

GUIによるグリフ画像コーパス編集用ソフトウェアの開発

王 一凡¹ 永崎 研宣² 下田 正弘³

概要: 複層的な伝承経路に由来する膨大な活字種を内包した『大正新脩大藏經』所収「一切経音義」「統一一切経音義」本文の分析にあたり、版面画像から各グリフ画像を自動的に切り出して全文コーパスに対応づけるシステムに加え、画像を手動で適切に分類・修正するためのクロスプラットフォームな GUI 環境を開発した。これによりコーパスの継続的な保守が可能になるばかりでなく、一連の手法は他の活字化仏典をはじめ戦前期和文活字本のコーパス構築に広く応用できると考えられる。

Developing GUI Editorial Softwares for Glyph Image Corpus

WANG YIFAN¹ NAGASAKI KIYONORI² SHIMODA MASAHIRO³

Abstract: In the study of *Yiqiejing Yinyi* and *Xu Yiqiejing Yinyi* in *Taisho Tripitaka*, which contain a vast repertoire of variant characters inherited from their antecedent documents, we have created a set of cross-platform programs to automatically extract each glyph from full-page images to associate with text corpus entries, and edit corpus glyph data in GUI. The system greatly improves efficiency in development and maintenance of the corpus and, we believe, is further applicable to other studies involving printed Japanese historical documents.

1. はじめに

1.1 大正新脩大藏經および『一切経音義』について

大正新脩大藏經（以下、大正蔵）とは、大正～昭和期に日本で編纂された漢文による經典である。一般に「大藏經」とは漢訳仏典を集成したものであり、中国では唐代以降歴代王朝により定められた形式で刊行された。大正蔵の特徴は、阿含部を冒頭に配するなど伝統的な漢文大藏經の枠組みを破った斬新な編集方針や、日本各地の古写本から敦煌文献といった当時の最新資料に至るまで網羅的に収集したことなどにあり、現在まで漢訳仏典の基本資料として用いられている。

『一切経音義』とは經典の字句を解説する語彙集の一つで、大藏經内に収録されている。『一切経音義』を冠する文献のうち最も良く知られているのは玄奘（唐、7世紀頃）撰の二十五卷本（いわゆる『玄奘音義』）であるが、ほかに

これを内包して増補した [1] 慧琳（唐、730 頃-820 頃）撰の百卷本（以下、慧琳音義）が存在する。また、これに加え希麟（遼、生没年不詳）撰『統一一切経音義』十卷（以下、希麟音義）も存在する。慧琳・希麟の両音義は中国では散逸して伝わらず、高麗大藏經（以下、高麗蔵）に保存されるのみであったが、江戸時代に日本で翻刻された（白蓮社本）のを皮切りに近代以降中国・日本でいくつかの刊本が作られた。

以前の研究報告 [2] でも述べた通り、大正蔵『一切経音義』は玄奘音義を収録しておらず、高麗蔵の慧琳音義および希麟音義を底本としているが、この版本は後世の校訂をほとんど経ておらず、唐代の多様な異体字を忠実に保存していると考えられている [3]。現存する慧琳・希麟音義の諸本は確認しうる限り大正蔵を含めすべて高麗蔵に遡るものであり、かつすべて刻本か活字本であるが、互いに校合を繰り返した結果、異体字の使用状況が複雑な様相を呈しているのが特徴である。

¹ 東京大学大学院人文社会系研究科・修士課程
Graduate School of Humanities and Sociology, University of Tokyo

² 人文情報学研究所
International Institute for Digital Humanities

³ 東京大学大学院人文社会系研究科
Graduate School of Humanities and Sociology, University of Tokyo

1.2 研究の目標

SAT テキストデータベースプロジェクト^{*1}（代表：下田正弘）では大正新脩大蔵経（大正蔵）のデジタル化に取り組んでおり、その一環として大正蔵出現外字の UCS 符号化を目指している [4]。『一切経音義』『統一経音義』部分は大正蔵においては正統蔵 85 巻中第 54 巻の半分を占めるにすぎないが、SAT から Unicode へ新規申請見込みの漢字のうち、約半数にあたる 3,000 字がこの範囲に集中しているため、目下の課題となっている。

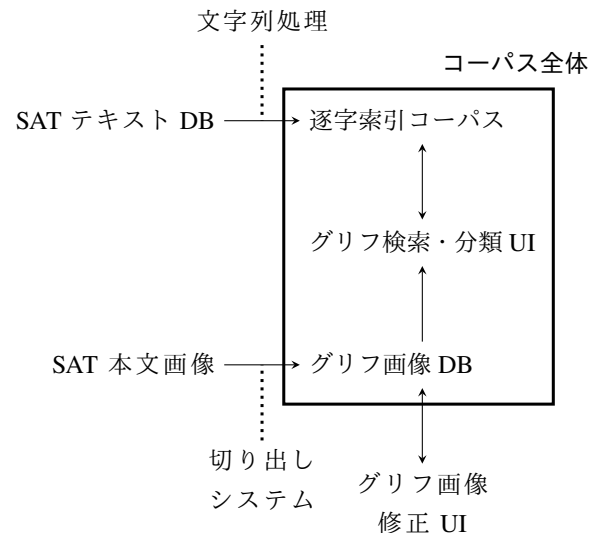
Unicode に申請する外字を選定するにあたっては、文字の包摂に関わる意味的な同一性が重視される [5]。大正蔵は活字本であり、一見活字の同一性を判断することは困難が少ないように思われがちだが、『一切経音義』部分の編集は他の巻とベースに合わせるために突貫で行われたという記述が残されており [5]、実際にその混乱を反映するかのよう統一性のない用字が至る所にみられ、中には違いの根拠を他の諸本に求めづらいうこともしばしばである。『一切経音義』部分の総文字数は延べ 100 万字超、2014 年時点の SAT テキストデータによれば異なり活字数は 3 万弱を数えるため、これらを網羅的に把握して理解することが難しかった。

また、大正蔵の編纂作業は関東大震災後の逼迫した状況で行われ [6]、ある程度残されたと思われる紙型その他の編集資料も戦災で失われたようで、戦後の復刻にあたって作成されたという正誤表 [7] も管見の限りでは確認できず、それらに手がかりを求めることもできない。

この状況を踏まえて、大正蔵『一切経音義』の用字体系を理解するため、現在 SAT が保有するテキストデータと原文画像に基づいたグリフ画像データベースを作成している。また、このコーパスの構築と保守を効率的に行うために、自動処理プログラムおよび各種 GUI による編集ソフトウェアを開発しており、本発表では主にこれらについて紹介する。

2. 仕様と設計

『一切経音義』『統一経音義』全文を網羅したコーパスを作成するにあたって、既存のデータからコーパスに適した形式に変換するための一連のシステムと、コーパスに変更を加えるためのシステムを設計した。全体の構成は以下の通りである。



既存のデータである SAT のテキストデータベースからは文字列処理により索引コーパスの雛形を生成し、本文画像からは文字を単離してグリフ画像のコレクションを生成する。自動生成されたものに対して必要な変更を加え、また逐次更新するためにグリフ画像の修正用画面およびグリフの分類メタデータ編集画面を用いる。

最終的な成果物となるのは太枠に囲まれた部分である。現段階での目標としては大正蔵『一切経音義』のみのコーパスだが、将来的には高麗蔵や白蓮社本との対照コーパスへ拡張することも視野に入れている。また、コーパスの入力が完了した際には一般公開し、他のデータベースとの連携などを通じて多目的な利用ができるようにしたいと考えている。

本システムの主要要素については 3 節で詳しく述べる。なおコーパスの設計に関しては [8] を、画像切り出しの詳細については以前の研究報告 [2] をあわせて参照されたい。

3. 構成

全体像を明らかにするため、関連する成果物となるコーパス自体についてもあわせて紹介する。

3.1 逐字索引コーパス

コーパスは SAT データベースから取得した正統『一切経音義』部分のテキストを基に、出現する各文字を一文字ずつ順に配列した形式となっている。本文部分はすべてひと続きのテキストとみなすことができ、文字の前後関係に曖昧さはない。脚注は本文とは不連続でページごとに別フローとなるが、脚注番号で明確な形に配列されており、同様に曖昧さはない。

各文字ごとには SAT データベースから引き継いだ出現位置などの情報に加え、字形情報を正確に記述するために表 1 に示した字形の分類などに関するメタデータを付与し

^{*1} <http://21dzk.1.u-tokyo.ac.jp/SAT/>

	docid	vol	page	col	line	charidx	ideoidx	side	sideidx	repres	gou	date	user	chapter
1691	2128	54	312	a	4	3	3	0		叙	4	2007...	SAT	0
1692	2128	54	312	a	4	4	4	0		之	4	2007...	SAT	0
1693	2128	54	312	a	4	5	5	0		云	4	2007...	SAT	0
1694	2128	54	312	a	4	6	6	0		爾	4	2007...	SAT	0
1695	2128	54	312	a	8	1	1	0		一	2	2007...	SAT	1
1696	2128	54	312	a	8	2	2	0		切	2	2007...	SAT	1
1697	2128	54	312	a	8	3	3	0		經	2	2007...	SAT	1
1698	2128	54	312	a	8	4	4	0		音	2	2007...	SAT	1
1699	2128	54	312	a	8	5	5	0		義	2	2007...	SAT	1
1700	2128	54	312	a	8	6	6	0		卷	2	2007...	SAT	1
1701	2128	54	312	a	8	7	7	0		第	2	2007...	SAT	1
1702	2128	54	312	a	8	8	8	0		一	2	2007...	SAT	1
1703	2128	54	312	a	9	1	1	0		爾	4	2007...	SAT	1
1704	2128	54	312	a	9	2	2	0		經	4	2007...	SAT	1
1705	2128	54	312	a	9	3	3	0		沙	4	2007...	SAT	1
1706	2128	54	312	a	9	4	4	0		門	4	2007...	SAT	1

図1 SQLite 上のコーパスのプレビュー



図2 グリフ画像 DB の一部のプレビュー (縮尺は等しくない)

ている。メタデータ部分は現在入力中であり、正確なデータ入力のために 3.2 節に述べるグリフ画像データベースを参照している。

コーパスは SQLite3 形式のデータベースファイルに格納されている (図 1)。

また、作業の進捗に伴う修正や SAT 側での変更を反映するために、SAT データベースから更新分を取得して更新する Ruby スクリプトを作成した。

3.2 グリフ画像データベース

SAT にて公開されている大正蔵原書のスキャン画像から文字を一字ずつ切り出したグリフ画像のコレクションである。各文字画像は画像形式 (JPG) で原書の段 (および脚注欄) ごとに区分されたディレクトリ内に格納される (2016 年 4 月現在)。

画像はその文字が 3.1 節の索引コーパス内で持つ位置情報と対応するよう名称を付けている (図 2) が、SAT データベースのテキストではレイアウト上空行に相当する番号を飛ばして行番号を付与しているのに対し、本研究的手法では実際に存在する行数しか検知できないため、章題などがある少数の段では対応づけを手動で修正しなければならず、目下作業中である。

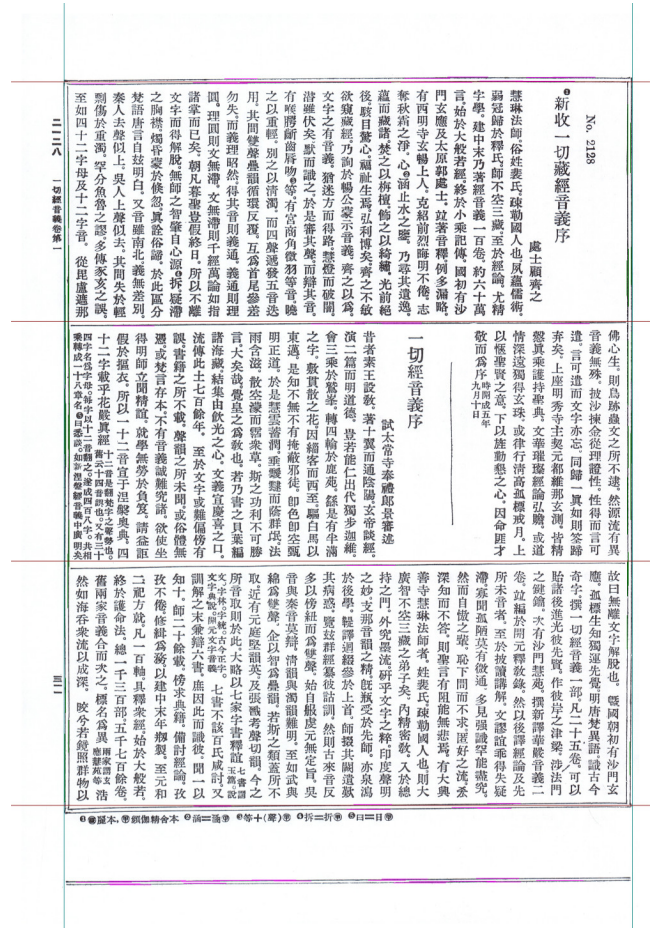


図3 切り出しシステムによる段組検出結果の模式図 ([2] より)

3.3 グリフ画像切り出しシステム

詳細は [2] に述べたが、SAT の本文画像からグリフ画像を効率良く切り出すために、自動処理による切り出しシステムを開発した。

これは画像処理ライブラリとしてオープンソースの OpenCV 2.4.10 を利用し、ruby-opencv^{*3} パッケージを介して Ruby 2.0 以降の処理系で動作するプログラムである。プログラムは 1 ページ全体の画像を与えられると、段組および余白を検出して分割 (図 3) し、次にそれぞれの段から行 (ただし割注は 2 ライン分 1 行) を検出して分割し、最後に各行の範囲内に存在する文字を検出する (図 4)。活字本という特性上、文字のサイズは限られたパターンしか存在せず、字面の重なりも原則として存在しないという前提で、四号・五号・六号^{*4} および補助記号などに合わせた定型サイズで検出させることができたので、比較的簡易なプログラムにすることができた。

処理結果は段ごとの画像と、各段の行区切り・文字の検出位置を示した YAML ファイルとして出力される。自動処理の性質上精度の問題が発生することや、再利用の便を

^{*3} <https://github.com/ruby-opencv/ruby-opencv>

^{*4} 四号・五号・六号活字とはそれぞれ現在の 13.75pt, 10.5pt, 8pt に相当し [9]、ソース画像内ではそれぞれ約 60px, 80px, 106px 四方に対応したが、滲み字対策の大きめの枠も用意した。

表 1 追加メタデータ

フィールド名	内容	備考
印影違い所属	字画の構成が同じだが活字の字面が異なるグループ名	既存字か外字番号で代表
印影違い符号	上のグループ内の区分	アルファベットで付番
構造違い所属	一部部品が置換した異体字グループ名（印影違いの上位）	既存字か外字番号で代表
構造違い符号	上のグループ内の区分（印影違い所属に対応）	数字で付番
和装本印影違い所属	SAT 底本と和装本とで印刷の状態が異なる、また修正の跡があるものに必要 記述は上と同じ	可能な範囲で入力
和装本印影違い符号		
和装本構造違い所属		
和装本構造違い符号		
構造データ	（参考情報）構造違い代表字の構造を表す	主に CHISE ^{*2} より取得
関連字	（参考情報）関連する文字（検索の便のために使用）	可能な範囲で入力
グリフ画像パス	文字画像へのリンク	詳細は 3.2 節を参照

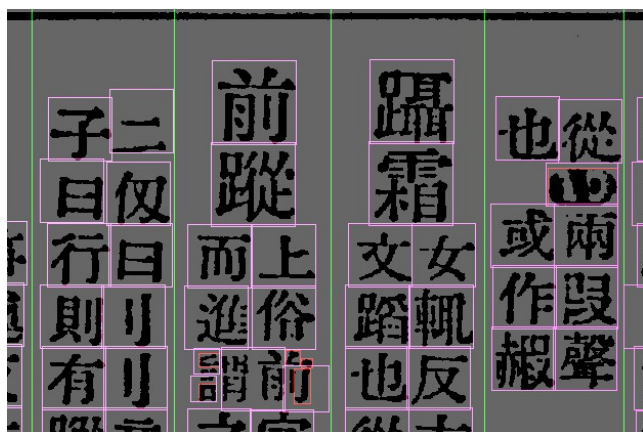


図 4 切り出しシステムによる文字検出結果の模式図（[2] より）

図するため、直接グリフ画像として出力することは避けた。また、以上のプログラムでは矩形を検出するのみで文字の順序情報は出力しないため、YAML データを解析し文字の配列関係を認識して正しい順序でグリフ画像データベースを生成する Ruby スクリプトも作成した。

自動切り出しの精度は、執筆時現在までに補正した結果を正しいものと仮定すると、脚注を除いた総検出行数 48,163 行につき 1,195 行（約 2.5%）の誤り、総検出文字数 1,128,548 字につき 68,403 字（約 6.1%）の誤り^{*5}が認められた。誤りの原因には版面の汚れ、滲み、傾きなどでアルゴリズム上対応できなかったものも含まれている。

3.4 グリフ画像修正用インターフェイス

3.3 節のシステムで出力した出現位置データを手動で直感的に確認・修正するために、GUI による修正用プログラムを開発した。UI ライブラリとして Qt 4.8.6 を利用し、qtbindings^{*6} パッケージを介して Ruby 2.0 以降の処理系で動作する。当初はウェブアプリケーションとしての開発

^{*5} この数には、現在までに変更された箇所のうち、手動で修正されたものが含まれる。すなわち検出の相対位置の誤り、サイズの誤り、区切り間違い、誤検出、未検出、またはこれらをつつ以上含むものなどである。なお段組画像の修正に伴う座標のずれは除いた。

^{*6} <https://github.com/ryanmelt/qtbindings>

も検討に入れていたが、動作速度の問題などがあり、ネイティブアプリケーションとして開発した。

このプログラムは段ごとに分割された画像と YAML ファイルを読み込み、編集画面上に画像とグリフに相当する矩形の位置を重ねて表示する。作業者は拡大・縮小や移動を行い、目視でこれを確認する。誤りのある箇所を発見した時は、矩形や区切り線を選択して移動・追加・削除を行うことができる。編集した結果は保存することで YAML ファイルに上書きされ、前後の段に切り替えたり任意の箇所から編集を開始したりすることが可能である。編集されたファイルを元にグリフ画像データベース（の一部）を再生成することで、データベースが更新される。

筆者らは実際にこれを用いて修正作業を行い、全ページの確認および修正を一通り行った。執筆時現在は、このような流れ作業での処理を念頭に置き、図 5 に示すように、モードを選択するだけの最低限のインターフェイスのみを表示してすべてマウス操作によって編集をするスタイルになっているが、Qt の機能により操作とインターフェイス部品のリンクを簡単な修正で組み替えることができる。今後継続的に保守していく際にも作業者や作業内容のニーズを汲んだカスタマイズが可能である。

3.5 グリフ検索・分類用インターフェイス

コーパスにおける各文字エントリに対応するグリフ画像を検索し、適切なメタデータを付与したり分類したりするために、閲覧・修正用のウェブアプリケーションを開発した。ウェブを採用したのはネイティブアプリケーションと比較して GUI の構築や更新が簡単に行え、また外部サービスとのリンクに便利なためである。

このプログラムでは 3.1 節のデータベースファイルおよび 3.2 節の画像ファイルを読み込み、検索条件にしたがってヒットした文字（SAT によるオリジナルデータまたは分類済みのカテゴリ）のグリフ画像を一覧表示する HTML ページを生成する。編集者は表示されたページ上で目視により文字の分類を確認し、必要があれば画像をドラッグし

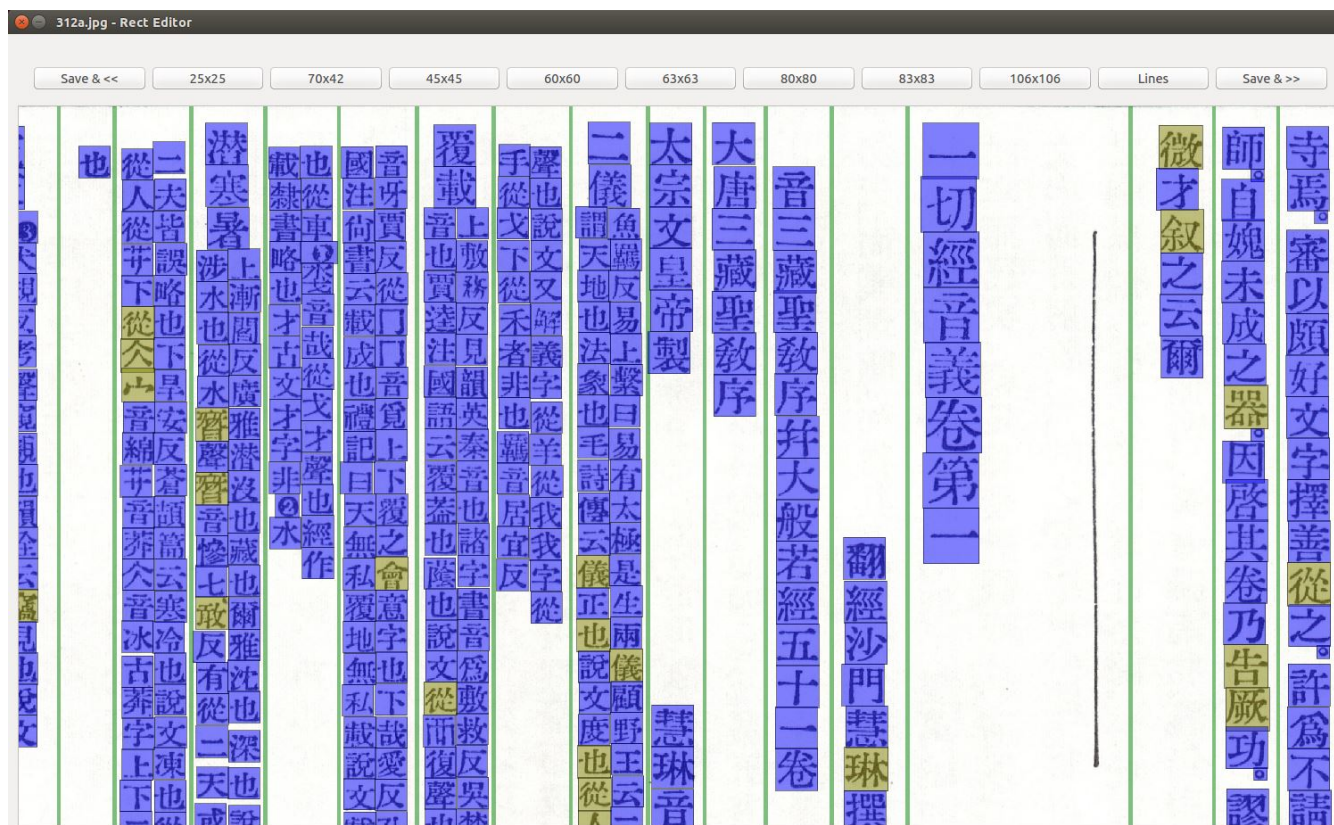


図5 グリフ画像修正用画面

て分類を変更したり新規の分類を作成することができる。変更した情報を保存することで、コーパス本体が更新される。一覧画面に表示されているのは各グリフ単体が切り出された画像とその出現位置番号のみだが、前後の文脈を確認したい場合にはクリックにより、3.3節で述べた段組画像とYAMLファイルから元画像内での出現位置を再現した画面を開くことができる(図6)。この確認画面は実質的に3.4節と同等の情報が表示されているが、パフォーマンスの問題からグリフ画像自体への編集は有効となっていない。その代わり、近傍の文字を選択してその文字が含まれる検索結果にジャンプすることができる。

バックエンドはRubyによるSinatra^{*7}フレームワークを用いたCGIアプリケーションとして構成されており、表示とクライアント側での操作補助にjQuery UI^{*8}ライブラリ(ver. 1.11.4)およびjQuery^{*9}ライブラリ(ver. 1.12.1)を利用している。

コーパスを外部公開する準備が整った際には、この画面を一般の訪問者が利用可能な検索インターフェイスに転用することを視野に入れ、外部サイトとの連携機能も適宜追加していく予定である。

4. 成果と課題

4.1 研究目標に対する成果

本発表で示したシステムの開発により、コーパス構築の高速化と省力化を図ることができ、ひいては最終的な目標である大正蔵『一切経音義』の用字体系解明の基礎となる膨大なデータ解析に多大な役割を果たした。特に出現グリフの拾い出しと比較は手動で行うと効率および網羅性を確保するのが非常に困難であり、出現頻度など統計的な分析に耐えうる多面的なデータをその都度収集するための労力は計り知れない。グリフ画像コーパスの自動生成および修正システムの存在により、作業者の負担を大幅に減少させることが可能になり、少ない作業人数・時間でコーパスの開発とメンテナンスを行うことができるようになった。これにより、本来の研究目標へより多くのリソースを振り分けられるようになり、従来よりも短期間で成果を得ることが可能になる。

また、本研究の題材である『一切経音義』は現存する最古の資料から刊本であるという特異な性格を持つため、同様の手法で頻伽精舎本や縮刷蔵、高麗蔵など関連諸本のグリフ画像を取得することでコーパスの拡張が可能であると考えられる。そのため、今後は諸本の対照コーパスの作成を通じて、原典研究の面でも寄与するところが大いと思われる。

^{*7} <https://github.com/sinatra/sinatra/>

^{*8} <http://jqueryui.com/>

^{*9} <http://jquery.com/>

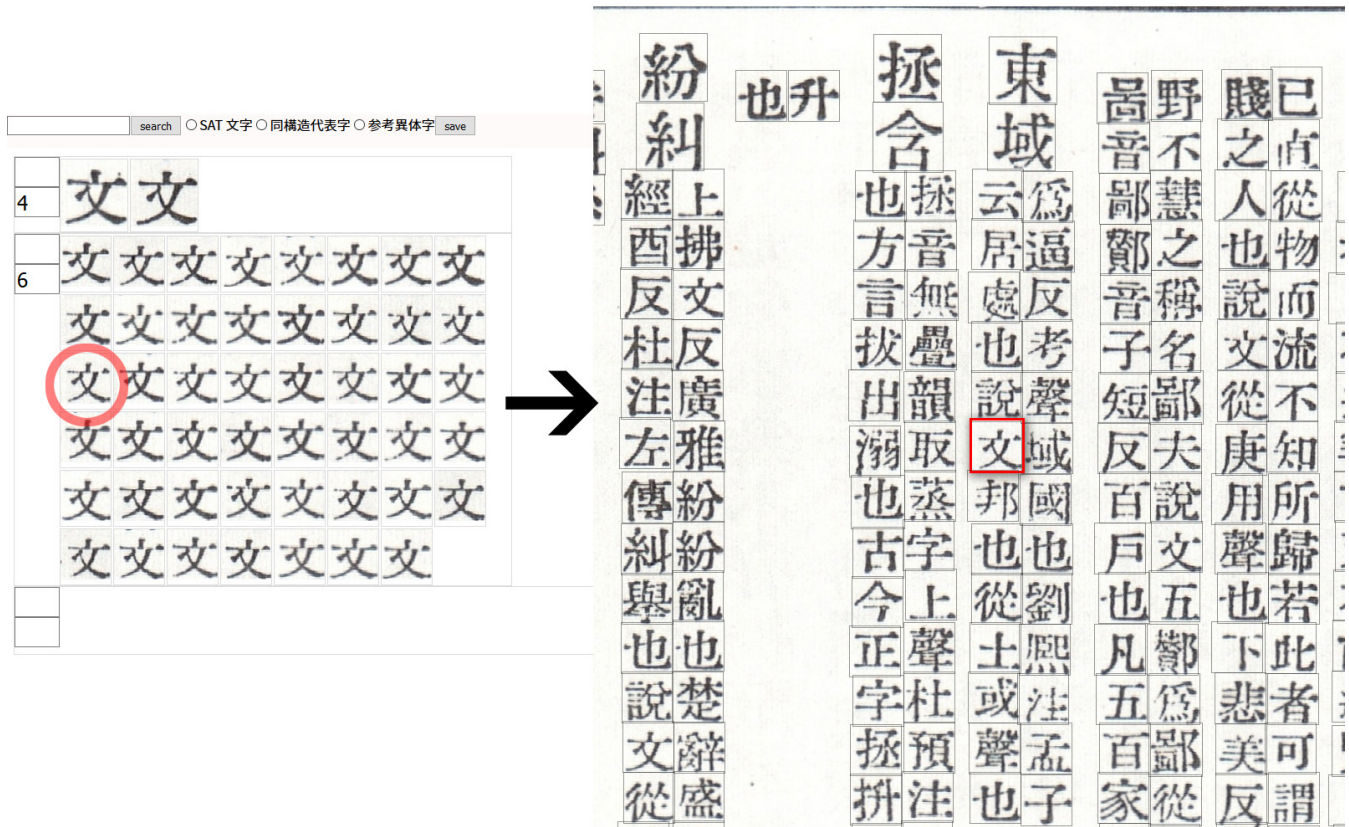


図6 グリフ画像検索用画面（2016年4月現在）

4.2 研究分野における意義

本研究におけるコーパスとその生成・編集システムは、類似する状況を有する大蔵經研究にとどまらず、戦前期の縦書き活字本、ないし割注を含む縦書きの活字本または版式の定まった版本一般のグリフコーパス作成に応用することができると思われる。この手法は、例えば和文活字で出版された文献の異なる版の比較など、出現字形の厳密なもしくは網羅的な比較を要する研究に対する一つのモデルケースを提示できるのではないかと考える。

技術的な面で特筆すべきは、本発表で紹介した各種ソフトウェアおよびそれらに使用したツールはすべてフリーなライセンスのもとで利用できるオープンソースソフトウェアであり、かつ Windows, Mac OS X, Linux いずれの環境でも動作するクロスプラットフォームなものであるということである（本研究自体は Ubuntu および Windows 環境で行われている）。その意味で、本発表での手法は研究の遂行に伴う技術的な制約を最小限に抑え、応用や移植に柔軟に対応できる汎用性のあるモデルを提供していると言える。

4.3 課題と展望

本発表で述べた内容は枠組みとしては相当程度に整備されていると思われるが、研究用ツールとしてより有用であるためには、今後も継続的にブラッシュアップしていく必要がある。特にインターフェイスとしての動作性能・使い

やすさに関して最適化の余地がまだまだ残っていると言える。最終的にコーパスを公開する際に必要となる変更も含め、設計を練っていきたいと考えている。

3.5 節で述べたグリフの分類については、全データに適用となった場合依然として膨大な作業量が必要である（当文献内には約3万の違うグリフが存在していると目されている）。そのため、手書き文書の解析に優れた文献研究支援ソフトウェアである SMART-GS^{*10}[10]を画像の照合に応用することで、省力化を図ることを検討している。

また、本発表の段階では SAT のウェブ閲覧用画像と同等の画質のものを素材として使用しているが、データを利用してより鮮明な TIFF 形式の画像から切り出すことも検討している。

参考文献

- [1] 高田時雄：藏經音義の敦煌吐魯番本と高麗藏，敦煌寫本研究年報，Vol. 4, pp. 1-13 (2010).
- [2] 王 一凡：OpenCV を利用した活字画像の切り出し—大正新脩大蔵經『一切經音義』の活字字形研究に向けて—，研究報告人文科学とコンピュータ (CH)，Vol. 2015-CH-106, No. 8, pp. 1-4 (2015).
- [3] 陳 五雲，徐 時儀，梁 曉虹：佛經音義與漢字研究，鳳凰出版社，1 edition (2010).
- [4] 王 一凡，永崎研宣，下田正弘：UCS 符号化という観点からの『慧琳撰一切經音義』の検討，漢デジ 2014: デジタル翻刻の未来 (予稿集) (京都大学人文科学研究所附属東

^{*10} <https://osdn.jp/projects/smart-gs/>

- アジア人文情報学研究センター, 編), pp. 61–66 (2014).
- [5] ISO/IEC JTC1/SC2: Annex S: Procedure for the unification and arrangement of CJK Ideographs, *ISO standards: Information Technology – ISO/IEC 10646:2003, Universal Multiple-Octet Coded Character Set (UCS)*, ISO, Geneva, pp. 2250–2260 (2012).
 - [6] 大藏小僧: 藏經逸聞集, ピタカ, Vol. 3, No. 1, pp. 35–52 (1935).
 - [7] 大正新脩大藏經刊行会: 編集室, 大正新脩大藏經会員通信, No. 9, p. 4 (1961).
 - [8] 王 一凡: 大正新脩大藏經所収『一切経音義』の文字同定と索引構築の試み, 東洋学へのコンピュータ利用研究セミナー, No. 第 26 回 (2015).
 - [9] 小宮山博史: 日本語活字ものがたり: 草創期の人と書体, 誠文堂新光社 (2009).
 - [10] 橋本雄太: 集合知で読む歴史史料-SMART-GS が実現するグループリーディング, 人文情報学月報, No. 37 (オンライン), 入手先 (http://www.dhii.jp/DHM/DHM37_smartgs) (2014).