Proceedings of the 2009 IEEE International Conference on Systems, Man, and Cybernetics
San Antonio, TX, USA - October 2009

Simulating Inertial and Centripetal Forces for Segmentation of Overlapped Handwritten Digits

Edward Roe and Carlos A.B.Mello, *IEEE Member*
Polytechnic School of Pernambuco, University of Pernambuco
Recife, Brazil
cabm@ieee.org

***Abstract*—This paper presents a new algorithm for segmentation of overlapped cursive handwritten digits. Segmentation is a common issue for automatic character recognition. The work in digits is common in applications as bank check or postal code processing. Our method is based on the segmentation achieved by the path traversed by a hypothetical ball which rolls over the digits under the influence of inertial and centripetal forces. The proposal has a computational cost of O(n^2) instead of O(2^n) as in classical graph-based approaches.**

***Keywords*—document processing, segmentation, overlapped handwritten digits**

I. INTRODUCTION

Automatic character recognition is still a challenge for applications that deal with handwritten text. In machine written documents, in general, the recognition is easy and there are a large number of tools developed for that. There are several basic issues that must be dealt for a correct recognition before the recognition phase itself. The main steps of automatic recognition systems are [9]: thresholding (in order to convert a document into black-and-white), skew estimation and correction (to avoid segmentation errors, the documents must have a zero degree inclination), noise removal (to prevent the recognition of noise as characters), segmentation (which provides the separation of the words or characters), feature extraction (the basic elements for recognition) and classification (the correct identification of each extracted symbol). Each one of theses steps is related; it is common that a fail in one step means fails in every further step.

Thresholding is the main subject of the paper [13] where a very complete survey is presented. A very robust algorithm for skew estimation and correction is presented in [3]. A technique for text segmentation and word extraction is shown in [12]. At this point however, we work in this paper with the results of this extraction in specific fields of some documents as bank checks or mail letters. It is our interest to provide handwritten digit segmentation. The main issues related to symbol segmentation are the connection or the overlapping of the symbols (Fig. 1).

The remainder of this paper is divided as follows: next Section briefly reviews some important algorithms for symbol segmentation. Section III presents the proposed algorithm. Section IV presents some results and experiments while Section V concludes the paper.

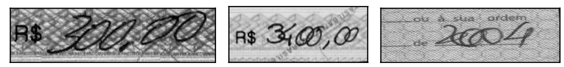

Figure 1. Sample images of overlapped or connected digits.

II. SEGMENTATION TECHNIQUES

There are several classical algorithms for segmentation. These algorithms are developed for connected symbols in general. Amongst them, we can cite *drop-fall*, *hit-and-deflect*, *min-max* or *structure feature based algorithms*. The first three of these techniques deal with the traversal of the border of the symbols. In drop-fall [2][4], the segmentation is done using the idea of a ball rolling as it follows the contour of the characters. It works as if one drops a ball from the top of a mountain (the symbol) and it falls down creating a path that separates the characters. There are several variations of the drop and fall algorithms according to the start point of the path. It can be descending or ascending from left or from right.

Hit and deflect methods [6] use a set of rules to separate the characters building a cutting path based again on the contour of the symbol. The algorithm starts from a peak in the lower contour and moves upwards, "reflecting" when the movement line "hits" a black pixel.

Another major type of segmentation heuristic involves the segmentation of a connected component based on structural features of the component itself (as corner points, local minima or maxima, etc). Min-Max algorithms are a sub-class of structure feature based algorithms. These techniques consider that most touching or overlapped characters are joined where the distance from the upper contour to the lower contour of the connected component is a minimum. The minima in the upper contour and the maxima in the lower contour define a straight line that separates the symbols.

A bad digitizing process can generate smeared characters which can be segmented by the method proposed in [5].

A method for separating connected digits is proposed in [1] using two agents. One agent works top-down and the other works bottom-up. Both agents look for cut-point candidates.

It is presented in [11] a general segmentation technique based on the evaluation of several candidates produced by an over-segmentation process. The method is efficient but it has a high computational cost (O(2^n)). To decrease this cost it

978-1-4244-2794-9/09/$25.00 ©2009 IEEE

proposes the use of heuristics which, by the other side, can also decrease the performance of the algorithm.

A survey about character segmentation methods can be found in [6].

III. Proposed Algorithm

The new algorithm works on bi-level images. Thus, the first step is the thresholding of the digits. The black-and-white images are seen as a path which is traversed by a hypothetical ball. The movement of the ball is defined based on physical forces and this movement causes the segmentation of the overlapped digits. The forces that we have implemented so far are inertia and centripetal. Next, we explain their use and the segmentation itself.

This is a complete version of the algorithm first proposed in [7]. In that paper, the authors presented an initial idea of the algorithm without the complete list of rules presented herein.

A. Inertial Movement of the Hypothetical Ball

As in any road, the object just needs the main path to roll over. It does not need the whole road. So, with the bi-level image, a thinning technique [10] is applied in order to reduce the digit to its skeleton. This skeleton is the path to be traversed by the ball; it contains the main aspect of the digits. A thinning algorithm makes a smoother skeleton than a skeletization algorithm. With the path defined, the ball needs a starting point (SP) to begin its movement. Before that, a salt-and-pepper noise removal is applied in order to eliminate unnecessary spots. This is also needed for other steps of the algorithm as it will be explained further. Fig. 2 presents an example of a pair of digits with overlapping and the skeleton after thinning.

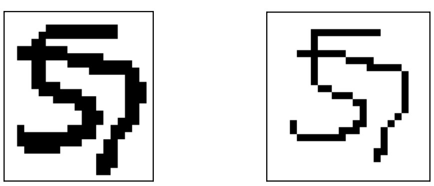

Figure 2. (left) Example of overlapped digits and (right) its skeleton.

The starting points from where the ball is launched can be:

1) points with just one neighbor or (Fig. 3-left) or;

2) points that have two neighbors that are adjacent to each other as in Fig. 3-top-center (otherwise they are not considered starting points as in Fig. 3-bottom-center) or;

3) the most left or the most right points case no one of the previous situations are satisfied (Fig. 3-right).

All images have a minimum of two SPs (Fig. 3-right); there is no maximum limit (Fig. 3-left, for example, with four SPs). Depending on the position of the SP, the ball will move to the left, to the right, to the bottom or to the top according to the rules defined next.

The ball starts its traversal from each staring point (one at a time). If the ball is in one position (SP or not) and there is just one black neighbor, it moves straight to this point which is its new position. The issues happen when there is more than one possible new position; when the ball's current position has more than one unvisited black neighbor and a choice must be done. In some of these cases, the inertial force defines the new position of the ball. According to Sir Isaac Newton in [8]: "The vis insita, or innate force of matter is a power of resisting, by which every body, as much as in it lies, endeavors to preserve in its present state, whether it be of rest, or of moving uniformly forward in a straight line.". This means that, in some situations, it is more probable that the ball continues to one way that changes to another. This situation can be seen in Fig. 4, where it is presented a zooming into part of the digits from Fig. 2-right (negated just for a better visualization). From a starting point, the ball continues its way down (Fig. 4-left) until it finds a crossing.

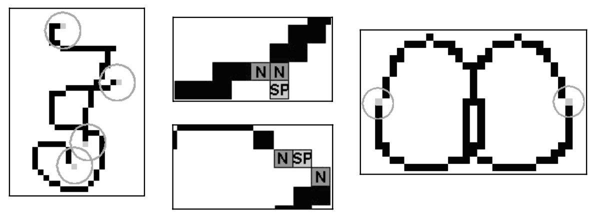

Figure 3. Starting points (SP): (left) SPs (circled) with just one neighbor; (top-center) a valid SP with two neighbors (labeled as N); (bottom-center) an invalid SP with two neighbors; (right) in this case, the SPs (circled) are the most left and the most right points of the figure.

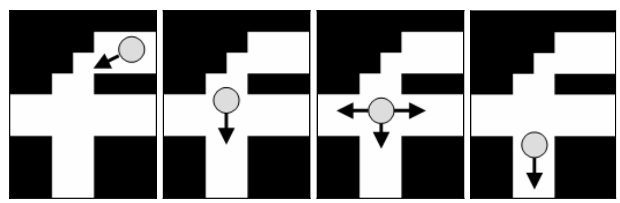

Figure 4. The inertial movement of the ball: (left) its traversal from a SP; (center-left) at this point, the ball reaches a crossing and (center-right) it has three possible ways. (right) The inertial force makes it continues its way down as it is more uniform considering from where it came from.

This situation can be summarized in Fig. 5. When the ball comes from one position, the next position is naturally a continuation from the path being created. So the path must be as smooth as possible. Consider Fig. 5: suppose in this figure that the ball is in the position (1, 1) – the center of the square. If it came from position (0, 0), it is more natural that the next step is point (2, 2). If it came from (1, 0), it continues to (1, 2), and so on.

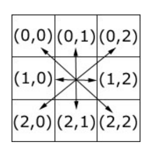

Figure 5. From the central position, the next point to be reached depends on where the ball came from.

When there is more than one possible position to follow, the first consideration is that the chosen path is the one that is closer to the ideal path, *i.e.*, the point that causes the smaller changes on the path. This can be seen in Fig. 6, where it is depicted a point with two possible paths, none of which is a continuation of the previous path (Pi). The same figure also presents the point chosen to continue the movement of the ball (Nb2). This is the choice that changes the path by an angle smaller than the angle created if we had chosen Nb1.

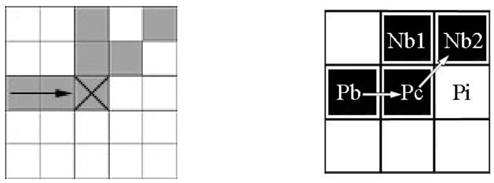

Figure 6. (left) Movement of the ball until the position marked with an 'X'; (right) from this position, the ball would naturally go to the position Pi, but it is not part of the object. Thus the ball must choose between positions Nb1 and Nb2. Nb2 is chosen as it creates a path closer to the best.

To increase the probability of the ball to get the correct path, a 3-length array with the history of the movement is stored. This history is updated at each new step and in the beginning of each new traversal, it is cleared. When its maximum size is reached, the older element is removed. Each element in the history has a corresponding weight which decreases as the ball goes far from this element. The length of the history was chosen to be 3 as higher values increase the inertia of the ball making possible changes in its path harder. This can be seen in Fig. 7. The ball comes from the path labeled 'B'. In this case, we are using a 5-length history array. From left to right, when the ball reaches the 'D' position, it continues through the incorrect path labeled as 'W'. As a 5-length history array indicates that the previous path was in the most part horizontal from left to right, the inertia makes the ball continue this movement instead of taking the correct path (labeled as 'C'). A 3-length history makes the ball rolls through this 'C' right path.

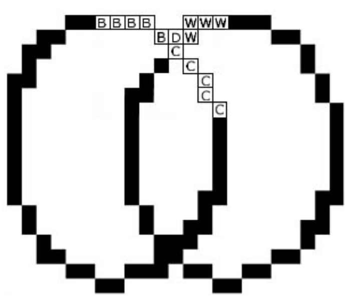

Figure 7. A history array of length equal to 5 makes the ball goes to the wrong way ('W') instead of the correct path ('C').

In the simplest situation, the movement follows the basic idea presented in Fig. 6: when there is just one black neighbor, the ball moves to that position; when there are two, it goes to the point that causes the minor change in the expected path.

When there is a situation where both neighbors have the same probability of being reached, the history is used to make a decision. This can be exemplified in Fig. 8. When the ball reaches the position marked with an 'X', it can rolls to any one of the other two positions with equal probability. In these cases, what will make a decision are its previous movements stored in the history and two rules heuristically defined based on the human hand movement to write digits:

1) If the ball has been moving in one vertical direction (upwards or downwards) and it finds a position where it can continue its movement or change its direction (from upwards to downwards and vice-versa) it chooses the way at which it continues its previous movement.

2) If the movement is horizontal, if the ball has been coming from the right and finds a point where it can change its direction to the left or continue to the right, it chooses to change and go to the left. Similarly, if it comes from the left, it changes and goes to the right.

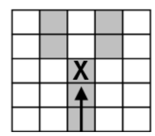

Figure 8. In an upwards movement, when the ball arrives at the position 'X', it can continue its movement with equal probability to the left or to the right. The history of the movement helps the decision.

When the movement ends (no possible neighbor not visited), the process stops and goes to another starting point not used yet if there is any.

B. The Use of Centripetal Force

Another situation comes when the leftmost or rightmost point of the image is chosen as a SP. It is assumed then that these points belong to a '0' or a '8' digit. At these situations a centripetal force is taken into account to calculate the next position in the cases where a choice must be made. Such a SP can generate two different paths: one clockwise and another counter-clockwise. The direction that the ball takes can generate different paths. Fig. 9 illustrates this.

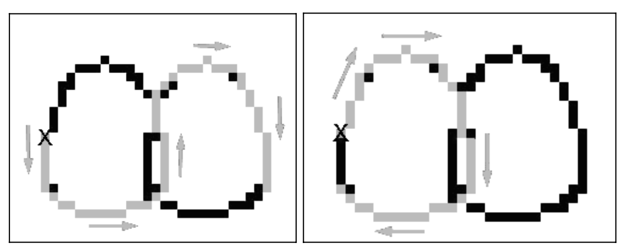

Figure 9. The 'X' indicates the SP. The clear gray path indicates the path which is followed (left) counter-clockwise or (right) clockwise.

To solve this problem, the next position is evaluated in a different way, making use of a *centripetal force*. This is an external force needed to keep an object following a curve path. At every movement, if the values of vertical and/or horizontal displacements change, they are stored. These displacements are used to calculate the next position to be visited when the ideal next point is not a black neighbor.

The calculation to find the nearest neighbor in this case takes into consideration the following factors:

- Euclidian distance between candidate neighbors and the next position;
- components of the movement (horizontal and vertical) stored in history list.

The difference is that the distance is not between neighbors and the next ideal position, based on values stored in history list, but between neighbors and the calculated next position, based on previous movement displacement. The next position is then evaluated using the current position as follow:

$$next\_position_x = current\_position_x + delta_x \quad (1)$$
$$next\_position_y = current\_position_x + delta_x \quad (2)$$

where $delta_x$ and $delta_y$ are evaluated as follows:

IF $current\_position_X - previous\_position_X \neq 0$ THEN
$delta_X = current\_position_X - previous\_position_X$
IF $current\_position_y - previous\_position_y \neq 0$ THEN
$delta_y = current\_position_y - previous\_position_y$

At every movement, new displacements ($delta_x$ and $delta_y$) values are calculated. For horizontal and vertical movements, the last non zero delta value is stored. If the calculated point is a neighbor, it is chosen as the next point otherwise, the nearest neighbor is chosen. If two neighbors have the same distance to the calculated next point calculated, the values in history list are used as before. This simple approach has no complications with sudden changes of direction because they are rare.

Fig. 10 presents how the next position is calculated. Fig. 10-top shows the pair to be segmented. Fig. 10-bottom-left shows the path followed without taking displacements into account. The path presented in Fig. 10-bottom-right uses displacements in calculations and lead to a correct path segmenting the digits into two zeros. The white arrows in Fig. 10-top shows the path being traversed and the square, the region zoomed in Fig. 10-bottom-left and Fig. 10-bottom-right. In these figures, the traversed path is marked in light gray.

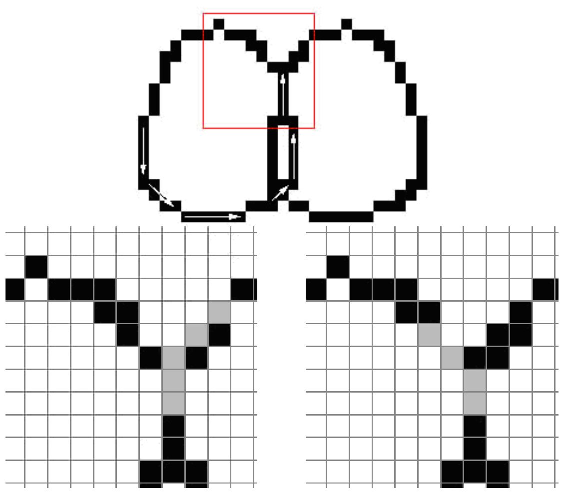

Figure 10. (top) Pair being traversed (the white arrows indicate the movement); the square marks the zoom area presented in: (bottom-left) without taking displacements into account and in (bottom-right) with the use of displacements. The movement appears in light gray.

Fig. 11 presents the complete example of the segmentation achieved with and without the use of the centripetal force for the image of Fig. 10-top. At first, using just the inertia, it is not possible to segment the zeros by any way from any starting point (the possible traversals are presented in light gray). The use of the centripetal force allows the correct segmentation of the pair of zeros.

Figure 11. (top) Wrong traversals (in light gray) from the possible starting points (labeled with an 'X') to segment the pair of zeros using just the inertial force; (bottom) correct digit segmentation with the use of centripetal force.

C. *Reconstruction*

The final step deals with remaining points of the digits or elements with few amounts of black pixels. As we removed all the noise in the beginning of the process, no isolated point is expected to be found. So, after segmentation, any isolated point is attached to the object that it was connected in the beginning of the process.

Besides single spots, we deal with small parts of numbers (as a small line that can be part of a '7' or a '1' as in Fig. 12-left). Objects are considered as isolated strokes (this means that they are part of a number) if:

1) Its height or width is less than 50% of the image's height or width;

2) It touches another object;

3) No part of it is used in a complete element.

If an element satisfies all these three conditions, it is considered an isolated element and it is attached to the object it touches.

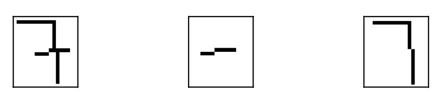

Figure 12. (left) A digit with a small line in its middle; (center) and (right) the results of the segmentation process: the isolated element (center) has the features needed to be attached to another object (right), restoring the complete number (left).

IV. RESULTS AND DISCUSSION

For the experiments, a database of overlapped digits was created from isolated digits extracted from true type cursive fonts and from digits available at: http://yann.lecun.com/exdb/mnist/. The overlapping was synthesized using these isolated numbers. Examples of the overlapping can be seen in Fig. 13 jointly with their segmentation by our proposal.

Fig. 14 presents a sample image and the segmentation achieved by the algorithms of Alhajj [1], Renaudin [11] and our proposal. It can be seen that the other algorithms always leave some part of one digit on the other.

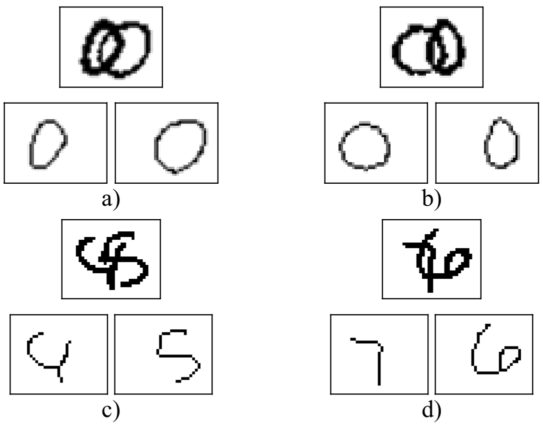

Figure 13. Samples of overlapping digits used and their segmentation by our new algorithm.

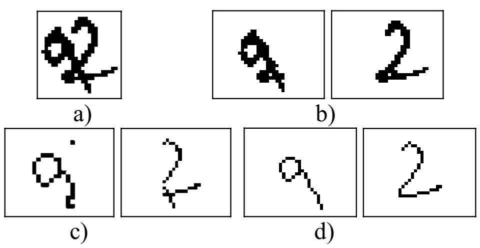

Figure 14. a) Original image and its segmentation by: b) Renaudim, c) Alhajj and d) our proposal.

One of the most important aspects of our approach is that it has a computational cost of $O(n^2)$ which is much better than most part of graph-based segmentation techniques which have cost of $O(2^n)$. This cost is mostly because of the thinning phase. Although the idea of the traversal of the hypothetical ball does not require this thinning, it is used to decrease the amount of pixels needed to be analyzed at each step of the movement.

We have tested the algorithm in a set of 50 overlapped synthetic overlapping digits with satisfactory results. These results also depend on an automatic digit recognition technique which is going to be part of the complete system and it is still under development [14].

V. CONCLUSIONS

In this paper, we presented an algorithm for segmentation of overlapping cursive digits based on physical forces as inertia and centripetal force. Segmentation is an important issue in automatic character recognition as it isolates the symbols that are going to be recognized by the system.

The proposed algorithm works with the idea of a hypothetical ball that rolls over the digits with a movement defined by physical forces. The inertia defines the general movement of the ball. Thus, if it comes from one way and reaches a crossing, the ball tends to continue its way according to the direction that it came from. For rounded digits as zeros, centripetal forces act in the ball keeping its circular movement.

Several rules define the way the ball moves so that the digits are correctly segmented. There are some specific cases where the algorithm does not provide a correct answer, but its low computational cost ($O(n^2)$) in comparison with $O(2^n)$ graph-based algorithms encourages its use.

Our algorithm fails in situations where the wrong path has higher probability than the correct one and when the thinning generates just one stroke for more than one digit (Fig. 15).

Figure 15. a) Original image and its incorrect segmentation by our proposal.

ACKNOWLEDGMENTS

This research is partially sponsored by CNPq (grant 470541/2008-3). The authors would also like to thanks Dr. Christophe Renaudin for his valuable help.

REFERENCES

[1] R.Alhajj and A.Elnagar, "A Novel Approach to Separate Handwritten Connected Digits", Int.Conf. on Document Analysis and Recognition, Scotland, 2003, pp. 1198-1202.

[2] R.G.Casey, and E.Lecolinet, "A Survey of Methods and Strategies in Character Recognition", IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 18, no 7, pp. 690-706, 1995.

[3] C.Chou, S.Chu and F.Chang, "Estimation of skew angles for scanned documents based on piecewise covering by parallelograms", Pattern Recognition, no 2, vol. 40, pp. 443-455, 2007.

[4] G.Congedo et al, "Segmentation of Numeric Strings", Int.Conf. on Document Analysis and Recognition, Canada, 1995, pp. 1038-1041.

[5] Jia Tse, et al, "An OCR-Independent Character Segmentation Using Shortest-Path in Grayscale Document Images", Int. Conf. on Machine Learning and Applications, USA, 2007, pp. 142-147.

[6] Y.Lu and M.Shridhar, "Character Segmentation in Handwritten Words - An Overview", Pattern Recognition, 1996, Vol 29, no. 1, pp. 77-96.

[7] C.A.B.Mello et al. "Segmentation of overlapping cursive handwritten digits", ACM Document Engineering, Brazil, 2008, pp.271-274.

[8] I.Newton, Mathematical Principles of Natural Philosophy, translated into English by Andrew Motte, First American Edition, New York, 1846.

[9] L.O'Gorman, R.Kasturi., Document Image Analysis, IEEE Press, 1996.

[10] J.R.Parker, Algorithms for Image Processing and Computer Vision, John Wiley and Sons, 1997.

[11] C.Renaudin, et al., "A General Method of Segmentation-Recognition Collaboration Applied to Pairs of Touching and Overlapping Symbols", Int.Conf. on Document Analysis and Recognition, Brazil, 2007, pp. 659-663.

[12] A.Sanchez, et al. "Text Line Segmentation in Images of Handwritten Historical Documents", IEEE International Workshops on Image Processing Theory, Tools and Applications, Sousse, 2008.

[13] M.Sezgin et al. "Survey over image thresholding techniques and quantitative performance evaluation", Journal of Electronic Imaging, 2004, vol. 13, no. 1, pp. 317-327.

[14] E.Silva JR, et al., "Feature Selection and Model Design through GA Applied to Handwritten Digit Recognition from Historical Document Images", Int.Conf. on Frontiers in Handwritten Recognition, Canada, 2008.