透明テキスト付き画像作成ツールの開発

安岡孝一*

1　はじめに

安岡と守岡が提唱した透明テキスト付き画像 [1, 2, 3, 4] は、東洋学の分野における有用性に関しては高い評価を受けているものの、その作成が難しいという点が問題となっている [5]。PDF や SVG を直接書く能力のある人にとっては、アイデアさえ理解できれば、透明テキスト付き画像を作るのは造作もないことだ [4] が、そのような能力のある人は、ごく少数である。透明テキスト付き画像の普及を考えるなら、グラフィカルユーザインターフェースを有する簡便な作成ツールは、必須のものだといえる。

そこで安岡は『透明テキスト付き画像作成ツール (漢文用) `ttext-kanbun`』を開発した。`ttext-kanbun` は、漢文の画像ファイルとその釈文をもとに、透明テキスト付き画像の作成をサポートするツールであり、Microsoft Windows XP 上の Internet Explorer 6 で動作する。出力は、透明テキストを含む SVG あるいは PDF である。以下では、`ttext-kanbun` の概略と、その出力仕様について述べる。

2　`ttext-kanbun` の使用法

`ttext-kanbun` は Microsoft Windows XP 上の Internet Explorer 6 で動作する。Microsoft Windows 2000 上の Internet Explorer 5.5 でも動作が確認されているが、作者としてはちょっと自信がない。Microsoft Windows Me 上の Internet Explorer 5.5 でも動くには動くようだが、使用できる漢字が JIS X 0208 と JIS X 0212 のものに限定されるように思われる。

下準備

最初に `http://kanji.zinbun.kyoto-u.ac.jp/~yasuoka/ftp/program/` から、`ttext-kanbun.lzh` を取り寄せて解凍する。中身は `editor.html` と `frame0.html` と `frame1.html` と `manual.pdf` である。

New Gulim フォント†もしくは SimSun (Founder Extended) フォント‡をインストールする。ただし、釈文に JIS X 0208 と JIS X 0212 に含まれる漢字しか使用しない場合は、インストールの必要はない。なお、SimSun (Founder Extended) フォントの CJK 統合漢字拡張 B を使用する場合は、Internet Explorer にサロゲートペア

*京都大学人文科学研究所附属漢字情報研究センター

†New Gulim フォントは、Microsoft Office XP の韓国語パッケージに含まれている。CJK 統合漢字拡張 A までが表示可能である。なお、New Gulim フォントをインストールすると、外字が全て化けてしまうという不具合が報告されている。

‡SimSun (Founder Extended) フォントは、Microsoft Proofing Tools の Chinese (Simplified) パッケージに含まれている。CJK 統合漢字拡張 A および B の多くが表示可能である。

パッチ http://www.microsoft.com/downloads/release.asp?ReleaseID=31114 をあてておく必要がある。

画像ファイルを準備する。jpg、png、gif のいずれでもかまわないが、とりあえずは jpg をオススメする。釈文も準備する。釈文は紙の形であってもかまわないし、テキストファイルとして入力してあってもかまわない。ただし、テキストとして準備する場合は、句読点は全角の「。」と「、」(Unicode の 3002 と 3001) で表し、割注は半角の丸カッコ (Unicode の 0028 と 0029) で囲んでおく。

起動と画像読み込み

editor.html のアイコンをダブルクリックして、ttext-kanbun を起動する。load image ボタンをクリックして、画像ファイルを選択する (図 1)。

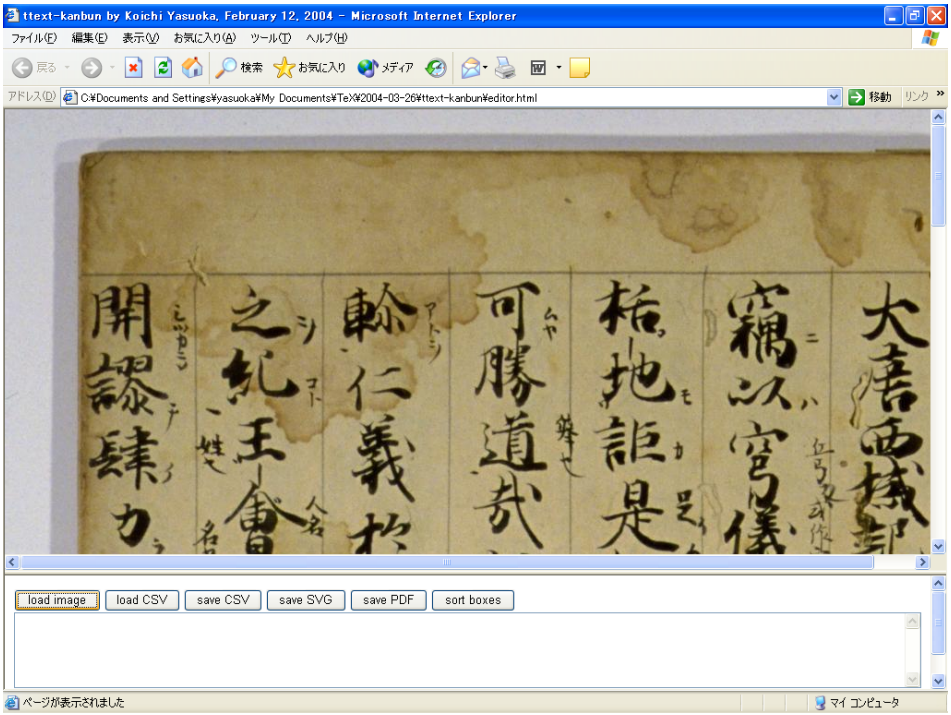

図 1: 画像読み込み

釈文の入力

最下段のテキストボックス中に、釈文を入力する (図 2)。テキストファイルからのカットアンドペーストでもかまわない。句読点は全角の「。」と「、」(Unicode の 3002 と 3001) で表し、割注は半角の丸カッコ (Unicode の 0028 と 0029) で囲んでおく。文字実体参照のうち、10 進数表記 (&#ddddd;) と 16 進数表記 (&#xhhhh;) が、釈文に使用可能である。ただし、CJK 統合漢字拡張 B に対する文字実体参照は、𠀀〜𪛖 あるいは 𠀀〜𪛖 を使用し、サロゲートペアは用いない。釈文中の改行は無視される。

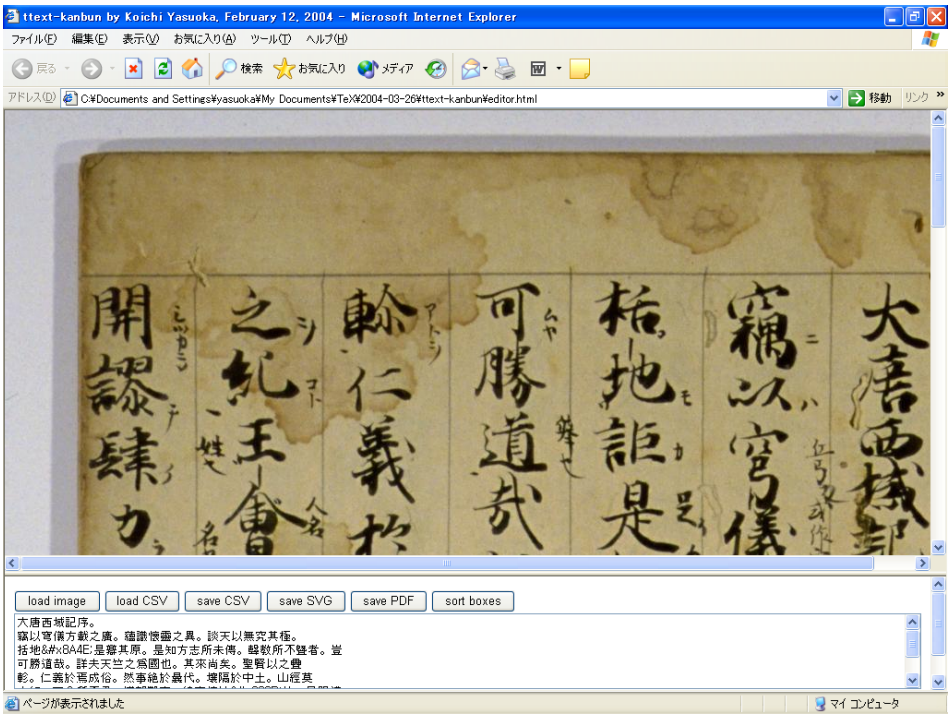

図 2: 釈文の入力

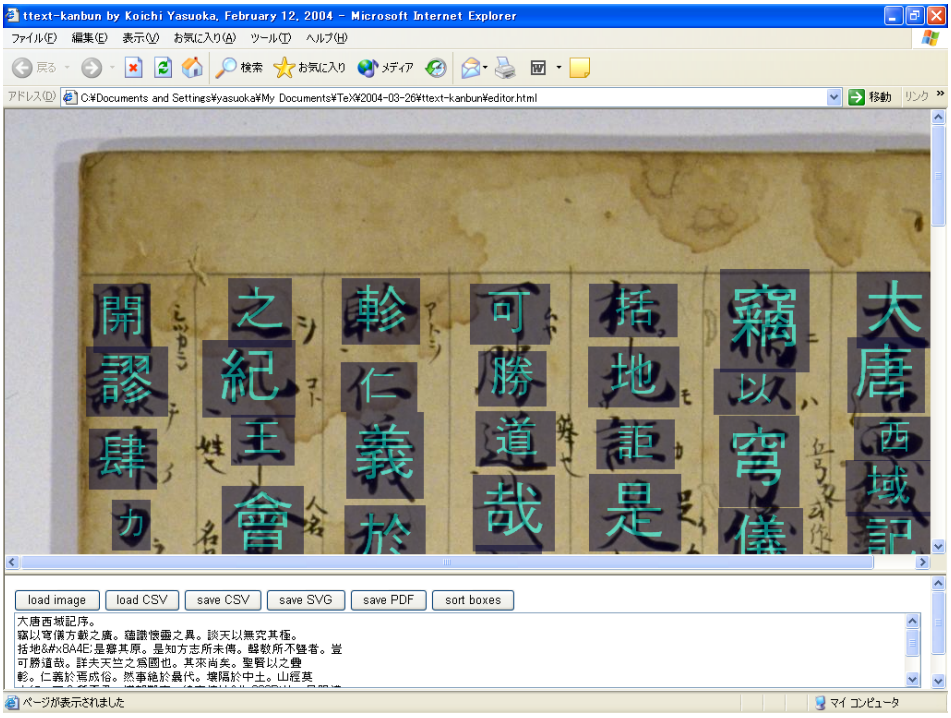

図 3: 文字ボックスの作成

マウスによる編集

画像上でのマウス操作により、文字ボックスの編集作業をおこなう。

- **左ボタンを押しながらマウスを移動**
  文字ボックスが作成される (図 3)。文字ボックス上で左ボタンを押した場合は、文字ボックスの移動あるいはリサイズがおこなわれる。

- **右ボタンを押しながらマウスを移動**
  画像全体をつかんで動かす。

- **左ボタンをクリック**
  直前の文字ボックスと同じ大きさの文字ボックスが作成される。文字ボックス上で左ボタンをクリックした場合は、文字ボックスの順序表示に切り替わる (図 4)。さらに同じ文字ボックスを左クリックすると、その文字ボックスは削除される。あるいは他の文字ボックスを続けて左クリックすると、文字ボックスの順序が変更される。

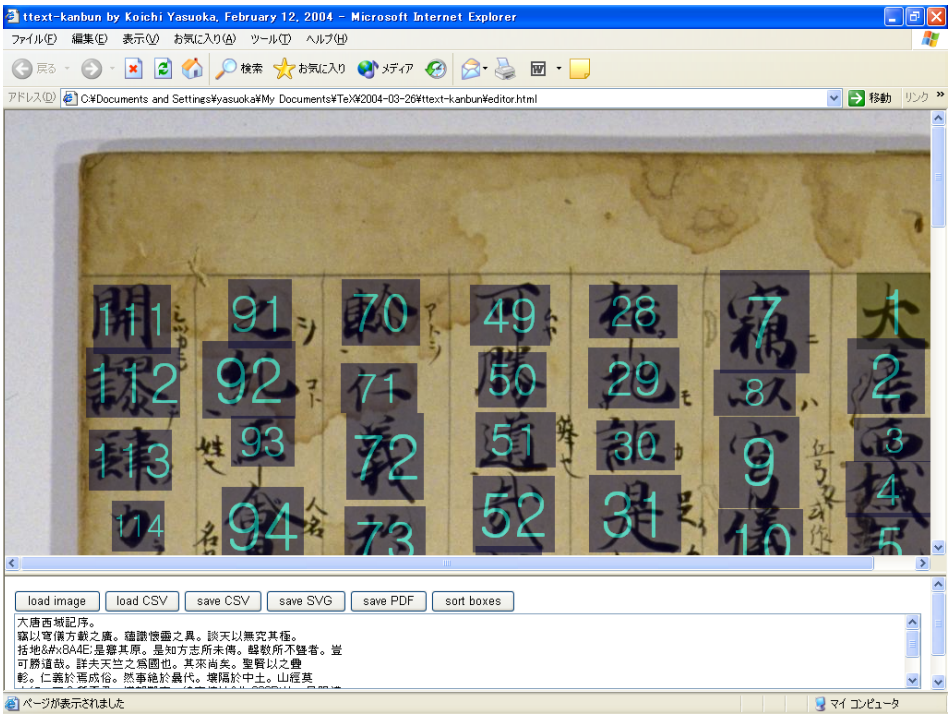

図 4: 文字ボックスの順序表示

文字ボックスの順序変更

sort boxes ボタンをクリックすることで、文字ボックスの順序は、右上から左下に並べ替えられる。ただし、この並べ替えは必ずしも完璧ではないので、各文字ボックスを順に左クリックしていくことで、部分的な並べ替えも可能になっている。

ファイルへの出力

save CSV ボタンをクリックすると、出力するファイル名を確認するウィンドウが表示され、さらに OK を押すと CSV ファイルが出力される。CSV ファイルは通常は、画像ファイルのサフィクスを csv に置き換えたものとなる。出力した CSV ファイルは load CSV ボタンで読み込めるので、作業途中でのデータ保存に用いることができる。

save SVG ボタンをクリックすると、出力するファイル名を確認するウィンドウが表示され、さらに OK を押すと SVG ファイルが出力される。また、jpg 画像に対しては、save PDF によって、PDF ファイルを出力することも可能となっている。

3 ttext-kanbun 出力ファイルの詳細

ttext-kanbun が出力する CSV、SVG、PDF 形式のファイルに関して、その内容が、画像ファイルや文字ボックスとどのように対応しているか、以下に述べる。

3.1 CSV 形式ファイル

各行が文字ボックス 1 個に対応している。1 行は 5 つのフィールドからなり、順に、文字ボックス左上の X 座標、文字ボックス左上の Y 座標、文字ボックスの幅、文字ボックスの高さ、文字の UTF-16 による 10 進数表現となっている。UTF-16 が複数ワードに渡る場合は、各ワードをスペースで繋げて並べた形となる。また、文字実体参照による文字表現は、&、#、など各文字の UTF-16 を、スペースで繋げて並べた形となる。なお、文字ボックスに対応していない文字、すなわち句読点や、文字ボックス数が不足している場合の釈文の残りに対しては、座標などは全て NaN となる。逆に、文字ボックス数が釈文より多い場合は、最後のフィールドが NaN となる。すなわち、釈文として入力されたテキストと文字ボックスは全て CSV 中に出力され、load 時に完全に復元されるようになっている。

```
923,176,85,83,22823
913,247,84,82,21776
920,333,89,48,35199
911,380,93,61,22495
913,436,92,73,35352
910,503,87,84,24207
NaN,NaN,NaN,NaN,12290
NaN,NaN,NaN,NaN,13
NaN,NaN,NaN,NaN,10
775,173,97,112,31434
768,281,89,50,20197
774,332,87,99,31353
773,432,92,78,20736
779,513,77,62,26041
779,571,87,93,36617
785,665,81,55,20043
778,722,86,81,24291
NaN,NaN,NaN,NaN,12290
      ⋮
```

3.2 SVG 形式ファイル

SVG 1.1 に準拠したファイルで、エンコーディングは utf-8 となっているが、実際には ASCII の範囲だけで出力している。画像ファイルは xlink で取り込んでいるので、画像ファイルを同じディレクトリに置いておく必要がある。fill-opacity:0 により、文字は全て透明となっている。文字ボックスは tspan に置き換えているが、割注の文字ボックスは本文の文字ボックスとは別に出力していることから、割注を飛び越した文字列検索が可能となっている。

```
<?xml version="1.0" encoding="utf-8" standalone="no"?>
<!DOCTYPE svg PUBLIC "-//W3C//DTD SVG 1.1//EN"
"http://www.w3.org/TR/SVG/1.1/DTD/svg11.dtd">
<svg width="100%" height="100%" viewBox="0 0 1050 1950">
<image x="0" y="0" width="1050" height="1950" xlink:href="saiiki.jpg"/>
<text style="writing-mode:tb; fill-opacity:0">
<tspan x="965" y="176" style="font-size:83"
>&#22823;</tspan><tspan x="955" y="247" style="font-size:82"
>&#21776;</tspan><tspan x="964" y="333" style="font-size:48"
>&#35199;</tspan><tspan x="957" y="380" style="font-size:61"
>&#22495;</tspan><tspan x="959" y="436" style="font-size:73"
>&#35352;</tspan><tspan x="953" y="503" style="font-size:84"
>&#24207;&#12290;
</tspan><tspan x="823" y="173" style="font-size:112"
>&#31434;</tspan><tspan x="812" y="281" style="font-size:50"
>&#20197;</tspan><tspan x="817" y="332" style="font-size:99"
>&#31353;</tspan><tspan x="819" y="432" style="font-size:78"
>&#20736;</tspan><tspan x="817" y="513" style="font-size:62"
>&#26041;</tspan><tspan x="822" y="571" style="font-size:93"
>&#36617;</tspan><tspan x="825" y="665" style="font-size:55"
>&#20043;</tspan><tspan x="821" y="722" style="font-size:81"
>&#24291;&#12290;
</tspan><tspan x="826" y="811" style="font-size:79"
      ⋮
>&#24432;</tspan></text>
<text style="writing-mode:tb; fill-opacity:0">
</text>
</svg>
```

3.3 PDF 形式ファイル

PDF-1.3 に準拠したファイルで、ASCII のみで書かれている。文字は白地に白で書かれており、その上に画像が重ねられている。各文字は UniJIS-UCS2-V エンコードで表現しているが、Adobe-Japan1[§]の制限により、JIS X 0208 と JIS X 0213 の文字くらいしか表現できない。jpg 画像を ASCII85Encode でファイル内に取り込んでいるので、ファイルサイズは多少大きくなってしまう。

```
%PDF-1.3
1 0 obj << /Type /Catalog /Pages 2 0 R >> endobj
2 0 obj << /Type /Pages /Kids [4 0 R] /Count 1 >> endobj
3 0 obj << /Type /Font /Subtype /CIDFontType0 /BaseFont /Ryumin-Light
/FontDescriptor << /Ascent 723 /CapHeight 709 /Descent -241 /Flags 6
```

[§]日本語 PDF で用いられている文字集合。

```
/FontBBox [-170 -331 1024 903] /FontName /Ryumin-Light /ItalicAngle 0
/StemV 69 >> /CIDSystemInfo << /Registry (Adobe) /Ordering (Japan1)
/Supplement 4 >> /DW 1000 >> endobj
4 0 obj << /Type /Page /MediaBox [0 0 252 468] /Parent 2 0 R /Resources
<< /ProcSet [/PDF /Text /ImageC] /Font << /F1 << /Type /Font /Subtype
/Type0 /BaseFont /Ryumin-Light-UniJIS-UCS2-V /Encoding /UniJIS-UCS2-V
/DescendantFonts [3 0 R] >> >> /XObject << /R1 6 0 R >> >> /Contents 5
0 R >> endobj
5 0 obj << /Length 6847 >> stream
q 0.24 0 0 0.24 0 0 cm 1 g
BT
/F1 83 Tf 102.409638554217 Tz 965.5 1774 Td <5927> Tj
/F1 82 Tf 102.439024390244 Tz -10.5 -71 Td <5510> Tj
/F1 48 Tf 185.416666666667 Tz 9.5 -86 Td <897f> Tj
/F1 61 Tf 152.459016393443 Tz -7 -47 Td <57df> Tj
/F1 73 Tf 126.027397260274 Tz 1.5 -56 Td <8a18> Tj
/F1 84 Tf 103.571428571429 Tz -5.5 -67 Td <5e8f> Tj
<3002> Tj
/F1 112 Tf 86.6071428571429 Tz -130 330 Td <7aca> Tj
/F1 50 Tf 178 Tz -11 -108 Td <4ee5> Tj
/F1 99 Tf 87.8787878787879 Tz 5 -51 Td <7a79> Tj
/F1 78 Tf 117.948717948718 Tz 1.5 -100 Td <5100> Tj
/F1 62 Tf 124.193548387097 Tz -1.5 -81 Td <65b9> Tj
/F1 93 Tf 93.5483870967742 Tz 5 -58 Td <8f09> Tj
/F1 55 Tf 147.272727272727 Tz 3 -94 Td <4e4b> Tj
/F1 81 Tf 106.172839506173 Tz -4.5 -57 Td <5ee3> Tj
<3002> Tj
      ⋮
ET
Q q 252 0 0 468 0 0 cm /R1 Do Q
endstream
endobj
6 0 obj << /Subtype /Image /ColorSpace /DeviceRGB /Width 1050 /Height
1950 /BitsPerComponent 8 /Filter [/ASCII85Decode /DCTDecode] /Length 7
0 R >> stream
s4IA0!"_al8O‘[\!<<*#!!*’"s4[N@!!ic5#6k>;#6tJ?#m^kH’FbHY$Odmc’+Yct)BU"@)B9_>
      ⋮
96&bef‘~>
endstream
endobj
7 0 obj 391170 endobj
xref 0 8
0000000000 65535 f
0000000010 00000 n
0000000060 00000 n
0000000118 00000 n
0000000432 00000 n
0000000730 00000 n
0000007631 00000 n
0000398978 00000 n
trailer << /Size 8 /Root 1 0 R >>
startxref
399001
%%EOF
```

4　おわりに

『透明テキスト付き画像作成ツール (漢文用) `ttext-kanbun`』を開発し、その使用法と出力ファイルについて述べた。なお、SVG と PDF 以外に、DjVu への出力も試みたが、JScript の FileSystemObject がテキストファイルしか出力できない仕様となっているため、バイナリである DjVu への直接の出力はできなかった。筆者は現在は、CSV から DjVuXML 1.1 への変換を awk スクリプトでおこない、`djvuparsexml`[¶]を使って透明テキスト付き DjVu を作成している。今後は、これを自動化するツールを開発したい。

参考文献

[1] Koichi Yasuoka and Tokio Takata: Digital Rubbings — Their Past and Future, 2001 Pacific Neighborhood Consortium Proceedings (January 2001), ECAI Rubbings Work Session.

[2] 守岡知彦: ポスト文字コード時代の文書処理技術に関する展望, 全国文献・情報センター人文社会科学学術セミナーシリーズ, No.12 (2002 年 11 月), pp.59-70.

[3] 守岡知彦: 文字画像のマークアップの試み, 第 14 回「東洋学へのコンピュータ利用」研究セミナー (2003 年 3 月), pp.21-30.

[4] 安岡孝一: 透明テキスト付き画像へのいざない, 第 14 回「東洋学へのコンピュータ利用」研究セミナー (2003 年 3 月), pp.31-42.

[5] 千田大介: InDesign の SVG 書き出し機能, 漢字文献情報処理研究, 第 4 号 (2003 年 10 月), p.131.

[¶]Solaris 用の Document Express Enterprise Edition に含まれている。