漢字字形データベース GlyphWiki によって漢字構造情報を生成する試み

劉　冠偉（京都大学人文科学研究所）

2024 年 7 月 26 日

1　はじめに

漢字の字形を共有するデータベース GlyphWiki[*1]は，すでに 140 万以上の字形を公開している．ユーザが漢字字形を登録・編集することで，インターネット上で漢字字形を共有できる．字形画像および外字表示用のフォントも同時に自動生成され，漢字の研究や教育に重要な情報源となっている．しかし，GlyphWiki の字形データには，字形を構成する部品の関係が記述されておらず，漢字の構造によって GlyphWiki に公開されている字形を検索することができない．

漢字構造情報は漢字構成記述文字列（Ideographic Description Sequence，以下 IDS と略す）によって表現できる [1]. IDS は，漢字構成記述文字（Ideographic Description Characters，以下は IDC と略す）を用いることによって，構成要素の関係を記述する [2].

本稿では，GlyphWiki の字形データを利用して，漢字構造情報を生成する方法を提案する．具体的には，字形データから構成部品の座標を取得し，各部品の筆画が実際に描画される座標を基に部品の位置を計算する．さらに，部品の位置関係を判断し，漢字構造情報としての IDC を生成する．

本稿の構成は次の通りである．まず，2章では，GlyphWiki の字形データについて説明する．3章では，漢字構造情報の生成方法について述べる．4章では，提案手法の実験結果を示す．

2　GlyphWiki 字形データ

2.1　データの取得

GlyphWiki では 1 つの字形を「グリフ」と呼び，グリフデータは名前の「グリフ名」，字形を検索するための漢字符号である「関連漢字」と字形画像に変換できる「KAGE データ」で構成される．さらに，バージョン管理のための「バージョン番号」が付与される．

GlyphWiki は日ごとにすべての字形データをまとめてダンプデータとして公開している[*2][3]．ダンプデータはすべてのバージョンが含まれる「dump_all_versions.txt」，最新バージョンのみが含まれる「dump_newest_only.txt」が存在する．

[*1] https://glyphwiki.org/

[*2] http://glyphwiki.org/dump.tar.gz

2.2　KAGE データの仕様

GlyphWiki の字形データは KAGE データという形式で提供される．KAGE データは，KAGE データは，字形の筆画の骨格を記録し，ほかの字形を引用することができる [4]．GlyphWiki には KAGE データの仕様が公開されている [4][5]．それによると，KAGE データは複数行からなり，筆画行，部品引用行，特殊行の 3 種類がある．

筆画行は，「直線」,「曲線」などの筆画の種類と，筆画の始点と終点などの制御点の座標が記されている．部品引用行は，ほかの字形を部品として引用し，その部品を配置する座標が指定される．これにより，字形に含まれる筆画を再利用することが可能である．特殊行は，字形を回転させたり反転させたりするための指示を行う．

KAGE データは GlyphWiki と同一の開発者が公開した「kage-engine」[*3]という JavaScript で構築されたライブラリによって，$(200, 200)$ サイズのカンバスにて上記の座標を用いて字形を描画して，最後に SVG または PNG 形式の画像に変換し，字形を表示する．

2.3　引用部品の配置座標

前述のように，部品引用行は，引用するグリフ名以外に，部品の配置位置を示す矩形の座標も含まれている．例として，グリフ名が「u5289-j」の「劉」字の KAGE データを次に示す．

```
99:0:0:0:0:224:200:u2896b-01:0:0:0
99:0:0:54:0:196:200:u5202-02:0:0:0
```

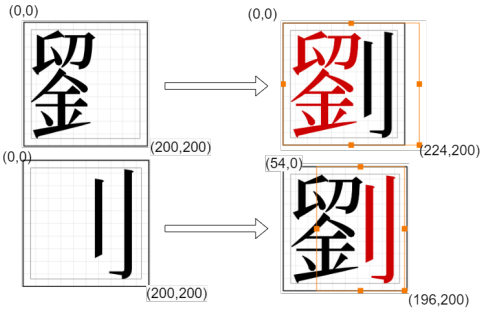

図1　引用部品の展開の一例

各行最初の「99:00:00」は部品引用行であることを示す．「u2896b-01」を $(0, 0)$ から $(224, 299)$ までの矩形に配置する[*4]．次に，「u5202-02」を $(54, 0)$ から $(196, 200)$ までの矩形に配置する（図1）．この矩形は以下に配置矩形と呼ぶ．

3　漢字構造情報生成の手法

3.1　漢字構造情報生成の流れ

漢字構造情報を生成する手法は次の通りである．

1. GlyphWiki のダンプデータから，各字形の KAGE データを取得する．
2. KAGE データから，部品引用行を取得し，部品の配置座標を取得する．
3. 部品の配置座標を利用して，部品の筆画の描画座標を計算する．
4. 部品の描画座標を基に，各部品の位置関係を分析し，漢字構造情報の IDC に変換する．

[*3] https://github.com/kamichikoichi/kage-engine

[*4] [5] では「展開」と呼ぶ．

3.2　筆画描画座標の取得

部品配置座標は，引用部品の配置位置を制御するための情報であるが，引用部品に含まれる筆画が実際に描画される座標ではない（図2）．引用部品の位置関係をを判断するには，筆画の描画座標を取得することが必要である．

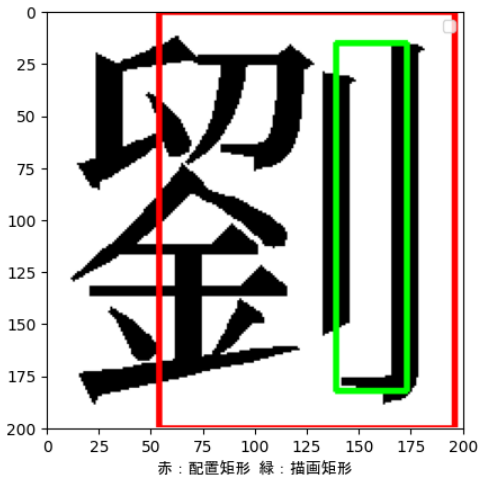

図2　配置矩形と描画矩形

配置座標は，引用した部品の元字形にある座標（以下，元座標と呼ぶ）に対して，平行移動，拡大・縮小（スケール）変換が行われる．具体的には，元座標 (x, y) から配置座標 (x', y') への変換は，以下のように表されます．

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} s_x & 0 \\ 0 & s_y \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} t_x \\ t_y \end{pmatrix} \quad (1)$$

ここで，s_x は x 軸方向の拡大縮小率，s_y は y 軸方向の拡大縮小率，t_x は x 軸方向の平行移動量，t_y は y 軸方向の平行移動量です．

引用部品矩形の左上座標 (x_{tL}, y_{tL}) が常に $(0, 0)$ であるため，配置矩形の左上座標 (x'_{tL}, y'_{tL}) は次のように表せれる．

$$\begin{pmatrix} x'_{tL} \\ y'_{tL} \end{pmatrix} = \begin{pmatrix} s_x & 0 \\ 0 & s_y \end{pmatrix} \begin{pmatrix} 0 \\ 0 \end{pmatrix} + \begin{pmatrix} t_x \\ t_y \end{pmatrix} \quad (2)$$

引用部品矩形の左上座標 (x_{bR}, y_{bR}) が常に $(200, 200)$ であるため，配置矩形の左上座標 (x'_{bR}, y'_{bR}) は次のように表せれる．

$$\begin{pmatrix} x'_{bR} \\ y'_{bR} \end{pmatrix} = \begin{pmatrix} s_x & 0 \\ 0 & s_y \end{pmatrix} \begin{pmatrix} 200 \\ 200 \end{pmatrix} + \begin{pmatrix} t_x \\ t_y \end{pmatrix} \quad (3)$$

配置座標から描画座標を求めるため，スケール変換と平行移動を逆変換する必要がある．

3.3　描画矩形の生成

前述のとおり，配置矩形は引用部品の実際の筆画の描画よりも広いため，漢字構造を正確に判断することが困難である．実際の筆画の描画矩形を生成するためには，引用部品が含むすべての筆画の座標の最大値を使用する．

その座標を (x_{max}, y_{max}) とすると，描画矩形の左上座標 (x_{dL}, y_{dL}) および右下座標 (x_{dR}, y_{dR}) は以下のように表される．

$$\begin{aligned} x_{rtL} &= t_x \\ y_{rtL} &= t_y \\ x_{rbR} &= s_x x_{max} + t_x \\ y_{rbR} &= s_y y_{max} + t_y \end{aligned} \quad (4)$$

3.4　IDC の生成

各部品の描画矩形の左上座標と右下座標を以下のように定義する．

部品 1 (x_1, y_1) (x_2, y_2)

部品 2 (x_3, y_3) (x_4, y_4)

部品 3 (x_5, y_5) (x_6, y_6)

3.4.1 ⿰ (U+2FF0)

⿰ (U+2FF0) の Unicode 文字名は「left to right（左→右[*5]）」であるため，部品 1 の右下座標 x_2 は部品 2 の左上座標 x_3 より小さい．部品 1 の右下座標 y_2 は部品 2 の左上座標 y_3 より大きい (図3)．

$$x_1 < x_3 \wedge y_2 > y_3 \tag{5}$$

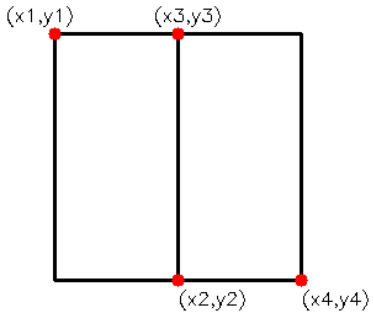

図3　⿰ (U+2FF0)

3.4.2 ⿱ (U+2FF1)

⿱ (U+2FF1) の Unicode 文字名は「above to below（上→下）」であるため，部品 1 の右下座標 y_2 は部品 2 の左上座標 y_3 より小さい．部品 1 の右下座標 x_2 は部品 2 の左上座標 x_3 より大きい（図4）．

$$y_2 < y_3 \wedge x_2 > x_3 \tag{6}$$

3.4.3 ⿲ (U+2FF2)

⿲ (U+2FF2) の Unicode 文字名は「left to middle and right（左→中央→右）」であるため，部品 1 の右下座標 x_2 は部品 2 の左上座標 x_3 より小さい．部品 1 の右下座標 y_2 は部品 2 の左上座標 y_3 より大きい．部品 2 の右下座標 x_4 は部品 3 の左上座標 x_5 より小さい．部品 2 の右下座標 y_4 は部品 3 の左上座標 y_5 より大きい（図5）．

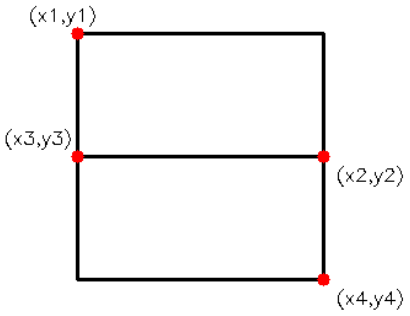

図4　⿱ (U+2FF1)

$$x_1 < x_3 \wedge x_2 < x_4 \wedge x_3 < x_5 \wedge x_4 < x_6 \tag{7}$$

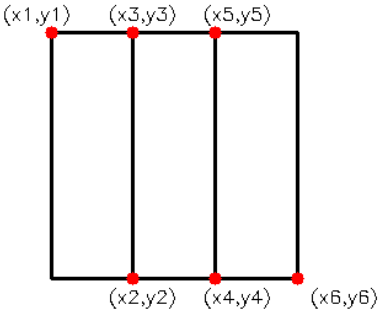

図5　⿲ (U+2FF2)

3.4.4 ⿳ (U+2FF3)

⿳ (U+2FF3) の Unicode 文字名は「above to middle and below（上→中央→下）」であるため，部品 1 の右下座標 y_2 は部品 2 の左上座標 y_3 より小さい．部品 1 の右下座標 x_2 は部品 2 の左上座標 x_3 より大きい．部品 2 の右下座標 y_4 は部品 3 の左上座標 y_5 より小さい．部品 2 の右下座標 x_4 は部品 3 の左上

[*5] 翻訳は [6] によるものである．以下同様．

座標 x_5 より大きい（図6）.

$$y_1 < y_3 \wedge y_2 < y_4 \wedge y_3 < y_5 \wedge y_4 < y_6 \quad (8)$$

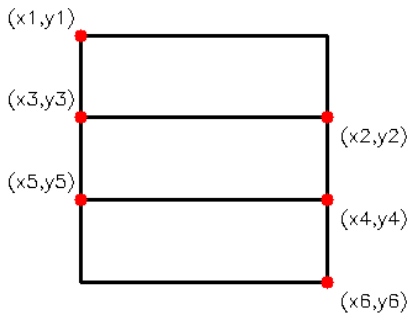

図6　(U+2FF3)

3.4.5　(U+2FF4)

(U+2FF4) の Unicode 文字名は「surround from above（四方→囲む）」であるため，部品 1 の右下座標 x_2 は部品 3 の左上座標 x_4 より小さい．部品 1 の右下座標 y_2 は部品 3 の左上座標 y_4 より小さい．部品 1 の右下座標 x_2 は部品 3 の左上座標 x_4 より小さい．部品 1 の右下座標 y_2 は部品 3 の左上座標 y_4 より小さい（図7）.

$$x_1 < x_3 \wedge y_1 < y_3 \wedge x_2 > x_4 \wedge y_2 > y_4 \quad (9)$$

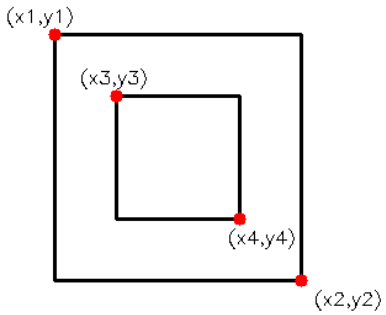

図7　(U+2FF4)

3.4.6　(U+2FF5)

(U+2FF5) の Unicode 文字名は「surround from below（上→囲む）」であるため，部品 1 の右下座標 x_2 は部品 3 の左上座標 x_4 より小さい．部品 1 の右下座標 y_2 は部品 3 の左上座標 y_4 より大きい．部品 1 の右下座標 x_2 は部品 3 の左上座標 x_4 より小さい．部品 1 の右下座標 y_2 は部品 3 の左上座標 y_4 より大きい（図8）.

$$x_1 < x_3 \wedge y_1 < y_3 \wedge x_2 > x_4 \wedge y_1 < y_3 \quad (10)$$

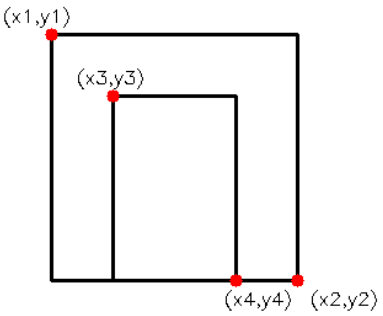

図8　(U+2FF5)

3.4.7　(U+2FF6)

(U+2FF6) の Unicode 文字名は「surround from left（下→囲む）」であるため，部品 1 の右下座標 x_2 は部品 3 の左上座標 x_4 より小さい．部品 1 の右下座標 y_2 は部品 3 の左上座標 y_4 より大きい．部品 1 の右下座標 x_2 は部品 3 の左上座標 x_4 より小さい．部品 1 の右下座標 y_2 は部品 3 の左上座標 y_4 より大きい（図9）.

$$x_1 < x_3 \wedge x_2 > x_4 \wedge y_2 < y_4 \quad (11)$$

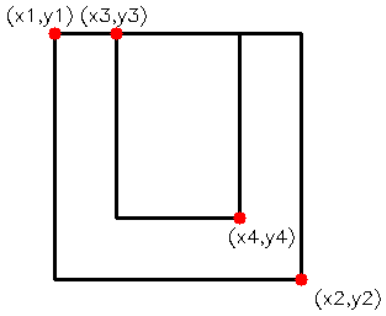

図9　(U+2FF6)

3.4.8　(U+2FF7)

(U+2FF7) の Unicode 文字名は「surround from upper right（左→囲む）」であるため，部品 1 の右下座標 x_2 は部品 3 の左上座標 x_4 より小さい．部品 1 の右下座標 y_2 は部品 3 の左上座標 y_4 より大きい．部品 1 の右下座標 x_2 は部品 3 の左上座標 x_4 より小さい．部品 1 の右下座標 y_2 は部品 3 の左上座標 y_4 より大きい（図10）．

$$x_1 < x_3 \wedge y_1 < y_3 \wedge y_2 < y_4 \qquad (12)$$

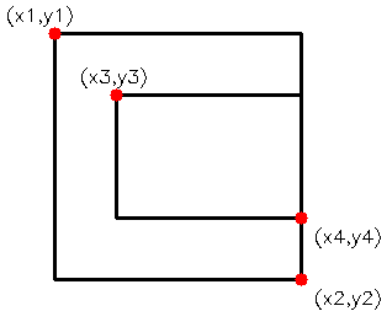

図10　(U+2FF7)

3.4.9　(U+2FF8)

(U+2FF8) の Unicode 文字名は「surround from upper left（左上→囲む）」であるため，部品 1 の右下座標 x_2 は部品 3 の左上座標 x_4 より小さい．部品 1 の右下座標 y_2 は部品 3 の左上座標 y_4 より大きい．部品 1 の右下座標 x_2 は部品 3 の左上座標 x_4 より小さい．部品 1 の右下座標 y_2 は部品 3 の左上座標 y_4 より大きい（図11）．

$$x_1 < x_3 \wedge y_1 < y_3 \qquad (13)$$

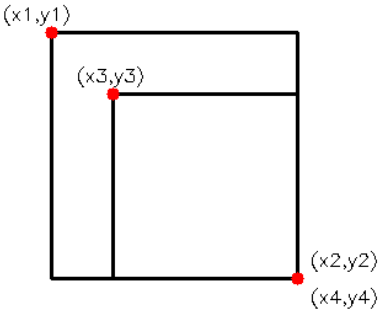

図11　(U+2FF8)

3.4.10　(U+2FF9)

(U+2FF9) の Unicode 文字名は「surround from lower right（右上→囲む）」であるため，部品 1 の右下座標 x_2 は部品 3 の左上座標 x_4 より小さい．部品 1 の右下座標 y_2 は部品 3 の左上座標 y_4 より大きい．部品 1 の右下座標 x_2 は部品 3 の左上座標 x_4 より小さい．部品 1 の右下座標 y_2 は部品 3 の左上座標 y_4 より大きい（図12）．

$$y_1 < y_3 \wedge x_2 > x_4 \qquad (14)$$

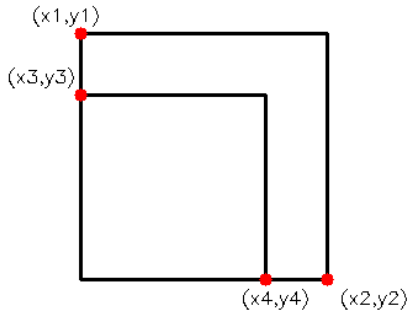

図12　□ (U+2FF9)

3.4.11　□ (U+2FFA)

□ (U+2FFA) の Unicode 文字名は「surround from lower left（左下→囲む）」であるため，部品 1 の右下座標 x_2 は部品 3 の左上座標 x_4 より小さい．部品 1 の右下座標 y_2 は部品 3 の左上座標 y_4 より大きい．部品 1 の右下座標 x_2 は部品 3 の左上座標 x_4 より小さい．部品 1 の右下座標 y_2 は部品 3 の左上座標 y_4 より大きい（図13）．

$$x_1 < x_3 \wedge y_2 > y_4 \tag{15}$$

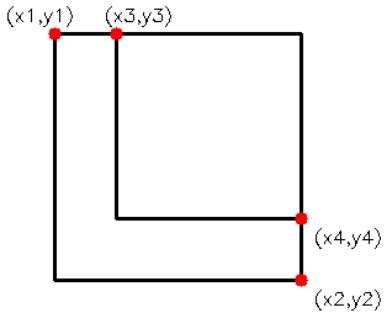

図13　□ (U+2FFA)

3.4.12　⿻ (U+2FFB)

⿻ (U+2FFB) の Unicode 文字名は「overlaid（重ねる）」であるため，部品 1 の右下座標 x_2 は部品 3 の左上座標 x_4 より小さい．部品 1 の右下座標 y_2 は部品 3 の左上座標 y_4 より小さい．部品 1 の右下座標 x_2 は部品 3 の左上座標 x_4 より小さい．部品 1 の右下座標 y_2 は部品 3 の左上座標 y_4 より小さい（図14）．

$$x_1 < x_3 \wedge y_1 < y_3 \wedge x_2 < x_4 \wedge y_2 < y_4 \tag{16}$$

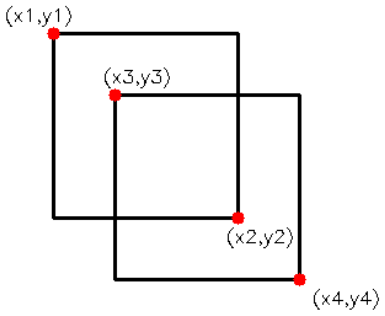

図14　⿻ (U+2FFB)

4　実験

CHISE 漢字構造情報データベース[*6]の‘IDS-UCS-Basic.txt’を検証データとして，生成した IDC と比較する．データの前処理として，以下の条件に合致するサンプルを除外する．

- IDS が「一」，「丁」など，自身の漢字であるもの
- GlyphWiki 字形が引用部品を持たず，筆画行のみで構成されたもの
- 引用部品数が検証データの IDC に対応しない字形[*7]

[*6] https://gitlab.chise.org/CHISE/ids.git
[*7] 引用部品と筆画行からなる字形は，すべての筆

実験の結果を表1に示す．全体的に正確率は 75.05% であるが，⿲，⿳，⿴，⿵，⿶，⿷，⿻はほぼ認識できなかった．これらの IDC は，生成手法を再考する必要がある．

表1　検証データ

IDC	サンプル数	正確数	正確率
⿰	14,350	10,690	74.49%
⿱	3,718	3,154	84.83%
⿲	4	0	0.00%
⿳	31	0	0.00%
⿴	115	0	0.00%
⿵	190	1	0.53%
⿶	12	0	0.00%
⿷	39	0	0.00%
⿸	629	523	83.15%
⿹	110	65	59.09%
⿺	446	312	69.96%
⿻	2	0	0.00%
合計	19,646	14,745	75.05%

5　おわりに

本稿では，本稿では，GlyphWiki の字形データを利用して漢字構造情報を生成する手法を提案した．提案手法では，部品の配置座標を取得し，部品の関係を判断することで，漢字構造情報を生成した．実験の結果，全体的な正確率は 75.05% であったが，⿲，⿳，⿴，⿵，⿶，⿷，⿻はほぼ認識できなかった．今後は，これらの IDC を正確に認識する手法を考案することが課題である．

画行を一つの部品として扱う．また，GlyphWiki では「エイリアス」である字形，つまり「99:0:0:0:0:200:200: 引用グリフ名」の一行からなった字形は，引用した字形の KAGE データを取得する．

参考文献

[1] The Unicode Consortium. *The Unicode Standard, Version 15.1.0*. The Unicode Consortium, South San Francisco, CA, 2023.

[2] 守岡知彦．漢字構造変換の試み．じんもんこん 2020 論文集, 第 2020 巻, pp. 197–202, dec 2020.

[3] Glyphwiki: 高度な活用方法．https://glyphwiki.org/wiki/GlyphWiki:%e9%ab%98%e5%ba%a6%e3%81%aa%e6%b4%bb%e7%94%a8%e6%96%b9%e6%b3%95, accessed: 2024-07-06.

[4] 上地宏一．Kage - an automatic glyph generating engine for large character code set．「書体・組版ワークショップ」報告書, 第 2004 巻, pp. 85–92, feb 2004.

[5] Glyphwiki:kage データ仕様．https://glyphwiki.org/wiki/GlyphWiki:KAGE%e3%83%87%e3%83%bc%e3%82%bf%e4%bb%95%e6%a7%98, accessed: 2024-07-06.

[6] Takahito Yamada. CHISE IDS 漢字検索の使い方―中国史研究のためのデジタルリソース入門. https://www.shuiren.org/chuden/toyoshi/syoseki/chise_ids.html, accessed: 2024-07-06.