

手書きレイアウト図面認識

Recognition of Handwritten Layout Drawings

高村 誠之*・高木 幹雄**

Seishi TAKAMURA and Mikio TAKAGI

1. はじめに

物の形や図形の認識や計測や検査を自動化できないだろうか、という要望に対して、「画像処理」という技術への期待は高まる一方である。この技術に関する論文や総説や個別の応用技術の事例報告の数も年々増加の一途である。筆者らは、特に擾乱の多く含まれる図面を対象とした図面認識アルゴリズムを研究しているが、その応用として手書きレイアウト図面を認識するシステムを試作した。本システムで認識の対象としているのは、印刷業者に持ち込まれる手書きのレイアウト図面である。以下に見るようにこの実現にはさまざまな困難がある。

レイアウト図面とは、出版に用いられる一種のフォーマット指定図面であり、最終的な頁に含まれる図や文字領域に関するさまざまな情報を含んでいる。DTP化が進んでいる現在、DTPツールにより作成されたレイアウト図面ファイルも無論あるそうだが、まだデザイナーが手で書いたものが大多数である。そのような手書きのレイアウト図面をCADで使えるデータに変換するのに現在はデジタイザを用いて人手で入力を行っている。レイアウト図面における、図形要素の形と製版上の意味を図1に示す。**定型図形**は、矩形や‘N’字形、‘Z’字形、マス形などの決まった形をしていて、たとえば矩形の位置が何かの枠の位置を表していたり、写真の入る位置決めの情報を与えたり、文章の領域などの位置情報を表している。**文字**は、この部分は文章が何行何列であるか、など補助的な情報を示すのに使われる。**自由図形**は、ラフなスケッチなどによりたとえば枠内に置く写真の位置を細かく指定したり枠のない写真の位置を決める手がかりとするために描かれたり(絵型)、

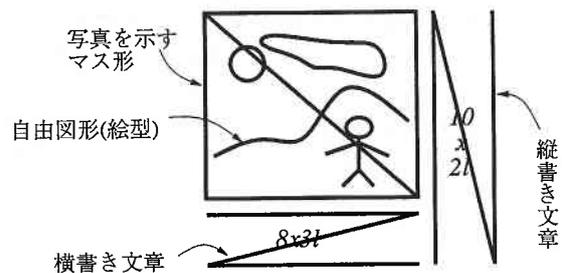


図1 定型図形

ロゴなどを描くのに使われる。このような手書きレイアウト図面の認識を特に困難にしている問題点として、以下のものが挙げられる。

1. 手書きに起因する乱れ
一線のかすれや切れ、はみ出し
一直線のゆらぎ、傾き
2. 書く人によりフォーマットや乱雑さの度合が多種多様であること
3. 方眼線などさまざまなノイズが存在する。またスキャナの読み取り位置によりノイズの性質が変わる
4. 同一種の定型図形でも縦長のものもあれば横長のものもある
5. 読みとったデータが膨大で、精度を求めて時間のかかる処理をはさむことは現実的でない

特に4番目の性質は、同一の回路部品は合同であると仮定できる回路図面などには見られない性質と言える。現在のところ、このような厄介な図面を高い認識率で認識できる実用的なシステムは存在しない。そこで、先にも述べたようにオペレータがデジタイザで欲しい座標を拾って入力しているのが現状である。また、ある程度の線の途切れ、

*東京大学生産技術研究所 第3部

**東京大学生産技術研究所 付属機能エレクトロニクス
研究センター

研究速報

ぶれに対応する手法はすでにいくつか出ているが、このように激しい場合でも認識できるアルゴリズムはまだ確立されていない。手書きレイアウト図面を認識するためのシステムの開発が強く望まれていると言えるだろう。

そこで、自動化できる部分をコンピュータで補わせることを目標に、以下の基本方針

- レイアウト図面あたりの膨大な入力量、および手書きによる多くの擾乱にも対応しうるシステムであること (高速性、雑音耐性)
- 図面の性質に依存した認識手法を用いることは多かれ少なかれ避け難いことではあるが、できる限り汎用的な手法を用いつつ実装すること (一般性)
- さまざまな形状を持つ定型図形や新しい定型図形にも容易に対応ができること (柔軟性)

に留意しながら開発を行った。

2. メッシュを用いた線分抽出アルゴリズム^{8),9)}

本システムでは、図面認識の基本処理である線分抽出にメッシュ分割による方法を用いている。これは文献⁵⁾の手法を大幅に改良したものである。この手法はもともと高速に図面の細線化を行うという目的で考案されたもので、

- 高速に線分が抽出可能
- 雑音に強い
- 線分の位置はメッシュサイズに量子化される
- 太さは保存されない

という特徴をもつ。メッシュに区切ってから周囲4方向との結合を見るというアイデアはrobustな線分抽出によく向いている。本手法はこの性質を生かしつつ、さらに

- 最も適したメッシュサイズの自動決定
- 抽出線分の位置は元の線の中心に一致する
- 太さ情報が得られる

という特長を持つものである。

3. 線分の管理・加工

特定の領域にある線分の検索・削除が高速かつ容易にできるようにするために、木構造により線分を管理する。これにより検索・削除に要する時間は線分数の対数になる。採用したのはBD木で、これは多次元データの効率的な管理、検索を目的として考案されたデータ構造で、完全平衡木でかつ実用時のメモリ効率は約70%である⁶⁾。全処理工程を通じ、すべての線分は二つの端点の座標および太さの属性を持たせて保存する。

現段階での線分は、メッシュサイズごとにおつ切りになった状態で保存されている。このままでは線分数が多すぎ、認識の効率が低下するので、一定の基準の下で直線と

みなせる折れ線は直線として認識していくことにする。具体的には、最大許容誤差の閾値内で、最も多く点列を近似する直線を点列の端から求めていく。本システムでは第一主成分直線を近似直線とした。この理由は、線分を端点から辿って距離を測定していく処理において、直線の計算が逐次的に行え、端点を更新していくたびに計算し直す必要がなく、高速な処理に向いていること、角度に関係なくよい近似の直線が得られること、などである。

ここで得られた直線を元に、次は少々端点が離れていても直線とみなせるものは次々につなぎ長直線を抽出する処理を行い、最終的な認識処理に渡される線分群を得る。

4. 定型図形認識アルゴリズム¹⁰⁾

ここまでの処理で得られた線分を元に、認識処理を行うわけだが、第一段階としてこれらの線分に通し番号をふり、両端に'Head', 'Tail'の印を任意につける。次いで、各線分の端点の近傍を探索し、他の線分端点が存在していればHead/Tailの別および線分同士のなす角度を接続関係表に記録する。

いったんこの表を作成した後は、これを用いて定型図形の認識を見通しよく行うことができる。基本的な方針は、表をもとに再帰的に総当たりで線分を辿っていき、定型図形をなしているものを抜き出すというものである。探索中に目的の図形から形状が外れた場合にはそれ以上の探索は行わないため、後に見るように総検索回数において組み合わせ爆発は起こらない。

4.1 積算角を用いた図形認識

積算角とは、「複数の線分を辿っていく際、各接続点で記述関係テーブルに示される線分間の角度を加算していった値」のことで、たとえば矩形を検索したい場合、「なす角度が約90°の角を4つ構成する閉図形」と考え、以下のような手順により検索が行われる。

1. 積算角 = 0°として、与えられた線分および端点から探索を開始する

```

;This is for figure 'N'.
states = 3 ;状態数の数(0..2)
0 { ;state0→state1の条件
angle = 110..175 ;大きく曲がったとき次の状態へ
}
1 { ;state1→state2の条件
angle = 0 ;元の角度に戻る
project 1 0 Intersection; 錨1と錨0を結ぶ直線へ該当線分
} ;の交点を下ろした足が錨0に近ければ次の状態へ
2 { ;終了条件
angle = 0 ;元の角度のまま、
project 0 1 Opposite; 錨1と錨0を結ぶ直線へ辿ってい
} ;る線分の逆端点を下ろした足が錨1に近ければ終了
end

```

図2 'N'形の簡易言語表現

2. 探索途中においては、近傍から角度が90°に近い線分を接続関係表から得る。もしその角度範囲に線分がなければ4.へ、線分があればその角度を積算角に加え、その線分のもう一方の端から引続き探索を行う
3. 2.からの処理を再帰的に繰り返し、もし探索開始点に戻ってきたら矩形が発見できたとして処理を終了する
4. 全線分について1.からの処理を繰り返す

4.2 出力の重複チェック

こうして抽出された図形群は、必ずしもおのおのが別々の図形を表しているとは限らず、中に同一の定型図形を表しているものがある。この原因の一つは、たとえば‘N’形の場合、二つある端のどちらから検索を行っても同じ結果が得られてしまうからである。(このとき、出力される線分は逆順になっている。)他にもさまざまな原因で出力に重複が生じてしまう。

この問題を解決するために、

1. 検索結果のすべての出力について、同じ図形を表していると思われるものをグループ化する
2. 同じグループの中から最も指定図形をよく表していると思われるものを選択する

という処理を行っている。

また、同一グループ内で最もよく指定図形を表しているものを選択のためにペナルティという値を導入する。ペナルティは、線分群から指定図形を再帰探索している時に、線分間のギャップや角度のずれに応じて加算していく。同一グループに属する各々の検索結果についてこのペナルティを比較し、最も小さいものを選択すればよい。

4.3 簡易言語による図形表現

本システムでは、図形を表す本質的な部分だけを抽出して記述し、それを元に認識できるようにするために、簡易言語をインプリメントしている。これにより、一般の線図

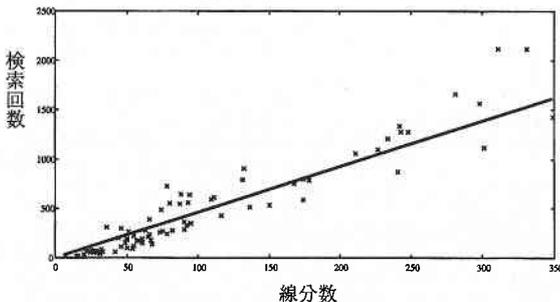


図3 図面に含まれる線分数と検索処理回数の関係

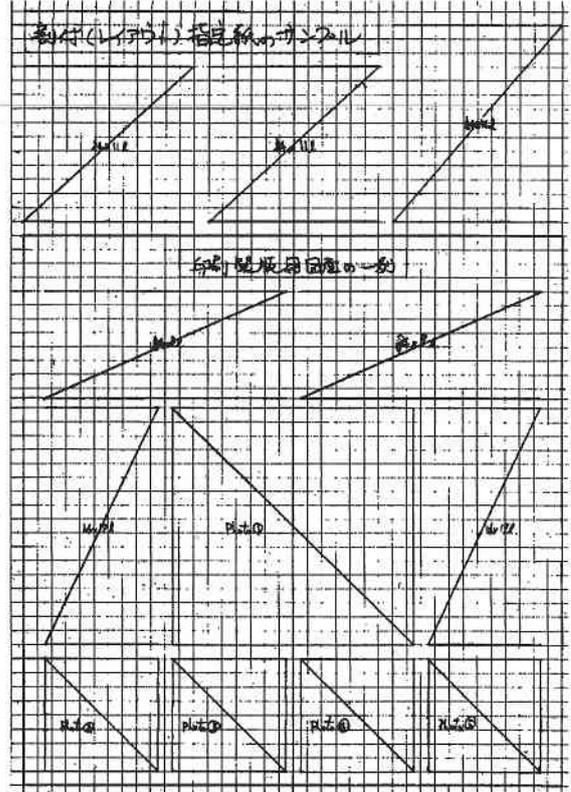


図4 レイアウト図面

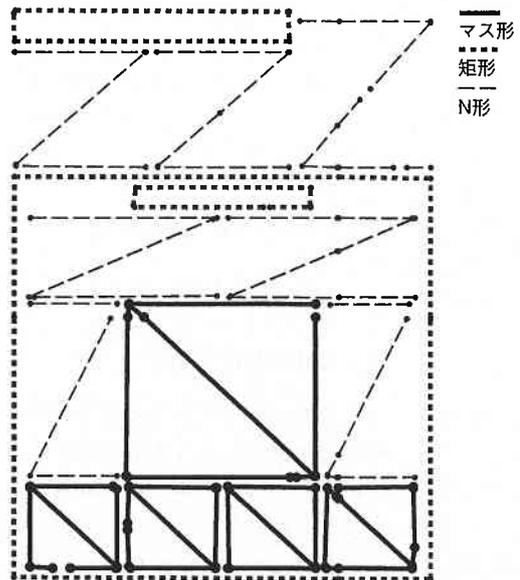


図5 認識結果

形の認識に容易に応用が可能となる。たとえば任意の縦横比および傾きを持つ‘N’形は、図2のように非常にシンプルに記述される。

研 究 速 報

5. 実 験 結 果

ワークステーション HP9000/720に C 言語を用いてシステムを実装し、実験を行った。

図面に含まれる線分数および定型図形 (矩形) を抽出するのに要する線分検索回数を比較した結果を図 3 に示す。線分数と検索回数はほぼ比例した関係が読みとれ、回帰直線を計算した結果、「検索回数 $\approx 4.64 \times$ 線分数」という関係が得られた。

図 4 にサンプルとした実際の手書きレイアウト図面を示す。これは 100dpi でスキャナ入力されたもので、横 776 画素、縦 1120 画素である。この図面の認識結果を図 5 に示す。二値化図面入力から認識結果出力までの全所要時間は 4.66 秒であった。バックグラウンドの雑音、手書き文字の擾乱にもかかわらず、すべての定型図形が正しく認識されていることがわかる。

6. お わ り に

以上、擾乱の多い図面を認識するためのアルゴリズムおよび手書きレイアウト図面の認識システムの認識結果について報告した。

線分抽出において、画素配列へのアクセスは、ラン長の計算時、およびメッシュに区切り各メッシュの重心パラメータを得る時のみに行われるため、線分抽出後の画素配列へのアクセスは不要となる。検索回数は線分数にほぼ比例する (線分本数のわずかに数倍) という値が得られ、効率よく探索が行われていると言える。

検索アルゴリズムにおいては、純粹に線分同士のなす角度のみに注目しているので、任意の角度に傾いた図形の抽出が可能である。また、積算角および状態数の概念を使えば、局所的な視野のみに注目しながら探索しているのにもかかわらず任意の閉図形の記述および検索をかなり簡単に行うことができる。また有り得るケースをすべて探索した

後に重複チェックを行うという方針で、最も確からしいと思われるものを残すため、認識誤りは最小限に抑えられている。

本システムには、バックグラウンドの細線を除くためのスレッシュホールドの決定法、線分をつないでいく時のパラメータなどにヒューリスティックなパラメータが存在する。理想的にはこれらを自動決定するアルゴリズムの開発も必要である。また現在は検索図形が一筆書きに限られているが、ここで用いている表による見通しのよい検索をさらに進めて、任意の枝分かれに対応した検索への発展も考察中である。
(1993年12月8日受理)

参 考 文 献

- 1) 中嶋, 安居院, 吉井, 今井: “地図・図面入力の技術動向”, 信学技報, IE88-37
- 2) 坂内: “図面の自動読取り技術”, PIXEL '91-1 (No. 100), pp. 74-77
- 3) 後藤, 吉田, 福村: “略図にもとづく市街地地図からの道路図の抽出”, 信学技報, PRU87-6, pp. 1-8
- 4) 山川, 小田: “会話型図面入力システム”, 日経コンピュータグラフィクス Apr. 1987 pp. 120-130
- 5) 山崎, 江島: “手書き論理回路図認識システムの高速化について”, 信学論, '90/10 Vol. J73-D-II pp. 1786-1788
- 6) 大沢, 坂内: “良好な動特性を持つ多次元データ管理構造の一提案”, 信学論 (D), J66-D, 10, Oct 1983, pp. 1193-1200
- 7) Rangachar Kasturi: “A System for Interpretation of Line Drawings”, IEEE Transactions on Pattern Analysis and Machine Intelligence. Vol. 12, No. 10, Oct 1990, pp. 978-991
- 8) 高村, 高木: “雑音に強い線分抽出手法”, 情報処理学会第45回全国大会 Oct. '92
- 9) 高村, 高木: “線分の太さ推定を用いた線分抽出手法”, 電子情報通信学会第45回全国大会 Mar. '93
- 10) 高村, 高木: “擾乱の多い線図面認識手法の一提案～手書きレイアウト図面への応用～”, 信学技報 HC93-44 Oct. '93