Text-Searchable Image and Its Applications

Koichi Yasuoka

Documentation and Information Center for Chinese Studies,
Institute for Research in Humanities, Kyoto University

1 Introduction

Since 1996 proposal of the Council for Science, the university libraries in Japan have progressed "The Digital Library Project". Nowadays the union catalogue database of the university libraries (NACSIS-CAT) is almost completely equipped, and we can easily find any books and magazines in the libraries through the database on the Internet. But we are still far and away from the goal of "The Digital Library Project", which is the digitalization of all the books and the magazines in the libraries. The university libraries have only made displays of images of the rare books without their digital texts, their digital tables of contents, or their digital indices. The digital libraries in Japan now are not "libraries" but something like "museums", since they don't give us the way to "read" the books digitally.

In this paper the author represents the concept of text-searchable images and its applications. The author shows two formats, Portable Document Format and Scalable Vector Graphics, to actualize text-searchable images, and also shows a JavaScript-based program "`ttext-kanbun`" to produce text-searchable images in these formats. The author contributes this paper toward the true progress of the digital "libraries".

2 Text-Searchable Images

In this section we examine two formats, Portable Document Format (PDF) and Scalable Vector Graphics (SVG), to actualize text-searchable images.

2.1 PDF for Text-Searchable Images

The author has studied long time about text-searchable images using PDF [2]. And Adobe adopted some results of the study into PDF-1.4 [3] as "transparent text". Now we have two ways to actualize text-searchable images

using PDF. The one is to put a transparent text upon an image, and the other is to put an image upon a text written in white characters. The former way is only available with the browsers of PDF-1.4 and after, and the latter way PDF-1.2 and after. In this paper we use the latter way for backward compatibility.

PDF can represent both images and texts, but has some limitations on its format. PDF supports only two compression methods for color images, that are JPEG and ZIP. PDF supports several character-sets for CJK texts, Adobe-Japan1-6 [7] (including 14663 漢字 characters), Adobe-GB1-4 [1] (including 27629 汉字 characters), Adobe-CNS1-4 [5] (including 17625 漢字 characters), and Adobe-Korea1-2 [6] (including 4620 漢字 characters) under Japanese, mainland Chinese, Taiwanese, and Korean circumstances, respectively. We need "Japanese Language Pack" to read and search PDFs written in Adobe-Japan1-6 character-set, so as mainland Chinese, Taiwanese, and Korean. This means that these character-sets are incompatible with one another, and that PDFs for text-searchable images actually cannot get across the borderlines. In this paper we use JPEG for color images and Adobe-Japan1-6 character-set for texts to produce text-searchable images with PDF.

2.2 SVG for Text-Searchable Images

Tomohiko Morioka has studied about text-searchable images using SVG [4]. He actualized a text-searchable image to put an image upon a text. But in this paper we put a transparent text upon an image to actualize a text-searchable image using SVG.

SVG can include both images and texts, but the most contemporary viewer "Adobe SVG Viewer 3.0" has some limitations. SVG supports any kind of formats for color images, but the viewer suports only JPEG, PNG, and GIF. SVG supoorts any text-encodings but prefers UTF-8. In this paper we use JPEG for color images and UTF-8 for texts to produce text-searchable images with SVG.

3 Experiment and Result

The author wrote a JavaScript-based program "`ttext-kanbun`" to produce text-searchable images using PDF or SVG. "`ttext-kanbun`" runs on Internet Explorer 6 under Microsoft Windows XP.

We, members of COE21-project at Institute for Research in Humanities, Kyoto University, tried to make text-searchable images of 大唐西域記 (ex-橘寺-collection) with "`ttext-kanbun`" (Figure 1). We prepared 319 JPEG images for 大唐西域記, where each image has 2100×1950 pixels and total size of all images is 196807821 bytes, and its text written in UTF-8 consisting of 104725 characters (3138 different).

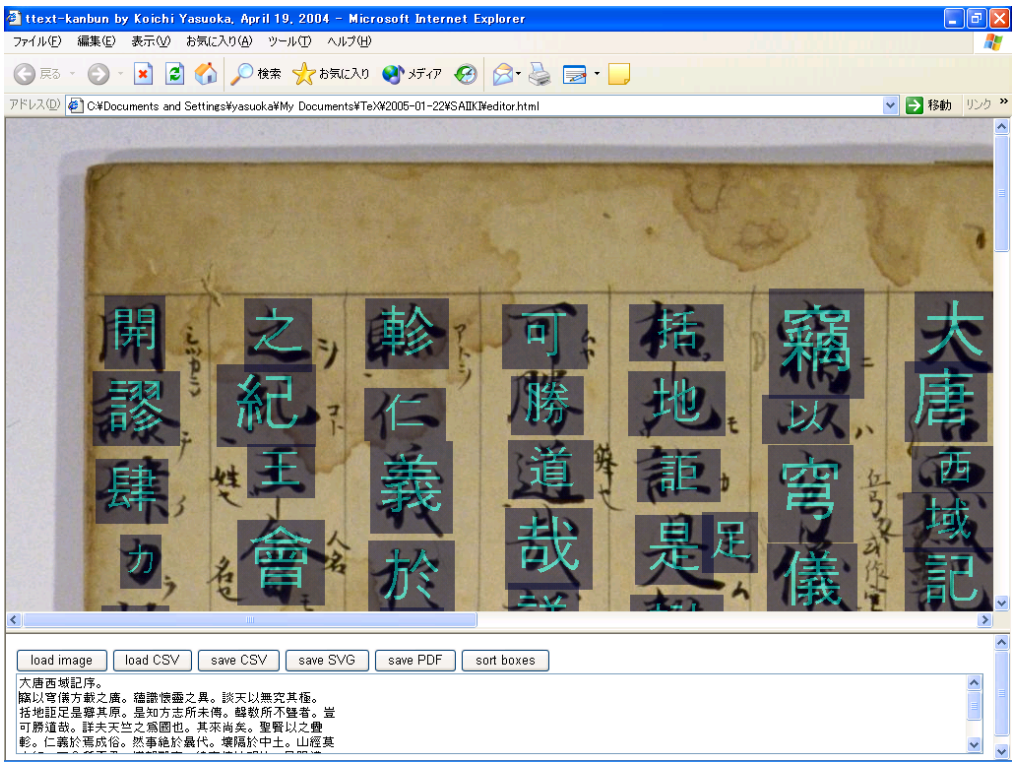

Figure 1: Snapshot of "`ttext-kanbun`"

First we produced text-searchable images using PDF (Figure 2). The total size of 319 PDF files was 202662390 bytes, 2.97% increasing from original JPEG images. We couldn't write 390 characters out of 104725 using PDF since they were not included in Adobe-Japan1-6. The 390 characters consisted of 51 different characters shown in Table 1. Then we combined the 319 PDF files into a multi-page PDF. The file-size of the combined PDF was 202440575 bytes, 2.86% increasing from original JPEG images.

Second we produced text-searchable images using SVG (Figure 3). The

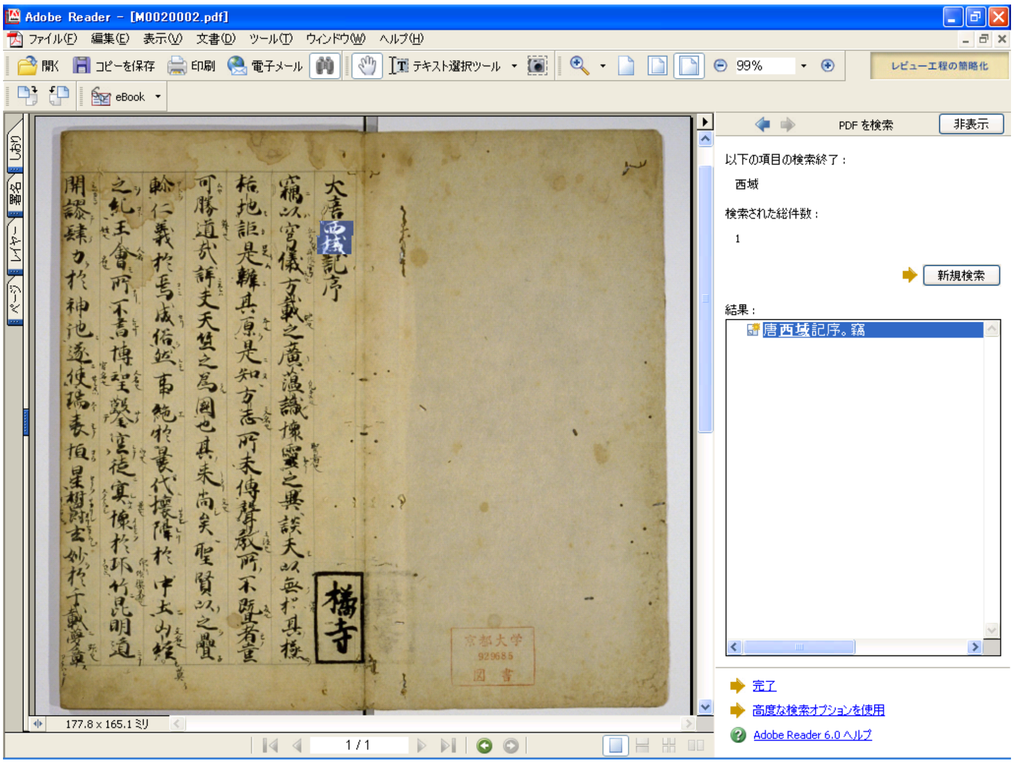

Figure 2: Searching "西域" on "Adobe Reader 6.0"

偘	句	呾	壀	蓯	睿	厝	暏	噭	槶	櫰	殑	礔	礕	笶	菆
菒	藮	薹	賷	奞	隒	霅	頜	鴡	鴐	黳	㤲	擢	柴	縠	㴔
穋	覔	駿	鷰	齌	務	匿	号	㦕	𢪃	斂	𣭘	烦	緝	𢍁	荓
衮	𩚃	餕													

Table 1: Characters not in Adobe-Japan1-6

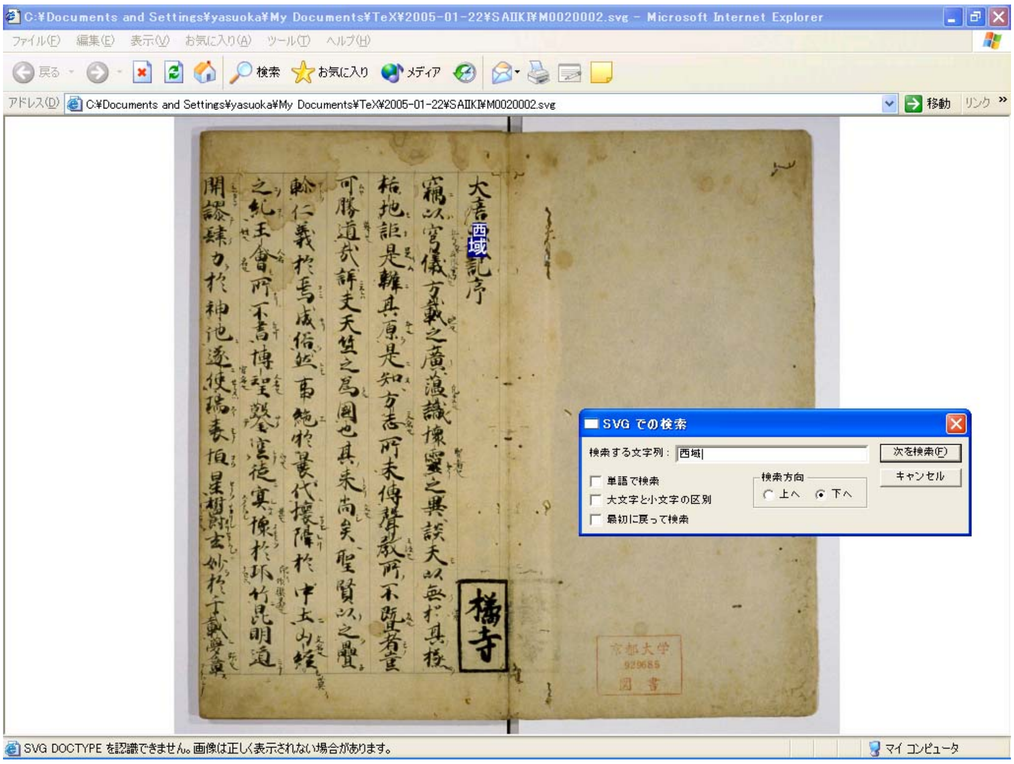

Figure 3: Searching “西域” on “Adobe SVG Viewer 3.0”

㽞	匸	㪾	㦝	㧗	敂	㲕	㶴	絹	丼	丼	㼭	飠	餕

Table 2: Invisible characters on “Adobe SVG Viewer 3.0”

total size of SVG files and JPEG images was 203752662 bytes, 3.53% increasing from JPEG images only. All characters could be represented in SVG files, but 14 characters shown in Table 2 couldn't be displayed on "Adobe SVG Viewer 3.0", since the Viewer didn't support Unicode Plane-2 fonts.

4 Conclusion

In this paper the author has represented the concept of text-searchable images and its actualization using PDF or SVG. The author wrote a JavaScript-based program "`ttext-kanbun`" to produce such text-searchable images. As a result we have found that only 3% to 4% file-size increase is needed to add texts on JPEG images. The author now distributes "`ttext-kanbun`" at `http://coe21.zinbun.kyoto-u.ac.jp/~yasuoka/ftp/program/` and is pleased to help anyone to produce such text-searchable images.

References

[1] Adobe-GB1-4 Character Collection for CID-Keyed Fonts, Technical Note #5079, Adobe Systems (November 2000).

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

[3] Adobe Systems Incorporated: PDF Reference third edition — Adobe Portable Document Format Version 1.4, Addison-Wesley (December 2001).

[4] 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 (November 2002), pp.59-70.

[5] Adobe-CNS1-4 Character Collection for CID-Keyed Fonts, Technical Note #5080, Adobe Systems (May 2003).

[6] Adobe-Korea1-2 Character Collection for CID-Keyed Fonts, Technical Note #5093, Adobe Systems (May 2003).

[7] Adobe-Japan1-6 Character Collection for CID-Keyed Fonts, Technical Note #5078, Adobe Systems (June 2004).