Usual Scenarios and Suitable Approaches Used in Automatic Merge of Scanned Images

Costin A. Boiangiu, Andrei C. Spataru, Andrei I. Dvornic and Ion Bucur

***Abstract*—**The most important step in automatic content conversion is the preprocessing step. Having a very good scanned document is almost a safe bet that the document will have the content extracted with a good confidence level. The current paper describes some preprocessing methods which can be used in large images that must be scanned by pieces because they simply don't fit entirely the scanner area. We propose a novel digital multi-pass scanned image merge scheme for newspapers or other historical documents, allowing further content exploitation in an efficient way. The goal is to combine multiple images with or without overlapping fields of view in order to produce segmented panorama or high resolution document.

Keywords—automatic content conversion, image merging, non-overlapping pages, page overlap, text characteristics.

I. Introduction

The future of cultural development throughout the world is demanding more than ever for a larger distribution and a better preservation in time of knowledge. Since in centuries the only support available for spreading and conservation of information has been paper, almost all legacy materials have the form of printed paper documents. Their degradation in time, as well as the limited availability of some of these documents is demanding for better techniques of document analysis and digitization in order to ensure preservation and access to the available material.

A vast amount of valuable historical information is contained in old-time newspapers and magazines. Hence, searching through a newspaper for particular information can be a real challenge, as this kind of documents are usually loaded with a large variety and amount of data. Libraries, apart from holding very few copies of historical newspapers, cannot make them available through interlibrary loan due to their degradation and sensitivity.

The available solution of microfilming such documents offers mass-user access to such papers but yet, unless the newspapers have been indexed in some way, the search for some desired information can be a tiresome quest. Local newspapers are suggested as being prime candidates for digitization, due to their vast amount of hard-to-access information. At its best, a digitized newspaper makes all its information searchable and accessible to anyone with internet access. But newspapers present some distinct challenges on the road to a successful outcome.

One of the main issues that arise during the digitization of such documents is the impossibility of fitting some of them entirely in available scanners because of their large size [10]. As a result of more and more improvements in the scanning technology, large-scale digitization projects are now possible. These kinds of projects are usually divided in two main modules:

- ✓ the preprocessing module - deals with the actual scanning and creation of a digital version of each work
- ✓ document understanding, content conversion and analysis module – interprets the images acquired by the preprocessing module by extracting structured content information for each document sub-part

Older scanning technologies used to require processed documents to be unbound so that each page could be fed automatically through a scanning machine. This procedure was very damaging for the document if they were not unbound, as the flattening on the scanner glass would corrupt the spines and binding of books (for example). Apart from that, this procedure required a person to manually position each page on the scanning surface. Modern scanning technologies perform image acquisition using digital cameras which point at open bound documents. Software products have been developed to work hand in hand with the scanning machines, such that to adjust the curvature of open pages in digital images, making the image flat even though the source document is not. Apart from that, software products can allow the scanning of text and illustrations altogether by adjusting resolution and other characteristics as needed. Images are usually captured at 600 DPI, the scan rates available on the market ranging from 1200 to about 4000 pages per hour. The scanning process usually produces very large files (for example one page can be 25 megabytes while a book can easily exceed 7 gigabytes), but the improvement in hard drive technology over the last decade means that the scanning systems can handle the files that result from the scanning activity.

As speed and quality of the digitization process are emphasized during such projects, there is the need of a

Paper submitted on December 10, 2008 for review.

Costin-Anton Boiangiu, Ion Bucur are with the Faculty of Automatic Control and Computers, "Politehnica" University of Bucharest, Bucharest, Splaiul Independentei 313, Romania, Postal Code 060042 (e-mail: Costin.Boiangiu@cs.pub.ro, Ion.Bebe.Bucur@gmail.com).

Andrei-Cristian Spataru, Andrei-Iulian Dvornic are with the Faculty of Engineering Taught in Modern Languages, "Politehnica" University of Bucharest, Bucharest, Splaiul Independentei 313, Postal Code 060042, Romania(e-mail:Andrei.Spataru@yahoo.com, Andrei.Dvornic@yahoo.co.uk).

laborious preprocessing phase that can solve these kinds of problems. To this goal, in this paper we explore a novel approach for improving the digitization process by means of trying to reconstruct documents from cropped parts obtained during the scanning process. Experimental results are presented in order to prove the validity of the presented techniques, as well as offer a more detailed description of each procedure. The presented multi-pass scanned image merge scheme assumes that the input documents are binary images (black and white documents).

A. Related work

Current image stitching algorithms used in the industry are combining multiple images based on the graphical projections of image segments that have been taken from the corresponding same points in the two parts of the document. Rectilinear projection algorithms consider the two images as viewed on a 2D plane, whereas algorithms using cylindrical projection consider panoramas in the projection that are meant to be viewed as though the image is wrapped into a cylinder and viewed from within. When viewed on a 2D plane, horizontal lines appear curved while vertical lines remain straight. Spherical projection is similar to the cylindrical one, panoramas being meant to be viewed as though the image is wrapped into a sphere. Considering a 2D plane in this case, horizontal lines appear curved as in a cylindrical projection, while vertical lines curve as they get closer to the poles of the sphere.

One of the essential problems that must be addressed in the case of development of a merge algorithm is how to blend overlapping sections, even in the presence of parallax, exposure differences or lens distortions. Apart from that, there is the issue of finding preprocessing methods which can be used on large images that must be scanned by pieces. This can be the result of the fact that such documents simply don't fit entirely the scanner area. Methods resulting in image merging of such documents have been developed for a long time now; however, most of the approaches focus on merging the image based on pixel-content comparison, which is a very slow process for high resolution images (a great number of picture-per-picture difference comparisons are required). This is simply because there is a difference in rotation angles between the two images and not only in planar displacement. The presented approaches are based on a new idea, of retrieving specific point-features in one image and "match" them in the other image.

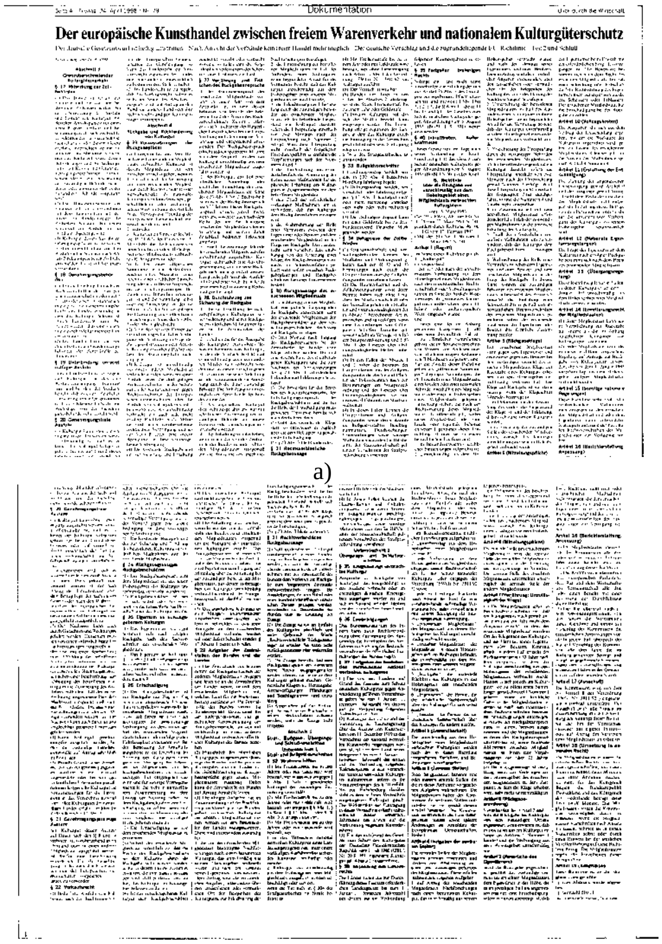
II. Problem Statement

The proposed algorithms for document-merging and recovery are based on two main scenarios that are encountered throughout the industry. The first one is that of intersecting components of a document. This is the case of scanned parts which are overlapping one another and contain common information that can be used during the merging process. The intersections that occur in such cases are included in one of the following cases:

- ✓ horizontally-aligned overlapping – e.g. the case of the top/bottom parts of a newspaper
- ✓ vertically-aligned overlapping – e.g. case of the left/right halves of a newspaper

The second case that is discussed in this paper is that of non-intersecting parts of a document that also need to be merged. The most common situations in which these problems are encountered are vertical separation of document components that need to be aligned and joined together before the actual digitization process.

The first case, describing intersecting parts of a page (usually halves) is the most common met in practice, even though higher separation orders can be encountered for very large documents like maps.



Dokumentation

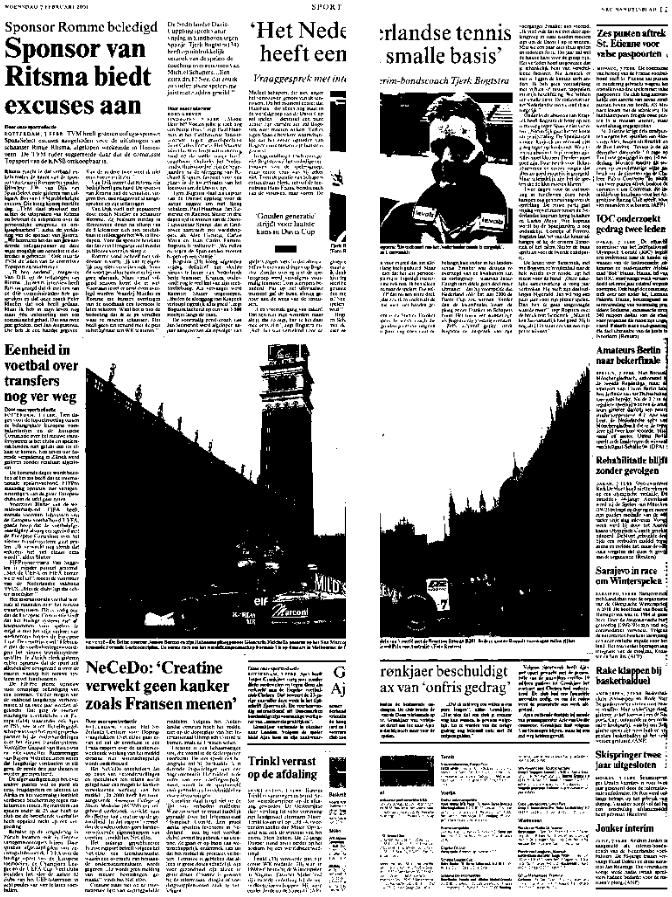
Der europäische Kunsthandel zwischen freiem Warenverkehr und nationalem Kulturgüterschutz

a)

b)

Fig. 1 overlapping sections of a newspaper
a) top half of the newspaper page
b) bottom half of the newspaper page

The main standard procedures used in the field of document analysis, input binarization and low-level component detection have a huge impact on the page-merging techniques presented in the following. The main purpose of the binarization stage is to make a clear difference between the background and the foreground (text, images, etc) of a document, removing during the process any kind of noise that obstructs its legibility. The binary image is desirable in the further analysis stages [12], as it is a better input for an OCR engines, as well as for the geometrical routines that extract the text characteristics and structure. Traditional binarization methods can be divided into two main categories: global thresholding methods and local thresholding methods. The global approaches apply an all-round threshold on the entire image and classify background and foreground pixels accordingly. This approach has the advantage of being simple and fast, but problems occur when applying it on images that have uneven background noise (as in the case of aged books newspapers) [3][13]. Local thresholding methods classify pixels into background or foreground according to a local threshold determined by their neighboring pixels; an adaptive approach but with a significantly lower time-performance and higher computational cost [4].


Sponsor Romme beledigd
Sponsor van Ritsma biedt excuses aan
'Het Nede heeft een
rlandse tennis smalle basis'
Zes punten aftrek St. Etienne voor valse paspoorten
IOC onderzoekt gedrag twee leden
Eenheid in voetbal over transfers nog ver weg
Amateurs Berlin naar bekerfinale
Rehabilitatie blijft zonder gevolgen
Sarajevo in race om Winterspelen
NeCeDo: 'Creatine verwekt geen kanker zoals Fransen menen'
Trinkl verrast op de afdaling
rønkjaer beschuldigt ax van 'onfris gedrag'
Rake klappen bij basketbalduel
Skispringer twee jaar uitgesloten
Jonker interim

a) b)

Fig. 2 non-overlapping sections of a newspaper
a) left half of the newspaper page
b) right half of the newspaper page

From another point of view, binarization approaches can be divided as follows:

- ✓ general-purpose methods - are able to deal with any type of input, without taking into consideration any specific characteristics of the document
- ✓ specifically-designed methods - take advantage of document's properties; these methods in many cases are a variation of general-purpose methods.

Low-level components extraction is usually performed by first extracting all horizontal run-length sequences of black pixels (called segments) found in the binary input. These segments, described by their row number and the numbers of the bounding columns, are grouped into clusters of connected black pixels (or entities). However, these groups of pixels cannot be considered characters for now – they could very well represent noise, punctuation marks or page separators. In order to differentiate between characters and other foreground elements in a document a series of filters is applied over the set of the previously extracted entities. The filters are applied in a predetermined order and are addressing common input scenarios (e.g. broken characters that must be recovered from pieces, punctuation marks etc.). Here are some examples of the most common filters used for performing the above task:

- ✓ check whether a cluster is completely included into another one and corrects this situation
- ✓ check whether two clusters inside the same line, one above another can be joined
- ✓ exclude separator lines or punctuation marks from the character set by applying a threshold based on cluster-width.

Some of the above filtering methods make use of line detection techniques which are described later in this paper.

III. Overlapping components

A. Algorithm description

This section presents our proposal for page-merging in the case of document building blocks that are intersecting. In order to achieve this, a strip of comparison is selected in each unit (document part), in order to be used as a reference during the recovery process. A ratio of 50% of the recoverable dimension for each building block (the height in the case of horizontally-aligned overlapping, width otherwise) is considered as a safe strip size for the algorithm.

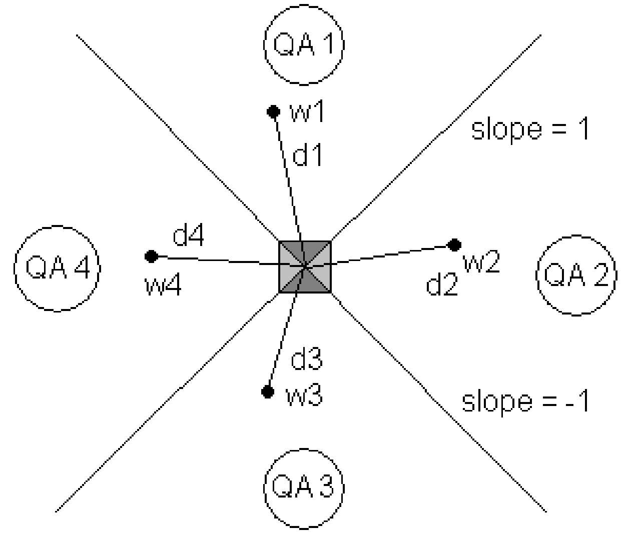


Fig.3 finding the four distances associated to the current entity

In order to correctly match entities in the two comparison bands and ensure the correctness of the results, a number of measurements must be performed. The notion of

"Neighborhood-Weighted Entity" is introduced, meaning that for each entity in the band, eight pairs of values are associated, these values being of two types: weights and distances, or, more precisely, rotationally invariant weights and smallest distances in four directions.

The previous figure explains the process of choosing the four distance values associated to the current entity:

In Fig.3 the current entity is represented by the centre square. The neighboring entities are represented by the four circles and are denoted by QA1, QA2, QA3 and QA4. Two axes are chosen, having the origin in the current entity, and the slopes +1 and -1 with respect to the normal position of the image. These axes form quadrant areas (QA) that are used as search zones for the smallest distance between the current and neighboring entities. The reason for choosing these slopes for the axes, as opposed to a regular layout, is to make the detection of distances as rotationally invariant as possible, by avoiding the misplacing of a relevant neighboring entity in another quadrant area. This is important because we expect the angle difference between the two images to be small (lower than five degrees in absolute) and every one of the four closest neighbors will tend to "keep" its quadrant area.

After finding the closest neighboring entities in the quadrant areas, their distances (d1...d4) to the current entity are recorded. The next step is associating weights to these four neighboring entities.

The weight of an entity can be chosen from a number of different geometrical measurements, as long as these measurements are rotationally invariant. Some examples of such measurements include the number of pixels of an entity (currently in use in the algorithm), or the area of the entity's convex hull. In following, the weights of the four closest neighboring entities are normalized, using the formula

$$w_i = \frac{w_i}{w},$$

where w_i represents the weight of a neighboring entity, $i = \overline{1,4}$, and w represents the weight of the current entity.

By performing these measurements, every entity in the two comparison bands will have associated eight values, (d1|w1 ... d4|w4). These values will be used for the comparison of entities in the two comparison bands.

The next step in the algorithm is performing a matching of entities from the two bands, based on the weights and distances associated to each entity. The "Neighborhood-Weighted Entity" is regarded in the matching process as having four weights only (those of the four neighbors normalized by performing a division with the centre weight). This is useful especially to compensate the effect of automatically adjustment of scanner contrast in which case many documents are darkened (or lightened) by a heuristically approach regarding their content. Entering the reports of weights into the match equation instead of the weights alone may compensate in most of the cases for this self-adjusting contrast side effect.

After the components-matching is completed for all entities in the selected strips, three histograms are computed in order to find the parameters of the transfer function that must be used in order to blend the building blocks together. In the case of documents with special characteristics (e.g. large size or character density) not all the connected components from the foreground are subject of matching, but only a limited number of them, considered reliable enough, in order to achieve acceptable results.

In following, the three types of histograms will be exemplified and explained. The histograms were computed by analyzing Fig.1.

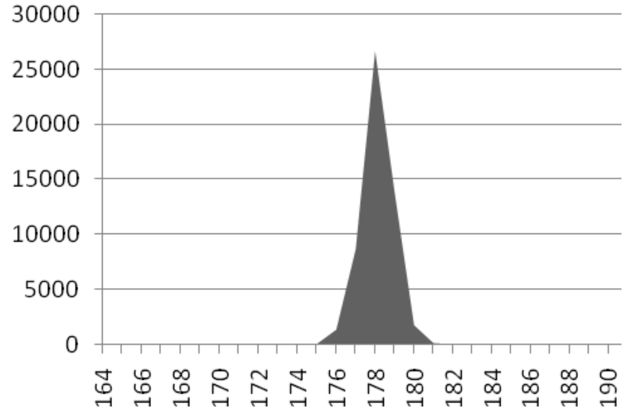


Fig.4 example of X-Offset histogram from the top-bottom overlapping scenario

The X-Offset histogram, shown in Fig.4, represents the number of appearances of certain values of the x-axis translation. These values are obtained by computing the difference in x coordinate between two matching entities in different comparison bands. As expected, this representation exhibits a clear peak around the most common x-axis offset value, because the whole page has been shifted by that value when it was scanned. The peak value (in this example 178 pixels) is the first parameter of the rotation-translation function used to blend the images.

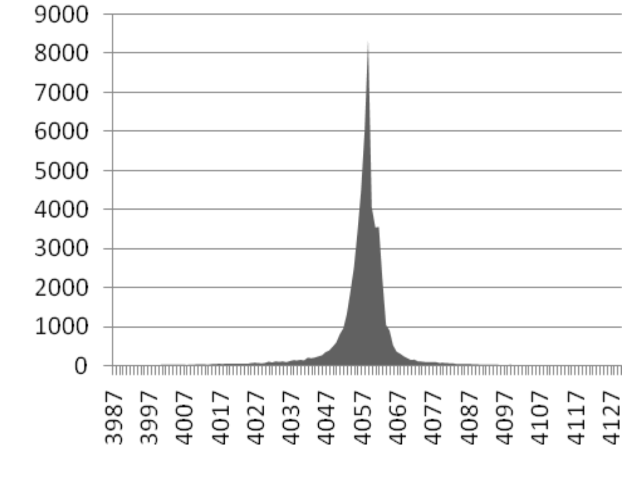


Fig.5 example of Y-Offset histogram from the top-bottom overlapping scenario

The Y-Offset histogram performs similarly to the X-Offset histogram, but on the Y axis. The difference in y-axis coordinate between matching entities in the two comparison bands is computed and plotted on the histogram, with the peak value being the most common one. The significant difference here is that the peak is found at much greater offsets than the X-Offset histogram (in this example the y-axis offset is 4055 pixels). This is due to the fact that on one image the entity is on the bottom of the page, while on the other image the matching entity is on the top of the page (in the case of top-bottom separation). When dealing with left-right separated images, the X-Offset will be significantly greater than the Y-Offset.

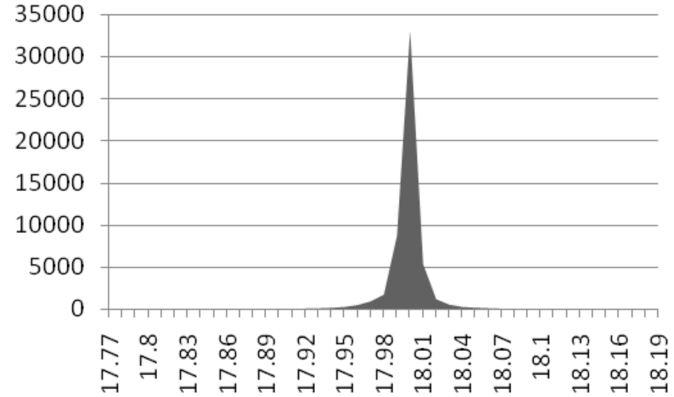


Fig.6 example of Angle histogram from the top-bottom overlapping scenario

In order to populate the Angle histogram with values, segments are considered between all pairs of entities from the same comparison band, along with their matching segments in the other comparison band. The angles that blend corresponding segments from the two comparison bands are plotted using the Angle histogram, with the histogram peak representing the most common angle. The angle between corresponding segments in different comparison bands is shown in Fig.8.

Coupled with the X-Offset and Y-Offset histogram peaks, the Angle histogram peak will result in the parameters of the rotation-translation function that will blend the two image pieces.

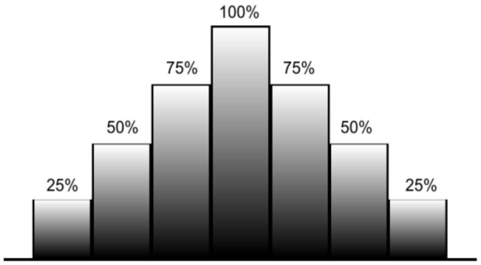


Fig.7 triangle filter

The three histograms mentioned above represent the geometrical gap between an entity (see: 2 Problem Statement) and its corresponding fit from the three points of view that are significant from the page-merging point of view: the translation over the x/y axis and rotation. In some cases, in order to increase the accuracy of results a filter triangle is applied over the histograms for a better parameter-detection (see Fig. 7).

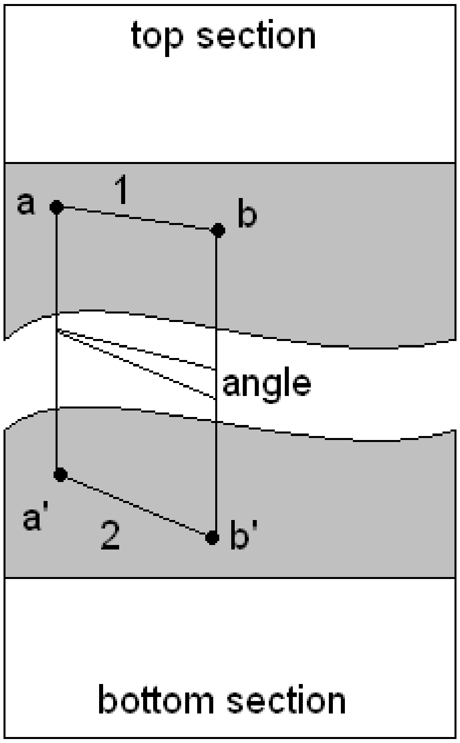


Fig.8 the angle between corresponding segments in different comparison bands for a top-bottom separation scenario

The final rotation-translation transformation is obtained by considering all plausible peaks from the previously computed histograms. These are used to blend in the two previously separated parts of a document which can be treated as one during the following stages of the digitization process.

B. *Entity matching*

The process of attempting to match entities in the two comparison bands can become time-consuming, especially when dealing with large input images. Therefore, a suitable matching approach must be used. When trying to match entities, we are actually comparing the weights and distances associated to each entity inside the comparison bands. A match is found between two entities when their weights and distances are about the same, within an adequate range.

There are two approaches available when matching these arrays of values. The first approach is the quadratic matching, with O(n^2) complexity, and involves an all-to-all comparison of values. This approach has the advantage of reliability, but may become time-consuming when the arrays are large.

As an alternative to the quadratic matching approach, the Linear Selective approach has a better complexity, O(n), but involves more algorithm steps and may become unreliable in some remote cases.

Given two arrays of weighted entities, $[w_{i1}, \dots, w_{iN}]$ and $[w_{j1}, \dots, w_{jM}]$, the first array corresponding to the first page and the second one containing weighted entities from the second page, the first step in the linear selective approach is sorting the two vectors, according to a $Match(W_i, W_j)$ function implementing a mathematical order relation between elements.

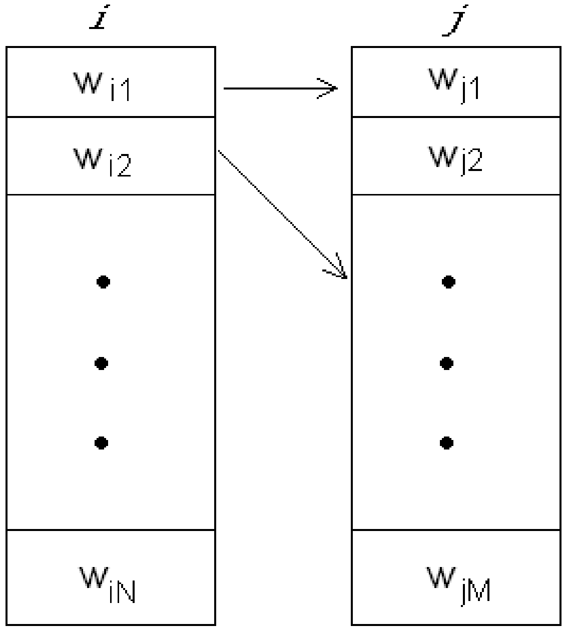


Fig.9 two vectors of weights, with the associated iteration variables i and j

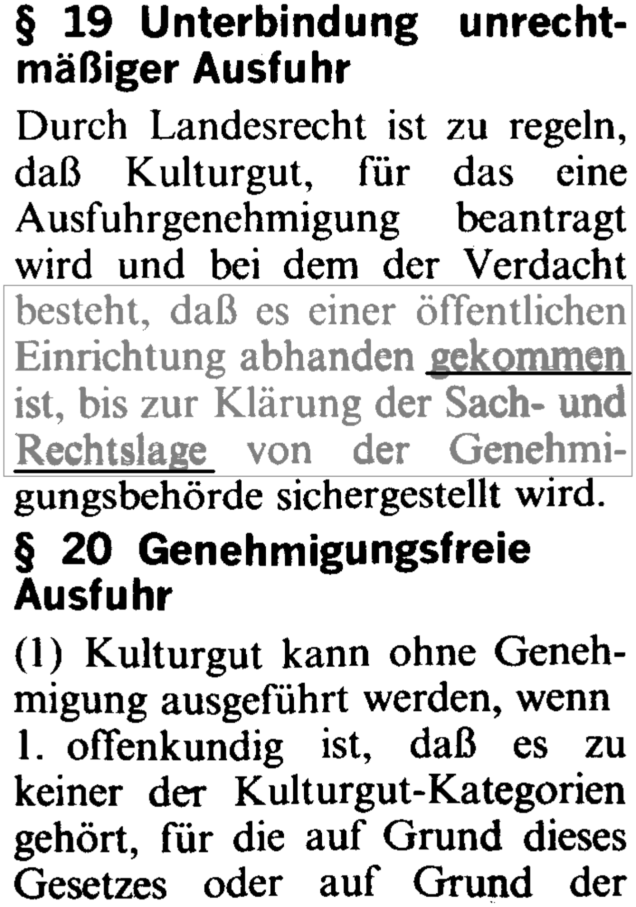
Afterwards, the $Match(W_i, W_j)$ function is applied between the elements of the two sorted vectors, in order to obtain the matching pairs of weighted entities. The function contains a range value, within which weights are considered to match so that the matching process should not become too restrictive. The matching steps are presented below, the processing resembling the navigation in ascending order of two sorted vectors:

```
1. While there are elements to process (i < N and j < M)
2. $If\ Match(W_i, W_j) == 0$
     • $i := i + 1$
     • $j := j + 1$
     • $Add\ (W_i, W_j)\ as\ matching\ pair$
     • Continue from Step 1
3. $If\ Match(W_i, W_j) < 0$
     • $i := i + 1$
     • Continue from Step 1
4. $If\ Match(W_i, W_j) > 0$
     • $j := j + 1$
     • Continue from Step 1
```

A zero return from the $Match(W_i, W_j)$ function means that the weighted entities in question are matching and as a result they can be paired.

C. Duplicate merge

After obtaining the rotation-translation function that blends the two overlapping images, the task of physically applying the function to the images (and verifying the results) remains.



§ 19 Unterbindung unrechtmäßiger Ausfuhr

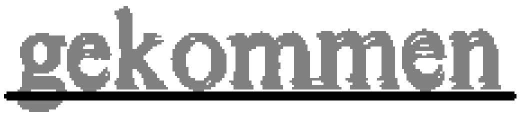
Durch Landesrecht ist zu regeln, daß Kulturgut, für das eine Ausfuhrgenehmigung beantragt wird und bei dem der Verdacht besteht, daß es einer öffentlichen Einrichtung abhanden gekommen ist, bis zur Klärung der Sach- und Rechtslage von der Genehmigungsbehörde sichergestellt wird.

§ 20 Genehmigungsfreie Ausfuhr

(1) Kulturgut kann ohne Genehmigung ausgeführt werden, wenn
1. offenkundig ist, daß es zu keiner der Kulturgut-Kategorien gehört, für die auf Grund dieses Gesetzes oder auf Grund der

Fig.10 the image result using the Sum approach

Upon closer inspection, some images that were merged using the presented algorithm exhibit a blurred, or “duplicate”, characteristic on some areas, as a result of a minor inaccuracy in the detection of the parameters or in the physical transformation and fitting of the images.



gekommen

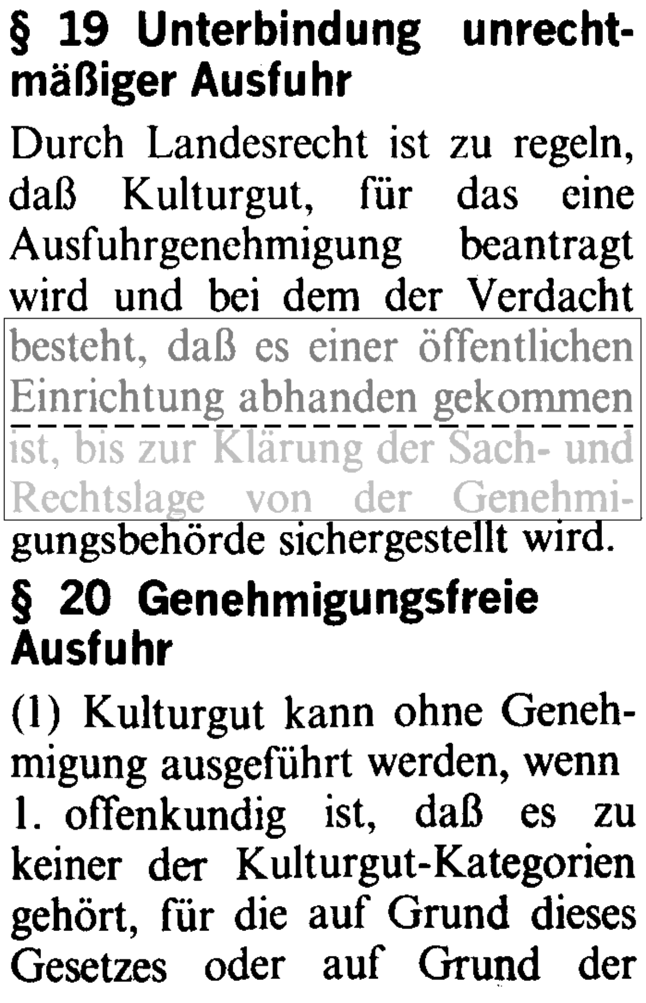
Fig.11 a word appears blurred when using the Sum approach

The presence of these artefacts is dependent on the approach applied when physically blending the images. There are two approaches that can be used in order to blend the images: the

Sum approach (an AND function), and the Choice approach (an OR function).

By using the Sum approach, some portions of the text may appear blurred, as it can be observed in Fig.10. This is the result of adding the two initial images. Although the translation and rotation are performed using the correct parameters, some entities will be distorted and not exactly matched as in the presented example. The underlined words in the figure appear "double-lined". For example:

This outcome can be avoided by using only one of the images' pixels when building the final image (the Choice approach). By comparing the output images from the two approaches, it can be observed that the problem of blurred components no longer appears when using this method, as can be seen in Fig.13. The choice of which individual part of the document to choose for a certain entity is performed by comparing the distance between the entity and the top bound of the comparison band in each part.



§ 19 Unterbindung unrechtmäßiger Ausfuhr

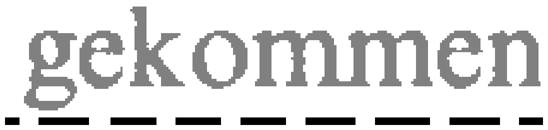
Durch Landesrecht ist zu regeln, daß Kulturgut, für das eine Ausfuhrgenehmigung beantragt wird und bei dem der Verdacht besteht, daß es einer öffentlichen Einrichtung abhanden gekommen ist, bis zur Klärung der Sach- und Rechtslage von der Genehmigungsbehörde sichergestellt wird.

§ 20 Genehmigungsfreie Ausfuhr

(1) Kulturgut kann ohne Genehmigung ausgeführt werden, wenn
1. offenkundig ist, daß es zu keiner der Kulturgut-Kategorien gehört, für die auf Grund dieses Gesetzes oder auf Grund der

Fig.12 the image result using the Choice approach

For example, in the case of a top/down overlapping case like in Figure 12, each entity will be outputted in the final result, either from the top of the bottom half as follows: the first two rows (in dark grey) are chosen from the upper part, while the bottom two rows (light grey) are chosen from the lower one.



gekommen

Fig.13 the same word no longer appears blurred

IV. Non-overlapping components

The algorithm aimed at recovering and merging non-intersecting partitions of a document is based on cost – computation of each text line using font properties. In order to achieve this, text-line detection and page de-skew must be performed before the actual process of calculating the characteristics of the font.

Binary image de-skewing can be obtained using either a simplification where shearing replaces an actual rotation, a true rotational of pixel coordinates or a sequence of shears to achieve a true rotation [11]. The key fact during this kind of pre-processing is the fact that jags placed into straight lines of the document are not removed, but rather additional compensating jags are added in the opposite direction, hence doubling their number. The most popular de-skewing algorithms used currently are the following:

- ✓ Bresenham-Style Shearing - these techniques typically rely on a variation of the Bresenham algorithm, which formulates a needed skew adjustment as a pair of pixel counts along the row and column axes of the document (similar to the way a builder sets the pitch of a roof using different measurements along the two legs of a builder's square).
- ✓ True Rotation - by using the sine and cosine of the skew angle, the ideal coordinate of the pixel in the input grid is found. This is the source of the current output pixel that is generated. The nearest neighbor method is then typically used, using the black or white state of the actual pixel nearest to that ideal coordinate to set the color of the output pixel.
- ✓ Rotation Via Multiple Shears - accomplishes a good approximation to a true rotation by means of an appropriate successive application of one dimensional shear.

Due to their importance to the layout of the document, and their use as iteration steps in the geometrical analysis, the text lines have to be detected as accurately as possible. In most real cases, the text line detection is often impaired by scan or printing flaws, resulting in noise or skewed images.

There is a large variety of methods available for the text line detection task, and they fall into a number of distinct classes, according to the input type. The most common class contains the documents with rectangular layout, according to [6].

The rectangular layout means that each page element can be enclosed into a rectangular shape. A number of specific text extraction methods which rely on this characteristic can be used in this case (e.g. projection and smearing). For the non rectangular printed documents, Kise et al. propose in [5] a Voronoi approach that can detect lines whatever their direction, based on area or distance estimations. Handwritten documents require a different approach, based generally on a bottom-up strategy. In [7] a method based on the Hough transform is used in order to detect lines in handwritten documents, while in [8] text lines are detected based on the extraction of aligned connected components.

An easier but still reliable method for line-detection aimed at font properties measurements, can be implemented using entities' properties. This hybrid approach makes use of a custom data structure and a routine that iterates through the input array of pixel clusters, building them into geometrically connected text lines. The line composition routine iterates through the input sets of connected pixels and either adds new components to an existing line or creates a new line with the current entity as reference. Based on the decision rule, clusters are on considered to be on the same text line if two conditions are fulfilled:

- ✓ both bounding rectangles are overlapping on the vertical axis
- ✓ the horizontal distance between them is smaller than a chosen threshold value. This threshold value is chosen such that vertical lines neighboring text columns will not be merged.

Because of the punctuation marks or noise in the page the detection of text lines sometimes fail, so a filtering function must be applied in the last stage, joining text lines that are either one completely inside another or horizontally adjacent.

The actual merging in the case of non-overlapping components is performed after the de-skewing step and uses the information obtained from the line – detection step. The bases for this algorithm are the font properties measurements performed for each individual line. Using this information the translation function is computed such that to best fit the lines in each individual component. As an observation, this approach is used for documents which are separated vertically (left/right parts), this being the most common case of non-overlapping document separation. The following font properties are considered during computations:

- ✓ boldness of characters
- ✓ italics property of characters
- ✓ font size

A. Font Boldness

Two different approaches can be used for the detection of the characters' degree of boldness.

The first one computes the ratio between the number of contour pixels, and the total number of pixels in an entity. This ratio will have a lower value in bold letters, as the contour pixels represent less of the total number of pixels from a bold entity.

The second approach, calculates the thickness of the entity in pixels. The routine iterates through the set of black entity pixels and searches for the largest vertical and horizontal black segments (from the entity) that the current pixel belongs to. The smaller of the two segments for each pixel is the segment of interest (called the dominant), being the actual thickness of the entity. By considering the segments for all the pixels, the most frequent value for the segment length is considered to be the input entity sets' thickness. This is done by plotting the dominant values of the input set onto a histogram and extracting the peak value.

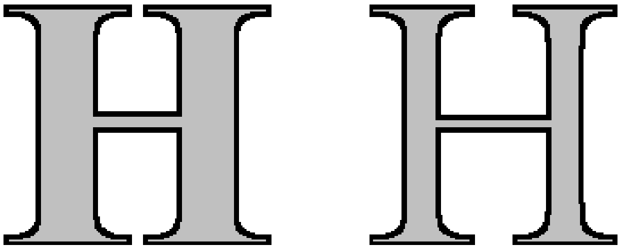


Fig. 14 the difference between a regular and a bold letter regarding the ratio between the number of outline pixels and the total number of black pixels

As it can be seen in Fig.15, the dominant can be either vertical or horizontal, depending on the pixel's position inside the entity. Theoretically, all segments passing through the current pixel should be investigated, at any angle (not just horizontal and vertical), but this approach proves to be too costly. In practical situations, the use of just vertical and horizontal segments coupled with the use of a histogram plot is enough to yield a reliable result.

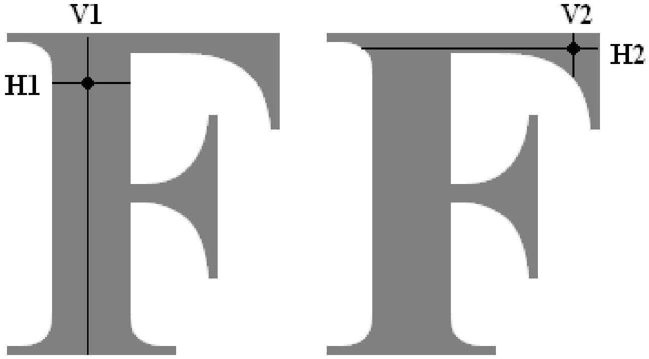


Fig. 15 for the pixel at the intersection of V1 and H1 the dominant will be vertical (V1), while for the pixel at the intersection of V2 and H2 the dominant will be horizontal (H2)

B. Font Italics

As in the case of the boldness measurement, for the italics measurements two approaches can been used, both based on the rotation of the entity by a reference value of 16 degrees.

Firstly, it can be easily observed that if an italic character is rotated counter-clockwise, the width of its' bounding rectangle decreases. In Fig.16, the regular letter clearly presents a narrower bounding rectangle compared to the italic character. Taking this into account, the algorithm rotates the current entity by -16 degrees and verifies this bounding rectangle's width against the initial one. If the width decreases after rotation, the letter is considered italic. Also, the initial letter is rotated by +16 degrees as to ensure the reliability of the result, as some letters have a "naturally italic" characteristic, such as the letter "a" in some fonts. A letter is considered "naturally italic" (and is skipped from the statistics) if it decreases its width when rotated both clockwise and counter-clockwise. A letter is considered italic only when its' bounding rectangle width decreases at -16 degrees and increases at +16 degrees.

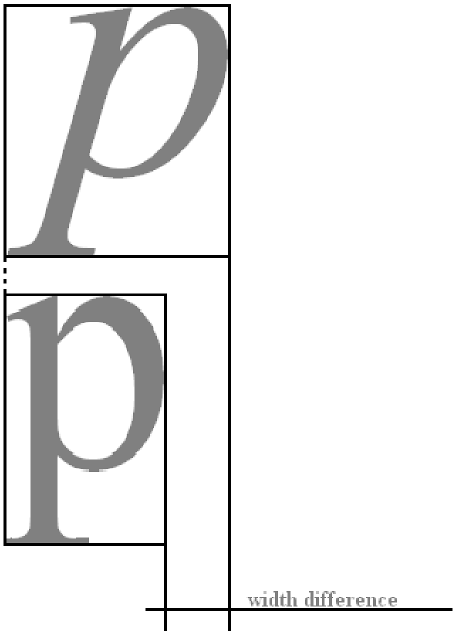


Fig. 16 the bounding rectangle of an italic letter (above) compared to the bounding rectangle of the same letter in regular form (below)

Secondly, it can be noticed that after the rotation, the italic characters' pixels tend to align vertically, forming longer vertical segments. By using these observations, it can be easily decided whether a character is italic or not.

This approach is theoretically correct, but in practical cases the vertical "chain" of black pixels may become interrupted, leading to inaccurate results, due to character particularities or the fragmentation of characters (due to poor quality of the input image or Black & White conversion faults).

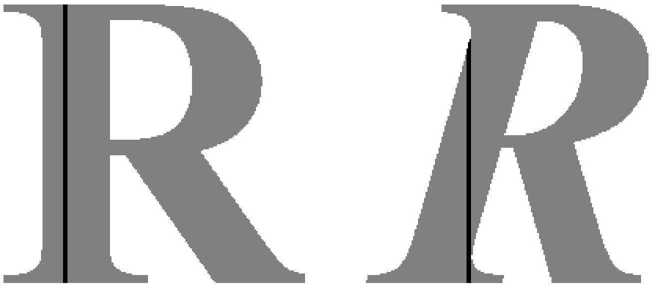


Fig. 17 the longest vertical line of black pixels in the case of a regular letter (left) compared to the longest line for an italic letter (right)

Using the entire above font properties assigned to each individual line, the to-be-merged components are compared line by line in the following way: a cost is computed for each line in one of the documents and then a corresponding match is found in the other document. When lines are matched from the characteristics point of view, the algorithm finds the vertical translation value necessary to match these lines geometrically.

As the non-overlapping merge scenario usually deals with vertical separation of images, the function that blends the images is significantly less complex than in the case of overlapping images. In this case, the function has only one parameter, a one-axis translation.

C. Font Size

The Font Size property of a line is computed by using the histogram representing the heights of the bounding rectangles of all entities (pixel clusters). The peaks of such a histogram represent the small and big caps sizes, and also punctuation marks. In regular texts, the small caps characters are predominant, and so the highest peak in the histogram represents them. Depending on the level of noise in the document, the noise peak varies in height, but can be identified as the leftmost peak on the histogram. Finally, the big caps peak is located in the rightmost region of the x-axis, but can be absent in some cases when the input text consists only of small caps characters. Before attempting to extract information, a triangle filter can be applied to the histogram in order to emphasizes the histogram peaks and obtain more accurate results.

V. EXPERIMENTAL RESULTS

In the following, we shall introduce some experimental results in order to underline the time complexity of the technique described in this paper. The below presented data are taken from newspapers found in the top-bottom separation case, this kind of separation being the most common in actual projects. A number of four experimental tests are detailed: the first three pairs of documents have basically the same resolution (~4960 x 3504 pixels), whereas the last one is from a pair of documents much larger (6614 x 4672 pixels).

Table 1. Experimental results

No. of Matches	**No. of Measurements**	**Time(s)**	**Resolution (Pixels)**
204	108	1.74	4959 x 3504
1780	154187	2.67	4960 x 3504
279	28952	1.66	4960 x 3507
502	53196	3.51	6614 x 4672

The number of matches in the above table represents the number of entities in one of the documents (the top part or the bottom part) which has a match in the other part. The number of measurements represents the total number of computations performed in order to determine the two parameters needed for the translation transformation and the angle for the rotation.

As shown in the figure below, the trend line obtained from the graph representing the total number of matches versus the total number of measurements is a polynomial of degree 2.

From the time complexity point of view, it can be easily noticed that the algorithm's performance is proportional to the size of the document. The least time is obtained for the last document, which is also the largest. Even though, this is not a very important fact, as in most of the projects, the resolution of all documents is the same depending on the scanning procedure and the preprocessing stage in the content conversion process.

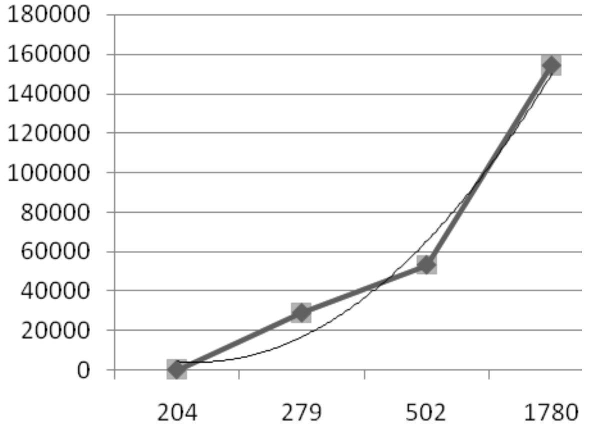


Fig. 18 number of matches vs. number of measurements

VI. Conclusions

The approach presented in this paper has a geometric basis and addresses the two most common scenarios for the images that have to be merged. If the sections are overlapping, comparison bands are chosen on the pair of images, and the algorithm attempts to obtain a function that matches corresponding entities in the two bands. When the input images are non-overlapping, higher-order geometric computations are performed, obtaining the characteristics of the text and finding a line-to-line match. A positive match in text characteristics results in the linear translation coefficient necessary for the page sections' alignment.

Regardless of the input type, a single output image is obtained, serving as a better starting point for any further processing task, such as logical structuring or content extraction, and reducing or eliminating the need for manual corrections.

VII. Future Work

The current paper described two preprocessing methods which can be used in large images that must be scanned in multiple steps because they are too large to be scanned in a single step. However, the described method used a content matching technique that is usable only after the recovery of the connected component in the image. For that, a binary conversion preprocessing phase must be performed and that may be a limitation for documents that contain not enough connected components in binary color space. As a result the merging techniques may be adapted in a future project to operate directly into continuous color spaces like grayscale or true-color, to extract the connected components directly into these spaces, thus enabling them to be less sensitive to the errors of image binarization.

Acknowledgment

We want to thank our colleagues from the University "Politehnica" of Bucharest, to the staff of Computer Science and Engineering department, for their ideas, research support and useful advices.

References

[1] C.A. Boiangiu, A. C. Spataru, A. I. Dvornic and I. Bucur, "Merge techniques for large multiple-pass scanned images", Proceedings of the 1st WSEAS Int. Conf. on VISUALIZATION, IMAGING and SIMULATION (VIS '08), WSEAS Press, Bucharest, Romania, November 7-9, 2008, pp. 72 – 76.

[2] C. A. Boiangiu, D. C. Cananau and A. C. Spataru, "Modern approaches in detection of page separators for image clustering", WSEAS Transactions on Computers, Vol. 5, Issue 7, July 2008, pp. 1071-1080.

[3] G. Leedham, S. Varma, A. Patankar and V. Goviandaraju, "Separating text and background in degraded document images" , Proceedings Eighth International Workshop on Frontiers of Handwriting Recognition, Ontario, Canada, September 2002, pp. 244-249.

[4] J. Bernsen, "Dynamic thresholding of grey-level images", International Conference on Pattern Recognition (ICPR86), Paris, France, October 1986, pp. 1251-1255.

[5] K. Kise, M. Iwata, A. Dengel and K. Matsumoto, "Text-line extraction as selection of paths in the neighbor graph", Document Analysis Systems, 1998, pp. 519-523.

[6] A. Lemaitre, B. Couasnon and I. Leplumey, "Using a neighborhood graph based on Voronoi tessellation with DMOS, a generic method for structured document recognition", Proceedings of GREC, Sixth IAPR International Workshop on Graphics Recognition, Hong-Kong, China, August 2005, pp. 260-271.

[7] L. Likforman-Sulem, A. Hanimyan and C. Faure, "A Hough based algorithm for extracting text lines in handwritten documents", 9th International Workshop on Frontiers in Handwriting Recognition (IWFHR), 2004, pp. 245-250.

[8] L. Likforman-Sulem and C. Faure, C., "Extracting text lines in handwritten documents by perceptual grouping", Advances in handwriting and drawing: a multidisciplinary approach, Paris, 1994.

[9] P. Bojarczak, S. Osowski, "Denoising of Images – a Comparison of different filtering approaches", WSEAS Transactions on Computers, Vol. 3, Issue 3, July 2004.

[10] Prateek Sarkar, Henry S. Baird, Xiaohu Zhang, "Training on Severely Degraded Text-Line Images", ICDAR, Vol. 1, 2003.

[11] B. Couasnon, L. Camillerapp and I. Leplumey, "Making handwritten archives documents accessible to public with a generic system of document image analysis", International Workshop on Document Image Analysis for Libraries, January 2004, pp. 270-274.

[12] F. Aurenhammer, " Voronoi diagrams - A survey of a fundamental geometric data structure", ACM Computing Survey, Volume 23, Issue 3, September 1991, pp. 345-405.

[13] B. Chen, and L. He, "Fuzzy template matching for printing character inspection", WSEAS Transactions on Circuits and Systems, Issue 3, Vol. 3, 2004, pp. 575 - 580.

[14] L. M. Sheikh, I. Hassan, N. Z. Sheikh, R. A. Bashir, S. A. Khan, and S. S. Khan, "An adaptive multi-thresholding technique for binarization of color images", WSEAS Transactions on Information Science and Applications, Issue 8, Vol. 2, 2005, pp. 1202 - 1207.

[15] C. A. Boiangiu, D. C. Cananau and A. C. Spataru, "Normalized text font resemblance method aimed at document image page clustering", WSEAS Transactions on Computers, Vol. 7, Issue 7, July 2008, pp. 1091-1100.

[16] F. Aurenhammer, " Voronoi diagrams - A survey of a fundamental geometric data structure", ACM Computing Survey, Volume 23, Issue 3, September 1991, pp. 345-405.

[17] Yuan, B.; Tan, C.L. (2003). "Skewscope: The Textual Document Skew Detector", Proceedings of the Seventh International Conference on Document Analysis and Recognition, Vol. 1, pp. 49-53, ISBN 0-7695-1960-1, Scotland, August 2003, Edinburgh.

[18] Y. Zheng, H. Li and D. Doermann, "A model-based line Detection algorithm in documents", Proceedings of the Seventh International Conference on Document Analysis and Recognition, Vol. 1, Edinburgh, Scotland, August 2003, pp. 44-48.