## 2 The improved Hollerbach model

### 2.1 The Hollerbach model

One of the first, if not the first, oscillatory model of handwriting was proposed by Hollerbach [1981]. It comes jointly with a spring-mass model of the arm apparatus, which will not be presented here.

In this model, handwriting is seen as the result of two superimposed oscillators on two distinct directions 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. Oscillators time evolution is defined as:

(1)

(2)

where and are the horizontal and vertical velocity amplitudes, , , and are respectively the frequencies and the phases associated to these directions. represent the constant displacement to the right when writing. Direction on which oscillators vibrates are not necessarily chosen perpendiculars according to usual horizontal and vertical 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 directions of the two oscillators.

Model parameters (ie. , , , , and ) are supposed to be piecewise constant. All these parameters change when the vertical velocity is null.

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

(3)

Another interesting value is the value of the horizontal velocity when the vertical velocity is null:

(4)

The sign and magnitude of this value indicates the particular shape of the trace at this point. If is next to zero then the top corner will look sharp. If it is positive, the top corner will become rounded. Conversely, a negative value of will result in a full loop. This behavior is shown in table 1.

|  |  |
| --- | --- |
| Shape |  |
|  | 1.262 |
|  | -43.74 |
|  | 37.41 |

Table 1: The shape of the top corner of the trace depends on the sign and magnitude of .

### 2.2 Improving the oscillatory model

Hollerbach’s model does not address important questions:

1. Is there a way to quickly extract parameters, given a recorded trace ?

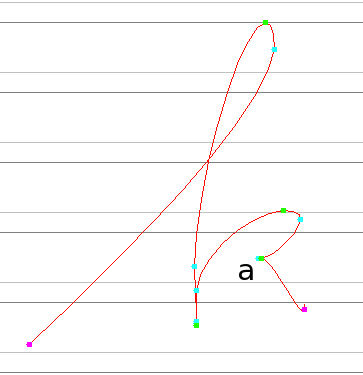
2. Is the ad-hoc parameter really necessary?

3. Parameters (, , , , and ) are only allowed to change at times of zero-crossing vertical velocity: is there any reason for this dissymmetry between x and y axes ?

The main computational drawback of the Hollerbach model is the way we can get the parameters from the trace: it is a non linear curve fitting problem. Usual optimizations methods are costly; we will present an algorithm that is much more efficient. This efficiency is possible because of our choice (that will be presented later) of the moments when parameters are allowed to change. For a comparison between our method and usual optimization methods, see [André,submitted].

The c parameter is of course representing the constant drift of the hand from left to right when writing, but it seems a rather ad-hoc way for taking it into account. Moreover, it imposes the X axis to be exactly aligned with the trace direction, and it introduces a dissymmetry between X and Y. We will show that it can be simply omitted, and that the drift can be accounted for by cumulating phase values between cycles.

About the dissymmetry between X and Y, it is true that the majority of shape changes occur at Y velocity zero-values, which is visible in the fact that most loops and peaks are directed upwards or downwards. However, an allograph such as “k” has clearly a parameter change on the X velocity zero crossing near point a, which Hollerbach’s model cannot handle properly.



### 2.3 The Improved Hollerbach Model

Our improved version of Hollerbach’s model is both simpler, symmetric and allows for a fast parameters extraction algorithm. It is composed of a set of modelling equations, a model of temporal changes and an algorithm of parameter extraction

1. Modelling equations. Oscillators time evolution is defined as:

(3)

(4)

1. Model of Parameters are constant by parts; they are allowed to change on X and Y separately; X (resp. Y) changes only when X (resp. Y) velocity is null.
2. A trace is represented by :
   1. A set of initial parameters values:

, , , , and

* 1. Two parameter series :

### 2.4 Extracting parameters from real traces

|  |  |
| --- | --- |
|  |  |
|  |  |
|  | |
|  | |

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

#### 2.4.1 An interesting Mathematical result

Before moving on and describe our fast method to find the model parameters out of recorded traces we need to demonstrate a proposition on harmonic functions. Consider the following function:

(5)

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

(6)

(7)

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

(8)

(9)

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

(10)

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

#### 2.4.2 Evaluating trace parameters

##### Algorithm

Suppose the recorded trace is represented by a chronological finite list of time stamped positions:

(11)

We apply the following steps on position components. Note that we present the steps for the component but it should be also applied to the y component.

**Step 1**

is differentiated according to :

(12)

**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 trace ; but we cannot infer this information from the time stamped positions only.

**Step 3**

We apply a zero-crossing algorithm on the derivative that we have previously low-pass filtered (using a flat window of size 6). This prevents this algorithm to find clusters of zeros due to acquisition irregularities or noise.

**Step 4**

Between two zeros, we said in section 2.2 that the parameters , and were constant. We now show that we can calculate these values easily. If and are the times of two successive zero velocities, we have:

(13)

(14)

From equation 13 we get and from equation 14 we get . This computation is extremely simple and fast to perform; however it is very sensitive to errors during the zero crossing algorithm. This point is developed further (5.1).

**Step 5**

The final step is to estimate the amplitude velocity (note that we still are between two zero velocity times and ). Let us define as the part of the derivative signal between and :

(15)

We approximate amplitude using result 3.1:

(16)

where is the sign of the middle element of (notice that theorically, all elements in are of the same sign).

#### 2.4.3 Reconstructing the trace

From the previously calculated parameters a trace can be reconstructed. Knowing the initial time, initial position and initial parameters , , , , and , we can calculate the synthetized trace using only , , and . Zero-crossing times, and , can be computed at reconstruction time, so there is no need to store them.

Conversely, the time stamped positions in the trace can be reconstructed using:

* the initial time and position, and the initial parameter values , , , , and
* the extracted parameter values , , , , and

The resulting function is a succession of half-periods of sinusoids whose parameters are given by the series , , , , and

Moreover, either the pair of , can be omitted in the series, because it can be inferred from all other parameters

Finally, the complete reconstructed trace needs only 4 parameters per half period of writing.

Figure 1 shows an example of parameters extraction and of the reconstructed trace.