What Knowledge about Handwritten Letters can be Used to Recover their Drawing Order ?

Laëtitia Rousseau, Jean Camillerapp, Eric Anquetil

What Knowledge about Handwritten Letters can be Used to Recover their Drawing Order ?

Laetitia Rousseau, Jean Camillerapp, Eric Anquetil
IRISA / INSA - Campus Universitaire de Beaulieu
Avenue du General Leclerc - 35042 RENNES Cedex - France
laetitia.rousseau@irisa.fr, jean.camillerapp@irisa.fr, eric.anquetil@irisa.fr

Abstract

A way to do off-line handwriting recognition is to generate an equivalent on-line signal of a letter image and to use an on-line recognition system to recognize this letter. To do this, the drawing order has to be recovered using handwriting knowledge. Our approach to recover the drawing order consists in proposing several starting and ending points. Several paths are generated and the best one is chosen. This paper presents the handwriting knowledge introduced at several steps of our method and its contribution for the recognition process. Experimentations have been carried out on isolated lower case multi-stroke letters from a database including both on-line and off-line signals. An on-line system has been used to recognize these letters.

Keywords: handwritten letters recognition, recovery of the drawing order, on-line and off-line recognition

1. Introduction

On-line handwriting recognition has better results than the off-line one: the off-line signal has lost the temporal information. A way to perform off-line handwriting recognition is to recover the drawing order using handwriting knowledge and to use an on-line recognition system to recognize letters. Our work is on Western handwriting. In general the used criteria are the curvature [3] [5] [7] and the left to right writing order. Doermann and Rosenfeld [2] introduce the direction of loops. Very little *a priori* handwriting knowledge is used. Other criteria could be added to improve results. Our approach to recover the drawing order consists in proposing several starting and ending points. Several paths are generated and the best is chosen. This paper presents the handwriting knowledge introduced at several steps of our method and its contribution for the recognition process. A database of handwritten letters including both on-line and off-line signals is used to learn handwriting characteristics and to assess our method.

The next section presents the problematic to recover the drawing order, section three explains tests methodology, section four shows how to use handwriting knowledge at each step of our method and section five presents experiments on handwritten letters.

2. Recovery of the drawing order in letters

In most methods to recover the drawing order, a drawing extractor applies a thinning algorithm to extract a skeleton in the ribbon portions of the drawing. A graph is then constructed, where edges correspond to ribbon drawings and nodes correspond to intersections and endpoints. Then the best sequence of edges is looked for. The Kato and Yasuhara's [4] algorithm consists in finding the good path knowing starting and ending points in mono-stroke drawings in general, but its application is limited. In [8], we have extended this algorithm to mono-stroke handwriting and several paths are proposed. In this paper, multi-stroke letters are also considered and more handwriting knowledge is used. The reconstruction algorithm needs a mono-stroke drawing and a localisation of starting and ending points. That is why segmentation has to be done (if necessary) and starting and ending points have to be proposed before applying this algorithm. Figure 1 presents the different steps of our method. Segmentation

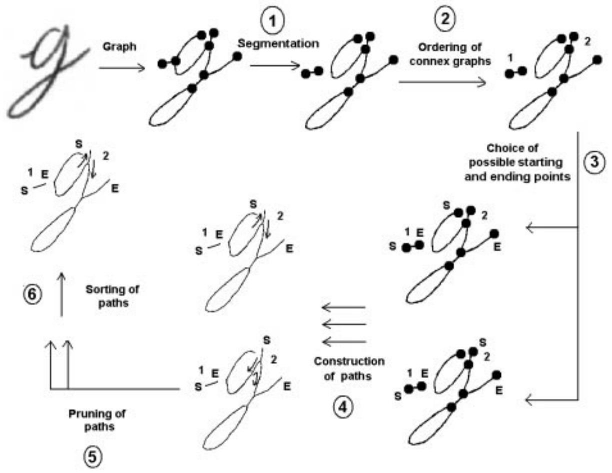

Figure 1. Steps to generate the drawing order

can be done (step 1) and in this case the different parts are ordered (step 2). Several starting and ending points are proposed (step 3). For each couple of points, the reconstruction algorithm is applied (step 4) and several paths between them are generated. Impossible paths are pruned (step 5) and the best path is chosen sorting remaining paths (step 6). Handwriting knowledge can be introduced at each step of this method. The minimisation of the curvature is the criterion the most used to choose the best path. Edges are linked if they globally minimize the average curvature [3] [5] [7]. But curvature is not always sufficient to choose the best path (see figure 2). About 75% of the letters can be reconstructed with only this criterion. When the thickness of the drawing is high, the curvature is less accurate because there are some distortions at the tips of ribbon portions. So other criteria have to be used and are presented in this paper.

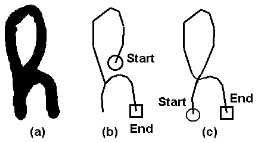

Figure 2. Curvature not sufficient (a) Image (b) Good path (c) Path with minimum curvature

3. Test methodology

To explain how to assess the contribution of handwriting knowledge, test methodology is presented in this section. The drawing extractor used is described in [6]. The on-line recognition system used is RESIFCar [1]. They have been designed by IMADOC project-team (www.irisa.fr/imadoc), from IRISA laboratory. Our method has been validated on lower case letters from the IRONOFF database [10] (10,000 letters) that includes both on-line and off-line signals for each letter. For each alphabetic letter, there is the same number of samples. We have removed unexpected models (10% of the database) like upper case letters and images with noise (8% of the database) like stains.

We aim at recognizing letters even if the generated signal is not the same as the original on-line signal. But to learn handwriting knowledge and to assess its contribution, the path corresponding to the on-line signal is looked for.

Handwriting knowledge depends on the models of letters learned by the on-line system. Models have been firstly learned on another on-line database. Models corresponding to possible paths but not used in on-line handwriting can be added. But if a consequent number of models are added, the confusion between letters increases. Besides, a sufficient number of examples is required to create a model. Consequently only stable models can be added.

Starting or ending points are sometimes hidden in the drawing, i.e. there is no node containing them (see figure 3). So the path corresponding to the on-line signal cannot be found and knowledge cannot be learned. These cases represent 11.1% of the database. It often occurs in letters a, b, d, g, m, n, p, q, r. To solve this problem, models have been secondly learned on reconstructed signals from IRONOFF database including new models. These new models have to respect handwriting knowledge described in this paper (minimum curvature...). Figure 4

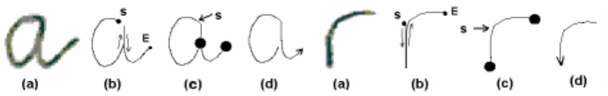

Figure 3. Hidden starting or ending points (a) Image (b) On-line signal (c) Graph (d) New model

contains typical models learned, including new models. A letter can be recognized if it corresponds to a model. It can be written in more or less strokes, it is recognized all the same if it respects downstrokes. The on-line system is based on downstrokes, so it is very important to well reconstruct them.

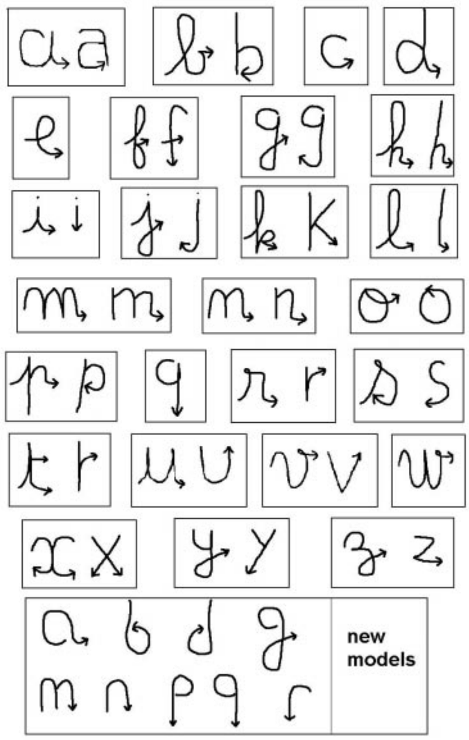

Figure 4. Models of letters learned

3/4 of the database (5,556 letters) is used for the learning (models of letters and handwriting knowledge) and 1/4 (1,852 letters) for tests. The contribution of each piece of handwriting knowledge is calculated on these test data.

4. Using handwriting knowledge

This section presents the handwriting knowledge used at each step of our method. Knowledge about segmentation (step 1), ordering (step 2), choice of starting and ending points (step 3) is presented. When a criterion can predict that a path is wrong, this path can be eliminated (step 5). When a criterion needs all the paths to determine which is the best, the decision is made when all the possible paths are generated (step 6).

4.1. How to segment

A multi-stroke letter can appear as only one connex graph if the strokes touch each other. So segmentation is done if necessary at step 1. A letter has to be segmented when a lack of segmentation would lead to a non-existing model. Four types of structures exist. An horizontal bar is detected as in figure 5 (c) at a node of degree four without cycle if two edges form an horizontal and straight stroke. Two oblique lines are detected as in figure 5 (a) at a node of degree four without cycle if the edges form two oblique straight lines. Two curved lines as in figure 5 (b) are detected at two nodes of degree three without cycle if the edges form two vertical curves. For these three cases, strokes are separated from each other, duplicating the common parts. A link is detected as in figure 5 (d) at a node of degree three if the edge on the left is straight and little and the other edges are curved. This edge is separated from the others.

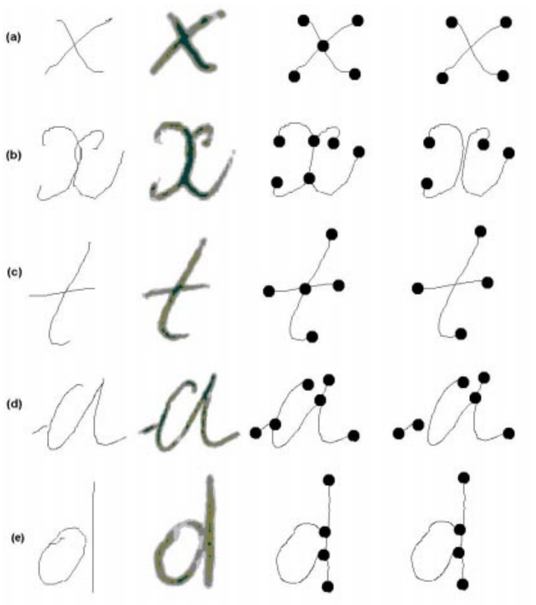

Figure 5. Examples of segmentation : on-line signal, image, graph and result of segmentation.

These cases represent about 9.2% of the test data. Our method makes only one proposition of segmentation. The same segmentation as the on-line signal is obtained in 93.8% of the test data. 0.7% are too segmented, 5% are not enough segmented and 0.5% are segmented differently. But we do not aim at having the same signal as the on-line one. The letters not well segmented have a recognition rate of 94.4%, so errors at this step do not affect a lot recognition rate. It is due to letters that can be segmented or not like the "d" from figure 5 (e). If this step was not done, good recognition rate would be 77.6% instead of 81%.

4.2. How to order the connex graphs

If there are several connex graphs, the order between them is determined at step 2. If the reason for segmentation is an horizontal bar, it is traced at the end. If it is a link, it is traced first. If the graphs do not overlap each other with the y projection, they are ordered from bottom to up ("i" case). Else graphs are ordered from left to right. The same order as the on-line signal is obtained in 99.6% of the cases.

4.3. How to choose possible starting and ending points

If there are several connex graphs, the same method is applied on each of them. Starting and ending points are proposed at step 3. The presence of cycles can sometimes help to localise starting or ending points. The criteria presented in this section can be used at step 3 because the global drawing is not necessary. Good starting and ending points are proposed in 98.4% of the cases (good localisation and good direction). 60.6% of the letters where the good starting and ending points are not proposed are recognized.

4.3.1. Knowledge about starting and ending points in circles

There are two types of particular cycles in graphs. We call *a loop* a stroke that crosses itself as the "l" from figure 6. We call *a circle* a loop with a double-traced edge as the "d" from figure 6.

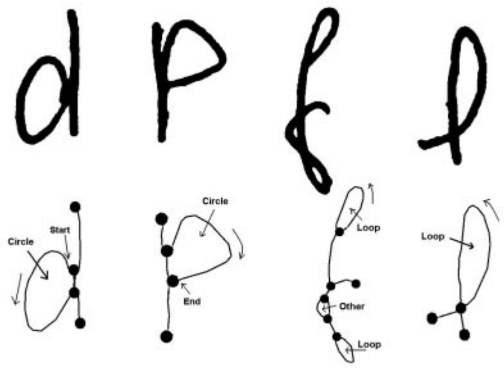

Figure 6. Labelling and crossing cycles

If a cycle contains two nodes and the longest edge is situated on the left or on the right and it is big (more than half of the letters width), then it is a circle. If a vertical bar is detected on the right, the starting point is localised on the top of the circle, like the "d" from figure 6. If a vertical bar is detected on the left, the ending point is localised at the bottom of the circle, like the "p" from figure 6. In cases of closed loops (like in an "o"), the graph contains only nodes of degree two. The starting and ending points are chosen on the upper-right node and it is drawn in counter-clockwise. Good starting or ending points are chosen with this criterion in 1.5% of the letters. It is few, but bad starting or ending points are never proposed and average number of paths generated after step 3 is 5.2 instead of 5.4 if this criterion is not used. So 3.7% of paths are not generated.

4.3.2. Knowledge about the position of starting and ending points

If starting or ending points have not been chosen with the knowledge about cycles, several couples of points are proposed. Starting and ending points correspond to nodes of degree one if they do not touch the drawing. They are on nodes of degree three or more if they are confused with a part of the drawing. In these cases, they belong to a cycle.

Several couples of starting and ending points are proposed and the direction has to be determinated (which point is the starting and which point is the ending). Figure 7 contains the direction of the vector that links the starting point to the ending point. It is always from left to right or from top to bottom. Limit angles have been learned. The good direction is chosen in 98.4% of the cases. Letters with a bad direction are recognized in 60.6%.

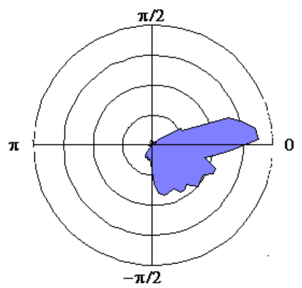

Figure 7. Direction of the drawing.

4.4. How to prune paths

For each couple of points, zero (if edges are not crossed), one or several paths have been generated at step 4. In average, 5.1 paths still remain. At step 5, paths incompatible with handwriting knowledge are eliminated. Pruning paths with reliable criteria has better results than only choosing the best path combining less reliable criteria. Knowledge presented in this section decreases the number of paths of 18%. Good pruning is done in 98.7% of letters. 70.2% of the letters where the good path has been pruned are recognized. Contributions of this step are presented in section 5.

4.4.1. Knowledge about double-traced edges

Kato and Yasuhara [4] have labelled double-traced edges (called D-lines) : LD-lines (Looped D-lines), SD-lines (Spurious D-lines) and PD-lines (Proper D-lines) (see figure 8).

A double-traced edge, in general, is not very curved and is short. A maximum curvature and a maximum length have been learned for each type of D-line to eliminate impossible paths. 8.3% of the paths are eliminated with these criteria, 6.4% with the criterion about only the

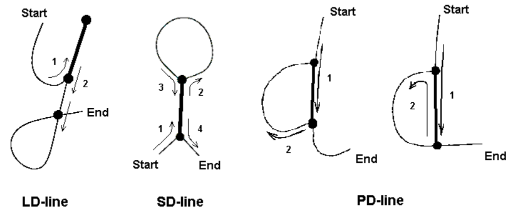

Figure 8. The three types of D-lines (in bold)

LD-lines.

4.4.2. Knowledge about the way to cross loops

A loop is constituted with one edge and one node of degree three or four. The way to cross these both types of loops (clockwise or counter-clockwise) has been studied. When there is a node of degree three (see figure 9 (a)) and it is oriented to the right, it is traversed clockwise. If it is oriented to the left, it is traversed counter-clockwise. When there is a node of degree four (see figure 9 (b)), only the top direction is reliable and it is crossed clockwise. Limit angles a1-a6 have been learned. This criterion allows pruning in average 13% of the paths.

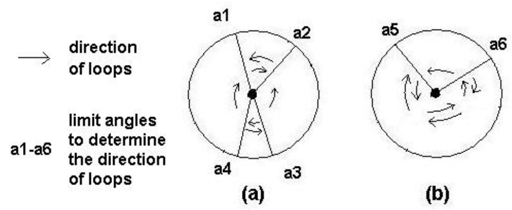

Figure 9. Crossing of loops with a node of degree three (a) or four (b).

4.5. How to sort paths

At step 6, the best path has to be determined. In average, 4.2 paths remain. The reliability of the criteria presented in this section is reported in section 5. No criterion is totally reliable and each criterion brings complementary knowledge. To sort paths, criteria have to be combined. It is a problem of classifiers combination with two classes (“good path” or ”bad path”). These classifiers are of measure type. Criteria values have been normalized. A linear method has been used to combine them. The good path is chosen in 95.4% of the cases. 36% of the letters in error are recognized.

4.5.1. Knowledge about double-traced edges

A LD-line is not very curved, is short, is often vertical and is situated in the middle of the letter. A RBFNN classifier (Radial Basis Function Neural Network) was used to determine if the position of LD-lines in a path is good or not. This knowledge about SD-lines and PD-lines is not reliable enough to be learned. Used alone, this criterion allows choosing the good path in 90.2% of the cases.

4.5.2. Knowledge about writing directions

The most frequent directions in handwriting are the up and the down ones, as said in [9]. Figure 10 represents the sum of the length covered for each direction for learning data (with a step of PI/32).

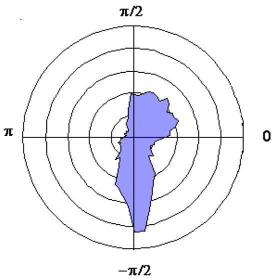

Figure 10. Most frequent directions

We can notice that the bottom direction is the most frequent. We can assume that if a path contains a lot of rare directions, it is a bad path. Thus an indicator of "ease of writing" is calculated. The length covered in one direction is multiplied by a coefficient depending on this direction. The path with the maximum value must be the best one. Used alone, this criterion allows choosing the good path in 77.5% of the cases.

4.5.3. Knowledge about downstrokes

Downstrokes are the fundamental structures of letters. Our on-line recognition system is based on them because they are relatively constant. In general, a letter has one or several downstrokes that are as high as the letter. For each path, the height of the longest downstroke is calculated. The path with the maximum value is probably the best one, because its principal downstroke is not broken. Used alone this criterion allows choosing the good path in 68.5% of the cases.

4.5.4. Knowledge about curvature

For each junction of edges, the angle difference is calculated. Not to penalize LD-lines, the junctions corresponding to a node of degree one are not concerned. The value of the global curvature corresponds to the average curvature of the other junctions. Used alone this criterion allows choosing the good path in 80.6% of the cases.

4.5.5. Knowledge about the position and the direction of starting and ending points

The position and the direction of starting and ending points can help to predict if a point is a starting, an ending or none. Figure 11 shows that the starting point is often on the left or on the top of the englobing shape and the ending point is often on the right or at the bottom. But there are no empty zones, so this criterion cannot eliminate a path.

Figure 12 shows that the direction of the starting is not reliable enough to be learned but the direction of the ending is relatively stable.

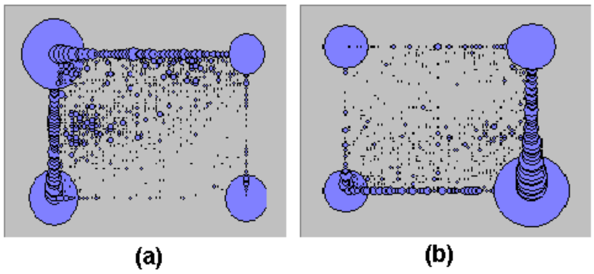

Figure 11. Position of (a) starting points and (b) ending points in the englobing shape. The diameter of the circles is proportional to the number of examples.

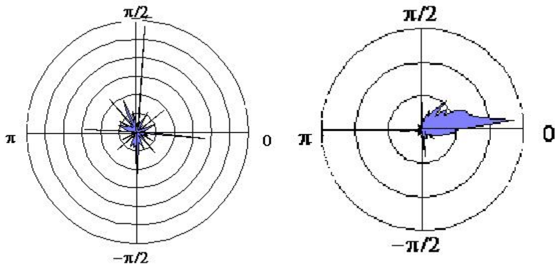

Figure 12. Directions of (a) starting and (b) ending

A RBFNN classifier has learned the position of starting points. Another RBFNN classifier has learned the position and the direction of ending points. Used alone, these criteria allow choosing the good path in 24.8% and 8.7% of the cases. These results are not good because a lot of paths have the same starting or ending points. But combined with other criteria, they help to choose the best path.

5. Experiments on multi-stroke letters

5.1. Tests about the good reconstruction

The path corresponding to the on-line signal and the path obtained with our method have been compared on 1,852 letter images. Table 1 contains, for each step, the rate of success, the average number of paths generated and the recognition rate of letters that contain an error resulting of this step. At each step, tested images were success-

Table 1. Success rate for each step

step	success rate	number of paths	recognition rate of letters in error
1-segmentation	93.8%	-	94.4%
2-ordering	99.6%	-	71.6%
3-starting/ending	98.4%	5.2	60.6%
4-reconstruction	97%	5.1	45.7%
5-pruning	98.7%	4.2	70.2%
6-sorting	95.4%	1	36%
global	**83.3%**	**1**	**59%**

ful at previous step. For example, in images where step 4 was successful, 98% of them are successful at step 5. Errors at steps 1 do not affect a lot recognition rates. Errors at steps 4 and 6 affect a lot recognition rates. Step 5 prunes 18% of the paths. It is important to notice that if this step is omitted, step 6 has 86.9% of good paths instead of 94.2% (corresponding to step5+step6). Finally, 83.3% of letters are well reconstructed.

Table 2 shows the rate of paths eliminated at step 5. Rules about cycles prune the maximum number of paths. Some paths are pruned by both criteria, that is why only 18% of paths are pruned globally.

Table 2. Contribution of each pruning criterion

Cycles	D-lines
13%	8.3%

Table 3 contains good reconstruction rate if each criterion of step 6 is used alone. Two paths can have the same value for a criterion. In these cases, this criterion cannot determine the best path. Column “with equality” contains good reconstruction rates when the “best” path is proposed in the first position and the column “without equality” contains good reconstruction rates when the “best” path is the only one proposed in the first position.

Table 3. Good reconstruction rate for each criterion

Criteria	with equality	without equality
D-lines	92.6%	90.2%
Curvature	88.8%	80.6%
Downstrokes	78.6%	68.5%
Direction	81.8%	77.5%
Starting	78.7%	24.8%
Ending	59.5%	8.7%
Combination	**93.3%**	**93.3%**

5.2. Tests about the good recognition

We aim at recognizing letters even if the generated signal is not the same as the original on-line signal. Table 4 contains the on-line recognition system rate for on-line and off-line signals. Firstly the on-line system has been used with models from another on-line database (models 1). Secondly, it has been used with well reconstructed signals of the learned data and new models presented at section 3 (models 2). 3/4 of the letters with starting and ending points hidden has been used for models and 1/4 has been tested. “S/E visible” correspond to letters with starting and ending points visible (1,852), “S/E hidden” correspond to letters with starting and ending points hidden (287), “Global” correspond to all these letters. With the second version, for letters with hidden starting or ending points, results are 83.6% instead of 9.4%. Besides, rates for other letters are a little greater.

Table 4. On-line recognition system rate

	On-Line	Off-Line
models 1, S/E visible	87.5%	80.8%
models 1, S/E hidden	88.5%	9.4%
models 1, global	**88%**	**75%**
models 2, S/E visible	85.6%	81%
models 2, S/E hidden	87.1%	83.6%
models 2, global	**85.7%**	**81.3%**

6. Conclusion

In this paper, contribution of handwriting knowledge to recover the drawing order in multi-stroke letters is presented. Our method consists in segmenting the graph if necessary and ordering connex graphs. Then several starting and ending points are proposed. Several paths are generated, impossible paths are pruned and the best is chosen. Handwriting knowledge has been used at several steps of this method and has improved recognition rates. Experimentations have been carried out on lower case multi-stroke letters. Models of letters non-existing in on-line handwriting have been added and that has also increased recognition rates. Our future work will focus on words.

References

[1] E. Anquetil and G. Lorette”On-line Handwriting Character Recognition System Based on Hierarchical Qualitative Fuzzy Modeling”, *Progress in Handwriting Recognition*, pp 109–116. World Scientific, 1997.

[2] D. S. Doermann and A. Rosenfeld, ”Recovery of Temporal Information from Static Images Handwriting”, *International Journal of Computer Vision*, 15:142–164, 1995.

[3] S. Jäger*Recovering Dynamic Information from Static, Handwritten Word Images*PhD thesis, Univ. of Freiburg, 1998.

[4] Y. Kato and M. Yasuhara, ”Recovery of drawing order from single-stroke handwriting images”, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22:938–949, sept 2000.

[5] P. Lallican and C. Viard-Gaudin, ”Off-line Handwriting Modeling as a Trajectory Problem”, *International Workshop on Frontiers in Handwriting recognition*, 1998, pp 347–356.

[6] J. Pettier and J. Camillerapp, ”Script representation by a generalized skeleton”, *Proceedings of the 2nd International Conference on Document Analysis and Recognition, (ICDAR’93)*, 1993, pp 850–853, Tsukuba, Japon.

[7] Y. Qiao and M. Yasuhara, ”Recovering dynamic information from static handwritten images”, *International Workshop on Frontiers in Handwriting recognition 9*, October 2004, volume 1, pp 118–123, Tokyo, Japan.

[8] L. Rousseau, E. Anquetil and J. Camillerapp, ”Recovery of a Drawing Order from Off-Line Isolated Letters Dedicated to On-Line Recognition”, *ICDAR*, August 2005, volume 2, pp 1121–1125, Seoul, Korea.

[9] H. L. Teulings, A. Thomassen and F. J. Maarse, ”A description of handwriting in terms of main axes”, *Computer recognition and human production of handwriting*, 1989, pp 193–211.

[10] C. Viard-Gaudin, P. M. Lallican, S. Knerr and P. Binter, ”The IRESTE On/Off (IRONOFF) Dual Handwriting Database”, *Proceedings of the Fifth International Conference on Document Analysis and Recognition (ICDAR’99)*, 1999, pp 455–458.