

マルチメディア文書入力のための文書画像認識システム：DRS

天野富夫 山下晶夫 伊東伸泰 小林芳直 豊川和治

日本アイ・ピー・エム(株)東京基礎研究所

既存の印刷文書からマルチメディアデータへの変換を支援するシステムDRS (Document Recognition System)の研究開発の一環として、パーソナルコンピュータ上に実験システムを作成した。このシステムは文書画像から文字列部分を抽出してタイトルや著者名などのラベルを与えるレイアウト解析機能、漢字OCRアダプターカードによる文字認識機能、及び日本語辞書と形態素レベルでの制約を利用した文脈後処理機能をサポートしている。本報告では実験システムの持つ諸機能について概説し、また各機能モジュールを組み合わせるための枠組みについて考察する。

Document Recognition System for Multimedia Documents Entry

Tomio AMANO Akio YAMASHITA Nobuyasu ITOH

Yoshinao KOBAYASHI Kazuharu TOYOKAWA

Tokyo Research Laboratory, IBM Japan, Ltd.

This paper describes an experimental system of DRS(Document Recognition System) which has been implemented on a personal computer. The DRS aims to support a conversion of existing printed documents into multimedia data. The experimental DRS can locate character strings, analyze layout structure, recognize characters, and automatically detect recognition errors by contextual post-processing. A flexible framework for cooperations of these function modules is also studied in this paper.

1 はじめに

光ディスクなどのハードウェア技術の発達によってオフィス等の文書をコンピュータを用いて蓄積検索するシステムが利用可能になってきた。現在製品化されているものの主流は文書の各ページを画像データとして光ディスク等に格納するシステムであるが、将来的には文書をマルチメディアデータとして管理するデータベースが普及すると予想される。マルチメディア文書データベースではテキスト情報と図や写真などの画像情報を統合して蓄積し、文書のレイアウト／論理構造も合わせて記録しておくことが必要とされる。これらの情報を用いれば、キーワードによる直接検索、ブラウジング検索、意味検索などの多種多様な検索手法が利用可能になる。しかし、さまざまな情報を蓄積し活用するということはその情報の入力に要する手間も増大することを意味している。このことは特に既存の印刷文書をデータベースに入力しなければならないという局面で問題になる。テキスト領域やレイアウトの構成要素を指定するのはユーザにとって手間のかかる作業であるし、フルテキスト検索等をサポートする場合のテキスト入力にかかる手間は膨大なものとなる。テキスト入力を書誌の情報（タイトルや著者名）に限った場合でも、依然としてこの部分の入力がボトルネックになると考えられる(1)。

上述のような問題点を解決するためレイアウト解析や文字認識をコンピュータによって行ない、既存印刷文書のマルチメディア文書への変換を支援しようという試みがいくつかなされている(2,3)。筆者らは今回画像の取り込みからレイアウト解析、認識、文脈後処理、ユーザインターフェイスまでのトータルな機能を備えたシステムDRS(Document Recognition System)のプロトタイプングを行ったので報告する。DRSは印刷文書からマルチメディア文書への変換に必要なさまざまな機能のうち、需要が大きいと思われる科学

技術文献の入力のためのいくつかの機能をサポートしている。このプロトタイプングの目的は1)各処理モジュールを実際に近い条件下でテストできる環境を提供し、2)入力対象文書やアプリケーションの要求に応じて処理モジュールが組み合わせられ協調的に動作できるような枠組みを確立しておくことである。

以下、2章ではDRSのシステム構成を簡単に述べ、3章では現在DRSのサポートしている機能について説明する。4章ではインプリメンテーションにおいて各機能が協調的に動作するための枠組みについて説明した後、実際にDRSで文書画像を処理した例を示す。5章では試作システムの拡張性について考察し、このような拡張性が実用に向けての重要な要件であることを述べる。

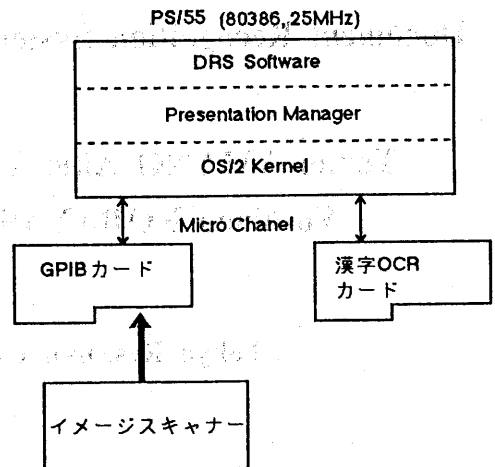


図 1: DRSのシステム構成

2 システム構成

図1にDRSのシステム構成を示す。文字認識以外のすべての処理はパーソナルコンピュータPS/55(80386,25MHz)上で行われており、OS/2上で動作している。文書の各ページはイメージスキャナーによって400dpiの解像度でスキャンされ、GPIBインターフェースを介してPS/55に二値化された画像データが取り込まれる。このデータはそのままPS/55のバスに接続された漢字OCR(Optical Character Reader)アダプターカードに送られる。漢字OCRカードはマイクロプロセッサ68020(16.5MHz)、画像データと認識用のテンプレートを格納するためのメモリー、特徴抽出と識別を行うための専用ハードウェアで構成されている。このOCRカードは手書き漢字用に開発されたものであるがDRSでは68020のプログラムを変更して印刷漢字用に用いている。68020のプログラムと認識用テンプレートはDRS立ち上り時にPS/55からロードされる。

3 各処理モジュールの概要

3.1 レイアウト解析モジュール

レイアウト解析モジュールは、文書画像から文字列や線分を表す矩形の位置情報(一次スケッチ)を作成した後、文書のレイアウトモデルと一次スケッチの照合を行い、文字列に対して論文タイトル、著者名などのラベルを付与する。

1) 一次スケッチの作成

図2に一次スケッチ作成の原理を示す。取り込まれた画像データはラインごとにランレンクス表現に変換され、黒ランの間の白ランの長さがある閾値以下の場合には左右の黒ランを接続して一本の黒

ランにするという処理(Smearing)が行われる。Smearing処理後、連続する2ライン分のランレンクス表現を比較することにより黒画素領域の上辺(Top Boundary)と底辺(Bottom Boundary)に相当する線分データを得る。レイアウト解析モジュールは画像をラスタ走査しながら検出された Top Boundaryを記録してゆく。Bottom Boundaryが検出されたら、対応する(重なりがあがる)Top BoundaryとBottom Boundaryとの間に黒画素の領域が存在すると判断し矩形情報を生成する。既に生成された矩形と接している場合はこれらを統合する。また、矩形の高さをもとにして文字列、水平線分、その他などの解釈を与える。

2) レイアウトモデルと一次スケッチの照合

文書画像のレイアウトモデルは部分-全体関係に基づく木構造によって表現されている。木の節点や葉はタイトルや著者名などの文書画像の構成要素を表しており、これらの上下関係によって下位要素が水平または垂直方向に並んで上位の要素を構成している状態が示されている。各節点にはその文書画像構成要素が満たすべき条件が記録されている。条件としては例えば、「その構成要素は

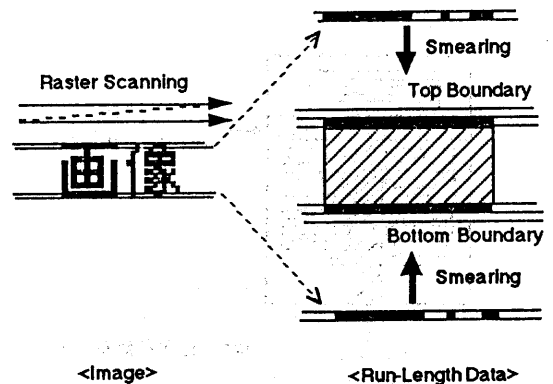


図 2: 一次スケッチの作成

必須か」、「上下左右にフィールドセパレータと呼ばれる白画素の領域があるか」、「最低／最大何個の下位要素で構成されるか」などがある。照合の最初の段階では、文字列矩形の集合を分割するフィールドセパレータが抽出される。レイアウト解析モジュールは一次スケッチの矩形情報（位置と大きさ）から白画素領域を表す矩形をもとめ、大きさがフィールドセパレータとしての条件を満たすものがあれば、これを用いて矩形集合を分割してゆく。分割された部分集合内で文字サイズやピッチが変わっている場合はさらに分割を行う。図3にフィールドセパレータによる分割の例を示す。次に分割された文字列部分集合群とモデルの木構造のすり合せが行われる。解析モジュールはモデルで定義されている位置関係に従って可能なラベルを文字列矩形に割り当ててゆき、弛緩法によって上述のような条件を満たすラベル付けを決定する。ラベル付けが一意に定まらない場合は、文字サイズやピッチの変化とラベルの変化がどの程度一致しているかを測る評価関数によって最適なラベル付けを決定している。

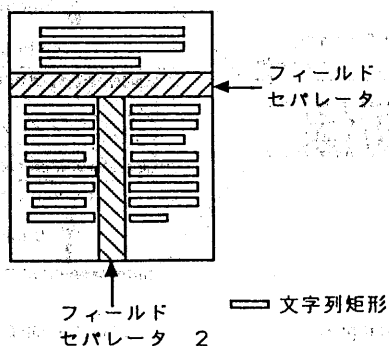


図 3: フィールドセパレータによる文字列集合の分割

3.2 文字認識モジュール

PS/55側の認識モジュールは認識すべき文字行の位置を漢字OCRアダプターカードに渡して認識要求を発行するのみで、実際の認識はOCRアダプターカードによって行われる。OCRアダプターカードはPS/55より文字行の位置情報を受け取り、カラムの切り出し及び認識(4)を行った後、各カラムに対して最大5位までの候補文字と識別時におけるテンプレートと各候補との距離を返す。カラムの切り出しにおいては、半角/全角文字の混合に対処するため認識結果のフィールドバックを行っている。

3.3 文脈後処理モジュール

印刷漢字の認識率は手書き漢字に比べて格段に高いとはいえ、実用環境下ではしばしば認識率の低下が問題となる(5)。画像の品質の低下や「一」（漢字の1）と「一」（長音）などの同形文字の存在などが原因と考えられるが、こういった問題にたいして単語あるいは文節単位での後処理によって認識率を向上させる試みが行われている(6,7)。DRSでは日本語辞書（約10万語）と形態素レベルでの制約を知識として利用した文脈後処理を行っている。文脈後処理モジュールは認識の結果得られた最大5個の候補文字を識別時の距離により平均2.3個に絞った後、単語辞書とのマッチング及びコスト関数に基づいて最適な単語列の選択を行う。後処理結果と認識結果を比較して、認識誤りの可能性がある文字については下位候補との入れ替えやオペレータへの警告が行われる。

また、各単語には辞書引きを行った時点で品詞コード（現在106に分類）が付与されるので選択された単語列から出現頻度の高い自立語（複合語を含む）を検出しキーワード候補として記録しておくことができる。

3.4 ユーザインタフェース

DRSのユーザインタフェースはOS/2のプレゼンテーションマネージャ上に構築されており、大部分の操作をマウスで行うようになっている。文脈後処理後の認識結果はテキストウィンドウに表示され、マウスで文字をクリックすることにより対応する文字イメージや2位以下の候補文字を表示することができる。マウスで2位以下の候補を選択して誤りを修正したり、キーボードからカナ漢字変換で正しい文字を入力することもできる。

4 インプリメンテーションと処理例

4.1 モジュールの協調的動作の実現

前章で説明した処理モジュールのいくつかは並行して動作しており、同期をとりながら協力して処理を行っている。各モジュールはいずれも一つの

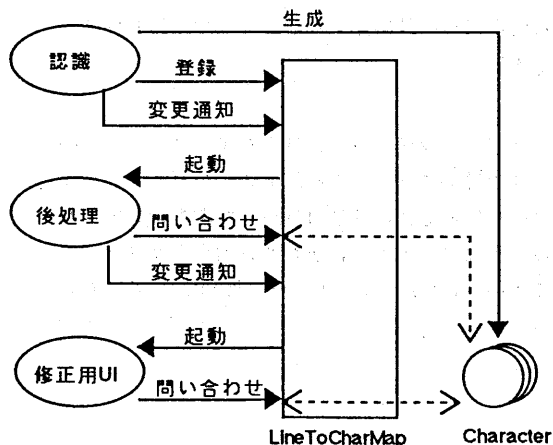


図4: 処理モジュールの協調動作の例

オブジェクトとしてインプリメントされており、メッセージによって起動やパラメータ設定が行われる。DRSのソフトウェアはこれらの処理モジュールと処理結果を格納しているオブジェクト(データオブジェクトと呼ぶことにする)及びマップと呼ばれるオブジェクトで構成されている。マップはデータオブジェクトの集合を管理しており、データオブジェクトの登録の機能や各種の問い合わせに対して該当するデータオブジェクトの集合を返答する機能を有している。また、マップは処理モジュールからのデータ変更の通知を受けて他の処理モジュールを起動することができる。各処理モジュールはマップから得たデータ(例えば行データ)をもとにして処理を行い、新しいデータオブジェクト(例えば各文字の認識結果)を生成したり、データオブジェクト内の情報を変更したりする。新たにデータオブジェクトを生成した場合、処理モジュールはそれらを適当なマップに登録し他の処理モジュールがそのマップを介してデータオブジェクトにアクセスできるようにする。

マップの導入によってインプリメンテーション上、次のような利点が得られる。

1. 並行動作しているモジュール間のデータの授受や同期のためのインターフェイスが共通化され、モジュールの追加や差し替えが容易に行えるようになる。
2. データの変更によって処理モジュールが起動されるといった事象駆動型の制御がある程度可能になる。
3. 処理モジュールからの問い合わせに高速に答えるための補助的なデータ構造をマップの内部データとして隠蔽できる。

図4に認識、後処理、修正用ユーザインタフェースが協調して動作しているようすを示す。認識

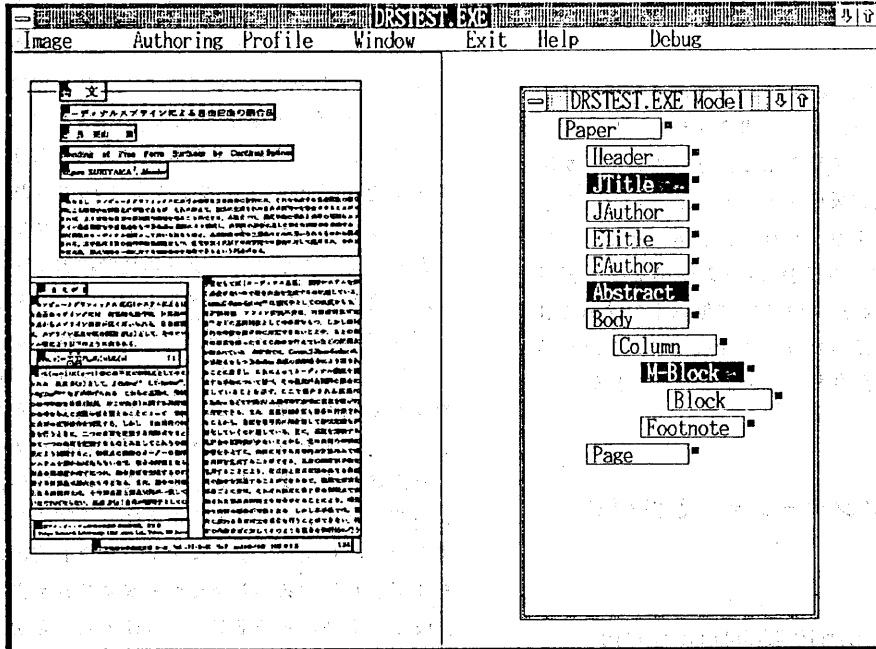


図 5: DRSによるレイアウト解析の実行例

モジュールは各文字に対する認識結果を表すオブジェクトCharacterを生成し、行データとの対応関係をマップLineToCharMapに登録する。認識モジュールはデータの変更通知メッセージをマップに送った後、次の行の処理を続行する。変更通知を受けたマップは後処理と修正用ユーザインタフェースに起動メッセージを送る。どの処理モジュールを起動するかという情報はマップ内に予め登録しておくことができる。起動メッセージの原因となったデータの変更が本当に自分に関係あるものかどうかは処理モジュールで判定する。データの変更（この場合はある行に対する認識結果が生成された）を知った後処理モジュールは文脈後処理を開始し処理が終わった行に対しては認識モジュールと同様にマップに変更通知を送

る。この変更通知によって発行された起動メッセージを受けた修正用ユーザインタフェースは後処理済みの認識結果をテキストウィンドウに表示する。

インプリメンテーションはデバイスドライバの一部を除いてすべてC言語によって行った。メッセージは構造体に貼り付けられた関数のコールとして実現されており、また並行動作をサポートするためOS/2のマルチスレッド機構を利用している。

4.2 文書画像処理の例

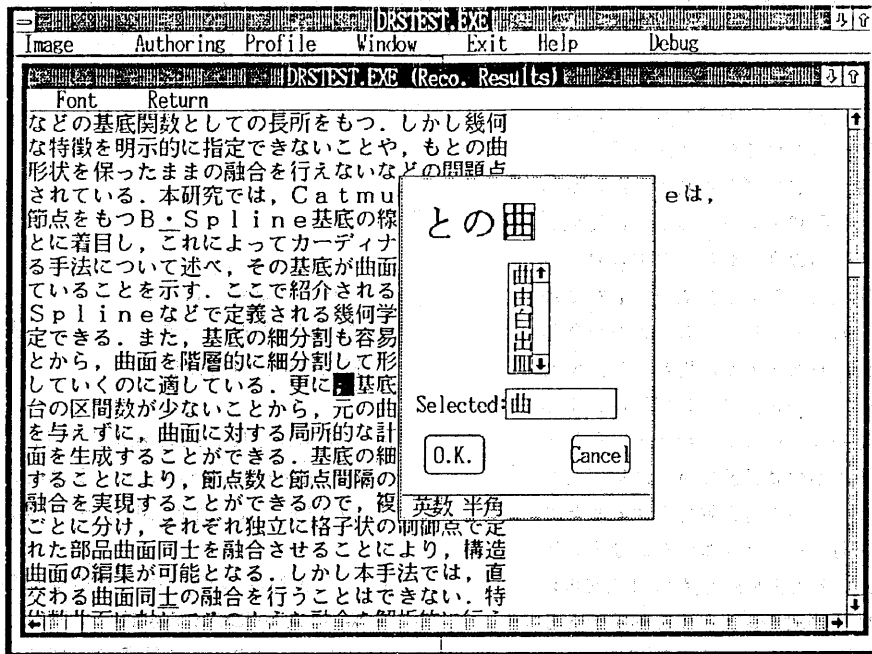


図 6: DRSによる認識、後処理及び修正の実行情例

図5はDRSによって電子情報通信学会論文誌の論文フロントページのレイアウト解析を行った結果を示している。画面右のウィンドウ内のレイアウト木構造の各構成要素に対応する矩形領域が左側の縮小画像上で枠で囲まれている。オペレータはマウスで認識する領域を指定することができる。図6は認識、文脈後処理された結果がウィンドウ内に表示されているようすを示す。並行処理を行っているので全ての行に対する認識及び後処理の完了を待つことなく確認修正を行うことができる。文脈後処理が一文字単位の認識結果を修正した部分には下線が引かれ(例えば最下行の「士」)、誤りの可能性があると判定された部分は白黒反転表示されている。図の中央付近に表示されているのは誤り修正用のダイアログボックスである。処理速度はOCRカードによる文字認識が30文字/秒、PS/55側での文脈後処理が27文字/秒であった。

5 むすび

文書画像のレイアウト解析、文字認識、文脈後処理等の機能を統合し、既存印刷文書からマルチメディア文書への変換を支援するシステムDRSのプロトタイピングを行った。試作したシステムではいくつかのモジュールがパイプラインを形成して、処理が終わったデータを順に次工程に渡していくための枠組みが提供されている。このときモジュール間のデータの授受や同期はマップというオブジェクトを介して行われており、各モジュールはパイプラインの前段あるいは後段のモジュールが何かということ意識しないですむ。例えば、修正用ユーザインタフェースは前段の文脈後処理の有無にかかわらず同一のインタフェースに従って情報を得、表示を行うことができる。したがって、処理モジュールを漸次追加してシステムを成長させていったり、アプリケーションに応じて必要なモジュールを組み合わせてシステムを構

成するといったことを容易に行なうことができる。

ユーザはDRSのようなシステムに対して非常に高い文字認識率を期待しているが、認識処理単体で文書の種類や画像の品質に影響されずに高い認識率を達成することは困難である。しかし、認識と他の補助的な処理を組み合わせることにより大幅な精度の改善が可能になるのは認識と文脈後処理の例にみられるとおりであり、この考え方が適用できるケースは他にもあるように思われる。例えば科学技術文献の文章の入力に際しては、日本語を対象とする認識モジュールにくわえて、英語部分を検出し英文字専用の認識及び英単語辞書との照合を行うモジュールや参考文献を示す肩番号を検出し数字認識を行うモジュールがあれば有効であろう。システムの肥大化を避け、かつ対象文書に応じて必要な補助的処理モジュールを利用できる汎用性を実現するためには、これらの処理モジュールが独立してインプリメントされ自由に組み合わせさせて使えるようになってるのが望ましい。上述の枠組みはこういった要件を満たしており、実用システムのためのプラットフォームになっている。今後はこのプラットフォーム上で処理モジュールの追加や改良を行ってゆく予定である。

謝辞

本研究の機会を与えていただいた高橋弘彦画像情報システム次長(現在IBMアルマデン研究所)、文字認識モジュールの開発にご協力いただいた加藤

真研究員、文脈後処理についてご教示いただいた当研究所日本語処理グループ諸橋正幸担当、丸山宏研究員に感謝します。

文献

1. 中野, 藤澤: 自動ファイリングのための文書理解の方式, 信学技報PRU86-30, 1986.
2. K.Wong, R.Casey, and F.Wahl: Document Analysis System, IBM J. Res. Develop., 26-6, pp. 647-656, 1980.
3. 宮原, 鈴木, 多田, 壁谷: 文書情報の蓄積検索システムに関する検討, 情処ヒューマンインタフェース研資29-3, 1990.
4. 高橋: 細線連結素方向による簡易手書漢字認識, 信学技報PRL82-8, 1982.
5. 青木: 「特集・次世代入力機器の開発状況」を読んで, 新聞技術, 132, pp. 34-38, 1990.
6. 高尾, 西尾: 日本語文書リーダー後処理の実現と評価, 情報処理学会論文誌, 30-11, pp. 1394-1401, 1989.
7. 伊東, 加藤, 高橋: 文字認識における簡易後処理方式, 情処第34回全国大会, 4E-9, 1987.