A Recognition based Approach for segmenting Touching Components in Arabic Manuscripts

Nabil Aouadi
University of Tunis
Tunis, Tunisia
Email: Nabil.aouadi@utic.rnu.tn

Afef Kacem Echi
University of Tunis
Tunis, Tunisia
Email: afef.Kacem@esstt.rnu.tn

Abdel Belaïd
University of Lorraine
Nancy, France
Email: abdel.belaid@loria.fr

***Abstract*—This work aims to segment touching components (TCs) which may occur between word letters of consecutive text-lines or those of words of the same line in Arabic manuscripts. The proposed approach is mainly based on two steps: 1) finding for a localized touching component its most similar model, stored in a dictionary with its correct segmentation, based on shape context descriptor, 2) segmenting the touching component based on central point of the found most similar model's parts. Tests are performed using a database of connection zones (1300 samples) and three metrics: Manhattan, Euclidean and Canberra distances. Experimental results have shown the effectiveness of the proposed touching component segmentation method in comparison to some related works. Our best achieved TC segmentation rate is of 94%.**

***Keywords*—Touching component; Segmentation; Shape context descriptor; Central point.**

I. Introduction

Depending on the writing style, characters and words may have unusual and varying spaces in manuscripts. These latter may also include narrow text-lines with overlapping and touching components (TCs). Note that these TCs may occur not only between consecutive text-lines, which are the most predominated case, but also between words of the same text-line or even between letters of the same word. This problem is common in unconstrained Arabic manuscripts since most of Arabic letters (21 from 28) are ascendant or descendant and contain special marks and dots. In addition, in ancient Arabic calligraphy, almost all letters contain big jambs from left to right or inversely. Segmenting Arabic manuscripts would ideally process without TCs. But, in presence of these TCs, a finer analysis process should be performed. Notice that a TC belongs to one of these three categories: 1) simple touching (see Fig. 1(a)), 2) multi-touching where the connection is between more than two letters (see Fig. 1(b)) and 3) overlapping in case of extension of the connection beyond letters (see Fig. 1(c) for inter-lines overlapping and see Fig. 1(d) and (e) for inter-words overlapping).

Several text-line segmentation methods have been reported in the literature by Likforman-Sulem and al. in [3]. In the method of Farnandez-Mota [30], a TC is localized when discontinuity in background skeleton image is observed. The segmentation of a TC follows a virtual edge between two endpoints. Other methods operate by making a horizontal cut at particular position [4], [5], [6]. In sum, the TC segmentation approaches can be basically classified into two categories namely, Recognition-free and Recognition-based.

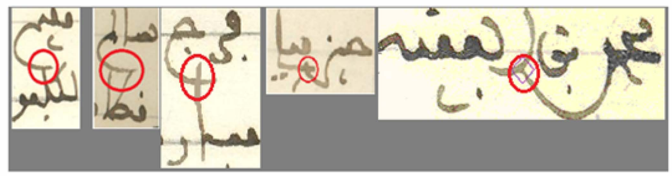

Fig. 1. Samples of TCs in Arabic and Latin Manuscripts: (a) simple touching, (b) multi-touching, (c) overlapping, (d) Example of inter-word connection,(e) non consecutive inter-word connection.

Recognition-free approaches are based on analyzing the component features and some structural information [7], [8], [9], [10], [11], [12], [13], [14], [15], [17], [18], [19] and [21]. Methods under this approach are generally performed in the TC's thinnest density region or based on the TC's baseline, TC's projection profile, contour or skeleton analysis.

Recognition-based methods search segmentation among generated ones using recognition step [16] and [20]. In Piquin and al. [16] all representative models are stored in a dictionary as a decision tree including geometrical and topological parameters. As there are many studies which are able to localize TCs in the manuscripts [21] and [22], we assumed that TCs are already extracted, as done by Kang [20], and we only focused on their segmentation. Our contribution is the recognition of the most similar model to a processed TC and its segmentation. This is done according to pixel's distribution and central points of the similar model's parts.

The paper is organized as follows. Section II describes our TC proposed segmentation method with a focus on the most similar model recognition step and the properly so called TC segmentation step. In Section III, experimental results are discussed. Finally conclusion and prospects are given in Section IV.

II. Proposed TC Segmentation Method

As mentioned before, two main steps are needed to TC segmentation. The first step looks for the most similar model using a matching with Shape Context (SC) descriptor. The second step uses an a priori knowledge of the correct segmentation of the found model. It exploits the central points of these model's parts to segment the TC. Models are stored in a TC database. We used the dataset of Kang and Doermann [20] for the availability of its ground truth. Then, we enriched it by working in two ways: 1) adding real connections extracted from ancient

Arabic manuscripts and 2) applying two algorithms: the first one switches randomly parts A with different others parts B and produces new TCs (A and B are respectively the first and the second model's parts). The second algorithm performs rotations of parts B and it associates them later with correspondent part A and vice versa. Notice also that our system is trained with different types of TCs that we called models. Models are organized into clusters with affinity propagation algorithm [23] which aims to let free cluster's number and do not force a new TC to a not homogenous group. A multi-threading research [29] becomes more intelligent and the performances are improving especially the run time. Another property of this set is the dynamism, thereby it can be static (fixed on a training phase) or semi-incremental so it can be extended by some of the input TCs having either a satisfying segmentation result (see Eq. 8) or when the distance between the best model and the TC (see Eq. 5) is over a fixed threshold (which means that for this TC, there is not a reproached model).

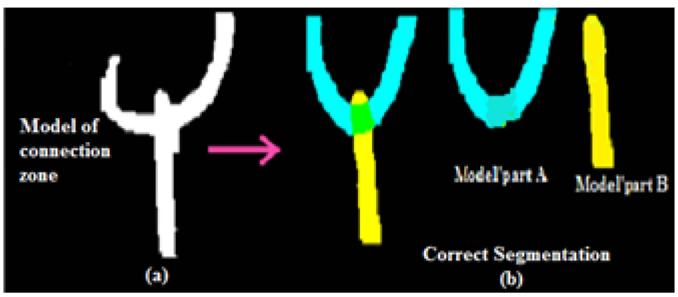

Fig. 2. An entry of dictionary (a) Model of TC, (b) Correct segmentation: two isolated parts A and B with the same size as the model (120×120).

In what follows, the focus is on steps of the most similar model selection and TC segmentation. We first explain how to find the appropriate model for an input TC, based on matching using the Shape Context descriptor. We also show how to handle with the outliers pixels problem by proposing regions matching instead of point to point matching. We then clarify how to estimate the aligning transform and compute similarity between TC and its model. We finally clear the step of TC segmentation based on central points of its model's parts.

A. *Selection of the most similar model*

To find the most similar model for an input TC, this latter is compared to all clusters representative models in the training set. We used the shape matching descriptor proposed by Belongie in [25] where similarity is computed by: a) solving the correspondence between TC and models by SC descriptor and b) estimating the aligning transform that maps one on the other by TPS function.

1) Correspondences with shape context: To match an input TC with models, stored in the dictionary, we compared their edge point's SC histogram. Canny [26] followed by sampling process were applied to get a fixed number of contour's point and make computation simpler. The SC at a point captures the distribution over relative positions of other shape points. Fig. 3 shows an example of two TCs and how to make correspondences with their SCs.

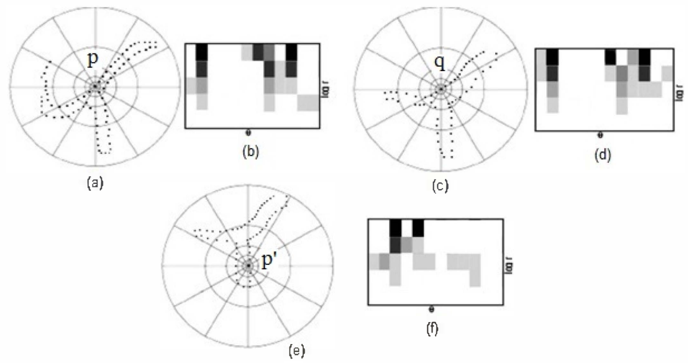

Fig. 3. Matching with SC (a) SC of p in TC_1,(b) log-polar histogram for p (c) SC of q in TC_2, (d) log-polar histogram for q which is similar to that in (b), but different of p' in (f). the best matching is between p in TC_1 and q in TC_2. Log-polar histogram similarity is computed according to the χ^2 distance. Black bins correspond to a higher number of pixels in that bin, gray bins contains fewer pixels than the black cells.

Our system computes, for each couple of points (p_i, q_j) where $p_i \in$ TC and $q_j \in$ Model, the matching cost by the χ^2 statistic test (see Eq. 1) and gets an $N \times N$ cost matrix C where $h_i(k)$ and $h_j(k)$ denote the K-bin normalized histograms at p_i and q_j.

$$C_{ij} = C(p_i, q_j) = \frac{1}{2}\sum_{k=1}^{K}\frac{[h_i(k) - h_j(k)]^2}{[h_i(k) + h_j(k)]} \tag{1}$$

In this step, finding the best correspondence between the shape's points becomes an assignment problem. The input is the cost matrix C with entries C_{ij} and the objective is to find a permutation π minimizing the total cost of matching C. To find the permutation π (see Eq. 2) minimizing the total cost of matching we applied the Jonker method [27] and got the correspondence between pixels of TC and models.

$$C = \sum_i C(p_i, q_{\pi(i)}) \tag{2}$$

- Outliers pixels: Even if the SC is a rich descriptor, the presence of outliers can affect the matching result especially when dealing with transformation. So, contrarily to the point to point matching process, as proposed by Belongie in [25] which takes into account all pairs (p, q) minimizing the cost defined in Eq. 2, we only considered the couples of well matched points (WMP) as shown in Eq. 3. The couple (p, q) qualified as outlier, will not be used in the aligning transform. From this point of view, we first, introduced a new definition of outlier pixels, proposed a constraint to filter the matching results and eliminated the doubtful ones. We then developed a new form of matching based on regions.
- Regions matching: Let denote MatchPoints (MP), the set of couples of pixels, using Eq. 2 as done by Belongie [25]. WMP, defines the couples of points

(p_i, q_j) such as:

$$V_P(p_i, r) \times V_Q(q_j, r) \subseteq \text{MP} \quad \text{with} \quad 1 \leq i, j \leq N$$
$$\text{and}$$
$$\text{Card}(V_P(p_i, r)) = \text{Card}(V_Q(q_j, r)) \geq n$$
$$\text{where}$$
$$V_H(h, r) = \{h' \neq h / h' \in H \text{ and } d(h, h') \leq r\} \tag{3}$$

P and Q are respectively a set of contour's points with size N, d is the Euclidean distance, r is the radius of circle centered on p_i or on q_j which defines the neighborhood region and n is the number of pixels in circle that must be matched. Table I displays parameters used to select the best model.

TABLE I. USED PARAMETERS.

Parameter	*Value*	*Interpretation*
N	200	Contour point
Neighbors	30	Pixels in circle of radius r (Neighborhood region:empirically evaluated)
Threshold_Neighbors	10	Number of pixels required to validate the matching (empirically evaluated)
Coeff	0.2	Empirical coefficient

Having correspondences between contours points, helps to get the SC distance called (D_{sc}) as explained in section C.

2) Estimation of the aligning transform: Given the matching result WMP, a transformation mapping the second set (Model) on the first set (TC) is necessary for two reasons: 1) to compute the similarity (D_{be}) and 2) to explore parameters of transformation in the segmentation step. Several transformation classes are available like polynomial, affine or Thin Plate Spline (TPS). We choose the Bookstein method [24] which is widely used for flexible coordinate transformation. In Fig. 4, an example of mapping the model on the TC in order to compute the bending energy (D_{be}).

Fig. 4. Model interpolated on TC (inter-word connection) in order to compute bending energy $(D_{be} = 0.010)$.

3) Similarity computing : The best model is the one which has the highest similarity with the input TC. To compute similarity between TC and all models stored in dictionary, three parameters are involved:

- The shape context cost (D_{sc}).
- The percentage (N_{match}) of WMP using region to region matching as explained in Eq. 3.
- The bending energy (D_{be})

The SC cost and the bending energy are used by Belongie in [25] for similarity computing (see Eq. 4).

$$\text{Similarity} = -D \text{ and } D = 1.6D_{ac} + D_{sc} + 0.3D_{be} \tag{4}$$

In this work, we have slightly modified the similarity computing (see Eq. 4) and we defined an algebraic distance including the number of WMP. In fact, we involved the percentage of the valid matched points N_{match} with a coefficient (Coeff, an empirically evaluated coefficient for computing similarity) as shown in Eq. 5. Note that we do not considered the image appearance D_{ac}, as done by [25], because we handle with binary images.

$$\text{Similarity} = -D \text{ and } D = D_{sc} + 0.3D_{be} - \text{coeff} * N_{match} \tag{5}$$

$$N_{match} = \frac{\text{Number of WMP}}{N} \tag{6}$$

N_{match} is a reliable indicator of the quality of the matching result and the term $N_{match} * \text{coeff}$ makes more accurate the distance. If the ratio N_{match} is large, it means that the model approaches the TC so the distance between them will decrease. But if the ratio is small, that means that the bending energy (D_{be}) of transformation function is not so significant since aligning transform will only act on a reduced set of couples. The term $N_{match} * \text{coeff}$ is used to balance the equation.

B. TC Segmentation

At this step, we have the most similar model, its correct segmentation (the two model's parts: parts A and B) and the TPS parameters. Using transformation with the same parameters, we can say that the two Model's parts A and B are almost similar to those of the TC. Let (P_a, P_b) be respectively the contours of the two model's parts after their transformation with the estimated TPS. Let (c_a, c_b) be their respective central points (see example in Fig 5). These centers are approximately those of the parts composing the input TC. Thus, to segment TC and reconstitute its parts, our system associates pixels to the closest center. However, the assignment can be ambiguous for the shared zone and the risk to assign pixels to the wrong part exists (see Fig 6, row(a)). For this reason, centers should be adjusted before being used. The system identifies the intersection points (I_a, I_b) of the segment L, connecting the two centers, with (P_a, P_b) respectively as defined in Eq. 7. Using each intersection point, the system creates a new center as the symmetric of the nearest one to the intersection and based on that, the attribution of pixels will be performed (see Fig 6 row(b)).

$$I_a = L \cap P_a \quad \text{and} \quad I_b = L \cap P_b \tag{7}$$

More precisely, if two parts are connected, to separate them and affect pixels to each part according to the closest center, a natural judgment says that we must measure the degree of emergency of each part in the other. To determine Part A of the TC, we first compare $d(C_a - I_b)$ to $d(C_b - I_b)$. If the first term is the highest, C_a changes to be C'_a as the symmetric of C_b according to I_b on L. Else, C_b moves to C'_b as the symmetric of C_a according to I_b on L. Afterwards, new centers are putted on TC and TC's part A will include all closest pixels nearest C_a or C'_a compared to C_b or C'_b (d is Euclidean, Canberra or Manhattan distance). The same treatment will be done on

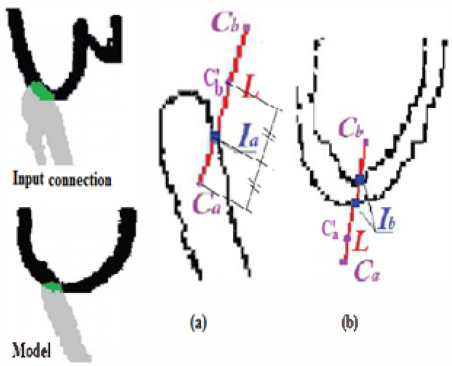

Fig. 5. (a) Contour of the part A after TPS transformation, (b) Contour of the part B after TPS transformation.

part B with based on comparing distance between $d(C_b, I_a)$ and $d(C_a - I_a)$. Finally, the shared zone is the intersection between part A and part B.

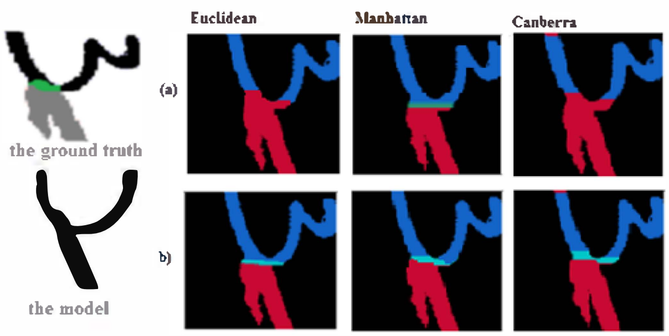

Fig. 6. Segmentation results: (a) Without centers transform, (b) With centers transform.

Note that this segmentation scheme is language independent. We just need to adjust the training set for the given scrip. Also, we do not have to find a very similar model for the input TC or make a perfect warping as done by Kang in [20]. It is sufficient that shapes are approached to have similar geometrical features. In addition, this proposed approach can be used not only to segment TCs in successive text-lines but also those which can appear between letters of the same word or even between letters of neighboring words in the same text-line.

III. Experiment Results and Discussion

For experimentation, we used 900 samples for training set (whose distribution accordingly to the connection's type is 220 for inter-word and 680 for inter-lines) and 400 for test. To evaluate the segmentation result, we used match score method (see Eq. 8) for each resulted segment [28]. It is based on a comparison of the segmentation of the input TC to its ground truth. Let (G_A, G_B) be the sets of all points inside the ground truth segments, (R_A, R_B) the set of all points inside the result segments and $T(s)$ a function that counts the elements of set s. MS_i represents the matching results of the i ground truth region and the i result region.

$$\text{MatchScore} = \frac{2 * MS_A * MS_B}{MS_A + MS_B} \quad (8)$$

$$MS_i = \frac{T(G_i \cap R_i)}{T(G_i \cup R_i)} \quad (9)$$

Fig. 8 resumes all steps of TC segmentation in Fig 8(a), with the Euclidean and the Manhattan distances where the point's distribution is respectively described by the gravity center and the mid-point. Results in Table II are the rates of accepted segmentation for a matching score upper than 0.8 and are satisfying for the three metrics. Note that the shapes of parts, constructed with the Manhattan distance, are more similar to their ground truth (see Fig. 6) than those obtained by Euclidean and Canberra distances. Using the Euclidean distance, the TC segmentation is similar to cutting according to the orthogonal on the segment connecting the two gravity points of the model's parts. So, the Manhattan distance is more suitable for a better character recognition. Otherwise, in Canberra distance one of the two parts is over segmented in two fragments (see the red fragment at the top of the blue part in the Canberra column of Fig. 6). That would distort text-line and word segmentation steps. Some problems, reported in [20], by Kang, are due to steps preceding the segmentation (the connection's extraction, recognition and transformation) are solved by this method because it is not so sensitive to the model's shape and these errors have a low rate especially for the transform deviation. For more details, see [1] and [2].

TABLE II. Used Parameters.

Metric	*Euclidean*	*Manhattan*	*Canberra*
Correct segmentation rate	92.6%	94%	92%

This method has been also tested on few Latin TCs (200 samples), since there 'is no ground truth TC dataset (see example in Fig. 7), results are good but, method needed to be tested on a ground truth set.

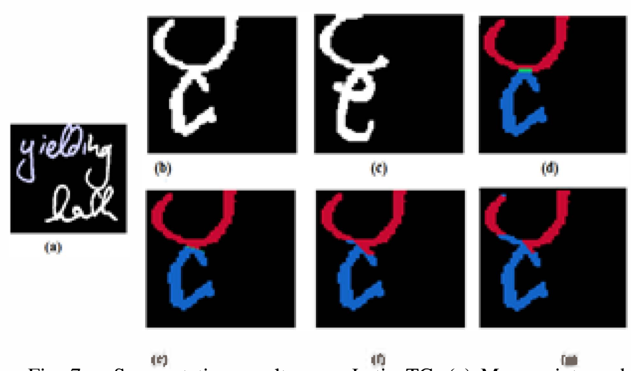

Fig. 7. Segmentation results on a Latin TC: (a) Manuscript used in [13], (b) TC, (c) selected model, (d) ground truth, (e),(f) and (g) are segmentation with Euclidean (Ms=0.98), with Manhattan (Ms=O.95) and with Canberra (Ms=0.92).

IV. Conclusion

In this paper we proposed a novel segmentation method for TC segmentation in unconstrained Arabic handwritten text. It mainly consists of two steps: finding the most similar model to an input TC and segment this latter based on central points of the two model's parts. This proposed method has the benefit to segments connections between successive text-lines as well as between neighboring words. It is also qualified

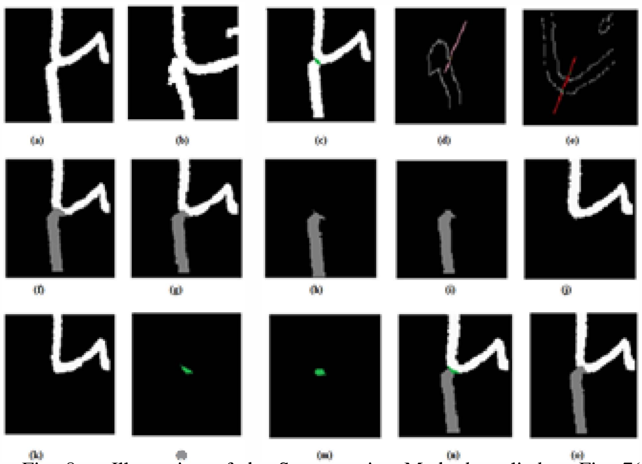

Fig. 8. Illustration of the Segmentation Method applied to Fig. 7(a) TC connection. (b) best selected model: similarity=0.26, WMP=148. (c) Ground truth. (d) Model's part A after transformation with TPS. (e) Model's part B after transformation with TPS, (f) Segmentation without center correction using Ed and Gravity center: MS= 0.917. (g) Segmentation without center correction using Manhattan distance and Midpoint, MS=0.949. (h) Extraction of the part A of the input connection using Euclidean distance after correction of Gravity center, C_a (60,75), New center C_b'(76,47). (i) Extraction of the part A of the input connection using Manhattan distance after correction of Midpoint: C_A (60,75), New center C_b'(76,47). (j) Extraction of the part B of the input connection using Euclidean distance after correction of Gravity center, C_a (60,75), New center C_b'(72,59), (k) Result of extraction of the part B of the TC connection using manhattan distance after correction of midpoints: C_A (61,75), new center C_b'(71,59). (l) extraction of the shared zone after correction of Gravity center, (m) Extraction of the shared zone after correction of Midpoint. (n) Final result of segmentation using Euclidean distance after correction of Gravity center, MS=0.980 (n) Final result of segmentation using Manhattan distance after correction of Midpoint, MS=0.987. C_a (60,75), C_b (80,42), I_a (66,67), I_b (68,61) are respectively centers of part A, part B, intersection of Part A with L and intersection of part B with L according to Euclidean distance. C_a (61,75), C_b (75,45.5), I_a (66,67), I_b (66,67) are respectively centers of part A, part B, intersection of part A with L and intersection of part B with L according to Manhattan distance.)

as a recognition-based on SC's region matching. The method is not dedicates to specific writers, styles or alphabets. It is also script independent. Experiments confirm effectiveness of the proposed method.

References

[1] N. Aouadi, A. Kacem, A. Belaïd, *Segmentation of Touching Component in Arabic Manuscripts*, In Proc. of ICFHR, 2014.

[2] N. Aouadi, S. Amiri, A. Kacem, *Segmentation of connected Component in Arabic Handwritten Documents*, In Proc. of CIMTA, Elsevier, Vol. 10, pp. 738-746, 2013.

[3] L. Likforman-Sulem, A. Zahour and B. Taconet, *Text-line segmentation of historical documents: a survey*, In IJDAR, pp. 123-138, 2007.

[4] L. Likforman-Sulem and C. Faure, *Extracting lines on handwritten documents by perceptual grouping*, published in Advances in Handwriting and Drawing: A Multidisciplinary Approach, pp. 21-38. Europia, Paris, 1994.

[5] L. Likforman-Sulem, C. Faure, *Une méthode de résolution des conflits d'alignements pour la segmentation des documents manuscrits*, Traitement du Signal, vol. 12, No. 6, pp. 541-549, 1995.

[6] A. Lemaitre, J. Camillerapp, B. Cousnon, *A perceptive method for handwritten text segmentation*, In Proc. of DRR XVIII - Electronic Imaging, San Francisco - United States, pp. 1-10, 2011.

[7] Y. H. Tseng and H. J *Recognition-based handwritten Chinese character segmentation using a probabilistic Viterbi algorithm*, Pattern Recognit. Lett. 20(8), pp. 791-806, 1999.

[8] A. Zahour, B. Taconet, P. Mercy, and S. Ramdane, *Arabic hand-written text-line extraction*, In Proc. of ICDAR, Seattle, pp. 281-285, 2001.

[9] U. Pal and S. Datta S,*Segmentation of Bangla unconstrained handwritten text*, In Proc. of ICDAR, pp. 1128-1132, 2003.

[10] A. Zahour, B. Taconet, S. Ramdane, *Contribution la segmentation de textes manuscrits anciens*, In Proc. of CIFED, La Rochelle, 2004.

[11] G. L. Ouloudis, B. Gatos, I. Pratikakis, C. Halatsis, *Text line and word segmentation of handwritten documents*, Pattern Recognition vol. 42, No. 12, pp. 3169-3183, 2009.

[12] J. Kumar, W. Abd-Almageed, L. Kang, D. S. Doermann, *Handwritten Arabic Text Line Segmentation using Affinity Propagation*, In Proc. of 9th IAPR International Workshop on Document Analysis Systems, New york, pp. 135-142, 2010.

[13] A. Alaei, U. Pal, P. Nagabhushan, *A new scheme for unconstrained handwritten text-line segmentation*, Pattern Recognition, vol. 44, No. 4, pp. 917-928, 2011.

[14] Chen, Y. A. et Leedham, G, *Independent component analysis segmentation algorithm*, In Proc. of ICDAR, pp. 680-684.

[15] P. Vassilis, S. Themos, K. Vassilis, C. George, *Handwritten document image segmentation into text lines and words, Pattern Recognition*, vol. 43, No. 1, pp. 369-377, 2010.

[16] P. Piquin, C. Viard-Gaudin, D. Barba, *Coopération des outils de segmentation et de binarisation de documents*, In Proc. of Colloque National sur l'Ecrit et le Document, Rouen, pp. 283-292, 1994.

[17] E. Bruzzone, M.C. Coffetti, *An algorithm for extracting cursive text lines*, In Proc. of ICDAR, 20-22 Sept., pp. 749-752, 1999.

[18] K. Takru and G. Leedham, *Separation of touching and overlapping words in adjacent lines of handwritten text*, In Proc. of Eighth International Workshop on Frontiers in Handwriting Recognition, Ontario, Canada, pp. 496-501, 2002.

[19] S. Rohini, R. S. Uma Devi, S. Mohanavel, *Segmentation of Touching, Overlapping, Skewed and Short Handwritten Text Lines*, International Journal of Computer Applications, vol. 49, No 19, pp. 24–27, 2012.

[20] L. Kang, D. Doermann,*Template based segmentation of touching components in handwritten text-lines*, In Proc. of ICDAR, Beijining China, pp. 417-422, 2011.

[21] N. Ouwayed, A. Belaïd, *Separation of overlapping and touching lines within handwritten arabic documents*, In Proc. of the 13th International Conference on Computer Analysis of Images and Patterns, Münster (North Rhine-Westphalia), Germany, pp. 237-244, 2009.

[22] J. Kumar, L. Kang, D. S. Doermann, W. Abd-Almageed, *Segmentation of Handwritten Text-lines in Presence of Touching Components*, In Proc. of ICDAR, Beijining China, pp. 109-113, 2011.

[23] B.J. Frey and D. Dueck, *Clustering by passing messages between data points*, Science, 315:972-976, 2007.

[24] F. L. Bookstein, *Principal Warps: Thin-Plane Spline and the Decomposition of Deformations*, IEEE transactions on pattern analysis and machine intelligence, 1989.

[25] S. Belongie, J. Malik, J. Puzicha, *Shape Matching and object Recognition Using Shape Context*, IEEE transactions on pattern analysis and machine intelligence, pp. 509-522, 2002.

[26] J Canny, *A computational approach to edge detection*, IEEE Trans. Pattern Anal. Mach. Intell., vol. 8, No. 6, pp. 679-698, 1986.

[27] R. Jonker, A. Volgenant, *A Shortest Augmentating Path Algorithm for Dense and Sparse Linear Assignment Problem*, Journal Computing, vol. 38, No. 4, March 1987.

[28] B. Gatos, N. Stamatopoulos, G. Louloudis, *ICDAR handwriting segmentation contest*, In Proc. of ICDAR, 2009.

[29] http://www.math.uni-6hamburg.de/doc/java/tutorial/essential/threads/group.html

[30] D. Farnandez-Mota, J. Lldos and A. Fornes, *Graph based Approach for Segmenting Touching Lines in Historical Handwritten Documents*, IJDAR, 2014.