Gaëtan
Jean-Christophe
Jean-Michel
Pier
Vivianne

Gaetan ANDRE 20/11/12 17:14
Comment [1]: FDSGFGS

Remarques: Le modèle est provisoirement appelé "Hollerbach". C'est probablement à changer.
Il y a probablement des problèmes de mise en page à la fin.
Je n'ai pas mis des tables dans ce texte car je n'ai pas réussi à la faire. Je les présente dans un fichier à part.

Improving the Oscillatory Model of Handwriting

Gatan Andre February 2, 2012

Abstract

After a brief and non-exhaustive review of existing models of handwriting in the literature, we try to define which properties should exhibit a valuable generative model of handwriting. Then, we propose a model based on Hollerbach model. We describe a fast method to extract the parameters of this model from real strokes; we compare this method to usual non linear optimization method. We then do a first statistical analysis of extracted parameters. Finally we rate our model respectively to our definition of a valuable generative model of handwriting.

1 Introduction

1.1 Modelling handwriting

As for any task a human being is able to tackle, handwriting is a center of interest of many fields of science such as computer science, psychology, history, biology and so on. Among them, some try to understand the process of handwriting (and the process of reading) as it may happens in our body, the others try to computationally recognize or synthesize handwriting for practical applications. These two approach led to a certain amount of models of handwriting, of its generation, of its recognition.

In computer science, models of handwriting are mainly due to real life applications such as automatic recognition, synthesis (ie. (copy a person's signature or text and imitate his handwriting style [6,19,20]). As far as recognition is concerned, we generally separate o-line and on-line recognition. Dynamics of the strokes are not available in o-line recognition which is a serious drawback but not the only one. On-line handwriting recognition is often given in input a precise timestamped se-rie of points (coming from digital pens, mouse or touchscreens). On the opposite o-line recognition often operates on poor quality scannings or photographies, so there are thickness problems). [11] is a survey on handwriting recognition (to our knowledge the most recent on such a wide area).
Models inherited from recognition are often based on hidden markov models, especially for o-line recognition [14]. The sliding window method seems to be the most used strategy to sequentialize handwritten images. Other models are based on neural networks. The very impressive TabletPC software embedded in Windows since Vista [10] is based on delayed neural networks ([14] considers on-line recognition as a solved problem).

Another type of models appeared within the same time. These are called generative models. They try to explain which commands and which function can be the ground for the generated dynamic movement of handwriting. This can be seen as a particular case of biological movement study. Commands (or inputs) can be seen at dierent scale from neural signals to more abstract commands such as write the letter a. Ideally such a model would account for some properties observed in movement and handwriting, we will present them in the following section. As examples of generative models we can give [1, 4, 5, 12, 16, 17] This work focuses mainly on two points. First, try to de-fine what would be a valuable generative model of handwriting; second, to provide a generative model whose parameters can be quickly extracted from existing strokes.

1.2 The problem

Is this section we try to define which properties of handwriting would a good generative model acquaint for, then we show the limitations of the models briefly presented in the previous section.

Two sets of criteria seem to be important. The first one relies on studies of handwriting from a biologico-movmentologue-truc point of view, these criteria allow us to measure if a model is or not close to, let us say, the human[1] properties of handwriting. The second set of criteria is more pragmatic and is linked to computer science. It tries to evaluate the computability and the usefulness of the model to the task presented before (recognition, synthesis and writer identification). Of course, model designers will favour some of the criteria depending of the use they plan to do with it.

First, we will depict the biologicomovmentologue-truc criteria.

1.3 Plan

In section 2 we first present the Hollerbach generative model (section 2.1) from which ours is derived (section 2.2).

The section 3 section explains our method to quickly extract model parameters from real strokes. We first need to show a mathematical result (section 3.1). Afterwars each step of the algorithm is detailed (section 3.2).

Proposition 1. Section 4 describes several experiments. Among them one in which real writers are given some words to write. The algorithm presented earlier is then run on their strokes. The outputs are then analysed.

Proposition 2. Section 4 describes an experiment, in which real writers are asked to produce several handwriting samples. The algorithm presented earlier is then run on their strokes. The outputs are then analysed.

In the last section (section 5.3 we discuss on the model, its strengths and its drawbacks. We give some direction to improve it and the costs of it. Then, applications based on this model basis are given.

2 Our model

2.1 The Hollerbach model

One of the first, if not the first, oscillatory model of handwriting was proposed by Hollerbach [4]. It comes jointly with a modelling of the arm apparatus using springs : this will not be developed further.

In this model, handwriting is seen as the result of two superimposed oscillators on two distinct di

[1]Human as to be thought there as a embodied whole

rections of the plane. Although any non-sinusoidal oscillators could work as well, it is more convenient to use sinusoids. Moreover, the choice was more compliant with the spring muscle model. Oscilla-tors time evolution is define as:

$$\frac{dx}{dt} = a \sin(\omega_x t + \varphi_x) + c \quad (2.1)$$

$$\frac{dy}{dt} = b \sin(\omega_y t + \varphi_y) \quad (2.2)$$

where a and b are the horizontal and vertical velocity amplitudes, ω_x, ω_y φ_x and φ_y are respectively the frequencies and the phases associated to these directions. c represent the constant displacement to the right when writing. Direction on which oscillators vibrates are not necessarily chosen perpendiculars according to usual X and Y axis. It would be advocable to choose the horizontal axis for one of them and the slant direction for the other. In the rest of this paper we use the canonical direction of plan space as the direction of the two oscillators.

un petit exemple en image...

Model parameters (ie. $a, b, \omega_x, \omega_y, \varphi_x$ and φ_y) are supposed to be piecewise constant. There value change at vertical zero-velocity points (both for pa-rameters concerning horizontal and vertical axis).

Interestingly, the slant described by the angle β can be expressed as :

$$\tan \beta = \frac{b}{a \cos \varphi} \text{ where } \varphi = \varphi_x - \varphi_y \quad (2.3)$$

Another interesting value is the value of the hori-zontal velocity when the vertical velocity is null:

$$\Psi = \frac{dx}{dt}(t_{y_0}) = c - a \sin \varphi \quad (2.4)$$

This value gives the shape the drawn object will have. If Ψ is next to zero then the top corner will look sharp. If it is positive, the top corner the top corner will become rounded. Oppositely a negative value of P si will result in a full loop. This behavior is shown in table 1.

2.2 Improving the oscillatory model

Our model is highly inspired by the Hollerbach model. In fact we saying that it is new model might be a bit exagerated. However important

Shape Ψ

1.262

-43.74

37.41

Table 1: The shape of the top corner of the stroke depends on the value of Ψ.

questions and problems have not been addressed enough deeply and this is the aim of that work. Here they are :

1. Is the adding of the c parameter useful ?
2. Keeping the other parameters (a,b,ω_x,ω_y,Φ_x and Φ_y) piecewise constant, what are the best moment for them to change ?
3. Is there a way to quickly extractparameters from real strokes ?

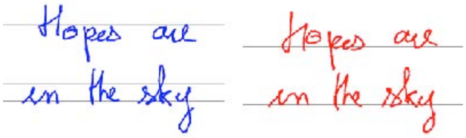
Answer to 1 : The parameter c is aims at representing the constant displacement to the right (or to the left) when writing...

Answer to 2 :

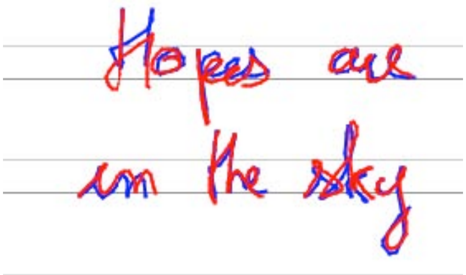
Answer to 3 : The main computational drawback of the Hollerbach model is the way we can get the parameters from the stroke : it is a non linear curve fitting problem. As we will see in section 4.1, usual optimization algorithms are not fast enough to give enough amount of data to study this model deeper. Thanks to the choice we made for ??, we were able to design a fast algorithm which will be presented in the next session.

3 From real stroke to the model

In this section we will present the algorithm used to retrieve parameters from a recorded stroke. In the remaining of this paper, this algorithm will be referred as the direct method. Figure 1 show the result of the direct method applied to a sentence. The method is applied on each stroke of the recorded sentence.



(a) Original (b) Synthesised



(c) Superimposed

Figure 1: In blue the original recording : Hopes are in the sky. In red the reconstructed signal using the extracted parameters. Note that the dynamic aspect can not be shown here but both original and synthesised signals can be replayed, we see that they are synchronised (ie. the oscillatory model is able too capture the dynamics of the movement).

3.1 An interesting Mathematical result

Before moving on and describe our fast method to find the model parameters out of recorded strokes we need to demonstrate a little result. Consider the following function:

$$f : x \rightarrow a \sin(\omega x + \varphi) \quad (3.1)$$

where a, ω and φ are independant to x. First, let us calculate the mean and variance of f between two successive zeros:

$$M = \frac{\omega}{\pi}\int_{\frac{-\varphi}{\omega}}^{\frac{\pi-\varphi}{\omega}} f(x)dx = \frac{2a}{\pi} \quad (3.2)$$

$$V = \frac{\omega}{\pi}\int_{\frac{-\varphi}{\omega}}^{\frac{\pi-\varphi}{\omega}} (f(x) - M)^2 dx = \frac{a^2(-8+\pi^2)}{2\pi^2} \quad (3.3)$$

Then, let us add the calculated mean and the square root of the calculated variance (ie. the standard deviation) and devide the result by a:

$$R = \frac{M + \sqrt{V}}{a} = \frac{2}{\pi} + \frac{\sqrt{2a^2(-8+\pi^2)}}{2\pi a} \quad (3.4)$$

$$R = \frac{4 + \sqrt{2}\sqrt{-8+\pi^2}\,\mathrm{sgn}(a)}{2\pi} \quad (3.5)$$

Which if we give a numerical approximation leads to (if a is positive):

$$R \approx 0.9443782250 \quad (3.6)$$

This result show that the amplitude of a sinusoidal signal can be approximated thanks to the sum and the standard deviation of this signal on a semi-period (zero to zero) independantly of the frequency and the phase.

3.2 Evaluating stroke parameters

3.2.1 Algorithm

Suppose the recorded stroke is represented by a chronological finite list of timestamped position:

$$S = (t_i, x_i, y_i)_{0 \le i \le N, N \in \mathbb{N}, \forall i > 0, t_i > t_{i-1}} \quad (3.7)$$

We apply the following steps on position components. Note that we present the steps for the x component but it is directly applicable to the other component.

Step 1 $x = (x_i)_{0 \le i \le N}$ is differentiated according to $t = (t_i)_{0 \le i \le N}$:

$$\frac{dx}{dt} = \left(\frac{x_i - x_{i-1}}{t_i - t_{i-1}}\right)_{0 < i \le N} \quad (3.8)$$

Step 2 Zeros are added to the beginning and to the end of the derivative signal. From a theoretical point of view this could be contested: it is clear (for example if you look at the pressure of the pen) that velocity is not always null when a writer begins or ends a written stroke ; but practically this allow to sensibly improve calculus of parameters.

Step 3 We apply a zero-crossing algorithm on the derivative that we have previously low-pass filtered. This prevent this algorithm to find clusters of zeros due to acquisition irregularities.

Step 4 Between two zeros, we said in section 2.2 that the parameters a, ω_x and φ_x were constant. We now show that we can calculate these values easily. Lets t_1 and t_2 the times of the two zeros. we have these equations :

$$\omega_x(t_2 - t_1) = \pi \quad (3.9)$$

$$\omega_x t_1 + \varphi_x = 0 \quad (3.10)$$

From equation 3.9 we can have ω_x and from equation 3.10 we can have φ_x. The way we obtain the parameters is excessively simple (not to say obvious), but the main drawback is quite an high sensitivity to zero finding. This point is developed further (5.1).

Step 5 The final step is to estimate the amplitude velocity a (note that we still are between two zeros timestamped t_1 and t_2). Lets define the arc A as the part of the derivative signal between t_1 and t_2:

$$A = \frac{dx}{dt}(i) \text{ between } t_1 \text{ and } t_2 \quad (3.11)$$

We approximate a thanks to result 3.1:

$$a = \mathrm{sign}(A)(\mathrm{mean}(A) + \mathrm{std}(A)); \quad (3.12)$$

where sign(A) can be evaluated as the sign of the middle element of A (notice that theoricaly, all elements in A are of the same sign).

4 Experiments and Results

In this section we present three experiments which are the earlier work of further developments of the exploitation of our model. The first experiment consist in comparing the algorithm (section 3.2.1) to usual optimization methods, both for accuracy of the result and heaviness of the computation. The second experiment is an example of how the reconstructed stroke can be used to drive a haptic arm from machine writing. Lastly, the ultimate experiment present a first step in the statistical analysis of the extracted parameters.

Kostrubiec Viviane 22/11/12 16:44
Comment [2]: Est-ce que cette introduction est toujours valable?

4.1 Rating the direct method

In order to rate the direct method a comparison with usual optimization methods is achieved. From the set of extracted parameters, a synthesis of the vertical and horizontal velocities signal can easily be achieved. We then compare the latter signal to the original signal in the least square paradigm.

4.1.1 A least square problem

The algorithm 3.2.1 can be modified in order to used a non-linear optimization method instead of the direct method. The loop on steps 4 and 5 are replaced by the method presented here.

The problem to solve can be expressed as a non-linear least square problem.

$$\hat{\theta} = \underset{\theta}{\mathrm{argmin}}\, f(\theta)$$

where

$$\theta = (a_i, \omega_{x_i}, \varphi_{x_i})_{1 \le i \le N}, (b_i, \omega_{y_i}, \varphi_{y_i})_{1 \le i \le N}$$

and

$$f(\theta) \to \sum_{j=0}^{N} \sum_{i=t_x(j)}^{N_{t_x}(j+1)-1} (x_i - a_j \sin(\omega_{x_j} t_i + \varphi_{x_j}))^2$$

$$N_{t_y}(j+1)-1$$

$$N \quad N$$

$+(y_i - b_j \sin(\omega_{y_j} t_i + \varphi_{y_j}))^2$

$j=0\ i=t_{b}(j)$

To find solutions we have used the large scale Reflective Trust-Region algorithm [2, 3].

You may notice that in θ parameters referring to horizontal speed velocity are independent from those referring to vertical speed velocity. Moreover, for each i,$(a_i,\omega_{xi},\varphi_{x_i})$ is independent from $(a_{i+1},\omega_{xi+1},\varphi_{x_{i+1}})$ and $(b_i,\omega_{y_i},\varphi_{y_i})$ id independent from $(b_{i+1},\omega_{y_{i+1}},\varphi_{y_{i+1}})$.

We therefore might be tempted to apply the Trust-Region algorithm to sub-parts of the problem instead of the entire problem. Several strategies have been explored, and applying Trust-Region to the whole problem gave the best results (both for time footprint and accuracy). Therefore we only present the latter strategy.

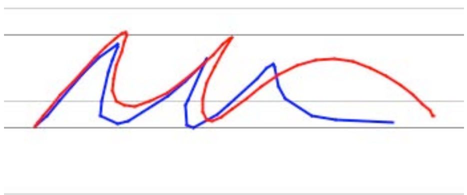
Trust-region needs a starting point θ_0. It was chosen randomly according to a normal distribution established from an earlier done statistical analysis of handwriting.

4.1.2 Comparison

The first point we can emphasize on is the setting up of this algorithm. Whereas the direct method is very straightforward with no tuning at all, reflective trust region needs a lot of parameters to be set before being run (such as stop conditions and much more). Moreover, it is nearly impossible to test all possible combinations of parameters. Hopefully the results don't change much with these parameters.

The first criteria we want to test is the accuracy of the two methods. In order to measure it we generate strokes thanks to parameters extracted from real ones. Then we get a measure of the accuracy by calculating the relative magnitude of the subtraction of the generated and the original signal. Results, presented in table numero and in figure, show that basically the the values given by both algorithm are quite similar (a compl´et´eer).

The other criteria studied is the computation time. For the Reflective Trust-Region method, Figure 2: The shape of the last spike is not what it should be.



these moments depends on the number of time the algorithm is launched for each stroke (because result depends on the starting point which is generated randomly). One run is barely su cient to get an optimal approximation but five to ten runs give good results most of the time (note that if we had not gone for normal distribution start points (cf ..), the number of necessary runs is much higher). Results are without appeal : the direct method is more than a thousand time faster than the usual optimization algorithm.

Our algorithm is as good as the usual optimizations method for calculating the parameters but clearly outperform them as far as computation time is concerned. Extracting parameters from real strokes has become clearly practicable for huge amount of samples, whereas it was not allowed with previous algorithms. Note that this gain is permitted only because we changed the point where these parameters change cf the improved model .

4.2 Common errors

Here we want to present three cases where the algorithm fails to approximate correctly the velocity signals.

4.2.1 Case 1 : straight line along the axes

4.2.2 Case 2 : ghost zero crossing

Figure 2 gives an example of what we called a ghost zero crossing bug. As explained all along this paper, our algorithm is very sensitive to zero crossing point search.

The reason for that error is that the horizontal velocity nearly touches zero but does not. The al-Figure 3: The shape of the last spike is not what it should be.

gorithm misses the zero crossing although it should put one 3.

4.2.3 Case 3: starting and ending points

4.3. Experimental Procedure

An experiment was carried out on real writers aiming to assess how Hollerbach model reconstructs real writing samples. Four unpaid volunteers, three male and one female, aged between 24 and 49, took part in the study. Two participants were self-claimed right handed, two others left handed.

The graphic task was performed on a computer-controlled graphic tablet (WACOM DTZ) with LCT screen of 261.1 × 163.2 mm size and 1280 × 800 resolution, inserted in a tablet (405.2 × 269.7 × 17 mm) which can be freely rotated just like a sheet of paper. A white sheet in landscape orientation, lined by blue lines spaced by 150 mm, was displayed on the screen. The stylus used was approximately the same size (174.8 mm long, with a diameter of 14.8 cm) and weight (17 g) as a normal ball point pen. Participants were seated in a high-adjustable chair, facing the graphic tablet posed on a table, and asked to adopt most comfortable writing posture. As soon as the stylus was brought of 5 mm the screen, the x and y spatial coordinates of the performed trajectories were digitized at 100 Hz with spatial resolution of 0.02 cm. The trace produced was displayed in real time on the screen and its coordinates stored for further analysis on a 3GHz PC. When the stylus raised 5 mm above the tablet, data recording stopped and produced trace removed.

The set of required forms was composed of the participant signature and of four handwriting prototypes distinguished by modellers (Edelman, & Flash, 1987; Edelman, Flash, Ullman, 1990; Grossberg, & Paine, 2000; Paine, Grossberg and Van Gemmert, 2004) and educators (Dumont, 2006) as basic components of all letters: a hook, a cup, a inversed gamma and an oval. Gamma appears in loop-based letters (b e f h k l), a cup in cup-based letters (m, n, u v w y i t), an oval in round-based letters (a c d g o p) and a hook in stroke-based letters (g y j, Dumont, 2006). The participant signature was selected as a sample of a natural, over-trained handwriting pattern. For each required form, participants were instructed to write six strings of unconnected, handwriting exemplars using cursive handwriting and their spontaneous writing speed. About 60 exemplars (SD = 12) were produced for each required form by each participant. To collect handwriting exemplars in as natural a setting as possible, participants were not asked to rest the pen at the starting position prior to beginning to write. As a result, their hands were already in motion when the pen contacted the writing surface.

4.4 Statistical analysis method

The aim of statistical analysis was to assess whether, despite the parsimony and genericity of our model, the degree of fit by Hollerbach model was in the range of that of Flash-Edelman model.

4.4.1. Data Reduction

For each participant and for each required shape, twenty central exemplars were selected for statistical analysis, in order to focus on the most natural tracing performance, free of warming-up and of fatigue effects. For each selected exemplar, the trace produced by participant and the two corresponding traces generated by Edelman-Flash and Hollerbach models were analyzed. For each trace, six dependent variables were studied: x-position, y-position, x-velocity, y-velocity, x-acceleration, and y-acceleration.

4.4.2. Goodness of fit

In line with Edelman and Flash (1987; Paine, Grossberg and Van Gemmert, 2004), numerical estimates of the degree of fit between the produced and the simulated traces was obtained by computing correlation index between the six dependent variables and their simulated counterparts (ie. r_{xx_pos}, r_{xx_vel}, r_{xx_accel}, r_{yy_pos}, r_{yy_vel}, r_{yy_accel}). Six correlation indexes captured the fit between the trace produced and the trace simulated by Hollerbach model (M_H), six others

Kostrubiec Viviane 22/11/12 12:06
Comment [3]: Le premier x (et le premier y) est un x (ou y) chapeau

between the trace produced and simulated Flash-Edelman model (M_{EF}). The classic formula for correlation index was used:

$$r_{ab} = \frac{\sum_{i=1}^{n}(a_i - \bar{a})(b_i - \bar{b})}{\sqrt{\frac{1}{n}\sum_{i=1}^{n}(a_i - \bar{a})}^2 \sqrt{\frac{1}{n}\sum_{i=1}^{n}(b_i - \bar{b})}^2}$$

where a and b represents a dependent variable and its simulated counterpart, respectively. This classic formula is distinct from that used by Flash and Edelman (1987), whose formula contains a possible artifact leading the authors to report correlations greater than 1.0 in some instances. Correlation index amounts to 1 for identical traces, to -1 between mirror-inversed traces, and tends toward zero when the goodness of fit deteriorates. For each participant, the 20 (Exemplar = {1, ..., 20}) × 4 (Form = {oval, gamma, cup, hook}) × 2 (Models = {Hollerbach, Flash-Edelman}) correlation indexes were computed for the six dependent variables and averaged over Exemplars.

4.4.3. Minkowski p-similarity

A global assessment of the (dis)similarity between Hollerbach and Edelman-Flash models was captured using Minkovski p-dissimilarity metric. This metric captured the distance between the two models in a 6-dimensional space, in which each dimension corresponds to the six correlation indexes. For each participant, the M_{EF} and M_H represents two points in the 6-dimensional hyperspace, and Minkowski distance, d(M_{EF}, M_H), corresponds to the vector joining them. Minkowski distance is a generalization of Euclidean distance:

$$d(M_{EF}, M_H) = (\sum_{i=1}^{p} \left|r_{ab\_EF} - r_{ab\_H}\right|^p)^{1/p}$$

when p is the number of dimensions, or the factor for the norm of the vector traced between the points M_{EF} and M_H. Minkovski distance equals 0 when the two methods are identical. In our data, it amounts to 1.348 when the distance is largest possible, because the largest distance between the best-fitting model (r_{ab}=1) and worst fitting model (r_{ab}=0) amounts to 1:

$$d_{max}(M_{EF}, M_H) = (\sum_{i=1}^{6} |1-0)|^6)^{1/6} = 1.348$$

4.4.4. Inferential statistics

Friedman ANOVA's, a nonparametric alternative to one-way repeated measures analysis of variance, were used. Each dependent variable was analyzed separately using a 2 (Model) Friedman ANOVA to test whether there was a statistically reliable difference between the fit provided by Hollerbach and by Flash-Edelman simulation. An additional 4 (Shape) Friedman ANOVA aimed to compare the Minkowski distances between the four Shapes. Statistically significant effects (at $p < 0.05$) are singled out by an asterisk.

4.5 Statistical analysis results

The Tables 1-4 display correlation indexes as a function of six dependent variable (column 1) for Hollerbach and Flash-Edelman model (column 2-5), the difference between the goodness of fit of both models (column 6) and Friedman ANOVA results (column 7-10). Positive difference signals that Hollerbach model fitted better the data than the Flash-Edelman one, and inversely.

4.5.1. Correlation index

Correlation indexes are displayed on the Tables 1-4. Overall, Hollerbach model led to larger r_{ab} for position and velocity than for acceleration. In Flash-Edelman model, this is true in case of circle, gamma and hook only. For all dependent variables but acceleration, the difference between Hollerbach and Edelman-Flash model was lower than 0.1.

Correlation indexes for circle are displayed on the Table 1. A 2 (Model) Friedman ANOVA revealed that there was a statistically reliable but marginal difference between the models for x-velocity, x-acceleration, y-position and y-velocity. Hollerbach model led to slightly larger correlation for position and velocities and to slightly lower correlation for acceleration.

Correlation indexes for gamma are displayed on the Table 2. A 2 (Model) Friedman ANOVA revealed that there was a statistically reliable but marginal difference between the models for x-position, x-velocity, x-acceleration, y-position and y-velocity. Hollerbach model led to slightly larger correlation for positions, velocities and to slightly lower correlation for x-acceleration.

Correlation indexes for hook are displayed on the Table 3. A 2 (Model) Friedman ANOVA revealed that there was a statistically reliable but marginal difference between the models for x-position, x-acceleration, y-velocity and y-acceleration. Flash-

Edelman model led to slightly larger correlation for all the four dependent variables.

Correlation indexes for hook are displayed on the Table 4. A 2 (Model) Friedman ANOVA revealed that there was a statistically reliable but marginal difference between the models for x-position, x-acceleration, y-velocity and y-acceleration. Flash-Edelman model led to slightly larger correlation for all the four dependent variables.

4.5.2. Minkowski p-similarity

Mean Minkovsky distance was of 0.242 (SD = 0.022) for circle, 0.298 (SD = 0.065) for gamma, 0.151 (SD = 0.063) for hook and 0.228 (SD = 0.041) for the cup, pertaining to the corresponding fractions of maximal distance: 0.179, 0.222, 0.111, and 0.169 of maximal distance. A 4 (Shape) Friedman ANOVA carried out on p-dissimilarities revealed no statistically reliable difference between shapes ($F(3) = 7.5$, $p < 0.057$).

4.5.3. Statistical analysis conclusion

The goodness of fit provided by Hollerbach and by Flash-Edelman was similar. Minkovsky distance between the models represented 0.229 of the maximal one. For position and velocity, the difference between Hollerbach and Edelman-Flash model was lower than 0.1. Acceleration was less well fitted than position and velocity by both models, and it was always less well fitted by Hollerbach than Flash-Edelman model.

The quality of the result highly depends on the way zero velocity crossing points are chosen. As we have seen in (ref bidule) missing a zero crossing point can be disastrous. If the algorithm finds too many points, the result will look OK but the spirit of the model is lost. ...

5 Discussion

In this section we want to address three points. First, we want to explain why our algorithm is so fast, ask if we can imagine to adapt it to cases where the moments of parameter changes of the oscilla-tory model would be di erent and try to answer that question. Second, we want to understand what is lacking in the present oscillatory model and try to give path for future work (param`atres pr´ef˜erentiels, points d'attraction, ...). Lastly, we try to undesrstand why generative models (oscillatory model as a particular case of) haven't been of much used in computer science for recognition, classification and synthesis tasks.

5.1 On the algorithm

The reason why our algorithm is so fast, is that the way we choose the dates where parameters changes (ie. zero vertical velocity for parameters linked to vertical direction and zero horizontal velocity for parameters linked to horizontal direction). Without this restriction our algorithm can not apply.

5.2 On the oscillatory model

Choosing changing times for the parameters at these moments leads to two serious drawbacks. If we estimate Φ (the relative phase), β (the slant according two Hollerbach model) and Ψ (giving the shape of the letter in Hollerbach's model); these parameters change every quarter period. Even worse, when studying them statistically,

Changins to simulate speed decrepancy

5.3 On the applications

Generative models are hardly used in practical ap-plications ; it is even truer for the Hollerbach based models (oscillatory approach). This may be because, these problems were earlier based on nonlinear regression. Thanks to our work, this approach is now more practicable.

We hope that from this work, we will be able to tackle a few problems. First, we think that know that we are able to extract parameters from huge amounts of data ,we hope that we will be able to to some classification (for writer, character or word recognition).

As we said in introduction, on-line recognition is a solved problem and there is a lot of work to try to extract temporal information from o -line strokes [7–9,13,15,18]. Generative models seem to be appropriate in these cases, we will try to explore that way.

Synthesis is also a interesting way of research. [6] gave a very good example of o-line style preserving synthesis base on glyphs and interpolation but we except to be able to do a on-line style preserving synthesis (that is respecting dynamics).

6 Conclusion

References

[1] H Chen, O.E. Agazzi, and CY Suen. Piecewise linear modulation model of handwriting. Analysis and Recognition, 1997., Proceedings of the, (908), 1997.

[2] TF Coleman and Y Li. An interior trust region approach for nonlinear minimization subject to bounds. SIAM Journal on Optimization, 6:418–445, 1993.

[3] T.F. Coleman and Yuying Li. On the convergence of interior-reflective Newton methods for nonlinear minimization subject to bounds. Mathematical programming, 67(1):189–224, October 1994.

[4] John M Hollerbach. An Oscillation Theory of Handwriting. Biological Cybernetics, 156:139– 156, 1981.

[5] HSR Kao and GP van Galen. Modelization of handwriting: A system approach. research in handwriting, 1986.

[6] Z Lin and L Wan. Style-preserving English handwriting synthesis. Pattern Recognition, 40(7):2097–2109, July 2007.

[7] E. Nel, J.a. du Preez, and B.M. Herbst. Estimating the pen trajectories of multi-path static scripts using hidden Markov models.
Eighth International Conference on Document Analysis and Recognition (ICDAR'05), pages 41–45 Vol. 1, 2005.

[8] E Nel, J Dupreez, and B Herbst. Verification of dynamic curves extracted from static handwritten scripts. Pattern Recognition, 41(12):3773–3785, December 2008.

[9] V Nguyen. Techniques for static handwriting trajectory recovery: a survey. Proceedings of the 9th IAPR International, pages 463–470, 2010.

[10] JA Pittman. Handwriting recognition: Tablet pc text input. Computer, pages 49–54, 2007.

[11] R Plamondon. On-Line and O-Line Hand-[20] Jue Wang, Chenyu Wu, Y.Q. Xu, H.Y. Shum, writing Recognition : A Comprehensive Survey. IEEE Transactions on Pattern Analysis And Machine Intelligence, 22(1), 2000.

[12] R Plamondon and Moussa Djioua. A multilevel representation paradigm for handwriting stroke generation. Human movement science, 25(4-5):586–607, October 2006.

[13] R Plamondon and C M Privitera. The segmentation of cursive handwriting: an approach based on o-line recovery of the motortemporal information. IEEE transactions on image processing : a publication of the IEEE Signal Processing Society, 8(1):80–91, January 1999.

[14] Thomas Pl¨otz and Gernot a. Fink. Markov models for oine handwriting recognition: a survey. International Journal on Document Analysis and Recognition (IJDAR), 12(4):269– 298, October 2009.

[15] Yu Qiao. Recovering dynamic information from static handwritten images. Handwriting Recognition, 2004. IWFHR, pages 118– 123, 2004.

[16] Y Singer and N Tishby. Dynamical encoding of cursive handwriting. Biological cybernetics, 71(3):227–37, January 1994.

[17] O. Stettiner and Dan Chazan. A statistical parametric model for recognition and synthesis of handwriting. In Pattern Recognition, 1994. Vol. 2-Conference B: Computer Vision & Image Processing., Proceedings of the 12th IAPR International. Conference on, volume 2, pages 34–38. IEEE, 1994.

[18] Christian Viard-Gaudin, Pierre-Michel Lallican, and Stefan Knerr. Recognition-directed recovering of temporal information from handwriting images. Pattern Recognition Letters, 26(16):2537–2548, December 2005.

[19] N Vincent, a Seropian, and G Stamon. Synthesis for handwriting analysis. Pattern Recognition Letters, 26(3):267–275, February 2005.

and Liang Ji. Learning-based cursive handwriting synthesis. Proceedings Eighth International Workshop on Frontiers in Handwriting Recognition, pages 157–162, 2002.