A comprehensive survey of mostly textual document segmentation algorithms since 2008

Sébastien Eskenazi, Petra Gomez-Krämer, Jean-Marc Ogier

A comprehensive survey of mostly textual document segmentation algorithms since 2008

Sébastien Eskenazi[a,*], Petra Gomez-Krämer[a,*], Jean-Marc Ogier[a,*]

[a]*L3i laboratory - La Rochelle University, Avenue Michel Crépeau, 17042 La Rochelle, France*

Abstract

In document image analysis, segmentation is the task that identifies the regions of a document. The increasing number of applications of document analysis requires a good knowledge of the available technologies. This survey highlights the variety of the approaches that have been proposed for document image segmentation since 2008. It provides a clear typology of documents and of document image segmentation algorithms. We also discuss the technical limitations of these algorithms, the way they are evaluated and the general trends of the community.

Keywords: Document, Segmentation, Survey, Evaluation, Trends, Typology

1. Introduction

Industrial document digitization, document archiving with destruction of the original copy and security technologies based on document processing create an increasing need for reliable document processing algorithms. A thorough list of the available algorithms would be of great use to choose them correctly. A typical paper document content extraction process is shown in Figure 1. Document segmentation aims at dividing the document image into meaningful parts. These parts can be glyphs, words, text lines, paragraphs, regions (usually with one type of content such as text or graphic). These parts are usually used for further content extraction such as text recognition, to determine the reading order, or to classify the document.

In order to split the document image properly we may need to understand its content

*Email: {sebastien.eskenazi, petra.gomez, jean-marc.ogier}@univ-lr.fr

and vice versa. This paradigm is related to the associationist and Gestalt theories of vision [1]. This is why document segmentation algorithms can include and work in symbiosis with a classification algorithm that will identify the content of the document.

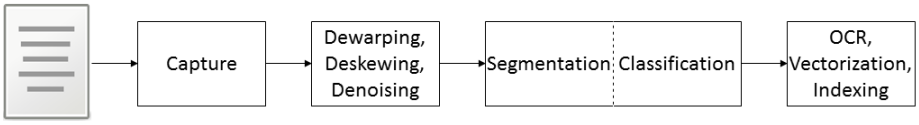


Figure 1: A classical document content extraction process

One of the first document image segmentation algorithms appeared in 1982 with the Run-Length Smoothing Algorithm (RLSA) [2]. It was followed in 1992 by the X-Y cut algorithm [3]. Many other algorithms have been presented since then and surveys have been done in 2000 [4] and in 2003 [5]. In 2007 and 2014, two surveys propose formal definitions and detail the main trends of document segmentation algorithms [6, 7]. There has also been several competitions and benchmarks providing an overview and a comparison of the state of the art [8–15]. However, these papers do not provide a coverage of all the available document image segmentation algorithms.

For the above reasons we present in this article a survey of recent document segmentation algorithms. Our work was motivated by the outstanding and very exhaustive review of natural image segmentation algorithms done by Vantaram and Saber [16].

We will first detail the scope of this survey in section 2. Then we propose a typology of document image segmentation algorithms in section 3. The next three sections survey the algorithms. Finally we discuss the limitations, the evaluation, the research trends and the industrial interest of the algorithms in section 7. The data used in this section can be found in the supplementary material. A conclusion completes this paper.

2. Scope of this survey

We will now detail the three scopes of this survey: the scientific scope (which algorithms are surveyed), the document scope (the type of documents on which they are expected to work) and the publication scope of this survey (for which publications we claim to be exhaustive).

2.1. Scientific scope of this survey

Segmentation algorithms can be applied to document images but also to a set of document images (in order to segment a book into its chapters by instance), to natural images [16], to medical images [17] and even to 3D meshes [18]. More generally a segmentation algorithm can be viewed as a specific kind of clustering or partitioning algorithm or of classification algorithm when it labels the parts that are segmented. More formally for the value of each input element $I(x)$ a segmentation algorithm associates a region number or a label $J(x)$ where x is the element index. x can be the page number, the node index, or even the pixel coordinates in an image.

$$\forall x, \quad I(x) \overset{Segmentation}{\rightarrow} J(x) \tag{1}$$

We only focus on offline document image segmentation algorithms. These algorithms' input excludes the history of the creation of the document such as writing strokes or typing/creation order. This kind of information is usually not available in a document analysis system such as the one described in Figure 1. However, we do not exclude algorithms that use the static image information to compute the document history. Such algorithms have been developed in the past [19] and in particular in paleography [20] but none was found during the time frame of this survey. Some algorithms such as some super pixel-based approaches [21] over-segment the images into small chunks that need a post processing merging step to make meaningful parts. These algorithms are not surveyed because they do not perform a complete segmentation on their own.

We include in this survey line segmentation algorithms as long as they explicitly detect the beginning, the end, the top and the bottom of each line. This excludes the algorithms that only draw a border between lines from one side of the page to the other.

Some algorithms are capable of identifying the type of content contained in each area. There are two kinds of identification: the physical layout and the logical layout. The physical layout relates to the nature of the content such as text, typewritten text, graphics, diagram, picture, decoration, etc. The logical layout relates to the function of the content such as header, footnote, main body, etc. In this survey we focus on algorithms that provide regions without labels or with a physical layout labeling. While

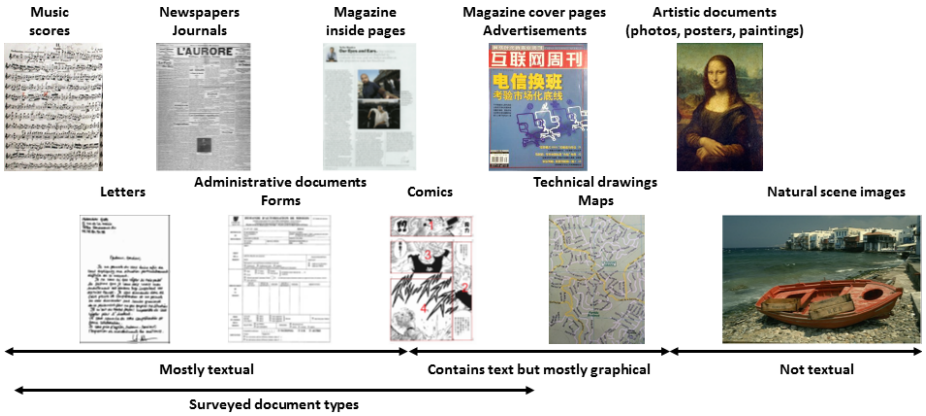


Figure 2: Document typology by increasing amount of text from left to right.

we don't exclude algorithms that provide a logical labeling, we do not discuss this capability. This is motivated by the fact that this kind of labeling is usually very specific to a document type or layout and cannot be generalized to other document types. We do not survey algorithms that start from a segmentation result and then label them.

While natural image segmentation algorithms are not made for document images, that does not mean that they cannot perform this task as shown in a recent study [22]. Nevertheless we do not survey them as they are usually not tested on document images. Neither do we consider the basics of segmentation algorithms. If needed [6, 7] provide a very good introduction to them.

2.2. Typology of documents

Since the main processing step of document analysis relies on extracting or indexing the text that they contain, we have chosen to sort them according to the amount of textual content they have. Figure 2 summarizes the main types of documents from the most textual on the left to the least textual on the right. It is difficult to make a generic and clear classification so the boundaries between the different categories should be considered as fuzzy. By instance, some comics do not contain any text and some magazine pages are only textual.

Music scores are classified as one of the most textual documents because we can consider music writing as a language and thus as textual content. However since the

layout of music scores is very constrained it is not a big challenge and no publication was made for analyzing it. Most publications relate to music symbol segmentation, staff removal and direct recognition of music scores without any layout analysis [23].

There is no specific category for historical documents. This is because all the categories on the figure contain both contemporary and historical documents. Moreover, defining a specific category for historical documents would require defining a boundary between historical and contemporary documents which would most likely not be feasible. The scientific community separates historical from contemporary documents on a subjective case by case basis, mostly because of the degradations and the variability they have and the specific algorithms needed to deal with them. Hence this separation should not be related to the time but to the degradations (and maybe the content) of the document. Similarly to natural scene images, it is not impossible that algorithms made for historical documents perform well on contemporary documents. Actually, many algorithms [24–32] have been evaluated on both modern and historical documents.

Similarly to the historical documents, handwriting can appear in all categories and as such does not have a category of its own. Moreover, documents that were written before the invention of printing were obviously handwritten but the writing style can be closer to machine print in some cases.

Color wise, there are three main types of color depth: black and white (BW), gray level (GL) and color (C).

We survey all mostly textual documents, the comics, the advertisements and magazines cover pages. We do not survey technical drawings or maps because the mere definition of what is their layout is already a challenge and they usually require very specific tools to be analyzed. We do not exclude an algorithm if it is tested on a data set that includes documents both in and out of the scope of this paper.

2.3. Publication scope of this survey

It is hardly possible to survey exhaustively all scientific journals, conferences and publications of the field. We have chosen to survey the algorithms published since 2008 since no survey includes papers after this date. Kise's survey in 2014 [7] - which is mostly and introduction to the field - only cites papers before 2007. We review

mostly but not only the papers published in the major venues for the document analysis community for which we claim to be exhaustive (in alphabetical order):

- ACM Symposium on Document Engineering (DocEng),
- Document Recognition and Retrieval (DRR),
- European Conference on Computer Vision (ECCV),
- International Conference on Computer Vision (ICCV),
- IAPR International Conference on Document Analysis and Recognition (ICDAR),
- IAPR International Workshop on Document Analysis Systems (DAS),
- IEEE Conference on Computer Vision and Pattern Recognition (CVPR),
- IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI),
- International Conference on Pattern Recognition (ICPR),
- International Journal on Document Analysis and Recognition (IJDAR),
- Pattern Recognition (PR),
- Pattern Recognition Letters (PRL)

3. Typology of segmentation algorithms

Before going through the in-depth survey of segmentation algorithms, it is wise to define a typology to organize them. Document image segmentation algorithms are typically classified into three groups [5, 6]: top-down, bottom-up and hybrid algorithms. Top-down algorithms start from the whole page and try to partition it. Bottom up algorithms start from a small scale and try to agglomerate the elements at this scale into bigger elements up to the the scale of the whole document. There are three main scales from which they start: pixels, connected components and "patches" which is a user-defined scale. This classification is very objective but does not reflect the capabilities and limitations of each algorithms. It only reflects the order of information processing.

From a different perspective, Kise [7] classifies first the algorithms according to their capability of segmenting documents with overlapping layouts such as a stamp on top of some text. This allows one to select a suitable algorithm based on the segmentation task at hand. However this typology only considers classification algorithms for

segmenting documents with overlapping layouts which is restrictive. Extending this typology for an exhaustive survey would lead to classifying some algorithms in several categories which would make each category and the whole classification less legible.

A given segmentation algorithm may not be able to segment any layout. This is the main limitation of such an algorithm and we use it to classify the surveyed algorithms into three groups. The layout segmentation limitation can come either from the way the algorithm itself works (group one) such as X-Y cut that has been written to segment a specific kind of layout with only horizontal and vertical regions (called Manhattan layout). It can also come from the parameters given to the algorithm (group two) such as Voronoi which is versatile but requires different parameters depending on the document style (font size, noise characteristics, connected component size distribution, etc.). A third group of algorithms attempts to overcome these limitations and could potentially not have any (such as neural networks). The overall algorithm type classification is shown in Figure 3. Thanks to the groups we defined it allows us to represent both the techniques and the limitations of the algorithms. Most algorithms rely on one main technique but also make use of other secondary techniques to obtain intermediate data. As such, the classification of an algorithm is necessarily fuzzy and we classified each algorithm based on its core technique.

3.1. The algorithms in group one

The algorithms of this group were the first to appear. They usually aim at segmenting a specific, predefined kind of layout such as a Manhattan layout for instance. Hence they can be used without any training. There are three main subcategories in this group:

The algorithms that make clear assumptions about the document layout [15, 26, 33–41]: they either define this layout with a grammar, a set of rules or they assume that it is a Manhattan layout and use projection profiles.

The algorithms that use filtering techniques to make the document regions appear [42–48]: They usually rely on RLSA, mathematical morphology or other filters. The filters characteristics reflect the assumptions made on the layout geometry.

The algorithms that try to identify straight lines [27, 49, 50]: This can be done

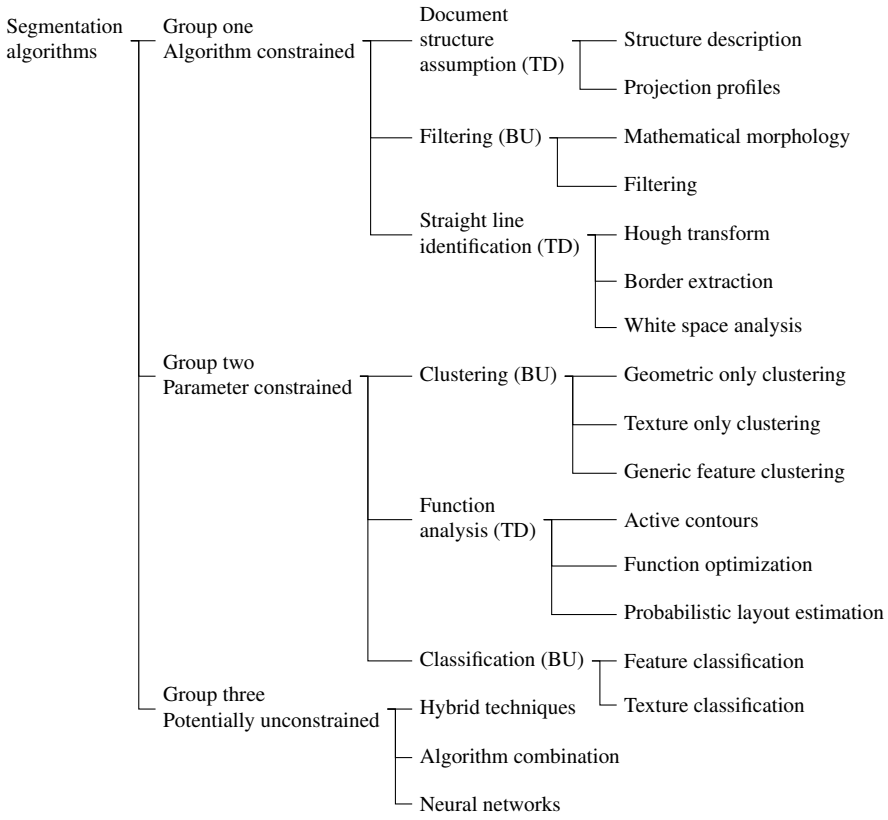


Figure 3: Typology of document segmentation algorithms. We also specify top-down (TD) and bottom-up (BU) algorithms.

with a Hough transform, by trying to identify straight lines or square borders, or by identifying white spaces alignment, in which case the "lines" may be invisible lines.

3.2. The algorithms in group two

They where the second to appear. Their difference with the algorithms in group one is that they try to adapt to local variations in the document in order to be able to segment a broader range of layouts with the same algorithm. This is achieved by using more flexible algorithms. The counter part of this is usually a higher number of parameters which are difficult to tune and may require training on a large data set. These algorithms are usually only limited by the values assigned to their parameters. There are four main subcategories in this group:

The clustering algorithms [9–12, 14, 25, 28, 30–32, 51–74]: They are bottom up algorithms and try to cluster elements based on geometric or texture or a more general set of features.

The algorithms based on function analysis [14, 15, 24, 75–87]: they mostly rely on function optimization e.g. trying to bring a function as close as possible to an objective value. Some specific cases are active contours, energy minimization and probabilistic layout estimation. They are top-down algorithms. In interesting fact about these algorithms is that while most algorithms work with the region areas, those based on function analysis usually work with the region boundaries.

The classification algorithms [9, 15, 29, 88–110]: they are trained to recognize the different types of elements based on a given set of features (purely texture or more generic features). As a consequence they need training and produce labeled elements. The regions are made of adjacent elements having the same label hence they are bottom up algorithms.

3.3. The algorithms in group three

They appeared last. They try to overcome the limitations of the other algorithms by combining them or by using artificial intelligence. Thus they cannot be considered as bottom-up or top-down algorithms. There are three subcategories in this group:

The hybrid algorithms [9, 14, 15, 111–114] combine several other algorithms in symbiosis: While they could potentially accumulate the strength of several other algorithms, some of them tend to be very complex without significant performance or versatility improvement.

The combination algorithm (only one has been presented) [115] combines the results of several algorithms to effectively improve them.

The neural network algorithms [15, 116] make use of artificial intelligence to automatically learn significant features and perform the required task. They need careful design and are subject to over training. They also tend to be a “black box” whose functioning is not easily explained.

4. Layout constrained by the algorithm (Group one)

The algorithms in this group are such that their internal mechanisms limit the variety of layouts that they can segment.

4.1. Segmentation based on document structure assumption

These algorithms are the most limited ones. They are made for a very specific type of layout hence they are only applicable to documents with a structured layout. This drawback is counterbalanced by their success rate in segmenting this specific layout in comparison with other more flexible techniques. They are also extremely fast.

4.1.1. Grammar

There are five algorithms of this type [26, 33–36]. They were first published in 2006 by Coüasnon [117] and tested on a data set of 88745 documents which is an unrivaled data set size. He designed a layout grammar language called DMOS. This language can describe any layout and the associated parser recognizes this layout in an image. The grammar also allows the association of a label to each region of the layout thus producing a labeled segmentation. Lemaitre et al. improved it in 2008 [26] by adding a multi resolution approach which made it flexible enough to segment handwritten letters (provided that their layout obeyed certain rules) and to identify text lines in administrative documents in French and Bangla. Carton et al. [34] continued this work with an interactive training step capable of creating automatically an exhaustive set of models for a large data set.

Shafait et al. [35], proposed another grammar algorithm based on a probabilistic layout formulation. The user defines a set of cuts (only horizontal and vertical in the paper) whose position is defined approximately. Then for each image a probabilistic fitting is performed to obtain the appropriate regions. This algorithm is capable of segmenting tight layouts with very small margins. Its results are not compared with those of DMOS.

4.1.2. Projection profiles

There are 7 algorithms of this type [37–41] and the two CASIA algorithms published in [15]. Ouwayed and Belaid [41] use projection profiles to segment multi-

oriented, text only documents. They make a paving of the document with rectangles. Then they compute the projection profile of each rectangle along several directions. The direction with the highest maximum of Wigner-Ville distribution is that of the text. Then, they use heuristics combined with local projection profiles to detect regions with non homogenous text orientation and text lines. They are also capable of segmenting curved text lines. Another feature of the algorithm is the capability to separate intricate text lines but this costs the generality of the algorithm as it requires typographic heuristics specific to Arabic handwriting.

Liu and al. [38] use [41] to segment Manhattan layouts. After a binarization that replaces text lines by black regions, they use projection profiles to remove border noise and detect text columns. The interest of the algorithm lies in the noise removal process. It first classifies the text lines into two levels of confidence. Then based on the most confident ones, it computes its internal feature parameters on the fly. These features help discriminate the lines as noise or text lines. The algorithm improves significantly the state of the art and is evaluated on 1922 documents containing four representative languages: Arabic, English, Chinese and Yiddish.

4.2. Segmentation based on filtering algorithms

These algorithms usually use specific predefined filters to segment a given type of content. They frequently rely on assumptions that the text lines are straight and/or horizontal.

4.2.1. Mathematical morphology

Six algorithms using mathematical morphology have been published [42–47]. Bockholt et al. [42] use several combinations of erosion and dilation to efficiently identify successively the pictures, the graphics and the text. While being a basic type of processing it proves very efficient for the task of document retrieval.

Ferili et al. [47] replace the logical AND of RLSA by an OR. This makes the algorithm more computationally efficient as only one run length is performed. They make the assumption that the text is horizontal. An interesting addition is the extension of the algorithm to natively digital documents based on their basic blocks.

Buckhari et al. [44] use Bloomberg's segmentation algorithm [118] which is based on mathematical morphology. They add a first step to merge broken horizontal and vertical lines with a hit-miss morphological transform and a second step to fill holes.

4.2.2. Filtering

The only contribution for this kind of algorithm was done by Shi et al. [48]. It was reused by A2iA in a competition [15] where it was ranked third out of 7 participants. The method is based on steerable filters (filters that can be rotated) to detect text lines along five orientations. A heuristic post processing is used to solve the issue of connected components spanning several lines.

4.3. Segmentation based on straight line identification algorithms

Three contributions are based on straight line identification [27, 49, 50]. Louloudis et al. [27] split the connected components horizontally into blocks based on the average character height. This allows the algorithm to work on handwritten text where several characters are merged into one connected component. However the horizontal partitioning assumes that the text is also horizontal. Once this partitioning is done, they apply a Hough transform on the centers of gravity of each block to detect text lines.

Wang et al. [50] attempt to reconstruct the border of the frames in comic books in order to segment them. Their algorithm is able to segment frames with only two apparent borders but is limited to quadrangle regions. They separate the background then they use the Douglas-Peucker algorithm [119] to fit quadrangles onto the candidate frames. This is followed by a classification of the frame complexity and specific heuristics are used to complete the frame border. This algorithm only improves the state of the art for difficult to very difficult layouts which is its original goal.

Chen et al. [49] analyze the white spaces to segment the document into text columns. The connected components are grouped into horizontal chains to create white spaces between these chains. Then the white spaces are grouped vertically to make white lines/column separators. The algorithm works well to detect text but not for graphical parts. It won the two segmentation competitions of ICDAR 2013 [12, 13].

5. Layout constrained by the parameters (Group two)

This group contains the majority of algorithms that have been published. They remain fairly simple while being flexible enough to address a wide range of problems.

5.1. Segmentation based on clustering

This is clearly the most popular type of algorithm with 40 algorithms.

5.1.1. Geometric only clustering

The vast majority (31 publications) of clustering algorithms uses only geometric features. Before further classifying them based on the color information level they can process, we can highlight the contribution of the Fraunhofer Institute and the team of Konya et al. who made several contributions to the field and participated in every document segmentation competition and won two of them [9, 10, 12–14, 30].

Black and white algorithms. Most of them only process black and white images [9–12, 28, 30–32, 59–71]. Liu et al. [63] use a Gaussian Mixture model to classify connected component triplets as text or non text. They use three geometric features (distance, area, density) and thus have trivariate Gaussian distributions. The first order neighborhood of a connected component is computed with the Delaunay triangulation and they use the second order neighbors to obtain all the possible triplets. They also use a specific training called MMS which maximizes the class separability. Although the algorithm is not made for color images, it is tested on binarized color advertisements and magazine cover pages. It performs well with a precision and recall over 90%.

Agrawal and Doermann improved the original Voronoi algorithm [120] with Voronoi++ which adapts the Voronoi parameters to the local spatial context [55]. Then they made a fuzzy version of it (with fuzzy edges) called CVS [56]. It formulates hypothetical regions and then validates them. The validation phase is done based on a distance and similarity (texture) contexts. This was evaluated on 350 documents with 5 evaluation schemes and consistently outperforms Voronoi and Voronoi++.

Gaceb et al. [61] use a custom binarization optimized for fast processing. They take a very novel stand in trying not to group dissimilar connected components. They do this

with a graph coloring technique where the dissimilarity constraint is reflected by that of two adjacent (dissimilar) nodes (connected components) having a different color. The connected components having the same color are the text lines of the documents. They benchmark their algorithm on 10000 envelopes and it outperforms RLSA and X-Y cut while providing a significant speedup.

Yin and Liu [70] use metric learning based on geometric features to compute the minimum spanning tree between the connected components of the binary image. A post processing is then applied to obtain the final text-lines.

Faure and Vincent [60] use geometric clustering to segment horizontal and vertical text plus technical drawings in historical documents. The interesting additions they have is the use of a confidence value for each alignment (text line) and a conflict resolution post processing when there is an inconsistency between two text lines.

Olivera et al. [67] improve their parallel line regression algorithm [66] by creating queues of horizontal and vertical neighbors of every connected component. They are processed by decreasing order of queue length and the parallel line regression clustering is applied on each queue. The parallel regression is based on geometric heuristics deduced from the typographic rules of six different fonts. This improvement allows them to significantly reduce the over segmentations of their previous algorithms and actually improves the state of the art.

Liu et al. made two contributions in the perspective of near-duplicate document image matching. In the first one [64] they index a document with a set of features among which is the distribution of distances between the segmented components of the document. This segmentation is performed by grouping the connected components based on a distance threshold. This was successfully tested on more than 24000 images. In [32] they consider several segmentations of a given document. They build a component hierarchical tree and then build every possible segmentation across all the tree levels. They consistently outperform the state of the art on 1425 modern and historical documents except for 20 text only documents.

Other algorithms. Out of the five algorithms that make use of more information [12, 14, 72–74], two stand out. Ouji et al. [72] have a versatile algorithm based on a

deep understanding of how a document is created. They identify color and non color regions in a document image. Then they separate the non color regions into binary (text) and gray-level (illustration) regions. The color regions are similarly separated into monochromatic and polychromatic ones based on a multi-level analysis. They outperform the state of the art on 448 challenging documents from magazine inside pages to advertisements with text overlapping natural images.

Clavelli and Karatzas [74] segment propaganda posters that have nearly uniform colors. They make use of this property to segment the image components with a pixel clustering based on pixel neighborhood and RGB color distance. Then they define a search region around each component in order to group them into text lines. This allows them to identify text lines with letters of different colors and of very curved shapes with a varying background.

5.1.2. Texture only clustering

Three algorithms use only texture features [25, 57, 58]. Journet et al. [25] use features at pixel level and tested them on both modern and historical documents. They highlight the importance of a multi-resolution approach to reduce the noise in pixel clustering techniques. Working at pixel level allows the clustering of many different types of objects such as drop caps, a specific kind of graphic, text, text fonts, etc.

Mehri et al [57] demonstrate that Gabor texture features outperform auto-correlation and co-occurrence texture features.

5.1.3. Generic feature clustering

Six algorithms use several types of features [51–56]. The most outstanding one is that of Chen and Wu [54]. Roughly, they cut the document into blocks which are then multi-thresholded to create several layers. The connected components of each layer are identified and grouped across blocks based on a predefined set of features. The evaluation data set is small (65 documents) but very challenging as it contains only magazine covers and advertisements that are multi-layered color documents with uneven background. They outperform significantly the state of the art and achieve both precision and recall above 99% for text extraction.

Carel [51] uses a multi-resolution color and spatial clustering to identify the color layers contained in a document and the connected components in each layer.

5.2. Segmentation based on function analysis

Sixteen algorithms rely on function analysis techniques. They have the advantage that, based on the "flexibility" of the functions, one can select how much they will follow the contours of the elements to segment. This can be helpful if we want to have a rough outline of the document regions or if we want to segment precise elements such as warped text lines.

5.2.1. Active contours

All these techniques were proposed by Bukhari et al. [75–81]. Their work can be considered as the state of the art for text line extraction based on active contours. It works by adding coupled snakelets (a kind of non closed active contour) on the top and bottom of a connected component and by deforming them based on the vertical component of the gradient vector flow. The snakelets are then extended laterally in order to include neighboring connected components. This algorithm has been evaluated on 10 different scripts in [78].

5.2.2. Function optimization

Seven algorithms use function optimization [14, 15, 24, 82–85]. They usually define a cost or energy function which needs to be minimized. So far, they work best for text line segmentation although the ISPL method was second in the last ICDAR document segmentation competition [14]. The state of the art with this technique is held by Ryu et al. [82] which also won the ICDAR 2013 Competition for handwriting segmentation [121] and 2015 Competition on text line detection [15]. Their contribution resides in over-segmenting connected components that do not fit a normalization criteria. From this they obtain a better estimation of the belonging of each connected component to a given text line which in turn allows them to build a better cost function. The optimization of this function is improved with dynamic programming. The over segmented components are then merged into proper components.

Shen et al. [84] use both intra and interline metrics to build a segmentation cost function. After an initial geometric clustering they perform a simulated annealing optimization. This means that the probability of accepting a new segmentation that increases the cost function is not null but decreases with each iteration. Combined with a custom binarization, this allows them to extract text line from challenging color documents with non uniform background (CD covers) or unusual layout (business cards).

Kim and Oh [83] highlight the interest of using interline information over intra-line information for Asian scripts.

5.2.3. Probabilistic layout estimation

Two algorithms make a probabilistic estimation of the layout [86, 87] but do not bring a significant improvement. Yin and Liu [86] perform an estimation of the number of text lines with a blur filter and then use a variational Bayes approach to segment the image rescaled at 75 dpi. This improves slightly the state of the art on a large but not very challenging data set.

Cruz and Terrades [87] proposed a method based on Conditional Random Field (CRF) and location features similar to [29] (see section 5.3.2) but without any improvement over the state of the art. This work is at the crossing between optimizing a probabilistic layout estimation and a classification.

5.3. Segmentation based on classification

This is the second most popular type of algorithms with 30 algorithms. A noticeable difference in the scientific work when compared with the clustering is the fact that classification algorithms all require training.

5.3.1. Texture classification

Three algorithms use only texture features [9, 88, 89]. Baechler and Ingold [89] string together three Dynamic Multi-Layer Perceptrons (DMLP) at three resolutions to segment historical documents. Each DMLP uses the label output of the DMLP at a lower resolution plus texture features at its resolution. Each level processes only part of the labels produced by the lower level in order to refine these specific labels.

5.3.2. *Feature classification*

Most classification algorithms (24 out of 27) use a generic set of features.

Black and white algorithms. Among them, 12 use only binary images [9, 29, 90–99]. Peng [99] and Pinson [93] focus on extracting overlapping handwritten and typewritten text. Pinson and Barret's algorithm [93] automatically selects the appropriate features based on the desired typewritten text classification accuracy. It selects the first 100 feature vectors of a Principal Component Analysis (PCA) of the character images. Then any new text is projected into this new space. If it is close enough to the typewritten or the handwritten templates then it is classified appropriately. Otherwise it is considered as made of several touching characters and split with a graph-cut thus making two new connected components to classify. They achieve 98% precision for typewritten text and 71% precision for handwritten text on 500 forms each coming from a different writer. The low handwritten precision is related to the training set that did not contain typewritten text of a small size. As a result small typewritten fonts were classified as handwritten. Replacing the small font size in the test set brings the precisions to 94% and 89% respectively. This highlights the limits of the training set and the versatility of the algorithm with respect to the writing style.

Peng et al. [99] work at connected component and patch level. Patches are found with a morphological closing. They use a first Markov Random Field (MRF) to classify the patches into typewritten, handwritten or overlapped text. Then, they use another MRF to reclassify them based on their context. The overlapped text is separated at a pixel level with a third MRF and by using Shape Context Features (SCF) [122]. It performs slightly worse than Pinson but the data set size (28 documents) hinders the significance of this performance.

Bukhari [92] focus on extracting text from documents that contain graphics illustrations such as circuit drawings. The challenge here is to identify correctly text and graphics. While the challenge seems easier than separating overlapped typewritten and handwritten text, many segmentation algorithms do not handle graphics well. Thus targeting this specific issue is a good addition to the state of the art. To do this, they rescale every connected component to a predefined size and do the same with a wider

image centered on the connected component in order to capture its context. Then they use a Multi-Layer Perceptron (MLP) to do the classification. The recall for text and non-text (graphic) is consistently above 93 and 96% respectively on 100 documents.

Fern [29] focuses on extracting the regions of structured documents. They use Gabor (texture) features with a CRF. Their contribution lies in the addition of relative location features which are the probability of a region being of a certain class given its position relatively to the regions of the other classes. These features (one per class and per region) have a significant impact for segmenting structured documents. The improvement remains less significantly on non structured documents.

Gray-level algorithms. Four algorithms use gray level information [100–103]. Diem [102] tackles the challenge of segmenting document fragments. After extracting candidate word blobs with projection profiles, they introduce Gradient Shape Features (GSF) to refine the segmentation and classify the text as handwritten or typewritten with a support vector machine (SVM). GSF are computed on a sliding window scaled to the size of the word blob. For a given window they are similar to shape context features applied on the inverted gradient image instead of the original image. Then they perform a geometric clustering of the word blobs into lines. A final global voting with another SVM classifies the candidate lines into typewritten or handwritten again. They include an error back-propagation to relabel the word blobs that were mistakenly labeled. They improve the state of the art for graphics classification while maintaining a similar performance on the text in ICDAR 2009 segmentation competition [9].

Zhong and Cheriet [100] devise a new tensor-based learning algorithm and apply it successfully to classify text and non text (borders, noise, background, etc.) on text only ancient manuscripts.

Color algorithms. Eight feature classification algorithms use color information [15, 104–110]. Garg et al. [110] separate text and graphics in challenging magazine covers. They use an SVM to classify Gabor and edge features followed by a CRF to include the local spatial context. The the CRF improves the performance by 2 points.

Wei et al. [106] compare the performance of SVM, MLP and GMM (Gaussian mixture model) classifiers. They find that SVM and MLP outperform GMM but cannot

conclude which one is best. This depends on the data. The dependency on the features is not studied.

Wang et al. [104] make a very interesting contribution by finding a way to automatically discover features to improve the performance of a 2-nearest neighbor classifier. Given a large set of features, they sample the feature space. From this sample they find a cluster of errors and project it into the quotient space with the already discovered features. Then they compute a new feature in this hyperspace with a linear combination of the already existing features.

We finish this section with an algorithm on the border between classification and neural networks (which belongs to the third group). Chen et al. [109] use convolutional autoencoders (a kind of neural network) to automatically discover distinctive features on image patches at three scales and train an SVM with these features. In their evaluation they demonstrate the usefulness of combining these features and their superiority to handcrafted features.

6. Layout potentially unconstrained (Group three)

This last group contains eleven algorithms with a majority of them published since 2014. These algorithms try to overcome the shortcoming of the others by hybridizing them, combining them or with advanced neural networks.

6.1. Segmentation based on hybrid techniques

A large majority of them make several techniques work in symbiosis to obtain better results [9, 14, 15, 111–114]. The most recent significant work was the MHS method developed by Tran et al. which won the last ICDAR complex document segmentation competition [14]. It works by iteratively classifying connected components based on multi-level homogenous regions and white space analysis.

Another significant contribution is the one by Barlas et al. [112] which can segment an extremely diverse and complex range of documents in several languages. They generate a feature codebook with self-organizing maps which serves to describe the training set and then train an MLP. Once they have obtained the class layers, they

create regions by combining RLSA and white space analysis. They tested it on 1000 documents from the Maurdor data set with three classes (graphics, typewritten and handwritten text) and won the evaluation campaign.

Wang et al. [114] propose another algorithm for extracting text lines from complex documents with multi-oriented text in several languages. They use MSER to extract candidate characters which are then classified as text or non text with a fast Adaboost classifier. The low confidence text is further evaluated with a Convolutional Neural Network (CNN). The next step is a coarse line extraction with geometrical grouping based on a linearity constraint. This graph is refined by a minimum spanning tree. The last step is an energy minimization refinement of the lines.

6.2. Segmentation based on algorithm combination

The only contribution to this type of algorithm was done by Stamatopoulos et al. [115]. They devise a procedure to significantly improve the performance of individual segmentation algorithms by combining their results. The procedure is based on the overlap of the regions produced by the algorithms. They are considered as good above 90% overlap. They use the regions that have more than 70% to compute the values of a task based set of features. All regions below 90% undergo a splitting based on their intersection followed by a merging starting from the regions with the highest overlap. This is successfully applied to text lines segmentation and improve single algorithm results by 15 to 25% depending on the metric.

6.3. Segmentation based on neural networks

Two algorithms use neural networks [15, 116]. The A2iA1 in the text line segmentation competition [15] did not won it and it is similar to that of Moysset et al. [116] which seems to have improved it. They normalize the width of a text column (or of a text only single column document) and use a bidirectional long-short term memory neural network (BLSTM) to detect text lines and paragraphs. They tackle the issue of modeling gaps and interlines and conclude that each should have its own class. They also show that using specialized neural networks trained on a specific set is better than using a single system trained on a more varied data set.

Figure 4: Comparison of thresholding and error diffusion binarization techniques. Left: with thresholding, right: with error diffusion.

7. Discussion

In this section we discuss the applicability scope of the surveyed algorithms, the evaluation practices of the community, its general trends. Finally we quickly review the surveyed algorithms from a user's point of view.

7.1. Remarks on the applicability of the surveyed algorithms

When choosing a segmentation algorithm there are a few important remarks to have in mind as they may impact its suitability for a given task.

As a general rule, all segmentation algorithms are script independent e.g. their performance does not depend on the language of the document. Yet they frequently make the assumptions that a language is made of disjoint characters and that each character is made of one connected component. These assumptions are not true for handwritten text and for several languages. Thus the script independence of an algorithm may be limited to scripts that satisfy one or both of the above assumptions.

Many algorithms working on binary images assume that the image is the result of a thresholding process similar to Otsu binarization. They usually do not work on binarization techniques that use error diffusion (dithering). In this case, the image should first be converted to gray level and then appropriately binarized. Figure 4 shows the difference between these two types of binarization. Also, most analyzes based on connected components require binary images.

One may think that being able to segment curved text is better than only straight lines. While this is generally true, it can also lead to merging straight lines that are close to each other and only separated by their orientation.

Many text line segmentation algorithms assume that the document contains only text and should be complemented with another algorithm capable of dealing with non textual content. Classification techniques provide labeled regions but often cannot cre-

ate a text line level segmentation. If one needs this level of segmentation, they should either use another algorithm or add a text-line segmentation algorithm.

Regarding the processing speed and independently from the implementations that can drastically impact it, bottom-up algorithms are usually slower than top down algorithms and the smaller their processing level, the slower they are. This is illustrated by the fact that there are many more pixels than connected components in an image. The more items there are to process the longer the processing will be.

Although most papers specify parameters in pixels it is better to use the resolution of the test data set to convert these parameters in absolute units such as millimeters. This will make the algorithm resolution independent and more versatile.

Not all algorithms absolutely require a training data set. However, they can have many parameters which may require training.

Finally, one should keep in mind that an algorithm that has been trained on a given data set is only supposed to segment correctly similar documents. While it may work for other kinds of documents the training data set is usually a limitation to the segmenting capability of the algorithm. This fact is clearly shown in the scientific papers by the difference of size between the training and testing sets. The training set is often several times larger than the testing set.

7.2. Evaluation of segmentation algorithms

Evaluating and comparing the performance of an algorithm is a difficult task. We summarize here the current state of the art for the evaluation of document segmentation algorithms. A first remark that can be made in the light of all the surveyed papers is that many authors tend to compare their algorithms with other algorithms based on the same technique. This prevents any cross-technique comparison to estimate which techniques are the most promising. Competitions and contest try to remedy this but it would be good if the community could base its future evaluation on the comparison with the best algorithms having the same functionality instead of the same technique.

7.2.1. Existing benchmarks

There has been three independent industry lead benchmarks: the MADCAT program[1] by DARPA, RIMES[2] by A2iA which is a document analysis company and MAURDOR[3] led by the French equivalent of DARPA, the DGA. MADCAT aims at creating a system to automatically categorize and translate any document into English. RIMES is a data set of handwritten letters that has been used for several competitions [123, 124]. The most comprehensive one is clearly the MAURDOR campaign and it is publicly available[4]. It contains more than 8000 color documents with a complete ground-truth (regions, region type, contained text, text type, text language and other meta data) as well as ready to use evaluation tools. It contains all types of document except for comics and in French, English and Arabic typewritten and handwritten text.

ICDAR also organizes recurrent document segmentation competitions [9–15]. Unfortunately they frequently use a data set that is too small (below 100 documents) and the evaluation tool is closed source, does not allow processing documents in batch mode and contains numerous parameters which may make the evaluation less objective [9, 10, 12–14]. The competition by Murdock et al. [15] is a very good addition as it comes from the industry and thus aims at addressing real case scenario issues. The competition that was organized by Lamiroy et al. [11] adopted an interesting point of view in that it divided an end to end system in several steps for which submissions could be made. Then the contribution of each algorithm was evaluated with the improvement it brought to the overall system.

7.2.2. Data sets

The evaluations in the different surveyed papers raise the question of the exhaustivity of the data sets. Clearly, the state of the art is the MAURDOR data set. Yet it does not contain Asian or Cyrillic scripts. Historical documents are also missing. For these, the St. Gall, Parzival and Washington triptych [105–107, 109][5] is recommended

[1] `http://opencatalog.darpa.mil/MADCAT.html`
[2] `http://www.a2ialab.com/doku.php?id=rimes_database:start`
[3] `http://www.maurdor-campaign.org/`
[4] `http://catalog.elra.info/product_info.php?products_id=1242`
[5] available at `http://www.fki.inf.unibe.ch/databases/iam-historical-document-database`

as it covers a wide time span. Adding copies of the same documents could allow the evaluation of the stability of segmentation algorithms as done in [22]. A last addition could be documents similar to comics such as those in [50, 68]

In section 2.2 we highlighted the difficulty of defining what is a historical document. A 16^{th} century manuscript is not the same as a decree from the 19^{th} century, yet both are historical documents. The exhaustivity in that regard should be considered with the sampling of documents in time not just sampling two classes of documents.

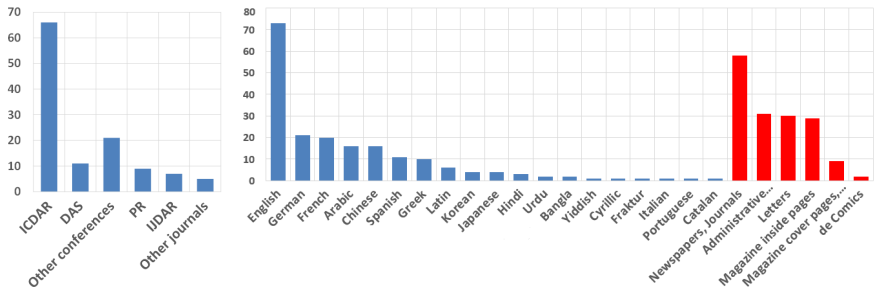
The testing data sets do not all have the same exhaustivity. Clustering algorithms are evaluated on data sets that are, on average, six times bigger than those used for classification algorithms. This is explained by the fact that classification algorithms require large training data sets in particular because of the curse of dimensionality. Thus for the same total data set size (training + testing) a clustering algorithm will usually be tested on a larger data set. However, this does not change the fact that, because of the testing data set size, the results obtained for clustering algorithms are more reliable than those for classification algorithms.

Moreover, Baird and Casey [125] advocate for versatile algorithms. These are algorithms capable of dealing with a wide range of documents. Thus, they will have to handle documents that they have not encountered before. Hence the training set should definitely not be bigger than the testing set and the classical k-fold evaluation and data sets that specify training and testing sets need to be updated accordingly.

7.2.3. Metrics

Most evaluations are based on the same principles of counting: false alarms (adding a region), misses (removing a region), merges (two or more regions in one), splits (one region in two or more) and matches (properly segmented region). Algorithms that tend to merge (respectively split) regions are said to under-segment (respectively over-segment) the documents. This is the most reasonable way to evaluate the performance of an algorithm on a single document. However this kind of metric does not evaluate the repeatability and performance stability of the algorithm over a range of documents. Baird and Casey [125] denote this as evaluation based on "confidence before accuracy".

We actually analyzed the repeatability of four state of the art segmentation algo-



(a) Number of publication per venue

(b) Number of algorithms that studied each language and document type.

Figure 5: Venue, language and document type publication trends.

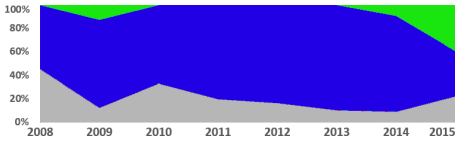
rithms [22] and found that they all have a very poor stability. This is to be expected since they where never evaluated with this criteria but we hope that it will be used in future evaluations.

7.3. Trends and statistics

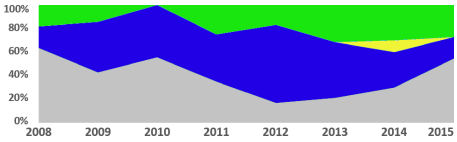
This survey gave us a very good insight on the trends, strengths and weaknesses of current algorithms. Figure 5a shows the main publishing venues.ICDAR stands as the flagship conference of the community. DAS is the second biggest venue of the community and its main journals are PR and IJDAR.

Figure 5b shows the detailed number of algorithms that studied each language and document type. The algorithms have been applied to 19 languages and scripts. This supports the worldwide applicability of the findings of the document analysis community. The top 5 most studied languages are in decreasing order: English, German, French, Arabic and Chinese. Regarding the document types, newspapers and journals have been studied twice as much as any other type. The least studied types are magazine cover pages and comics.

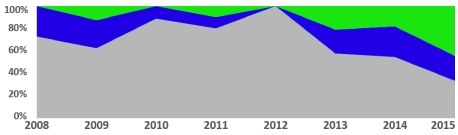
Figure 6a summarizes the number of publications based on each technique group. The papers are counted in the group in which they were presented above. The techniques in groups 1 and 2 are disappearing to the profit of the third group. The main techniques used for document segmentation are bottom up techniques, in particular geometric clustering and feature classification. They cover 24% and 19% of all surveyed



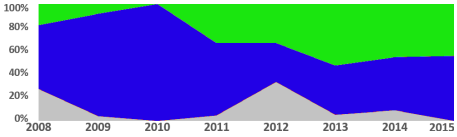
(a) Shares of the algorithms groups. From top to bottom: group three, two and one.



(b) Shares of processing level for the algorithms in the first two groups. From top to bottom: pixel, patch and connected component level for the bottom-up algorithms. In gray at the bottom, top-down algorithms.



(c) Shares of the color depths that are used by the surveyed algorithms. From top to bottom: color, gray level, black and white.



(d) Shares of the algorithm that are evaluated on historical documents. From top to bottom: historical, modern, both.

Figure 6: Trends of algorithm techniques and evaluations between 2008 and 2015

algorithms respectively. All geometric clustering algorithms work at the connected component level.

For the first two groups, we can analyze the shares of top-down and bottom-up algorithms and among the bottom-up algorithms the processing scale. Figure 6b shows that there is majority of bottom-up algorithms but top-down algorithms are gaining a new interest. Overall, most of the bottom-up algorithms work at the connected component level but pixel level analysis is becoming more popular.

Figures 6c and 6d show the proportion of algorithms that use a specific color depth and the proportion of algorithms that are evaluated on historical, modern or both types of documents. More and more algorithms make use of color information and are evaluated on historical documents. Currently there is a tie between testing on modern or historical documents. Testing on both types of documents (historical and modern) has a significant share although it seems to become less active.

Figure 7 summarizes the number of publications that have been tested on a given number of different document types (according to the typology of section 2.2) and of different languages for a given data set size. We can see with the big blue and red bubbles that most algorithms that are tested on a data set below 1000 images are only

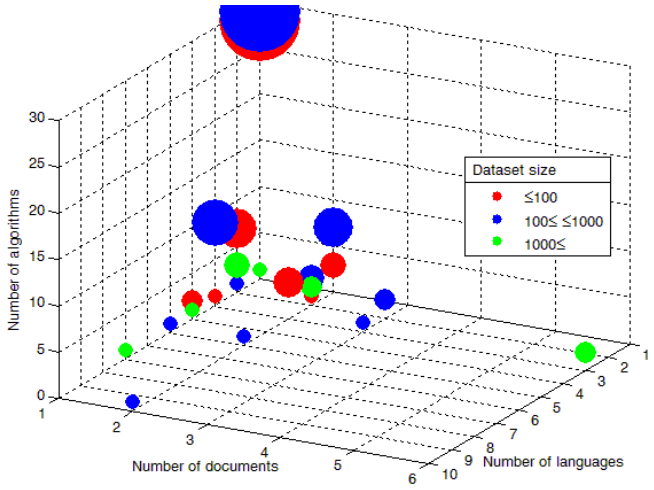


Figure 7: Breakdown of the number of algorithms based on the number of languages and document types in the evaluation corpus. The radius of each bubble is proportional to the number of algorithms which is also the vertical coordinate. The color of the bubbles relates to the size of the corpus.

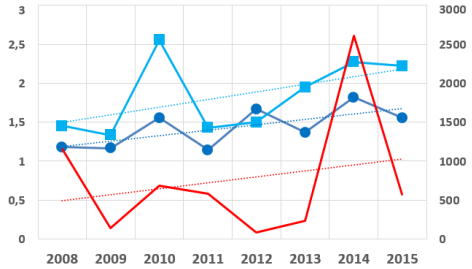


Figure 8: From top to bottom: evolution of the average number of different languages, the average number of different documents and the average data set size (scale on the right). The dotted lines show the linear tendency of these values.

tested on one language and one document type. The algorithms tested on more than 1000 images are mostly tested on one type of document and two languages. The most extensive testings are the blue bubble on the left (2 document types, 10 languages, data set between 100 and 1000 images) and the green bubble on the right (6 document types, 3 languages, data set bigger than 1000 images).

Time wise, Figure 8 shows the tendency for the extensive nature of the evaluation of segmentation algorithms. The data sets on which they are tested tend to include more document types, more languages and tend to be bigger. Finally, nearly all papers are now compared to the state of the art.

7.4. A user's point of view

This survey reviewed the algorithms from a scientific stand point. We would now like to see them from a user/industrial perspective. This would be the place for a qualitative analysis of the algorithms in order to identify which ones may be the most suitable for a given task. However the lack of cross-technique comparison and of diversity in many data sets prevents us from doing such an analysis. Furthermore, the performance differences of the algorithms between experimental and real data would make such an analysis irrelevant.

Instead, we try to answer two questions related to this: what do they do (functionality) and what do they require to do it (requirements)? Table 1 summarizes this for the main algorithms of this survey. The kind of documents that an algorithm can take as input (layout, multi-layered, color depth, text orientation and alignment) is both a functionality and a requirement. A constraining requirement is the need for training. The output (type of output and labels) produced by the algorithm is a functionality. We ordered them from what seemed to be the most to the least critical from a user/industrial point of view. We added the last three columns (data set size, number of languages and of document types) as a guidance to estimate how extensively the algorithms have been tested and thus how reliable they are.

Table 1: Summary of the characteristics of the main document segmentation algorithms.
The type of input is the kind of document that can be processed. Multi-layered indicates whether it can process documents with overlapping content.

Algorithm	Input layout	Multi-layered	Color depth	Labels	Training	Type of output	Text orientation	Text alignment	Data set test size	Nb of languages	Nb of doc types
[72]	Any	Yes	Color	Yes	No	Text lines	Horizontal	Straight	448	1	2
[104]	Any	Yes	Color	Yes	Yes	Regions	Any	Curved	87	1	2
[110]	Any	Yes	Color	Yes	Yes	Regions	Any	Curved	16	2	1
[51]	Any	Yes	Color	No	No	Regions	Any	Curved	2000	2	2
[74]	Any	Yes	Color	No	No	Text lines	Any	Curved	50	1	1
[84]	Any	Yes	Color	No	Yes	Text lines	Any	Straight	21	2	1
[54]	Any	Yes	Gray	Yes	No	Regions	Horizontal	Straight	65	2	1
[112]	Any	Yes	BW	Yes	Yes	Regions	Horizontal	Straight	1000	3	6
[89]	Any	No	Color	Yes	Yes	Regions	Any	Curved	100	1	1

Continued on next page

Table 1 – continued from previous page

Algorithm	Input layout	Multi-layered	Color depth	Labels	Training	Type of output	Text orientation	Text alignment	Data set test size	Nb of languages	Nb of doc types
[109]	Any	No	Color	Yes	Yes	Regions	Any	Curved	49	3	2
[106]	Any	No	Color	Yes	Yes	Regions	Any	Curved	49	3	2
[114]	Any	No	Color	No	Yes	Text lines	Any	Curved	214	3	3
[116]	Any	No	Color	No	Yes	Text lines	Horizontal	Straight	1072	3	6
[100]	Any	No	Gray	Yes	Yes	Regions	Any	Curved	86	3	1
[102]	Any	No	Gray	Yes	Yes	Regions + text lines	Any	Curved	501	1	2
[25]	Any	No	Gray	No	No	Regions	Any	Curved	400	1	2
[57]	Any	No	Gray	No	No	Regions	Any	Curved	25	1	1
[14] (MHS)	Any	No	BW	Yes	No	Regions	Horizontal	Straight	70	1	1
[42]	Any	No	BW	Yes	No	Regions	Horizontal	Straight	3742	2	1
[30]	Any	No	BW	Yes	No	Regions + text lines	Horizontal	Straight	185	2	2
[29]	Any	No	BW	Yes	Yes	Regions	Any	Curved	75	2	2
[56]	Any	No	BW	No	No	Regions	Any	Curved	350	2	3
[64]	Any	No	BW	No	No	Regions	Any	Curved	24339	2	2
[47]	Any	No	BW	No	No	Regions	Any	Straight	100	1	1
[60]	Any	No	BW	No	No	Text lines	Any	Straight	52	1	1
[83]	Any	No	BW	No	No	Text lines	Any	Straight	95	4	1
[78]	Any	No	BW	No	No	Text lines	Horizontal	Curved	649	10	2
[32]	Any	No	BW	No	Yes	Regions	Any	Curved	1425	2	3
[26]	Structured	No	BW	Yes	No	Regions	Any	Curved	100	2	2
[61]	Structured	No	BW	No	No	Regions	Horizontal	Straight	10000	1	1
[35]	Structured	No	BW	No	Yes	Regions	Any	Curved	260	1	1
[93]	Text only	Yes	BW	Yes	Yes	Regions	Any	Curved	500	1	1
[48]	Text only	No	Gray	No	No	Text lines	Horizontal	Straight	45	1	1
[99]	Text only	No	BW	Yes	Yes	Regions	Any	Curved	28	1	1
[41]	Text only	No	BW	No	No	Text lines	Any	Curved	100	1	1
[86]	Text only	No	BW	No	No	Text lines	Any	Straight	853	1	1
[115]	Text only	No	BW	No	No	Text lines	Horizontal	Straight	50	2	1
[67]	Text only	No	BW	No	No	Text lines	Skewed	Curved	202	1	2
[27]	Text only	No	BW	No	No	Text lines	Skewed	Straight	120	2	1
[82]	Text only	No	BW	No	Yes	Text lines	Horizontal	Straight	150	3	1

8. Conclusion

In this survey we give a complete and yet thorough overview of the current state of the art for document image segmentation. We have seen the major trends for the techniques and the challenges that are tackled. It appears that the community is very dynamic and has extended its area of research to every conceivable problem from "simple" fully constrained documents to the most complicated ones that may seem impossible to segment.

The techniques have also evolved. From the first algorithms that were dedicated to a specific type of document, we have moved towards fully flexible algorithms capable of handling a wide range of layouts and that are nearly fully language independent.

Some improvements still remain to be done in particular on three topics: the versatility of the algorithms, their stability and their cross-technique comparison. The training and testing data sets need to be extended to include an even wider range of documents and languages. At the same time, the size balance between the training and the testing set should be reversed in favor of the testing set. These two measures will increase the versatility of the algorithms. The stability of the algorithms has not been much investigated and should be included in future work as current algorithms are not stable. Finally, algorithms should be compared with the best other algorithms that have the same functionality independently from their underlying technique. This will allow a proper identification of the best solution(s) for a given problem.

The recurrent organization of competitions is a very good way to establish a state of the art and a strength of this community. We can be grateful to the organizers and the competitors for contributing in that way to the scientific excellence of the community.

This excellence is also reflected by the strong involvement of industrial partners that recognize the expertise of the document image analysis community. A significant part of the contributions and of the benchmarks was done by them. They regularly provide real world data which allows a more reliable evaluation of the algorithms as their performance is usually much worse on real data sets than on experimental data sets. This is also why we added a functionality/requirement summary. A proper qualitative analysis could not be done because of the remarks that we made on the evaluation of

the algorithms but we hope that in a near future it will become feasible.

For the future we can expect algorithms to make an even wider use of neural networks. The automated discovery of features is also a promising field of research in particular when combined with a multi-scale analysis. It is also clear that future state of the art algorithm will make use of the full color information in order to handle the most challenging documents.

Acknowledgment

This work is financed by the French National Research Agency (ANR) project SHADES referenced under ANR-14-CE28-0022 and by the Town community of La Rochelle.

References

[1] A. M. Treisman, G. Gelade, A feature-integration theory of attention, Cognitive Psychology 12 (1) (1980) 97. arXiv:9605103.

[2] K. Y. Wong, R. G. Casey, F. M. Wahl, Document analysis system, IBM Journal of Research and Development 26 (6) (1982) 647.

[3] G. Nagy, S. Seth, M. Viswanathan, A prototype document image analysis system for technical journals, Computer 25 (7) (1992) 10.

[4] G. Nagy, Twenty years of document image analysis in PAMI, Pattern Analysis and Machine Intelligence 22 (1) (2000) 38.

[5] S. Mao, A. Rosenfeld, T. Kanungo, Document structure analysis algorithms: a literature survey, in: Proc. of DRR X, Elsevier, 2003, p. 197.

[6] A. M. Namboodiri, A. K. Jain, Document structure and layout analysis, in: Digital Document Processing, Springer London, 2007, p. 29.

[7] K. Kise, Page segmentation techniques in document analysis, in: Handbook of Document Image Processing and Recognition, Springer London, 2014, p. 135.

[8] F. Shafait, D. Keysers, T. Breuel, Performance evaluation and benchmarking of six-page segmentation algorithms, Pattern Analysis and Machine Intelligence 30 (6) (2008) 941.

[9] A. Antonacopoulos, S. Pletschacher, D. Bridson, C. Papadopoulos, ICDAR2009 Page segmentation competition, in: Proc. of 10th ICDAR, IEEE, 2009, p. 1370.

[10] A. Antonacopoulos, C. Clausner, C. Papadopoulos, S. Pletschacher, Historical document layout analysis competition, in: Proc. of 11th ICDAR, IEEE, 2011, p. 1516.

[11] B. Lamiroy, D. Lopresti, T. Sun, Document analysis algorithm contributions in end-to-end applications: report on the ICDAR 2011 contest, in: Proc. of 11th ICDAR, IEEE, 2011, p. 1521.

[12] A. Antonacopoulos, C. Clausner, C. Papadopoulos, S. Pletschacher, Competition on Historical Newspaper Layout Analysis (HNLA 2013), in: Proc. of 12th ICDAR, IEEE, 2013, p. 1454.

[13] A. Antonacopoulos, C. Clausner, C. Papadopoulos, S. Pletschacher, Competition on Historical Book Recognition (HBR 2013), in: Proc. of 12th ICDAR, IEEE, 2013, p. 1459.

[14] A. Antonacopoulos, C. Clausner, C. Papadopoulos, S. Pletschacher, ICDAR2015 Competition on recognition of documents with complex layouts, in: Proc. of 13th ICDAR, IEEE, 2015, p. 1151.

[15] M. Murdock, S. Reid, B. Hamilton, J. Reese, ICDAR 2015 Competition on text line detection in historical documents, in: Proc. of 13th ICDAR, IEEE, 2015, p. 1171.

[16] S. R. Vantaram, E. Saber, Survey of contemporary trends in color image segmentation, Journal of Electronic Imaging 21 (4) (2012) 040901.

[17] J. Peng, J. Wang, D. Kong, A new convex variational model for liver segmentation, in: Proc. of 21st ICPR, IEEE, 2012, p. 3754.

[18] S. Katz, G. Leifman, A. Tal, Mesh segmentation using feature point and core extraction, The Visual Computer 21 (8-10) (2005) 649.

[19] D. S. Doermann, A. Rosenfeld, Recovery of temporal information from static images of handwriting, International Journal of Computer Vision 15 (1-2) (1995) 143.

[20] P. T. Daniels, A Calligraphic Approach to Aramaic Paleography, Journal of Near Eastern Studies 43 (1) (1984) 55.

[21] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, S. Süsstrunk, SLIC Superpixels compared to state-of-the-art superpixel methods, Pattern Analysis and Machine Intelligence 34 (11) (2012) 2274.

[22] S. Eskenazi, P. Gomez-Krämer, J.-m. Ogier, Evaluation of the stability of four document segmentation algorithms, in: Proc. of DAS XII, IEEE, 2016, p. 1.

[23] A. Rebelo, I. Fujinaga, F. Paszkiewicz, A. R. S. Marcal, C. Guedes, J. S. Cardoso, Optical music recognition: state-of-the-art and open issues, International Journal of Multimedia Information Retrieval 1 (3) (2012) 173.

[24] X. Du, W. Pan, T. D. Bui, Text line segmentation in handwritten documents using Mumford-Shah model, Pattern Recognition 42 (12) (2008) 3136.

[25] N. Journet, J.-Y. Ramel, R. Mullot, V. Eglin, Document image characterization using a multiresolution analysis of the texture: application to old documents, IJDAR 11 (1) (2008) 9.

[26] A. Lemaitre, J. Camillerapp, B. Coüasnon, Multiresolution cooperation makes easier document structure recognition, IJDAR 11 (2) (2008) 97.

[27] G. Louloudis, B. Gatos, I. Pratikakis, C. Halatsis, Text line and word segmentation of handwritten documents, Pattern Recognition 42 (12) (2009) 3169.

[28] G. Lazzara, R. Levillain, G. Thierry, Y. Jacquelet, J. Marquegnies, A. Cr, The SCRIBO Module of the Olena platform : a free software framework for document image analysis, in: Proc. of 11th ICDAR, IEEE, 2011, p. 252.

[29] F. C. Fern, O. R. Terrades, Document segmentation using relative location features, in: Proc. of 21st ICPR, IEEE, 2012, p. 1562.

[30] I. Konya, Adaptive methods for robust document image understanding, Ph.D. thesis, Rheinischen Friedrich-Wilhelms-Universität (2012).

[31] M. Diem, F. Kleber, R. Sablatnig, Text line detection for heterogeneous documents, in: Proc. of 12th ICDAR, IEEE, 2013, p. 743.

[32] L. Liu, Y. Lu, C. Y. Suen, Near-duplicate document image matching: A graphical perspective, Pattern Recognition 47 (4) (2014) 1653.

[33] J. Camillerapp, C. D. Beaulieu, A generic method for structure recognition of handwritten mail documents, in: Proc. of DRR XV, SPIE, 2008, p. 68150W.

[34] C. Carton, A. Lemaitre, B. Coüasnon, Automatic and interactive rule inference without ground truth, in: Proc. of 13th ICDAR, IEEE, 2015, p. 696.

[35] F. Shafait, J. Beusekom, D. Keysers, T. M. Breuel, Structural mixtures for statistical layout analysis, in: Proc. of DAS VIII, IEEE, 2008, p. 415.

[36] F. Shafait, J. van Beusekom, D. Keysers, T. M. Breuel, Background variability modeling for statistical layout analysis, in: Proc. of 19th ICPR, IEEE, 2008, p. 1.

[37] N. Ouwayed, A. Belaïd, Multi-oriented text line extraction from handwritten arabic documents, in: Proc. of DAS VIII, IEEE, 2008, p. 339.

[38] Z. Liu, H. Zhou, N. Yang, Semi-supervised learning for text-line detection, Pattern Recognition Letters 31 (11) (2010) 1260.

[39] N. Ouwayed, A. Belaïd, F. Auger, General text line extraction approach based on locally orientation estimation, in: Proc. of DRR XVII, SPIE, 2010, p. 75340B.

[40] V. Papavassiliou, T. Stafylakis, V. Katsouros, G. Carayannis, Handwritten document image segmentation into text lines and words, Pattern Recognition 43 (1) (2010) 369.

[41] N. Ouwayed, A. Belaïd, A general approach for multi-oriented text line extraction of handwritten documents, IJDAR 15 (4) (2012) 297.

[42] T. C. Bockholt, G. D. C. Cavalcanti, C. a. B. Mello, Document image retrieval with morphology-based segmentation and features combination, in: Proc. of DRR XVIII, SPIE, 2011, p. 787415.

[43] S. S. Bukhari, F. Shafait, T. M. Breuel, High performance layout analysis of Arabic and Urdu document images, in: Proc. of 11th ICDAR, IEEE, 2011, p. 1275.

[44] S. S. Bukhari, F. Shafait, T. M. Breuel, Improved document image segmentation algorithm using multiresolution morphology, in: Proc. of DRR XVIII, SPIE, 2011, p. 78740D.

[45] A. Lemaitre, J. Camillerapp, B. Coüasnon, A perceptive method for handwritten text segmentation, in: Proc. of DRR XVIII, SPIE, 2011, p. 78740C.

[46] Y. Tang, X. Wu, W. Bu, Text line segmentation based on matched filtering and top-down grouping for handwritten documents, in: Proc. of DAS XI, IEEE, 2014, p. 365.

[47] S. Ferilli, M. Biba, F. Esposito, T. M. Basile, A distance-based technique for non-Manhattan layout analysis, in: Proc. of 10th ICDAR, IEEE, 2009, p. 231.

[48] Z. Shi, S. Setlur, V. Govindaraju, A steerable directional local profile technique for extraction of handwritten arabic text lines, in: Proc. of 10th ICDAR, IEEE, 2009, p. 176.

[49] K. Chen, F. Yin, C.-l. Liu, Hybrid page segmentation with efficient whitespace rectangles extraction and grouping, in: Proc. of 12th ICDAR, IEEE, 2013, p. 958.

[50] Y. Wang, Y. Zhou, Z. Tang, Comic frame extraction via line segments combination, in: Proc. of 13th ICDAR, IEEE, 2015, p. 856.

[51] E. Carel, J.-c. Burie, V. Courboulay, J.-m. Ogier, V. Poulain d'Andecy, Multiresolution approach based on adaptive superpixels for administrative documents segmentation into color layers, in: Proc. of 13th ICDAR, IEEE, 2015, p. 566.

[52] J. Kumar, W. Abd-Almageed, Handwritten arabic text line segmentation using affinity propagation, in: Proc. of DAS IX, IEEE, 2010, p. 135.

[53] M. Ziaratban, K. Faez, An adaptive script-independent block-based text line extraction, in: Proc. of 20th ICPR, IEEE, 2010, p. 249.

[54] Y. L. Chen, B. F. Wu, A multi-plane approach for text segmentation of complex document images, Pattern Recognition 42 (7) (2009) 1419.

[55] M. Agrawal, D. Doermann, Voronoi++: a dynamic page segmentation approach based on Voronoi and Docstrum features, in: Proc. of 10th ICDAR, IEEE, 2009, p. 1011.

[56] M. Agrawal, D. Doermann, Context-aware and content-based dynamic Voronoi page segmentation, in: Proc. of DAS IX, IEEE, 2010, p. 73.

[57] M. Mehri, P. Gomez-Krämer, P. Héroux, A. Boucher, R. Mullot, Texture feature evaluation for segmentation of historical document images, in: Proc.of 2nd International Workshop on Historical Document Imaging and Processing (HIP), ACM Press, 2013, p. 102.

[58] M. Mehri, P. Heroux, P. Gomez-Kramer, A. Boucher, R. Mullot, A pixel labeling approach for historical digitized books, in: Proc. of 12th ICDAR, IEEE, 2013, p. 817.

[59] C. Clausner, A. Antonacopoulos, S. Pletschacher, A robust hybrid approach for text line segmentation, in: Proc. of 21st ICPR, IEEE, 2012, p. 335.

[60] C. Faure, N. Vincent, Simultaneous detection of vertical and horizontal text lines based on perceptual organisation, in: Proc. of DRR XVI, SPIE, 2009, p. 72470M.

[61] D. Gaceb, V. Eglin, F. Lebourgeois, H. Emptoz, Application of graph coloring in physical layout segmentation, in: Proc. of 19th ICPR, IEEE, 2008, p. 1.

[62] J. Kumar, L. Kang, D. Doermann, W. Abd-Almageed, Segmentation of handwritten textlines in presence of touching components, in: Proc. of 11th ICDAR, IEEE, 2011, p. 109.

[63] X. Liu, H. Fu, Y. Jia, Gaussian mixture modeling and learning of neighboring characters for multilingual text extraction in images, Pattern Recognition 41 (2) (2008) 484.

[64] L. Liu, Y. Lu, C. Y. Suen, Novel global and local features for near-duplicate document image matching, in: Proc. of 22nd ICPR, IEEE, 2014, p. 4624.

[65] V. Malleron, V. Eglin, H. Emptoz, S. Dord-Crouslé, P. Régnier, Text lines and snippets extraction for 19th century handwriting documents layout analysis, in: Proc. of 10th ICDAR, IEEE, 2009, p. 1001.

[66] D. M. Oliveira, R. D. Lins, G. Torreão, A new method for text-line segmentation for warped documents, in: Proc. of 7th International Conference on Image Analysis and Recognition (ICIAR), Springer Berlin Heidelberg, 2010, p. 398.

[67] D. Oliveira, R. Lins, G. Torreao, J. Fan, M. Thielo, An efficient algorithm for segmenting warped text-lines in document images, in: Proc. of 12th ICDAR, IEEE, 2013, p. 250.

[68] C. Rigaud, N. Tsopze, J.-C. Burie, J.-M. Ogier, Robust Frame and Text Extraction from Comic Books, in: Graphics Recognition. New Trends and Challenges, Vol. 7423 LNCS, Springer, 2013, p. 129.

[69] P. P. Roy, U. Pal, J. Lladós, Text line extraction in graphical documents using background and foreground information, IJDAR 15 (3) (2012) 227.

[70] F. Yin, C. L. Liu, Handwritten Chinese text line segmentation by clustering with distance metric learning, Pattern Recognition 42 (12) (2009) 3146.

[71] A. Winder, T. Andersen, E. H. B. Smith, Extending page segmentation algorithms for mixed-layout document processing, in: Proc. of 11th ICDAR, IEEE, 2011, p. 1245.

[72] A. Ouji, Y. Leydier, F. LeBourgeois, A hierarchical and scalable model for contemporary document image segmentation, Pattern Analysis and Applications 16 (4) (2013) 679.

[73] F. Zirari, A. Ennaji, S. Nicolas, D. Mammass, A document image segmentation system using analysis of connected components, in: Proc. of 12th ICDAR, IEEE, 2013, p. 753.

[74] A. Clavelli, D. Karatzas, Text segmentation in colour posters from the spanish civil war era, in: Proc. of 10th ICDAR, IEEE, 2009, p. 181.

[75] S. S. Bukhari, F. Shafait, T. M. Breuel, Segmentation of curled textlines using active contours, in: Proc. of DAS VIII, Springer, 2008, p. 270.

[76] S. S. Bukhari, F. Shafait, T. M. Breuel, Coupled snakelet model for curled textline segmentation of camera-captured document images, in: Proc. of 10th ICDAR, IEEE, 2009, p. 61.

[77] S. S. Bukhari, F. Shafait, T. M. Breuel, Text-line extraction using a convolution of isotropic Gaussian filter with a set of line filters, in: Proc. of 11th ICDAR, IEEE, 2011, p. 579.

[78] S. S. Bukhari, F. Shafait, T. M. Breuel, Towards generic text-line extraction, in: Proc. of 12th ICDAR, IEEE, 2013, p. 748.

[79] S. S. Bukhari, F. Shafait, T. M. Breuel, Coupled snakelets for curled text-line segmentation from warped document images, IJDAR 16 (1) (2013) 33.

[80] S. S. Bukhari, F. Shafait, T. M. Breuel, Script-independent handwritten textlines segmentation using active contours, in: Proc. of 10th ICDAR, IEEE, 2009, p. 446.

[81] S. S. Bukhari, F. Shafait, T. M. Breuel, Ridges based curled textline region detection from grayscale camera-captured document, in: Proc. of Computer Analysis of Images and Patterns (CAIP), Springer-Verlag, 2009, p. 173.

[82] J. Ryu, H. I. Koo, N. I. Cho, Language-independent text-line extraction algorithm for handwritten documents, Signal Processing Letters 21 (9) (2014) 1115.

[83] M. Kim, I.-S. Oh, Script-free text line segmentation using interline space model for printed document images, in: Proc. of 11th ICDAR, IEEE, 2011, p. 1354.

[84] X. Shen, C. Liu, X. Ding, Y. Zou, Text line extraction in free-style document, in: Proc. of DRR XVI, SPIE, 2009, p. 72470L.

[85] H. I. Koo, I. Cho Nam, State estimation in a document image and its application in text block identification and text line extraction, in: Proc. of 11th ECCV, Vol. 6312, Springer, 2010, p. 421.

[86] F. Yin, C.-l. Liu, A variational bayes method for handwritten text line segmentation, in: Proc. of 10th ICDAR, IEEE, 2009, p. 436.

[87] F. Cruz, O. R. Terrades, EM-based layout analysis method for structured documents, in: Proc. of 22nd ICPR, IEEE, 2014, p. 315.

[88] M. Benjelil, S. Kanoun, R. Mullot, A. M. Alimi, Complex documents images segmentation based on steerable pyramid features, IJDAR 13 (3) (2010) 209.

[89] M. Baechler, R. Ingold, Multi resolution layout analysis of medieval manuscripts using dynamic MLP, in: Proc. of 11th ICDAR, IEEE, 2011, p. 1185.

[90] X. Peng, S. Setlur, V. Govindaraju, R. Sitaram, K. Bhuvanagiri, Markov random field based text identification from annotated machine printed documents, in: Proc. of 10th ICDAR, IEEE, 2009, p. 431.

[91] P. Sarkar, E. Saund, J. Lin, Classifying foreground pixels in document images, in: Proc. of 10th ICDAR, IEEE, 2009, p. 641.

[92] S. S. Bukhari, M. I. A. Al Azawi, F. Shafait, T. M. Breuel, Document image segmentation using discriminative learning over connected components, in: Proc. of DAS IX, IEEE, 2010, p. 183.

[93] S. J. Pinson, W. a. Barrett, Connected component level discrimination of handwritten and machine-printed text using eigenfaces, in: Proc. of 11th ICDAR, IEEE, 2011, p. 1394.

[94] X. Peng, S. Setlur, V. Govindaraju, S. Ramachandrula, Using a boosted tree classifier for text segmentation in hand-annotated documents, Pattern Recognition Letters 33 (7) (2012) 943.

[95] M. Benjlaiel, R. Mullot, A. M. Alimi, Multi-oriented handwritten annotations extraction from scanned documents, in: Proc. of DAS XI, IEEE, 2014, p. 126.

[96] V. P. Le, N. Nayef, M. Visani, J.-m. Ogier, C. D. Tran, Text and non-text segmentation based on connected component features, in: Proc. of 13th ICDAR, IEEE, 2015, p. 1096.

[97] T. Yamaguchi, M. Maruyama, Feature extraction for document image segmentation by pLSA model, in: Proc. of DAS VIII, IEEE, 2008, p. 53.

[98] D. Hebert, T. Paquet, S. Nicolas, Continuous CRF with multi-scale quantization feature functions application to structure extraction in old newspaper, in: Proc. of 11th ICDAR, IEEE, 2011, p. 493.

[99] X. Peng, S. Setlur, V. Govindaraju, R. Sitaram, Handwritten text separation from annotated machine printed documents using Markov random fields, IJDAR 16 (1) (2013) 1.

[100] G. Zhong, M. Cheriet, Tensor representation learning based image patch analysis for text identification and recognition, Pattern Recognition 48 (4) (2015) 1211.

[101] F. Montreuil, E. Grosicki, L. Heutte, S. Nicolas, Unconstrained handwritten document layout extraction using 2D conditional random fields, in: Proc. of 10th ICDAR, IEEE, 2009, p. 853.

[102] M. Diem, F. Kleber, R. Sablatnig, Text classification and document layout analysis of paper Fragments, in: Proc. of 11th ICDAR, IEEE, 2011, p. 854.

[103] A. Garz, R. Sablatnig, M. Diem, Layout analysis for historical manuscripts using SIFT features, in: Proc. of 11th ICDAR, IEEE, 2011, p. 508.

[104] S.-Y. Wang, H. Baird, C. An, Document content extraction using automatically discovered features, in: Proc. of 10th ICDAR, IEEE, 2009, p. 1076.

[105] M. Baechler, M. Liwicki, R. Ingold, Text line extraction using DMLP classifiers for historical manuscripts, in: Proc. of 12th ICDAR, IEEE, 2013, p. 1029.

[106] H. Wei, M. Baechler, F. Slimane, R. Ingold, Evaluation of SVM, MLP and GMM classifiers for layout analysis of historical documents, in: Proc. of 12th ICDAR, IEEE, 2013, p. 1220.

[107] K. Chen, H. Wei, M. Liwicki, J. Hennebert, R. Ingold, Robust text line segmentation for historical manuscript images using color and texture, in: Proc. of 22nd ICPR, IEEE, 2014, p. 2978.

[108] A. Fischer, M. Baechler, A. Garz, M. Liwicki, R. Ingold, A combined system for text line extraction and handwriting recognition in historical documents, in: Proc. of DAS XI, IEEE, 2014, p. 71.

[109] K. Chen, M. Seuret, M. Liwicki, J. Hennebert, R. Ingold, Page segmentation of historical document images with convolutional autoencoders, in: Proc. of 13th ICDAR, IEEE, 2015, p. 1011.

[110] R. Garg, E. Hassan, S. Chaudhury, M. Gopal, A CRF based scheme for overlapping multi-colored text graphics separation, in: Proc. of 11th ICDAR, IEEE, 2011, p. 1215.

[111] R. W. Smith, Hybrid page layout analysis via tab-stop detection, in: Proc. of 10th ICDAR, IEEE, 2009, p. 241.

[112] P. Barlas, S. Adam, C. Chatelain, T. Paquet, A typed and handwritten text block segmentation system for heterogeneous and complex documents, in: Proc. of DAS XI, IEEE, 2014, p. 46.

[113] A. Asi, R. Cohen, K. Kedem, J. El-sana, Simplifying the reading of historical manuscripts, in: Proc. of 13th ICDAR, IEEE, 2015, p. 826.

[114] L. Wang, W. Fan, J. Sun, S. Naoi, Text line extraction in document images, in: Proc. of 13th ICDAR, IEEE, 2015, p. 191.

[115] N. Stamatopoulos, B. Gatos, S. J. Perantonis, A method for combining complementary techniques for document image segmentation, Pattern Recognition 42 (12) (2009) 3158.

[116] B. Moysset, C. Kermorvant, C. Wolf, Paragraph text segmentation into lines with recurrent neural networks, in: Proc. of 13th ICDAR, IEEE, 2015, p. 456.

[117] B. Coüasnon, DMOS, a generic document recognition method: application to table structure analysis in a general and in a specific way, IJDAR 8 (2-3) (2006) 111.

[118] D. S. Bloomberg, Multiresolution morphological approach to document image analysis, in: Proc. of 1st ICDAR, IEEE, 1991, p. 963.

[119] D. H. Douglas, T. K. Peucker, Algorithms for the reduction of the number of points required to represent a digitized line or its caricature, Cartographica: The International Journal for Geographic Information and Geovisualization 10 (2) (1973) 112.

[120] K. Kise, A. Sato, M. Iwata, Segmentation of page images using the area voronoi diagram, Computer Vision and Image Understanding 70 (3) (1998) 370.

[121] N. Stamatopoulos, B. Gatos, G. Louloudis, U. Pal, A. Alaei, ICDAR 2013 Handwriting segmentation contest, in: Proc. of 12th ICDAR, IEEE, 2013, p. 1402.

[122] S. Belongie, J. Malik, J. Puzicha, Shape matching and object recognition using shape contexts, Pattern Analysis and Machine Intelligence 24 (24) (2002) 509.

[123] E. Grosicki, H. E. Abed, ICDAR 2009 Handwriting recognition competition, in: Proc. of 10th ICDAR, IEEE, 2009, p. 1398.

[124] E. Grosicki, H. El-Abed, ICDAR 2011 French handwriting recognition competition, in: Proc. of 11th ICDAR, IEEE, 2011, p. 1459.

[125] H. S. Baird, M. R. Casey, Towards versatile DAS, in: Proc. of DAS VII, Springer Berlin Heidelberg, 2006, p. 280.