on a connected component. **c** Snakes' pair after first deformation cycle. **d** Snakes' pair after second deformation cycle. **e** Snakes' pair after the third (last) deformation cycle

larly, bottom snake is deformed by using the vertical component of the GVF with γ, where the GVF is calculated form the bottom points of all connected components within the selected region.

Snakelets Coupling: The top and bottom snakes are composed of the same number of points with similar values of x-coordinates. For each common value of x-coordinate of the top and bottom snakes, absolute distance is calculated from the corresponding values of y-coordinates. Then, average distance is computed. Now, for each common value of x-coordinate of both snakes, the corresponding values of y-coordinates are increased or decreased proportionally such that the distance between them becomes equal to the average distance. Snakelets coupling procedure is illustrated in Fig. 10.

Snakelets Extension: First, the average slope of the pair of snakes is calculated. Then, each of the top and bottom snake is extended by a length equal to the average width (W) from both left and right sides, and the slope of these extended lengths is kept the same as the average slope. Similarly, the rectangular region around the connected component is extended, such that its width and height become twice as big compared to its previous width and height after extension.

After snakelets deformation, coupling, and extension steps, the first deformation cycle of the snakes' pair is completed, which is shown in Fig. 9c. The pair of snakes is further processed by a few number (N) of deformation cycles. We empirically found that three iterations of snakelets extension are sufficient for printed Latin script documents. The results of coupled snakelets for two more deformation cycles are shown in Fig. 9d, e, respectively.

The pair of snakes approximates a local pair of x-line and baseline on the connected component, as shown in Fig. 9e. Now all the connected components that are overlapped/touched by the pair of snakes are marked as processed. Afterward, the same process is repeated for another unprocessed connected component and is continued until no more unprocessed connected components are left. Some example images with all computed pairs of coupled snakelets are shown in Fig. 11. In these example images, each group of overlapping/touching pairs of snakes can be considered as a segmented text-line. In these example images, it is also visible that our method can handle a high degree of curl/skew and different font sizes within a document image.

Coupled snakelets-based text-line segmentation algorithm may also cause undersegmentation failures. Some examples of undersegmentation failures of our algorithm are shown in

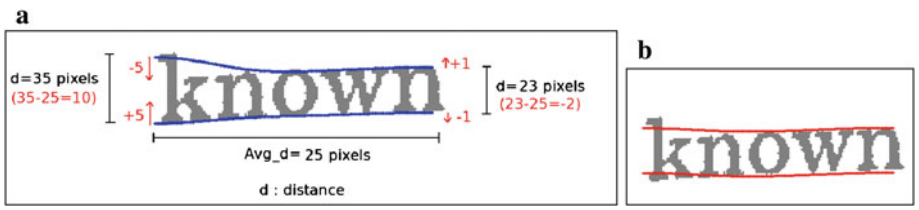

Fig. 10 An illustration of snakelets coupling procedure. **a** A pair of snakelets, **b** The points of the *top and bottom* snakelets are adjusted with respect to the average distance between them. **a** Before snakelets coupling. **b** After snakelets coupling

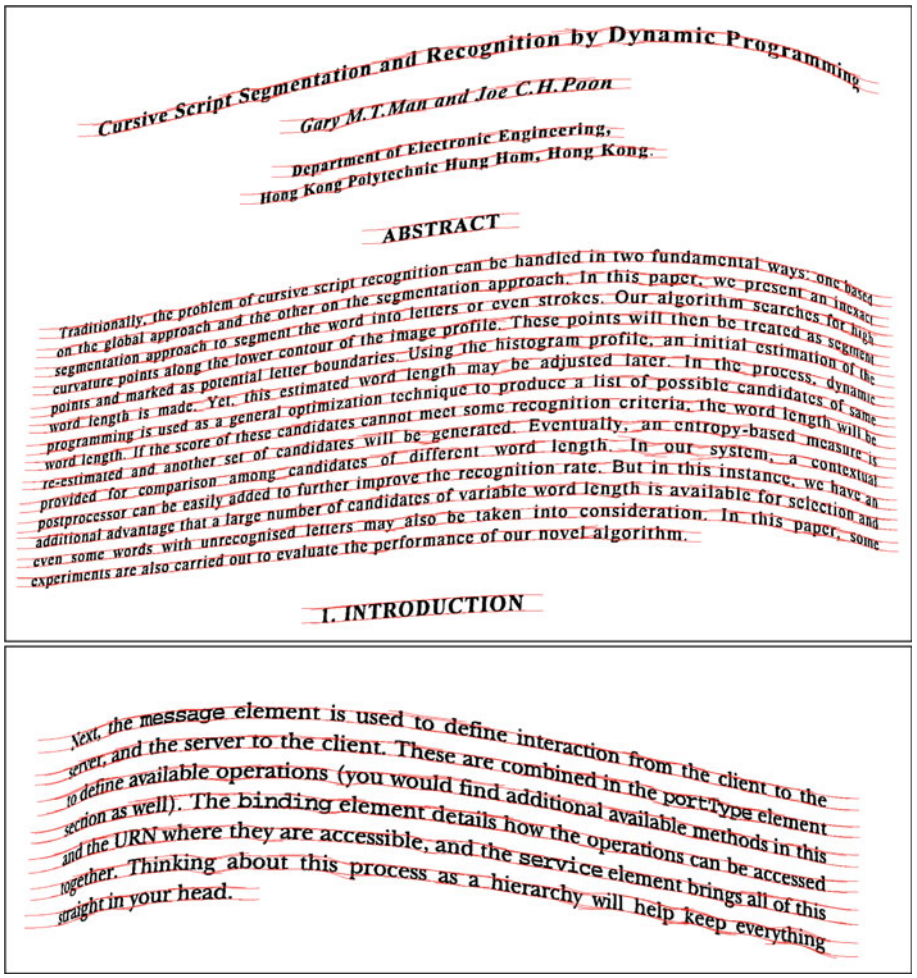

Fig. 11 A few example images of curled text-line segmentation using coupled snakelets algorithm where a pair of snakes estimates a local pair of x-line and baseline and a group of overlapping pairs of snakes represents a segmented text-line with its x-line and baseline information. Our method can handle different fonts sizes within a document image (*top* figure) and high degrees of different directions of curls/skews

Fig. 12a. Such type of segmentation failures occur because of some *badly deformed* pairs of snakes, as marked in Fig. 12a. By badly deformed pairs of snakes, we mean those pairs of snakes that are not correct with respect to the estimation of their corresponding local x-line and baseline pairs and are not uniform with respect to their neighboring pairs of snakes. Such type of badly deformed pairs of snakes mainly occur because of some inherent properties of a document image:

- a document image may contain text-lines with different starting and ending positions with respect to each other, as shown in the left image of Fig. 12a.
- a document image may contain slightly big connected component(s) near to comparatively small connected component(s) as shown in the right image of Fig. 12a.

We have introduced a post-processing step for cleaning up badly deformed pairs of snakes and for achieving better segmentation results.

Snakelets Cleaning (post-processing): We develop the following observations from a close examination of the coupled snakelets (pairs of snakes) in Fig. 12a: (i) the slope of each pair, except the marked ones, is approximately the same as that of the neighboring pairs, (ii) the thickness (average distance) of each pair, except the marked ones, is approximately the same as that of other neighboring pairs. Both or either of these observations do not hold for badly deformed coupled snakelets as shown in Fig. 12a. Therefore, badly deformed coupled snakelets can

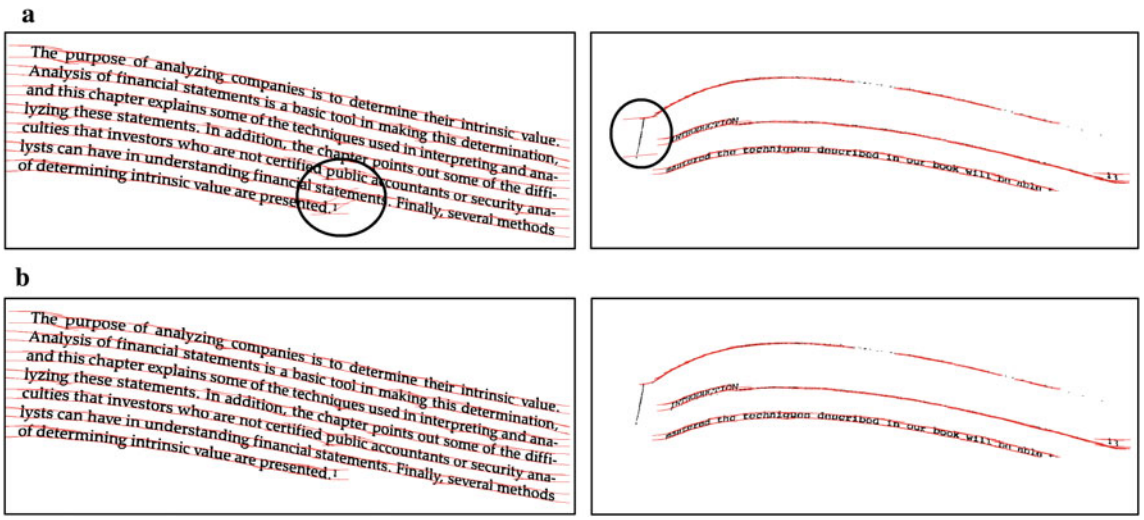

Fig. 12 Example of undersegmentation failures of our algorithm due to badly deformed coupled snakelets and their improvement through snakelets cleaning (post-processing) **a** [*Left*] a badly deformed coupled snakelet due to the presence of a superscript letter in between text-lines that have different ending positions, **a** [*Right*] a badly deformed coupled snakelet due to a big connected component near to comparatively small connected components. **b** Improved text-line segmentation results after removing badly deformed coupled snakelets through snakelets cleaning (post-processing). **a** Under-segmentation errors due to badly deformed coupled snakelets (marked ones). **b** Improved text-line segmentation results after post-processing

be removed by using slope- and thickness-based statistical analysis. Each coupled snakelet is removed as a badly deformed pair if the difference between its slope and the mean slope of neighboring-coupled snakelets is greater than a predefined threshold, or if the difference between its thickness value and the mean thickness value of neighboring coupled snakelets is greater than a predefined threshold. The slope threshold can be set equal to a small value such as 10° or 15° and can be represented by T_S. The thickness threshold can be relatively selected with respect to the average height of connected components (H) and can be represented by $T_T \times H$. The height and length of the neighboring window for estimating the mean thickness and mean slope can also be relatively selected with respect to H and can be represented by $R_{pp} \times H$. Altogether, our snakelets cleaning (post-processing) step contains three free parameters: T_S, T_T, and R_{pp}. The snakelets cleaning process is applied to the examples in Fig. 12a for predefined values of these free parameters, and the remaining coupled snakelets are shown in Fig. 12b. It is clearly visible from these examples that our post-processing cleaning step removed badly deformed coupled snakelets and overcame undersegmentation failures. Furthermore, in performance evaluation (Sect. 5), we have evaluated our text-line segmentation algorithm for different possible values of T_S, T_T, and R_{pp}.

Text-Lines Labeling: As shown in Figs. 11 and 12b, each group of overlapping or touching pairs of snakes represents a group of connected components that belong to a particular text-line. Each group of connected components is assigned a unique text-line label. Each small noisy components that was removed in the noise cleanup step is assigned the label of its nearest text-line. A few example results of curled text-lines segmentation using coupled snakelets algorithm followed by post-processing and text-lines labeling are shown in Fig. 13.

Behavior of Coupled Snakelets under Challenging Conditions: Our algorithm is designed for text-lines segmentation which cannot handle tables and/or formulas segmentation, but it detects text-lines correctly even in the presence of tables and/or formulas as shown in Fig. 14a, b, respectively. Grayscaling camera-captured document images are usually composed of shadows that are captured using a hand-held camera in an unconstrained environment. A local adaptive thresholding produces a small amount of noise from shadows, but a global thresholding technique produces a large amount of shadow-based noise. Our text-line detection algorithm starts with binarization step using a local adaptive thresholding technique, and then, it performs cleanup step in order to remove noise (that are originated from shadows, borders, etc.) and other non-text components (like graphics, drawings). It is also possible that the cleanup process is unable to remove noise and/or non-text components completely. Our algorithm works well even in the presence of the remaining amount of non-text noise and/or shadow noise as shown in Fig. 14c, d, respectively. A binarized camera-captured document image may contain broken or joined characters. Coupled snakelets algorithm can give satisfactory text-lines segmentation result under these conditions as shown in Fig. 14e, f, respectively, until characters are broken into a large number of pieces and/or a large number of characters

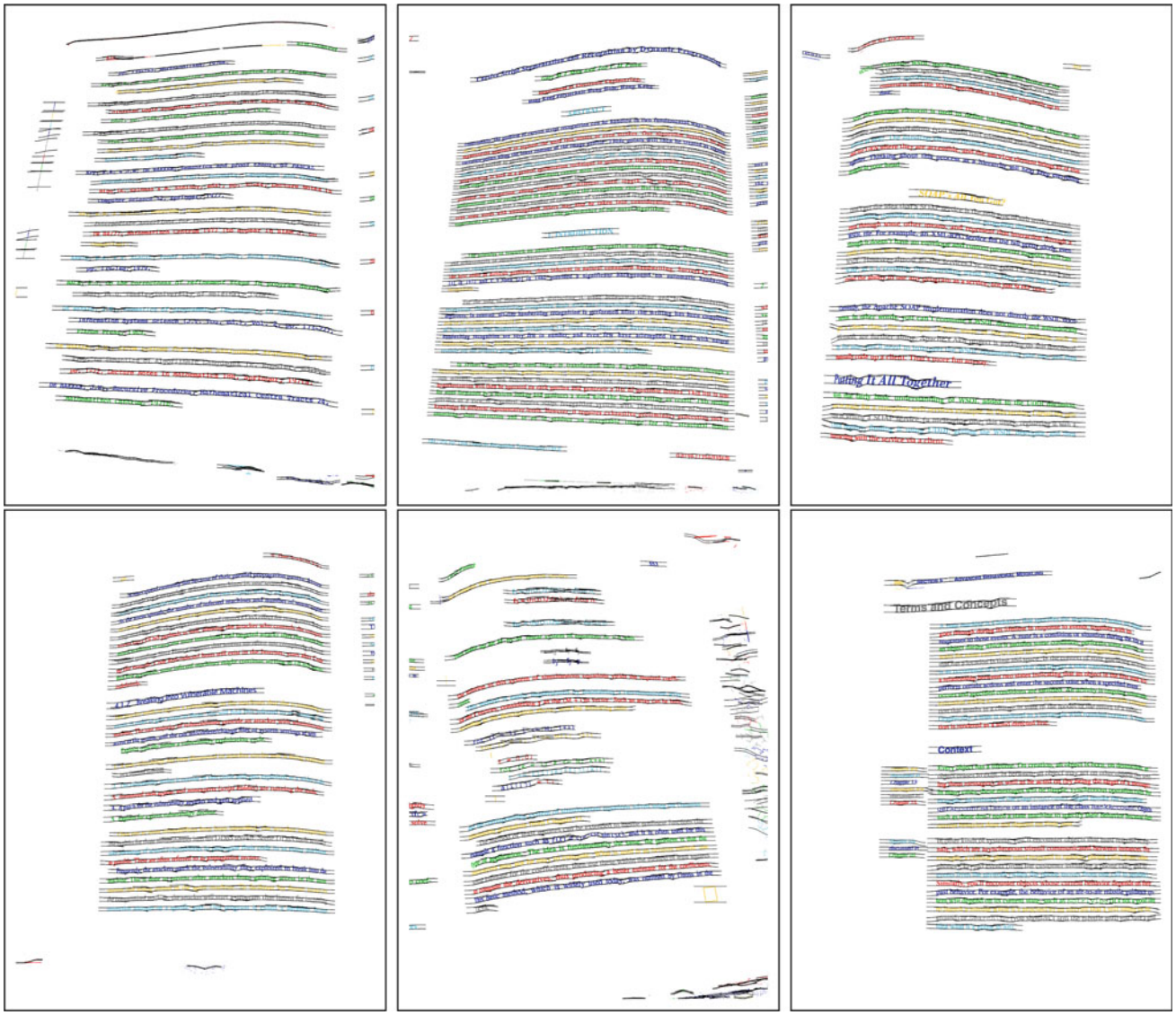

Fig. 13 Accurate curled text-line segmentation results of coupled snakelets algorithm for camera-captured document images of DFKI-I (CBDAR 2007 dewarping contest) dataset. [*Note*: text-line segmentation results are shown in color coded form using repetition of six different colors. Two or more text-lines with same color do not necessarily mean undersegmentation error. In order to avoid this confusion, coupled snakelets are also drawn here to mention the boundary of each segmented text-line]

are joined together. In the presence of big connected components of joined characters, text-line segmentation results can further be improved by using a character segmentation algorithm like dynamic-programming-based curved-cut segmentation [38].

5 Performance evaluation

We have evaluated our coupled snakelets-based curled text-line segmentation algorithm on publicly available DFKI-I (CBDAR 2007 dewarping contest) dataset [39] by using Shafait et al. [10] performance evaluation metrics. This section is further divided into the following three subsections: (1) description of hand-held camera-captured document images DFKI-I dataset that was used in CBDAR 2007 dewarping contest, (2) performance evaluation methodology, and (3) performance evaluation and experimental results.

5.1 DFKI-I (CBDAR 2007 dewarping contest) dataset

DFKI-I (CBDAR 2007 dewarping contest) dataset contains 102 grayscaling and binarized document images of pages from several technical books captured by an off-the-shelf hand-held digital camera in a normal office environment. The captured documents were binarized using a local adaptive thresholding technique, which is described in [37]. Docu-

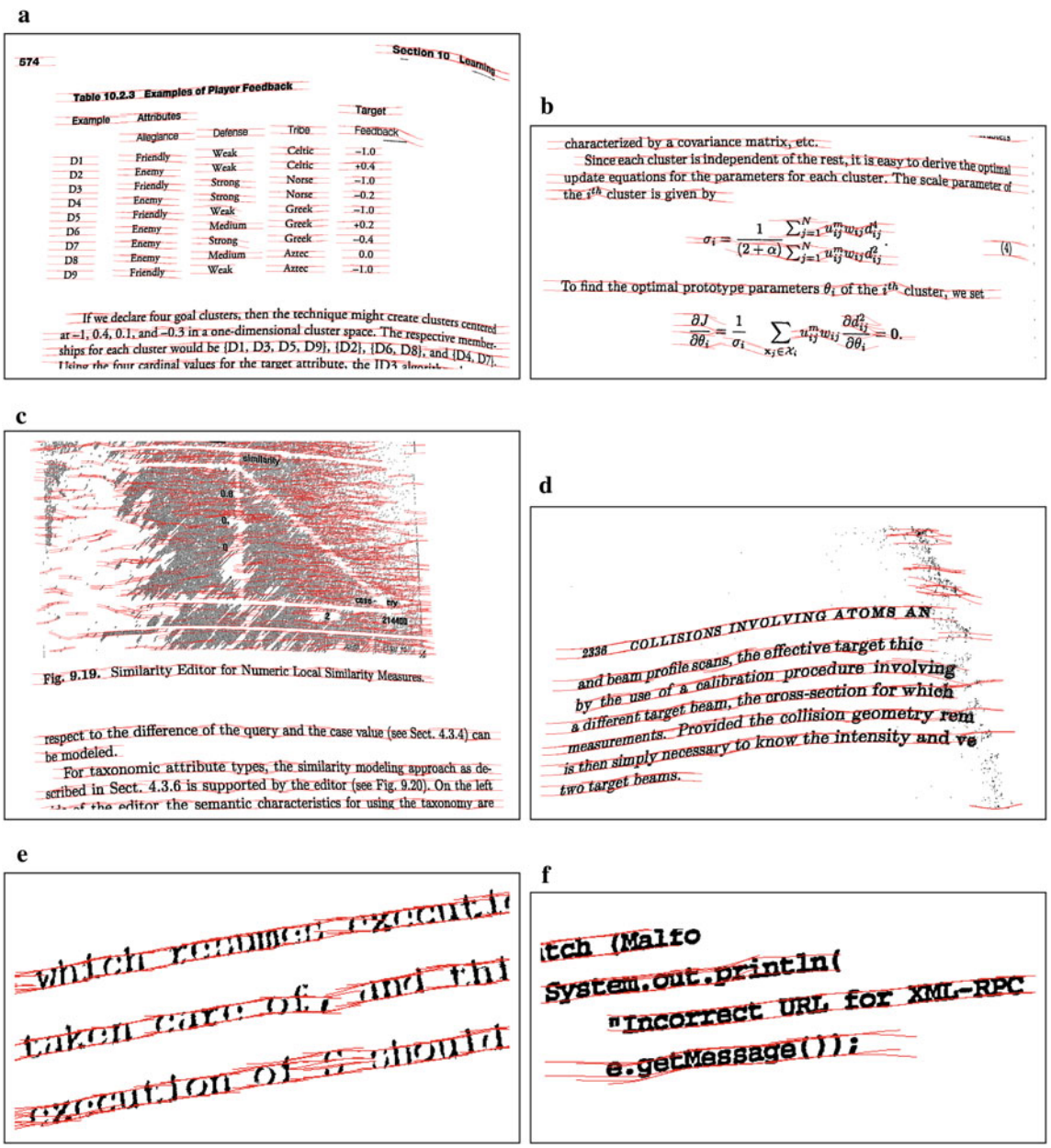

Fig. 14 Behavior of coupled snakelets text-line detection method under challenging conditions. Coupled snakelets-based text-line segmentation algorithm segments text-lines correctly in the presence of tables, formulas, remaining portions of non-text noise, or remaining portion of shadow noise as shown in (**a**)–(**d**). Here, It is also important to note that the snakelets over tables, formulas, or remaining amount of noise produce false alarms. Coupled snakelets-based text-line segmentation algorithm also work well in the presence of broken and/or joined characters as shown in (**e**) and (**f**), respectively

ment images in this dataset consist of warped text-lines with high degree of curl, different directions of curl within an image, non-text (graphics, halftone, etc.) components, and a lot of textual and non-textual border noise. The average size of a document image in this dataset is equal to 7.8 megapixels. The values of mean and standard deviation of the length (L) and height (H) of connected components in this dataset are as follows: $\mu_{\mathrm{L}} = 19$ pixels, $\sigma_{\mathrm{L}} = 10$ pixels, $\mu_{\mathrm{H}} = 25$ pixels, $\sigma_{\mathrm{H}} = 9$ pixels. These values give some information about the font sizes of characters in this dataset, because most of the connected components in this dataset belong to text class.

Together with ASCII-text ground-truth, this dataset also contains pixel-based ground-truth for zones, text-lines, formulas, tables, and figures. These pixel-based ground-truth images are embedded in color-coded form, where red chan-

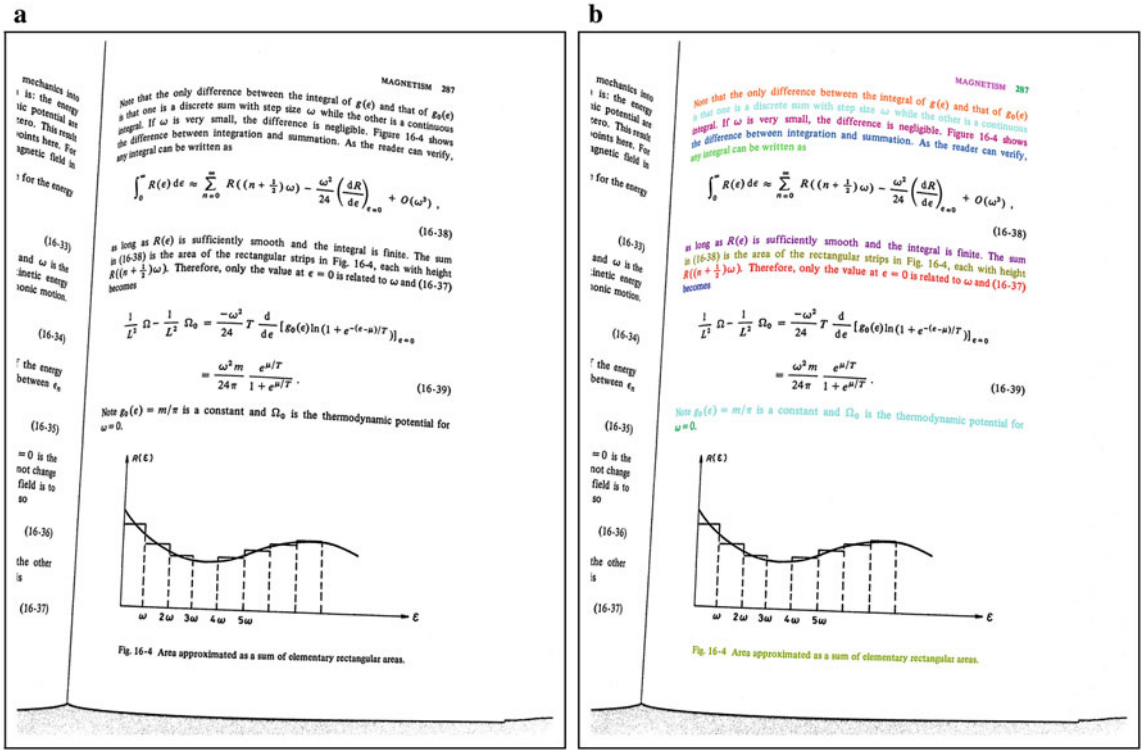

Fig. 15 An example image of DFKI-I (CBDAR 2007 dewarping contest) dataset [39] and its corresponding text-line-based ground-truth image. Note that non-text elements (equation, graphics) as well as partial text-lines from the neighboring page have been considered as "noise" in the ground-truth. **a** Original Image. **b** Labeled text-lines

nel contains zone class information, blue channel contains zone number (in reading order) information, and green channel contains text-line number information. Textual and non-textual border noise are marked as noise with black color. For the performance evaluation of text-lines segmentation algorithms, we generated text-line-based ground-truth images from the original ground-truth images. A text-lines-based ground-truth image contains labeling only for text-lines, and all the other foreground objects, like formulas, tables, and figures, are marked as noise with black color. We did it automatically by using original ground-truth information as follows: in an original ground-truth image, green color channel value is zero for all pixels except those that belong to text-lines. We marked all pixels, that contain zero value for green color channel, as noise. An example image and its corresponding text-lines-based ground-truth image is shown in Fig. 15.

5.2 Performance evaluation methodology

Performance evaluation of a text-line segmentation algorithm is based on vectorial performance evaluation metric that was presented by Shafait et al. [10]. One of the importance of these vectorial metric is that it not only represents one-to-one segmentation accuracy, but also represents most important classes of segmentation errors, such as over-, under-, and miss-segmentation.

Performance evaluation metrics are described as follows. Consider we have two segmented images, the ground-truth G and hypothesized segmentation H. We can compute a weighted bipartite graph called "pixel-correspondence graph" between G and H for evaluating the quality of the segmentation algorithm. Each node in G represents a text-line (*ground-truth component*), and each node in H represents a segmented text-line (*segmented component*). An edge is constructed between two nodes such that the weight of the edge equals the number of foreground pixels in the intersection of the regions covered by the two segments represented by the nodes. The matching between G and H is considered perfect if there is only one edge incident to each component of G or H, otherwise it is not perfect, i.e., each node in G or H may have multiple edges. The edge incident to a node is significant if the value of $w_i/P \geq t_r$ and $w_i \geq t_a$, where w_i is the edge-weight, P is the number of pixels corresponding to a node (segment), t_r is a relative threshold, and t_a is an absolute threshold. In practice, $t_r = 0.1$ and $t_a = 100$ are good choices for text-lines-based performance evaluation for typed-text document images [10]. We have also used same parameter values for the performance evaluation of our coupled snakelets and other text-line segmentation algorithms.

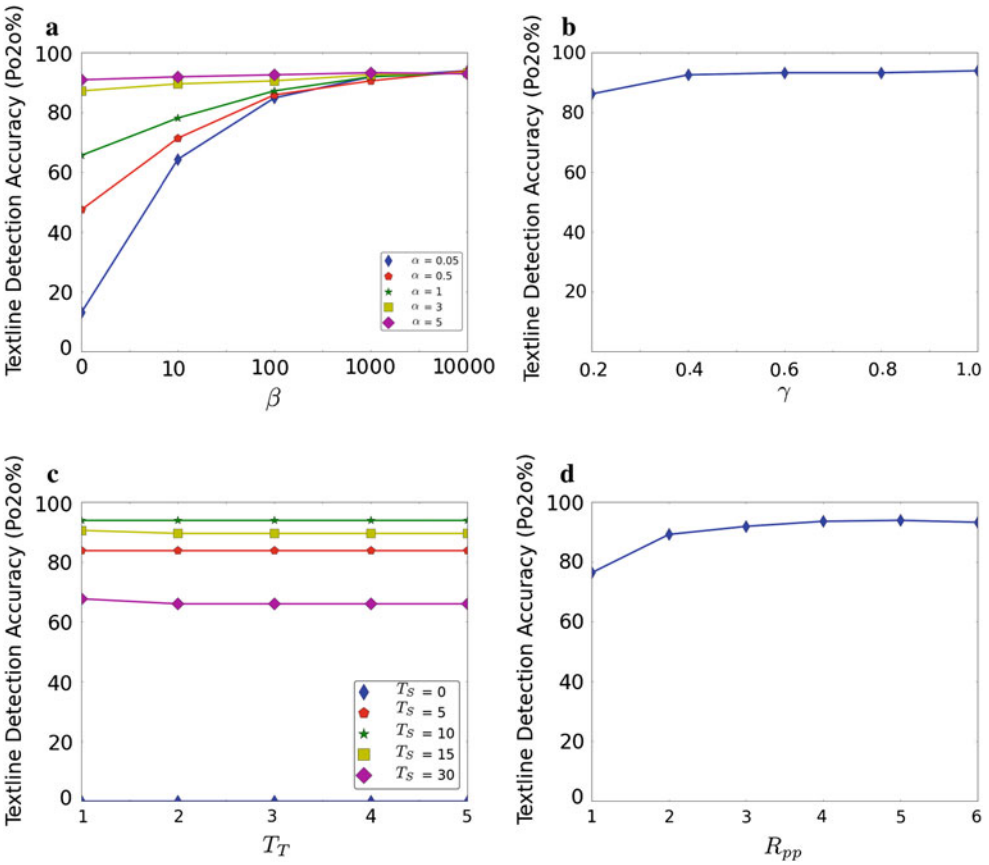

Fig. 16 Curled text-line segmentation accuracy (P_{o2o}) of our coupled snakelets-based algorithm for different values of free parameters, α, β, γ, T_S, T_T, and R_{pp}, on a subset of 11 images from DFKI-I (CBDAR 2007 dewarping contest) dataset. **a** Accuracy vs. α and β. **b** Accuracy vs. γ. **c** Accuracy vs. T_T and T_S. **d** Accuracy vs. R_{pp}

Let N_g represents total number of ground-truth components, and N_s represents total number of segmented components. Based on the above description, the performance evaluation metrics are:

- **Total correct segmentation** (N_{o2o})**:** the number of one-to-one matches between the ground-truth components and the segmented components. The one-to-one match accuracy is calculated by $P_{o2o} = N_{o2o}/N_g$.
- **Oversegmented components** (N_{ocomp})**:** the number of ground-truth lines having more than one significant edge. The percentage of oversegmented components is calculated by $P_{ocomp} = N_{ocomp}/N_g$.
- **Undersegmented components** (N_{ucomp})**:** the number of segmented lines having more than one significant edge. The percentage of undersegmented components is calculated by $P_{ucomp} = N_{ucomp}/N_g$.
- **Missed components** (N_{mcomp})**:** the number of ground-truth components that match the background in the hypothesized segmentation. The percentage of missed components is calculated by $P_{mcomp} = N_{mcomp}/N_g$.
- **Total oversegmentations** (N_{oseg})**:** the number of significant edges that ground-truth lines have minus the number of ground-truth lines.
- **Total undersegmentations** (N_{useg})**:** the number of significant edges that segmented lines have minus the number of segmented lines.
- **False alarms** (N_{falarm})**:** the number of components in the hypothesized segmentation that did not match any foreground component in the ground-truth segmentation.

5.3 Performance evaluation results

Our coupled snakelets-based curled text-line segmentation algorithm contains three free/tunable parameters (α, β, and γ) for the coupled snakelets estimation, and three parameters (T_S, T_T, and R_{pp}) for the post-processing step. All of these parameters have been explained in detail in Sect. 3. A brief description of these parameters are as follows. Parameters α and β are used to control snake's internal energy during deformation, and γ is used to control snake's external energy. The parameter α is usually set to a value like 0.05, 0.5, 5, etc. The parameter β is set to a small value when no stiffness is required and to a large value when high snake's stiffness is required during deformation steps (like our coupled snakelets model). The possible range of values for parameter γ

Table 1 Performance evaluation results of our coupled snakelets-based and other previously reported curled text-line segmentation algorithms as well as a straight text-line segmentation algorithm(Docstrum [6]) on binary camera-captured document images of DFKI-I (CBDAR 2007 dewarping contest) dataset [39] by using performance evaluation metrics [10]

Algorithm	Performance evaluation metrics[a]									
	N_g	N_s	N_{o2o}	N_{falarm}	N_{useg}	N_{oseg}	P_{ucomp} (%)	P_{ocomp} (%)	P_{mcomp} (%)	P_{o2o} (%)
Docstrum [6][b]	3,091	6,852	657	6,066	2,096	4,383	51.05	66.90	0	21.26
Nearest-Neighbor [19]	3,091	6,983	898	307,601	16	1,365	0.49	22.94	44.74	29.05
Neighbor modified [19][c]	3,091	3,256	2,780	42,15	102	208	3.17	6.05	0.03	89.93
Rule-based [22]	3,091	2,924	2,816	785	57	682	1.81	21.71	4.43	91.10
Baby-snakes [26]	3,091	3,371	2,707	13,199	117	294	2.91	5.79	0	87.58
Ridges-based [27]	3,091	3,115	2,771	2,183	110	144	3.30	4.40	0.29	89.65
Coupled snakelets	3,091	3,106	2,940	3,328	51	61	1.58	1.84	0	95.12

This dataset contains 102 document images captured using hand-held camera in an uncontrolled environment

[a] N_g ground-truth components, N_s segmented components, N_{o2o} one-to-one matched components, N_{falarm} false alarms, N_{useg} undersegmentations, N_{oseg}: oversegmentations, N_{ucomp} undersegmented components, $P_{ucomp} = N_{ucomp}/N_g$; N_{ocomp} oversegmented components, $P_{ocomp} = N_{ocomp}/N_g$, N_{mcomp} missed components, $P_{mcomp} = N_{mcomp}/N_g$, $P_{o2o} = N_{o2o}/N_g$

[b] Docstrum [6] is used for scanned document image segmentation with straight text-lines. Here, it is used for curled text-lines segmentation to show that: (i) straight text-lines algorithm is not directly application on camera-captured documents and (ii) the DFKI-I (CBDAR 2007 dewarping contest) dataset is challenging with respect to curled text-lines

[c] The original version of Gatos et al. [19] nearest-neighbor-based curled text-line detection algorithm and our proposed modification are described in Sect. 2

is in between 0 to 1. The parameter T_S is the slope threshold, T_T is the relative thickness threshold (with respect to the mean height of connected components in a document image H), and R_{pp} is the relative window size with respect to H. Experimental results show that the post-processing step does not require a very small value (like 0°) or a comparatively large value (like 45°) for parameter T_S. Similarly, the relative values for T_T and R_{pp} can be set in between 1 to 10.

For optimization of these parameters and showing their effects on text-line detection accuracy, we have evaluated our coupled snakelets-based curled text-line segmentation algorithm on 11 images from the DFKI-I (CBDAR 2007 dewarping contest) dataset (that start with name *dsc00*) for different values of these free parameters. The one-to-one text-line segmentation accuracy (P_{o2o}) of our algorithm for the different values of these free parameters is shown in Fig. 16. Here, we have adopted a sequential procedure for evaluating and optimizing the performance of our text-line detection method with respect to the different values of these parameters. In Fig. 16a, the text-line detection accuracy is shown for different values of α and β with empirically chosen values for other parameters ($\gamma = 1$, $T_S = 10°$, $T_T = 30$ pixels (absolute value), $R_{pp} = 150$ pixels (absolute value)). Similarly, in Fig. 16b, the text-line detection accuracy is represented for different values of γ with optimized values for $\alpha = 0.05$ and $\beta = 10,000$ (from Fig. 16a), and chosen values for other parameters ($T_S = 10°$, $T_T = 30$ pixels, $R_{pp} = 150$ pixels). Likewise, in Fig. 16c, the text-line detection accuracy is shown for different values of T_T and T_S with optimized values for $\alpha = 0.05$, $\beta = 10,000$, and $\gamma = 1$ (from Fig. 16a, b), and chosen value for $R_{pp} = 150$ pixels. Finally, in Fig. 16d, the text-line detection accuracy is shown for different values of R_{pp} with optimized values for others ($\alpha = 0.05$, $\beta = 10,000$ and $\gamma = 1$, $T_T = 1$ and $T_S = 10°$; from Fig. 16a–c).

From Fig. 16, we can conclude that the performance of our text-line segmentation method is not sensitive to the values of most of the free parameters (like α, γ, T_T and R_{pp}), except β and T_S. The optimized values of these parameters for a subset of 11 images from DFKI-I (CBDAR 2007 dewarping contest) dataset are as follows: $\alpha = 0.05$, $\beta = 1,000$, $\gamma = 1$, $T_S = 10°$, $T_T = 1$, and $R_{pp} = 5$.

We have compared the performance of our coupled snakelets-based curled text-line segmentation algorithm on the complete dataset of DFKI-I (CBDAR 2007 dewarping contest) dataset with other previously reported curled text-line segmentation algorithms: (i) nearest-neighbors (Gatos et al. [19]), (ii) baby-snakes (Bukhari et al. [26]), (iii) ridges detection (Bukhari et al. [27,28]), (iv) rule-based (Oliveria et al. [22]), (v) Docstrum [6]. In the literature review (Sect. 2), we also proposed a minor modification for the nearest-neighbor-based algorithm [19] by introducing the free parameter T. The average height of a connected components in DFKI-I (CBDAR 2007 dewarping contest) dataset is approximately equal to 20; therefore, we set $T = 20$ for the modified version of nearest-neighbor-based algorithm [19]. We also evaluated a straight text-line segmentation algorithm (Docstrum [6]) for curled text-line segmentation from camera-captured document images. Docstrum is one of the state-of-the-art page segmentation algorithm for scanned document images with straight text-lines. The main reason of

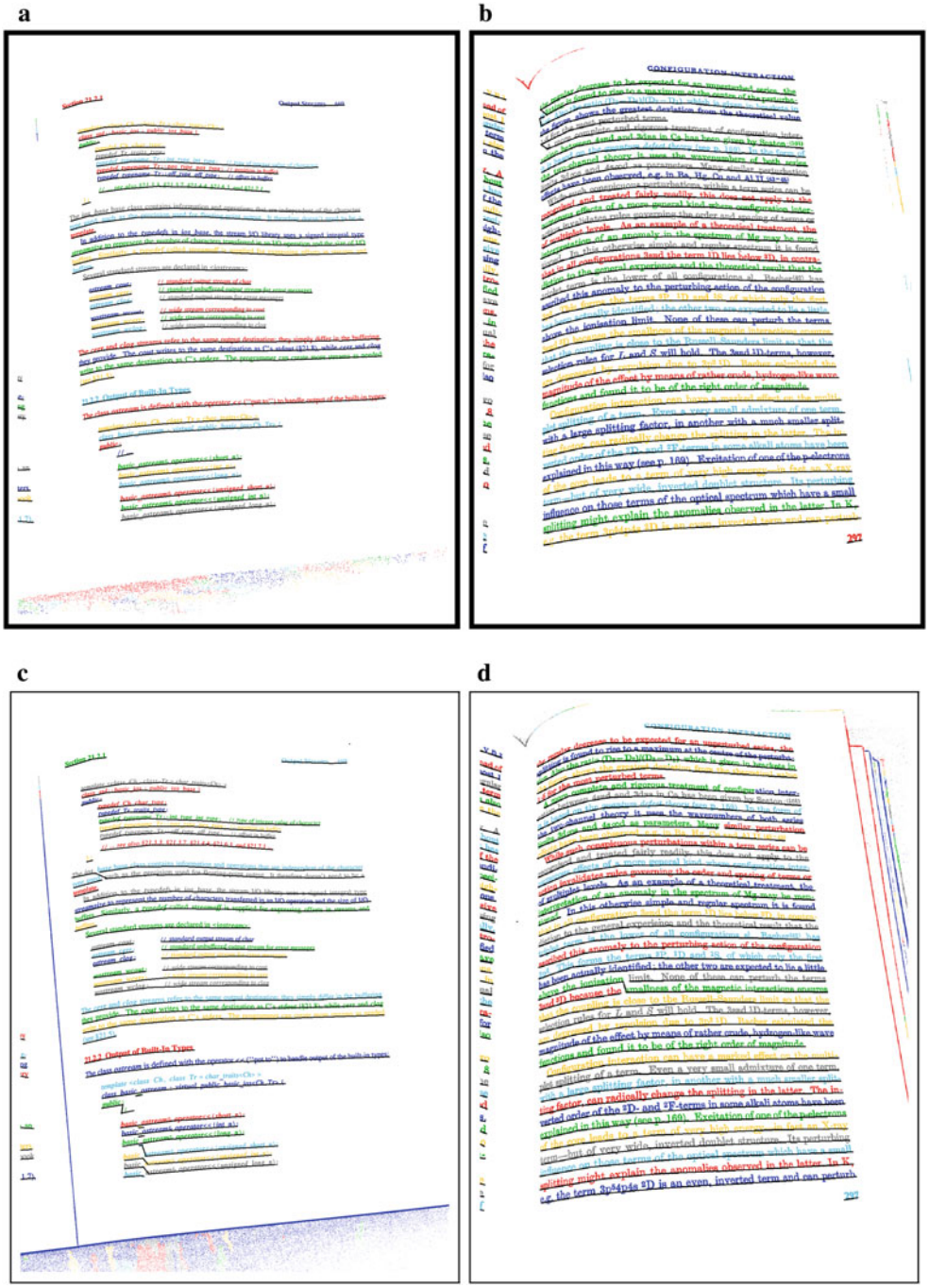

Fig. 17 Text-line segmentation failures: *Top Row* the largest number of text-line segmentation errors on some sample documents from the DFKI-I (CBDAR 2007 dewarping contest) dataset for our coupled snakelets algorithm. *Bottom Row* the corresponding results of modified version of nearest-neighbor-based curled text-line segmentation algorithm [19] for comparison. [*Note*: in order to highlight the segmentation results, the color-coded segmented text-lines are also *underlined* manually by *black line*]. **a** $P_{o2o} = 81.40\%$. **b** $P_{o2o} = 88.64\%$. **c** $P_{o2o} = 69.77\%$. **d** $P_{o2o} = 86.36\%$

including it here is to show that how challenging the dataset is, and that straight text-lines segmentation algorithms cannot be directly applied for curled document images. Performance evaluation results of all algorithms for DFKI-I (CBDAR 2007 dewarping contest) dataset are shown in Table 1.

Among all curled text-line segmentation algorithms that are shown in Table 1, our coupled snakelets algorithm achieved the highest percentage of one-to-one segmentation accuracy and the lowest percentages of oversegmentation and missed text-line errors. Our algorithm also achieved

the second lowest percentage of undersegmentation errors. Almost all of the algorithms have produced large numbers of false-alarm errors. In general, a large number of false alarms can be reduced by using an appropriate pre-processing or post-processing step, for example a page boundary detection method [2,40] can help in removing textual and non-textual border noise.

A few sample documents from the DFKI-I (CBDAR 2007 dewarping contest) dataset having the largest number of text-line segmentation errors for our coupled snakelets algorithm are shown in the top row of Fig. 17, and for comparison, the corresponding results of modified nearest-neighbor-based algorithm [19] are shown in the bottom row of Fig. 17. Even in these examples, our algorithm has overall performed better than modified nearest-neighbor-based algorithm [19]. The document image in Fig. 17a contains total 43 text-lines, and our method detected 35 of them correctly. In this example, most of the errors belong to oversegmentation category. These oversegmentation errors mainly occur because of big gaps between words within some text-lines, which can be seen in the middle of the page. The document image of Fig. 17b contains very small gaps between text-lines, resulting in undersegmentation errors. For this image, our coupled snakelets method detected 39 text-lines correctly out of 44 text-lines and produced some undersegmentation errors, which can be seen in the top area of the document image in Fig.17b.

The average size of a document image in DFKI-I (CBDAR 2007 dewarping contest) dataset is around 8 Mega-pixels, and the average size of a character is this dataset is 19 pixels wide (with a standard deviation of 11 pixels) and 25 pixels tall (with a standard deviation of 9 pixels). The execution time of our text-line detection method is directly proportional to the size and the number of connected components in a document image. The main problem with active contour model is that it takes large amount of execution time because of a large number of deformation cycles. Our coupled snakelets model processes many snakes sequentially, and therefore it also takes a large amount of processing time. On weighted average with respect to the number of connected components, our algorithm takes around 38 min per page with text-line detection accuracy of around 95%. We have implemented the code using Python programming language without using any Python-specific and/or active contour–specific optimization techniques. The execution time can be reduced by using these types of optimization techniques, which is one of our future research goals. Here, we have tested a basic strategy for reducing the execution time of our method such that a document image is downscaled for coupled snakelets calculation and then the estimated snakelets are upscaled proportionally for text-lines labeling step. Figure 18 shows the text-line detection accuracy of our coupled snakelets method and the corresponding average execution time for different size of

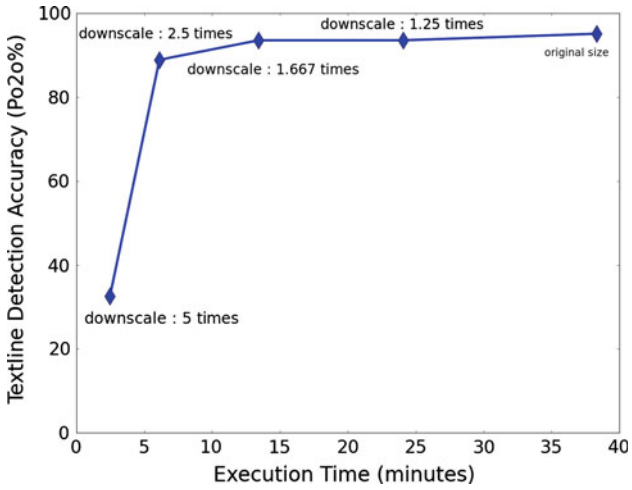

Fig. 18 The execution time and the corresponding text-line detection accuracy of our coupled snakelets method for different downscaling factors on DFKI-I (CBDAR 2007 dewarping contest) dataset

downscaled images for DFKI-I (CBDAR 2007 dewarping contest) dataset. This process also demonstrates how well our text-line segmentation algorithm can perform for small character sizes. We have achieved around 3 times speed up gain (i.e., 13 min per page) with around 1% reduction in text-line detection accuracy (i.e., 94%) for around 2 times image downscaling factor. It also shows that our algorithm can work gracefully up to a minimum character size of around 10 pixels wide and around 12 pixels tall.

6 Conclusion

Hand-held camera-captured document images usually contain warped/curled text-lines because of geometric and/or perspective distortions. In this paper, we introduced a novel curled text-line segmentation algorithm by adapting active contour (snake) [29]. We refer to our adapted active contour (snake) model for text-line segmentation as *coupled snakelets*. Our algorithm uses only top and bottom points of connected components within a document image for detecting text-lines. It jointly estimates a local pair of x-line and baseline on each connected component using top and bottom points, and then each group of overlapping and/or touching pairs of x-line and baseline is considered as a segmented text-line. We used DFKI-I (CBDAR 2007 dewarping contest) dataset [21] for performance evaluation and compared our results with other state-of-the-art approaches: (i) nearest-neighbors—original (Gatos et al. [19]) and our proposed modified version, (ii) baby-snakes (Bukhari et al [26]), (iii) ridges detection (Bukhari et al. [27,28]), (iv) rule-based (Oliveria et al. [22]), and (v) Docstrum [6]. Our algorithm is less sensitive to a high degree of curl and skew and produces a less number of over- and undersegmentation errors when compared to other state-of-the-art curled text-line segmen-

tation methods. Unlike previous approaches, our algorithm performs text-lines segmentation and their x-line and baseline pairs estimation simultaneously that results in improved segmentation with better estimation of x-lines and baseline than other approaches. The performance evaluation results are shown in Table 1. Our algorithms achieve the highest one-to-one text-line segmentation accuracy when compared to other methods. It also yields the lowest oversegmentation and missed text-lines errors, and a smaller number of undersegmentation errors. Our method contains 6 free/tunable parameters. Most of these parameters are non-sensitive with respect to the performance the presented method.

References

1. Bukhari, S.S., Shafait, F., Breuel, T.M.: Adaptive binarization of unconstrained hand-held camera-captured document images. J. Univers. Comput. Sci. **15**(18), 3343–3363 (2009)
2. Shafait, F., van Beusekom, J., Keysers, D., Breuel, T.: Document cleanup using page frame detection. Int. J. Doc. Anal. Recognit. **11**, 81–96 (2008)
3. Kao, C.-H., Don, H.-S.: Skew detection of document images using line structural information. In: International Conference on Information Technology and Applications, vol. 1, pp. 704–715. Los Alamitos (2005)
4. Arvind, K., Kumar, J., Ramakrishnan A.: Entropy based skew correction of document images. In: Pattern Recognition and Machine Intelligence, vol. 4815 of Lecture Notes in Computer Science, pp. 495–502 (2007)
5. van Beusekom, J., Shafait, F., Breuel, T.M.: Combined orientation and skew detection using geometric text-line modeling. Int. J. Doc. Anal. Recognit. **13**(2), 79–92 (2010)
6. O'Gorman, L.: The document spectrum for page layout analysis. IEEE Trans. Pattern Anal. Mach. Intell. **15**(11), 1162–1173 (1993)
7. Marti, U.-V., Bunke, H.: Text line segmentation and word recognition in a system for general writer independent handwriting recognition, In: Proceedings of the 6th International Conference on Document Analysis and Recognition, pp. 159–163. Los Alamitos (2001)
8. Bukhari, S.S., Shafait, F., Breuel, T.M.: Dewarping of document images using coupled-snakes. In: Proceedings of Third International Workshop on Camera-Based Document Analysis and Recognition, pp. 34–41. Barcelona (2009)
9. Tan, C.L., Zhang, L., Zhang, Z., Xia, T.: Restoring warped document images through 3D shape modeling. IEEE Trans. Pattern Anal. Mach. Intell. **28**(2), 195–208 (2006)
10. Shafait, F., Keysers, D., Breuel, T.M.: Performance evaluation and benchmarking of six page segmentation algorithms. IEEE Trans. Pattern Anal. Mach. Intell. **30**(6), 941–954 (2008)
11. Glauberman, M.H.: Character recognition for business machines. Electronics **29**, 132–136 (1956)
12. Nagy, G., Seth, S., Viswanathan, M.: A prototype document image analysis system for technical journals. Computer **25**(7), 10–22 (1992)
13. Fletcher, L.A., Kasturi, R.: A robust algorithm for text string separation from mixed text/graphics images. IEEE Trans. Pattern Anal. Mach. Intell. **10**(6), 910–918 (1988)
14. Wong, K.Y., Casey, R.G., Wahl, F.M.: Document analysis system. IBM J. Res. Dev. **26**(6), 647–656 (1982)
15. Breuel, T.M.: Robust least square baseline finding using a branch and bound algorithm. In: Proceedings SPIE Document Recognition and Retrieval IX, pp. 20–27. San Jose (2002)
16. Liang, J., Doermann, D., Li, H.: Camera-based analysis of text and documents: a survey. Int. J. Doc. Anal. Recognit. **7**, 84–104 (2005)
17. Zhang, Z., Tan, C.L.: Recovery of distorted document images from bound volumes. In: Proceedings of the International Conference on Document Analysis and Recognition, pp. 429–433. (2001)
18. Lu, S., Tan, C.L.: The restoration of camera documents through image segmentation. In: 7th IAPR Workshop on Document Analysis Systems, pp. 484–495 (2006)
19. Gatos, B., Pratikakis, I., Ntirogiannis, K.: Segmentation based recovery of arbitrarily warped document images. In: Proceedings 9th International Conference on Document Analysis and Recognition, pp. 989–993. Curitiba (2007)
20. Fu, B., Wu, M., Li, R., Li, W., Xu, Z.: A model-based dewarping method using text line detection. In: 2nd International Workshop on Camera Based Document Analysis and Recognition (2007)
21. Shafait, F., Breuel, T.M.: Document image dewarping contest. In: 2nd International Workshop on Camera-Based Document Analysis and Recognition, Curitiba (2007)
22. Oliveira, D.M., Lins, R.D., Torreao G., Fan J., Thielo M.: A new method for text-line segmentation for warped document. In: Proceedings of International Conference on Image Analysis and Recognition, pp. 398–408. Povoa de Varzim (2010)
23. Liang, J., DeMenthon, D., Doermann, D.: Geometric rectification of camera-captured document images. IEEE Trans. Pattern Anal. Mach. Intell. **30**, 591–605 (2008)
24. Goto, H., Aso, H.: Extracting curved text lines using local linearity of the text line. Int. J. Doc. Anal. Recognit. **2**, 111–119 (1999)
25. Loo, P.K., Tan, C.L.: Word and sentence extraction using irregular pyramid. In: Document Analysis Systems V, vol. 2423 of Lecture Notes in Computer Science, pp. 307–318. Springer, Berlin (2002)
26. Bukhari, S.S., Shafait, F., Breuel, T.M.: Segmentation of curled textlines using active contours. In: Proceedings of the 8th IAPR Workshop on Document Analysis Systems, pp. 270–277. Nara (2008)
27. Bukhari, S.S., Shafait, F., Breuel, T.M.: Ridges based curled textline region detection from grayscale camera-captured document images. In: Proceedings of the 13th International Conference on Computer Analysis of Images and Patterns, vol. 5702/2009 of Lecture Notes in Computer Science, pp. 173–180. Muenster (2009)
28. Bukhari, S.S., Shafait, F., Breuel, T.M.: Curled textline information extraction from grayscale camera-captured document images, In: Proceedings of the 13th International Conference on Image Processing, Cairo (2009)
29. Kass, M., Witkin, A., Terzopoulos, D.: Snakes: active contour models. Int. J. Comput. Vis. **1**(4), 1162–1173 (1988)
30. Bukhari, S.S., Shafait, F. Breuel, T.M.: Coupled snakelet model for curled textline segmentation of camera-captured document images. In: Proceedings of the 10th International Conference on Document Analysis and Recognition, pp. 61–65. Barcelona (2009)
31. Strouthopoulos, C., Papamarkos, N., Chamzas, C.: Identification of text-only areas in mixed-type documents. Eng. Appl. Artif. Intell. **10**(4), 387–401 (1997)
32. Hsieh, C.T., Lai, E., Wang, Y.C.: An effective algorithm for fingerprint image enhancement based on wavelet transform. Pattern Recognit. **36**(2), 302–312 (2003)
33. Xu, C., Prince, J.L.: Snakes, shapes, and gradient vector flow. In: IEEE Transaction of Image Processing, pp. 359–369 (1998)
34. Gunn, S.R., Nixon, M.S.: A robust snake implementation; a dual active contour. IEEE Trans. Pattern Anal. Mach. Intell **19**(1), 63–68 (1997)
35. Hohnhaeuser, B., Hommel, G.: 3D pose estimation using coupled snakes. J. WSCG **12**(1–3), 1213–6972 (2003)

36. Wang, Q., Ronneberger, O., Schulze, E., Baumeister, R., Burkhardt, H.: Using lateral coupled snakes for modeling the contours of worms. In: Pattern Recognition Lecture Notes in Computer Science, vol. 5748, pp. 542–551. (2009)
37. Ulges, A., Lampert, C., Breuel, T.: Document image dewarping using robust estimation of curled text lines. In: Proceedings of the Eighth International Conference on Document Analysis and Recognition, pp. 1001–1005 (2005)
38. Breuel, T.M.: Segmentation of handprinted letter strings using a dynamic programming algorithm. In: Proceedings of the Sixth International Conference on Document Analysis and Recognition pp. 821–826 (2001)
39. Shafait, F., Breuel, T.M.: Document image dewarping contest. In: 2nd International Workshop on Camera-Based Document Analysis and Recognition, pp. 181–188. Curitiba (2007)
40. Stamatopoulos, N., Gatos, B., Kesidis, A.: Automatic borders detection of camera document images. In: Proceedings of Second International Workshop on Camera-Based Document Analysis and Recognition, pp. 71–78. Curitiba (2007)