Vol.2011-DD-79 No.8
2011/1/21

Type1 埋め込みされた TrueType フォントの同定手法の考察

鈴　木　俊　哉[†1]

PDF ドキュメントは TrueType フォントをそのまま埋め込むことができるが、TrueType ラスタライザを持たない処理系や、フォントの抽出・転用を防ぐために PostScript Type1 形式に変換して埋め込まれることがある。Type1 変換された TrueType フォントはフォント名や文字コード符号位置が元の TrueType と異なるため、使用されたフォントが本来は何であったかを特定することが非常に困難となる。ISO/IEC 10646 規格票の漢字表改訂作業を通じて得られたフォントおよびグリフ同定手法の知見について報告する。

A Study of Font Identification for Type1 Converted TrueType in PDF

SUZUKI TOSHIYA[†1]

Although PDF can include TrueType font in it since its version 1.3, there are many PDF including PostScript Type1 fonts that are converted from TrueType font. There are a few motivations for such conversions: the portability with PDF rendering systems without PostScript rasterizer (today, the most of the desktop computing environments have TrueType rasterizers, but the number of legacy printers without TrueType rasterizers are still non-negligible), the prevention of the font piracy by the extraction of the embedded fonts. When TrueType font is converted to PostScript Type1 format, often the original font family names and the original codepoints assigned to the glyph are removed. Therefore, it is quite difficult to identify which font is used and what string is rendered. This report summarizes a study based on the experiments for the reviewing process for CJK Unified Ideograph charts for ISO/IEC 10646.

†1 〒 739-8511 東広島市鏡山 1-4-2　広島大学大学院総合科学研究科
Faculty of Integrated Arts and Science, Hiroshima Univ., Kagamiyama 1-4-2, Higashi-Hiroshima-shi, 739–8511 Japan

1. 背　　景

通常、デジタルドキュメントに用いられているフォントを特定する必要が出てくるのは以下のような特殊な環境である。

- ハードコピーや編集不可能な文書を再度編集する必要があり、データ入力し直さなければならない場合。
- 文書中に一般の情報交換の中では用いられない文字が外字等で使われており、その出所を明らかにしなければならない場合。

編集可能なデジタルドキュメントの多くはフォントを埋め込んでおらず[⋆1]、フォントをフォントファミリ名によって参照しているものが多い。

再編集可能なの文書形式では、表示されている文字列は以下のような属性情報を持っている。

- 符号化文字列
- 書式
  - フォント (フォントファミリ名により指定する)
  - フォントスタイル (太字、斜体など)
  - 文字サイズ
  - 行書式 (行幅、行間、インデント)

書式つきテキスト編集のためのライブラリは、テキスト表示の際にこれらの情報全てを揃えた上で表示しているので、部分的なコピー・ペーストを行なった場合にも書式情報が伝播し、フォント指定も受け継がれる。

一般的な文書では、見出しと本文の 2 種類程度のフォントしか使い分けず、さらに多数のフォントを用いる理由は以下のような特殊な場合が多い。

- 言語研究や辞書など、本文とは異なる文字集合のテキストを多数挿入する場合。
- 広告や書籍表紙などのデザイン的な性格が強い文書において、単語の印象づけを変えるために書体の印象を変えたい場合。

前者の場合は印字する文字列、後者の場合は書体の印象を手掛りにフォントを特定することができ、字形の詳細に踏み込む必要はない。

これに対し、国際文字符号 ISO/IEC 10646 の規格票における漢字票は、図 1 に示すように、中国・台湾・香港、日本、韓国、ベトナムなどが実装の際に参照されることを念頭に置いて字形の詳細に配慮したフォントを提出し、これによって漢字表が印刷される[1)]。

ISO/IEC 10646 の印刷においては、多くの場合、新規に追加される漢字があるが、標準化作業の過程で文字の提案取り下げがありうるため、規格票を印刷するフォントはその規格

⋆1 OOXML など文書形式仕様はフォントを埋め込めるが実際の処理系は埋め込まなかったり、埋め込んでも無視するものは少なくない

Row/Cell	C		J	K	V
Hex Code	G-Hanzi-T		Kanji	Hanja	ChuNom
143/208	运	运	运	运	
8FD0	0-544B	4-2826	1-613D	2-6555	
	0-5243	4-0806	1-6529	2-6953	
143/209	近	近	近	近	近
8FD1	0-3D7C	1-4F54	0-3661	0-504E	1-6929
	0-2992	1-4752	0-2265	0-4846	1-7309
143/210	远	远	远	远	
8FD2	3-7159	2-286C	1-613E	2-6556	
	3-8157	2-0876	1-6530	2-6954	
143/211	迓	迓	迓	迓	
8FD3	0-6542	2-286E	1-613F	1-662A	
	0-6934	2-0878	1-6531	1-7010	
143/212	返	返	返	返	返
8FD4	0-3735	1-4F53	0-4A56	0-5A77	1-692A
	0-2321	1-4751	0-4254	0-5887	1-7310

Row/Cell	C		J	K	V
Hex Code	G-Hanzi-T		Kanji	Hanja	ChuNom
143/224	迠	迠	迠	迠	
8FE0	E-4034	2-2D57	1-6141	2-6559	
	E-3220	2-1355	1-6533	2-6957	
143/225	迡	迡		迡	迡
8FE1	5-702C	2-2D55		2-655A	1-692C
	5-8012	2-1353		2-6958	1-7312
143/226	迢	迢	迢	迢	迢
8FE2	0-4C76	1-535F	0-6D6B	1-6F7C	1-692D
	0-4486	1-5163	0-7775	1-7992	1-7313
143/227	迣	迣	迣	迣	
8FE3	3-715A	2-2D54	1-6142	2-655B	
	3-8158	2-1352	1-6534	2-6959	
143/228	迤	迤	迤	迤	
8FE4	0-6546	1-5364	1-6143	1-6A57	
	0-6938	1-5168	1-6535	1-7455	

Row/Cell	C		J	K	V
Hex Code	G-Hanzi-T		Kanji	Hanja	ChuNom
143/240	述	述	述	述	述
8FF0	0-4A76	1-535D	0-3D52	0-627B	1-6930
	0-4286	1-5161	0-2950	0-6691	1-7316
143/241	迮	迮	迮	迮	
8FF1	5-702B	3-2F30	1-6146	2-655F	
	5-8011	3-1516	1-6538	2-6963	
142/242	迭			迭	
8FF2	1-7D25			0-4C26	
	1-9305			0-4406	
143/243	迳				
8FF3	0-6549				
	0-6941				
143/244	迴	迴	迴	迴	迴
8FF4	1-783C	1-584F	0-6D6F	1-7451	1-6931
	1-8828	1-5647	0-7779	1-8449	1-7317

図 1　ISO/IEC 10646:2003 の統合漢字表

票が定義する符号位置に例示字形を持たないことが多い (取り下げが 1 文字であっても、多数の文字の符号位置がくり上がるなどが生じるため)。

基本的には、ISO/IEC 10646 の漢字表は新規に追加されたものだけを印字すれば良い筈であるが、ISO/IEC 10646 規格票や Unicode 規格票を参照して実装するベンダが増えたため、規格票の例示字形に規範性を求める圧力が高まり、過去の漢字表で用いたフォントをさしかえたいという要求も多い。ISO/IEC 10646 で文字表を改訂した場合、図 2 に示すように、多くの表は数文字の改訂であってもブロック全体を追補で印刷しなおすが、漢字表だけは膨大であるため差分のみが追補に含めることが多い。

これらの例示字形変更は逐一 ISO/IEC JTC1/SC2/WG2/IRG でレビューが行なわれ、統合範囲を越えたり、他の符号位置の例示字形と混乱を招かないか確認した上で変更の可否が決定されることになっている。しかし、漢字表を組んでいるのは IRG ではなく SC2/WG2 のプロジェクトエディタであるので、最終的に提出したフォントでどのような字形変更が行なわれているのかは IRG では確認できない。そのため、現在はプロジェクトエディタが PDF で印刷した漢字表を IRG にさしもどし、IRG のメンバがレビューするという構造になっている。

1.1 PDF へのフォント埋め込み手法

TrueType フォントの PDF への埋め込みには以下のような方法がある。

- TrueType 描画プログラムのまま埋め込む
  - 255 個ずつのグリフごとにフォントオブジェクトを生成し、8 ビット単位の Type42 フォントとして埋め込む。
  - 文字符号とグリフ変換表も作成し、この変換表を参照する 16 ビット単位の Type11 フォントとして埋め込む

図 2　修正に際して 1 文字だけの修正でも表全体を再印刷する例 (ISO/IEC 10646:2003/Amd.1:2005 の Latin Extended B) と、追加分だけ付記する例 (ISO/IEC 10646:2003/Amd.1:2005 の CJK Unified Ideographs の JIS X 0213:2004 対応)

- PostScript 描画プログラムに変換し、グラフィクスとして埋め込む
  - Type1 アウトライン描画プログラムに変換し、255 個ずつのグリフごとに 8 ビット単位の Type1 フォントとして埋め込む
  - Type2 アウトライン描画プログラムに変換し、8 ビットの CFF フォントまたは 16 ビット単位の Type9 フォントとして埋め込む
  - ビットマップに変換し、255 個ずつのグリフごとに 8 ビット単位の Type3 フォントとして埋め込む

大別して TrueType 描画プログラムのまま埋め込む方式と、PostScript 描画プログラムに変換して埋め込む方式がある。ただし、「8 ビット単位の TrueType を 16 ビット単位の Type11 フォントとして埋め込む」「8 ビット単位の TrueType フォントを 16 ビット単位の Type9 フォントとして埋め込む」というようなグリフ数を拡大する方向での変換を行なう実装はほとんどない。また、Type3 フォントは PDF の描画命令を全て処理しなければならないためフォントラスタライザのようなコンパクトな処理系による高速な描画ができない。そのため、PostScript データを変換して PDF を生成する際に元の PostScript データに含まれていた Type3 フォントを埋め込む以外の用途で用いられることは殆んどない。

PostScript および PDF は当初 TrueType フォントを含むことができなかったため[*1]、まず後者が最初に実装され、後に TrueType フォントを埋め込むようになった。後者の方式は以下の難点があるので[2]、処理系の制限がない限り現在では推奨されていない。

- TrueType 描画プログラムに書き込まれているヒント情報をそのまま PostScript 描画命令に翻訳できないため、中・低解像度での表示品質が元のフォントよりも低下する。
- TrueType 描画プログラムでの曲線描画は 2 次スプライン曲線であるのに対し、PostScript 描画プログラムでの曲線描画は 3 次ベジエで曲線であるため、正確に翻訳できないため、ヒントがないグリフでも完全に同一な図形を描画できない。

そのため、処理系も「漢字などの大規模文字集合の TrueType フォントを Type9 フォントとして埋め込む」という手法は提供しているものは殆どない[*2]。Type1 フォントはマルチバイト文字符号とグリフの対応表を持てないため、Type1 埋め込みした場合、元の文書で用いていた符号化文字列は保存されない。また、多くの処理系では Type1 フォントを生成する際に元の TrueType フォントのファミリ名を反映せず、全て記号的なフォント名に書き変えられる[*3]。フォント名および文字符号が保存されないため、Type1 埋め込みされたフォントを特定することは非常に難しくなる。

さらに、TrueType フォントの場合は合成グリフ (たとえば、ローマ字アルファベットとアクセント記号を別々のグリフとして格納しておき、アクセント記号つきアルファベットはこの 2 文字を合成した図形を描画するという仕組) があり、複数のグリフを重ねて描画する場合も考慮されている[*4]が、PostScript フォントでは描画命令の一部を共用するためのサブルーチンしかなく、複数のグリフを重ねて描画した場合の特殊な扱いは考えられていない。このため、何も考えずに TrueType フォントの合成グリフを Type1 描画プログラムに翻訳すると、要素グリフの重なりが発生するため、処理系によっては白抜けなどが発生する。

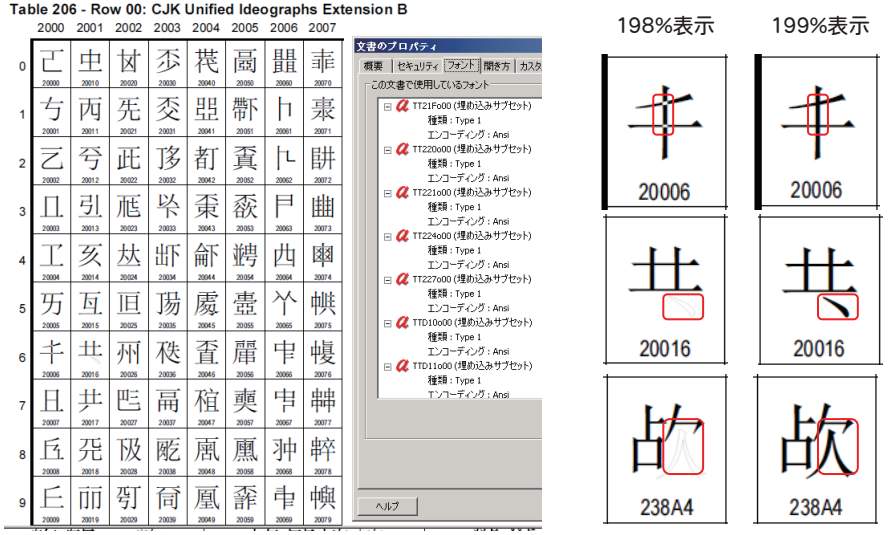

図 3 ISO/IEC 10646:2003 の CJK Unified Ideographs Extension B の埋め込みフォントと、合成グリフのバグによる白抜けの例

2. ISO/IEC 10646 規格票と例示字形のフォント特定

2.1 ISO/IEC 10646 規格票の漢字表の構造

ISO/IEC 10646 規格票 PDF 版[*5]におけるフォントは、出版当初から TrueType 埋め込みではなく、Type1 埋め込みである。特に、2003 年版は CJK Unified Ideographs Extension B の追加により短期間で作成された 2 万字の漢字フォントが埋め込まれ、レビューが不完全だったため図 3 に示すような上記の合成グリフの問題が発生した[*6]。

2003 年版は長い間参照されてきたが、漢字表については以下の要望があり、ISO/IEC 10646 の改訂第 2 版、第 3 版に向けて変更作業が進められている。

- 本来の統合漢字 (URO) や、CJK Unified Ideographs Extension A のマルチカラム表で示される各提案元の例示字形について、出版後の国内規格改訂により追加すべき文字や、字形変更などが希望されている。多くの文字では、変更が行なわれたブロック全体

⋆1 TrueType フォントは PostScript によらないスケーラブルフォント技術として PostScript level2 以降に開発されたものなので当然と言える。

⋆2 これは PostScript に変換して埋め込むという方式が TrueType ラスタライザを持たない処理系への後方互換にすぎないという位置付けの他に、初期の PostScript 処理系はマルチバイト文字符号の処理のための文字符号位置-グリフ番号対応表を持つという仕組を欠いており、Type9 フォントを埋め込んでも処理できないという問題があるためと考えられる。これらの初期の PostScript 処理系ではマルチバイト文字符号からの文字切り出しを状態遷移プログラムによって処理していたため、フォントのサブセット化と同時にプログラムを生成しなければならず、非常に困難である。

⋆3 ページごとに個別のフォントオブジェクトに分割して表示速度を向上させるなどの目的で、1 個の TrueType フォントから多数の埋め込みフォントオブジェクトが生成することが多く、この場合元のフォント名をそのまま用いることはできない。アドビによる実装では、Type42 または Typ11 埋め込みする場合には元のフォント名に乱数を追加して埋め込み、また、ページごとに参照している埋め込みフォントを切り換えた仮想フォントに元のフォント名を付加するなどして元の TrueType フォント名をできる限り維持しようとしている。Type1 埋め込みした場合にはこのような配慮は為されない。

⋆4 TrueType の glyf テーブルの合成グリフ用フラグの中にオーバーラップ制御のフラグが存在する。

⋆5 PDF 仕様および Adobe 実装がが CJK TrueType フォントの埋め込みに正式対応したのは 2001 年なのに対し、ISO/IEC 10646 の PDF 版の出版は 2003 年からで、PDL ではなく電子文書として見た場合には TrueType フォント埋め込みは既に一般的になっていたと言える。

⋆6 当時、合成グリフの白抜け問題が広く認識されていなかったため、白抜きの状態が規格が定義するグリフだと誤解される場合もあった[3]

の文字表が提供されているが、漢字の場合は表が巨大すぎるため、追加・変更した文字のリストだけが追補に分散しており、見づらい。

- CJK Unified Ideographs Extension B もマルチカラムとしたい。
- 統合漢字 (URO) や、CJK Unified Ideographs Extension A のマルチカラム表の一部では、CJK Unified Ideographs Extension B 以前の漢字表のために作成された画像データで印刷されており、現在では品質が低く見える。アウトラインであっても品質は低い。
- 統合漢字 (URO) や、CJK Unified Ideographs Extension A のマルチカラム表での字喃フォントは台湾フォントをそのままコピーしており、ベトナムの参照字形としては不適切。ベトナムでもフォントを内製できる環境が整いつつあり、さしかえたい。

漢字表は本来 JTC1/SC2/WG2 の漢字関連 Working Group である IRG でメンテナンスされるもので、拡張 B のマルチカラム化作業も IRG で作業準備を進めていたが[4]、近年では IRG が CJK Unified Ideographs C, D などの標準化に注力したため、2007 年以降進展していなかった。そのため、漢字表の組版作業は ISO/IEC 10646 のプロジェクトエディタが直接行なうこととなり、各提案者はフォントをプロジェクトエディタに提出することとなった[⋆1]。

これと同期して、マルチカラム漢字表の欄として、中国、台湾、日本、韓国、ベトナムが列挙されるが、CJK Unified Ideographs Extension A、CJK Unified Ideograpsh Extension B などは提案元が 2,3 のものが大半であり、単純に列挙すると空欄の割合が増えるため、マルチカラム漢字表といっても図 5 のように空欄は作らないよう圧縮し、提案元の情報は例示字形の下の典拠情報によって識別することとなった[⋆2]。漢字によっては提案元が多数であるため折り返し表示となり、漢字表は符号位置あたり 1 行消費するという構造ではなくなった。さらに、図 4 行間について開始行と継続行で異なるために機械的な分解が著しく困難となっている。

3. ISO/IEC 10646 第 2 版漢字表のレビュー作業の難点

ISO/IEC 10646 第 2 版では、URO および CJK Unified Ideographs Extension A の表構造は上記のように変更されるが[⋆3]、CJK Unified Ideographs Extension B については図 6 のように表構造は維持し (正確には第 1 版では 1 ページあたり 128 字だったものを Unicode

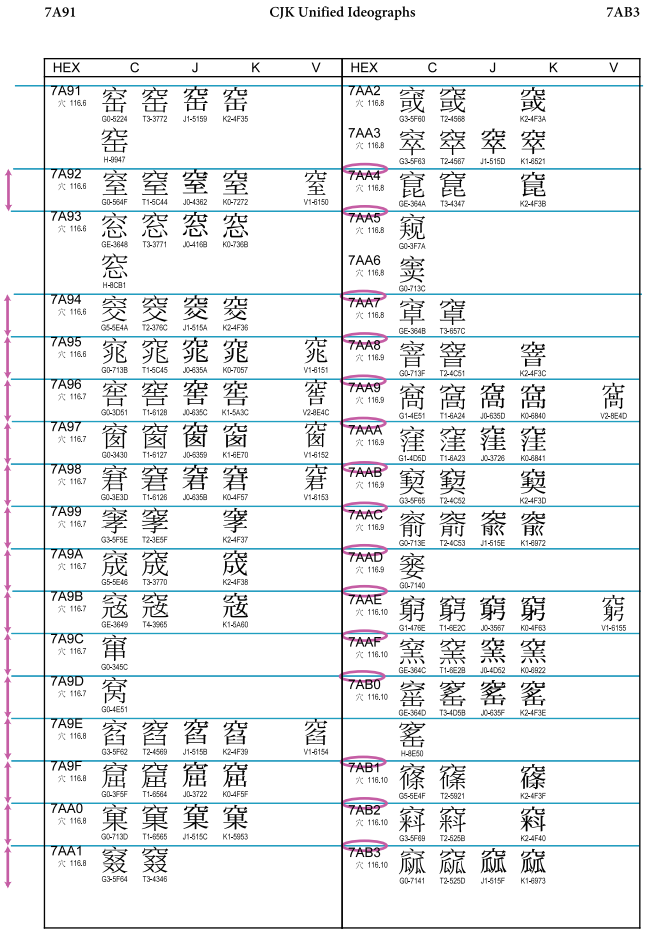

図 4　折り返しを導入したことによる漢字表の構造崩れ

の漢字表と同様に 1 ページあたり 256 字に変更しているが基本的な構造は同じと言える)、印刷に用いるフォントのみ 2003 年版以降の追補を反映した字形に変更することとなった。しかし、IRG は 2003 年版の CJK Unified Ideographs Extension B を印刷したフォントも、今回の印刷に用いるフォントも持っておらず、本当に変更点が追補で定義されたものだ

⋆1 日本は IRG が漢字表をメンテナンスするべきで、作業が間に合わない以上は古い漢字表をそのまま用いることが規格の安定のために望ましいと主張したが、受け入れられなかった。

⋆2 CJK Unified Ideographs Extension A は 1 符号位置あたり 3 カラム、CJK Unified Ideographs Extension B は 1 符号位置あたり 2 カラムとなっている。

⋆3 当初は、URO についても中国、台湾、日本、韓国、北朝鮮、ベトナム、香港の字形を折り返して列挙する予定であったが、北朝鮮の担当者と連絡がとれない状態が 5 年以上続いてフォントが提出されなかったため、過去に提出された北朝鮮提案漢字との典拠情報は残すが漢字表の北朝鮮欄は削除された。北朝鮮互換漢字に関しては維持されている。

図 5 CJK Unified Ideographs Extension A の形式

図 6 ISO/IEC 10646:2010(第 2 版) の CJK Unified Ideographs Extension B の形式

けなのか根本的な確認をすることはできない[⋆1]。フォントの大半は変更されていないと仮定し、第 1 版と第 2 版の CJK Unified Ideographs Extension B の漢字表を 300dpi でビットマップ化し比較すると、以下のような差が見られた[⋆2]

- グリフあたりの異なりピクセル数 0 個である文字 211 字
- グリフあたりの異なりピクセル数 1 ～ 49 である文字 34203 字
- グリフあたりの異なりピクセル数 50 ～ 99 である文字 6764 字
- グリフあたりの異なりピクセル数 100 個以上である文字 1524 個

ここで、ISO/IEC 10646 Amd.1 のように明確に字形を修正した場合の異なりピクセル数は 100 を越える (たとえば、図 7 に示すような修正の場合、U+20BF6 は 299 個、U+21BA7 は 796 個、U+21E45 では 586 個)。従って、大半の文字について統合範囲を越えないような微細な字形修正が行なわれていると予想された。

CJK Unified Ideographs Extension B のフォントはシンニョウの字形などから明らかなように、特定の国の規範に従っているわけではないので[⋆3]、特定の国の市場を考慮してフォントを修正したとは考えられない[⋆4]。しかし、異なりピクセル数の数によって統合範囲を越える修正かどうかを判断する材料が少ないため、アウトラインの比較を試みた。

第 1 版と第 2 版の PDF に埋め込まれている Type1 フォントを抽出して比較した結果、以下の違いが見つかった。

- グリフ名が異なっている
- アウトラインの回転方向が全て逆転している
- 閉曲線の描画において、最終制御点への移動後に初期座標位置に自動的に戻る機能を利用するのではなく、最終制御点を初期描画点と同一位置において閉曲線を描画する
- 300dpi ビットマップには現われないような微細な制御点移動 (300dpi で 1/4 ピクセル程度) がほぼ全てのグリフにある
- アウトラインのパス構築命令が相対座標系であったものが絶対座標系になっている

特にグリフ名が第 1 版と第 2 版で異なっているため、同一符号位置に印字される例示字形

In the CJK Unified Ideographs Extension B code table, replace the graphic symbol for the following entries:

UCS value	10646-2: 2001	10646: 2003	New graphic symbol
20BF6	冃	冃	冃
21BA7	罶	罶	罶
21E45	亜	亜	亜
23031	數	數	數
230D4	斷	斷	斷
25962	窒	窒	窒
25ACD	龕	龕	龕
26165	縄	縄	縄
2630B	罊	罊	罊
264AB	鼜	鼜	鼜
26CD8	蒼	蒼	蒼
285ED	遼	遼	遼
29FCE	鴉	鴉	鴉

図 7 ISO/IEC 10646:2003/Amd.1:2005 の CJK Unified Ideographs Extension B の修正

⋆1 漢字表の改訂において、規格で定義される「統合可能な字形差」の範囲を越える変更に関しては問題視されるが、統合可能な範囲の変更については議論する場がない。

⋆2 この調査は日本 SC2 委員会の関口正裕による。

⋆3 各国から提出されたものではなく、拡張 B の例示字形フォントは複数のフォントベンダに発注された。統合範囲よりも細かなデザイン整合性はとられていない。

⋆4 たとえば中国国内規格の GB 18030:2000 では全て中国の標準字形デザインに揃えたもので印刷されている。

を特定して比較することが困難であった。グリフ描画プログラムを以下のようなオブジェクトへ変換することにより特定を行なった。

- contour(**閉曲線**) 閉曲線を為す制御点の二次元座標を列挙したもの。位置関係を相対ベクトルを列挙する。同一制御点は削除する。描画の順序に関係なく、座標中もっとも座標原点に近いものを初期座標とする。
  - **制御点数** 閉曲線を為す制御点の個数を返す
  - **閉曲線比較** 別の閉曲線オブジェクトを与えた場合、初期座標から順に各制御点を比較し、位置のずれが指定された誤差範囲内であることを確認する。
- path(パス) 複数の閉曲線を持つオブジェクト
  - **パス比較** 別のパスオブジェクトを与えた場合、構成する閉曲線数が同一であれば、閉曲線の制御点数が同一である閉曲線を比較し、ずれが指定された誤差以内であることを確認する。

制御点の 0.01em 未満の差異を許すとして U+28C00~U+299FF の区間 (3583 文字) を調査すると、494 個のグリフに対して違いが検出された。その例を図 8 に示す。

上で列挙した差異、特にアウトラインの回転方向の逆転と、300dpi ビットマップでは現われないような微細な制御点移動が多数あることは、意図的なフォントの修正の結果とは考え難い。考えられる要因として、第 1 版の作成以降に TrueType フォントを Type1 変換するアルゴリズムが変更され[⋆1]、結果としてグリフ描画プログラムも変わったと考えられる。

4. ISO/IEC 10646 第 3 版に向けた作業と今後の課題

ISO/IEC 10646 第 3 版では CJK Unified Ideographs Extension B もマルチカラム化され、図??に示すような折り返し整形表となる。

URO および CJK Unified Ideographs Extension A は各提案元の国内規格を典拠とするため、ある程度安定した集合と考えられるが、CJK Unified Ideographs Extension B は康熙字典の見出し字を全て個別の文字として符号化するという動機があったため、字形の安定性について以下のような問題点がある。

- 直接に康熙字典を典拠として提案しているのは中国だが、中国が提出するフォントは中国の印刷標準字形に則るようにデザインされている。
  - シンニョウや草冠の画数など、日本で当用漢字以降の新字体と区別するという意味での「康熙字典字形・字体」と言う場合に期待されるデザイン的な特徴がない場合が多い。
  - 現代漢字として殆ど用いられない部首は印刷標準字形が定義されていないが、それらの部首のデザインについて中国が提出するフォントも康熙字典字形と完全に一致するデザインではない。

⋆1 True1 埋め込みされたフォントの命名規則から判断すると Adobe PS driver を使用していると思われる。

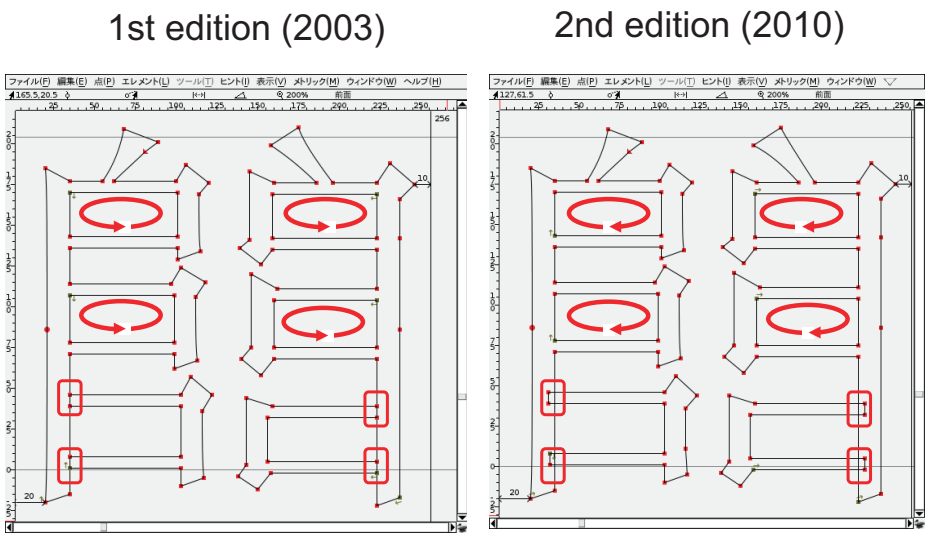

図 8 ISO/IEC 10646:2003(第 1 版) と ISO/IEC 10646:2010(第 2 版) の CJK Unified Ideographs Extension B の漢字アウトラインデータの差異例

- CJK Unified Ideographs B には台湾のローカル規格である CNS 11643:1992 からも多数収録された。ISO/IEC 10646 上はその典拠は CNS 11643:1992 としているが、CNS 11643 策定時には康熙字典から採録していたと思われるものも多く、実際には康熙字典の同一項目を参照しながら、上記の中国提案字形とデザイン差があり別個に符号化されてしまっているものが少なくない。

前者の「康熙字典典拠とされているが康熙字典字形と異なる」問題は、CJK Unified Ideographs Extension C にも中国から康熙字典典拠での漢字が再度提案されるなどの問題を引き起こしており、先頃マカオ会議で「康熙字典と異なる字形になっているものについて記録をとる」ことが合意された。後者は CNS 11643 の典拠についての議論は ISO/IEC 10646 の担当範囲外ということもあり[⋆2]、典拠との差異を先に調査しておくことはできず、台湾から提出されるフォントの字形変更としてレビューしなければならない。

例示字形 3 個で折り返し整形する表のため、今後、典拠が移動・追加された場合 (たとえば現在では中国提案しか典拠がないが、日本や韓国から別の典拠のものが統合可能として追

⋆2 中華民国教育部の精査により CNS 11643:1992 の誤字を修正したものが CNS 11643:2007 として公布された。IRG は国際工業標準であって漢字研究ではないので、「典拠の典拠」を追跡して字形の正確さを議論することはスコープ外ということもあるが、CNS 11643 の各符号位置の漢字の典拠は明らかにされていないので、何を参照して字形を修正したのか、IRG では議論ができない。

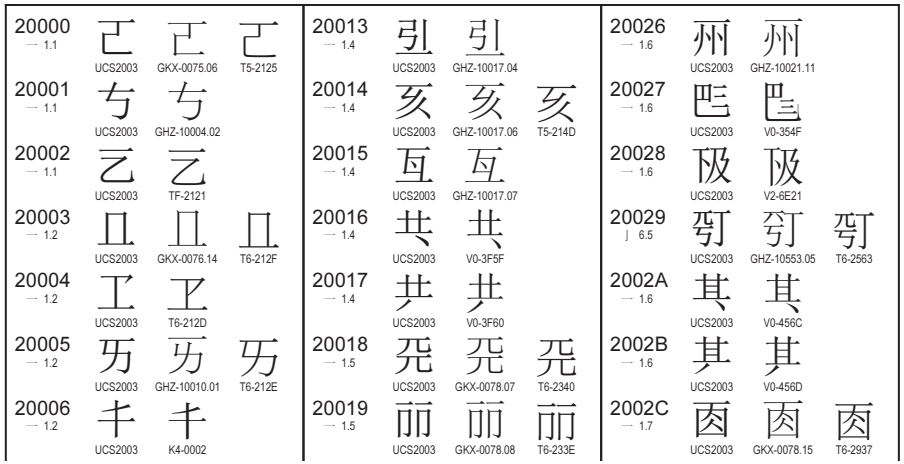

図 9 ISO/IEC 10646:2012(第 3 版) の CJK Unified Ideographs Extension B の形式

加されるなど)、表示位置の変更が多数の位置に及ぶ可能性が高い。従って、作業中の漢字表の比較は同一ページ同一位置の字形の図形比較としては困難で、どの文字は、ページ上のどの位置で、どのフォントのどのグリフで表示されているのか正確に把握しなければ機械的な前処理ができない。

これは PDF からフォントオブジェクトのみを抽出しても解決できず、文書構造を反映してアウトラインを抽出しなければならない。PDF は描画時の座標変換が可能であり、また、文字描画の際にも様々な位置指定が可能であり、テキスト描画は必ずしも一括して同じ座標空間で書かれるわけではないので、文字を描画する命令の近辺を解析するだけでは描画している文字の位置を把握できない。特に、ISO/IEC 10646 規格票でも文字表の部分はアクセシビリティに配慮した文書ではないので、描画順序などは印刷用途に最適化された状態であり、あくまでも PDF 全体の構造を解析した上で位置を判定する必要がある。現在、PDF のテキスト化プログラムである pdftotext[5] を拡張することで解決を目指している。

謝　　辞

本研究は科学研究費補助金 若手研究 (B) 課題番号 21700113 の補助を受けました。

参 考 文 献

1) ISO/IEC JTC1/SC2: *ISO Standards: Information Technology – ISO/IEC 10646:2003, Universal Multiple-Octet Coded Character Set (UCS)*, ISO (2003).
2) Adobe Systems Inc.: *Adobe Technote 5012: The Type42 Font Format Specification*, Adobe Systems Inc., San Jose (1998). http://partners.adobe.com/public/developer/en/font/5012.Type42_Spec.pdf.
3) アンテナハウス：PDF 千夜一夜: PDF と文字 (19) - 漢字統合問題再検討. http://blog.antenna.co.jp/PDFTool/archives/2006/01/10/.
4) Group, I.R.: *IRG N1381: Ext. B Visual Reference Table.* http://appsrv.cse.cuhk.edu.hk/~irg/irg/irg29/CJKB_0601-0700.pdf.
5) Cid, A. A.: *Poppler, a PDF rendering library based on the xpdf-3.0 code base.* http://poppler.freedesktop.org/.