Implementation of Automatic Font Generation

Lakshmi M. Gadhikar, Ajinkya Shukla
Information Technology Department
Fr.CRIT, Vashi, Navi Mumbai

Abstract

Persons handwriting reflects his identity in many cases. However, due to wide use and easy access to electronic media most of the document exchange now-a-days is in the form of typewritten or computerised documents which mask person's handwriting. Sending and receiving emails or SMS or chatting in one's own personalized handwriting or communication through social networking would indeed be a great experience. We have developed an Automatic Font Generation System which can be used to create a font style of an individual's handwriting. The Automatic Font Generation System takes a scanned copy of user's handwriting and builds a model of his/her handwriting for most of the symbols used generally in writing like alphabets , numbers , punctuation characters , special symbols etc. The system then creates a standard font file that can be installed on a computer system and used like any other standard font style.

1. Introduction

Automatic Font Generation system takes as input a scanned hand written document and generates a font out of it. Unlike font editor application in which person has to sketch the font our system automatically generates a replica of hand written text.

1.1 Font [1]

A computer font is an electronic data file containing a set of glyphs, characters.

A glyph represents an individual letter, number, punctuation mark, or other symbol. Collection of glyphs is called as typeface.

A font family is typically a group of related fonts which vary only in weight, orientation, width, etc., but not design. For example, Arial is a font family, whereas Arial Narrow, ArialCondensed and Arial Rounded are individual fonts making up the Arial family. [1]

1.2 Types of Fonts [2]

As discussed in our previous work [2] there are various types of fonts like bitmap font that stores each glyph as an array of pixels i.e. a bitmap , outline font or vector fonts which are collections of vector images, i.e. a set of lines and curves to define the border of glyphs. Outline fonts come in various formats like Type 1 and Type 3 fonts , TrueType font, Open Type font, Metafont etc

Bitmap fonts are faster and easier to use in computer code, but non-scalable, requiring a separate font for each size. Outline and stroke fonts can be resized using a single font and substituting different measurements for components of each glyph, but are somewhat more complicated to render on screen than bitmap fonts, as they require additional computer code to render the outline to a bitmap for display on screen or in print. Although all types are still in use, most fonts seen and used on computers are outline fonts. A raster image can be displayed in a different size only with some distortion, but renders quickly; outline or stroke image formats are resizable but take more time to render as pixels must be drawn from scratch each time they are displayed. In this work, we generate outline font since they can simulate the handwriting with a high degree of realism.

2. Implementation Details

Automatic Font Generation Software comprises of four modules i.e. Contour Tracing Module, Polygon Tracing Module, Curve Smoothing module and finally the .ttf font generation module. The input to the software is a handwritten template and the output of the software is a .ttf font file.
The process of generating font from user"s handwriting comprises of following steps :
1. The user fills the template in his/her handwriting

2. The template is then scanned at minimum 300 dpi resolution

3. The scanned template is given as an input to automatic font generation software

4. The automatic font generation software thresholds the template to remove noise

5. The template is then segmented into each of 96 characters

6. The contour tracing module then traces the boundary pixels of each character

7. The polygon tracing module joins the pixels traced by contour tracing module

8. The curve smoothing module smoothens the sharp edges of the polygon formed by polygon tracing module

9. Finally the .ttf font generation module generates the .ttf font file
The complete process in explained in detail below.

2.1 Filling the template in user's handwriting

A template is a pre-developed page layout. The template is used to inform the machine i.e. computer at which position each of the character is located. For example the character that will appear in the first top-left block will be a blankspace character, the next character will be an exclamation character. It is necessary to inform the computer position of each character because computer is a dumb machine and it does not understand what is a particular shape called unless it is instructed.
The template for automatic font generation software is developed for an A4 size paper. It consists of 8 columns and 12 rows which constitutes 96 blocks. The width and height of each block is exactly identical
The user has to fill the template i.e. 96 characters in the sequence defined on page no 31. The template has a red line called as base line. The baseline is used for horizontal alignment of handwriting. The user should write each of the character in the respective block. The user should take care that the written character does not overflow the block. Figure 1 shows the blank template.

The user can fill in the blank template in his/her own handwriting in the order shown in figure 2. The handwritten document should consists all the characters including capital letters, small letters, special characters, numbers and space. Each character is called a glyph. The document should be well written with a thick pointed pen.

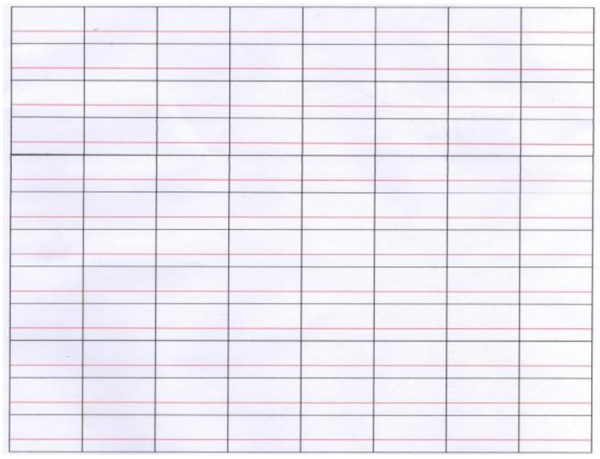


Figure 1. Blank A4 page size template

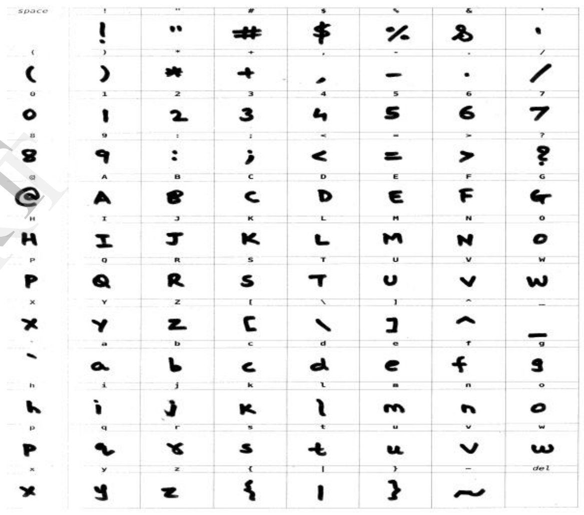


Figure 2 : Handwritten Template [2]

2.2 Scanning the handwritten template

Once the template has been filled in user‟s hand writing the template is scanned at 300 dpi resolution. If the resolution of scanning is high, the size of the scanned template increases and as a result the processing speed of template becomes slow especially while removing the noise. The template shown in Figure 1 was scanned at 300 dpi resolution and it resulted in an image size of approximately 1 to 1.5 MB. When the template was scanned at resolution of 2400 or 4800 dpi the size of scanned template exceeded 10 MB. Thus it is recommended to scan the template at 300 dpi resolution.

2.3 Input the scanned template

The scanned template is provided as an input to the automatic font generation system.

2.4 Threshold the scanned template

After the image has been read by the automatic font generation system, thresholding is used to remove noise from the image. Automatic Font generation system used inverted binary thresholding to remove the red baseline from the scanned template. Inverted Binary thresholding is the opposite of Binary thresholding where the pixel value is set to 0 if it is greater than threshold value. Thus this thresholding operation can be expressed as follows. If the intensity of the pixel src(x,y) is higher than threshold, then the new pixel intensity is set to 0 . Otherwise it is set to maximum value.

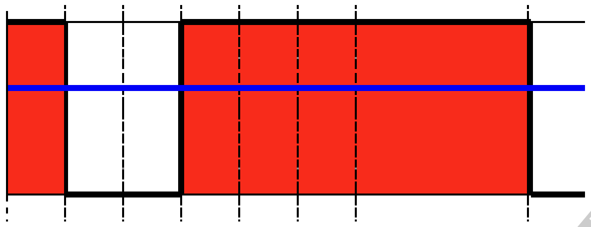


Figure 3. Inverted Binary Thresholding

2.5 Segment the scanned template

Once the scanned template is noise free it is segmented into 96 images each corresponding to a character in the template as shown in figure 4.

2.6 Contour Tracing Module

Once the scanned template is segmented into 96 images, the boundary pixels of each of the character are traced using Moore-Neighbor tracing algorithm [3].

2.7 The polygon tracing module

Once the boundary pixels are traced by contour tracing module the pixels are joined using polygon tracing module.

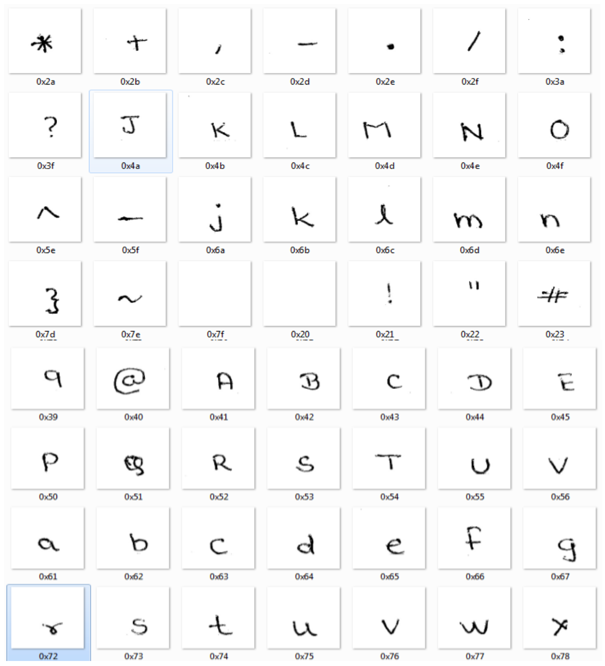


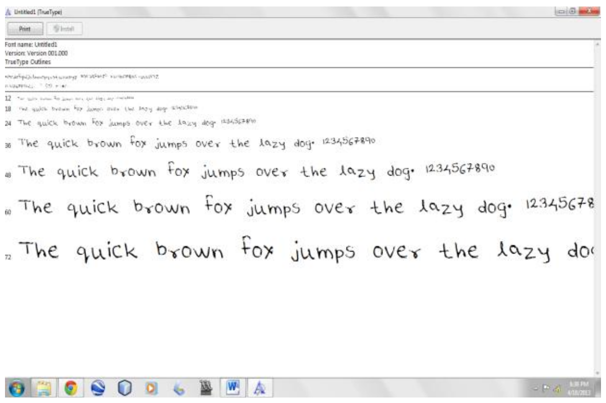
Figure 4. Segmented and noise free image

2.8 The curve smoothing module

The polygon obtained by polygon tracing module has sharp edges. This polygon with sharp edges is given as input to a curve smoothening module which smoothens the edges using Bezier curves.

2.9 .ttf font generation module

The .ttf font file generated is as follows.



Font name: Untitled1
Version: Version 001.000
TrueType Outlines

```
12 [illegible]
18 The quick brown fox jumps over the lazy dog 1234567890
24 The quick brown fox jumps over the lazy dog 1234567890
36 The quick brown fox jumps over the lazy dog. 1234567890
48 The quick brown fox jumps over the lazy dog. 1234567890
60 The quick brown fox jumps over the lazy dog. 12345678
72 The quick brown fox jumps over the lazy do
```

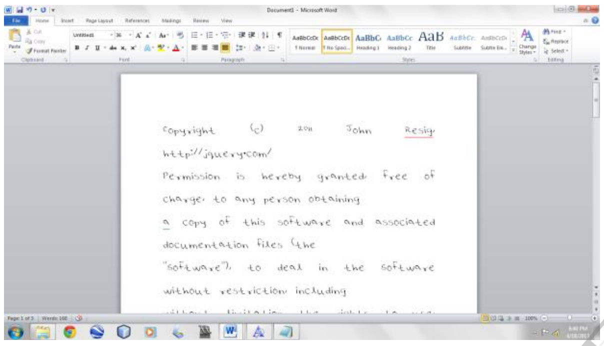
Figure 5. .ttf Font File

2.10 Installation and use of Font File

The .ttf font file is installed on machine by clicking on Install button in top left area of the font file. The font file gets added to the already existing system fonts. On a Windows operating system, the fonts can be viewed under C:\Windows\Fonts. Once the font is installed on machine it can be used in any text editor

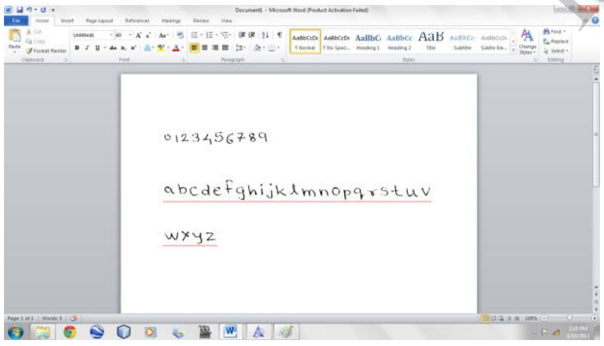
3. Results

The use of the above generated font in MS Word 2007 is as follows. Select the font in the font drop down box and type in text, number or any special character.



Copyright (c) 2011 John Resig
http://jquery.com/
Permission is hereby granted free of
charge, to any person obtaining
a copy of this software and associated
documentation files (the
"software"), to deal in the software
without restriction including

Figure 6. Font use in MS Word

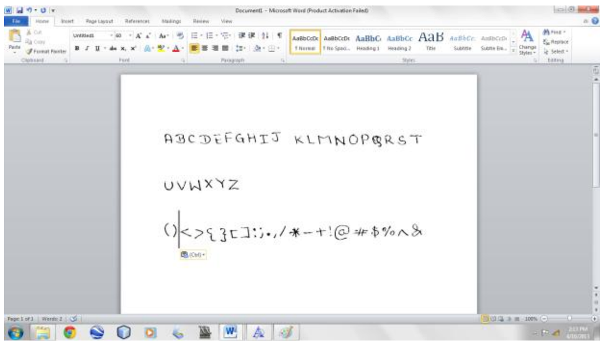


0123456789

abcdefghijklmnopqrstuv

wxyz

Figure 7. Numbers and lowercase alphabets in mystyle font in MsWord

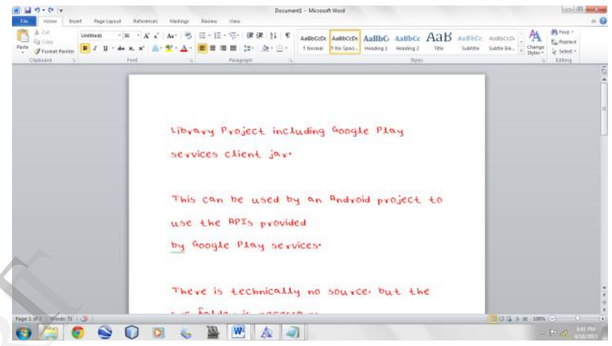


ABCDEFGHIJ KLMNOPQRST

UVWXYZ

()<>{}[]:;.,/*-+!@#$%^&

Figure 8. Uppercase and Special characters in mystyle font in MsWord

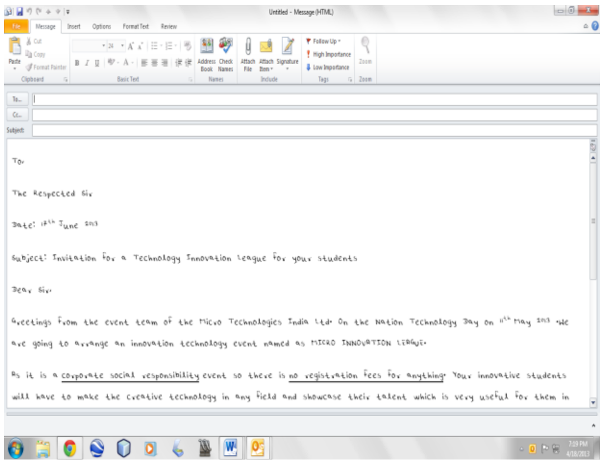


Library Project including Google Play
services client jar.

This can be used by an Android project to
use the APIs provided
by Google Play services.

There is technically no source, but the

Figure 9. Bold and Colored mystyle font in MsWord



To,

The Respected Sir

Date: 17th June 2013

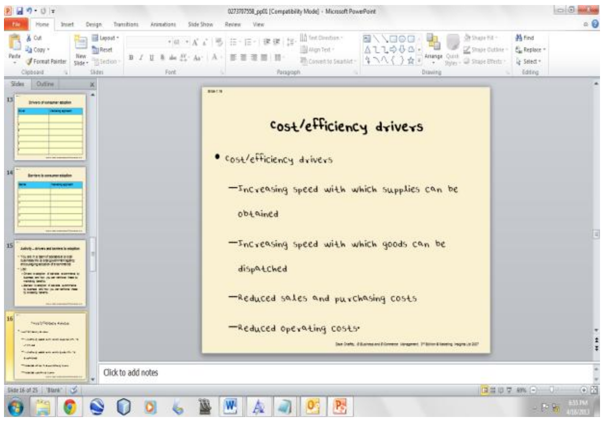
Subject: Invitation for a Technology Innovation League for your students

Dear Sir,

Greetings from the event team of the Micro Technologies India Ltd. On the Nation Technology Day on 11th May 2013 we are going to arrange an innovation technology event named as MICRO INNOVATION LEAGUE.

As it is a corporate social responsibility event so there is no registration fees for anything. Your innovative students will have to make the creative technology in any field and showcase their talent which is very useful for them in

Figure 10. Use of font for email application



cost/efficiency drivers

- cost/efficiency drivers
  - Increasing speed with which supplies can be obtained
  - Increasing speed with which goods can be dispatched
  - Reduced sales and purchasing costs
  - Reduced operating costs

Figure 11. Use of font in Ms PowerPoint

4. Related work

Luc Devroye and Sandro Mazzucato[4] have presented a method for creating a PostScript type one Bezier outline font from a scanned bitmap of all characters. The output consists of a type one PostScript font file and corresponding AFM and TFM files with full sets of kerning pairs. They faced following technical difficulties. Characters drawn with a thin pen are often just a few pixels wide, and are very sensitive to variations in thickness. Characters with highly irregular contours, such as from old typewriters require more Bezier sections for faithful reproductions. Characters with many intersections require crisp rendering where strokes cross.

Koichi Itoh and Yoshio Ohno [5] presented an algorithm that automatically generates outline fonts from a grey-level image of a character obtained by a scanner. The algorithm begins by extracting contour points from the image and dividing the points into a number of segments at the corner points. The next step is fitting a piecewise cubic Bezier curve to each segment. To fit cubic Bezier curves to segments, they use least-squares fitting, without fixing the end points of the curves. They locate the end points by computing the intersection of the adjoining curves. This algorithm greatly improves the shape of the corner of the outline fonts. However, this method is especially suitable to Gothic (in the Japanese sense) or Kaisho style (a font based on calligraphic characters but which has a very stiff impression).

High-Logic Font Creator [6] and FastFont [7] are similar to our automatic font generation system but are proprietary.

Our System i.e. Automatic Font Generation creates a font style of an individual's handwriting with minimal effort ,cost and time . It generates a .TTF font file format that can be installed on any machine and it achieves high degree of realism.

5. Conclusion

Automatic Font Generation system generates a font out of user's own handwriting. The user's handwritten document is scanned and a standard font file is generated from it. This font file can then be installed on any compatible computer. The user can thus type in his/her handwriting in any text editor or applications like emails, chat, IM (Instant Messaging) etc.

6. References

[1]http://www.adobe.com/devnet/opentype/afdko.html
[2] Lakshmi Gadhikar , Ajinkya Shukla , Rohan Wani , Sonal Mumbaikar , Monica Ahuja, " Automatic Font Generation" , International Journal on Multidisciplinary Research , Volume 1 : Issue 12(VI), March 2013
[3] P.Rajashekar Reddy, V.Amarnadh, Mekala Bhaskar "Evaluation of Stopping Criterion in Contour Tracing Algorithms" , IJCSIT- International Journal of Computer Science and Information Technologies, Vol. 3 (3), 2012, 3888-3894.
[4] Luc Devroye , Sandro Mazzucato, School of Computer Science, McGill University, Montreal, Canada, "Random search in Automatic font generation" , hhttp://luc.devroye.org/mazzucato.pdf
[5] Koichi itoh1 and yoshio ohno , "A Curve Fitting Algorithm For Character Fonts" , Electronic Publishing, Vol. 6(3), 195–205 (September 1993)
[6] http://fontforge.org/
[7]http://www.high-logic.com/fontgenerator/ scanahand.html.