Vol.2015-CH-106 No.8
2015/5/16

OpenCV を利用した活字画像の切り出し —大正新脩大藏經『一切経音義』の活字字形研究に向けて—

王一凡[†1]

大正新脩大藏経所収『一切経音義』は、活字本でありながら多様な差異を有する膨大な異体字群を内包している。当資料を適切にデジタル化・UCS 符号化するためには、活字の異同を検討したうえで用字に関する体系的な理解を得る必要があるが、総字数 100 万字超、異なり活字約 3 万種と推定される本文を直接点検しながら、一貫性のある分析を行うことは困難である。したがって、活字の集計を省力化する手段が求められる。本報告では、オープンソースライブラリ OpenCV による自動処理を適用することでこれを実現する試みを紹介し、もって特定分野への汎用ライブラリの応用の可能性を提示する。

Isolating Typeset Characters Using OpenCV: Applied to *Yiqiejing Yinyi* in *Taisho Tripitaka*

WANG YIFAN[†1]

Yiqiejing Yinyi in *Taisho Tripitaka* is a document embraces a vast range of character variants contrary to its letterpress nature. Accurate digitalization and UCS proposal of its characters require correct understanding of its internal writing system through empirical and consistent research, which would be prohibitively onerous against a book which contains over one million characters and est. 30,000 of different character types. This report seeks a solution by automatic image processing using an open-source library OpenCV, as a case study on application of general-purpose library to a specific humanities field and objective.

1. はじめに

筆者は 2013 年末より、大正新脩大藏経（大正藏）のデジタル化を進める SAT テキストデータベースプロジェクト*[a]（代表：下田正弘）で『一切経音義』出現外字の UCS 符号化を目指した外字の調査に参加している。『一切経音義』とは大藏経の字句を解説するための語彙集である。大正藏においては正続藏 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*[b]の漢字のうち、約半数にあたる 3000 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。

大正藏における『一切経音義』の特異性は、その伝承経路によるところが大きい。『一切経音義』または類似の書名で呼ばれる文献は複数存在するが、大正藏はそのうち慧琳（唐、730 頃-820 頃）撰の百巻本（『慧琳音義』）、および希麟（遼、生没年不詳）撰『続一切経音義』十巻のみを採録している。より知られているのは玄応（唐、7 世紀頃）撰の二十五巻本（いわゆる『玄応音義』）であるが、慧琳音義はこれと同等の内容をも内包している[1]。しかし歴代大藏経に収録された玄応音義と異なり、慧琳音義は早くに中国から散佚し、わずかに高麗大藏経（高麗藏）に残るのみであり、現在確認できる諸本はすべて高麗藏に遡るとされている。このため慧琳音義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することになった[2]。

大正藏は、そのやや前に上海で出版された頻伽精舎本大藏経を主要な底本としている[3]。頻伽精舎本は概ね大日本校訂縮刷大藏経（縮刷藏）を下敷きにしているが、『一切経音義』のみは諸般の事情により江戸時代の刊本である白蓮社本（監修者の名を取って忍澂本とも呼ばれる）を底本にしている[4]。縮刷藏所収の慧琳音義は、用字を校訂して刊行されたが、大正藏は結局これを継承しなかったために、高麗藏に遡る膨大な異体字を引き継ぐこととなった。

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。大正藏は活字本であり、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ページ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に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、昨年時点の 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。

この状況を踏まえて、大正藏『一切経音義』の用字体系を理解するため、筆者は昨年より SAT テキストデータをリレーショナルデータベースに格納した単字索引 DB を作成している[6]。これにより外字の出現傾向などをある程度統計的に観察しやすくなったが、現段階ではあくまでデータを

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

a) http://21dzk.l.u-tokyo.ac.jp/SAT/

b) なお『一切経音義』以外の一部の字を含む CJK Extension F は現在 ISO において議論が行われている。

テキストに依存しているため字形に関する情報がない。研究の基礎となる実際の字形に基づいた分類データを作成するには、字面の異同を確認するために元画像と照合しなければならない。手動で画像を確認するのはあまりに労力がかかり非効率なため、画像データをすぐに目視できる状態で提示できるインターフェイスが必要であると考えていた。そこで、自動処理によって画像から文字単体を切り出す方法を検討する。

2. 手法

2.1 目的

今回の研究においては、SAT プロジェクトページから取得した大正蔵『一切経音義』のページ画像の本文部分から、自動処理により単字ごとに分割した画像を行・位置に関する情報を伴って切り出すことを目標とした。対象が活字資料である性質上、文字の外形が定まっていることが期待され、また文字の同定それ自体は研究上議論される問題であり、今回の自動処理の対象ではない。したがって、通常の OCR とは異なり、外形的な特徴から正しく文字の輪郭を切り出すことに重点を置いた。

処理の正確さは、前述の SAT 文字列データを格納したデータベースと照合して評価を行った。

2.2 大正蔵『一切経音義』の特徴

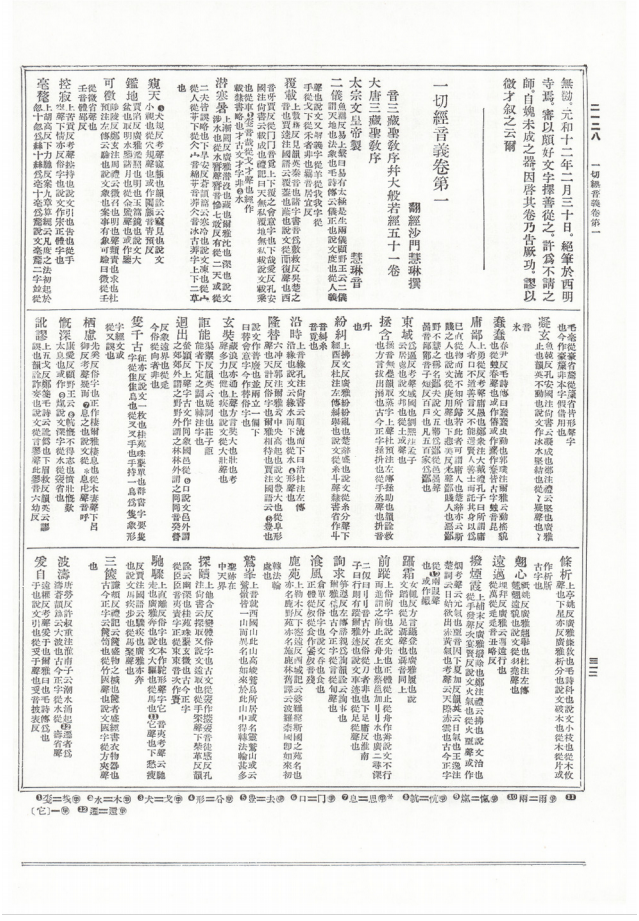

図 1　SAT 上の大正蔵『一切経音義』の版面

大正蔵『一切経音義』『続一切経音義』の紙面は、図 1 のようなものである。本文が縦書き三段組で、脚注を除いて二重枠で囲まれているのは大正蔵を通して共通のスタイルであるが、本文のほとんどが他の経文部分にはあまり見られない割注主体の組版となっているのが対照的である。

本文は、タイトル行の量によっても変動するが、1 段につき 25 行前後である（割注 2 行分を 1 行と数える）。割注によって 1 行が膨らむため、他の箇所と比べて行数は少ないが、1 行あたりの字数は割注換算で 48 字前後と非常に密集している。ほとんどの行は見出し語で始まり、その後に割注で解説が続く。少数の行のみ割注が次の行へ持ち越される。

使用されている活字のサイズは、号数制で大きい方から順に四号・五号・六号である。それぞれ大見出し（実質的に巻のタイトルのみ）が四号、小見出し（主に経名）と見出し語が五号、割注が六号となっている。

原本は実際に鉛活字を用いて組まれているため、外枠や行の向きが完全に水平・垂直ではなく、ページによって不規則に歪んでおり、その程度はまちまちである。また、ほとんどのページは印刷の濃度が比較的均一であるが、まれに他のページと比べて明らかに薄い、あるいは濃いものが存在する。特定の文字だけがかすれていることもある。

原本固有の、または SAT によるスキャン時に生じた揺れによって、ページ全体に対する枠や文字の座標は一定しない。

2.3 作業環境

画像処理にはオープンソースの画像処理ライブラリである OpenCV*[c] (ver. 2.4.10)を利用した。当該ライブラリには公式の C, C++, Python のバインディングが存在するが、今回は別に配布されている ruby-opencv*[d]パッケージ(ver. 0.0.14)を利用して Ruby 処理系から操作した。

作業環境は当初 Windows 7 64bit を考えていたが、執筆時点では ruby-opencv の動作に問題があったため、VMware Player 上の Ubuntu 14.04 に Ruby 2.2.1 をインストールした環境で作業を行った。

2.4 作業手順

2.4.1 ページ画像の分割

図 1 にみられる通り、大正蔵のページ画像は三段組であるため、行を区分けるためには段で分割する必要がある。

元画像を二値化し、Canny アルゴリズムで輪郭を抽出したデータを元に枠線を抽出することを試みたが、単純な Hough 変換ではページのわずかな歪みのために十分な長さの線分が得られず、輪郭検出では印刷のかすれによって輪

c) http://opencv.org/
d) https://github.com/ruby-opencv/ruby-opencv

郭が分断されるケースがあった。最終的には、ページの上枠より上に直線がなく、各段の高さは概ね一定であると仮定して、閾値を長くとった Hough 変換によって最も上で検出された直線から縦に一定間隔ごとに領域を切り出すことが最も確実と思われた。

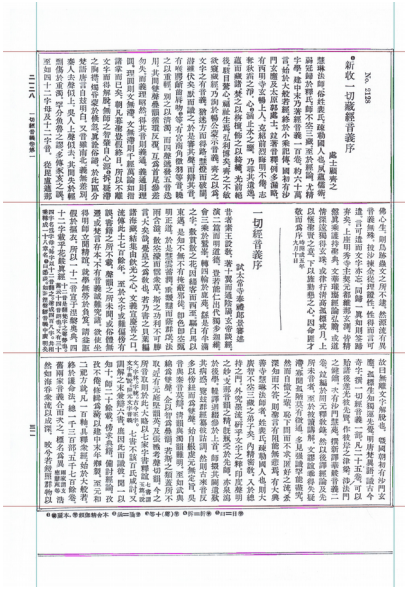

図 2　段組検出と分割の視覚化

2.4.2 行の分割

段ごとに分割した画像から、切り出す文字の属する行を特定するために、あらかじめ行の区切りを検出する必要がある。

二値化した画像を横方向に走査すると、文字の含まれる縦列は黒い画素が多く現れ、文字のない間隔は黒い画素がほとんど見られないと仮定し、黒い画素の谷部分を行の区切りとみなすよう処理を行った。ただし、線などの細いノイズや閾値の関して若干の調整を行い、文字の少ない行の誤分割を避けるために付近の列との平均を取るなどした。

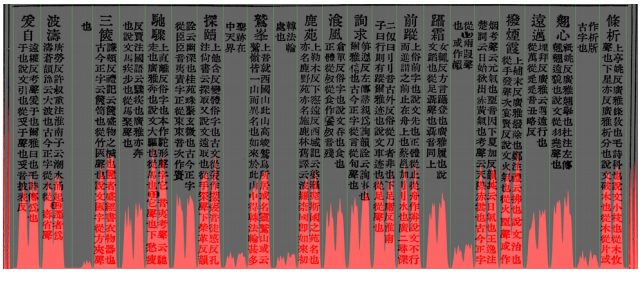

図 3　行の検出基準の視覚化

2.4.3 文字の検出

活字で組まれている大正蔵の文字は、大きさが定型的であり、比較的少ない種類の型枠に当てはめることが容易であると考えられる。

原本への実測でも確かめられる通り、本文には四号・五号・六号活字が用いられているが、それぞれ 13.75pt, 10.5pt, 8pt に相当し、SAT 画像のメタデータにある 600dpi で換算すると約 114px, 87px, 66px 四方のボディサイズとなる計算である[7][8]。しかし実際の画像では六号が 60〜61px 間隔で並んでおり、四号・五号も等倍率の 105px, 80px で換算する方が適していた。また、活字組である関係上、輪郭の重複は発生しないはずであり、複雑な形状は考慮しなくてもよいと仮定し、矩形ベースで処理を行った。

実際の処理では、Canny 化データから輪郭を抽出し、重複する外接矩形を統合した矩形を文字の候補とみなし、上から順に文字枠の適用を試していった。活字のサイズにはわずかな差（主に外字など）があり、また活字がやや回転しているため画像の辺と平行な矩形では不都合なケースが散見されたため、最終的には五号・六号については通常枠とわずかに大きめの枠を 2 種類用意し、わずかな領域の重複は許容して照合した。その他、丸囲み数字などの補助的な記号のためのサイズも用意した。

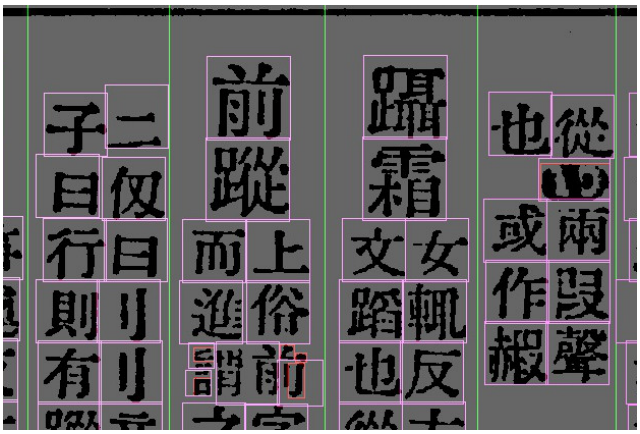

図 4　文字検出の視覚化

3. 結果

サンプルとして 100 ページ（300 段）分の画像を処理させた出力結果（処理データ）と、当該部分に相当する SAT テキストデータ（元データ）を突き合わせて結果を比較した。本来 SAT の元データは誤りを含む可能性があり、研究を通して校正していくべきものであるが、ここでは収録行数・文字数については基本的に誤りがないものとして基準に用いた。

行数検出に関しては、処理データが 6,848 行を検出したのに対し、元データは 7,182 行存在した。段ごとに集計すると、不一致行数は 364 行（5%*e）であった。行数が完全に一致した段が 300 段中 187 段（62%）、違いが 1 行以下の段が 244 段（81%）と比較的正確だが、一方で差異が 10 行以上

e) パーセンテージは全て元データ比、節内以下同じ。

という極端な段が 8 段（2%）存在し、これらの行だけで 137 行の不一致を示している。確認したところ、いずれも傾きや歪みが大きいか、活字サイズが混在しているために目視でも行の判別が難しいといった事情が明らかなページであった。少なくとも傾き補正に関しては処理上考慮に入れなければならないと思われる。また、同じページ内でも歪みの具合で一番上または下の段のみ突出して成績が悪いものがあり、これらは閾値のボーダーライン上に存在していたと考えられるが、なお詳細な調査が必要である。

文字枠検出に関しては、処理データが 175,693 文字を検出したのに対し、元データは 183,355 文字存在した。段ごとの集計では、不一致字数は 7,682 字あり、不一致がなかった段が 4 段（1%）のみ、不一致が 5 字以下の段が 48 段（16%）、不一致が 1%以内の段が 73 段（24%）と、まだ改善の余地があると思われる。不一致が 10%以上という成績の悪い段（29 段）を調査すると、活字の傾きや歪みのほか、画像の汚れによるものがあった。また、文字の場合は印刷のにじみによって微妙に輪郭が拡張したために認識できなくなるなどのケースがあり、パラメータにデリケートな調整が必要であろう。

文字検出の正確性は統計的な差異のみならず、実際に意図した文字の輪郭を取得できているかの検証も不可欠であるが、全ページを目視で精査することは時間的に厳しいため、1 ページのみサンプル調査した。このページ内には元データ 2,071 字中、処理データ 2,034 字検出された（28 字不一致）が、加えて輪郭の誤りを 7 字発見した。特に、前後に文字が少ない環境において字画が分散している字（一・二・三など）の輪郭を正しく読み取れなかったケースが特徴的だった。ただし不一致の原因のうち、12 字が底本のシミまたは修正による字画の重複に起因するものであった。底本の印刷問題については現在のアプローチでは対応できないため、多発するようであれば別の手法を取り入れる必要があると思われる。

4. まとめ

4.1 OpenCV について

OpenCV を使うことで、画像解析の専門知識のない筆者でもある程度簡便に画像処理を行うことができた。OpenCV は豊富な道具立てを提供するばかりではなく、ドキュメントが充実しており、またネット上でも多くの情報が提供されている点で、「使いやすい」ソフトウェアであるという印象を受けた。一方で、アーキテクチャに（恐らく専門家にはなじみがあるのだろうが）必ずしも初心者にわかりやすいとは言えない*f概念を使用しているのも事実で、こうした背景知識についての解説は充実していると言えないため、チュートリアルを離れて応用すると、目的に対応する手段がわからない、機能や挙動を検索・把握しにくい、という状況がある。そのため現段階では「敷居の低い」ソフトウェアとは言い難いと思われる。このような事柄は必ずしも全て OpenCV 自体に帰するものではないとは思うが、ここのフォローアップが充実すればより親しみやすい間口の広いツールとなるだろうと思われる。

4.2 今後の課題

本研究のために実用的といえる水準を達成するためには、より精度の高い文字位置検出アルゴリズムを実現する必要があると思われる。今回の範囲では単一の基準を固定して全場面に適用するという単純なアプローチのみであったが、必要に応じて場合分けなどの精緻化や、確率的・機械学習的なアプローチも取り入れ、組み合わせていければと考えている。また、OpenCV などを通じて画像処理のより高度なアルゴリズムを活用していくことも検討していきたい。

別の面からは、現在の実行環境では試行に時間がかかるという問題がある。今後継続的に改良していくためには、作業環境や処理効率の面でも見直しが必要であると思われる。

なお、本報告ではページ下脚注部分の検出については扱わなかったが、本文と同様の手法を援用することはさほど難しくないと考えられるので、今後取り組んでいきたいと思う。

参考文献

1) 高田時雄 (2010)「藏經音義の敦煌吐魯番本と高麗藏」高田時雄（主編）『敦煌寫本研究年報』6: 1-13.
2) 陳五雲・徐時儀・梁曉虹 (2010)『佛經音義與漢字研究』. 南京: 鳳凰出版社.
3) 王一凡・永崎研宣・下田正弘 (2014)「UCS 符号化という観点からの『慧琳撰一切経音義』の検討」京都大学人文科学研究所附属東アジア人文情報学研究センター『漢デジ 2014：デジタル翻刻の未来』61-66.
4) 黎養正 (2007)「重校一切經音義序」一誠長老（総主編）『頻伽精舍校刊大藏經』為: 3-4. 長春: 吉林出版.
5) 大藏小僧（編） (1935)「藏經逸聞集」『ピタカ』3(1): 35-52.
6) 王一凡 (2015)「大正新脩大藏経所収『一切経音義』の文字同定と索引構築の試み」京都大学人文科学研究所附属東アジア人文情報学研究センター（編）『東洋学へのコンピュータ利用 第 26 回研究セミナー』27-31.
7) 小宮山博史 (2009)『日本語活字ものがたり――草創期の人と書体』. 東京: 誠文堂新光社.
8) 「活字の大きさ」朗文堂 アダナ・プレス倶楽部『活版印刷用語集』. http://robundo.com/adana-press-club/glossary/ka.html#ka_size-type.

f) 例えば、各種機能名が原論文著者名をそのまま冠していたり、動作の似た複数の型が共存していたりという点が挙げられる。