領域分割済み PDF 文書からの 内部座標情報を用いた文章抽出に関する研究

重松杏美[1]　伊東桂佑[2]　鶴田直之[1]　乙武北斗[1]

概要：地方自治における住民参画を促進する取り組みとして，地方自治体の会議資料をテキスト化し，自然言語処理が可能なデータベースを構築する試みが進められている．本研究では，片田江ら(2026)の手法によって抽出された文書の領域分割情報を手掛かりとして PDF 文書から文字単位の座標情報を用いて意味的に正しい順序で文章を抽出することを試みた．具体的には，PyMuPDF を用いて取得した文字単位の座標情報を統計的に解析・再構成することで，PyMuPDF で生じやすかった文字位置のずれや読順の誤認識を抑制することができた．実験では，正しく領域分割されている箇所については 98%の精度で文章を抽出することができた．

キーワード：PDF 文書，文章抽出，内部座標情報，領域分割情報

How to Typeset Your SIG Technical Reports in MS-Word (Version 3.5)

AMI SHIGEMATSU[†1], KEISUKE ITO[†2], NAOYUKI TSURUTA[†1], HOKUTO OTOTAKE[†1]

Abstract: As an initiative to promote resident participation in local governance, efforts are underway to convert local government meeting materials into text and build a database suitable for natural language processing. This paper attempts to extract sentences from PDF documents in a semantically correct order using character-level coordinate information, guided by the document segmentation information extracted by Katada et al. (2026). Specifically, by statistically analyzing and reconstructing the character-level coordinate information obtained using PyMuPDF, we were able to suppress character position shifts and misrecognitions in reading order that frequently occurred with PyMuPDF. In experiments, we achieved 98% accuracy in extracting sentences from correctly segmented regions.

Keywords: PDF documents, text extraction, internal coordinate information, segmentation information

1. はじめに

近年，地方分権の進展により，住民が行政の意思決定を理解する上で，自治体が公開する PDF 形式[1]の議会資料の重要性が増している．そこで，地方自治体が公開する会議資料を自然言語処理によって構造化し，データベース化する研究が進められている[2][3]. しかし, PDF は印刷物の再現を目的とした形式であり，内部データが論理順ではなく描画命令として記録されるため，読み順の崩れや文字位置の誤認が発生しやすい．特に自治体資料では，縦書き・横書きの混在，複雑な表レイアウトの存在がテキスト抽出をさらに困難にしている．

生成 AI や大規模言語モデルを用いた高度な自然言語処理が注目される中，これらの PDF を正確にデータ化するためには，意味順序に基づく構造的なテキスト抽出が不可欠である．しかし，従来の画像認識ベースの領域検出や一般的な PDF テキスト抽出ツールでは，レイアウト情報が十分に扱えず，抽出テキストが断片化する問題がある．例えば，Python で利用可能な PDF 解析ライブラリ PyMuPDF[4]は, 段組みなどのレイアウト情報が十分に扱えず，図 1 の右側のような例では正しい順序で文字が抽出できない．一方，文字座標を用いた構造推定手法も存在するが，画像ベースのレイアウト解析との統合や，一貫した構造化テキスト生成までを扱う枠組みは十分に確立されていない．例えば，GROBID[5]による学術文書構造解析は，単純な文字列抽出に比べて文書構造をより正確に復元できる可能性を示しているが，画像ベースのレイアウト解析結果と統合的に扱う枠組みは十分に確立されていない.

本研究では，PDF 文書から「意味的に正しい順序」で情報を抽出するため，先行研究[6]の画像認識ベースの手法（以下，領域検出）による領域分割結果（図 1 の左側）と PyMuPDF による PDF 内部の詳細な文字情報取得結果とを統合する. これにより, 座標変換と文字座標の統計的解析，領域内文字の厳密な確定が可能になり，画像解析のみでは難しかった境界領域の判定や，縦横書き混在文書における正確な読み順の復元を行うことができる.

1 福岡大学
Fukuoka University
2 福岡大学大学院
Fukuoka University Graduate School

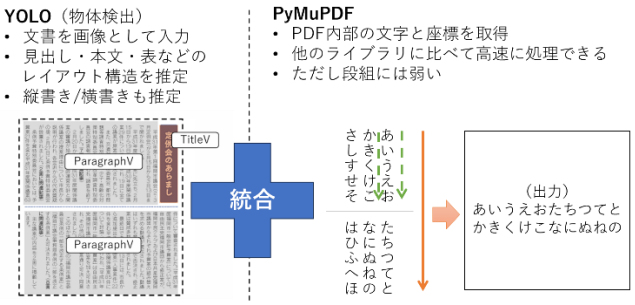

図 1 提案手法の基本概念

Figure 1 A Basic Concept of the Proposed Methodology

2. 提案手法

本研究の提案手法は，物体検出モデルによる「視覚的な領域把握」と，PDF 内部に保持された文字座標情報に基づく「精密な文字確定」を組み合わせたハイブリッドアプローチである．本手法の全体フローは，大きく 4 段階で構成される（図 2 参照）．なお，以下では，図 2 中の②〜④を構造化アルゴリズムと呼ぶことにする．

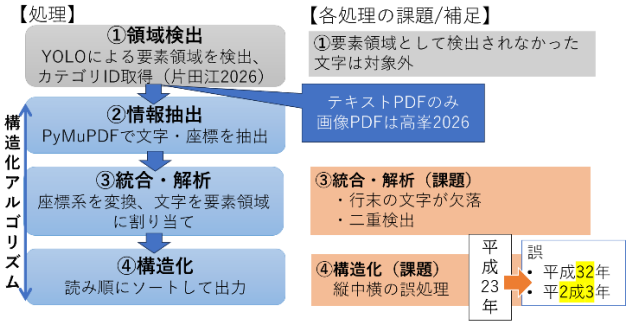

図 2 提案手法の処理フロー

Figure 2 Process Flow of the Proposed Method

2.1 領域検出

後続処理のガイド情報として，領域分割結果（カテゴリ ID およびポリゴン座標を含む JSON ファイル）を受け取る．ここで，カテゴリは以下のように階層的に定義されている．なお，H (Horizontal)と V (Vertical) はそれぞれ横書き，縦書きを示す．

Page（ページ）

- PTitle（ページ内に一つだけ存在する大見出し）
- PSegment（タイトル，文章，リード文を持つかたまり）
  - TitleV（PSegment 内に 1 つだけのタイトル，見出し文）
  - TitleH
  - LeadV（Title,Paragraph を補足する文，リード文）
  - LeadH
  - ParagraphV（文章，本文，箇条書きの文）
  - ParagraphH
- FSegment（図や表とその説明文のかたまり）
  - Figure（イラスト，挿絵，写真）
  - Table（表，罫線で囲まれた文字や数字 ※箇条書きは含まない）
  - CaptionV（図や表を簡潔に説明する 1〜2 行の文）
  - CaptionH

本研究では，FSegment 内の Figure 以外の全ての文字領域を対象とする．また，ページ全体が画像として保存されている文書を除き，文字情報が保存されている PDF ファイルを対象とする．

2.2 情報抽出

解析ライブラリ PyMuPDF を用い，PDF 内部から全文字のテキスト情報と詳細な座標情報を取得する．この段階では，PDF ファイル構造上の記録順序に従った生データを抽出する．

具体的には，入力された PDF がテキスト情報を保持している「テキスト PDF」であるか，あるいはスキャン画像のみで構成される「画像 PDF」であるかを判定する．テキスト PDF である場合，PyMuPDF を用いてページ内の全文字のバウンディングボックス(x_0, y_0, x_1, y_1)を抽出する．

次に，画像認識上のピクセル座標系（JSON 側）と PDF 内部のポイント座標系を同期させるため，以下の式によりスケーリング係数（scale_x,scale_y）を算出する．

$$scale_x = \frac{PDF_{Width}}{JSON_{Width}} \ , scale_y = \frac{PDF_{Height}}{JSON_{Height}}$$

この係数を用いて，領域検出により検出した矩形やセグメンテーション（多角形）を PDF 上の座標へ正確に投影し，以降の処理の基盤とする．

2.3 統合・解析

画像上の検出領域（ピクセル座標）と PDF 内部の文字情報（ポイント座標）を，PDF ページサイズと画像解像度の対応関係から算出した線形スケーリング係数に基づいて同一座標系へ変換する．各検出領域に対して「最大重なり面積」や「重心点判定」など複数の判定基準を組み合わせることで，境界線上の文字を正しい要素へ厳密に割り振る．また，統計的解析により，ページ内の主要な文字サイズや行間を特定し，抽出の閾値を動的に決定する．

(1) 文字情報の統計的解析

PDF 文書は作成環境によってフォントサイズや文字間隔が大きく異なるため，一律の閾値では正確な行・列の判定が困難である．そこで本研究では，ページ内の文字情報を統計的に解析し，処理に必要なパラメータを動的に決定する手法を採る．

ページ内に含まれるすべての文字情報から平均文字幅（avg_char_width）および平均文字高さ（avg_char_height）を算出する．この平均値に基づき，文字同士が同一の行またはカラムに属しているかを判定するための許容誤差を，例えば avg_char_width×0.8 のように算出する．この動的な

パラメータ調整により，文字密度の異なる多様な自治体資料に対しても，一貫した精度で文字の結合判定を行うことが可能となる．

すなわち，平均文字サイズに基づく動的パラメータ決定と，局所的な文字配置構造に基づく並び替え処理を組み合わせることで，文書全体にわたる文字サイズのばらつきに対しても頑健な構造復元を実現している．

(2) 座標変換と最適領域への帰属判定

画像認識によって得られた領域情報（ピクセル座標）とPDF 内部の文字情報（ポイント座標）を同期させ，各文字がいずれの要素に属するかを一意に決定する．

まず，算出されたスケーリング係数を用いて，検出領域を PDF 上の座標系へ投影する．次に，変換された領域（ポリゴン）と各文字のバウンディングボックスの重なり具合を検証する．本手法では，ある文字が複数の検出領域に含まれる可能性がある場合に備え，「最大重なり面積」による判定ロジックを導入する．

具体的には，文字の矩形領域と各ポリゴン領域との交差面積を算出し，最も大きな面積を記録した領域をその文字の所属先（最適領域）として確定させる．ここで，データの整合性を保ち，テキストの重複抽出を物理的に排除するため，「1 文字につき帰属可能な領域は最大で 1 つのみ」という排他的な帰属規約を課している（図 3）．

面積による判定が困難な微細な文字については，重心点の包含判定を併用することで救済を行う．この厳密な紐づけプロセスにより，境界線上の文字の重複抽出や欠落を防ぎ，視覚的レイアウトと電子データの正確な統合を実現する．

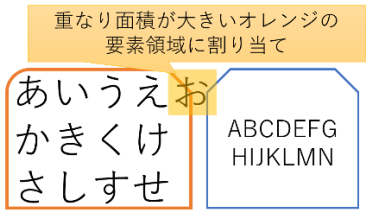

図 3 排他的な帰属規約の概念図

Figure 3 Conceptual Diagram of Exclusive Attribution Rules

2.4 構造化

抽出・分類された文字集合をカテゴリ（縦書き・横書きなど）に応じた適切な順序（読み順）に並べ替え，生成 AI を活用する際の前処理として機能する構造化テキストを出力する．

前節まで確定した領域ごとの文字集合を，意味的に正しい順序に従って再構成し，機械処理に適した構造化テキストとして出力する．PDF 内部の文字情報は記録順序が不規則であるため，文字の空間配置と書字方向に基づいた読み順推定が不可欠である．

本研究では，横書き・縦書き別に応じた基本的なソート規則を適用したうえで，縦書き文書中に現れる数値列に対して独自の補正処理を導入する．

(3) 基本的な読み順ソート規則

まず，各検出領域（ポリゴン）ごとに割り振られた文字集合に対して，書字方向に応じた基本的な読み順ソートを行う．

図 4 に示すように，横書き領域においては，日本語文章の一般的な読み順に従い，文字を上から下（Y 座標昇順），同一行内では左から右（X 座標昇順）の順で整列させる．行判定には，前節で算出した平均文字高さおよび許容誤差を行い，Y 座標が一定範囲内にある文字を同一行として扱う．

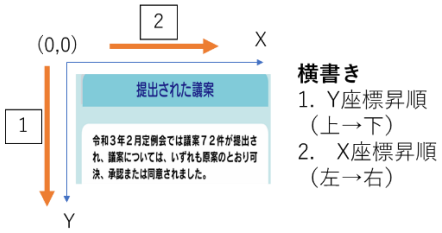

図 4 横書きのソート規則

Figure 4 Sorting Rules for Horizontal Writing

一方，図 5 に示すように，縦書き領域においては，右から左のカラム順（X 座標降順）を優先し，同一カラム内では上から下（Y 座標昇順）の順で文字を整列させる．カラム判定についても，平均文字幅に基づく動的な閾値を用いることで，文字間隔のばらつきに対応する．

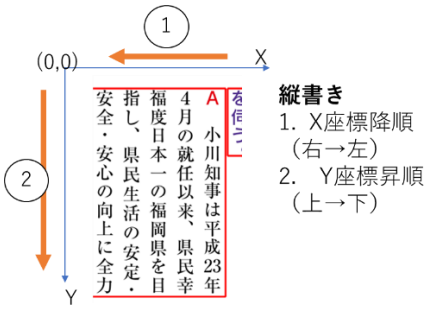

図 5 縦書きのソート規則

Figure 5 Sorting Rules for Vertical Writing

これらの処理により，通常の文章部分については，人間が自然に読む順序に近い文字列を安定して復元することが可能となる．

(4) 縦書き文書中の数値列（縦中横）に対する独自補正アルゴリズム

縦書き文書では，本来 1 文字ずつ縦に並ぶべき文字の中で，数値や記号を例外的に横方向に配置する「縦中横」の表記が頻出する．これらは視覚的には横方向にまとまって配置されるが，前節の基本ソートのみを適用すると，本来一つの数値列として読まれるべき文字が分断されたり，順序が入れ替わったりする問題が生じる．

そこで本研究では，縦書き領域内の文字集合に対して，数値に特化した読み順補正アルゴリズムを導入する．本ア

ルゴリズムでは，以下の条件に基づき「数値列候補」を判定・抽出する．

- 文字種別判定：対象文字が数字（0-9）または数値に付随する記号であること
- 近傍判定：同一カラム内，直前の文字との Y 座標差が定義された閾値以内であること
- 連続性判定：文字間の X 座標間隔が平均文字幅以下であり，水平方向に連続性が認められること

抽出された数値列候補については，縦書きの基本規則を一時的に無効化し，X 座標昇順に再ソートすることで，人間が視覚的に認識する横方向の並び順を優先する．これにより，縦書き中に挿入された数値列（縦中横）をまとまりの情報として正しく復元することが可能となる．

この補正処理は，通常の文字列に対しては影響を与えず，数値列に限定して適用されるため，文書全体の読み順整合性を保ったまま，情報の意味的破綻を防ぐことができる．

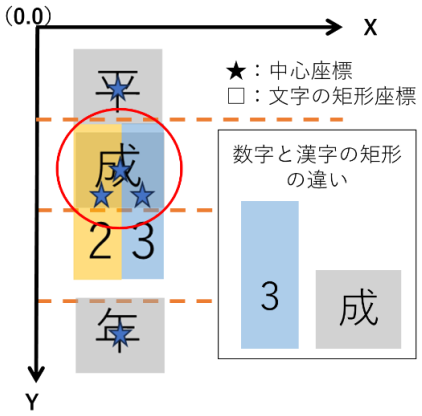

図 6 縦書き文書中の数値列に対する補正

Figure 6 Correction for numerical sequences in vertical-writing documents

2.5 出力形式

最終的に再構成されたテキストは，領域種別（タイトル，本文，図表等）に応じたタグ情報を付与し，Markdown 形式で出力する．これにより，生成 AI（LLM）や検索・要約システムなどへの入力として，そのまま利用可能な高品質な構造化テキストデータを実現する．

3. 実験

3.1 実験の概要

本実験の目的は，提案手法である「物体検出結果と PDF 内部座標情報の統合」が，複雑なレイアウトを持つ地方自治体資料のテキスト抽出においてどの程度の精度を持つかを検証することである．なお，前処理の性能は適合率 84.33%，再現率 77.35%，F 値 80.54%である．本実験での性能評価では，前処理が正しく行われている領域のみを対象とした．

提案手法の性能評価に際しては，以下の 3 点について評価を行う．

- 画像認識による領域検出と PDF 内部文字情報の対応付けの正確性
- 縦書き・横書きが混在する環境下での読み順（ソート規則）の妥当性
- 独自補正アルゴリズムによる縦書き中の数値情報の復元精度

評価対象として，実際に地方自治体で公開されている PDF 形式の議会報（例:『ちくじょう議会だより』等）から，縦書き・横書き・数値・図表が混在するページを任意に抽出したデータセットを使用する．実験の手順は以下の通りである．

3.2 評価指標および評価方法

本研究では，テキスト抽出及び読み順ソートの精度を定量的に評価するため，正解データとの一致率を表す Accuracy を評価指標として用いる．評価は，提案手法全体の性能評価に加え，領域検出の影響を除外した場合の構造化アルゴリズム単体の性能についても行う．

(1) 評価パラメータ

正解データと抽出結果を比較し，以下の 5 つのパラメータを算出する．

- N（Total Characters）：正解データにおける総文字数
- S（Substitution）：置換誤り（文字は存在するが，誤った文字として抽出された数）
- D（Deletion）：欠落誤り（正解データにあるが，抽出されなかった文字数）
- I（Insertion）：挿入誤り（正解データにないが，余分に抽出された文字数）
- T（Transposition）：文字の入れ替え（文字自体は正しく抽出されているが，読み順ソートの誤りにより位置が不適切な文字数）

本研究における Transposition（T）は，抽出された文字が正解データと同一であるにもかかわらず，読み順ソートの誤りによって配置順が一致しない文字数を表す．例えば，同一行内の電話番号などにおいて，一部の数字のみ順序が入れ替わった場合には，当該数値文字列全体を 1 件として扱うのではなく，正解位置と一致しない各文字を独立に T として計数した．

(2) 精度算出式

上記のパラメータを用い，正解データに対する一致率を以下の式で定義する．この式では，単なる文字の有無ではなく，本研究の主眼である「構造化（読み順）」の妥当性を厳密に評価するため，入れ替え（T）を独立した誤りとして減点対象としている．

$$Accuracy = \left(1 - \frac{S + D + I + T}{N}\right) \times 100 \quad [\%]$$

(3) 評価の妥当性

一般的な文字認識の評価では S,D,I のみが用いられることが多いが，本研究のように PDF の内部構造から「正しい

順序」を復元する手法においては，順序の誤りを示す T の評価が不可欠である．この指標を用いることで，独自補正アルゴリズム（4.5 節）が読み順の適正化にどの程度寄与したかを定量的に示すことが可能となる．

また，本研究では，領域検出結果に起因する誤りと，その後の読み順補正アルゴリズムに起因する誤りを区別するため，評価条件を分けて実験を行った．これにより，提案手法全体の性能と，独自に設計した構造化アルゴリズムの性能をそれぞれ定量的に評価することを可能としている．

3.3 実験結果

(1) 提案手法全体の性能評価

提案手法全体の評価結果を表 1 に示す．Accuracy は約 98.05％であった．

表 1 提案手法全体の評価結果

Table1 Evaluation Results of the Proposed Method

項目	個数
全ての正解データの文字数(N 合計)	46255
S 合計（置換）	31
D 合計（欠落）	351
I 合計（挿入/二重検出）	120
T 合計（隣接文字の入れ替え）	398

誤りの内訳を分析した結果, カテゴリによってエラーの傾向が明確に分かれることが判明した．

- 置換誤り（S）と入れ替え（T）：これらは主に「表（Table）」カテゴリで発生した. 特殊文字（例:「吉」）が連続して誤変換されたり，記号（▲）の向きが反転して抽出される例が確認された．また，表全体を一つのポリゴンで囲んでいるため，内部の複雑な列構造を単一な行としてソートしてしまい，大きな入れ替え（T）が発生した．
- Paragraph カテゴリの精度：特筆すべきは，Paragraph カテゴリにおいては S（置換）および T（入れ替え）が一切発生しなかった点である．本文領域における文字の正しさと読み順の復元については，極めて堅牢な性能を示した．
- 二重ポリゴンによる文章の分断（D および I）：領域検出において, ある領域を覆う小さなポリゴンと，それを包含するように重複して検出された大きなポリゴンが存在する場合，境界付近の文字が「奪い合い」の状態となる．本手法の帰属判定ロジックにより，末尾の文字がより面積の大きい（重なりに余裕がある）広域ポリゴン側へ割り振られ，本来の領域側ではその文字が欠落（D）し，広域側では不要な文字が混入（I）する現象が多発した．
- ライブラリの座標変換誤差による欠落（D）：文章の行末に位置する句読点が，一貫して欠落する傾向がみられた. これは領域検出の精度ではなく, PDF 内部座標（ポイント）を画像上のピクセル座標へ変換するライブラリ（PyMuPDF）の処理において，微細な座標の不一致が生じていることに起因すると考える．

(2) 構造化アルゴリズムの性能評価

本節では，提案手法の中核である Python アルゴリズムの性能を明確にするため，領域検出由来の「ポリゴンの二重検出」及び構造的解析が困難な「表カテゴリ」を除外した条件下での評価を行った．この条件下での Accuracy は 98.89%に達した．この結果から，境界領域の句読点や記号の欠落という微細な課題を除けば，本研究のアルゴリズムは論理的な文章構造の復元において高い有効性が確認された．

(3) 考察

実験結果から，領域検出の品質がテキスト抽出精度を左右する最大の要因であることが明らかになった．デバックの結果，多くの欠落（D）と，I（挿入）は，人間が視覚的に認識する領域と，領域検出が出力したポリゴンのわずかなズレによって引き起こされていた. 特に「1 文字 1 領域」という厳密な帰属ルールを採用しているため，ポリゴンが二重に重なっている箇所では，面積のわずかな差によって文字が隣接領域に「奪われる」事態が生じる．これを防ぐためには，領域検出の検出結果に対して，ポリゴン同士の重なりを解消するクリッピング処理や，境界付近の文字をどちらの領域に入れるか判断するヒューリスティックなルールを設けるなどの検討が必要である．

Paragraph 領域において発生した句読点の欠落については，前述したポリゴン検出のズレに加え，PDF 内部の文字座標取得に起因する問題が重なっていると考えられる．デバック用に文字バウンディングボックス及び領域ポリゴンを PDF 上に可視化した結果, 視覚的には当該文字はポリゴン内部に収まっているように見えるにもかかわらず，PyMuPDF が取得した文字座標ではポリゴン外と判定されている確認された．さらに，ペイントツールを用いて確認した視覚的な文字中心座標と, PyMuPDF から取得した文字中心座標を比較したところ，縦書き文書においてのみ，PyMuPDF が返す座標が下方向に数ポイントずれていることが判明した．一方で，横書き文書および領域ポリゴンの描画においては同様のズレは確認されなかった．これらの観測結果から，本現象は単純な描画処理やスケーリング誤差によるものではなく，縦書きテキストに対する描画命令レベルの座標解釈，すなわち回転行列やテキストマトリクスの適用方法に起因する可能性があると考えられる．PDF 内部では，縦書き文字が回転を伴う描画命令として記述される場合があり，これらの情報を度の座標系で文字バウンディングボックスとして取得するかによって，視覚的位置との差異が生じる可能性がある．本研究では，PDF 描画命令の完全な再解釈までは行っていないため，この座標ズレ

を完全に解消するには至らなかった．本研究の重要な知見である．今後は，描画命令レベルでの座標計算や回転行列を厳密に考慮した文字配置解析手法の導入が必要であると考えられる．

一方で，特筆すべき成果は，一般的な文章（paragraph）において文字の置換（S）および入れ替え（T）が一切発生しなかったことである．特に，「平成 23 年」が「平 2 成 3 年」と入れ替わらずに抽出できたことは，縦書きと数値が混在する自治体資料解析において，本アルゴリズムが解決策になることを示している．

4. おわりに

本研究では，PDF 文書中の文書要素を意味的に正しい順序で構造化することを目的とし，そのための補助情報として，先行研究である画像領域検出手法により抽出された領域情報を活用した．特に，縦書き文書や数値を含む文章においては，単純なテキスト抽出では文字位置や読み順の誤認識が生じやすい．そこで，領域検出によって得られた領域情報を前提とし，PDF 内部に記録されている文字座標情報に着目した解析手法を提案した．提案手法では，文字単位での座標変換および配置情報の再構成をおこなうことで文書中の文字配置を忠実に再現する．

その結果，縦書き・横書きが混在する PDF 文書に対しても，領域ごとの文字位置のずれや順序誤りを抑制した文書要素抽出が可能となり，抽出精度の向上を確認した．
一方で，PDF の内部座標情報による文字配置の再現は完全ではなく，描画命令に含まれる回転行列や座標変換処理を十分に再現できないことから，一部領域において文字抽出が困難な事例が確認された．今後は，描画命令レベルでの座標計算手法の検討が必要である．

謝辞 本研究は JSPS 科研費 JP22K12740 の助成を受けたものです．

参考文献

[1] PDF32000.book（オンライン）入手先 https://developer.adobe.com/document-services/docs/assets/35e4369068f86065372c18787171a17e/PDF_ISO_32000-1.pdf?utm_source=chatgpt.com （参照 2026-01-28）.
[2] Yasutomo Kimura, Keiichi Takamaru, Takuma Tanaka, Akio Kobayashi, Hiroki Sakaji, Yuzu, Uchida, Hokuto Ototake and Shigeru Masuyama: Creating Japanese Political Corpus from Local Assembly Minutes of 47 Prefectures, ALR12, The COLING 2016 Organizing Committee, pp.78-85, 2016.
[3] 乙武北斗，内田ゆず，高丸圭一，木村泰知: 構造化データ作成を 目的とした PDF 地方議会資料のテキスト抽出に関する分析， 第 37 回ファジィシステムシンポジウム講演論文集，pp.431- 436，2021.
[4] PyMuPDF ドキュメント（オンライン）入手先 https://pymupdf.readthedocs.io/ja/latest/index.html （参照 2026-01-28）.
[5] GROBID Documentation（オンライン）入手先 https://grobid.readthedocs.io/en/latest/Introduction/ （参照 2026-01-28）.
[6] 片田江啄門，PDF 文書の階層構造と YOLO を用いた領域分割の性能向上に関する研究，福岡大学電子情報工学科卒業論文，2026.1.