A SURVEY OF FONTS AND ENCODINGS FOR INDIAN LANGUAGE SCRIPTS

Dipali B. Choudhary Sagar A. Tamhane Rushikesh K. Joshi
Department of Computer Science and Engineering,
Indian Institute of Technology Bombay
Powai, Mumbai-400076.
Email:{dipali, sagar, rkj}@cse.iitb.ac.in

Abstract
In this paper, some of the existing fonts and font encoding techniques for Indian languages are surveyed. A classification of fonts based on their properties and technologies used is presented. A survey of the tools and techniques for font creation, editing, font display and Indian language text input is also presented. We also list some of the existing products for Indian languages.
Key Words: *Fonts, Indian languages, Encodings, Tools, Input Technologies*

1 Introduction

Today, almost all of the software systems in India are written in English language. For the past two decades, efforts are being carried out to enable Indian languages on computer systems. While the language of software development in India primarily remains to be English, document development systems have been enabled with Indian languages with the help of fonts and encodings designed for Indian languages. Many input and output technologies have been developed for Indian languages on different platforms. The area of localizing software and desktop and Internet publishing is currently very active. Hence this survey paper has been felt timely. In the next section, we present a classification of fonts based on various criteria, and subsequently describe each class in Sections 3 to 8. The classification of fonts is based on features such as their internal representation, installation method, width of glyphs, physical existence, font encoding and font file readability. Section 9 surveys existing encoding techniques like ASCII, LATIN, ISCII, TSCII, ISFOC and Unicode. We present the input technologies used for Indian languages in Section 10. A survey of available tools for Indian languages is given in Section 11.

2 A Font Classification System

We now present a classification of fonts based on their different properties. According to internal representation, fonts can be categorized into Bitmap fonts and Vector fonts. In Bitmap fonts, glyphs are represented as matrices of dots whereas in Vector fonts, glyphs are represented as curves. Fonts can be classified into static and dynamic fonts as per the installation method. Dynamic fonts are automatically downloaded per viewing, whereas static fonts are always available on the local machine. According to width of glyphs, fonts can be classified into fixed width and variable width. Fixed width fonts are typically used for consoles, while most desktop publishing applications use variable width fonts. Fonts can be classified into physical fonts and logical fonts according to physical existence. Logical fonts are used for portability. According to font encodings, fonts are classified into 8-bit fonts and 16-bit fonts. These are the bits required to identify the position of a character (glyph) in the alphabet. In addition to this, fonts can be classified into ASCII fonts and Binary fonts according to the human readability of the font file. Some fonts are editable as ASCII files. Figure 1 captures this classification tree of fonts. Each of these classes are described below in detail.

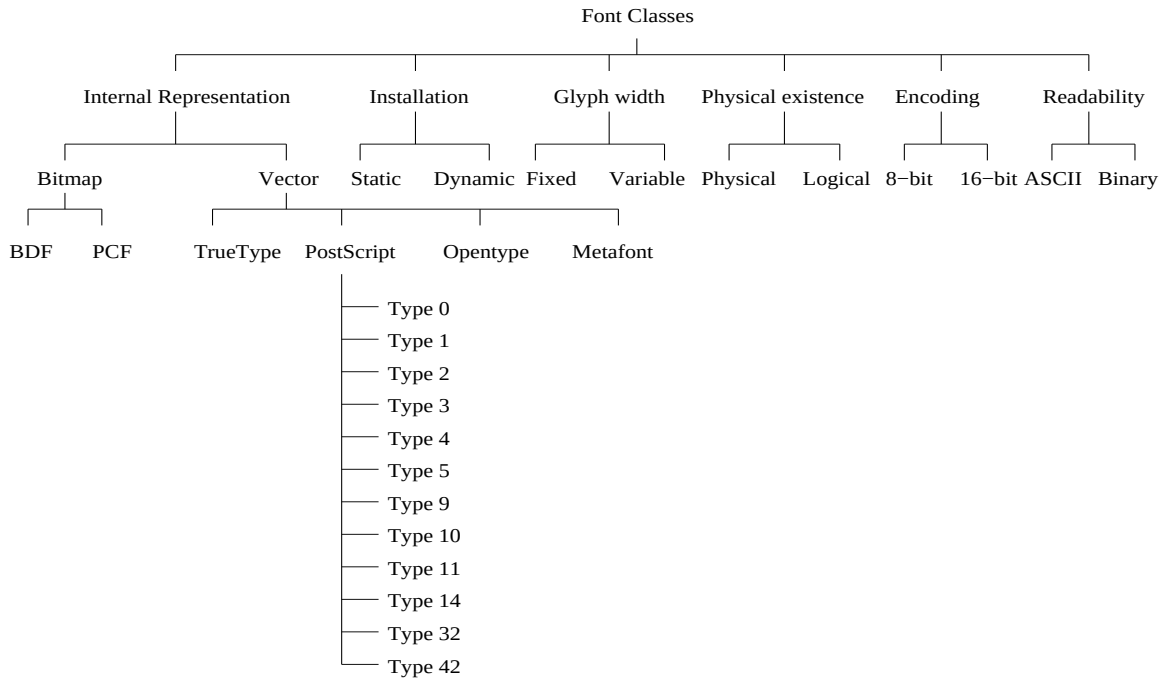

Figure 1: The Font Classification tree

3 Classification According to Internal Representation

The shape of the glyph can be stored in the form of a bitmap or in the form of vectors resulting in Bitmap and Vector fonts. Bitmap fonts are rendered faster than Vector fonts. However, Vector fonts do not cost extra storage for scaling to higher dots per inch (dpi) unlike Bitmap fonts.

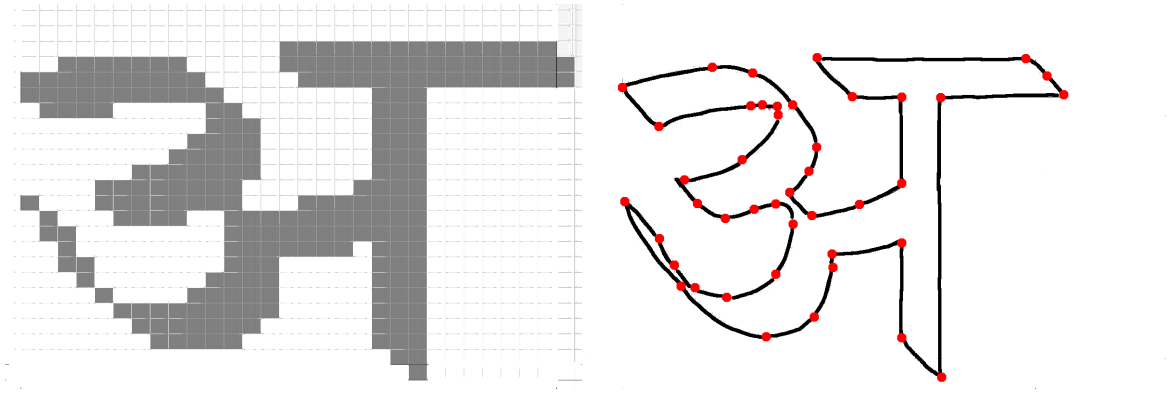

Figure 2: (a) A Bitmap font glyph (b) A Vector font glyph

3.1 Bitmap Fonts (BDF and PCF)

Glyphs in bitmap fonts are represented as matrices of dots, which limits them to a particular resolution. For example, we may have bitmap fonts for point sizes such as 10, 12, 24, etc. Bitmap fonts are further divided into two types based on the font file format, *Bitmap Distribution Format (BDF)* and *Portable Compiled Font (PCF)*. Typically, font files contain details about glyphs and metadata about the font files. BDF [1] files are in an ASCII encoded, human-readable form. They are composed of meta-information in the header followed by a series of hexadecimal encoding of the glyphs in the font. PCF Format is a binary storage format for bitmap fonts files [2]. The binary font file contains all the information about a font such that it is compact and efficient for machine readability.

3.2 Vector Fonts (Outline Fonts)

Glyphs in Vector fonts are described by one or more curves. These curves are described using Bezier curves. Each Bezier curve has 'on-curve' and 'off-curve' control points. These Bezier curves can be of either quadratic or cubic type. Vector fonts are scaleable. These are classified into TrueType [3], Postscript [4], OpenType [5] and Metafont [6] categories. Truetype fonts use quadratic curves, whereas, PostScript fonts use cubic curves. OpenType font is an extension of the TrueType font, which can contain either only TrueType or only Type 1 data or both. Metafont supports creation of family of fonts through parameterization. In addition to shape details, a Vector font also consists

of metric data such as character width and kern pairs [7]. A kern pair is a pair of glyphs for which spatial adjustment is required when they occur in sequence.

3.2.1 TrueType Fonts

TrueType is a font technology designed by Apple Computer Inc. It uses quadratic Bezier curves [8] to represent the outlines of the glyphs. TrueType fonts store the metric and shape information in a single file with a *.ttf* extension. Examples of TrueType fonts are *XDVNG*, *Shusha*, *DV-TTYogesh* and *Subak*

3.2.2 PostScript Fonts

PostScript font technology was developed by Adobe Systems. These fonts use cubic Bezier curves [9] to represent outlines of the glyphs. PostScript fonts can be further classified into Type 0, Type 1, Type 2, Type 3, Type 4, Type 5, Type 9, Type 10, Type 11, Type 14, Type 32 and Type 42 fonts [10]. PostScript fonts use two separate files for containing character outlines and metric data. For example, outlines may be stored as Printer Font Binary (pfb) or Printer Font ASCII (pfa). Examples of metric data files are Printer Font Metrics (pfm), Adobe Font Metrics (afm), Information files (inf) and Multiple Master Metric (mmm).

3.2.3 OpenType Fonts

OpenType font technology was developed by Adobe and Microsoft. These fonts are based on Unicode. They allow wide range of characters through a set of glyphs with the help of mapping between characters and glyphs. This mapping supports conjuncts such as क्व, forms requiring positioning such as in के and कि, alternatives such as in आर्य and आऱ्य and substitutions such as क + ् + श = क्ष. Positioning of glyphs can be in two dimensions. Open type fonts may have extension *.otf* or *.ttf*.

3.2.4 Metafont

Metafont is part of Donald Knuth's Tex system. They are parameterized and can generate different font families. Scaling is graceful due to their way of nonlinear scaling (different parameters scale differently). For example, at 10 point and 20 point, shapes may be different. These fonts are not preferred for WYSIWYG publishing due to slower rendering. Metafont files have a *.mf* extension. Example of Metafont is the *Computer Modern* Metafont.

4 Classification According to Installation

Fonts can be classified into two categories, *static* and *dynamic*, based on the way they are installed on a computer. Static fonts are manually installed. If the user visits a webpage that has a font not installed on his system, the browser typically uses some default font to

render the text. Text editors allow text to be composed with only statically installed fonts. The dynamic font technology [11] is created by Netscape and Bitstream. A dynamic font on a web server hosting a page is automatically downloaded by the browser and installed temporarily for viewing. They take download time overheads. Dynamic fonts are of two types, Embedded OpenType (eot) and Portable Font Resource (pfr). Unicode supported eot fonts are supported by both IE and Netscape while pfr fonts are only supported by Netscape.

5 Classification According to Width of Glyphs

Fonts can be classified as *fixed width* and *variable width* fonts [12] [13]. Each character in the fixed width font has same width. Fixed width fonts are used for console and to represent computer programs. An example of fixed font is *courier*. Most of commonly used fonts are of variable width fonts. Fonts for Indian languages are of variable width type since many of the words have conjuncts.

6 Classification According to Physical Existence

Fonts can be classified into *physical* and *logical* fonts to take into account portability issues [14]. Physical fonts are the actual fonts libraries which can be installed on a computer system. Examples of physical fonts are *Times New Roman*, *Helvetica*, *Courier New*. JAVA 1.0 supports Logical fonts called Serif, Sans serif, Monospaced, Dialog, and DialogInput. These logical fonts are not actual font libraries. They are names recognized by the Java runtime and must be mapped to existing physical fonts installed on the system.

7 Classification According to Font Encoding

Fonts are classified into two categories of *8-bit* and *16-bit* based on the number of bits required by the font to store encoding of each glyph. Each 8-bit font contains maximum 256 characters. The same character number can represent a different character in different encoding schemes. For example, Indian language fonts of this type are *DV-TTYogesh, SUBAK* and *Shusha*. A 16-bit font can represent 65536 characters. An example of 16-bit encoding is that of Unicode which contain all major languages in the world. Examples of 16-bit fonts supporting Indian languages are *Arial Unicode MS, Code2000, Mangal, Raghu8, TITUS Cyberbit Basic, Devanagari MT* for Macintosh, *Devanagari MTS* for Macintosh, *Sibal Devanagari* for Linux.

8 Classification According to Human Readability of Font File

Fonts can be classified into *ASCII* and *Binary*. The ASCII font files are in English format. They can be edited easily with an editor. Examples of ASCII format fonts are BDF fonts

and PostScript Font ASCII (.pfa). Binary font files require special tools for viewing and editing. PostScript printers generally contain binary fonts in .pfb format in ROM or on disk. Examples of binary format fonts are TrueType Fonts and PostScript Font binary (.pfb).

9 Encoding Techniques

Character encoding is a method of converting a sequence of bytes into a sequence of characters. The standard character sets for Indian languages are, ASCII, Latin, ISCII and Unicode. We explain each of these encodings below.

9.1 ASCII and Extended ASCII

American Standard Code for Information Interchange (ASCII) is a 7-bit scheme allowing for 128 character positions. Extended ASCII uses 8-bits to represent 256 characters. All of the 8-bit Indian language fonts use extended ASCII as the encoding for the Indian language glyphs.

9.2 LATIN

Latin-1 (ISO-8859-1) is a version of extended ASCII. It is applied mainly for English and Western European languages. However, ISO-8859 standardizes character sets for other scripts also. The family includes Latin1 (Western European), Latin2 (East European), Latin3 (South European), Latin4 (North European), Latin-Cyrillic, Latin-Arabic, Latin-Greek, Latin-Hebrew, Latin5 (Turkish) and Latin6 (Nordic).

9.3 ISCII

Department of Electronics (DoE) published ISCII (Indian Script Code for Information Interchange) as a standard in 1983 for Indic scripts based on their common phonetic structure [15]. ISCII is an ISO 8-bit code standard, which contains ASCII character set in the lower half, and the Indic script character sets in the upper half. Two more systems based on ISCII have also been approved: ACII (Alphabetic Code for Information Interchange) and ISFOC (Intelligence based Script Font Code).

9.4 TSCII

Tamil Standard Code For Information Interchange(TSCII) [16] was proposed by Internet Tamil Community in 1998. In ISCII and Unicode the encoding schemes for all the Indic languages use Devanagari script as the reference language, but phonology and script usage of Tamil (Dravidian) language is different. Hence the need for a separate standard for Tamil arises. TSCII uses 8-bit bilingual scheme with ASCII character set in lower half. Standards for Tamil Computing Group (STC) has developed many fonts, keyboard drivers, editor and converters using this encoding.

9.5 ISFOC

Intelligence Based Script Font Code (ISFOC) [17] is a standard proposed by C-DAC. These code-charts are different for different scripts and are represented by 8 bits. It contains basic shapes required for rendering a script. These shapes can be overlapped to compose any word in the font.

9.6 Unicode

Unicode [18] is based on 16-bit encoding that permits 65,535 characters. Unicode assigns the value U+nnnn to a character where nnnn is a four-digit number in hexadecimal notation referred to as a code point. Unicode encodes text by script, not by language. This avoids duplication of letters. For instance, a letter in Devanagari script could be used in Sanskrit, Hindi or Marathi. Unicode has reserved positions for Devanagari (range: 0900 - 097F), Bengali (range: 0980 - 09FF), Gurmukhi (range: 0A00 - 0A7F), Gujarati (range: 0A80 - 0AFF), Oriya (range: 0B00 - B7F), Tamil (range: 0B80 - 0BFF), Telugu (range: 0C00 - 0C7F), Kannada (range: 0C80 - 0CFF) and Malayalam (range: 0D00 - 0D7F) scripts.

10 Keyboard Input Techniques

Another important aspect in enabling of Indian languages over computer systems is the method for inputting Indian language text. Using keyboard for input is the most common way of inputting text. Keyboard input method can be classified into Inscript, English Phonetic and Typewriter layouts based on the mapping of actual keys on the keyboard to Indian language characters. For example, in the Devanagari typewriter layout, the character 'G' of the keyboard maps to ळ , while in Inscript layout it maps to उ.

10.1 Inscript:

In the Inscript keyboard layout, all the vowels are placed on the left side of the keyboard layout and the consonants on the right side.

10.2 English Phonetic:

This layout uses transliteration between languages. Transliteration refers to the written representation of one language by another. An example of transliterator is Jtrans [19], which is a script for transliteration from English to Devanagari. When the word 'tukArAm' is typed, the corresponding word in Devanagari, तुकाराम is shown.

10.3 Typewriter Layout:

This layout is similar to the corresponding layout of the Indian language typewriter and hence is useful for typists and other people familiar with that typewriter layout.

11 Tools

In this Section, we enlist various font related tools such as font creation and editing tools, font display engines and terminal emulators.

11.1 Font Creation and Editing Tools

There are many font creation and editing tools out of which some are commercial whereas some are freeware or shareware. We have listed some tools and tabulated their properties in Table 1.

Software	Creator Details	Operating System	Commercial or Shareware or Freeware	Open Source	Type of the font supported
Font Creater	High-Logic http://www.high-logic.com/fcp.html	Windows	Shareware	No	.ttf
Fontographer	Macromedia Inc. http://www.macromedia.com/software/fontographer/	Windows, Macintosh	Commercial	No	Type1, Type2, .ttf
Softy	Dave Emmett http://users.iclway.co.uk/l.emmett/	Windows	Shareware	No	.ttf, .bdf
Type Tool	FontLab Ltd http://www.pyrus.com/html/typetool.html	Windows, Macintosh	Commercial	No	Type1, Type2
Web Embedding Tool (WEFT)	Microsoft Corporation http://www.microsoft.com/typography	Windows	Freeware	No	.ttf
Web Font Maker	Bitstream Inc. http://www.bitstream.com	Windows	Freeware	No	.ttf
BDF Font Editor	Thomas A. Fine http://hea-www.harvard.edu/~fine/Tech/bdfedit.html	Linux	Freeware	Yes	.bdf
GOTE	Robert Brady http://gote.sourceforge.net/	Linux (POSIX)	Freeware	Yes	.ttf
PfaEdit	George Williams http://pfaedit.sourceforge.net	Linux	Freeware	Yes	.ttf, .bdf, .pfa, .pfb, .bin, .otf
xmbdfed	Computing Research Laboratory, http://crl.nmsu.edu/~mleisher/xmbdfed.html	Linux	Freeware	Yes	.bdf

Table 1: Comparision between font editing tools

11.2 Font Servers

A font server is a background process which makes fonts available to XFree86. Font servers are used on Linux machines. An advantage of font servers is that they can send fonts to remote displays. There are three font servers: xfs, xfs-xtt and xfstt. All of these font servers can provide TrueType font rendering. Currently xfs supports Type1, Truetype and bitmap font formats. Also, modification of the font path is made easier with the utility chkfontpath. If xfs is not installed then FreeType (TrueType font handler) and Type1 (Adobe Type 1 font handler) modules are required. XFree86 has two different TrueType font rendering engines, FreeType engine and X-TrueType (xtt) engine. The FreeType (formerly xfsft) is based on the FreeType Library. xfstt can handle TrueType fonts. It cannot handle X bitmap, Type1, CJKV and so on.

11.3 Terminal Emulators for X

Terminal emulators are programs that allow a computer to act like a terminal. We look at two most common emulators:

1. xterm: xterm [20] is the most commonly used terminal emulator for X, which supports input and output of English text. xterm emulates VT100 and Tektronix4014 terminals. The character encoding used for xterm is ASCII. On xterm Indian text characters get widely separated because xterm can handle only fixed width characters. It takes the maximum width of the character for displaying all characters.

2. Kterm: Kterm [21] program is a multilingual terminal which emulates VT102 and Tektronix4014. It has support for input-output of Chinese, Japanese, and Korean text.

11.4 Glyph processing on Windows

Windows displays .ttf fonts using TrueType Font Technology. This technology consists of two parts, the TrueType font files and the TrueType Rasterizer [22]. The job of the TrueType Rasterizer is to generate character bitmaps for screens and printers It accomplishes this by performing the following tasks:

1. Reading the outline description of the character from the TrueType font file.

2. Scaling the outline description of the character to the requested size and device resolution.

3. Adjusting the outline description to the pixel grid (based on hinting information).

4. Filling the adjusted outline with pixels (scan conversion).

For displaying Unicode Fonts, Windows operating system provides Uniscribe rendering engine with services [23] like OpenType Layout Services Library. The Unicode Script Processor (USP10.DLL), also referred to as Uniscribe, is new to Windows 2000. Uniscribe is a collection of APIs that enable a text-layout client to format complex scripts. It

supports complex rules found in scripts such as Arabic, Indian and Thai. Uniscribe also handles scripts written from right-to-left, such as Arabic and Hebrew. RichEdit is a higher-level collection of interfaces that may be used to call Uniscribe or other shaping engines or routines. RichEdit hides the complexities of certain scripts from the text layout clients. The OpenType Layout Services Library (OTL Services) is a set of text processing helper functions. This library is used for reading information from font files. The text layout functions in this library can be used to perform glyph substitution and glyph positioning.

11.5 An Additional List of Tools

1. Text Processors: Hindi Word Processor (HWP), Unitype Global Office (http://www.krishnasoft.com/hwp.htm), Global Writer (http://www.unitype.com/unitype.htm), AKSHARA (http://www.LanguageTechnologies.ac.in), Akruti (http://www.akruti.com), Aksharamala (http://aksharamala.com/about/), Yudit (http://www.yudit.org/), IWrite32 (http://members.tripod.com/~sbiswas/IWrite32/IWrite32.html), AATHAMI and THIRU (http://www.geocities.com/Athens/Acropolis/1427/index.htm), WILIO (http://www.languagetechnologies.ac.in/wilio/wiliomain.html).

2. Transliteration: Acharya (http://acharya.iitm.ac.in), Jtrans (http://www.sibal.com/sandeep/jtrans/), Baraha (http://www.baraha.com/baraha2000.htm).

3. Text Viewer: TAMILVU (http://www.geocities.com/Athens/Acropolis/1427/ index.html)

4. Character Recognisation: A Devanagari Pen-written Character Recognition system [24], Writing Pad for Indian Languages On-Line Character Recognition System [25].

5. Mailing System: http://www.epatra.com, http://www.mailjol.com, INDOMAIL (http:// www.lastech.com).

6. Localization of Linux: Apple's Indian Language Kit (http://www.apple.co.jp/datasheet/software/ilk.html), Indian Pango (Available at: http://indlinux.org/downloads/downloads.php), Indix (Available at: http://rohini.ncst.ernet.in/indix), iterm (Available at: http://www.cse.iitk.ac.in/~moona/isciig/iterm/main.html)

7. Teaching Tool: HindiGuru (http://specialitysoftware.com/hguru/)

8. Other Tools: Arthalekha (http://www.modular-infotech.com/html/arthalekha.htm), Itrans (http://www.aczone.com/itrans/), Bangtex (http://tnp.saha.ernet.in/~pbpal/bangtex/bangtex.html), Dishaa (http://velankar.hypermart.net/dismainl.html)

12 Summary

In this paper we have surveyed fonts and font technologies with the objective of summarizing the developments in this area of enabling Indian languages over the computer

systems. We have presented a classification of fonts based on their properties. Many standards like ISCII and Unicode have been developed by organisations to facilitate multilingual documents. Some of the tools developed to create Indian language documents have been listed in section 11.
Note: Any suggestions or additional information can be directly mailed to the authors.

13 References

[1] Adobe System Incorporated, Bitmap Distribution Format Version 2.2, 1993. Available at: http://partners.adobe.com/asn/developer/pdfs/tn/5005.BDF_Spec.pdf
[2] PCF File format. Available at: http://www.wotsit.org
[3] Microsoft Corporation, What is TrueType, 1997. Available at: http://www.microsoft.com/typography/what/what.htm
[4] Adobe Systems Incorporated, PostScript Language Reference third edition, Reading, Massachusets, Addison-Wesley Publishing Company, 1999. Available at: http://partners.adobe.com/asn/developer/PDFS/TN/
[5] Microsoft Corporation, OpenType initiative FAQ, 1997. Available at: http://www.microsoft.com/typography/faq/faq9.htm
[6] Donald E. Knuth, Computers & Typesetting, Volume C, The METAFONTbook, Reading, Massachusetts: Addison-Wesley, 1984.
[7] Adobe Solutions Network: Developer Program: Kerning, 2000. Available at: http://partners.adobe.com/asn/developer/type/kerning.html
[8] TrueType Outlines, http://www.truetype.demon.co.uk/ttoutln.htm
[9] George Williams, PfaEdit: A PostScript Font Editor, 2000. Available at: http://pfaedit.sourceforge.net/bezier.html
[10] Adobe Solutions Network, Adobe Font Formats, File Types and Q&A, 2000. Available at: http://partners.adobe.com/asn/developer/type/ftypes.html
[11] Bitstream Incorporated, Dynamic Fonts. Available at: http://www.truedoc.com/webpages/intro/
[12] Paul Neubauer, Monospaced Fonts, 2000.
Available at: http://home.bsu.edu/prn/monofont/
[13] Tim North, Better Writing Skills, 2002. Available at: http://www.thewritemarket.com/articles/north.htm
[14] Sun Microsystem, Font Overview. Available at: http://java.sun.com/j2se/1.3/docs/guide/intl/addingfonts.html
[15] Bureau of Indian Standards, Indian script code for information interchange - ISCII, Indian Standard IS 13194:1991, 1991.
[16] (STC) TSCII - The Tamil Encoding Standard. Available at: http://www.tamil.net/tscii/tscii.html
[17] C-DAC, ISFOC standard for fonts. Available at: http://www.cdacindia.com/html/gist/standard/isfoc.asp
[18] Unicode Inc., Unicode Home Page, 1991. Available at: http://www.unicode.org
[19] Sandeep Sibal, The JavaScript transliterator (Jtrans), 1996. Available at http://www.sibal.com/sandeep/jtrans/

[20] Richard S. Shuford, xterm: Software terminal emulation under the X Window System, 1995. Available at: http://www.cs.utk.edu/~shuford/terminal/xterm.html
[21] Mike Fabian, kterm, 2002. Available at: http://packages.debian.org/unstable/X11/kterm.html
[22] Microsoft Corporation, The TrueType Rasterizer, 1997. Available at: http://www.microsoft.com/typography/what/raster.htm
[23] Microsoft Corporation, Windows Glyph Processing, 2000. Available at: http://www.microsoft.com/typography/developers/opentype/default.htm
[24] Abhiram Ranade, Meghanad Ranade, "Devanagari Pen-written Character Recognition", Proceedings of the Ninth International Conference on Advanced Computing and Communications (ADCOM 2001), PP 255-261, 2001
[25] R.S.R Kunte, R.D.S. Samuel, "Writing Pad for Indian Languages On-Line Character Recognition System for Handwritten Characters/Script with Bilingual Facility Employing Neural Classifier and Wavelet Features.", In Knowledge Based Computer Systems, PP 1-12, 2000.