How a Font Can Respect Basic Rules of Arabic Calligraphy

Abdelouahad BAYAR
Ecole Supérieure de Technologie - Safi
Cadi Ayyad University
Dar Si Aissa Road, P.O.B 89, Safi 46000
Morocco
bayar@ucam.ac.ma

Khalid SAMI
Faculty of sciences - Marrakech
Cadi Ayyad University
12 BD. My Abdellah, P.B 2390, Marrakesh 40000
Morocco
k_sami@ucam.ac.ma

Abstract This paper describes formal bases for designing a PostScript dynamic font respecting some basic rules of Arabic calligraphy. This font will provide, for instance, a vertical curvilinear characters dynamic stretching that depends on their horizontal stretching according to the Calligraphic rules. Vertical and horizontal ligatures will also be supported by the proposed font. The techniques used can be applied in various digital typography contexts such as in typesetting mathematics, to get big delimiters for instance. The results presented here give both extensions and corrections of some already adopted techniques.

Keywords: Dynamic Font, Curvilinear Stretching, PostScript, Complex ligatures.

1. Introduction

Processing documents written in an Arabic alphabet-based script is not an easy business. For instance, in order to justify texts, stretching characters replaces the ordinary inserting spaces between characters. This stretching is almost mandatory in a cursive writing such as Arabic. Systems such as ArabiTEx [18] or Ω [31] can help in typesetting a quite complex Arabic documents, say for example mathematical texts, in a quite good quality. However, most of the software designed for typesetting documents based on this script don’t take sufficiently into account the complex rules of characters stretching. The CurExt [5] system provides some solutions for dynamic fonts and extensible symbols in Arabic mathematical formulas. So, it gives, by the way, solutions for the composition of the Arabic keshideh (a small flowing curve stretching characters in Arabic-alphabet based scripts). This package combined to RydArab [4] run with both ArabiTEx or Ω and provides support for the composition of right to left mathematical equations based on Arabic symbols.

Actually, the stretching of characters according to the Arabic calligraphy rules requires that the fonts development languages support what can be called ‘fonts dynamic’ as PostScript language does [2]. Unfortunately, this support is not offered by Metafont [10] the first system used for developing TEx’s fonts. The CurExt system generates curvilinear stretching of Arabic characters through repeated calls of Metafont. The D. Berry system [8] also offers the possibility of characters horizontal stretching up to some level. However, the Arabic calligraphy rules require a vertical stretching also.

In Arabic calligraphy, some parts of the letters are written, directly with the nib’s head, and other parts are drawn: the contour is set first, with the right up corner of the nib’s head (the qalam’s tooth) and therefore, it is darkened as with a brush [3, 22, 25]. The figure 1 presents a letter REH where the written part is in black and the drawn part, to be darkened, in a different gray. This figure comes from the calligraphy handbook [22]. This fact is due to the rectangular shape of the nib’s head and to its constant inclination with respect to the vertical direction. Such considerations are not to take into account in traditional typography. Most of the pens have a circular head and they can go with any inclination. In the proposed font, the written parts would be shaded
Figure 2. a) The nib’s head in 192 points, b) The standalone REH in 192 points, c) The written part of the letter REH in 192 points, d) The drawn part of the letter REH in 192 points.

through considering the nib’s head motion (see later) and the drawn ones are considered as simple contours to darken. The figure 2 shows the standalone REH in our font in 192 point size. The figure 2(a) presents the nib’s head in the size 192 points for the Naskh style. The corner marked with a solid circle models the part of the nib’s head used by a calligrapher to blacken the drawn parts of letters. The standalone REH is given on 2(b), in the same size. The written and the drawn parts are showed in the figure 2(c) and 2(d) respectively. Some states of the nib’s head motion are given on the written part of the letter REH. One important thing to underline is that the surface in the written part never exceed the width of the nib’s head nor it is exceeded by the nib’s head width (kept at 70° inclination). The standalone REH implemented in the font is slightly different from the one presented in the figure 1 because the letter scanned and used to determine the encoding of the letter REH in the font comes from another calligraphy book [3].

In [8], particular curves for representing stretchable pieces of the characters were to find out. The approach adopted in [8] to stretch letters can lead sometimes, to write some parts of the letter, in a width exceeding, or exceeded by, the nib’s head width. See, for instance, the stretched QAF in [2, p.1433] (see the figure 3). The figure 3 presents the letter QAF with a null stretching and a stretched QAF from [8] (a slightly different variant from the same letter). We tried to show the same states (for the same parameters of the Bézier curves representing the contours) of the nib’s head in the two cases of the letter QAF. We remark that in 3(a), the width of the nib’s head is approximately equal to the surface. The letter QAF with a null stretching is a genuine written QAF. In the stretched QAF (see figure 3(b)), on the right, the nib’s head width exceeds largely the surface darkened of the letter and conversely on the left of the same letter. So, some parts that are written in the null stretching case can not be obtained when the letter is stretched unless they are drawn. This is due to the curves to stretch choice. Actually, these curves should verify a dependency property since they represent trajectories of two corners of the nib’s head following the same movement with respect to a translation (take into account the motion in shading the surface razed with the nib’s head).

A significant progress in the development of Arabic mathematical dynamic fonts has been presented in [27]. A type 3 PostScript font has been developed for the purposes of CurExt in order to produce documents respecting the Arabic calligraphy rules and allowing the support of stretchable mathematical symbols. In CurExt, the variable sized character, in the suitable size, is produced via repeated calls of the program. Although our dynamic type 3 PostScript font aims at producing variable sized characters too, the approach adopted is different. The dynamic is inherent to the font, not in the font generator. The stretching’s encoding is not the same. Moreover, some proposed modules can be integrated into certain typesetting systems such as LyX [21]. The text displayed in screen fonts will be used with these formatting mod-
ules through the parameters of pagination and the font would help in generating directly a PostScript document respecting the rules of Arabic calligraphy.

In Arabic calligraphy, some kinds of ligatures are mandatory and others are only for aesthetic. All of the systems cited before use fonts providing some mandatory ligatures such as the horizontal ligatures exactly on the baseline and some vertical ligatures for aesthetic. However, there are other not provided mandatory and also aesthetic ligatures like oblique, big strictly above the baseline, strictly above the baseline and horizontal below the baseline ligatures. Those ligatures are studied in the paper.

We'll begin by presenting the curvilinear stretching in Arabic calligraphy. This will lead to some general Arabic calligraphy constraints. Then, we will focus on ligatures, their categories and kinds. The support of the keshideh can then be formalized in mathematical model, implemented in a font and thus compared to other approaches.

2. Curvilinear stretching in Arabic Calligraphy

2.1 Stretching in Arabic Calligraphy

As it has been mentioned before, the keshideh is a small flowing curve that stretches characters or junctions between characters in Arabic-alphabet based scripts. It is used, for instance for justification. So, instead of inserting spaces (blanks) between words, characters and/or ligatures are stretched according to a structured set of calligraphic rules. More rules of justification in Arabic documents can be found in [26]. In that paper, the authors present most of the existing methods of justifications especially the keshideh. In our article, we give a method to implement the keshideh in a PostScript font. This stretching goes in both the horizontal and vertical directions. Horizontal curvilinear stretching can go to a maximal value of 12 diacritic points. To this horizontal maximal value corresponds a maximal vertical stretching of a half diacritic point downwards. Moreover, we can distinguish two kinds of stretching:

- Stretching the ligature between two connected letters and
- Stretching inside a letter.

For instance, consider the words presented in figure 4. The first line displays the word طارم with three extensions, zero, 6 and 12 diacritic points (from right-to-left). The keshideh takes place between the letters و and ر. In the following line, the word حما is stretched with zero, 9, 11 and 12 diacritic points. Then, the stretching is performed inside the last letter QAF ق. In the calligraphy books [3, 22, 25], the letter QAF without stretching is different from the stretched one. That is also true for all letters that undergo stretching inside them. This property hadn't been respected in [8] (see figure 3 and 4 to compare). In the figure 4, the keshideh is darken in Gray and the boxes are reproduced for more clarity. Most of Arabic letters, are composed in two parts. One is static and an other that is dynamic. The dynamic part is the set of curves that undergo really the stretching. As example, we have to look to the letter أ on the figure 4. The static part is in black whereas the dynamic one is in Gray.

2.2 Dynamic fonts and the support of stretching

The stretching amount is the difference between the length of the current line and the sum of the lengths of the words added to the sum of the normal blanks (a normal blank is the minimal blank used to separate normally two words). This stretching amount depends on the line context. A good font to support the stretching may allow continuous variations in stretching. We have here the same need in type of font like in [8, 13, 14]; a dynamic font.

2.3 Implications for the design of the font

A font with a good support for the stretchability may be based on a language that provides the possibility to use variables in character procedures that can be computed on the fly (dynamically). Postscript dynamic fonts [12] and TrueType fonts [11] are examples of such kind of fonts. As we are here extending and improving what has been done in [8], we will rather opt for PostScript. The PostScript fonts of type 3, as proposed in [12] use the full PostScript language and violates the assumption for the type 1 fonts [1], namely the printing in a fast way thanks to the concept of catching the character bitmaps for ulterior usage, the ability to provide hints to improve low resolution or small point size rasterization and also the ability to be handled by the Adobe Type Manager (ATM). Here I agree with the opinion of the author of [8] that the beauty of what can be done with dynamic fonts, and so write an Arabic text with the keshideh outweighs the disadvantages.
2.4 Characteristics of the stretching in Arabic Calligraphy

In the Naskh style, the nib’s head behaves as a rectangle of width $l$ and thickness $e = \frac{l}{6}$. This rectangle moves with a constant inclination angle of about $70^\circ$ with regard to the baseline. A nib’s head with $l = 12$ mm (and $e = 2$ mm) is presented in the figure 5.

As mentioned before, the stretching goes in both the horizontal and vertical directions. In addition, the stretching ought respect the following constraints:

- The horizontal stretching can go from 0 to 12 diacritic points (the dot in the letters ب and ج is a diacritic point and it can be used as a metric unit).
- The vertical stretching depends on the horizontal one and varies from 0 to half of a diacritic point. The stretching model provided in [8] considers stretching only in the horizontal direction.
- In the connection of two stretched zones, the state of continuity of degrees 0 and 1 is preserved. This will be well explained through considering the Bézier geometrical representations. Prematurely, the reader can see the figures 41 and 46. The set of curves in the figure 46 is obtained after a stretching in the set of curves in the figure 41. We can remark that the angle on the point $L_{10}$ is conserved on $L_{20}$ and the alignment on $R_{13}$ is conserved on $R_{23}$.

In the figure 6, the word طفف is presented with a null stretching in the first line, whereas, in the second line the same word is displayed with an horizontal stretching of 12 diacritic points and the corresponding vertical stretching that is a half of a diacritic point. The nib’s head (in black) appears in the gray of the keshideh. We can remark that the thickness (in the direction of the nib’s head inclination) doesn’t change and consequently, the stretching model is an optical scaling in opposition to the linear scaling (see [14] about optical scaling). In the following, the stretching model is presented in the same way.

Let us look at the relationship between the external curves delimiting the surface razed with the nib’s head. We have to darken the surface taking into account this relationship. That has not been considered in [8]. And now, we’ll consider that:

The character is defined by the curves representing the nib’s movement not by those representing the limits of the surface razed with the nib.

This means that a character results from blacking the surface razed with the rectangle representing the nib’s head. As the four corners of the nib’s head follow the same movement, the movement can be represented by the curve corresponding to the way followed by one of the four corners. Say for instance, the extremity surrounded with a small circle in the figure 5. Of course, the movement of this corner when writing a character is determined by a set of Bézier curves. So, we’ll consider a unique Bézier curve.

Let $B_1$ be a Bézier curve with the four control points...
Consider $B_2$, the Bézier curve with the four control points $M_{23}, M_{22}, M_{21}$ and $M_{20}$ such that:

$$M_{2i} = t_{\overrightarrow{u}}(M_{1i}), \quad i \in \{0, 1, 2, 3\}$$

where $t_{\overrightarrow{u}}$ is the translation of vector $\overrightarrow{u}$ such that

$$\overrightarrow{u} = (l \cos (\alpha), l \sin (\alpha))$$

The surface $S_{t1}$ delimited by the curve $B_1$, the segment $[M_{13}, M_{23}]$, the curve $B_2$ and the segment $[M_{20}, M_{10}]$ coincides exactly with the surface razed by the edge $l_1$. The surface is filled in thin gray in the figure 7.

The surface $S_{t2}$ razed by the edge $l_2$ can be obtained through translating the surface $S_{t1}$ with vector $\overrightarrow{v} = (l \cos (\alpha + \frac{\pi}{2}), l \sin (\alpha + \frac{\pi}{2}))$ (see figure 8).

Let $S_{t2} = t_{\overrightarrow{v}}(S_{t1})$ be the result of this translation.

$S_{e1}$: surface bounded by $B_1$, $[M_{13}, M_{33}]$, $B_3$ and $[M_{30}, M_{10}]$, that is the surface razed by edge $e_1$.

$B_3$: a Bézier curve with control points $M_{33}, M_{32}, M_{31}$ and $M_{30}$ so that:

$$M_{3i} = t_{\overrightarrow{v}}(M_{1i}), \quad i \in \{0, 1, 2, 3\},$$

and $S_{e2} = t_{\overrightarrow{v}}(S_{e1})$.

These four surfaces allow the reconstitution of the surface razed by the nib’s head. The figure 10 shows an illustration of this situation. It gives the control points of the Bézier curve modeling the movement of the left bottom corner.

This way of blacking the surface razed by the nib does not give always good result. Actually, consider the example on the figure 11. In order to simplify, consider a nib with a very thin head (with negligible thickness). The surface obtained through applying the technique used previously (see figure 11) does not coincide exactly with the surface razed with that nib (see figure 12). The geometric reason why it is so is the existence of an element $t_0 \in ]0, 1[$ such that $B'_1(t_0)$ ($B'_1(t_0)$ is a vector) and the
vector $\overrightarrow{M_{10}M_{20}}$ are collinear (such condition was not satisfied in the case of the previous examples). More formally, consider the function $R_1$ defined as follows:

$$R_1 : [0, 1] \rightarrow \mathbb{R}$$

$$t \mapsto (\overrightarrow{M_{10}M_{20}} \land \overrightarrow{OB_1(t)}) \cdot \overrightarrow{k}$$

when the considered Bézier’s curves are defined on the affine plane $\mathbb{R}^2$ reported to the orthonormal basis $R\left(O, \overrightarrow{i}, \overrightarrow{j}\right)$ containing the origin $O$ and that is a subspace of the affine space $\mathbb{R}^3$ with the orthonormal basis $R\left(O, \overrightarrow{i}, \overrightarrow{j}, \overrightarrow{k}\right)$ and the operator $\land$ stands for the vectorial (cross) product.

$R_1(t)$ is the direct orthogonal range of the Bézier curve $B_1$ according to the vector $\overrightarrow{k}$. If $R_1(t)$ is null at $t_0 \in [0, 1]$, that means that $B_1'(t_0)$ and $\overrightarrow{M_{10}M_{20}}$ are superposed. As that $R_1'(t) = \left((\overrightarrow{M_{10}M_{20}} \land \overrightarrow{OB_1(t)}) \cdot \overrightarrow{k}\right)'(t) = (\overrightarrow{M_{10}M_{20}} \land B_1'(t)) \cdot \overrightarrow{k}$, we get the interpretation:

- if $R_1$ is monotone on an interval $[0, 1]$ then the razed surface can be obtained through applying the method mentioned before,
- the method fails if the monotony’s sense of $R_1$ changes.

We can remark that the study of the function $R_1$ can help to blacken the surface razed in the figure 12. The curve $B_1$ is a perfectly determined Bézier curve. The study of $R_1$ over $[0, 1]$ shows that $R_1$ admits an extremum at $t_0 = 0.6184$ and the monotony’s sense doesn’t change over the sub-intervals $[0, 0.6184]$ and $[0.6184, 1]$.

Now, let us decompose $B_1$ with respect to the generalized Bézier algorithm of refinement (case of none median decomposition) [7, 15, 23, 29] with respect to the coefficient $t_0 = .6184$. So we’ll get two curves $B_{11}$ and $B_{12}$ whose corresponding razing can be obtained through proceeding as in the example in figure 7. The two razings are on the figures 13 and 14.

The two surfaces overlap and reconstitute a surface that fit exactly with the surface razed with the segment $[M_{10}M_{20}]$ (see figure 15). This way of proceeding can be very useful for the design of characters shapes in PostScript or Metafont so that the trajectories of the summits nib overlap.

The difficulties to find out a way to represent and to coat the nib’s head movement reminds those met in the knuth’s approach in metafont [10] or those found in the kinch’s approach in MetaFog [28]. The comparison will be considered in the sub-paragraph 4.3.
3. On some Arabic calligraphy constraints

3.1 The notion of baseline

The baseline is one of the basic notions in digital typography [9, 30, 10, 2]. It constitutes the basic reference for positioning the characters, or symbols in a mathematical formula and so on. In Arabic writing normative studies, there is no general agreement on the necessity of this reference. Some calligraphers think that the notion of baseline is meaningless. Others think that its position is not steady [27]. Nevertheless, the reference to the baseline can be very helpful and even necessary for positioning signs. Actually, some parallelism among words and even among letters, with regard to a virtual horizontal line, can be noticed in any written line. We will use the common methodology related in the A. El Husseiny’s calligraphy’s book [3] to define the baseline used in the development of a Naskh font providing the possibility of characters curvilinear extension. According to this reference, two groups of isolated Arabic letters may be distinguished: those to place above the baseline and those containing pieces above and/or below the baseline (see figure 16).

3.2 Arabic calligraphic metrics

The measure unit in Arabic Calligraphy is the diacritic dot used to differentiate some Arabic letters with the same shape. For example, the letters HAH, JEEM and KHAH (from the right to left), in initial position, are presented in the figure 17. The diacritic point is the small square filled in thin gray. The same dot is the measure unit used to define the letters dimensions. In the Naskh style, the diacritic point is a filled in square with sides of a width equal to the width of the nib’s head and rotated about 60°. In the figure 18, some metric indications to write the letter HAH (in the initial form with oblique ligature) are in charge of the diacritic dots. In calligraphy, the space between two consecutive dots in a measure is about 0.45 to 0.5 of a calligraphic dot. But, in our work, all the metric values are given in terms of diacritic points, without spaces (see figure 19). In the figure 19, an example of a stretching of the word طفـر is presented. The first line displays the word طفـر with the keshideh in gray in its initial size i.e with a null stretching, the width’s keshideh is two diacritic points and a fraction. In the following line, the same word is stretched 12 diacritic points horizontally. To the horizontal stretching corresponds a vertical stretching of half a diacritic point downwards with respect to the baseline (see later for the rules of stretching). We can also get information about metrics from [26].

3.3 Connections and ligatures

The Arabic writing is cursive. The notion of ligature is of a particular importance. Handling ligatures in Arabic is of a strong complexity in comparison with Latin alphabet based scripts. Now, let us give some details on connections of letters since they are the bases of the ligatures. In Arabic calligraphy, a connection is a fine curve following the nib’s head inclination that connects a letter to another before or after it. So, we can talk about
Figure 19. A stretching of 12 diacritic points horizontally and 0.5 vertically preceding and succeeding connections (see figure 20 and 21). In the sequel, categories and examples of connections are given. Actually, two categories of connections can be distinguished, namely:

- **Horizontal connection**: horizontal connections can be regarded as tools to tie letters so that the boxes containing these letters are horizontally positioned. In the figure 20 and 21 are showed the horizontal succeeding connection of the letter KHAH and the preceding connection of the letter TAH respectively. The connections are in thin gray. In the figure 22, the word beer formed of the letters  and  with their digital boxes is presented. So, the definition of the horizontal connection is more clear.

- **Vertical connection**: a vertical connection allows connecting two letters so that their boxes can be positioned vertically (see figure 25). The figure 23 and 24, displays the succeeding and preceding connections of the two letters NOON and HAH.

When they are in the middle of the word, some Arabic letters can have both a succeeding connection and a preceding connection at the same time. For instance, consider the letter  in the middle of the word  (see the figure 26).

We define a ligature as a set of one or more pair of connections. A pair of connections is composed of a succeeding connection and a preceding one. There are two categories of ligatures namely:

- **Horizontal ligature**: it is a combination of a horizontal succeeding connection and an horizontal preceding one. In the figure 22, the ligature between the letters KHAH in initial position and TAH in final one of the word  is an example of such ligature.
• **Vertical ligature**: it is a set of pairs of vertical connections. Every pair is formed of a vertical succeeding connection and a vertical preceding connection. In the figure 25, a vertical ligature appears between the letters NOON and HAH. It contains one pair of vertical connections. It is then a simple vertical ligature. In the figure 27, the ligature is composed of two pairs of connections connecting the letter LAM to MEEM and MEEM to JEEM. The ligature is then double. Whereas, on the figure 28, the ligature connects four letters and thus is formed of three pairs of vertical connections. It is a triple ligature.

Some difficulties occur in handling vertical ligatures. Solutions can be found through considering the connected letters as a single character [30]. We’ll adopt the same strategy concerning this kind of ligature. We will refer to those characters as compound letters. In this way, the compound letters are considered as single characters with horizontal preceding connections, horizontal succeeding connection or both of the two connections. They can also be considered as standalone letters. All of these cases are presented in the figure 30. So, the compound letters can accept preceding or succeeding connections or contribute simply as standalone characters in words (see figure 31). They can also contribute in stretching (see figure 29).

We can remark that the handling of vertical ligatures can be brought to the study of the horizontal ligatures. For that reason, we will give in the following all kinds of horizontal connections and ligatures. We have seen in the subsection 2.4 that the movement of the nib's head is characterized by one of its corners. We present then the kinds of connections and ligatures in their geometrical forms through considering the trajectory of the left bottom corner of the nib's head. In designing the letters of our font, the connections are formed of one cubic Bézier...
In the following, we suppose that the preceding connections have \( L_0, L_1, L_2 \) and \( L_3 \) as control points whereas, the succeeding connections have \( R_0, R_1, R_2 \) and \( R_3 \) as control points. There are various kinds of horizontal connections. A lot of them are, in general, not respected for reasons of simplification when that is not due to ignorance. Horizontal connections can be one of the five kinds:

- **Oblique connection**: in the figure 32 are presented in the left a preceding connection and in the right a succeeding one. These connections are oblique because the segments \([L_2, L_3]\) and \([R_0, R_1]\) are not parallel to the baseline.

- **Horizontal connection big-strictly above the baseline**: a connection is horizontal big-strictly above the baseline when the segment \([L_2, L_3]\) for the preceding connection and segment \([R_0, R_1]\) for the succeeding connection are parallel to the baseline and over it about one and 0.54 diacritic Point (see figure 33).

- **Horizontal connection strictly above the baseline**: the connections presented in the figure 34 are horizontal strictly above the baseline because the segments \([L_2, L_3]\) and \([R_0, R_1]\) are parallel to the baseline and over it about a half of a diacritic point (0.55 precisely).
• **Horizontal connection exactly on the baseline**: in this case \([L_2, L_3]\) and \([R_0, R_1]\) are exactly on the baseline as in the figure 35.

![Figure 35. a) Horizontal preceding connection exactly on the baseline, b) Horizontal succeeding connection exactly on the baseline](image)

• **Horizontal connection below the baseline**: \([L_2, L_3]\) and \([R_0, R_1]\) are parallel to the baseline and below it, about an amount that goes from zero to a half of diacritic point. In general, the depth of this connection depends on its width. These connections are used in certain cases of the keshideh. An example is given in the figure 36.

![Figure 36. a) Horizontal preceding connection below the baseline, b) Horizontal succeeding connection below the baseline](image)

Now, we can define what does **horizontal ligature** means geometrically. An horizontal ligature is a combination of two horizontal connections \(C_1\) and \(C_2\) such that:

- \(C_1\) and \(C_2\) are of the same kind (oblique, big strictly above the baseline, ... etc.),
- \(C_1\) is a preceding connection,
- \(C_2\) is a succeeding connection and
- if \(L_0, L_1, L_2\) and \(L_3\) are the control points of \(C_1\), and \(R_0, R_1, R_2\) and \(R_3\) are control points of \(C_2\) then \(L_3\) and \(R_0\) coincide and, \(L_2, L_3\) and \(R_1\) are aligned.

Note that the kind of the ligature is entirely determined by the kind of its connections. So, there are five kinds of horizontal ligatures. A font respecting a minimum of the Naskh calligraphic rules may support at least the horizontal connections (ligatures) exactly on the base line and below the baseline. The connections below the baseline are used in to stretch. We can state that these two kinds of horizontal ligatures are mandatory whereas other horizontal and vertical ligatures are rather aesthetic. However, Arabic handbooks calligraphy show more aesthetic ligatures than mandatory ones. That’s one of the motivations behind the design of a font that allow producing documents closer to handwritten texts. The horizontal oblique connections big strictly above the baseline, strictly above the baseline and strictly below the baseline are not supported in [18, 8, 4, 31].

4. Model and support of the keshideh in Arabic calligraphy

4.1 Keshideh: the mathematical model

In this sub-section, we give the geometrical representation and processing of the curvilinear stretching. In order to present how to handle geometrically the keshideh, we’ll first define the way of stretching some particular curves that constitute basic elements of the keshideh. Consider the following notations:

- The notation \([M_0, M_1, M_2, M_3]\) is the Bézier curve with control points \(M_0, M_1, M_2\) and \(M_3\),
- The set \(\mathcal{B}_1\) is the set of Bézier curves \([M_0, M_1, M_2, M_3]\) with an invariant concavity verifying:
  \[
  \overrightarrow{M_2M_3} = \lambda \overrightarrow{i}, \lambda \in \mathbb{R}_+^* \text{ and } 0 \leq \left(\overrightarrow{M_0M_1}, \overrightarrow{i}\right) \leq \frac{\pi}{2},
  \]
  (see figure 37), where \(\overrightarrow{i}\) is the axis \(X\) director vector and \(\left(\overrightarrow{u}, \overrightarrow{v}\right)\) stands for the angle between the two vectors \(\overrightarrow{u}\) and \(\overrightarrow{v}\).

![Figure 37. Curves of type 1 (in \(\mathcal{B}_1\))](image)
• The set $B_2$ is the set of Bézier curves $[M_0, M_1, M_2, M_3]$ with an invariant concavity verifying:

$$\overrightarrow{M_0 M_1} = \lambda \overrightarrow{I}, \lambda \in \mathbb{R}^+_0 \text{ and } 0 \leq \left( - \overrightarrow{I}, \overrightarrow{M_3 M_2} \right) \leq \frac{\pi}{2} \text{ (see figure 38).}$$

![Figure 38. Curves of type 2 (in $B_2$)](image)

The keshideh is a juxtaposition of two Bézier curves, $B_1$ from the set $B_1$ and $B_2$ from the set $B_2$. If $L_0$, $L_1$, $L_2$ and $L_3$ are the control points of $B_1$ and $R_0$, $R_1$, $R_2$ and $R_3$ are the control points of $B_2$ then $L_3$ and $R_0$ are equal.

The keshideh's stretching results from the definition of the functions that stretch curves in $B_1$ and $B_2$. So, we give the definition of two stretching functions $E_{be}$ that stretches curves in $B_1$ and $E_{af}$ that stretches curves in $B_2$ as follows. Let $E_{be}$ be the stretching function defined by:

$$E_{be} : B_1 \times [0, h_m] \times [0, v_m] \rightarrow B_1$$

$$(B, h, v) \mapsto E_{be}(B, h, v)$$

The transformation $E_{be}$ stretches curves in $B_1$. The details of its definition are given below:

Let $B_1 = [M_{10}, M_{11}, M_{12}, M_{13}]$ and $B_2 = [M_{20}, M_{21}, M_{22}, M_{23}]$ be two curves in $B_1$. Let $h, v \in [0, h_m] \times [0, v_m]$. $h$ stands for the horizontal stretching amount whereas $v$ is the vertical one.

**Definition 1 ($E_{be}$ transformation).**

$B_2 = E_{be}(B_1, h, v)$ if and only if the control points of $B_2$ are:

$$M_{20} = M_{10} - (h, 0)$$

$$M_{23} = M_{13} - (0, v)$$

$$M_{21} = (1 - c_1) M_{20} + c_1 J$$

$$M_{22} = (1 - c_2) J + c_2 M_{23}$$

With $c_1, c_2 \in [0, 1]$ satisfying:

$$M_{11} = (1 - c_1) M_{10} + c_1 I$$

$$M_{12} = (1 - c_2) I + c_2 M_{13}$$

Where

$$\{I\} = (M_{10} M_{11}) \cap (M_{12} M_{13})$$

and

$$\{J\} = \Delta_1 \cap \Delta_2 \text{ with}$$

$$\Delta_1 : \text{the parallel to } (M_{10} M_{11}) \text{ passing through the point } M_{20},$$

$$\Delta_2 : \text{the parallel to } (M_{12} M_{13}) \text{ passing through the point } M_{23}.$$

An example of a stretching using the function $E_{be}$ is presented in the figure 39.

![Figure 39. Stretching a curve belonging to the set $B_1$ with $E_{be}$ ($E_{be}(B_1) = B_2$)](image)

Now, we have a function that allows stretching a curve belonging to the set $B_1$, what about curves in $B_2$?

Let $E_{af}$ be the stretching function defined as follows:

$$E_{af} : B_2 \times [0, h_m] \times [0, v_m] \rightarrow B_2$$

$$(B, h, v) \mapsto E_{af}(B, h, v)$$

The details of the definition of $E_{af}$ are:

Let $B_1 = [M_{10}, M_{11}, M_{12}, M_{13}]$ and $B_2 = [M_{20}, M_{21}, M_{22}, M_{23}]$ be in $B_2$.

Let $(h, v) \in [0, h_m] \times [0, v_m]$.

**Definition 2 ($E_{af}$ transformation).**

$B_2 = E_{af}(B_1, h, v)$ if and only if the control points of $B_2$ are:

$$M_{20} = M_{10} - (h, v)$$

$$M_{23} = M_{13}$$

$$M_{21} = (1 - c_1) M_{20} + c_1 J$$

$$M_{22} = (1 - c_2) J + c_2 M_{23}$$

With $c_1, c_2 \in [0, 1]$ satisfying:

$$M_{11} = (1 - c_1) M_{10} + c_1 I$$

$$M_{12} = (1 - c_2) I + c_2 M_{13}$$

Where

$$\{I\} = (M_{10} M_{11}) \cap (M_{12} M_{13})$$
\{J\} = \Delta_1 \cap \Delta_2

with

\Delta_1 : \text{is the parallel to } (M_{10} M_{11}) \text{ passing through the point } M_{20},
\Delta_2 : \text{is } (M_{12} M_{13}).

As for the function \(E_{be}\), an example of stretching with the function \(E_{af}\) is given in the figure 40.

Figure 40. Stretching a curve belonging to the set \(B_2\) with \(E_{af}\) (\(E_{af} (B_1) = B_2\))

Now, consider a set of curves \(C\) (see figure 41) to be stretched. More exactly, the stretching will be done with a keshideh modeled with the curves \(B_{11}\) and \(B_{21}\). Consider the following curves:
\(C_1\) : a curve bound to the left of \(B_{11}\) (see figure 41),
\(C_2\) : a curve bound to the right of \(B_{21}\).

Figure 41. initial state of a set of curves \(C\) to be stretched

The stretching or the keshideh with regard to \(h\) horizontally and \(v\) vertically results from the following transformations:

- \(E_{af} (B_{21}, \frac{h}{2}, v)\) to get \(B_{22}\) (see figure 42),

Figure 42. Stretching the curve \(B_{21}\) of \(\frac{h}{2}\) and \(v\)

- \(t_{\hat{u}} (B_{11})\) to get \(B_{11}'\), where \(t_{\hat{u}}\) is a translation of vector \(\hat{u} = (-\frac{h}{2}, 0)\) (see figure 43),

Figure 43. Translation of \(B_{11}\) towards the right-side of \(\frac{h}{2}\)

- \(E_{be} (B_{11}', \frac{h}{2}, v)\) to have \(B_{12}\) (see figure 44)

Figure 44. Stretching the curve \(B_{11}'\) of \(\frac{h}{2}\) and \(v\)

- \(t_{\hat{v}} (C_1)\) to have \(C_1'\), where \(t_{\hat{v}}\) is a translation of vector \(\hat{v} = (-h, 0)\) (see figure 45).

After all these transformations, the set of curves \(C\) gives a stretching as in the figure 46.
In the PostScript encoding, not all these transformations are to be handled explicitly. In some cases, some of them (some translations) are automatically processed with the PostScript interpreter. We remark that before stretching, \(R_{12}, R_{13}\) and \(V_{11}\) are aligned. After the stretching \(R_{22}, R_{23}\) and \(V_{11}\) are aligned too. The angles \((L_{10} P_{12}, L_{10} L_{11})\) and \((L_{20} P_{12}, L_{20} L_{21})\) are equal. The derivatives of degree 0 and 1 are preserved in the points where the keshideh is connected to other curve sets.

4.2 Keshideh: The support in the font

As it has been said before, the keshideh is a small flowing curve that stretches and/or connects Arabic letters. Even tough it is not a character it is supported as parts of the stretched letters. And, as it has also been said before, the stretching can appear inside a letter or between two connected letters (see figure 4). When a word, in a text, is to be stretched, if the last letter of this word
is a stretchable one, the stretching occurs in this letter otherwise the stretching will be in the ligature between
the two last connected letters of the word. The two cases
are presented separately in the following.

4.2.1 Inter-letters keshideh

The keshideh is a juxtaposition of two curves of type 1 and type 2. When the keshideh appears between two
connected letters, the first curve is the horizontal preceding connection below the baseline of the letter in the
left and the second part of the keshideh is the horizontal succeeding connection below the baseline of the letter in
the right. When we designed the font, we parameterized
the horizontal connections exactly on the baseline in a
way such that they can be stretched horizontally to the
right and vertically downwards. So, when the stretching
is null, the connections are simply those exactly on the
baseline, otherwise they are horizontal strictly below the
baseline. From now, we mean by parameterized hori-
tzontal connection exactly on the baseline a connection
that support the connection exactly on the baseline and
below the baseline. For example, when the word طفر
is to be stretched, the stretching occurs between the letters
أ and ﺎ. The letters أ and ﺎ have horizontal connec-
tions exactly on the baseline that are parameterized to
support stretchability. Then, the Stretching consists on
keeping the word as it is and communicating a horizontal
and vertical amounts of stretching as parameters to the
procedures for writing these two letters. Suppose that
we want to stretch the word طفر of an amount $h$
horizontally and $v$ vertically downwards, then, we have to
perform the two actions:

- to stretch the horizontal succeeding connection of
  the letter أ of $\frac{h}{2}$ horizontally and $v$ vertically and
- to stretch the horizontal preceding connection of the
  letter ﺎ of $\frac{h}{2}$ horizontally and $v$ vertically.

In the figure 47, an illustration of a stretching in a font
size equal to 64 points is given. In the first line, the
word طفر with a null stretching is presented. In the last
line, the same word is stretched 12 dp (diacritic point) horizontally and 0.5 dp vertically. In the second line the
letters أ and ﺎ in their initial shapes are showed whereas
in the third line, both of the two letters have a stretching
of 6 dp horizontally and 0.5 dp vertically.

Most of the handbooks of calligraphy [3, 22, 25], teach
that : in the beginning of a word in Arabic letters, the
commonly used horizontal connections are oblique, big
strictly above the baseline and strictly above the base-
line. That's an aesthetic need. When we have to write
the word طفر formed of the letters BEH in initial position
and AIN in the end without stretching, a letter BEH and
AIN with horizontal succeeding connection and prece-
ding connection respectively strictly above the baseline
are used. Let طفر be such a word (the word to the right
on the figure 48), when a stretching is needed, the alter-
natives of these two letters with horizontal connections
below the baseline (they are horizontal connections ex-
actly on the baseline when the amount stretching is null)
are used. In this way, the word طفر would be used. The
two alternatives of the word are in clear on the figure 48.
We can introduce these alternatives in the font to solve the problem of stretchability needs so that the texts processed with the font satisfies the Calligraphic rules. Of course, these alternatives are allowed by the Calligraphic rules and exist. Consider the word \textit{طافر} the letter \( \text{ة} \) admits an horizontal connection after exactly on the baseline because \( \text{ة} \) is a strongly rising letter from the baseline. It is the same for the letter \( \text{ر} \) that admits a preceding connection exactly on the baseline. It is mandatory in a situation such as after the letter \( \text{ب} \) in the word \textit{تبت} or after \( \text{ح} \) inside the word \textit{ته}. We can apply the alternatives of letters also when the ligature is oblique, in the case of a null stretching. But it doesn’t work in some particular situations such that concerning the stretching of the word \textit{تبت}, we may keep the letter \( \text{ت} \) without changes and replace the letter \( \text{ب} \) (see figure 49) by \( \text{ت} \) (see figure 50) and then, the stretching can be performed. With this alternative, the stretching of the word \textit{تبت} has been appreciated by a calligrapher and became good in comparison to the handwritten proof.

\[ \text{ة} \]
points. The shape of the letter QAF  without stretching is quite different from the stretched one, the character  is only a variant of . So, it has the same width as . The vertical downward stretching varies in the range from zero to three diacritic points. The keshideh is performed inside . The part modeling the keshideh is displayed in gray on the figure 52. The letter QAF in final position is in font size 128 points. The control points of the left bottom corner of the nib’s head are given to make it more understandable.

![Figure 52. The keshideh in the letter QAF in final position](image)

In the development of our PostScript font, the horizontal stretching is a value communicated to character procedures as a global variable \( h \). As in [8], at the end of each call of procedure, the value of \( h \) is put back to zero. The value \( h \) is determined by means external to the font, such as the text formatting programs. Then it is communicated to the character procedures in the font. The vertical stretching \( v \) is determined from the character procedures in the font. The vertical stretching \( v \) is communicated to the character procedures in the font. To compute the stretching, the letter \( L \) is used and can be stretched horizontally between zero and three diacritic points. The keshideh is performed inside . The part modeling the keshideh is displayed in gray on the figure 52. The letter QAF in final position is in font size 128 points. The control points of the left bottom corner of the nib’s head are given to make it more understandable.

4.3 The nib’s head motion modeling approach versus other approaches

In this sub-paragraph, the approach for modeling of the nib’s motion is compared in one hand to the Knuth’s approach in Metafont [10] and to the Kinch’s one in MetaFog [28] in the other hand. These two approaches seek the envelope of an ellipse stroking (a nib with an elliptic head) in two different ways. The Knuth’s approach follows the Hobby method [16] to represent the envelope in terms of the raster instead of scalable curves. Kinch solve the problem through representing the envelope in an algebraic and topological way. In our approach, the envelope of the characters static parts is determined using tools outside the PostScript interpreter. A specific program (based on a mathematical idea presented in the sub-paragraph 2.4) helps to determine this envelope. Of course, we also can use Metapost [17], to determine the envelope and cope with other programs to eliminate the fact that some zones are painted twice. The characters static parts have true outlines (there are no overlapping curves), the envelope is generated as with MetaFog.

Concerning the characters dynamic parts, such as the parts of the keshideh, the envelope is determined in a way different from the Knuth’s and Kinch’s approaches. In order to well explain and justify our approach, let us consider a left component of a keshideh (see figure 53).

The curve \( B_1 \) of type 1 in the figure 53 has the control points (169,205), (170,151), (249,134) and (340,134). The nib’s head width and thickness are 31.9643 and 15.9821 points respectively. Here, we have considered a thickness equal to half of the width for clarity.

The points where the slopes are parallel to the vectors defining the nib when the thickness is a sixth of the width are the same. The vectors \( \overrightarrow{u_1} \), \( \overrightarrow{u_2} \) and \( \overrightarrow{u_3} \) are respectively (10.91,77,30.042), (-4.1067,35.4913) and (-15.0244,5.44933). \( B_1'(t) \) is parallel to \( \overrightarrow{u_2} \) at 0.0323 and to \( \overrightarrow{u_3} \) at 0.4923. After applying a stretching with the \( E_{be} \) to \( B_1 \) with an horizontal stretching equal to 109.1601 points and a corresponding vertical one 9.0966 points, we obtain the curve \( B_2 \) with the control points (59.839872,205), (60.968076,144.081482), (190.549225,124.903397) and (340,124,903397). Then \( B_2'(t) \) is parallel to \( \overrightarrow{u_2} \) and \( \overrightarrow{u_3} \) at 0.0224 and 0.3980 respectively. The coefficients where the slopes are parallel to the vectors of the nib’s head change from a stretching state to another. The reason for this is simply that the functions of stretching used to compute the keshideh are not a linear scaling. In order to determine these coefficients and therefore, to be able to determine the envelope as the Kinch’s approach seek through computing these coefficients in the PostScript character procedure. Then, printing the character will be very slow. To avoid such problem, we opt for painting the surface razed with ev-
Figure 53. Identification of points where slopes are parallel to vectors defining the nib’s head.

Every edge of the nib’s head. So, some parts are painted twice or more causing a larger CPU time consuming but it is still less than the time used to compute the outlines of the curve. As we have seen before, some parts of the surface razed with the nib’s head can be lost in the neighborhood of the points where the slopes are parallel to the vectors $\vec{u}_2$ and $\vec{u}_3$. In the figure 54, the real surface razed with the nib’s head according to the curve $B_1$ is showed. Whereas, in the figure 55, we have drawn the surface that is on the figure 54 and then drawn the envelope (in gray) with our adopted technique on it. We remark that we can not distinguish the lost zones (that must be in black). The reason is that the keshideh curves have curvature vectors [6] with small magnitudes (or the radiuses of the osculating circles are big) especially on the points where the derivative is collinear to $\vec{u}_2$ and $\vec{u}_3$. We can accept then this inaccuracy because the results are considered more in their visual aspects. Reeves [32] call this phenomenon "Visual Accuracy". When we consider a nib’s head that respect the ratio existing between the width and the thickness of the nib’s head (one a sixth) the result would be more satisfying visually.

Figure 54. The real razed surface.

Figure 55. The razed surface with our approach.

Figure 56. a) Arabic text lines before justification, in 18 pt, b) Arabic text lines after justification.

5. Conclusion and perspectives

Our main goal consisted on designing a font that helps in producing Arabic documents looking like handwritten proof. So, it was necessary to identify all kinds of ligatures and to introduce keshideh in an adequate way. We were brought to scan the handwritten proofs and then to work on these proofs with simple graphic tools.
like Kontour [19] under Linux [24], in order to get the encoding of the letters in terms of PostScript Bézier’s curves. The existing font tools were not enough to take into account the motion of the nib’s head, the metrics in diacritic point, the possibility to correct some errors allowed for the calligrapher in handwritten proofs etc. We worked almost manually to design the PostScript encoding of some characters that represent most of the cases and we got so a mini-font. This font has been used to give the example in the figure 56. This is a set of lines (with no meaning in Arabic) to assess justification through keshideh. The requirements, support and development of a tool to assist in designing Arabic fonts should be a separate work. In any way, there will be no satisfying compromise to respect the rules of Arabic calligraphy unless a geometrical model for all the characters and symbols is built.

References


Abdelouahad BAYAR is professor of mathematics and computer science at Cadi Ayyad University (école Supérieure de Technologie) in Safi, Morocco since 1996. He gets a CEUS (Certificat d’Études Universitaire Supérieures) in applied mathematics and computer science in 1996. He integrates then the university as a research-teacher. He holds his Doctorat of the third cycle in 1999 in Computer science in communication networks. Since 2007, Bayar is a member of the group of Information Systems and Communication Networks LSNC in Information Systems Ingeniering Laboratory ISIL. His actual investigations deal especially with dynamic fonts and calligraphic rules formalization.

Khalid SAMI is professor of mathematics at Cadi Ayyad university (Faculté des Sciences) in Marrakech, Morocco since 1983. Since years, he’s interested in multilingual scientific text formatting. Some of the work relevant to this area is presented in the url www.ucam.ac.ma/fssm/rydarab.