

仏典データベースのための画像処理について

張 蓉⁽¹⁾ 仁野 洋平⁽¹⁾ 田中 猛彦⁽²⁾ 中川 優⁽²⁾

青木 進⁽³⁾ 宇都宮 啓吾⁽⁴⁾ 落合 俊典⁽³⁾

⁽¹⁾和歌山大学システム工学研究科

⁽²⁾和歌山大学 システム工学部

⁽³⁾国際仏教学大学院大学

⁽⁴⁾大谷女子大学 文学部

金剛寺一切経を対象としたデジタルアーカイブ化を目的として、デジタルカメラで撮影された経典画像を解析し、認識された文字位置の座標情報をデータベースに格納する一連の画像処理システムを開発している。画像を縦方向または横方向にのみ走査することで、本文でない領域を求め、文字領域を認識する。求めた座標情報を用いて、重なり位置の推定や、合成して1枚の画像として見せるための座標の計算を試みている。

Image Processing for Buddhist Canon Database

Rong ZHANG⁽¹⁾ Yohei NINO⁽¹⁾ Takehiko TANAKA⁽¹⁾ Masaru Nakagawa⁽¹⁾

Susumu AOKI⁽³⁾ Keigo UTSUNOMIYA⁽⁴⁾ Toshinori OCHIAI⁽³⁾

⁽¹⁾Graduate School of Systems Engineering, Wakayama University

⁽²⁾Faculty of Systems Engineering, Wakayama University

⁽³⁾International College for Postgraduate Buddhist Studies

⁽⁴⁾Ohtani Women's University

To construct a database system for Buddhist canons of Kongoji Issaikyo, we develop an image processing system which takes images taken by digital cameras as input and outputs coordinate data of the image. By scanning the images vertically or horizontally, the margin to be removed is calculated and each character regions are derived. Using the coordinate data, we attempt to guess the position of joint between two adjacent images and to calculate the Y coordinates for combining a series of images of a scroll.

1 まえがき

大阪府河内長野市にある真言宗御室派大本山・天野山金剛寺には、「金剛寺一切経」と呼ばれる仏教経典約4,500巻が今日に伝えられている。この一切経は平安時代の仏教を知る上で

非常に貴重な資料である。また、経文（文章）の読み方を朱の符号で示した「訓点」や振り仮名、また奥書と呼ばれる日付や書き写した人間のコメントは、国文学・歴史学・国語学の分野においても非常に重要な史料である。

我々は、この金剛寺一切経を対象として、デ

デジタルカメラによる撮影から歴史的遺産である經典のデジタルアーカイブ化を行っている[1,2]. “アーカイブ”とは本来、文書保管所を意味する. デジタルアーカイブ化とは、歴史文化財や伝統文化遺産などを高精細デジタル映像で記録し、マルチメディアデータベースとして保管するものである. デジタルアーカイブ化は次のような手順で進められる[3].

1. デジタルアーカイブ化するための価値があるかどうかの検討
2. 貴重資料をもつ資料所有者の同意
3. 悪影響を与えないような資料の撮影、デジタル化（カメラのフラッシュなど）
4. デジタル情報の整理（データベース化を含む）
5. 研究者等への情報提供および一般公開

最近では、撮影されたものを図書館などに電子記録として保存し、電子図書館として運営しているところもある.

石碑などに直接彫られ残された碑文と呼ばれる古典資料や、古代の蔵書として扱われる書物形式のものは、一枚の画像に収めることが可能である[4,5]. しかし巻子として保存されている經典を、一枚の画像で収めることは非常に困難である. Photoshopなどの画像処理ツールを用いて手作業で構成することは可能であるが、手間がかかる上に、出来上がった画像の幅が数万ピクセルにもなり、閲覧や印刷が容易に行えない. ここで、撮影された各コマから情報抽出しておいてデータベースで管理し、利用者のニーズに応じて取り出し、合成などができるシステムを構築できれば、重要な経巻資料の電子化が大きく進む画期的なアーカイブとなると考える.

現在までに我々は約2,100巻余りのカラー撮影を終えた. 撮影枚数は約4万4,000コマ、データ量は320GBに及んでいる. 今後の計画では、

最終的に全巻の撮影を完遂すれば12万コマ、約1.2TBに達する見通しである. カラー撮影によって得られた画像は、視認性に優れている. しかし、内容を精査する上で画像上の文字が自動的に活字化されることを、せめて検索可能になることを研究者は切望している. 検索性を持たせるため、研究者自身がこの画像の文字を読み、テキストを入力しようとしても、先述の通り、金剛寺一切経は4,500巻以上もの量があり、現実的ではない. 幸いにも金剛寺一切経については『大正新脩大藏經』という活字本との対応があり、これにはテキストデータが存在する. この既存のテキストデータと画像の文字領域を文字単位で対応付け、データベースと検索閲覧インタフェースを提供して読解を支援するのが、本研究の目的である. 木簡等を対象とし、文字単位での解読とデータベース登録を行うシステム[6,7]も存在するものの、1文書あたり数千字以上の文字を持つ手書き古典籍への解読には別のアプローチを要すると判断し、独自のシステムを構築している.

2 仏典画像

図1に、撮影された放光般若経巻第一（『大正新脩大藏經』第8冊，No. 221）の1コマを示す.

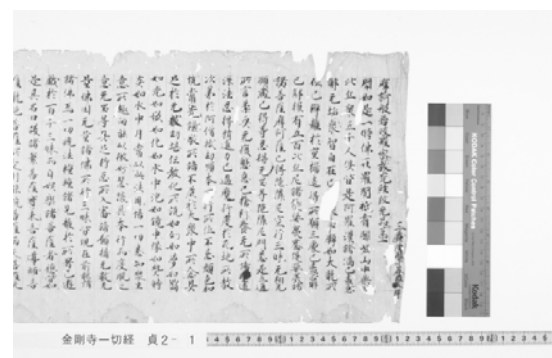


図1：撮影された仏典画像の例

撮影の対象となる経典は卷子と呼ばれる紙の巻物の体裁を取っており、撮影に際しては各コマ断片的にならざるを得ない[8]。通常、1本の巻物は17字詰で450～600行、7,500～10,000字程度である。1コマは大体30行ずつとして撮影を行い、1巻は20～30コマ程度で構成される。各コマの左右両端は、隣のコマとのつながりを示すために3～5行ほど重複を持たせている。人間が見ればこの重複は容易に分かるが、文字単位でデータベースに登録するためには、この重なり箇所、言い替えると、コマごとに左右でどの位置からは不要になるかも導出する必要がある。撮影された画像のフォーマットと画素数は、初期はRAW形式で500万画素であったが、カメラを変更して、現在では1200万画素のNEF形式として保存している。このままのフォーマットではコンピュータで使用するのには難があるため、閲覧や画像処理に適した形式に変換している。画像サイズも200万～250万画素程度に減らしている。

画像処理のためのフォーマットとして、PNG (Portable Network Graphics)を使用している。これにより、情報量を落とすことなく画像を処理、保存できる。PNG画像は、標準的なWebブラウザで閲覧可能であるとともに、画像処理をするためのフリーのライブラリやソフトウェアも充実している。

しかしデジタルカメラで撮影された画像には一応の規則性があるもののコンピュータで機械的に処理する場合、複数の画像にまたがって撮影されているためどうしても下準備として画像の加工が必要となってくる。1行17字を標準とするが、そうでない行も頻繁に見られる。碑文と異なり、文字の位置には規則性がない。そのため文字の位置の認識には困難が生じてくる。そこで筆者らは、縦方向および横方向へのスキャン(走査)によって、文字領域を認識

する方法を検討し、試作した。

3 画像処理

最初に、画像処理の必要性について述べる。前節で述べたように、一つの経典に対して、重なり合う20～30枚程度のコマで撮影されている閲覧者に提示する際には、重なり部分や、上下左右の余白を取り除いて提示したい。また、画像では検索が困難であるが、大正新脩大蔵經のテキストデータと対応付けることで、仏典の全文検索や、仏典内での検索が容易になる。そこで、文字認識まで行わなくても、文字領域認識、すなわち先頭からどこに文字領域があるかを順番に求め、その座標を保持することで、大まかながらテキストデータとの各文字との対応付けが可能となる。

以上の構想のもと、図1の流れで、撮影された仏典画像から文字領域などの座標情報を抽出する処理を実装した。なお、処理時間については、「行領域の抽出」と「文字領域の抽出」で時間がかかる。筆者らの環境では、260万画素の画像1枚あたり40秒程度かかっている。

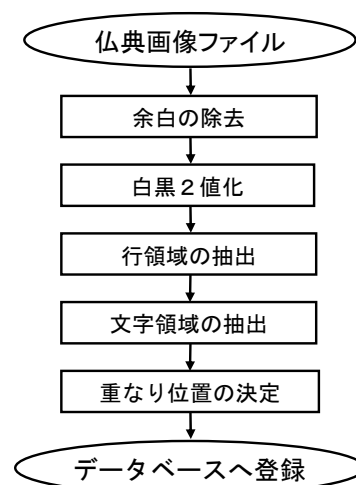


図1. 画像処理の流れ

3.1 余白除去

撮影された画像には、上下に余白がある。経巻の最初のコマは右端、最後のコマは左端にも余白がある。下には長さ確認のための目盛りがある(この目盛りは経典の位置を示すものではないため、この情報を用いて重ね合わせ処理をすることはできない)。さらに、巻の最初のコマには色見本もある。これらの部分は以下の処理で不要であるほか、減色処理に影響を及ぼすので、最初の段階でカットし、本文のみの画像を抽出しておく。

上側の余白の検出手順を説明する。残り3方向の処理も、色見本を識別するための処理を右余白に施している点を除き、同じである。

(1) 探索位置を決める。予備実験の結果、画像の幅の平方根の間隔で、上端(Y座標は0)から下方向に探索し、列ごとに余白を求める。例えば、画像の幅が $1025=32^2+1$ ピクセルであれば、X座標は0, 32, 64, ..., 1024である。

(2) 探索の開始位置から1ピクセルずつ下方向に探索して画素(座標のRGB情報)を求める。画像が紙の色と思われる画素が3点連続すれば、その連続した箇所直前のY座標を、その列での「上余白の下端」とみなす。紙の色には黄色、褐色、灰色などがあり、それぞれ紙の色となる色の特徴は異なるため、紙の色として認識する条件を別々に設定している。

(3) 探索列ごとに得られた、上余白の下端をソートして、3番目に小さい値を、この画像の上余白の下端とする。最小や2番目に小さい値で切ると、上の余白が小さくなり、無駄が多いことを、いくつかの画像に対する予備実験で確認している。

3.2 2値化

余白を切り落とした画像に対して、白黒2値化を行う。ただし、この状態で直ちに白黒2値にしても質の良い画像は得られない。

この時点での画像の色数は1画像当たり数万色であり、まずこれを、256階調のグレースケール画像に減色する。次に、減色した画像ファイルから、色濃度(0が黒、255が白に対応する)と頻度(その画像にその色が何ピクセル含まれているか)を求める。重み付け平均により画像の平均色濃度を求め、定数Mを掛けて、白黒分離の閾値とする。Mの値は、画像にもよるが、筆者らの環境では0.6~0.75の間から選んでいる。Mの値が小さくなければ、生成される画像は白っぽくなる。

最後に、減色した画像ファイルから、各色の色濃度が閾値以上であれば白、そうでなければ黒に、色情報を変更する。

3.3 行領域の認識

前節で白黒2値化した画像に対して、行領域の抽出を試みる。簡単に言えば、列ごと(縦方向)に走査して黒の点の個数を求め、これが画像ごとに決められた閾値以上であれば、その列は文字のある行と推定する。以下、文字のある行と判断した列を「黒の列」、そうでない列を「白の列」と呼ぶ。

あらかじめ、文字領域と判定するための閾値を設定しておく。これは、画像の高さの定数倍とすればよい。その定数として、筆者らは1/50を選んでいる。例えば、画像の高さが1,200であれば閾値は24となる。

列ごとに黒の列か否か(論理値)を求め、配列にする。この配列は、理想的には「連続した白の列」と「連続した黒の列」の繰り返しになる。実際にはノイズや誤差などもあるため、補正を行う。具体的には、「連続した黒の列」の長さが定数値(例えば10)以下なら、それをすべて白の列に変更する。逆に、白の列を黒の列に変更することはない。

最後に、列ごとの論理値の配列を先頭から走査して、連続した黒の列ごとに「その先頭位置

「(X座標)」と「列の長さ(幅)」を求め、その配列を、行領域の抽出処理の結果とする。



図2. 行領域の認識例

図2は、抽出した行領域を箱で囲ったものである。ここまでの処理に関して、40枚程度の画像で適用したところ、大きな問題となる誤認識は見当たらなかった。

3.4 文字領域の認識

前で求めた行領域 (X座標と幅) 一行ずつに対して、文字領域を求める。そのX座標をleft, 幅をwidthとしたとき、白黒2値画像を行単位で走査する。具体的には、Y座標ごとに、画像中の領域 $\{(X,Y) \mid \text{left} \leq X < \text{left} + \text{width}\}$ (線分になる)で黒の点の数を求める。

行領域の抽出方法と同様に、文字領域と判定するための閾値を設定しておく。これは、widthの定数倍とすればよい。その定数として、筆者らは1/8を選んでいる。例えば、width=40であれば閾値は5である。Y座標ごとに求めた黒の点の数と、この閾値を比較すれば、黒の行か否か(論理値)がY座標ごとに求められる。

列ごとに黒の列か否か(論理値)を求め、配列にする。この配列は、理想的には「連続した白の行」と「連続した黒の行」の繰り返しになる。ここでもノイズや誤差などを考慮して、補正を行う。具体的には、「連続した白の行」の長さ

が定数値(例えば9)以下なら、それをすべて黒の行に変更する。逆に、黒の行を白の行に変更することはない。

最後に、行ごとの論理値の配列を先頭から走査して、連続した黒の行ごとに「その先頭位置(Y座標)」と「行の長さ(高さ)」を求め、この配列、left, およびwidthを、ひとつの行領域における文字領域の抽出結果とする。図3に、ある行領域に対する抽出結果を示す。



図3. 文字領域の認識例

3.5 重なり部分の検出

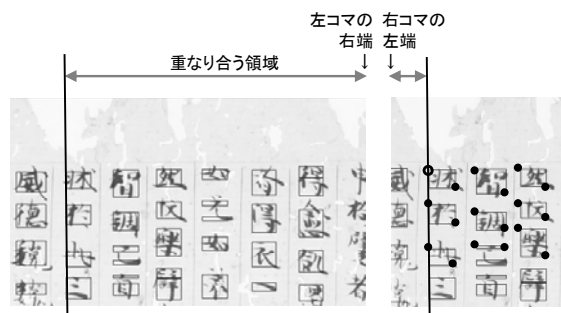


図4: 重なり部分

3.4節で求めた文字領域情報のいくつかを用いて、重なり部分の検出を試みた(図4)。まず、図4右のコマのように、ある行頭の文字領域を基点として、3行3列の文字領域について、それぞれの左上と右下の位置(図4で丸のついた箇所)について、基点の左上を原点とした相対

座標を求める。そして、各点のX座標とY座標を1列に並べて、36項からなる非負整数ベクトルを作る。これを検査ベクトルと呼ぶ。

右側のコマでは、文字領域として認識されている中で左上の文字を基点として、検査ベクトルを作る。左側のコマでは、すべての連続する3行について、検査ベクトルを作る。そして、右側のコマの検査ベクトルと、左側のコマの各検査ベクトルとで距離を求める。ここでの距離として、マンハッタン距離、すなわち各要素の差の絶対値の合計を用いる。この値が小さいほど、検査ベクトルの元となった18個の点の形状が類似していることを意味し、実際、左側のコマについては距離が最小となった行の左端X座標と、右側で検査ベクトルを求めた領域の左端X座標とを重ね合わせて、不要部分を除去すればよい。

1回の検出処理で生成する、検査ベクトルの個数は、右側のコマは行領域数に関わらず1であり、左側のコマは行領域数から2を引いた数となる。文字領域情報のみを用い、画像処理は行わないので、検査ベクトルの生成、およびその比較は極めて高速に行える。

3.6 データベースへの登録

3.5節までで獲得できた座標情報は、余白を取り除いた領域(左上XY座標, 幅, 高さ), 各行領域の左端X座標と幅, 行領域ごとの上端Y座標と高さ, そして左右との重ね合わせX座標である。これらをPostgreSQLのデータベースサーバに登録している。データベースを構築するに当たり、テーブルを設計した。主要なテーブルは、画像, 行領域, 文字領域であり、画像と行領域に1対多, 行領域と文字領域にも1対多の関係を持たせている。したがって、文字領域のテーブルには、そのX座標と幅は登録されおらず、行領域テーブルと合わせて参照するこ

とで獲得できる。これは一見煩雑に見えるが、あらかじめ適切なSQL文を用意しておけば、実行時間が大きくかかることはない。その一方で、「一つの行領域に属する文字領域のX座標と幅は共通である」という性質を利用して、データ管理の効率をよいものになっている。

4 評価

4.1 文字領域の認識

文字領域認識の精度を評価するため、仏典画像に対して、文字領域として認識された箇所を箱で囲っておく。図5はその一部である。

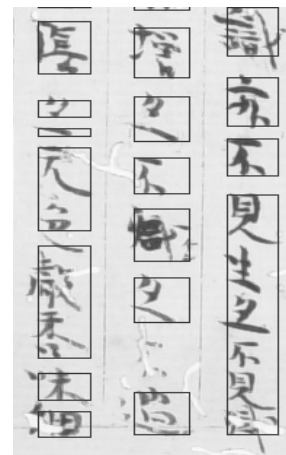


図5. 文字領域認識の例

この箱を目視により検査し、それぞれの文字に対して優, 良, 不可, 失敗のいずれかに判定する。その基準としては、90%以上を箱で囲んでいれば「優」(図5の右の行, 上から2番目と3番目の文字), 60%以上であれば「良」(中央の行で最も下の字が該当する。しんじょうが認識できていない), 60%未満, または2文字以上に囲ってたり、一つの文字を複数の文字領域で囲っている場合は「不可」, 箱がまったく囲めていないものは「失敗」(中央の行, 下から2番目の文字)とした。

2種類の経典について文字領域認識を行い、

重ね合わせで排除されることになる文字領域も含めて、精度を判定した結果を、表1に示す。

表1：文字領域認識の結果

	摩訶般若波羅蜜放光品第一		大方広佛華嚴經第三	
	優	1691	17.17%	754
良	5852	59.42%	2652	30.10%
不可	2303	23.39%	5375	61.01%
失敗	2	0.02%	29	10.33%
合計	9848		8810	

摩訶般若波羅蜜放光品第一は、文字間の切れ目が人間の目で見ても分かりやすく、文字領域認識も比較的成功的と考えられる。とはいえ、優の文字は多いとはいえ、改善していかなければならない。

4.2 重なり部分の検出

画像を切断、合成するのではなく、検出した重なり位置に対して、図4のように線を引き、目視により、重なり位置が正しいかどうかを判断する。ここでも、2種類の経典に対して評価した。摩訶般若波羅蜜放光品第一は20枚の画像からなるので、重なり位置は19箇所である。大方広佛華嚴經第三は18枚のため17箇所である。結果は表2の通りである。大方広佛華嚴經第三は重なり位置の判断がよくない。これは、文字領域の認識精度が悪かったためと考えられる。文字領域認識の精度が向上すれば、重なり位置の精度も向上することが期待できる。

表2：重なり位置の結果

	摩訶般若波羅蜜放光品第一		大方広佛華嚴經第三	
	成功	18	94.7%	11
失敗	1	5.3%	6	35.3%
合計	19		17	

5 画像の結合

卷子としての経典を1枚の写真として撮影するのは不可能であるため、何枚かに分けて撮影していた。ここでは、各コマの画像と、座標情報を活用して、1枚の経典画像に合成する方法を検討する。重ね合わせが適切に行えれば（失敗のものは人手で座標を修正しておく）、X座標については単純に右から左に並べていけばよい。ここでは、Y座標の調整を考える。

まず、連続する2枚の画像で、Y座標の調整方法を考える。この処理で使用するのは、余白を削除した画像と、重ね合わせ座標と重なる文字領域の行頭の文字について、余白を削除した画像からみたY座標である(図6)。左側のコマと右側のコマについて、この座標をleftBoxY, rightBoxYと表記する。このとき、 $\Delta Y = \text{leftBoxY} - \text{rightBoxY}$ が、2枚のコマをY方向に調整するための値となる。 ΔY が正のときは左側画像が上に配置され、負のときは右側画像が上に配置される。

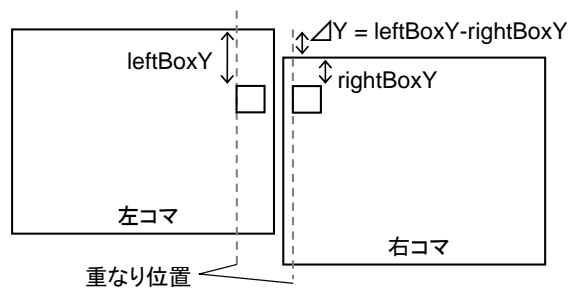


図6：重なりにおけるY座標

この方法ですべての重なり位置について ΔY を求め、重なるべき文字を合わせて配置すると、図7のようになる。画像サイズは、まず暫定的に左端画像のY座標を0として配置し、それぞれの上端Y座標と下端Y座標を求めていく。すべてを(仮想的に)重ね合わせたときに、

下端Y座標の最大値(bottom)と上端Y座標の最小値(top)の差を求め、 $bottom - top$ を、合成してできる画像の高さとすればよい。さらに配置する際には、左端のコマのY座標を、0ではなく $-top$ とすればよい($top \leq 0$ に注意)。

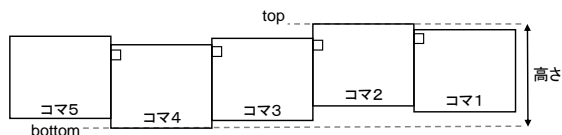


図7：仏典の合成におけるY座標

6 あとがき

本稿では、金剛寺一切経画像データベース構築に不可欠な画像処理について述べた。文字領域認識を行い、その座標情報を用いることで、コマ同士で重なり合う位置を推定し、合成して1枚の画像にすることが可能となった。

今後の課題としては、認識精度と処理速度の向上、他の仏典画像への適用、仏典検索・閲覧システムの開発などが挙げられる。

謝辞

仏典画像のデータベース化に関して、大谷女子大学 大倉孝昭教授よりアドバイスをいただきました。また、画像の結合については、和歌山大学交換留学生 Martin Rolland 君との議論により得られたものです。深く感謝します。

本研究は部分的に日本学術振興会科学研究費補助金 基盤研究(A) 課題番号 15202002 の補助を受けた。

参考文献

[1] 落合俊典：「金剛寺一切経の基礎的研究と新出仏典の研究」，平成12年度～15

年度科学研究費補助金基盤研究(A)・(1) 研究成果報告書，課題番号 12301001，364p (2004)

[2] 仁野他：「仏典画像閲覧のためのデータベースシステムの構築」，情報知識学会誌, Vol.15, No.2, pp.15-18 (2005)

[3] 阿部研究室古文書電子テキストアーカイブ (AFC) 研究グループ，
<http://monju.elec.ryukoku.ac.jp/abe/AF/C/tategaki/tategaki.html>

[4] 安岡孝一：「透明テキスト付き画像のいざない」，東洋学のコンピュータ利用第14回研究セミナー，京都大学学術情報メディアセンター第71回研究セミナー (2003)

[5] 坪井 昭憲，八村 広三郎，吉村 ミツ：「江戸期版本画像からの文字切り出しの試み」，情報処理学会研究報告，2005-CH-066, pp.53-60 (2005)

[6] 石川正敏，波多野賢治，天笠俊之，植村俊亮，勝村哲也：「歴史文献のための電子スクラップシステムの設計」，人文科学とコンピュータシンポジウム(じんもんこん 2003), pp.227-234 (2003)

[7] 末代誠仁，斎藤恵，蜂谷大翼，中川正樹，馬場基，渡邊晃宏：「木簡解読支援システムの基本設計と試作」，人文科学とコンピュータシンポジウム(じんもんこん 2004), pp.215-220 (2004)

[8] 鈴木 昭夫，吉田 成，岡宮 誠一，田口 榮一，鷺野谷 秀夫：「研究者のための資料写真の撮り方」，理工学社，pp.2-30 - 2-31 (1991)