A New Segmentation Approach for Handwritten Digits

Luiz E. Soares de Oliveira, Edouard Lethelier, Flávio Bortolozzi
PUCPR - Pontifícia Universidade Católica do Paraná (LARDOC)
Rua Imaculada Conceição 1155, Prado Velho, 80215-901 - Curitiba - Pr Brazil
{soares,edouard,fborto}@ppgia.pucpr.br

Robert Sabourin
ETS - Ecole de Technologie Supérieure (LIVIA)
1100, rue Notre Dame Ouest, Montréal (Quebec) H3C 1K3, Canada
sabourin@gpa.etsmtl.ca

Abstract

This article deals with a new segmentation approach applied to unconstrained handwritten digits. The novelty of the proposed algorithm is based on the combination of two types of structural features in order to provide the best segmentation path between connected entities. This method was developed to be applied in a segmentation-based recognition system. In this article, we first present the features used to generate our basic segmentation points. Then, we define our segmentation paths depending on the encountered configurations with only few heuristic rules. Finally, we evaluate the output of our segmenter using a neural network trained with isolated digits.

1 Introduction

Off-Line handwritten numeral string recognition has been a topic of intensive research for recent years, due to its large number of potential applications. Moreover, the various peculiarities of the unconstrained handwriting stay as an open problem in handwriting recognition field.

The challenge of a segmentation technique lies in the decision of the best cut path to localize an entity to be recognized as a correct isolated character by the recognition system. The literature usually shows three different strategies to perform the segmentation : Segmentation-Recognition [6] Segmentation-based Recognition [3] [5] and Segmentation-Free systems [8].

In the first approach, the segmentation module provides a single sequence hypotheses where each sub-sequence should contain an isolated character, which is submitted to the recognizer. This technique shows rapidly its limits when the correct segmentation does not fit as well as the pre-defined rules of the segmenter. Very often, contextual information is used during the segmentation process in order to improve the robustness of the system.

The second strategy is based on a probabilistic assumption where the final decision must express the best segmentation-recognition score of the input image. Usually, the system yields a list of hypotheses from the segmentation module and each hypothesis is then evaluated by the recognition. Finally, the list is post-processed taking into account the contextual information. Although this approach gives better reliability than the previous one, the main drawback lies in the computational effort needed to compare all the hypotheses generated. Moreover, the recognition module has to discriminate various configurations such as fragments, isolated characters and connected characters.

In the third technique, the recognizer is trained with the peculiarities of the segmentation. For example, two connected characters of class “00” correspond to a specific class. Due to the large variability of the handwriting, we can easily imagine the problem of considering all the possible configurations to be learned by the system. But in particular applications where connections are restricted to a few class-set, this approach is certainly well suited.

Our works are based on the segmentation-based recognition strategy. The aim of this article is to show how we defined a new segmentation algorithm taking into account two complementary sets of structural features. The final objective of the module is to provide the best hypotheses list of segmentation paths without any a priori knowledge of the context, such as the number of characters to be segmented. Therefore we focused our segmentation work on the limitation of heuristic rules to consider most of configurations in connected characters. We worked with binary images with a 300 dpi resolution.

2 Generation of the Segmentation Features

Depending on the context, a lot of features are available to find plausible segmentation points in a character image. Certainly some of the most used in the literature are the contour and profile features [2] [6].

Indeed, these features are easy to be provided and they usually express directional variability of the character strokes and then possible cuts. However, they are not even fully informative to localize any kind of connection between characters, mostly when the handwriting is strongly skewed or overlapped. Therefore, we opted to consider a second set of features provided by the skeleton: the *intersection points*. These points are often located in the neighborhood of stroke connections where contour and profile features are not always available.

Let us define the contour as the image envelope. This is a bi-dimensional data where each contour point C_i is associated with the coordinates (X_i, Y_i) of the image. The profile image is obtained from a vertical projection of the first encountered transition, in both ways top-down and bottom-up. From these both sets of features, we are able to localize the first list of potential cuts which correspond to the local minima of the contour and profile (Figure 1a and 1b). We define these points as *Basic Points* (BPs).

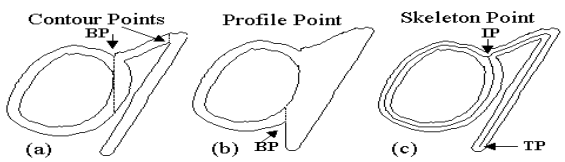

Figure 1. Segmentation points generated: (a) Contour (b) Profile (c) Skeleton Points (Intersections and Terminations)

Let us define now characteristics of the skeleton :

Definition 1: In a binary image, the intensity of a pixel p, denoted $I(p)$, is either 0 (white) or 1 (black). A pixel p is a *foreground pixel* iff $I(p) = 1$. Conversely, a pixel p is a *background pixel* iff $I(p) = 0$.

Definition 2: Using the Freeman directions, when the eight neighbors of a pixel p are traced clockwise, the neighborhood of p is denoted $T(p)$, such as:

$$T(p) = \sum_{i=1}^{8} I(p'_i) \tag{1}$$

where p'_i is the ith neighbor of p. We can then define the *terminal point* (TP) when $T(p) = 1$ and the *intersection point* (IP) when $T(p) > 2$. The *intersection point* and *terminal point* are called *characteristic points* of the skeleton.

Definition 3: A skeleton path $(P_{skeleton})$ is a pixel sequence of the skeleton where each extremity corresponds to a *characteristic point*.

The skeleton image is obtained by using the thinning algorithm proposed by Jang et al [1]. The best advantage of this algorithm relies on the limited number of IPs generated in the neighborhood of connected strokes of different characters. During the detection process of the IPs, two particular configurations are first selected as showed in Figure 2. We have noticed that particular digit connections, such as "00" for example, often generate these IPs classes. All other types of IPs are considered as a single IP class (Class 3). The first class (Figure 2a) contains all IPs with one segment in its lower part and two segments in its upper part. Depending on the Freeman directions, each class contains all the variations which respect this definition. The second class (Figure 2b) is the symmetric of Class 1.

Figure 2. Skeleton intersection points: (a) Class 1 (b) Class 2

3 Determination of Segmentation Paths

Once selected from the image, the BPs and the IPs are compared altogether in order to determine the list of segmentation hypotheses of the character image. The algorithm scans all possible relationships between BPs and IPs and generates a set of segmentation paths where the goal is to get the correct segmentation paths (for connected characters) in the list of hypotheses.

This association takes into account the distance between BPs and IPs. When an IP is located in the BP neighborhood, this could indicate a possible stroke connection between two characters. Then our algorithm uses only the relationship between IPs and BPs to provide the local cut path. To determine the proximity between points, we based our comparison on the estimation of the thickness of the strokes E_t, obtained with the projection of the density histogram. Then, two points BP_i and IP_j belong to the same neighborhood if :

$$d_E(BP_i, IP_j) \leq E_t \tag{2}$$

or

$$d_E(proj_y(BP_{ik}), IP_j) \leq E_t \quad \text{for} \quad k = 1, 2 \ldots n \tag{3}$$

is verified, where d_E is the Euclidian distance, $proj_y(BP_{ik})$ is the vertical projection of BP_i at the

step k on the segment whose height is n. Note that the equation 3 is checked only if equation 2 is not verified. The Figure 3 shows both configurations of neighborhood verification.

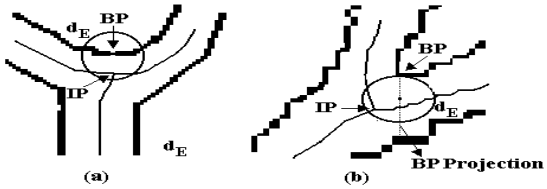

Figure 3. Distance verification between the BP and IP (a) Distance verified by equation 2 (b) Distance verified by equation 3

Depending on the local configurations of points, the segmentation path (P_{seg}) can be generated straight away from the $P_{skeleton}$ or being orthogonal to it. In the first case, the system has found particular points of class 1 and class 2, as described in Figure 2 and tries to link the sequence of $P_{skeleton}$ between these two points. The figure 4 shows an example where $P_{skeleton}$ (Figure 4b) is associated with the joining segments between BPs and IPs, denoted (P_{bp-ip}) (Figure 4c) to form the final and correct segmentation path (P_{seg}) (Figure 4d). Thus, we can define the segmentation path : $P_{seg} = P_{skeleton} \bigcup P_{bp-ip}$

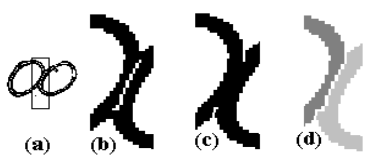

Figure 4. Particular conections: (a) Skeleton (b) $P_{skeleton}$ (c) P_{bp-ip} (d) P_{seg}

In the second case, where no such configuration is found, but where it exists some IPs and BPs, the segmentation path is orthogonal to $P_{skeleton}$. To determine the possible cuts around each IP, the algorithm performs the following tests :

1. If the considered segment of the skeleton is a stroke-end (TP exists on the segment), then the cut is allowed if the segment length is significant;

2. The cut length should be minimum in the neighborhood of IP. To find the best position, the cut is evaluated in $2r_e$ pixels, where r_e is the IP radius, for the minimum distance where a transition is encountered.

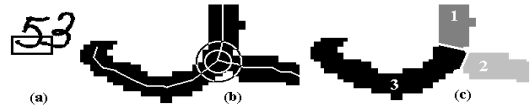

Figure 5. Orthogonal cuts: (a) Original image (b) Potencial region (c) Cuts realized

The Figure 5b shows the specific regions where the search of the cut path is performed around the IP. For each pixel of the $P_{skeleton}$ included in the biggest circle, the orthogonal cut path is evaluated and compared to the previous minimum length already calculated. Finally, the algorithm performs the cut path for each segment around the IP. When the configuration does not verify the first test, the cut path is not performed as shown in Figure 5c for the segment 3.

In some cases, none IP is detected in the image (see Figure 6), even if it exists a connection between two characters. To solve the lack of redundancy of BPs and IPs, the algorithm enables cut path directly from BPs in order to avoid segmentation errors.

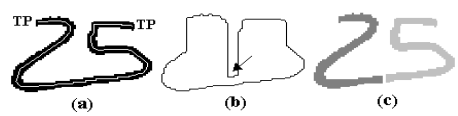

Figure 6. Lack of IPs: (a) Original image with the TPs only (b) BP from profile (c) Segmented image

4 Evaluation of the Segmentation

To be able to evaluate our segmentation algorithm with the best objectivity, we used a neural network (Multi-Layer Perceptron) trained with a set of 8.500 images (10 classes) of isolated handwritten digits extracted from our laboratory database. We used a standard backpropagation algorithm and the features extracted from the images are the concavity measures described in [4]. The best learning experiment provided a rate of 99.9% in training and 98.5% in test.

The evaluation of the segmentation was runned on a set of 2.000 images of brazilian bank checks (laboratory database), where we extracted a subset of 900 images of connected characters (pairs and triples for the most). For an input image, the best segmentation-recognition choice is given by the highest product of recognition scores of each sequence image.

In a first evaluation, we did a visual analysis and we verified that 98.5% of connected characters were correctly segmented. For most cases, the pure segmentation errors are due to the lack of BPs in the neighborhood of the connection strokes: under-segmentation (see Figure 7c). In a second experiment, the performance of our Segmentation-based Recognition system was evaluated and 90.8% of good classifications has been observed.

The analysis of the confusions shows two main categories of errors : segmentation errors (1.5%) and classification errors (7.7%). We define a segmentation error as the best score configuration where one (or more) character is mis-segmented and provokes a mis-classification (Figure 7a). Classification error is the best score configuration where the segmentation is correct but one (or more) character is mis-classified (Figure 7b).

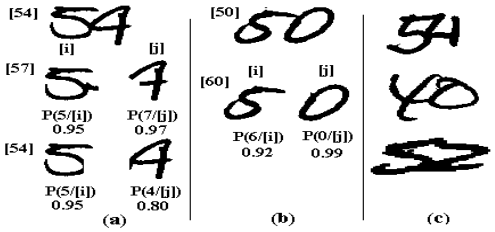

Figure 7. Cases of errors (a) Segmentation error (b) Recognition error (c) Cases not solved

In both cases, errors can be reduced if the recognizer is not only trained with naturally isolated digits, but also with images obtained from the output of the segmentation module. Moreover, an evaluation of various feature sets is necessary to avoid the mis-classification errors. Considering our new segmentation approach, we have developed a simple technique with few rules which enables us to perform the correct segmentation in most cases of connected characters, even if the symbols are strongly overlapped or skewed.

To obtain the same results, current structural approaches need to use a lot of heuristics to consider the large variability of handwriting [3]. Moreover, any particular pre-processing technique is necessary as we find in other approaches such as windowing [5] or histogram projection [7]. If we compare with algorithms only based on contour and profile features [2], our method is more accurate in case of strong connections. Figure 8 shows examples of particular configurations where segmentation is not obvious with classical methods.

Figure 8. Correct segmentation in complex cases

5 Conclusion and Perspectives

We have presented a new segmentation-based recognition approach applied to unconstrained handwritten digits. The segmentation technique uses two sets of structural features to provide the possible segmentation paths, with any contextual information of the input image. Due to the small number of rules used in the cut path generation, the system is able to provide the correct segmentation in most cases of connected characters. The first results show that we can easily improve the recognition accuracy by considering the segmentation outputs in the training step of the recognizer.

The next studies we plan to implement in this system deal with the combination of segmentation information in the recognition module in order to perform a real segmentation-based recognition system. Moreover, we will focus on a new recognition system where various types of features will be used to improve the rejection rate.

References

[1] B.K.Jang and R.T.Chim. Reconstructable parallel thinning. In C.Y.Suen and P. Wang, editors, *Thinning Methodologies for pattern recognition*, pages 181–217. World Scientific, 1994.

[2] E.Lethelier, M.Leroux, and M.Gilloux. An automatic reading system for handwritten numeral amounts on french checks. In 3^0 *ICDAR*, pages 92–97, 1995.

[3] H.Nishida and S.Mori. A model-based split-and-merge method for character string recognition. In P. Wang, editor, *Document Image Analysis*, pages 209–226. World Scientific, 1994.

[4] L.Heutte, J.Moreau, B.Plessis, J. Plagmaud, and Y.Lecourtier. Handwritten numeral recognition based on multiple feature extractors. In 2^0 *ICDAR*, pages 167–170, 1993.

[5] O. Matan and C.J.C.Burges. Recognizing overlapping hand-printed characters by centered-objects integrated segmentation and recognition. In *IJCNN*, pages 504–511, 1991.

[6] M.Shridhar and A.Badreldin. Recognition of isolated and simply connected handwritten numerals. *Pattern Recognition*, 19(1):1–12, 1986.

[7] R.Fenrich. Segmentation of automatically located handwritten words. In 2nd *International Workshop on Frontiers in Handwriting Recognition*, pages 33–44, 1991.

[8] S.Choi and I. Oh. A segmentation-free recognition of two touching numerals using neural networks. In 5^0 *ICDAR*, pages 253–256, 1999.