Typesetting Rare Chinese Characters in LaTeX

Wai Wong, Candy L.K. Yiu, Kelvin C.F. Ng
Department of Computer Science
Hong Kong Baptist University
Kowloon Tong, Kowloon
Hong Kong
wwong,candyyiu,ncfkelvin@comp.hkbu.edu.hk
http://www.comp.hkbu.edu.hk/~wwong

Abstract

Written Chinese has tens of thousands of characters. But most available fonts contain only around 6 to 12 thousand common characters that can meet the needs of everyday users. However, in publications and information exchange in many professional fields, a number of rare characters that are not in common fonts are needed in each document. This paper describes a method of typesetting such rare characters in LaTeX. The document author describes a rare character in HanGlyph when such need arises. A Chinese character synthesis system renders the glyph according to this description and collects the newly created glyphs into a font so they are available in the body of the LaTeX document.

Résumé

Le chinois écrit possède des dizaines de milliers d'idéogrammes. Mais la plupart des fontes disponibles ne contiennent que quelques 6 et 12 mille idéogrammes standard. Néanmoins, les publications en sciences humaines, comme la littérature classique, l'archéologie, ou, tout simplement, les registres municipaux de noms, nécessitent l'utilisation de grand nombre d'idéogrammes rares.

Cet article décrit une méthode de composition de de tels caractères rares sous LaTeX. L'auteur du document décrit le caractère rare dans une syntaxe spéciale, appelée HanGlyph. Un système de synthèse de caractères chinois génère le glyphe d'après cette description et assemble les glyphes ainsi produits dans une fonte pour être immédiatement utilisés par LaTeX dans le corps du document.

Introduction

The Chinese written script is ideographic. Each ideograph, known as *hanzi*, or more commonly 'Chinese character', has its own visual structure and carries certain meaning. There are tens of thousands of *hanzi* or Chinese characters.

The most notable difference between an ideographic script and a phonographic script, such as the Latin alphabet, Slavonic alphabet, and so on, is the huge number of characters that the former script contains.

Historically, the number of *hanzi* in existence increases with time. This is reflected in many dictionaries published in various times. For example, the first influential book on *hanzi* 說文解字 (shuōwénjiĕzì[1]) published around 100 AD collected 10,516 characters. By the time of the Qing dynasty, the more famous dictionary 康熙字典 (kāngxīzìdiăn) (published in 1716) contains 47,043 characters. The largest contemporary dictionary 中華字海 (zhōnghuázìhăi) documents a staggering 86,000 characters [5].

In order to facilitate the machine processing of the Chinese script, coded character sets have been developed [8]. Commonly used coded Chinese character sets are GB2312-80, CNS11643, Big5, JIS X 0208-1983 and KS X 1001:1992. The number of Chinese characters encoded in these character sets varies from around 4,888 (in KS X 1001:1992) to 48,027 (in CNS 11643-1992).

The main reason behind the selection of characters in these encoded character sets is the frequency with which each character appears in common documents, such as newspapers, textbooks and business correspondence. A statistical study reveals that around 6,600 Chinese characters can cover 99.999% of daily use [5]. It seems that having a large enough encoded character set will solve the problem.

Recent international effort in character set standardization results in the Unicode (version 4.0 released in May 2003) [12] and ISO/IEC 10646 standards. Unicode 3.0 [9] encoded 27,484 Chinese characters. 42,711 characters were added in version 3.2 [11].

Although the new international standards include a

[1] The word shuōwénjiĕzì following the *hanzi* is in *pinyin*, a phonetic transcription of Chinese characters.

huge number of Chinese characters, in practice, several problems remain unsolved. The first is that the design and creation of fonts of a huge character set is very expensive. It is also not economical to process and maintain such large fonts given that many of the characters in them are rarely used. Furthermore, a large number of existing systems may not be able to handle such a large character set. Transferring a document to such older systems will result in missing characters or even crash the system.

One possible solution is to synthesize the characters when needed. We have designed a Chinese character description language called *HanGlyph*. It is a high-level abstract description language which captures the essential features of characters. These features are the topology of the strokes and their relative size and location. We are developing a Chinese character synthesis system (CCSS) to render the characters from the *HanGlyph* description. The *HanGlyph* language and the method of synthesizing Chinese characters will be described in the next section.

Using the *HanGlyph* description and the character synthesis system, we can typeset rarely used Chinese characters in LaTeX documents. We developed a simple macro package to allow the document author to embed *HanGlyph* descriptions in a LaTeX document. During the first time the LaTeX document is formatted, a *HanGlyph* file containing all character descriptions is generated. This file is then processed by the synthesis system to create a font. The next time the LaTeX document is formatted, the newly generated Chinese character font is accessible. Thus, the character can be typeset. Later sections will describe how to use this macro and outline its implementation.

Chinese character synthesis

The HanGlyph language is a Chinese character description language. It provides a means of describing the topological arrangements of strokes in Chinese characters. Each Chinese character can be decomposed into a number of parts called *components*. Each component consists of a number of *strokes*. For example, as illustrated in Figure 1, the character 明 (míng meaning bright) can be decomposed into two components 日 (rì) and 月 (yuè). The first component consists of four strokes: 丨(豎 shù), ㇕(橫折 hèngzhé), 一(橫 héng) and 一.

A *HanGlyph* expression describes the arrangement of the strokes in an abstract fashion. This means that only topological information is captured and no geometric information is included. For example, the *HanGlyph* expression describing the character 二 is `h h =` which means the character is composed by two héng strokes, one above the other. We do not need to specify the exact coordinates of the starting point of each stroke. These will be worked out by the character synthesizer.

One of the criteria for writing a good *HanGlyph* expression is to be able to distinguish similar characters. For example, the characters 土 and 士 are composed of the same strokes and in the same arrangements. The only difference is the relative length of the two 一 strokes. The *HanGlyph* expressions to describe these characters are `h h=< s+_` and `h h=> s+_`, respectively. The dimensional relation symbols < and > specify the relative length of the two 一 strokes.

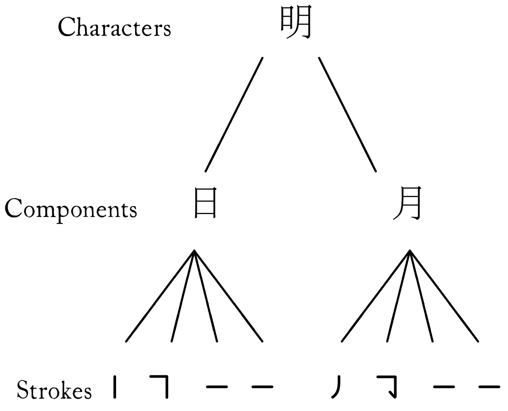


Fig. 1: Composition of a Chinese character

Because the smallest building blocks of a Chinese character are the strokes, the *HanGlyph* language takes the strokes as its primitives. According to many studies of Chinese characters, we selected 41 such primitive strokes. *HanGlyph* composes characters using a small set of five operations which are illustrated in Figure 2:

- top-bottom,
- left-right,
- fully-enclosed,
- partially-enclosed, and
- crossing.

It would be very tedious if every character description contains details down to each single stroke. Using the fact that characters can be decomposed into components and many components appear in a number of characters, *HanGlyph* allows macros to be defined to stand for components. Thus, expressions can be very concise.

In summary, *HanGlyph* provides an abstract way to describe Chinese characters which captures their essential characteristics so that they can be rendered. Details of the *HanGlyph* language can be found in our paper to be presented at *TUG* 2003 [14].

The Chinese character synthesis system (CCSS) is responsible for rendering the character glyphs from their *HanGlyph* descriptions. The core of CCSS is implemented in METAPOST [6]. It consists of a library of METAPOST macros implementing various *HanGlyph* operations. The input to CCSS is a sequence of *HanGlyph*

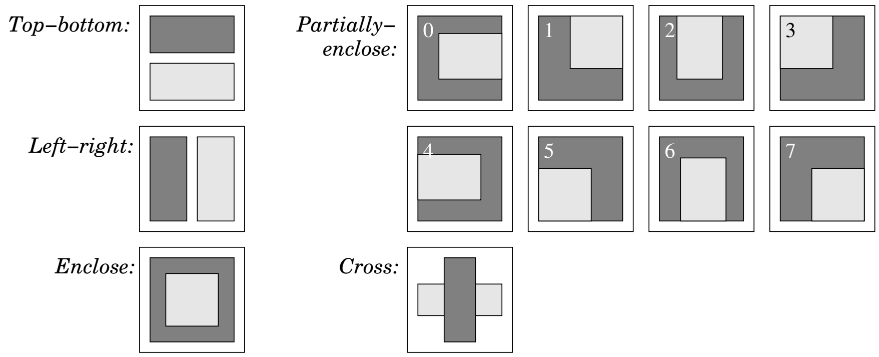
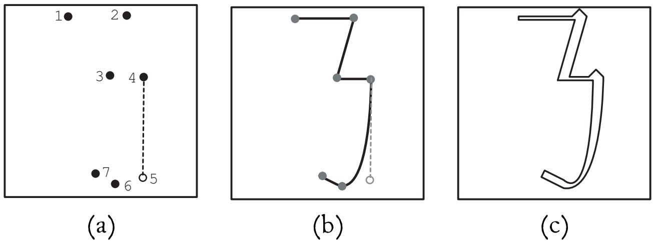


Fig. 2: Graphical representation of operators



(a) (b) (c)

Fig. 3: Basic stroke macros for the stroke ㇋

expressions. A front end translator converts these *HanGlyph* expressions into a METAPOST program. Running METAPOST on this program will then generate a set of PostScript files each of which contains a single glyph.

The METAPOST macro library is organized into two major parts: the primitive strokes and the composition operations. Each primitive stroke is implemented as three METAPOST macros, namely a control point macro, a skeleton macro and an outline macro. The control point macro defines the control points, the skeleton macro specifies the skeletal path that connects the control points, and the outline macro draws the outline which are defined relative to the control points and the skeletal path. Figure 3 shows a sample stroke.

The composition operation macros perform the composition. This is done by transforming the control points of each stroke and position them to the correct location within a character bounding box. When all strokes of a character are positioned at the correct location, the skeleton macros are called to define the skeletal strokes. Then, the outline macros are called to draw the outline.

Using HanGlyph in LaTeX

The *HanGlyph* language provides a means of describing Chinese characters, and the CCSS system renders the characters so that we can have visual output. How can we then use them in the context of typesetting professional documents in LaTeX? Figure 4 illustrates the process flow.

First of all, the document author will embed the *HanGlyph* expressions of the required Chinese characters in the LaTeX source files. Typesetting the document will generate a *HanGlyph* file that contains all *HanGlyph* expressions in the source. This file is then processed by the CCSS fontmaker to generate a font in `tfm` and `pk` format so that the next time LaTeX is run it can find the font.

To allow the author to embed *HanGlyph* expressions and to use the synthesized characters in a LaTeX document, we developed a macro package `hanglyph`. This provides a very simple interface for authors to use *HanGlyph*. To define a new character, the author writes the *HanGlyph* expression in the preamble of a LaTeX document as below:

```
\hgchar{tu}{h h=< s+_}
\hgchar{shi}{h h=> s+_}
```

Each call to the macro `\hgchar` defines a new *HanGlyph* character. The macro takes two arguments: The first is a character name which can be used in the document to typeset the character; the second is the *HanGlyph* expression describing the character. For example, the first call to `\hgchar` above defines a new macro `\tu` to represent the character 土.

When processing the LaTeX document, each call to the macro `\hgchar` also writes a *HanGlyph* character definition to a *HanGlyph* file. Each character definition associates a character code to its *HanGlyph* expression.

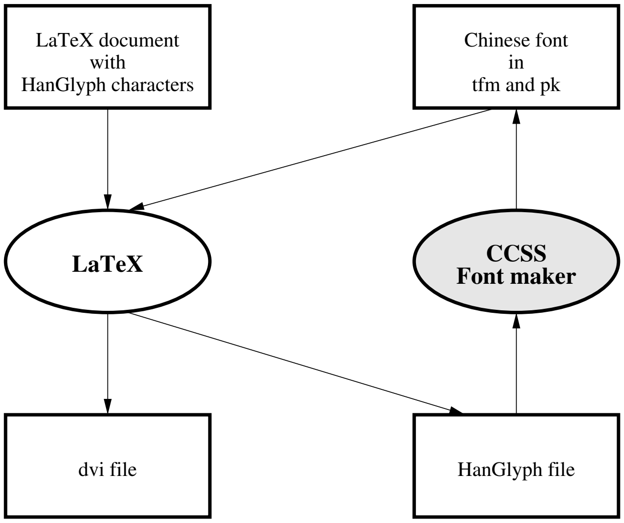


Fig. 4: Typesetting process

The hanglyph package automatically keeps track of the character code. By default, the first character, defined by the first call to the macro \hgchar, has the code 0.

In the body of the LaTeX document, the character names defined using the macro \hgchar can be used to typeset these characters. So the author can type

```
The Chinese characters \tu{} and \shi{}
are composed of the same strokes and in
the same arrangements.
```

to typeset the following:

> The Chinese characters 土 and 士
> are composed of the same strokes and in
> the same arrangements.

To use the \hgchar macro, the author should include the hanglyph package in the LaTeX source file. After running LaTeX once, a *HanGlyph* file having the suffix hgc and the same base name as the LaTeX document is generated. The author should execute the CCSS fontmaker to generate the tfm and pk files needed by LaTeX and the dvi driver. The fontmaker should be executed each time the embedded *HanGlyph* expressions are changed so that the font is up-to-date. Figure 6 shows some sample characters generated by our system.

This approach of typesetting Chinese characters aims at applications where only a small number of rarely used characters appear in a document. By 'rarely used characters', we mean those characters that cannot be found in commonly available fonts, such as those that come with the popular operating systems or typesetting systems. Our method provides a simple and convenient way to solve this missing character problem. Therefore, the current implementation of the hanglyph package is able to handle up to 256 characters in a document.

The implementation

The implementation is in two parts: the macro package hanglyph, and the CCSS fontmaker implemented as a suite of scripts and C programs.

The hanglyph *package* defines the macro \hgchar for the document author to specify the Chinese character and a number of auxiliary macros to manage the Chinese character font.

The package works as follows: All characters defined by calling \hgchar will be collected into a font with the default font name hgfont. Each character is assigned a character code in the order it is defined. A counter named hg@charcode keeps track of the number of characters that have been defined.

The character definition macro \hgchar takes two arguments, namely the character name and a *HanGlyph* expression. It performs two tasks: defining a character macro and writing the character description to the *HanGlyph* file.

In order to allow the character name macros to be used in the document body, the definitions must be placed in the preamble. At 'begin document', the *HanGlyph* file will be closed and all character macros have been defined. Each character macro is defined to represent a single character in the default *HanGlyph* font.

By default, the font metric file and the glyph file have the same basename as the document file. The hanglyph package provides a macro \hgfontname for the author to change the font file name.

The CCSS fontmaker takes a *HanGlyph* file and generates a font to be used in the LaTeX document. The process is slightly long-winded but the philosophy is to use as many existing tools and file formats as possible, so that the generated files are compatible with existing systems. This process is depicted in Figure 5.

The core of the process is the Chinese character synthesis system. The output of CCSS is a set of encapsulated PostScript (`eps`) files. Each file contains a single glyph. They are then converted to bitmaps in portable bitmap format (`pbm`). This task is accomplished by Ghostscript. The set of `pbm` files is then merged into a large bitmap, which is then converted to a TEX font in a pair of `gf` and `pl` files. This pbm-to-gf conversion is performed by a utility called `pbmtogf` developed by the first author several years ago [13] and since been made available on CTAN. The program to merge a set of pbm files is `pbmmerge`, which is relatively simple to implement. The last two programs, `pltotf` and `gftopk`, are available in all TEX implementations. This process is easily integrated in a single script.

Discussion and conclusion

Using the *HanGlyph* description language and CCSS, one can generate Chinese character glyphs that are not found in commonly available fonts. To typeset such characters requires generating a font in the format understood by the typesetting system. The `hanglyph` package enables users of the LATEX system to typeset rarely used Chinese characters.

At this stage, we have experimented with a small number of characters. The results (some of which are illustrated in Figure 6) show that this approach is feasible, and very promising. We are planning to carry out more experiments to typeset a reasonably large set of characters. The purpose is to study the effect of rasterization and to improve the quality of the glyphs generated by CCSS. We are confident that our method of character synthesis can produce reasonably good quality and readable Chinese character glyphs.

It is obvious that the current font making process is a bit inefficient. Another shortcoming is the loss of scalability by converting the vector representation of character glyphs into a bitmap form. These weaknesses will certainly be eliminated as the development of CCSS progresses.

The current stage of the development of CCSS concentrates on the experiment and improvement of glyph algorithms. It is planned that future versions of CCSS will generate outline fonts for *HanGlyph* input. This will certainly improve quality and usability.

This paper describes only one of many possible applications of *HanGlyph* and Chinese character synthesis. We envisage that CCSS will contribute greatly to easing a major difficulty in Chinese textual information exchange and presentation in a heterogenous environment.

References

[1] 蘇培成 (sūpéichéng).《二十世紀的現代漢字研究》(èrshíshìjìde xiàndài hànzì xiānjiù) 書海出版社 (shūhǎi chūbǎnshè), 2001. [SU Pei Cheng, *20th Century Research on Modern Chinese Characters*, Su Hai Press, 2001.]

[2] 劉連元 (liúliányuán).〈漢字拓扑結構分析〉(hànzì tuōpǔ jiékoù fēnxī). In《漢字》(hànzì). 上海教育出版社 (shànghǎi jiàoyù chūbǎnshè), 1993. [LIU Lian Yuan, *Analysis of the Topological Structure of Chinese Characters*, Shanghai Education Press, 1993.]

[3] 傅永和 (fùyónghé).《漢字結構和構造成分的基楚研究》(hànzì jiékoù he koùzào chéngfènde jī chǔ yānjiù). 上海教育出版社 (shànghǎi jiàoyù chū bǎnshè), 1993. [FU Yong He, *Basic Research on the Structure of Chinese characters and Their Constituents*, Shanghai Education Press, 1993.]

[4] 傅永和 (fùyónghé).《中文信息處理》(zhōngwén xìnxī chǔlǐ). 廣東教育出版社 (guǎngdōng jiàoyù chūbǎnshè), 1999. [FU Yong He, *Chinese Information Processing*, Guangdong Education Press, 1999.]

[5] 馮志偉 (féngzhìwěi).《計算語言學基礎》(jìsuàn yǔyánxué jīchǔ). 商務印書館 (shāngwù yìnshū guǎng), 2001. [FENG Zhi Wei, *Foundations of Computational Linguistics*, The Commercial Press, 2001.]

[6] J. D. Hobby. *A METAFONT-like system with PostScript output*, *TUGboat*, Vol. 10, No. 4, pp. 505–512, December 1989.

[7] John D. Hobby. *A User's Manual for METAPOST*, AT&T Bell Laboratory Computer Science Technical Report, No. 162, 1992.

[8] Ken Lunde. *CJKV information processing*. O'Reilly, 1999.

[9] The Unicode Consortium. *The Unicode Standard, Version* 3.0. Addison-Wesley, 2000.

[10] The Unicode Consortium. *Unicode Version* 3.1, The Unicode Standard Annex No. 27. `http://www.unicode.org/reports/tr27/`

[11] The Unicode Consortium. *Unicode Version* 3.2, The Unicode Standard Annex No. 28. `http://www.unicode.org/reports/tr28/`

[12] The Unicode Consortium. *The Unicode Standard, Version* 4.0. Addison-Wesley, 2003.

[13] Wai Wong. *pbmtogf — converting bitmap to font*. `http://www.comp.hkbu.edu.hk/~wwong/typeset/pbmtogf/`

[14] Candy L.K. Yiu, Wai Wong. *Chinese character synthesis using METAPOST*. *TUGboat*, Vol. 24, No. 1, pp. 85–93, 2003 (TUG 2003 proceedings).

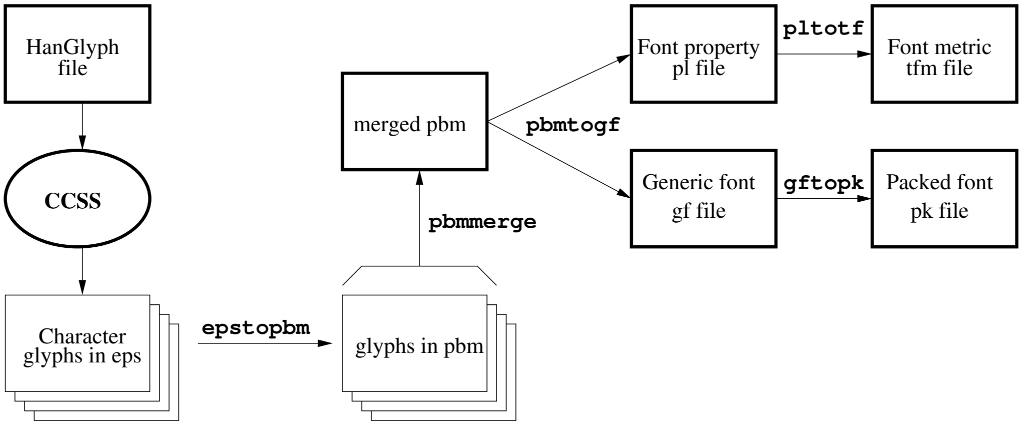


Fig. 5: The fontmaking process

<table>
<tr><th>Glyph</th><th>HanGlyph expression</th></tr>
<tr><td>八</td><td>p n |</td></tr>
<tr><td>把</td><td>hSt ba_1 | !~</td></tr>
<tr><td>班</td><td>wang_b_2 d q | ~ | !~ wang_a_2 | !~</td></tr>
<tr><td>般</td><td>zhou_1 shu_1 |</td></tr>
<tr><td>版</td><td>pian_4 /Pq kn ^7 _]/|</td></tr>
<tr><td>半</td><td>h s+ d Z|!~ h= @ ’</td></tr>
<tr><td>壁</td><td>qih sih ^7 0.8 0.8 _ xin_1_b | hhs1 =</td></tr>
<tr><td>避</td><td>div qih sih ^7 0.8 0.8 _ xin_1_b | ^1</td></tr>
<tr><td>辨</td><td>xin_1_b D q | ~ | xin_1_b |</td></tr>
<tr><td>昌</td><td>ri_4 ri_4 = <</td></tr>
<tr><td>长</td><td>p h=0.4<!~n=>]e+#[</td></tr>
<tr><td>吃</td><td>sih ph m = |!~ 0.5 0.5</td></tr>
<tr><td>吹</td><td>sih qian_4 |!~ 0.5 0.5</td></tr>
<tr><td>辞</td><td>she_2 xin_1_b |</td></tr>
<tr><td>此</td><td>zhi_3_a wp |</td></tr>
<tr><td>寸</td><td>h S+<’ d^5</td></tr>
<tr><td>到</td><td>h rd = hst = sS |!~</td></tr>
<tr><td>敌</td><td>she_2 pu_1_b |</td></tr>
<tr><td>笛</td><td>zhu_b_2 ri_4 s + _ < =</td></tr>
<tr><td>第</td><td>zhu_b_2 ihC s + ’ < p ^5 0.5 _ [=</td></tr>
<tr><td>独</td><td>pXp chong_2 |!~</td></tr>
</table>

Fig. 6: Some characters generated using *HanGlyph* and CCSS