

Segmentation of Touching Component in Arabic Manuscripts

Nabil Aouadi, Afef Kacem, Abdel Belaïd

► **To cite this version:**

Nabil Aouadi, Afef Kacem, Abdel Belaïd. Segmentation of Touching Component in Arabic Manuscripts. International Conference on Frontiers in Handwriting Recognition, Sep 2014, Crète, Greece. <10.1109/ICFHR.2014.82>. <hal-01111750>

HAL Id: hal-01111750

<https://hal.inria.fr/hal-01111750>

Submitted on 3 Feb 2015

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Public Domain}

Segmentation of Touching Component in Arabic Manuscripts

N. Aouadi, A. Kacem
 University of Tunis, LaTICE-ENSIT
 Tunis, Tunisia
 Nabil.aouadi@utic.rnu.tn
 Afef.kacem@esstt.rnu.tn

A. Belaïd
 University of Lorraine, LORIA
 Nancy, France
 abdel.belaid@loria.fr

Abstract— Touching components are connection zones occurring between text-lines or words of the same line and are one of the problems that make unconstrained handwritten text segmentation greatly hard. In this paper, we propose a recognition based method to separate these components once localized in Arabic manuscript images. It first identifies, for a given touching component, a similar model stored in a dictionary with its correct segmentation, using shape context descriptor and an interpolation function. Then, it segment the touching component based on the distance from the midpoints of the identified model's parts. Tests are performed using a database of touching components and two metrics: Manhattan and Euclidean distances. Experimental results show the effectiveness of the proposed segmentation method.

Keywords—Arabic manuscript segmentation; Touching component; Shape context descriptor; Midpoint.

I. INTRODUCTION

In the context of unconstrained handwriting, text-lines are hard to segment especially when manuscript pages include narrow spaced lines with overlapping and touching components (TC). In manuscripts, characters and words may have unusual and varying spaces, depending on the writer. Text-line extraction would ideally process manuscript images without TCs. This problem is especially obvious in Arabic manuscripts (see Fig. 1) since most of Arabic letters (21 from 28) are ascendant or descendant and contain special marks and dots (see Fig. 2(a)). Notice that TC generally happens when interlines spacing is narrow (see Fig. 2(b)) or when the calligraphy contains big jams (descenders) (see Fig. 2(c)).

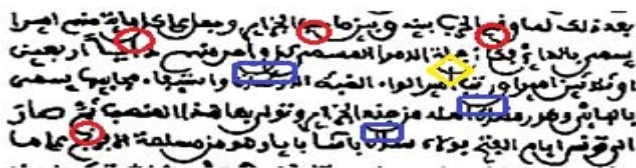


Fig. 1. Example of old Arabic manuscript having touching components.

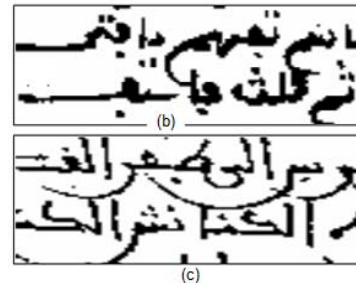


Fig. 2. (a) Arabic letters, (b) and (c) Examples of connections.

To separate connected components there are many approaches based on component structural analysis. The process, as reported in [2], starts by detecting the intersection points. Then, it looks for the starting point of the ligature which is generally the highest point near the baseline of the top line. Next, it follows, from this point, the descending character respecting an angular variation which corresponds to the curvature of the descending direction. Others approaches look for the best segmentation using recognition based on template matching. In [3], authors used templates formed by TCs with their correct segmentations. They computed similarity, using shape context descriptor, to find the best match in templates, and then they applied an overlaying technique. Note that recognition based approaches are more robust against the irregularity aspect of TCs but they only success when the correct segmentation exists in the candidates set. The method, proposed in this paper, concerns segmentation of TCs in unconstrained Arabic handwritten text. We assume that TCs are already localized in manuscript images. In fact, many works [2, 6] are able to extract these components. The main idea is to separate the connection by approximation to models stored in a dictionary (see Fig. 3) with their known correct segmentation. Thus, there are two

fundamental steps before TC can be segmented: 1) the recognition step to find the most similar model stored in a dictionary and 2) the approximation step to estimate the transformation aligning the selected model to the TC. Finally, we use the midpoints of the model's parts to segment the TC.

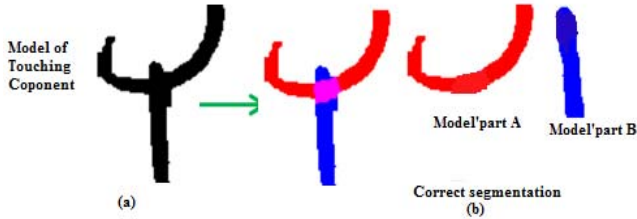


Fig. 3. An entry of dictionary (a) Model of TC, (b) Correct segmentation.

The remainder of the paper is organized as follows. Section II describes some related works. In section III, the recognition and the transform steps for detecting the best similar segmented model are presented. Section IV is dedicated to TC segmentation using midpoints. Experimental results are reported in section V and some conclusions are drawn in section VI.

II. RELATED WORKS

Many text-line segmentation methods have been reported in the literature, but most of them do not mention the problem of TCs or how to solve it. The process of TC separation passes unnoticed or be treated as second-class. Table I summarizes some related works.

TABLE I. METHODS OF TC SEGMENTATION IN MANUSCRIPTS.

	Ref.	Used concept	Script	Results
Recognition-free approaches	[1]	Projection based	Latin	-
	[2]	Angular variation	Arabic	Avg. 94%
	[4]	Geometrical constraint	Latin	-
	[5]	Base lines	Latin	-
	[6]	Convex hull	Arabic	-
	[7]	Distance from the center of gravity	Arabic	-
	[8]	Statistical method	-	-
	[9]	Structural knowledge	Latin	-
	[10]	Run length algorithm	Latin	-
	[11]	Contour based shape decomposition	Arabic	71.9%
	[12]	Contour tracing	-	82%
	Recognition-based approaches	[3]	Dictionary, Shape context and TPS transformation	Arabic
[19]		Dictionary and decision tree	-	90%

Hereafter, we discuss our proposed TC segmentation method based on three main steps: recognition, transformation and segmentation.

III. RECOGNITION AND TRANSFORMATION

The objective of recognition and transformation is to find, for an input TC, the best similar model, stored in the dictionary, and to estimate a transformation aligning them. To find the appropriate model, our system compares the input TC

to elements of the training set (models) and selects the most similar one. Comparison is made based on the shape matching descriptor proposed in [14]. It covers both steps of recognition and transformation since the similarity is computed by solving the correspondence between an input TC and models and then estimating the aligning transform. Shape Context (SC) is a feature descriptor used in object recognition. It is captured by a finite subset of its edge points. The SC of a point characterizes the configuration of the entire shape relative to this reference point. It is a histogram computed from logarithmic coordinates. Notice also that SC is intended to be a way of describing shapes that allows for measuring shape similarity and the recovering of point correspondences As can be seen in Fig. 4, since Fig. 4 (d) and Fig. 4 (e) are the shape contexts for two closely related points, they are quite similar, while the shape context in (f) is very different.

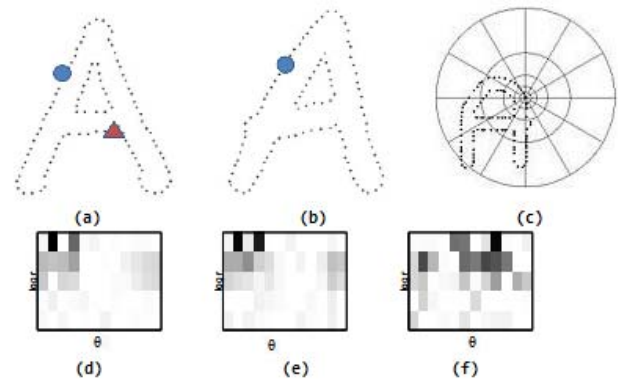


Fig. 4. (a) and (b) are the sampled edge points of the two shapes, (c) is the diagram of the log-polar bins used to compute the shape context, (d) is the shape context for the circle, (e) is that for the diamond, and (f) is that for the triangle.

A. Training set

Our system is trained with different models of TCs, of size 120 * 120 pixels, using a segmented connection database. These models can be saved without any hierarchical structure (see Fig. 5(a)) or organized into levels with a representative element for each group or cluster (see Fig. 5 (b)). In the first case, the research is a browsing of the whole training set. But, in the second case and with affinity propagation algorithm, as proposed in [17], the research for the best model is based on comparison with only the representative element of each cluster. Thus, research becomes more intelligent and the run time gets better. Also note the dynamic aspect of the training set. Thereby it can be static (fixed on a training phase) or dynamic so it can be extended by further input connections having a satisfying segmentation result (see Eq. 7).

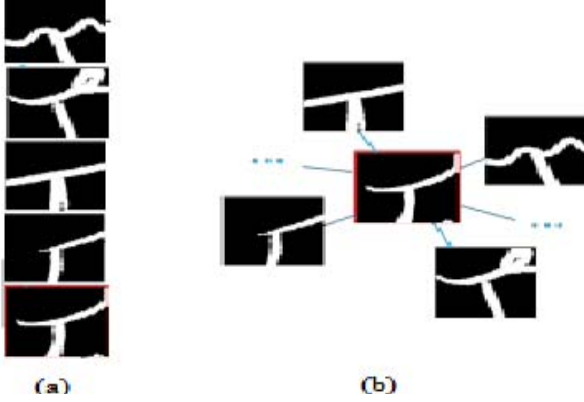


Fig. 5. (a) TCs saved without any hierarchical structure (b) TCs organized into cluster with representative element.

B. Shape Matching

To match two TCs, the system compares their edge point's shape context histogram, as illustrated in Fig. 4. To extract edge points, we applied the Canny edge detector [18] followed by a sampling process to get a fixed number of contour's point and make computation easier. Fig. 6(a), Fig. 6(b) and Fig. 6(c) show respectively an example of an input TC, its extracted edge point and the shape context histogram of a point M . We used 5 bins for $\log r$ and 12 bins for θ .

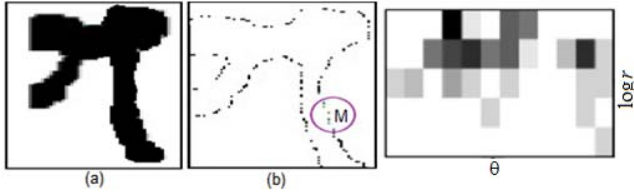


Fig. 6. (a) Input TC (b), Extracted edge points, (c) Point's shape context histogram.

The shape context at a point captures the distribution over relative positions of other shape points. For each point p_i on the shape, we compute a coarse histogram of the relative coordinates of the remaining $n-1$ points. To compute the shape context for all N edge points and get the shape descriptor, we combined vectors of all point's shape context. It is an $N * P$ matrix where P is the product of the θ and the distance bins (60 in general). Using two shape descriptors (one for an input connection and the second for a model stored in the dictionary), the system computes, for each couple of points (p_i, q_j) from the two shapes, the matching cost by the χ^2 statistic test (see Eq. 1) and gets an $N * N$ cost matrix C where $h_i(k)$ and $h_j(k)$ denote the K -bin normalized histograms at p_i and q_j . K is fixed to 60.

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

In this step, finding the correspondences between the shape's points becomes an assignment problem, which can be solved by the method proposed in [15]. The input is the matrix

C and the objective is to find the set of couples (p, q) minimizing the cost.

C. Outliers pixels

Even if the shape context is a rich descriptor, the presence of outliers is a problem that can affect the matching result especially when working with small and finical shapes. For this reason, we defined a neighborhood relation between points of the same shape. Hence, for every point M , we also considered its neighbors (see Fig. 6(b)) in the objective to remove points having a doubtful matching. So, contrarily to the point to point matching process, as proposed in [14] which takes into account all pairs (p, q) minimizing the cost, we only considered the points well matched according to neighbors' relation. So, a matching between the points: p and q is considered valid if and only if a sufficient number of p neighbors are associated to a sufficient number of q neighbors. Otherwise the couple (p, q) is considered as outliers and it will not be used in next steps. This amounts to a matching by set of points instead of a point to point matching.

D. Estimation of the Aligning Transform

The estimation of the plane transformation between two shapes accordingly to the matching result (couples (p, q) minimizing the cost in Eq. 1) is necessary for two reasons: first for computing the similarity (D_{be}) and then for transforming the separated parts of the model to approximately coincide into the input connection (and so they will have close geometrical features). There are several transformation techniques, among them the affine transform which is the most known and the Thin Plate Spline (TPS) which is widely used for flexible coordinate transformation. In our system we used the TPS transformation [16] as an interpolation technique. Hence, the system computes from two sets of points, having a correspondence relation, the function f which is defined everywhere in the \mathbb{R}^2 space by:

$$f(x, y) = a_1 + a_x x + a_y y + w_i U(|P_i - (x, y)|) \quad (2)$$

where $U(r) = r^2 \log r$ and $\sum_{i=1}^N w_i = \sum_{i=1}^N w_i x_i = \sum_{i=1}^N w_i y_i = 0$ verifies the relations below. This TPS interpolating $f(x, y)$ minimizes the bending energy I_f .

$$I_f = \iint_{\mathbb{R}^2} \left(\left(\frac{\partial^2 f}{\partial x^2} \right)^2 + 2 \left(\frac{\partial^2 f}{\partial x \partial y} \right)^2 + \left(\frac{\partial^2 f}{\partial y^2} \right)^2 \right) dx dy \quad (3)$$

E. Computing Similarity

To measure the similarity between two shapes, we used three parameters:

- The shape context cost (D_{sc}),
- The number of matched points maintained after the neighborhood test (N_{match}) and
- The bending energy (D_{be}).

The shape context cost and the bending energy are used in the similarity defined in [14] as the distance between shapes where

D_{ac} is the image appearance distance, D_{sc} is the shape context distance and D_{be} is the bending energy.

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

In this work, we have slightly modified the classic similarity measure as shown in Eq. 4 to include the number of points well matched. We did not use D_{ac} because it is not really significant for binary images. Instead, we involved the percentage of the valid matched points N_{match} with an empirically evaluated coefficient. So the used similarity is defined in Eq. 5.

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

For the input TC (see Fig. 7(a)) and its best selected model (see Fig. 6(b)), let's compute similarity for this example according to both equations. The matched point is equal to 200. The similarity is evaluated to $S = -0.116847$ according to Eq. 4. Using Eq. 5, the similarity has significantly increased to -0.0568 , where the number of well matched points is 139 (using neighborhood relation) and coeff is usually static, evaluated to 0.2.



Fig. 7. (a) Input connection, (b) the selected best model.

IV. SEGMENTATION

A. Shape Description with the distance from midpoint

Here we are interested to analyze a point set distribution. In what follows, we consider P a set of 2D points distributed on a space E , defined by their coordinates. To evaluate the relative positions and characterize P , we used two metrics: Euclidean and Manhattan distances. Using a metric, we can define a central point to characterize the distribution of a point set. The most known central point is the gravity center. It minimizes the sum of squared Euclidean distances. The midpoint, unlike the gravity center, is defined by a general property and not a particular formula. For a given point set P to which we associate the metric D , the midpoint M is the most accessible point (the one that minimizes the sum of the distances of all the set). This point is easy to find if the used metric D is the Manhattan distance, but to find it we have to browse the whole space E .

B. Segmentation using midpoints

At this stage, we have the most similar model, its correct segmentation and the estimated TPS parameters. Let's consider (P_a, P_b) the contours of the models separated parts after the transformation with the estimated TPS and (C_a, C_b) their respective midpoints (see example in Fig 8).

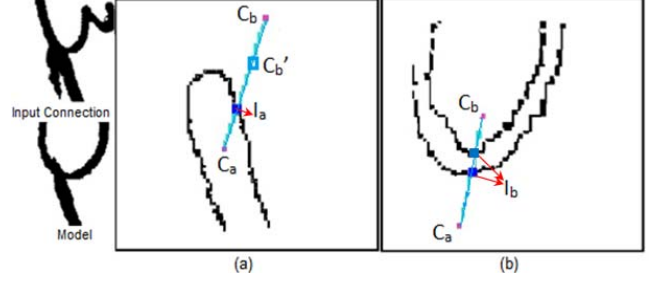


Fig. 8. (a) Contour of the part A after TPS transformation, (b) Contour of the part B after TPS transformation.

These midpoints are approximately those of the parts composing the input connection. Thus to reconstitute these parts, the system associates pixels to the closest midpoint. For points in the touching zone, the assignment can be ambiguous and they risk to be assigned to the wrong part (see Fig. 9(a)). For that reason, midpoints must be adjusted before being used. The system identifies the intersection points (I_a, I_b) between the line L , connecting the two initial midpoints, and (P_a, P_b) respectively as indicated in Eq. 6. Using each intersection point, the system creates an adjusted midpoint as the symmetric of the nearest one to the intersection (see C_b and C_b' in Fig. 8(a)) and based on that, the pixel attribution to parts A and B will be performed (see Fig. 9 (b)).

$$I_a = L \cap P_a \text{ and } I_b = L \cap P_b \quad (6)$$

Extracting the shared zone for the two parts is important for the character recognition. To detect this common area, the system uses the intersection point I_b of the line connecting the initial midpoints and the model modified part P_b to compute the new midpoint C'_a and identify the part P_a of the input connection and vice versa for the other part. The advantages of this segmentation scheme are as follows. First, we do not need to study the connections relative to the language. It is only necessary to use the appropriate training set. Second, we do not have to find the same shape or make a perfect warping. It is sufficient that shapes are approached to have similar geometrical features.

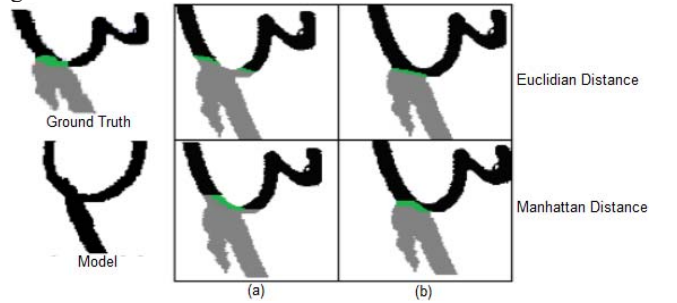


Fig. 9. Segmentation results: (a) Without centers transform, (b) With centers transform.

V. EXPERIMENT RESULTS AND DISCUSSION

To test the proposed segmentation method, we used 840 TCs whose distribution accordingly to the connection's type is 220 for inter-word and 620 for inter-lines. Some of them are

extracted manually from Arabic documents belonging to the Tunisian National Archive. Others are selected from existing database [3].

In the recognition and transformation stages, we used a multi-threading approach, which reduces the computational time to almost the half. Values of used parameters are set as shown in Table II, where N is the point set's size defining the shape's contour for each TC. $Neighbors$ is the number of points defining the neighborhood of each point (as shown in circle in Fig. 6(b)) and $Threshold_Neighbors$ is the minimum number of point's neighbors well assigned to consider the matching as valid. $Coeff$ is explained in section III, used to compute similarity (empirically fixed).

TABLE II. USED PARAMETERS.

Parameter	Value
N	200
$Neighbors$	30
$Threshold_Neighbors$	10
$Coeff$	0.2

To evaluate the segmentation results, we used match score method for each resulted segment [13]. 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.

$$MatchScore = \frac{2 * MS_A * MS_B}{(MS_A + MS_B)} \quad (7)$$

$$MS_i = \frac{T(G_i \cap R_i)}{T(G_i \cup R_i)} \quad (8)$$

TABLE III. TESTS RESULTS.

Metric	Euclidean	Manhattan
Segmentation rate	92.6%	94%

The proposed segmentation method is tested with the Euclidean and the Manhattan distances, where the point's distribution is respectively described by the gravity center and the midpoint. The results in Table III are the rates of accepted segmentation for a matching score upper than 0.8. We think that obtained results are satisfying using the two metrics. The shapes of the segments, constructed with the Manhattan distance, are more similar to their ground truth (see Fig. 10). From this finding, Manhattan distance is more appropriate in our case, because the pictures are more similar to a grid plan. Using the Euclidean distance, the separation of the TC is similar to cutting according to the orthogonal on the line connecting the two gravity points of the model's components. Thus, the Manhattan distance is more suitable for a better character recognition. Fig 11, illustrates the kinds of segmentation errors mentioned in [3]: (a) Bad template, (b) Ambiguity, (c) Noise (disturbing components) and (d) Transform deviation. These problems are due to the steps prior the segmentation (the connection's extraction, recognition and

transformation). It is convenient to note that the segmentation method, proposed here, is not so sensible to the shape recognition because even if not the best model was found for an input TC, the method generally succeeds to correctly segment it based on midpoints. Notice also that few are errors related to transform deviation. Moreover, shape contexts are empirically demonstrated to be robust to deformations, noise, and outliers especially when using synthetic point set matching as we done.

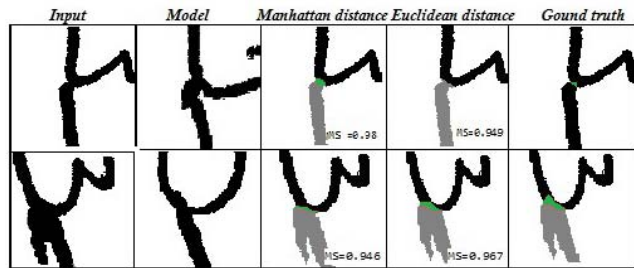


Fig. 10. Segmentation samples for Matching Score ≥ 0.9 .

VI. CONCLUSION

In this paper we proposed a segmentation method for the TCs in unconstrained Arabic handwritten text. This method can be applied for the connections between text-lines as well as between words. It is mainly based on a recognition and transformation steps in which we select a similar segmented model for the TC and estimate the transformation aligning them. Then, we use the midpoints of the Model's transformed parts for TC segmentation. Experiments confirm effectiveness of our system.

REFERENCES

- [1] L. Likforman-Sulem and A. Zahour and B. Taconet, "Text line segmentation of historical documents: a survey," IJDAR, 2007, pp. 123-138.
- [2] N. Ouwayed, A. Belaid, "Separation of Overlapping and Touching Lines within Handwritten Arabic Documents," CAIP, 2009, p. 237-244.
- [3] Le Kang and David Doermann, "Template based Segmentation of Touching Components in Handwritten Text Lines," ICDAR, 2011.
- [4] P. Vassilis, S. Themos, K. Vassilis, C. George, "Handwritten document image segmentation into text lines and words", Pattern Recognition 43(1), 2010, pp. 369-377.
- [5] A. Lemaitre, J. Camillerapp, B.Cousnon, "A perceptive method for handwritten text segmentation," DRR, Electronic Imaging, San Francisco - United States, 2011, pp. 1-10.
- [6] J. Kumar, L. Kang, D. S. Doermann, W. Abd-Almageed, "Segmentation of Handwritten Text-lines in Presence of Touching Components," ICDAR, Beijing China, 2011, pp. 109-113.
- [7] J. Kumar, W. Abd-Almageed, L. Kang, D. S. Doermann, "Handwritten Arabic Text Line Segmentation using Affinity Propagation," International Workshop on Document Analysis Systems, New York, 2010, pp. 135-142.
- [8] G. L. Ouloudis, B. Gatos, I. Pratikakis, C. Halatsis, "Text line and word segmentation of handwritten documents," Pattern Recognition 42(12), 2009, pp. 3169-3183.
- [9] K. Takru and G. Leedham, "Separation of touching and overlapping words in adjacent lines of handwritten text," International Workshop on Frontiers in Handwriting Recognition, Ontario, Canada, 2002, pp. 496-501.

- [10] S. Rohini, R. S. Uma Devi, S. Mohanavel, "Segmentation of Touching, Overlapping, Skewed and Short Handwritten Text Lines," International Journal of Computer Applications 49(19), 2012, pp. 24-27.
- [11] L. Kang, D. S. Doermann, H. Cao, R. Prasad P. Natarajan, "Local segmentation of touching characters using contour based shape decomposition," International Workshop on Document Analysis Systems, Gold Coast, Queensland, Australia, 2012, pp. 460-464.
- [12] A. Alaei, U. Pal, P. Nagabhushan, "A new scheme for unconstrained handwritten text-line segmentation," Pattern Recognition 44(4), 2011, pp. 917-928.
- [13] B. Gatos, N. Stamatopoulos, G. Louloudis, "ICDAR handwriting segmentation contest," ICDAR, 2009.
- [14] S. Belongie and J. Malik and J. Puzicha, "Shape Matching and object Recognition Using Shape Context," IEEE transactions on pattern analysis and machine intelligence, 2002, pp. 509-522.
- [15] R. Jonker and A. Volgenant, "A Shortest Augmenting Path Algorithm for Dense and Sparse Linear Assignment Problem," Journal Computing volume 38 Issue 4, March 1987.
- [16] F.L. Bookstein, "Principal Warps: Thin-Plane Spline and the Decomposition of Deformations," IEEE transactions on pattern analysis and machine intelligence, 1989.
- [17] Satu Elisa Schaefer, "Graph clustering survey," ELSEVIER, 2007.
- [18] J Canny, "A computational approach to edge detection," IEEE Trans. Pattern Anal. Mach. Intell., 8(6), 1986, pp. 679-698.
- [19] P. Piquin, C. Viard-Gaudin, D. Barba, "Cooperation des outils de segmentation et de binarisation de documents," Colloque National sur l'Écrit et le Document, Rouen, 1994, pp. 283-292.

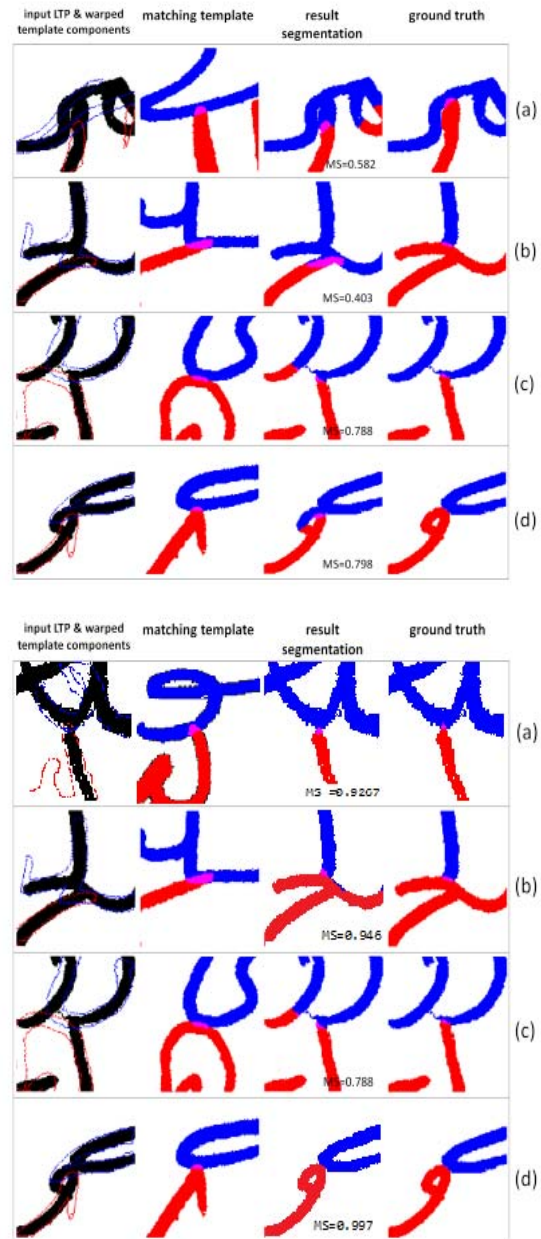


Fig. 11. Problems related to a based template method followed by obtained results.