

新しい線分抽出方法

2D-1

李圭宰 國井利泰
東京大学

1. はじめに

最近さまざまな分野でCADシステムが実用化されるようになり、その結果、作業能率の飛躍的な向上が図られている。ところが一方では、CADシステムの導入に起因する新しい問題も生じてきている。その一つにデータの互換性の問題がある。例えば図面を考えた場合、CADシステム導入以前から既に多数の図面が蓄積されている。これらの図面を再利用するためには、既存の蓄積方法とCADシステムとの間でデータの互換性を取らねばならない。また、他のシステムを用いて作成された図面を用いようとする場合にも同様な問題が生じる。

これらの図面を描くようにCADの入力システムで入力するには人間の手間や時間がかかり過ぎる。そこで、図面をスキャナで読み取り、それを自動的にCADシステムに適したデータ形式に変換するという方法が採られている。このような図面の自動認識において、最も重要かつ基本となるのが線分の抽出であり、この論文では、スキャナで読み込んだ図面のイメージ・データから線分を抽出する新しい方法を提案し、実現した。また実際の問題に応用することにより、有効性を確認した。

2. 線分抽出プロセス

図面から線分を抽出するプロセスのアウトラインは次のようになる。

1. 図面をスキャナで読み取る。
2. 細線化する。
3. ピクセルからセクターを構成する。
4. セグメントから直線および曲線を構成する。

読み取った図面は1と0の2次配列データである。ここでは、配列の単位をピクセルと呼び、値が1のピクセルを'1'ピクセル、0のピクセルを'0'ピクセルと呼ぶことにする。

読み取った図面の線幅は(スキャナの精度によるが)ピクセル3、4個以上になるため、まずこれをピクセル1個

の線幅にする必要がある。この作業を細線化という([1])。ここで、細線化後に残った'1'ピクセルを図1のように、コーナ・ピクセル、エンド・ピクセル、そしてコネク・ピクセルの3種類に分類する。コーナ・ピクセルとエンド・ピクセルは、それぞれ図形の角点、(線分と線分の)交差点に当たるピクセルで、これら以外に線分を構成している'1'ピクセルがコネク・ピクセルになる。エンド・ピクセルとコネク・ピクセルは図2のようにまわりの'1'ピクセルの数と位置関係ですぐ分類できるが、この際、コーナ・ピクセルはコネク・ピクセルとして認識される。コーナ・ピクセルの認識方法は3節で詳しく述べる。

次に、細線化された2次配列データからセグメントを構成する。セグメントとは'1'ピクセルの1次配列で、その両端のピクセルはコーナ・ピクセルかエンド・ピクセルであり、それ以外はコネク・ピクセルで構成されるものである(図3)。

最後に、このように構成されたセグメントを、向きなどを考えながらつなぎ、本来の目的である直線および曲線を抽出する。

3. セグメントの構成

この節では、セグメントの構成方法、特にコーナ・ピクセルの認識方法について詳しく述べる。現在、コーナ・ピクセルの認識には、直線抽出方法の、始点と終点を結ぶ近似方法、2乗誤差最小化法、追跡法などが使われているが、これらには次のような問題点がある。

1. 1回の計算の量が多い
2. ピクセル1個に対して2回以上の計算を必要とする
3. セグメントの両端点の位置をあらかじめしらないと適用できない
4. もとの角点あるいは交差点の座標値が線分抽出後変わる

上記した問題点を解決するために、ここで、直線をピクセルレベルで表した際の性質([2])を利用した、コーナ・ピクセルを認識する方法について説明する。前述したように、セグメントは、エンドあるいはコーナ・ピクセルから出発して、また新しいエンドあるいはコーナ・ピクセルに出会うまで、隣のコネク・ピクセルをつないでいて構成する。この際、コーナ・ピクセルの認識は大きく次の

2つのルールに基づいて行う。

ルール1

図4のように3番目の新しい方向を創るピクセルはコーナ・ピクセルとして認識する。

ルール2

図5のように、セグメント構成中に守られてきた規則性をやぶるピクセルはコーナ・ピクセルとして認識する。

4. むすび

読み取った図面は単なる1と0の配列データ(ラスタ-データ)であり、それらをCAD上の直線や曲線データ(ベクターデータ)などに変換するためには、シンタクティックなアプローチだけでは無理がある。そのため、何らかの形でセマンティックなアプローチを加えなければならない。その解決策として、属性文法を用いて、セマンティックな情報を属性(ここでは、セグメントの向きなど)で表し、その属性に従って変換を行う方法を採用している。さらに、属性として抽出されたデータをデータベースに蓄えることにより、図面データベースの構築も可能となる。

■参考文献

- [1] Nakayama A, Kimura F, Yoshida Y and Fukumura T (1984) "An Efficient Thinning Algorithm for Large Scale Image Based upon Pipeline Structure": The Transactions of the Institute of Electronics and Communication Engineers of Japan J67-D(7):761-767
- [2] Rosenfeld A, (1974) "Digital Straight Line Segments": IEEE Transactions on Computers C-23:1264-1269

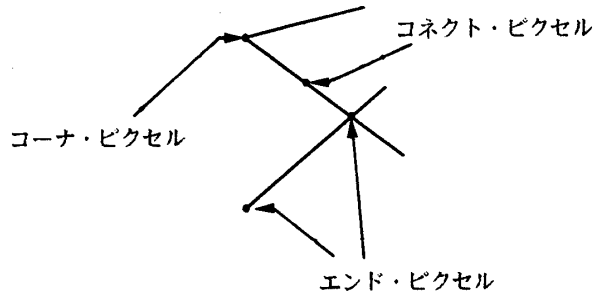


図1. ピクセルの分類

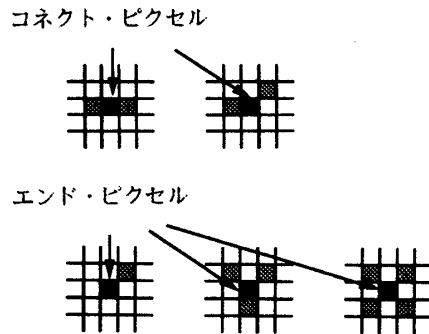
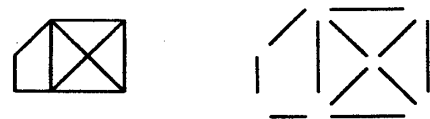


図2. エンド・ピクセルとコネクト・ピクセル



(a) (b) 図aのセグメント

図3. セグメント

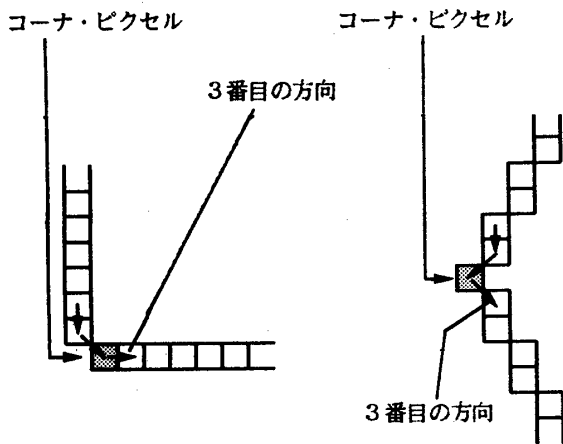


図4. 3番目の方向を創るコーナ・ピクセル

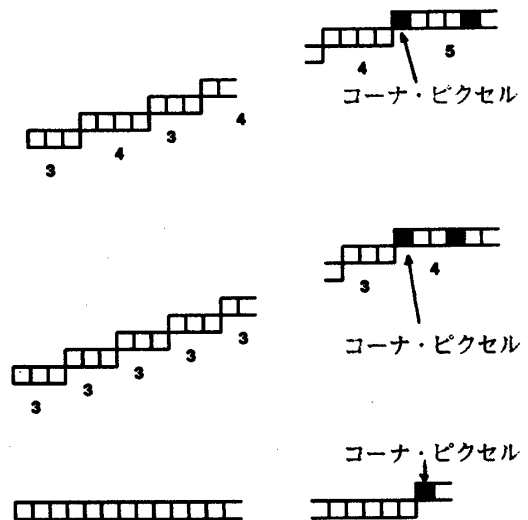


図5. セグメント構成中の規則性を破るコーナ・ピクセル