

系図処理プログラムの開発および公開

村川猛彦^{†1}

概要：系図には人物の詳細や人間関係などが記載されており、人文学の研究において有益な資料である。系図の読解や分析を支援するため、本研究では、系図画像および位置情報付きテキストを入力にとり、系図画像上の線分を認識しながら、人物同士および人物と付随情報の対応付けを行うプログラムを、Ruby 言語を用いて開発し、その一式をGitHub に公開した。同梱のプログラムを実行することで、『豊臣秀吉譜上』の系図画像を処理し、得られた関係を出力できる。認識結果の修正をプログラムから指示することも可能にしており、これを用いて、『真言宗付法血脈』に見られる横線を結ぶ斜線について対処を図った。

キーワード：デジタルアーカイブ、系図、画像処理、ソフトウェア開発

Development and Publication of a Genealogy Processor

TAKEHIKO MURAKAWA^{†1}

1. はじめに

系図には人物の詳細や人間関係などが記載されており、人文学の研究において有益な資料である。系図は史料の撮影画像（例えば[1]）、出版物（例えば[2]）、文書ファイルなどの形態で表現されているが、JPEG などの画像にしておけば、ブラウザを用いた閲覧が容易となる（最近では、ブラウザの機能により画像の拡大・縮小表示も可能である）。しかし画像のままでは、検索を行うことが困難である。そこで画像に記載の文字や、系図における情報どうしのつながりを自分の計算機もしくは外部のサーバに保持しておき、必要なときに取り出せるようにしておけば、より効果的な系図のブラウジング環境が期待できる。

本研究で対象とする系図は、先行研究[3][4]と同様に、一族の代々の系統を書き表した図表とし、親子関係のみならず、学芸の師匠から弟子への師資相承の関係（師弟関係）を表したものが該当する。ただし婚姻関係は対象としない（婚姻関係を含む系図へのアプローチに関しては[5][6]などが詳しい）。また系図の人間関係は基本的にグラフ理論における根付き木であるという観点から、師弟関係についても親子や兄弟などの表記を用いる。

本研究では、[3]からの発展として、改良したプログラムの概略を、適用例とともに報告する。系図画像と位置情報付きテキストを対象とした処理プログラムを統合・整理して“Keizu Deconstructor”と名づけ、GitHub にて公開した[7]。このプログラムを用いることで、コマンド実行により、系図画像に対する処理結果を生成でき、情報の関連付けを把握できるようにしている。先行研究では想定していなかった、系図に見られる描かれ方に対し、本プログラムにおける対応の方針についても述べる。

開発したプログラムの命名についてその意図を述べておく。これまでの系図ソフトウェアおよび系図に関する情報処理分野からのアプローチ[5][8][9]はいずれも、系図を「構築する(construct)こと」に主眼が置かれてきた。本研究でも、画像などから獲得できた情報を、TEI (Text Encoding Initiative)のガイドラインやGEDCOM (Genealogical Data Communication)、IIIF (International Image Interoperability Framework)などのフォーマットに沿うようファイルに保存し、閲覧やさならなる分析などに活用することを意識しつつも、[3]および本稿では、いかにして既存の系図を「解体する(deconstruct)か」に関心を持っており、それを表す最適な語として、“deconstructor”を採用した次第である。

2. 準備

2.1 系図画像

図1に『豊臣秀吉譜上』の画像の一部を示す。本稿執筆時点では、[1]のWeb ページにブラウザでアクセスを行い、「デジタル画像とメタデータの一括ダウンロード (ZIP 形式、約 880MB)」によりファイル (200021823.zip) を取得してから伸張し、200021823/image/200021823_00171.jpg として作られるファイルより参照できる。系図に書かれる文字列は人物名と、人物を説明するための情報（付随情報）に大別され、図1においては、「秀吉」「秀長」「女子」が人物名、「不詳其父」などが付随情報となる。

^{†1} 和歌山大学
Wakayama University

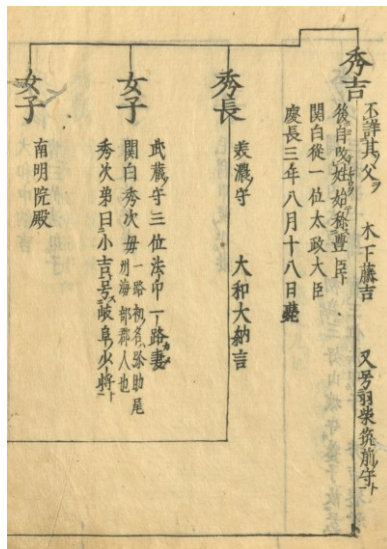


図1 系図画像例 (『豊臣秀吉譜上』)

Figure 1 Example of genealogical image (1).

出版物 (『真言宗付法血脈』[2]) の系図記載例を図2に示す。この系図は、原本をもとに翻刻者が読みやすくし、また必要に応じて情報を付記している。「大法師泰範」「大法師智泉」などが人物名、「天台傳教大師同法山別当」などが付随情報にあたる。なお、系図が出版物、またはPDF形式などの文書ファイルの場合には、画像に変換しておくものとする。

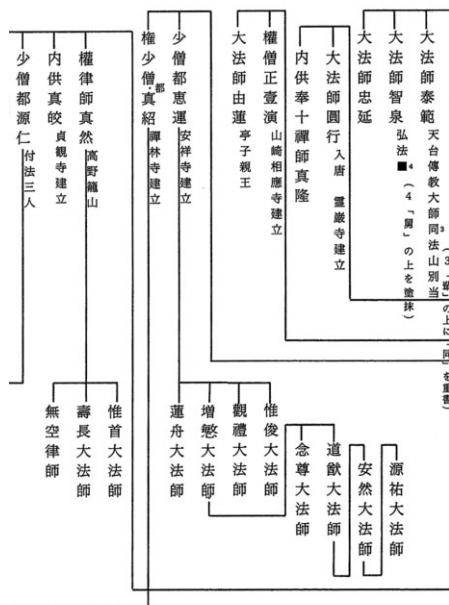


図2 系図記載例 (『真言宗付法血脈』)

Figure 2 Example of genealogical image (2).

これらの系図を参照して得られた共通点を整理する。まず、人物名の近くにその人物の付随情報が記載されており、人物の親子関係や兄弟関係は、縦と横の線分の組み合わせでつながっている。

大部分において、人物から伸びる線分は縦の線分であるが、醍醐寺本『伝法灌頂師資相承血脈』[10]では、図3において「實耀」と「仁杲」の間に見られるような、左右に隣り合う人物をつなぐ横線が多数出現する。そのほか図3では、「乗海」の最初の子の「實海」に下線が伸びているが、その先は人物名ではなく2行からなる付随情報であり、別の箇所(図3の左側)に、「實海」を筆頭とする枝分かれのない親子関係が図示されている。また斜線が[2]に見られる(図4-5)、人物名の上の斜線は対象外とする一方、2本の横線を結ぶ斜線は取り入れる必要がある。

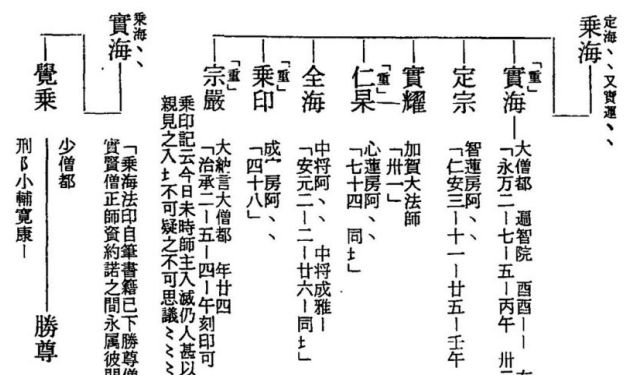


図3 注意を要する系図例

Figure 3 Genealogical tree including noteworthy elements.

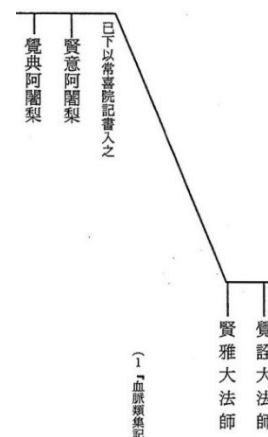


図4 斜線の出現例(1)

Figure 4 Diagonal line seen in genealogical tree (1).

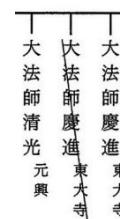


図5 斜線の出現例(2)

Figure 5 Diagonal line seen in genealogical tree (2).

2.2 位置情報付きテキスト

本研究では系図画像に加え位置情報付きテキストを使用する。位置情報付きテキストはプレーンテキストもしくは XML 文書であり、1つの行に、人物名もしくは付随情報の記載文字列、画像上の位置を表す左上 XY 座標、文字列の大きさを表す高さ・幅の数値情報を記述する。

例えば図 1 に見られる「秀吉」に対し、「String CONTENT=" 秀吉 " WIDTH="111" HEIGHT="178" HPOS="1032" VPOS="122"」と記述しておけば、左上座標は (1032,122)、幅と高さはそれぞれ 111 と 178 と読み取ることができる。

CONTENT をはじめとする属性名は、ALTO (Analyzed Layout and Text Object) [11]と呼ばれる XML のスキーマによるものであり、本研究の実施にあたって、凸版印刷の OCR ソフトウェアが生成した、このフォーマットに基づく XML ファイルを使用してきた。ただし、GitHub で公開した際に、『豊臣秀吉譜上』の 3 ページ分について独自に作成した文字列情報および数値情報をプログラム内に入れ、実行により位置情報付きテキストを生成するようにしている。開発プログラムの処理の都合上、文字コードは UTF-8 に限定される。

このテキストでは人物名と付随情報の区別がなされておらず、系図上の線分の情報も記載されていない。系図画像と位置情報付きテキストの各特性を考慮することで、自動処理によって効率よく情報の抽出が行えると考え、認識プログラムの開発を進めてきた。

2.3 使用するソフトウェアおよびサービス

プログラムの実行はコマンドラインを使用する。Ubuntu のほか、Windows と Cygwin の組み合わせで動作確認を行ってきた。プログラミング言語には Ruby を採用しており、キーワード引数を使用しているため Ruby 処理系 (ruby コマンド) のバージョンは 2.0.0 以上を必要とする。

Ruby で記述したプログラムの内部より、画像処理ソフトウェア ImageMagick のコマンド convert および identify を呼び出している。convert は画像の抽出やフォーマットの変換、および既存の画像上に線分や図形を描き足して新たな画像を作成するのに使用しており、identify コマンドは画像の幅と高さを獲得する際に実行することとなる。ZIP ファイルの処理に関しては、rubyzip ライブラリが導入されていれば使用し、なければ zip および unzip のコマンドを実行する。

プログラム公開先の GitHub は、ソフトウェア開発プロジェクトのための共有ウェブサービスである。個別のファイルを参照するだけであれば、ブラウザのみでも行えるが、一連のファイルをダウンロードしコマンドライン上で実行するには、通常、git コマンドを必要とする。Git は分散型バージョン管理システムとして広く普及しており、“Git flow”や“GitHub flow”などと呼ばれる、ソフトウェアを

効率よく開発し改善するための手法も確立している。

Docker は、コンテナ型仮想化技術を使ったアプリケーションのためのプラットフォームである。他の仮想化方式に比べ、軽量・高速で動作するという特長がある。Docker は Linux 上で動作するものであるが、Windows や Mac 向けの実行用ソフトウェア (Docker Toolbox) が提供されているほか、クラウド環境でも動作する。このコンテナに、上述のソフトウェア (Ruby 処理系、ImageMagick、Git など) を格納しておくことで (それぞれを自分の計算機にインストールしなくても)、コンテナの中で一連の処理を行い、結果のファイルをホスト計算機内に保存することができる。

3. 処理の流れ

3.1 『豊臣秀吉譜上』への適用例

作成したプログラムを『豊臣秀吉譜上』の系図に適用したときの処理およびデータ取得の流れを述べる。処理は、(1)初期ファイル (画像および位置情報付きテキスト) の作成、(2)位置情報付きテキストの処理、(3)画像処理および線分認識、(4)人物関係の取得に大別される。

この節ではできるだけ実装に立ち入らないようにしているが、若干の用語[12]を使用しているので簡単に述べておく。「オブジェクト」は、いくつかの値 (数値・文字列・他のオブジェクト) で構成される「ひとまとまりのもの」を意味し、「線分オブジェクト」のように、何のオブジェクトであるかを明示することもある。「配列」は、値の並びを表し、順番に格納していくほか、先頭から順に、または何番目かを指定して値を参照することができる。

(1) 初期ファイル (画像および位置情報付きテキスト) の作成

初期ファイルのうち画像は、2.1 節で述べた将軍家譜の ZIP 形式ファイルを伸張して得られる 200021823_00171.jpg および 200021823_00172.jpg から、convert コマンドを用いて対象領域を指定し取り出す。200021823_00171.jpg からは画像の左側を、また 200021823_00172.jpg からは画像の右側と左側について、2400x3400 のサイズで抽出のち 50% に縮小し、1200x1700 のサイズで JPEG 形式として保存する。なお 200021823_00172.jpg の左側 (一連の系図画像の第 3 ページ) に関しては、抽出の前に時計と反対方向に 2° の回転を行い、線分ができるだけ水平・垂直方向になるようにしている。

位置情報付きテキストの作成にあたり、元となるデータはプログラムに記述している。簡単な変換処理で 2.2 節に記載した書式に変換し、ページごとにテキストファイルとして保存する。

(2) 位置情報付きテキストの処理

位置情報付きテキストを開き、行単位で読み出して「文字列、左端 X 座標、上端 Y 座標、幅、高さ」を獲得し、そ

こから、領域に関するオブジェクトを構成する。ページごとに、領域オブジェクトの配列を管理する。

領域オブジェクトそれぞれに対して、人物名か付随情報かの判別を行う。『豊臣秀吉譜上』では、2文字以下の文字列は人物名、そうでない文字列は付随情報となる(文献[3]で述べた、文字列領域の幅と高さに基づく方法でも、認識ミスなく判別できることを、別途プログラムを作って確認している)。オブジェクトごとに、人物名または付随情報の種別を設定しておき後で参照する。

人物と付随情報の対応付けを行う。今回は単純に、各文字列領域の上端中央座標をもとに、付随情報から見て画像上の2次元ユークリッド距離が最も小さい、人物名の文字列領域と対応づけることとした。その結果、1箇所だけ対応付けのミスが生じたため、処理を入れて対応付けを修正している。

(3) 画像処理および線分認識

初期画像について、以下の処理で行う対象(処理領域)の指定を行う。その処理領域の外部、および各文字列領域を白塗りにして、いったんJPEG画像として保存し、convertコマンドを用いてPBM形式の2値画像に変換する(当初はNetpbmのコマンドを使用していたが、依存ソフトウェアを減らすため変更した)。白黒2値化の閾値は0.6としている。

プログラムではPBM形式のファイルを開いて読み出し、ピクセルごとの白黒2値情報の2次元配列を構成する。線分候補の取得においては、この2次元配列を行単位で走査し、20ピクセル以上連続する箇所について、始点および終点のXY座標をもとに、始点および終点のY座標が等しい線分オブジェクトとして構成する。そのようなオブジェクトの配列が、横方向の線分候補の集合となる。縦方向についても同様に処理し、始点および終点のX座標が等しい線分オブジェクトの配列を得る。

横方向の線分候補について、配列の要素(線分オブジェクト)間で距離を求め、12ピクセル(画像の幅の1/100)以内のものを結合する。結合により作られる線分の座標は次のとおりである。すなわち、左端X座標は結合対象の線分の中で最も左のX座標、右端X座標は結合対象の線分の中で最も右のX座標、Y座標は結合対象の線分の平均とする。これにより、始点および終点のY座標が等しい線分オブジェクトを新たに構成する。縦方向の線分についても、同様にして結合処理を行う。

結合処理を行った、縦方向・横方向の線分オブジェクト間で、交差の判定を行う(3.2節)。交差ごとに、縦方向の線分のインデックス(配列の出現位置)、横方向の線分のインデックス、交点XY座標からなる配列を構成し、交点の集合を配列として保持する。

ここまでの処理では、線分の誤認識が生じ得る。初期画像の濃さのため、2値化で黒となった箇所の誤認識(false-

positive)については、他の線分と交点を持たず、かつ人物名の文字列領域からの距離が離れているものを、線分候補から除外する。それでもまた残ってしまっている誤認識箇所、および、初期画像の薄さのため、2値化で白になり適切に認識できていない箇所(false-negative)については、プログラムで指示を与えて線分の削除・追加・結合・伸縮を行う(4.4節)。

隣接ページとまたがる横線の検出を行う。一方の交点がない横方向の線分について、その左端が処理領域の左端に近い(端から1/10以内の場合)場合には、次のページの横線と接続するものとみなす。右端が処理領域の右端に近い場合には、前のページの横線と接続するものとみなす。前後のページで、前ページの左端に届く線分、次ページの右端に届く線分が同数であることを確認したのち、Y座標の小さいものから順にそれぞれの線分を対応づけるよう、配列に格納する。

以上の処理について、処理結果の妥当性確認のため図1の画像上に、文字列領域、認識した線分(両端には強調のため×印)、交点の○印、および人物名と付随情報の対応付けを描いたものを図6に示す。

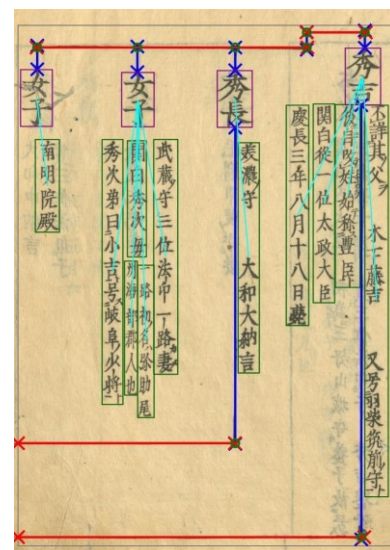


図6 認識結果例

Figure 6 Example of recognition result.

(4) 人物関係の取得

人物名の文字列領域オブジェクトから、領域の下段中央の座標を算出し、上端座標が最も近い、縦方向の線分オブジェクトを起点として、その線分とつながる線分オブジェクトの集合を、交差の配列および隣接ページとまたがる横線の対応付けをもとに探索し全て求める。起点となる縦線とつながる縦線(起点となる縦線自身を含む)のうち、下端座標と、他の人物名の上段中央の座標との距離が40ピクセル以下であれば、2人を親子関係と判断し、親子関係を保持する配列に格納する。

人物名の文字列領域オブジェクトから、領域の上段中央の座標を算出し、下端座標が最も近い、縦方向の線分オブジェクトを起点として同様に処理して、到達する人物については、2人を兄弟関係として配列に格納する。

「同一の親ならば兄弟」という推論を用いなかったのは、図1の「秀吉」の上に伸びる線分に見られるように、親となる文字列領域を持たない線分のつながりが存在するためである。

3.2 線分の交差判定

系図の人物同士の関係（親子関係，兄弟関係）を導出するための基礎として，系図画像上の線分どうしの「連結」を，「交差」に帰着して判定することとした．縦線および横線の「連結」の形状について，『豊臣秀吉譜上』ではL字型とT字型を見ることができ，また[2]などにおいては十字型の交差も見られるが，判定においては線分を少し伸ばした上で，十字型の交差判定を行う．このようにするのは，処理をL字型・T字型・十字型で分けることなく一括して行えるほか，現実には太さを持ち（前節で作成した初期画像では最大8ピクセルの太さが見られた），水平・垂直方向に真っ直ぐとは限らない，系図画像上の線分に対して，それまでの処理で得られた線分候補は，水平・垂直方向のいずれかに関して真っ直ぐであるとともに，太さを持っておらず，この点を考慮しない単純な判定では連結とならない場合が考えられるためである．連結と判定すべき，線分認識の例を，図7に示す．

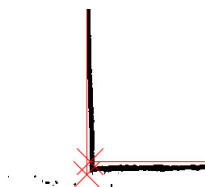


図7 連結すべき線分の認識例

Figure 7 Recognized lines that should be connected.

座標を用いて，判定方法を説明する．判定対象の縦線の上下端の座標を (x_1, y_{11}) および (x_1, y_{12}) ，横線の左右端の座標を (x_{21}, y_2) および (x_{22}, y_2) とする．ただし $y_{11} < y_{12}$ ， $x_{21} < x_{22}$ である．ここで2つの線分の各両端を，それぞれ定数 τ だけ伸ばす（前節で述べた適用例では $\tau = 10$ ピクセルである）．得られる縦線の上下端の座標を (x_1, y'_{11}) および (x_1, y'_{12}) ，横線の左右端の座標を (x'_{21}, y_2) および (x'_{22}, y_2) と表したとき， $y'_{11} = y_{11} - \tau$ ， $y'_{12} = y_{12} + \tau$ ， $x'_{21} = x_{21} - \tau$ ， $x'_{22} = x_{22} + \tau$ となる．そして， $x'_{21} \leq x_1 \leq x'_{22}$ および $y'_{11} \leq y_2 \leq y'_{12}$ の両方が成り立つとき，この縦線と横線は交差すると判断する．L字型で連結することになる状況の座標例を図8に示す．

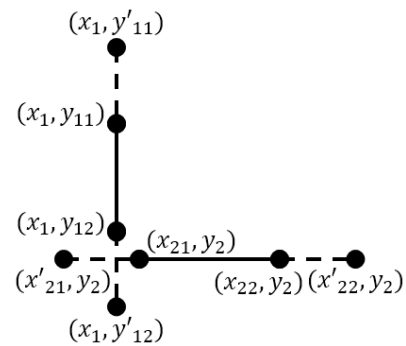


図8 交差判定の座標例

Figure 8 Example of coordinates for intersection.

4. Keizu Deconstructor

4.1 プログラム概要

開発・公開したプログラムについて概略を述べる．プログラミング言語は Ruby を用いて作成しており，処理部は本稿執筆時点で8ファイル，合計約2,400行からなる．

実行にあたっては，2.3節で述べた Ruby の処理系および ImageMacgick のコマンドに加えて，処理対象となる初期ファイル，および実行のための Ruby スクリプトファイルも必要となる．公開した一式の中に，3.1節で述べた処理の流れを記述したサンプルプログラムも収録している．実行する際には，あらかじめ2.1節で述べた，将軍家譜の ZIP 形式ファイルを，実行するディレクトリに置いておく必要がある．筆者の検証環境（OS: Ubuntu 16.04, CPU: Intel Core i7 @ 3.60GHz（マルチコアに配慮した処理はしていない），主記憶: 64GB）でこのサンプルプログラムの実行時間は約13秒であった．

将軍家譜[1]の一部，大正大学附属図書館所蔵『真言宗付法血脉』[2]，および東寺観智院金剛蔵本『真言付法血脉仁和寺』[13]についても処理を行い，各要素の認識・判別や，結果出力が行えているのを確認している．

4.2 実行方法

シェル（コマンドライン）と，git および docker のコマンドが利用可能な環境において，以下の手順でファイルを作ることができる．コマンドパラメータなどの詳細は[7]を参照されたい．

1. URL[7]をもとにプログラム一式をダウンロードする．
2. プログラム実行のための Docker イメージを作成（ビルド）する．
3. 将軍家譜の ZIP 形式ファイル（200021823.zip）をダウンロードし作業用ディレクトリに置く．
4. パラメータを与えて docker コマンドを実行することで，Docker コンテナ内で処理を行い，実行結果を zip ファイルとして作業用ディレクトリに作成する．

4.3 クラスの概略

プログラムで定義したクラスの概要を述べる。なお本稿に記載のクラス名や提供する機能は、コードの改良に伴い変更される可能性もある。

Rubyのクラスおよびモジュールの慣例に基づき、定義したすべてのクラスおよびモジュールは「Kzd::」から始まる。系図の1ページは複数の文字列領域を持つなど、クラス間のhas-a関係も多い(is-a関係は使用していない)。UMLのクラス図(属性および操作は省略)による、主要クラス間の相互関係を図9に示す。

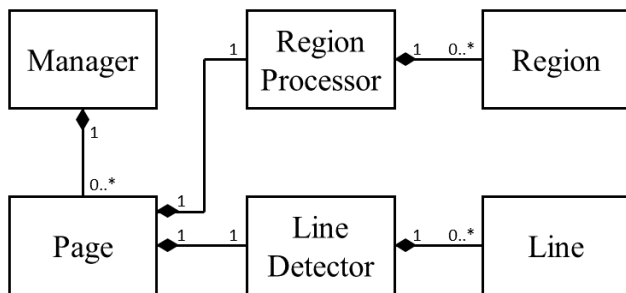


図9 クラス構成(主要部のみ、「Kzd::」は省略)

Figure 9 Classes defined in the program.

クラスごとに説明する。まずKzd::Managerクラスは、複数ページからなる1つの系図を管理する。Kzd::Pageクラスのオブジェクトの配列を持ち、ページ順に走査したり、前後のページを返したりするメソッドを提供する。

Kzd::Pageクラスは、1つのページを管理する。Kzd::RegionProcessorクラスのオブジェクトと、Kzd::LineDetectorクラスのオブジェクトを、1つずつ保持する。

Kzd::RegionProcessorクラスは、1つのページにある全ての文字列領域を管理する。Kzd::Regionクラスのオブジェクトの配列を持ち、文字列領域が特定の条件(文字列が正規表現にパターンマッチする、サイズが一定の範囲に収まるなど)を満たす場合に、「人物名」「付随情報」「どちらでもない」の種別を指定する処理は、このクラスのメソッドとして定義されている。3.1節で述べた、2文字以下の文字列を人物名とする処理は、Kzd::RegionProcessorクラスのオブジェクトを参照する変数rpに対しrp.set_person_by_regexp(/¥A..?¥Z/)と書けばよい。

Kzd::Regionクラスは、1つの文字列領域を管理する。領域座標および記載文字列を保持するほか、種別を設定したり読み出したりすることができる。

Kzd::LineDetectorクラスは、1つのページにある線分を管理する。縦線の集合と横線の集合は、異なる配列としてそれぞれインスタンス変数が保持する。

Kzd::Lineクラスは、1つの線分を管理する。始点と終点の座標を値として持つ。ただし縦線か横線かは、属性として保持するのではなく、このクラスで定義されたメソッド

の呼び出しのたびに、座標情報をもとに判定している。

その他に、初期ファイルをもとに中間処理や最終段階のファイル名を得るためのKzd::FilenameHelperモジュール、および線分や交点の修正処理を取り扱うKzd::LineModifierモジュールを別のファイルに定義し、Kzd::LineDetectorクラスに取り入れている。系図全体の結果出力を行うため、Kzd::Reporterモジュールを定義している。

4.4 認識結果の修正

系図画像に対する線分認識結果の修正の必要性およびその方法を述べる。1回限りの画像処理であれば、誤認識が見られた場合にはその都度、手作業で(テキストエディタなどを用いて)修正すれば十分である。しかし本研究のように、さまざまな素材に適用できることを目指した処理プログラムの開発・維持・改良をする場合には、同一の素材(本研究では『豊臣秀吉譜上])に対して繰り返し実行しながら、プログラム変更の前後での結果の違い(しばしば、変更前と結果が変わらないこと)を確認する必要がある。そのため、「その都度、手作業で」という修正方法をとるのは非効率であり、間違いを含む認識結果後に、プログラム内で処理を行うこととした。

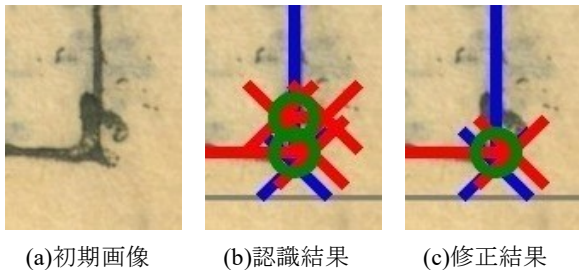
修正の指示は、ラベルと対象領域によって与えられる。定義したラベルを表1に示す。対象領域は処理画像における長方形領域であるが、その使われ方は、ラベルにより異なる。例えばdel_h、merge_h、del_v、merge_vに関しては、その対象領域に完全に含まれる横線と縦線を除去または結合する。add_hでは、対象領域の中心を通り、左右端は対象領域と同じとする線分を追加する。expandまたはshrinkから始まるラベルでは、座標を変更しない方の端点がある領域に入っている場合に、処理を行う。

表1 修正のラベル

Table 1 Labels for corrective action.

ラベル	処理
add_h	横線を1本追加する
del_h	横線を除去する
merge_h	横線を結合する
expand_left	横線の左端を伸ばす
expand_right	横線の右端を伸ばす
shrink_left	横線の左端を縮める
shrink_right	横線の右端を縮める
add_v	縦線を1本追加する
del_v	縦線を除去する
merge_v	縦線を結合する
expand_up	縦線の上端を伸ばす
expand_down	縦線の下端を伸ばす
shrink_up	縦線の上端を縮める
shrink_down	縦線の下端を縮める
no_cross	交点を除去する
nop	何もしない

『豊臣秀吉譜上』への適用にあたり、del_h, del_v, merge_v, expand_up を用いて計 12 箇所 of 修正を指示している。修正例を図 10 に示す。墨のつき具合により、短い横線を誤認識してしまっており、除去する処理を入れることで、横線 1 つと縦線 1 つが L 字型で連結するようになった。



(a)初期画像 (b)認識結果 (c)修正結果

図 10 修正による横線除去の例

Figure 10 Removal of horizontal line.

『真言宗付法血脈』の処理プログラムでは、表示上は交差しているが連結と見なさない箇所に対し、no_cross ラベルを用いた指示を取り入れている。ただし、線分修正を行ったのちに、改めて交差判定を行い、その後適用して除去している。

4.5 注意を要する線分への対応

図 3-5 (2.1 節) で例示した系図への対処について方針を示す。

図 3 に見られる、横並びの人物を結ぶ横線や、人物の下に伸びる縦線が人物名とつながっていない場合については、それぞれ、Kzd::RegionProcessor クラス内にインスタンス変数を配置して管理すればよい。図 4 のような、2 本の横線を結ぶ斜線については、前節で述べた修正指示を入れながら、折れ線に変更して連結することで解決を図ることができる(図 11)。人物名に施された斜線については、特別な処理をしておらず、縦横の走査により線分候補とならない。

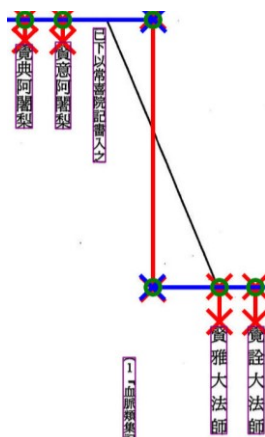


図 11 斜線の編集結果

Figure 11 Result of replacing diagonal line.

5. おわりに

本稿では、系図画像および位置情報付きテキストを対象として処理を行い、人物名と付随情報の判別、人物名どうしおよび人物名と付随情報を対応づけるプログラムについて報告した。

今後の課題として、他の系図への適用や、最適な閾値の算出などにより人手による処理の削減などが挙げられる。

謝辞 本研究を進めるにあたり、大阪大谷大学 宇都宮啓吾教授および凸版印刷 大澤留次郎氏よりコンテンツならびにご助言をいただきました。心より感謝申し上げます。また永井謙也氏は和歌山大学大学院在籍中、系図処理に関して先駆的な成果を挙げてくれました。御礼申し上げます。

本研究は JSPS 科研費 JP17H02342 の助成を受けたものです。

参考文献

- [1] 人文学オープンデータ共同利用センター準備室日本古典データセット 将軍家譜. <http://codh.rois.ac.jp/pmjtbk/200021823/> (2018-04-16 参照).
- [2] 苔米地誠一, 大正大学附属図書館所蔵『真言宗付法血脈』紹介と翻刻, 川勝守・賢亮博士古稀記念東方学論集, 汲古書院, pp.431-452 (2013).
- [3] 永井謙也, 村川猛彦, 大澤留次郎, 宇都宮啓吾: 系図からのデータ自動取得の試み, 人文科学とコンピュータシンポジウム論文集, 情報処理学会シンポジウムシリーズ, Vol.2017, No.2, pp.15-22 (2017).
- [4] 永井謙也: 系図データベースシステムの構築—系図からのデータ自動取得の試み—, 和歌山大学大学院システム工学研究科修士論文 (2018).
- [5] 杉山正治, 生田敦司, 柴田みゆき, 松浦亨: イベント指向データ管理手法を用いた系図表示の研究, 情報処理学会研究報告人文科学とコンピュータ, Vol.2011-CH-89, No.5, pp.1-8 (2011).
- [6] Sugiyama, S., Ikuta, A., Yokozawa, D., Hiratsuka, S., Shibata, M. and Matsuura, T.: Displaying Genealogy with Mythological Relations by Using the WHiteBasE Method, 人文科学とコンピュータシンポジウム論文集, 情報処理学会シンポジウムシリーズ, Vol.2017, No.2, pp.75-82 (2017).
- [7] Keizu Deconstructor (Kzd). <https://github.com/takehiko/kzd> (2018-04-16 参照).
- [8] 田中猛彦, 冨金原賢次, 宇都宮啓吾, 中川優: 平安・鎌倉を対象とした僧侶データベースシステム, 情報知識学会誌, Vol.13, No.2, pp.18-31 (2003).
- [9] 朴明哲, 森本雅史, 立花純児, 村川猛彦, 宇都宮啓吾, 中川優: 人文研究を支援するデータベースシステム—聖教検索および系図表示—, 情報知識学会誌, Vol.17, No.2, pp.105-110 (2007).
- [10] 築島裕, 醍醐寺蔵本「伝法灌頂師資相承血脈」解題, 醍醐寺文化財研究所研究紀要通号 1, pp.27-135 (1978).
- [11] ALTO Schema Version 3.1. <http://www.loc.gov/standards/alto/v3/alto.xsd> (2018-04-16 参照).
- [12] プログラム言語 Ruby, JIS X 3017:2013 (2013).
- [13] 武内孝善, 東寺観智院金剛蔵本『真言付法血脈仁和寺』, 高野山大学密教文化研究所紀要通号 6, pp.39-131 (1993).