Segmentation of Curled Textlines using Active Contours

Syed Saqib Bukhari[1], Faisal Shafait[2], Thomas M. Breuel[1]

Image Understanding and Pattern Recognition (IUPR) Research Group
[1]Department of Computer Science, Technical University of Kaiserslautern, Germany
[2]German Research Center for Artificial Intelligence (DFKI), Kaiserslautern, Germany
{bukhari, tmb}@informatik.uni-kl.de,{faisal.shafait}@dfki.de,

Abstract

Segmentation of curled textlines from warped document images is one of the major issues in document image dewarping. Most of the curled textlines segmentation algorithms present in the literature today are sensitive to the degree of curl, direction of curl, and spacing between adjacent lines. We present a new algorithm for curled textline segmentation which is robust to above mentioned problems at the expense of high execution time. We will demonstrate this insensitivity in a performance evaluation section. Our approach is based on the state-of-the-art image segmentation technique: Active Contour Model (Snake) with the novel idea of several baby snakes and their convergence in a vertical direction only. Experiment on publically available CBDAR 2007 document image dewarping contest dataset shows our textline segmentation algorithm accuracy of 97.96%.

1 Introduction

Document images are usually captured through a scanner. Capturing document images using a camera in an uncontrolled environment is a fast, easy and less expensive alternative as compared to capturing document images with a scanner. Unfortunately, this capturing mechanism produces distorted document images. Non-linear warping is one of the major distortions. It badly affects document readability and reduces OCR accuracy. We will refer to these types of images as curled or warped document images.

A number of techniques have been reported in literature for warped document restoration or dewarping. These can be classified into two main groups based on the method of document capture to which they are applied: (i) approaches in which specialized hardware is needed for recovering the 3D shape structure of warped documents [3, 4, 21], (ii) approaches in which no auxiliary hardware is needed, and documents are captured with a hand-held camera in an uncontrolled environment [7, 9, 10, 14, 15, 16, 22]. The approaches requiring specialized hardware have not come to widespread use due to the restrictions they impose on document capture. Dewarping approaches for documents captured with a hand-held camera can be further divided into two categories: (a) approaches based on the document geometry [14, 16], (b) approaches based on curled textline segmentation [7, 9, 10, 13, 15, 22]. Accurate segmentation of curled textline is essential for good performance of the later dewarping approaches.

The textline detection technique described in [7] proceeds as follows. First, characters are combined based on [11] and are referred to as sub lines. A mid point of each sub line is calculated and the image is thinned. For each mid point, left and right neighboring points are searched in the direction of previously calculated slope. A curve fitting algorithm is applied for each set of points. The algorithm [7] works well only for moderate slope in document images. The drawback of the algorithm [7] is that a single outlier point in a thinned image leads to the wrong slope direction and starts grouping wrong neighborhoods.

The textline detection technique described in [15] uses morphological operations to calculate the top point, bottom point, and vertical stroke of each character. Neighboring points are searched for each point within a specified area (based on the function of average height of vertical strokes and the slope of the already grouped points). This approach [15] has two problems. Firstly, the selection of the structuring elements for the calculation of top and bottom points are based on the document skew (top-left to bottom-right or bottom-left to top-right). This means it can't handle curled documents which have different types of skew. Secondly, the grouping of neighboring points goes in the wrong direction if the distance between words is greater than the distance between the base and top lines.

The textline detection approach described in [10] performs horizontal smoothening [23] to combine characters into words. Words are then selected in a top down scanning fashion. For each word having bounding box (x_1,x_2,y_1,y_2), neighboring connected components (in the right direction, satisfying condition: [y_1,y_2] intersection [$y_1^{'}$,$y_2^{'}$] is not equal to null) are searched. A Component with the smallest distance is grouped with the selected word. This approach [10] works well only if the curl is moderate, otherwise it combines the components that are present in different top or bottom lines.

The textline detection approach in [9] calculates words and textline by using the modified “box hand” [25, 20] approach. For detecting words, quadrilaterals of equal width and height are attached to the left and right side of each character. Letters having overlapping hands are combined to form words. A Similar approach is used for combining words into textline, but the slope of the quadrilateral is based on the slope of the word. The problem with this approach [9] is that it does not make sure that the hands of each connected component touch the others even in a moderate line curl. This approach [9] does not work on documents with high curl.

The textline detection technique proposed in [13] based on the “Level Set” algorithm is also one of the popular image segmentation techniques but different from the “Active Contour Model: Snake” algorithm. In [13] the level set algorithm is applied to handwritten documents which can’t be directly applied to printed curled textline documents.

The dewarping technique proposed in [22] is based on local textline approximation using RAST [2]. It works on an assumption that the line spacing is the same between all lines. Therefore finding paragraphs is one of the preprocessing steps in [22]. There is no information about the global approximation of textlines.

Most of the above approaches are highly sensitive to the directions and degrees of page curl. Our approach can handle high degrees of variable curls. It is based on the state-of-the art image segmentation technique, ***Active Contour Model (Snake)*** [12].

The rest of this paper is organized as follows: Section 2 deals with the technical details of active contour models (Snakes), novelty of our algorithm and its description. Section 3 highlights the performance evaluation and results. Section 4 comprises the discussion about future goals.

2 Textline Segmentation using Baby Snakes

2.1 Active Contour Model: Snakes

Active contour models [12], are curves which are defined via points over an image and can slither under the influence of internal and external energies. Internal energy tries to keep the curve points together and external energy tries to move the curve points toward the boundary of the image or other desired features/object. These energies are defined in such a way that move the curve continuously toward the object and at the end curve will align itself over the object. The behavior of the active contour model is dynamic because it always minimizes its energy function. The active contour model is also called ”Snake” because of the way it slithers while minimizing its energy. Snakes are used in both computer vision and image processing fields for edge-detection, shape modeling, segmentation, and motion tracking.

There are two general types of active contours models: Parametric active contour [12] and Geometric active contour [5]. We will use the parametric active contour model in our algorithm.

2.1.1 Parametric Snake Model

Parametric active contour models were introduced by [12]. In the parametric active contour model, snake is a curve $S(s) = [x(s), y(s)]$, where $s \in [0, 1]$, that moves through the spatial domain of an image to minimize the energy function:

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

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

E is the energy function (i.e. the sum of the snake’s internal and external energies). α and β are weight parameters that control the snake’s tension and rigidity or stiffness, respectively, and $S^{'}(s)$ and $S^{''}(s)$ denote the first and second derivatives of $S(s)$ with respect to s. The snake that minimizes E must satisfy the Euler equation [6]:

$$\alpha S^{''}(s) - \beta S^{''''}(s) - \partial/\partial_s E_{ext} = 0 \tag{3}$$

Equation 3 yields two independent Euler equations:

$$\alpha x^{''}(s) - \beta x^{''''}(s) - \partial/\partial_x E_{ext} = 0 \tag{4}$$

$$\alpha y^{''}(s) - \beta y^{''''}(s) - \partial/\partial_y E_{ext} = 0 \tag{5}$$

Let, $\partial/\partial_x E_{ext} = f_x$ and $\partial/\partial_y E_{ext} = f_y$

The external energy function (or external force) f_i is derived from the image so that it has smaller values at the image object. Traditional external forces are ’Gradient’ and ’Gradient of Gaussian’, but they have two main problems. Firstly, the range of these external forces is very limited and only exist near the object boundary. Therefore snake should be initialized near the object otherwise it will not converge towards the object. Secondly, at the boundary of concavities these forces are parallel and in opposite directions prevent the snake from moving inside the concavity. *Gradient Vector Flow (GVF)* [24] is another type of external force

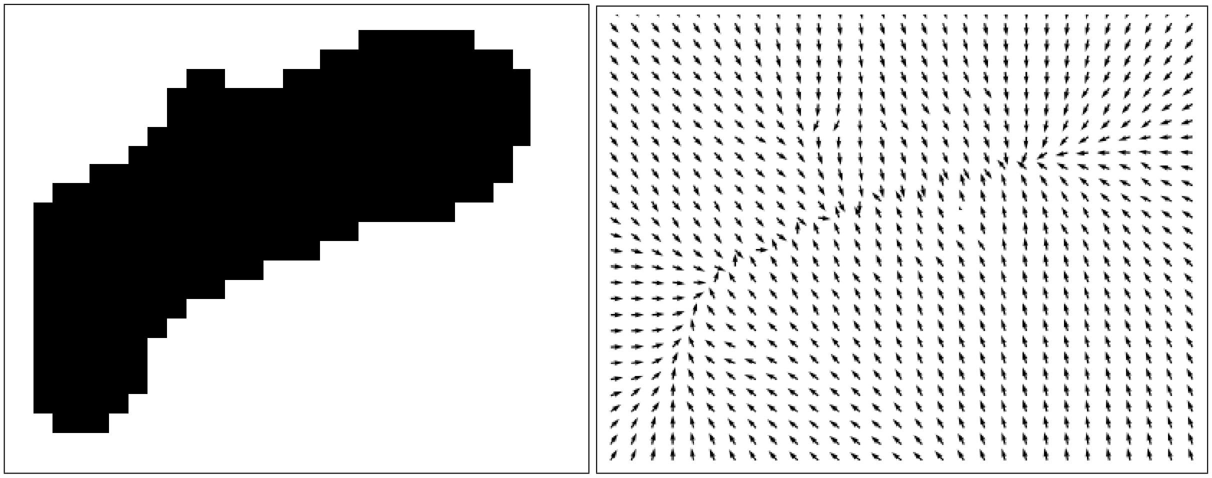


Figure 1: (a) Smeared word block. (b) GVF external forces of smeared word block is presented in a vector form.

that solves both the above problems of traditional external forces. It is calculated as a diffusion of gradient vectors of an object. We will use GVF as an external force in our algorithm because of its advantages over gradient and gradient of Gaussian.

2.2 Baby Snakes

Snake is used for detecting the edge boundary of object in an image. It is a closed curve of points and usually initialized around the edge map of the image object. External forces of the edge map cause the snake to move towards the edge map and internal forces keep the snake's points together, while updating horizontal and vertical components of the snake's points. At the end it encloses the edge boundary of an object. Our approach using snakes in textline segmentation is novel in four respects:

1. **Concept of straight line snake :** Closed curve snakes can't be used in our situation due to the topology of textlines. Here we introduce the concept of straight line snakes initialized over each connected components. We call them "baby Snakes" (see Section 2.3.3).

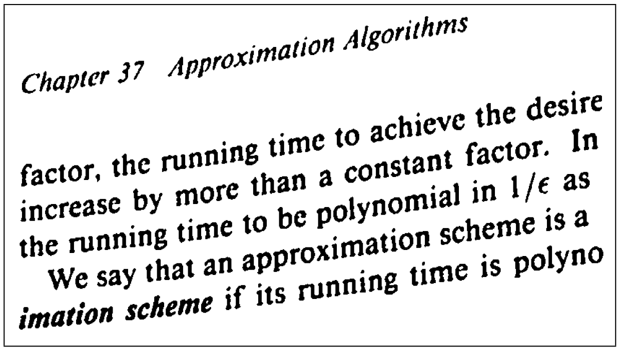
2. **Slop-aligned baby snakes :** We want to converge the straight line baby snakes in such a way that causes the adjacent baby snakes to stick to each other or to the adjacent connected component that help in textline segmentation. For straight textlines, it would be sufficient to initialize a horizontal snake at each connected component. Page curl distortion introduces a slope in each word depending on the amount of page curt at that position. Therefore, we initialize slope-aligned baby snakes over each connected component (see Section 2.3.3).

3. **Updating only vertical components of snake's points :** If we update both the horizontal and the vertical components of a straight line snake's points, it will converge to a single point after some iterations. Therefore, we update only the vertical components of a snake's points during each iteration through internal and external forces.

4. **Calculating external forces from smeared image object:** Usually external forces are calculated from an edge map of the image [12, 24]. Our experiments showed that using the smeared image for computing GVF instead of an edge map gave more stable results for our purpose. Figure 1a shows a smeared word. GVF external force vectors of this image are presented in Figure 1b. Notice the direction of all vectors; they are pointing towards the line passing through the center of the smeared word. This property ensures the joining of the slope-adjacent snakes on the left and right of the image component.

2.3 Textline Segmentation Algorithm

In our approach, slope-aligned baby snakes (parametric active contour models) are initialized over each connected component of a smeared document image. Only vertical components of baby snakes are deformed using GVF as an external snake force. Due to this deformation, slope-aligned baby snakes are joined together which results in segmented textlines. The basic steps of the algorithm are given below:



Chapter 37 Approximation Algorithms

factor, the running time to achieve the desire
increase by more than a constant factor. In
the running time to be polynomial in 1/ε as
We say that an approximation scheme is a
imation scheme if its running time is polyno

Figure 2: Input-image: A part of camera-captured document image is used for the demonstration of steps of algorithm.

1. Image smearing
2. Finding connected components
3. Initialization of slope-aligned baby snakes
4. Transformation of baby snakes into father-chains.
5. Extraction of textlines

Each step is discussed in detail in the following subsections.

2.3.1 Image Smearing:

The smearing technique is applied to join adjacent letters into words. These words or smeared components are candidates for baby snake initialization. The smearing approach is based on a combination of binary morphological operations, i.e dilation of an image followed by hole-filling and then erosion. Figure 2 and 3 show a document image sample and its corresponding smeared image respectively.

2.3.2 Noise Removal

Connected components of the smeared document image are calculated (Figure 3). Very small components (e.g. punctuation marks) or very big components (e.g. noisy page borders in camera captured documents) are considered as noise. These noisy components are not taken into account for baby snake initialization. Small noisy components may be part of a textline (some small noisy components are shown using blue circles in Figure 3). All noisy components are removed from the smeared document image. A record of small noisy components is maintained. In the last step of this algorithm we will check the association of these small noisy components with the segmented textlines using minimum distance criteria. The given thresholding criterion is used for finding noisy components:

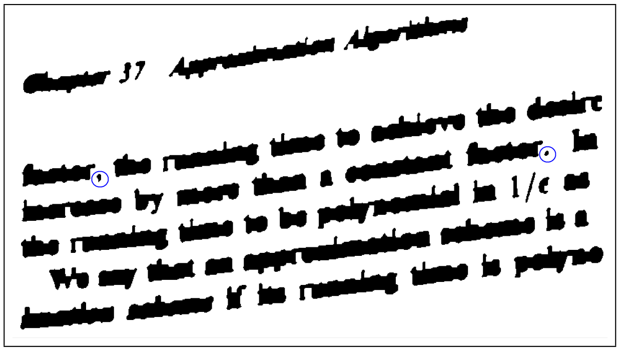


Figure 3: Result of image smearing. Small noisy components are marked with blue circles.

Let,
CC: set of connected components.
FCC: set of filtered connected components.
SNC: set of small noisy components.
LNC: set of large noisy components.

$$\text{SNC} = \{ \text{ x} \in \text{CC : width(x)} < \text{median(width}\{\text{CC}\}).a \text{ and height(x)} < \text{median(height}\{\text{CC}\}).b \ \}$$

$$\text{LNC} = \{ \text{ x} \in \text{CC : height(x)} > \text{document-height}.c \text{ or perimeter(x)} > \text{document-perimeter}.d \ \}$$

$$\text{FCC = CC - (SNC + LNC)}$$

Where a, b, c and d are the free parameters and their values are set according to the desired thresholding. In our case their values are 0.25, 0.25, 0.10 and 0.25 respectively.

2.3.3 Slope-aligned baby snakes initialization

Baby snakes are initialized over each connected component in such a way that each snake has the same slope as the component and pass through the middle of a component with some extra length on both sides of a component. The algorithm for the initialization of baby snakes is given below:
For each connected component repeat the following steps:

1. Decide the length of a snake based on a function of the component's width, average width of components and minimum and maximum widths among all components.

   Length = ComponentWidth + ExtraLength

   where,

   ExtraLength = AverageWidth.(1+((ComponentWidth-minWidth)/(maxWidth-minWidth)))

2. Initialize horizontal top and bottom straight line snakes' points for a component.

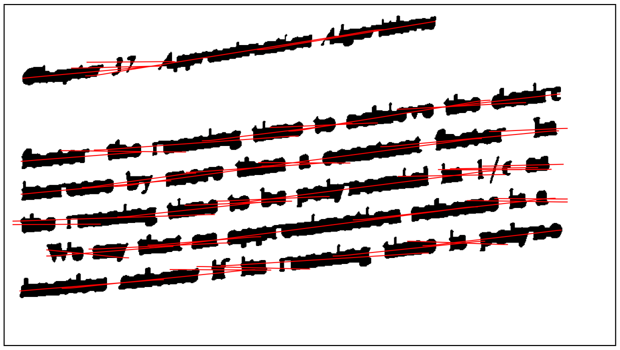


Figure 4: Slope-aligned horizontal Initial baby snakes over each connected components.

3. Compute gradient vector flow (GVF) of a component.
4. converge both snakes and update their y coordinates in each iteration using the active contour convergence approach with GVF as an external force and high stiffness parameter(β).
5. Approximate top and bottom lines from converged top and bottom snakes' points and find their slopes and y-intercepts.
6. Compute average straight line snake based on average of y-intercepts and weighted average of slopes of top and bottom lines. Average slope can be calculated as:

   $$\text{AverageSlope} = w_1.\text{SlopeOfTopLine} + w_2.\text{SlopeOfBottomLine}$$

   Where w_1 and w_2 are equal to 0.5 if both slopes are equal. $w_1 = 0.75$ and $w_2 = 0.25$ if SlopeOfTopLine $<$ SlopeOfBottomLine. $w_1 = 0.25$ and $w_2 = 0.75$ if SlopeOfTopLine $>$ SlopeOfBottomLine.

Slope-aligned baby snakes are presented in Figure 4.

2.3.4 Transformation of baby snakes into father-chains

In the previous step, baby snakes are initialized for each component. A mechanism is needed that joins together each baby snake with its slope-adjacent left and right neighborhood components and/or baby snakes making one connected chain for each textline. We call such a connected chain as father-chain. This is achieved by calculating the GVF of the smeared image (smeared image is shown in Figure 3) and allows all baby snakes to converge with a very low stiffness parameter (β) in the vertical direction only. This GVF external force with a very low stiffness parameter (β) allows each baby snake to converge towards its left and right (slope-adjacent) neighborhood components and/or baby snakes. Converged baby snakes are shown in Figure 5. For all baby snakes' points black pixels are drawn on the smeared image. An example result is shown in Figure 6. Each connected chain of components, which is shown in Figure 6, is referred to as father-chain.

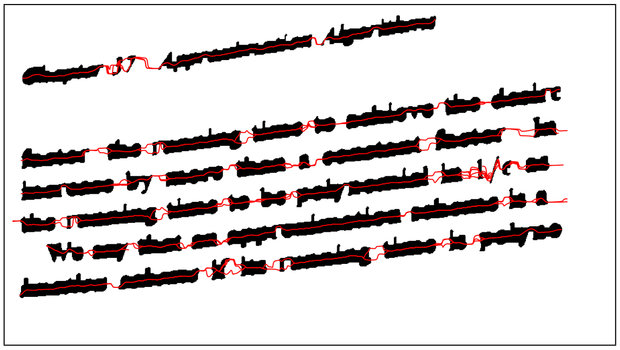


Figure 5: Baby snakes are converged by using an active contour convergence approach with GVF as an external force.

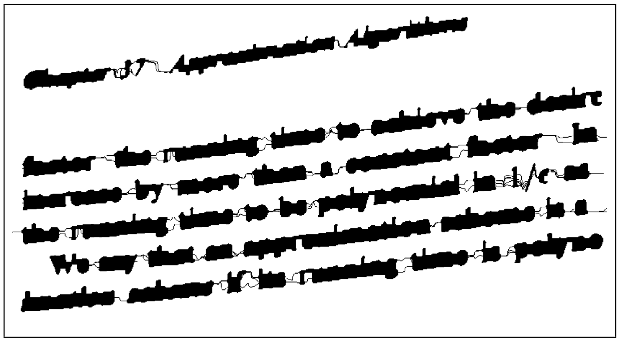


Figure 6: Father-chains: Black pixels are drawn in place of all baby snakes' points on a smeared image.

2.3.5 Textline extraction

It is clearly visible in Figure 6 that each father-chain is composed of all the components of a single textline. Father-chains are extracted using a connected component algorithm. The result is shown in Figure 7 (different connected components are represented by different colors).

All the pixels in the original document image (Figure 2) are compared with father-chains and the corresponding value of father-chains are assigned to them. This comparison results in segmented textlines shown in Figure 8 with different colors.

It is noticeable in Figure 8 that some small components (i.e. ',' and '.' in lines 2 and 3 respectively, represented by blue circles) are not a part of segmented textlines. This is because they were counted as noisy components in step 2 of the algorithm and were removed from the smeared image. The record of their locations is already maintained in

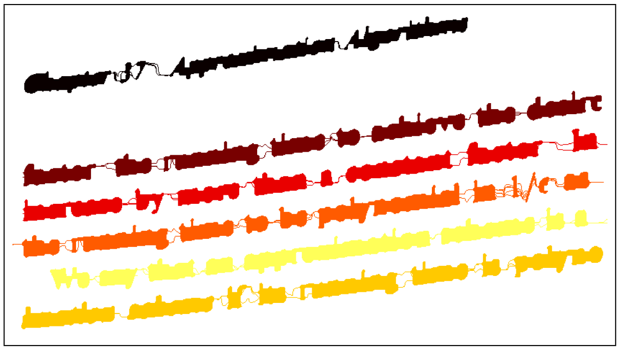


Figure 7: Segmented father-chains are represented by different colors.

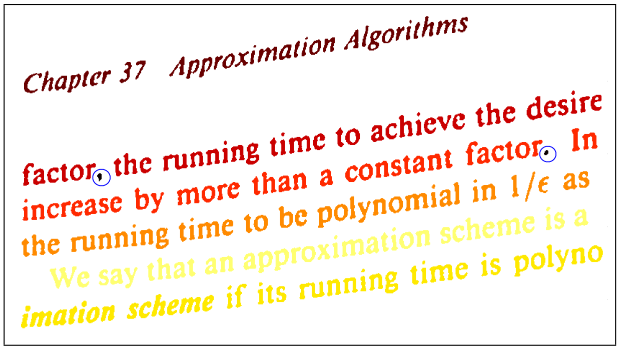

Chapter 37 Approximation Algorithms
factor, the running time to achieve the desire
increase by more than a constant factor. In
the running time to be polynomial in 1/ϵ as
We say that an approximation scheme is a
imation scheme if its running time is polyno

Figure 8: Different colors represent segmented textlines. Some small connected components, which are considered as 'noise', are marked with blue circles. These components are not part of any segmented textlines.

step 2. Minimum distance criteria (based on a specified area equal to the median area of FCC (Filtered connected components), calculated in step 1) is used to find the association of these noisy components with segmented textlines. The Final output of an algorithm is presented in Figure 9

3 Performance Evaluation and Results

To demonstrate the performance of our algorithm on real-world documents, we evaluated it on the data set used in the CBDAR 2007 document image dewarping contest [17]. This data set consists of 102 documents (approximately 3020 textlines) captured with a hand held camera in an uncontrolled environment. The ground truth images are given in a color coded format i.e. different textlines are represented by different colors. Snake-segmentation results are also represented in the same format.

Performance evaluation of our algorithm is based on the metrics which are defined in [8, 18, 19].

Descriptions of performance evaluation metrics for textline segmentation based on [18, 19] are as follows. Consider we have two segmented images, the ground truth G and hypothesized snake-segmentation H. We can compute a weighted bipartite graph called "pixel-correspondence graph" [1] between G and H for evaluating the quality of the segmentation algorithm. Each node in G or H represents a 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 perfect if there is only one edge incident on 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 on a node is significant if the value of (w_i/P) meets some thresholding criteria, where w_i is the edge-weight and P is the number of pixels corresponding to a node (segment).

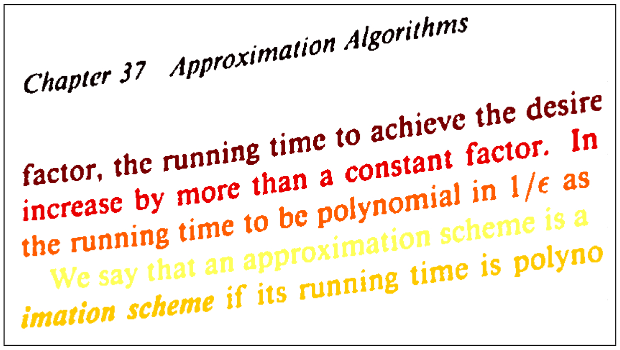

Chapter 37 Approximation Algorithms
factor, the running time to achieve the desire
increase by more than a constant factor. In
the running time to be polynomial in 1/ϵ as
We say that an approximation scheme is a
imation scheme if its running time is polyno

Figure 9: Small noisy connected components become a part of the nearest segmented textline. This is the final segmented output of an algorithm.

On the basis of the above description the performance evaluation metrics are:

- **Total oversegmentations (N_{tos}):** the total number of significant edges that ground truth lines have, minus the number of ground truth lines.
- **Total undersegmentations (N_{tus}):** the total number of significant edges that segmented lines have, minus the number of segmented lines.
- **Oversegmented components (N_{oc}):** the number of ground truth lines having more than one significant edge.
- **Undersegmented components (N_{uc}):** the number of segmented lines having more than one significant edge.
- **Missed components (N_{mc}):** the number of ground truth components that matched the background in the hypothesized segmentation.

Similarly, the description of performance evaluation metric for textline segmentation based on [8] is:

Table 1: Performance evaluation results

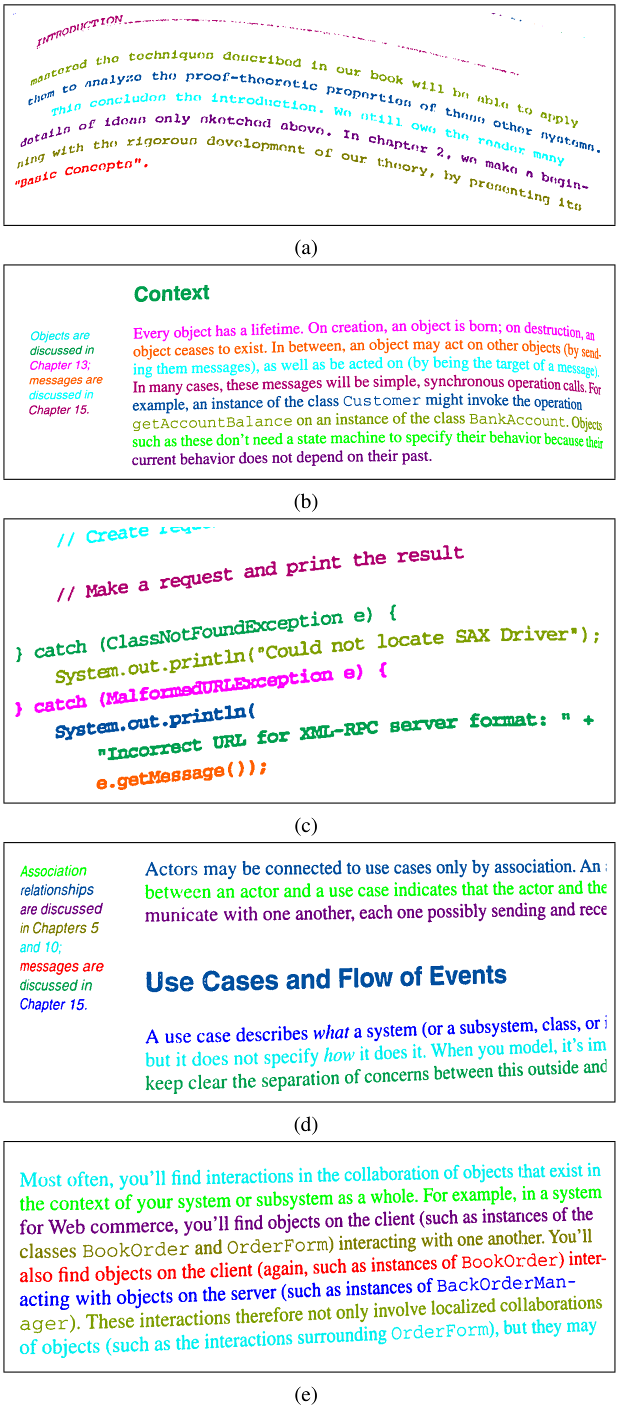
Number of ground truth lines (N_g)	3020
Number of segmented lines (N_S)	3227
Total oversegmentations ($100 * N_{tos}/N_G$)	8.91%
Total undersegmentations ($100 * N_{tus}/N_H$)	4.52%
Oversegmented components ($100 * N_{oc}/N_G$)	5.43%
Undersegmented components ($100 * N_{us}/N_H$)	3.63%
Missed components ($100 * N_{mc}/N_G$)	0%
Match-score ($100 * S_{match}$)	97.96%

- **Match-score (S_{match}):** the sum of maximum significant edge-weight of all ground truth lines divide by the total number of ground truth lines.

Performance evaluation results of our textline segmentation algorithm, based on the above metrics, are given in a Table 1. The results show that our algorithm achieved a match-score of about 97.96%, implying high segmentation accuracy of curled textlines in the CBDAR 2007 contest dataset. Our algorithm can handle several complex cases in which other segmentation algorithms usually fail. Our approach can do a proper segmentation in the case of a high degree of document curl, as shown in Figure 10a. It can correctly segment textlines in a two (or more) column document image. Most of the textline segmentation algorithms are unable to segment margin notes correctly. Our algorithm is capable enough of handling this situation as shown in Figure 10b. Our approach can handle small line spacing properly as shown in Figure 10c. Document images with variable font size and variable line spacing are also handled as shown in Figure 10d. Variable spacing within a line pair is also solvable as shown in Figure 10e. Segmentation result of the algorithm on a sample document from CBDAR 2007 dataset is shown in Figure 11.

4 Discussion

This contribution presented an algorithm for the segmentation of curled textlines from camera-captured documents. Most of the curled textline segmentation techniques in literature are highly dependent on document-language. The algorithm we present has a high potential of becoming general textline segmentation algorithm which is independent of document-language. After some modifications, our algorithm can be used for segmenting handwritten document images. Our future goal is to make our algorithm document-language independent and make it equally applicable to handwritten documents as well.



INTRODUCTION

mastered the techniques described in our book will be able to apply
them to analyze the proof-theoretic properties of these other systems.
This concludes the introduction. We still owe the reader many
details of ideas only sketched above. In chapter 2, we make a begin-
ning with the rigorous development of our theory, by presenting its
"Basic Concepts".

(a)

Context

Objects are
discussed in
Chapter 13;
messages are
discussed in
Chapter 15.

Every object has a lifetime. On creation, an object is born; on destruction, an
object ceases to exist. In between, an object may act on other objects (by send-
ing them messages), as well as be acted on (by being the target of a message).
In many cases, these messages will be simple, synchronous operation calls. For
example, an instance of the class Customer might invoke the operation
getAccountBalance on an instance of the class BankAccount. Objects
such as these don't need a state machine to specify their behavior because their
current behavior does not depend on their past.

(b)

```
    // Create requ
    // Make a request and print the result
} catch (ClassNotFoundException e) {
    System.out.println("Could not locate SAX Driver");
} catch (MalformedURLException e) {
    System.out.println(
        "Incorrect URL for XML-RPC server format: " +
        e.getMessage());
```

(c)

Association
relationships
are discussed
in Chapters 5
and 10;
messages are
discussed in
Chapter 15.

Actors may be connected to use cases only by association. An
between an actor and a use case indicates that the actor and the
municate with one another, each one possibly sending and rece

Use Cases and Flow of Events

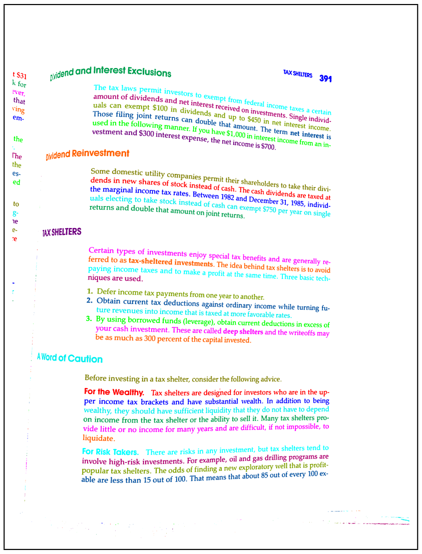
A use case describes *what* a system (or a subsystem, class, or i
but it does not specify *how* it does it. When you model, it's im
keep clear the separation of concerns between this outside and

(d)

Most often, you'll find interactions in the collaboration of objects that exist in
the context of your system or subsystem as a whole. For example, in a system
for Web commerce, you'll find objects on the client (such as instances of the
classes BookOrder and OrderForm) interacting with one another. You'll
also find objects on the client (again, such as instances of BookOrder) inter-
acting with objects on the server (such as instances of BackOrderMan-
ager). These interactions therefore not only involve localized collaborations
of objects (such as the interactions surrounding OrderForm), but they may

(e)

Figure 10: Proper segmentation efficiency of algorithm in complex situations. (a) High degree of curl. (b) Text-note aligned left to paragraph. (c) Small line spacing. (d) Variable font size with variable spacing. (e) Variable spacing within adjacent line pair.



Dividend and Interest Exclusions

TAX SHELTERS 391

The tax laws permit investors to exempt from federal income taxes a certain amount of dividends and net interest received on investments. Single individuals can exempt $100 in dividends and up to $450 in net interest income. Those filing joint returns can double that amount. The term **net interest** is used in the following manner. If you have $1,000 in interest income from an investment and $300 interest expense, the net income is $700.

Dividend Reinvestment

Some domestic utility companies permit their shareholders to take their dividends in new shares of stock instead of cash. The cash dividends are taxed at the marginal income tax rates. Between 1982 and December 31, 1985, individuals electing to take stock instead of cash can exempt $750 per year on single returns and double that amount on joint returns.

TAX SHELTERS

Certain types of investments enjoy special tax benefits and are generally referred to as **tax-sheltered investments**. The idea behind tax shelters is to avoid paying income taxes and to make a profit at the same time. Three basic techniques are used.

1. Defer income tax payments from one year to another.
2. Obtain current tax deductions against ordinary income while turning future revenues into income that is taxed at more favorable rates.
3. By using borrowed funds (leverage), obtain current deductions in excess of your cash investment. These are called **deep shelters** and the writeoffs may be as much as 300 percent of the capital invested.

A Word of Caution

Before investing in a tax shelter, consider the following advice.

For the Wealthy. Tax shelters are designed for investors who are in the upper income tax brackets and have substantial wealth. In addition to being wealthy, they should have sufficient liquidity that they do not have to depend on income from the tax shelter or the ability to sell it. Many tax shelters provide little or no income for many years and are difficult, if not impossible, to liquidate.

For Risk Takers. There are risks in any investment, but tax shelters tend to involve high-risk investments. For example, oil and gas drilling programs are popular tax shelters. The odds of finding a new exploratory well that is profitable are less than 15 out of 100. That means that about 85 out of every 100 ex-

Figure 11: Segmentation result of the algorithm on sample document from CBDAR 2007 dataset.

References

[1] T. M. Breuel. Representations and metrics for off-line handwriting segmentation. In *8th International Workshop on Frontiers in Handwriting Recognition*, pages 428–433, 2002.

[2] T. M. Breuel. Robust least square baseline finding using a branch and bound algorithm. In *9th Conference on Document Recognition and Retrieval*, pages 20–27, 2002.

[3] M. S. Brown and W. B. Seales. Image restoration of arbitrarily warped documents. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 26(10):1295–1306, 2004.

[4] H. Cao, X. Ding, and C. Liu. Rectifying the bound document image captured by the camera: A model based approach. In *Proceedings, International conference on Computer Vision*, pages 71–74, 2003.

[5] V. Caselles, R. Kimmel, and G. Sapiro. Geodesic active contours. In *International Conference on Computer Vision*, pages 694–699, 1995.

[6] C. G. Fraser. The origins of euler's variational calculus. *Archive for History of Exact Sciences*, 47(2):103–141, 1994.

[7] B. Fu, M. Wu, R. Li, W. Li, and Z. Xu. A model-based dewarping method using text line detection. In *2nd International Workshop on Camera Based Document Analysis and Recognition*, September 2007.

[8] B. Gatos, A. Antonacopoulos, and N. Stamatopoulos. Icdar2007 handwriting segmenentation contest. In *9th International Conference on Document Analysis and Recognition*, pages 1284–1288, 2007.

[9] B. Gatos and K. Ntirogiannis. Restoration of arbitrarily warped document images based on text line and word detection. In *4th IASTED International Conference on Signal Processing , Pattern Recognition, and Applications*, pages 203–208, 2007.

[10] B. Gatos, I. Pratikakis, and K. Ntirogiannis. Segmentation based recovery of arbitrarily warped document images. In *Proceedings, International Conference on Document Analysis and Recognition*, 2007.

[11] C. T. Hsieh, E. Lai, and Y. C. Wang. An effective algorithm for fingerprint image enhancement based on wavelet transform. *Pattern Recognition*, 36(2):302–312, 2003.

[12] M. Kass, A. Witkin, and D. Terzopoulos. Snakes: Active contour models. *International Journal of Computer Vision*, 1(4):1162–1173, 1988.

[13] Y. Li, Y. Zheng, and D. Doermann. Detecting text lines in handwriten documents. In *18th International Conference on Pattern Recognition*, pages 1030–1033, 2006.

[14] S. Lu and C. L. Tan. Document flattening through grid modeling and regularization. In *18th International Conference on Pattern Recognition*, pages 971–974, 2006.

[15] S. Lu and C. L. Tan. The restoration of camera documents through image segmentation. In *7th IAPR workshop on Document Analysis Systems*, pages 484–495, 2006.

[16] A. Masalovitch and L. Mestetskiy. Usage of continuous skeletal image representation for document images dewarping. In *2nd International Workshop on Camera Based Document Analysis and Recognition*, September 2007.

[17] F. Shafait and T. M. Breuel. Document image dewarping contest. In *2nd International Workshop on Camera-Based Document Analysis and Recognition*, Curitiba, Brazil, Sep 2007.

[18] F. Shafait, D. Keysers, and T. M. Breuel. Pixel-accurate representation and evaluation of page segmentation in document images. In *International Conference on Pattern Recognition*, pages 872–875, Hong Kong, China, Aug 2006.

[19] F. Shafait, D. Keysers, and T. M. Breuel. Performance evaluation and benchmarking of six page segmentation algorithms. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(6):941–954, Jun 2008.

[20] C. Strouthopoulos, N. Papamarkos, and C. Chamzas. Identification of text-only areas in mixed-type documents. *Engineering Applications of Artificial Intelligence*, 10(4):387–401, 1997.

[21] C. L. Tan, L. Zhang, Z. Zhang, and T. Xia. Restoring warped document images through 3d shape modeling. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 28(2):195–208, 2006.

[22] A. Ulges, C. H. Lampert, and T. M. Breuel. Document image dewarping using robust estimation of curled text lines. In *International Conference on Document Analysis and Recognition*, pages 1001–1005, 2005.

[23] F. M. Wahl, K. Y. Wong, and R. G. Casey. Block segmentation and text extraction in mixed text/image documents. *Computer Graphics and Image Processing*, 20:375–390, 2006.

[24] C. Xu and J. L. Prince. Snakes, shapes, and gradient vector flow. In *IEEE Transaction of Image Processing*, pages 359–369, 1998.

[25] Z. Zhang and C. L. Tan. Recovery of distorted document images from bound volumes. In *Proceedings, International Conference on Document Analysis and Recognition*, pages 429–433, 2001.