Use of Connected Components for automatic digitalized documents restoration.

Marisa R. De Giusti[1]; María Marta Vila; Villarreal Gonzalo Luján

Comisión de Investigaciones Científicas de la Provincia de Buenos Aires (CIC)
Proyecto de Enlace de Bibliotecas (PrEBi)
Servicio de Difusión de la Creación Intelectual (SeDiCI)
Universidad Nacional de La Plata
La Plata, Argentina

marisa.degiusti@sedici.unlp.edu.ar, {vilamm, gonetil}@sedici.unlp.edu.ar

1. Introduction

The process of digitalizing documents has been simplified through the advance of technology and the development of devices for images capture (scanners, digital cameras), which has provoke the constant digitalization and on line movement of huge amounts for documents. This has generated great advantages in many environments, thanks to the ability of preserve very large document databases and transfer them almost immediately through digital mediums, allowing to a huge number of people the access to information that, in other way, would have been impossible. A clear example of this statement is given inside the Digital Library frame, where PrEBi and SeDiCI projects are strongly involved generating services, developing software tools and carring out many research lines related to common subjects.

Despite all mentioned advances, the digitalization of documents keeps generating some problems originated from human or technical mistakes, and that result in images with redundant information, hard-to-read documents, extremely big files, etc. Among these problems we can stand out:

- Documents with black or diffuse borders: It usually happens when the original documents are in poor conditions, or when the scanner has not been well closed. This implies bigger documents, heavier digital files, more ink consumption and some kind of difficulty for reading them.
- Random dots along the document: It might happen because of original documents poor quality as well as problems with the scanner; this generates from dirty or untidy documents to hard-to-read texts due to the points among the text (sandy effect)
- Images slant: due to distraction of the operator, automatic scanners malfunction or lack of space when digitalizing big documents (such as large books), many documents present some kind of slop compared to the base line. This slop might affect the reader, specially if it is grater than 5 degrees and under 85 degrees.

In this article, we will present a optimized and automatized mechanism for improve digitalized documents that present some or even all problems described below. First we introduce briefly the concept of Connected Components, which are essential for the whole process. Then, we carry on with a technique for slant detection of document as well as a basic algorithm for image rotation. Next, we explain an easy method for detection and elimination of black borders and random points. Last, we propose an automated mechanism that integrates all these tasks in a big and organized restoration process.

2. Connected Components.

2.1 Description.

The use of connected components for digital image processing is not new, but given all the information that these components may retrieve from a determined image, new applications are constantly found for image processing,, restoration, patter recognition, and many other fields.

Basically, connected components allow to associate elements or parts (zones) of images with each other and then perform operations based on the composition of the images in different parts. A connected component is nothing but a set of image points or pixels grouped by some characteristic that identify them. This

[1] Investigador Comisión de Investigaciones Científicas de la Provincia de Buenos Aires – CIC.

grouping permits, for example, to discriminate the different parts of an image, to find relationships among the elements from a given component or even from many components, to apply certain operations and filters to segments of images, etc.

As has been explained in [1], a set of connected components (CC) can be seen as a set of points grouped in Equivalence Classes, where a determined point p belongs to exactly one equivalence class, and where there not exist empty equivalence classes. Also in [1] it has been developed a method for CC computting and in [2] it has been explain some useful applications for it. In this document, some modifications to the algorithm for CC calculation will be introduced, adding new data structures that improve the response time and optimize the obtantion of certain parmeters.

2.2 Optimizing the technique.

The first change introduced is the use of an extra data structure - a bi-dimenisional array- with the same size of the original image, with integer values and initialized in zero. Even though this increase memory usage, specially when a large amount of good-quality-images are being processed in parallel, it permits to delay the actual application of some operations by putting *marks* in the points to change and doing all changes at the end of the process.

Let **I** be the original image, sized MxN, and let CC[M,N] be the matrix that contains all connected components, sized MxN as well. We assume that **I** has at least one black point (otherwise it would be a useless all-white document image).

The process of CC calculations are just as explained in [1], but now the CC matrix will have information about all CC, as follows:

Let $p_{i,j}$ be a point of the image I, where $0 <= i < M$; $0 <= j < N$

Let CCs be the set of **all** CC if the image; Ccs not empty (as assumed above). Let Ck ∈ CCs, Ck $\neq \phi$ is a connected component and k > 0 its label. Then, the matrix CC is composed as:

- CC[i,j] = 0 => $p_{i,j}$ does not belong to any CC
- CC[i,j] = k => $p_{i,j}$ belongs to CC Ck ∈ CCs

The main advantage of this structure is that it permits to know immediately to which CC belongs any given point p ∈ I. As a disadvantage, it requires to allocate an extra amount of memory to store a matrix as big as the image in order to register all CC labels.

Many operations require to know the size the CC to which a given point p belongs. Let then size(Ck) be a function that returns the amount of points that form the CC Ck. Clearly, size(Ck) > 0 given that Ck $\neq \phi$ always. The required time to compute this function is a very important factor when processing components, since many operations require repeatedly this datum and therefore it might provoke an important degradation of the whole process.

In order to speed up this operation and take it to a minimum execution order, we have added another data structure that holds the size of each CC. Let S be an array with length equal than the amount of elements in CCs. Then, if S[k] = m it means that the component Ck ∈ CCs has m elements. This way, the execution order of function size() remains constant. Again, as a disadvantage, an extra amount of memory is required for this structure, besides the fact that the calculation of CC takes a bit longer since it now needs to fill this array of sizes. Anyway, we have found that the improvement this extra structure generates is so clear that the "disadvantages" are minimized.

3. Operations for image improvement.

3.1 Slant Correction

As mentioned previously, the slant problem is quite common when digitalizing documents and it is evident that a slanted document is a bit harder to read.

The aim here is to automatically correct the slant in the pages of the document, avoiding human intervention to modify each page by hand. To achieve this objective, we have to main problems to solve:

- to detect the current slant of each page (slant detection)
- to straighten the pages (slant correction)

3.1.1 Slant Detection in Image Documents

Automatic detection of slant angle in a document is a complex process, but it does not always ensure the achievement of expected results. In the current literature there are many methods developed for slant correction, differentiated by their robustness and complexity of the underlying method, exactitude of the results and the efficiency of the algorithms, in execution time as well as system resources.

In [3] there is described a robust method based on connected components which consist of lowing down the quality of the image in order to obtain big squares that represent words and then detect the slope of these words. In [4] and [5] it has been described two methods in which random parts of the documents are taken and it is used cross correlation (horizontal and vertical), which allows to detect the slant of document with Chinese or Japanese characters too. These methods, although half or very complex, can achieve very good results very efficiently.

There exists besides simpler methods, as described in [7] and [8], in which the aim is to maximize the difference between peaks and valleys of the horizontal histogram of the image. To do this, many rotations are made calculating horizontal projections and analyzing results from all histograms and adjusting the rotation angle based on a dichotomic search. Even though the results are not as precise as other methods (this method tends to be strongly noise-sensitive) and execution time a bit higher, the methods are extremely easy to carry out allowing a quick software development, appropiate in some environments.

Among all analyzed methods, the one introduced in [6] has a good balance between complexity and efficiency, achieving high quality results. We have implemented the algorithms but with some modifications which pretend to solve some problems described in the original article.

The whole algorithm consist of 7 serial steps, with probably many cycles or iterations until the desired angle is reached. These steps are:

1. Detect the set of all CC in the image;
2. For each CC, calculate its centroid (Cx,Cy), its height (h), its width (w) and its bounding box (left higher and right lower poitns).
3. For each CC, calculate the NNC (*nearest-neighbor chain*) :

a. Find the nearest CC.
Let C_1 and C_2 CC of I. C_2 is the nearest neighbor (NN) of C_1 if $dc(C_1, C_2) = m$; where $dc(C_a,C_b) = \Delta x^2 + \Delta y^2$ is the distance function between CC and $\Delta x = |x_{c1} - x_{c2}|$

b. If a near component has been found (C_2), it must verify

- $hC_1 \simeq hC_2$ for $\Delta x > \Delta y$ (with hC_i height of CC i) ó $wC_1 \simeq wC_2$ for $\Delta y > \Delta x$ (with wC_i width of CC i)
- $Cx_2 > Cx_1$ for $\Delta x > \Delta y$, ó
$Cy_2 > Cy_1$ for $\Delta y > \Delta x$
- $dg(C_1 , C_2) < \beta .$ max (hC_1 , hC_2)

$\beta = 1.2$ (constant).

c. If C2 accomplishes with all conditions, then it is registered as NN of C1 and it is also registered that C2 can not be the root of the chain.

For example:

# component	NNC	Is chain root?
1	16	SI
8	22	SI
16	32	NO
24	30	SI
30	40	NO

4. Identify all K-NNC (NNC with length K)
For each root:

a. Calculate the length of chain L.
b. Increment the amount of L-length chains.

With these numbers, a new one dimensional array is generated in which each position i will have the amount of NNC with size i and the roots of these NNCs.

For example, from point 4 we obtain this result:

k	Number of chains	roots
3	2	1, 24
2	1	8

As long as each chain is processed, it is stored the k with highest amount of chains.

5. Look for the biggest k: K is initialized as the biggest number of CC from all NNC, and the amount of K-NNCs is calculated. If the value is lower than 3, the procedure is repeated with K = K - 1. At this point, Lu et al. in [6] suggest the use of a minimum threshold of 3. Sometimes it is required to low down this value to 2, specially when processing documents with few text and many images inside.

6. Process each K-NNC from step 5 to obtain the slant angle of each one.

7. Calculate the slant angle of the document (S_D) as the medium of all angles from 6):

$$S_D = \arctan (S_D) \times 180 / \pi .$$

3.1.2 Document Rotation.

Once the angle of the document has been found, the document must be rotated at an inverse angle in order to correct the problem[2]. The rotation angle is very simple, and the only restriction it has is that it must keep the center of the image I. The transformation to apply follows as any other affine transformation:

$$R(x,y) = A(f1(x,y),f2(x,y));$$
$$f1,f2 : (N\square,\square N) \rightarrow \square R$$

This affine matrix is known as Rotation Affine Matrix or simply Rotation Angle with is calculated as follows:

$$M = \begin{bmatrix} a & \beta & \mid (1-a) \cdot x_p - \beta \cdot y_p \\ -\beta & a & \mid \beta \cdot x_p + (1-a) \cdot y_p \end{bmatrix}$$

where $a = scale * \cos(\alpha)$, $\beta = scale * \sin(\alpha)$ and *scale* represents the scale of the rotated image (ideally, the original size must be kept). We can then define a based coordinate for each point in the space, through tangent vectors to coordinates lines. This new base line can be related with the fundamental base of cartesian coordinates through some relationships, and then we can apply the transformation by specifying how each point of the origin system is mapped to the destiny system.

$$w' = h . |sen(\alpha)| + w . |cos(\alpha)|$$
$$h' = w . |sen(\alpha)| + h . |cos(\alpha)|$$

where w' y h' represent the required size of the rotated image. This allows us to calculate the right scale, as follows:

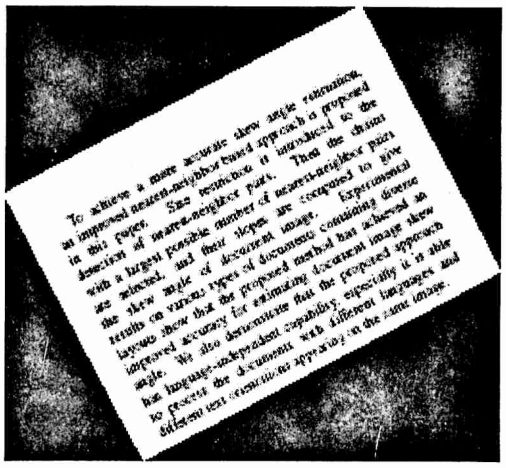
$sw = w_d / w'$; width scale, w_d width of destination image
$sh = h_d / h'$; height scale, h_d height of destination image
$scale = \min(sw,sh)$; the real scale must be the one that adjust the most to the destination image

Finally, the rotation angle is adjusted to the center of the destination image.

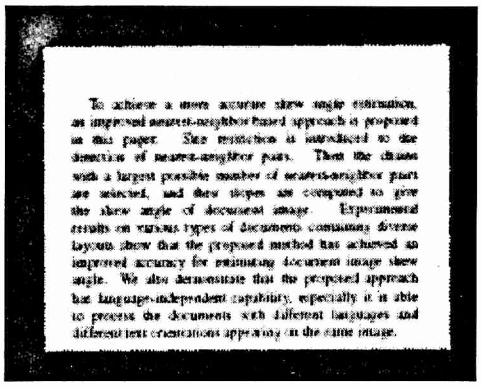
[2] The rotation angle is represented in radians. If not, it must be translated: $\alpha = (degree . \pi)/180$

Once matrix M is calculated, we can apply the rotation transformation R.

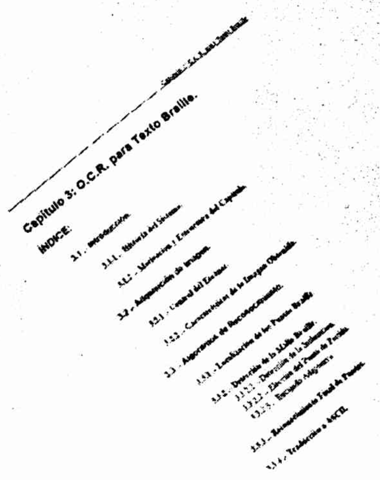
In the following pictures we can observe the results of applying this method. In each, we have the original rotated image to which we calculate the slant angle and then the result of the rotation in that angle.

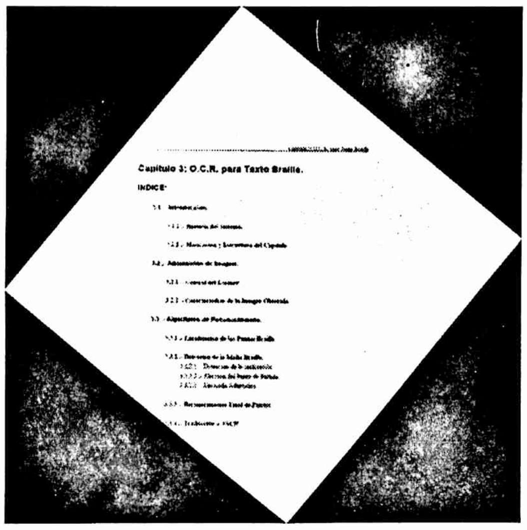

To achieve a more accurate skew angle estimation, an improved nearest-neighbor based approach is proposed in this paper. Size restriction is introduced to the detection of nearest-neighbor pairs. Then the chains with a largest possible number of nearest-neighbor pairs are selected, and their slopes are computed to give the skew angle of document image. Experimental results on various types of documents containing diverse layouts show that the proposed method has achieved an improved accuracy for estimating document image skew angle. We also demonstrate that the proposed approach has language-independent capability, especially it is able to process the documents with different languages and different text orientations appearing in the same image.

Original image


To achieve a more accurate skew angle estimation, an improved nearest-neighbor based approach is proposed in this paper. Size restriction is introduced to the detection of nearest-neighbor pairs. Then the chains with a largest possible number of nearest-neighbor pairs are selected, and their slopes are computed to give the skew angle of document image. Experimental results on various types of documents containing diverse layouts show that the proposed method has achieved an improved accuracy for estimating document image skew angle. We also demonstrate that the proposed approach has language-independent capability, especially it is able to process the documents with different languages and different text orientations appearing in the same image.

Rotated image



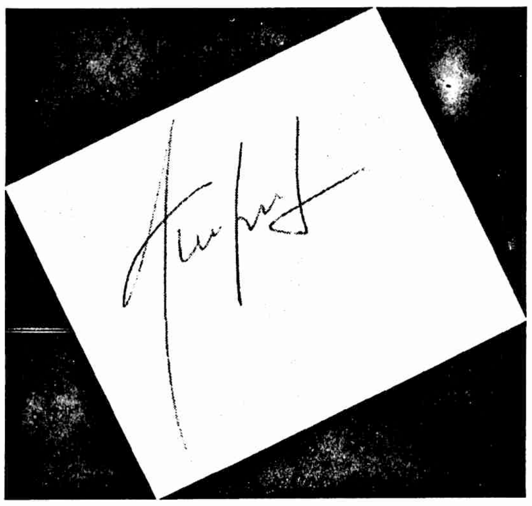


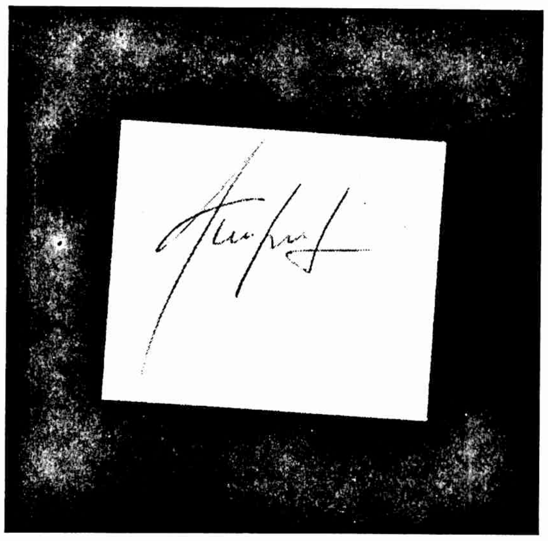
Capitulo 3: O.C.R. para Texto Braille.

INDICE:

[illegible]

Above original image. Below rotate image.





Above original image. Below rotated image.

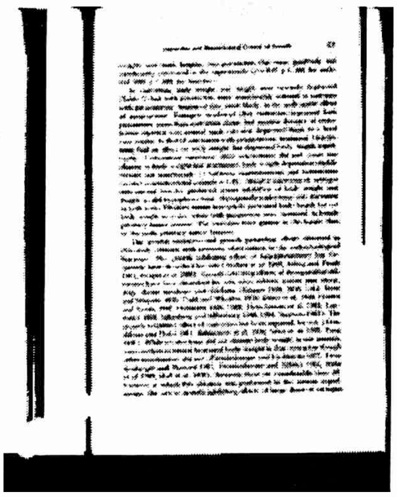
3.2 Black borders cleaning

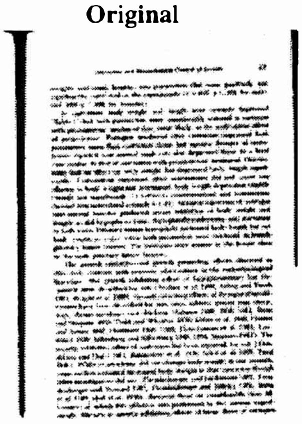
Black borders in the images may arise as a consequence of an incorrect use of the scanner (not properly closed) maybe because of negligence of the operator or probably the need of digitalizing big-volume books. As a result, this generates a black border around the image which results in bigger files as well as waste of ink or toner when printing the document.

The use of connected components can be very useful when looking for an automated solution for this sort of problems. A black border is, or at least a black bar, will form one single connected component which indeed will be considerably big, specially if it is compared to the average size of the rest of the components.

Even though pictures or tables inside the image will also be part of very large CC, they do not belong to the periphery of the image so they are not considered as "erasable zones".

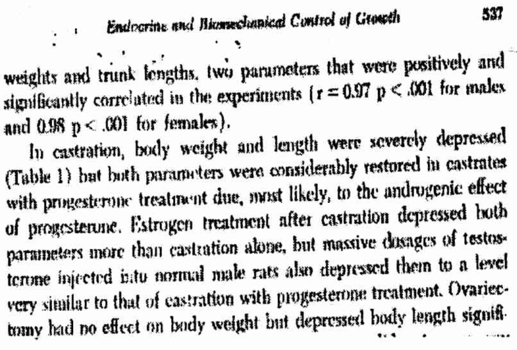
Another common problem arises when black borders do not affect the whole periphery, or there exist more than one black zone surrounding the image. As shown in the following picture, the elimination of these peripheral components will not ensure the total cleaning of black borders. To solve this problem, a step-by-step cleaning procedure can be done, starting from the most outer components and moving forward the center of the image, until no huge component is found or the text of the document is reached.


Original


First cleaning

3.3 Random points elimination (despeckle)

A very common problem that frequently comes up after documents digitalization is the generation of random black points along the image, as shown in the picture below. These may happen because of bad conservation of the original document, low quality image files and dust in the scanner bed or in the paper.


Endocrine and Biomechanical Control of Growth 537

weights and trunk lengths, two parameters that were positively and significantly correlated in the experiments ($r = 0.97$ $p < .001$ for males and 0.98 $p < .001$ for females).

In castration, body weight and length were severely depressed (Table 1) but both parameters were considerably restored in castrates with progesterone treatment due, most likely, to the androgenic effect of progesterone. Estrogen treatment after castration depressed both parameters more than castration alone, but massive dosages of testosterone injected into normal male rats also depressed them to a level very similar to that of castration with progesterone treatment. Ovariectomy had no effect on body weight but depressed body length signifi-

Text with random points

Connected component let us find an easy and fast solution to this problem, since this points will belong to very small components, probably no bigger than 2 or 3 points. which means we can solve our problem by searching this tiny CC. Given that we have already store the size of each CC in a special data structure, we can get a really efficient solution quite fast.

Let p be a random point of image I, and let μ be threshold that indicates the minimum allowed size of a CC. Let cc = CCs[p_x,p_y]. Then, if size(cc) < μ => $I_{x,y} = b$ (b is a clean - white point)

4. Automatized process.

Cleaning operations permit to obtain very good results, but they require many passings through (at least one for each operation). We propose here a automatized process that minimizes the amount of passings and that focuses on the relevant zones of the images.

The first operation will be, necessarily, the slant detection and rotation; it must be done before any other operation because it will provoke changes in the CC array, which will need to be recalculated once this operation has ended. Then, we can proceed with the noise cleaning. We can speed up the process by minimizing the area over we will work and minimizing the amount of passings through the image.

The minimize the passings, we can take advantage of the fact that CC are fully eliminated; instead of eliminating them every time we can, we just mark them as *erasable* in the different steps and then move along the whole image only once eliminating all marked CC. If we use again an extra array of marks, we can perform the operation of searching erasable components and deleting them in a constant time.

We can improve the algorithm a bit more. As mentioned before, black borders are eliminated in sequence going to the center of the image. But, once we have detected the most inner border to eliminate, we could consider it as the minimum bounding box of the image. Then we can trim the image, ignoring both black borders and random points in zones outside this box. Besides, we optimize the size of the image, resulting in smaller files and with only relevant information inside. This is very useful if we are going on working with pattern recognition or post-processing over these images.

4.1 Algorithm execution orders.

If we know how efficient our system is, we must measure it. This is a good way to improve it in speed and

accuracy.
Se have calculated the maximum of all execution orders O(n) in each step of the algorithm. First, we make some considerations:

1. Images are all square, say NxN. This is the same as considering the worst case, in a MxN image when N > M.
2. Because of 1), each passing through the image will have a minimum quadratic O(n), say $O(n^2)$, since n^2 points are analyzed.
3. We always consider the worst case, but this is not 100& real. For example, an image may have up to $n^2/2$ CC (with 4 vicinity), but this would not have any sense because it would be an image with only black and white point in a mesh.

Rotation technique consist of a sequence of well defined steps:

- Connected Components calculation
- Slant angle detection
- Image rotation

1. The execution order for CC calculation is $O(n^2)$.
2. If the the worst case there would be $n^2/2$ CC, then the slant detection algorithm would result in $O(n^4)$, since all CC are processed, NNC calculated, and then K-NNC are searched, calculated and used to get the angle. In each step we have this executions orders:
   - Step 1: $O(n^2/2)$
   - Step 2: $O(n^4)$
   - Step 3: $O(n^4)$
   - Steps 5 y 6: O(1)
   - Step 7: $O(n^2/2)$
   - Step 8: O(c), with c constant
3. Finally, execution order of image rotation is $O(n^2)$, since we only need to obtain the rotation matrix - $O(n^2)$ - and apply it to the image – $O(n^2)$ -.

From 1, 2 y 3 we can easily conclude that the point 2 determines the final execution order, say $O(n^4)$.

5. Conclusions.

In the digital image processing field there exist many algorithms and techniques to make the most different tasks; in particular, techniques related with the area of digitalized document improvement have considerably improved and very good results have been obtained. Anyway, a lot of techniques can be optimized yet, grouped together with other techniques that make similar or related tasks, or adjusted/adapted for new applications.

In this document, the technique introduced in [6] has been adapted for, on one side optimize its efficiency thinking specially in processing many documents in a sequence, an on the other side to document with images on it, or even manuscripts.

We have use this technique to decompose PDF formatted documents in a set of images (one by page), to which these algorithms have been applied to finally rebuild the original PDF file. Results have permit to generate an automatized sequential process, and we have seat the basis to continue developing focused on printed and manuscript document improving, and in parallel we continue improving all developed algorithms, thinking in resource optimization and concurrent image processing, using multi-processor architectures or processing in cluster.

6. References.

[1] Marisa R. De Giusti, Maria Marta Vila, Gonzalo Luján Villarreal. "Manuscript Document Digitalization and recognition: a first approach". *Journal of Computer Science and Technology.* Vol 5. No. 3. 158-163,2005.
[2] Marisa R. De Giusti, Maria Marta Vila, Gonzalo Luján Villarreal. "Manuscript Character Recognition Overview of features for the Feature Vector". *Journal of Computer Science and Technology.* Vol 6. No. 2. 92-98,2006.
[3] Oleg Okun, Matti Pietikäinen and Jaakko Sauvola, "Robust Document Skew Detection Based on Line Extraction". 'Machine Vision and Media Processing Group, Infotech Oulu and Dept. of EE, University of Oulu
[4] Avanindra and S. Chaudhuri, Robust Detection of Skew in Document Images, *IEEE Trans. on Image Processing,* Vol. 6, No. 2, 1997, pp. 344-349.[5] Ming Chen, Xiaoqing Ding "A Robust Skew Detection algorithm for Grayscale Document Image". Fifth International Conference on Document Analysis and Recognition (ICDAR'99). p. 617
[6] Yue Lu, Chew Lim Tan. "A nearest-neighbor chain based approach to skew estimation in document image". Pattern Recognition Letters 24 (2003) 2315–2323
[7] Pastor Gaeda, Moises, Aportaciones al reconocimiento automático de texto manuscrito, Dpto. de Sistemas Informáticos y Computación, Universidad de Valencia.
[8] H.S. Baird. The skew angle of printed documents. In *Proc. of the Conf.Society of Photographic Scientists and Engineers*, pages 14–21, 1987.