

考察： 「パターン認識における文脈処理」

現在、鋭意、英語続け字認識に挑んでいるのですが、その前にパターン認識全般について考察してみました。パターン認識における曖昧性とはどう管理し処理していけばいいのでしょうか。この点について明確にしておくことは、パターン認識の大きな問題をクリアするときには必須のことですので、これからの文字認識作業においても大いに有意義な一歩となることでしょう。

この考察は、論文「英子文字認識の試行」の続編でもあります。

第1章 はじめに

パターン認識ではどんな曖昧性があるでしょうか。以下に整理してみたいと思います。次のような曖昧性があります。

- (1) 形の曖昧性
- (2) 重ね合わせからの曖昧性
- (3) 雑音、図形修飾による曖昧性
- (4) 変形による曖昧性

パターン認識ではこうした曖昧性を解決していかなばなりません。それは、大局を示す情報を利用して、ここの図形要素の解釈を決定していくという作業が必要になるということです。大局情報は、全体画面で与えられる場合も有りますし、知識や常識といった知識ベースな情報を導入しなくてはならない場合もあります。

こうした大局情報を文脈情報とよび、この曖昧性を処理していくのを文脈処理と呼ぶことにしましょう。

まずは、各曖昧性の具体例を挙げてみます。

1.1 形の曖昧性

形をどう捉えるかはものすごい自由度があります。次の例のようです。

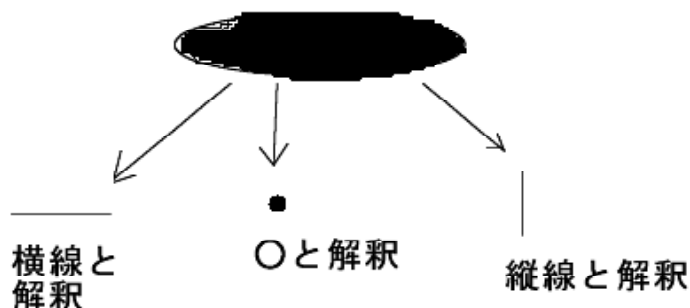


図1.1 塗りつぶし領域の解釈

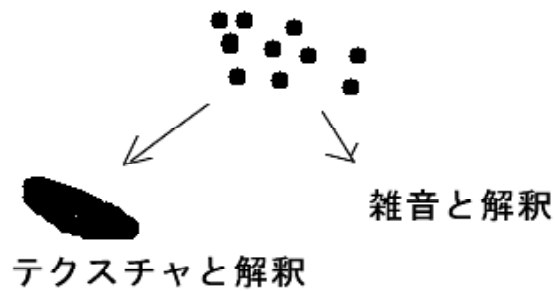


図1.2 ドットパターンの解釈

1.2 重ね合わせからの曖昧性

大局的な情報が無いと、交差線の解釈は複数あり得ます。

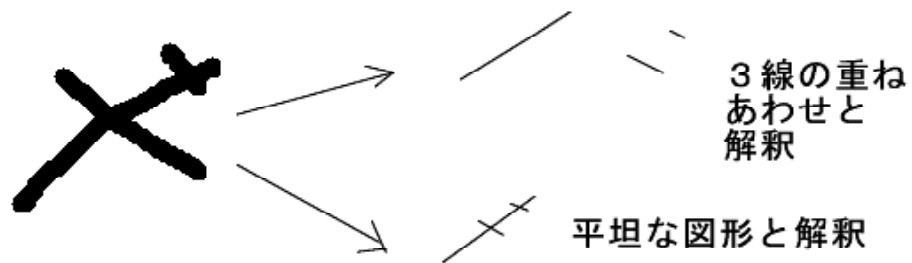


図1.3 重ね合わせの曖昧性

1.3 雑音、図形修飾による曖昧性

雑音なのか、実際に重要な図形要素なのかは判断に困るところです。

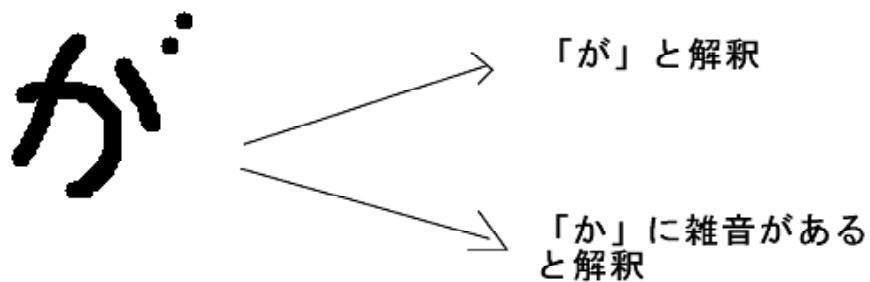


図1.4 雑音、図形修飾による曖昧性

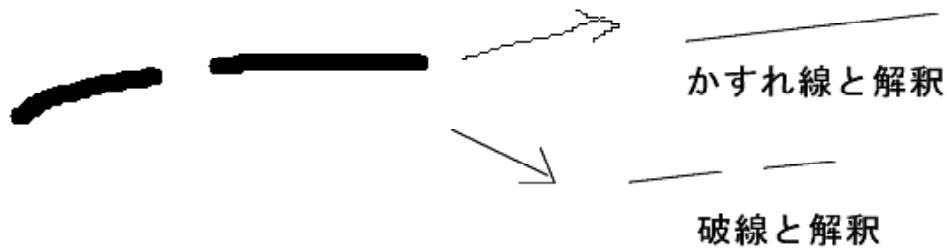


図 1 . 5 雑音による曖昧性

1 . 4 変形による曖昧性

現実世界では様々な変形が図形に加えられます。そのため、テンプレートマッチングができないとか、かなり厳しい解析をしないと同一性が把握できないことが多々あります。そもそも、変形なのか、それが新しい別の図形なのかという曖昧性にもなります。



aとbは同じ魚とも、別の魚ともとれる

図 1 . 6 変形による曖昧性

第 2 章 各曖昧性の処理方法

各曖昧性について解法のアイディアを記していきたい。曖昧性管理フレームというものを導入して、そこに複数の解釈を並記して状況により利用していけるようにします。

2 . 1 形の曖昧性

細線化情報との突き合わせで、端点、分岐点、交差点を求めていくのがよいようにみえます。

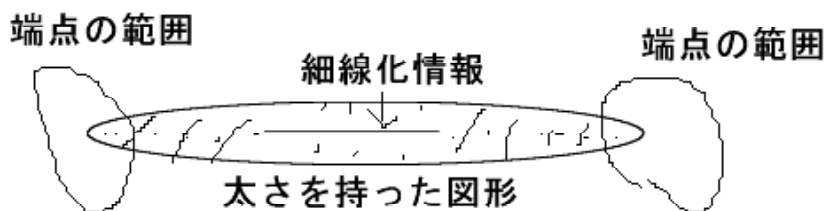


図 2 . 1 細線化情報の利用イメージ

細線を延長して元の図形との交点を求めて判断します。また太さを持ちますから、黒丸とも解釈できると曖昧性管理フレームに記しておきます。

2.2 重ね合わせからの曖昧性

交差点を見いだしたら、重ね合わせ点との解釈と特徴点との解釈で曖昧性管理フレームに並置して記していきます。このとき、解釈に応じてフレームを追加し、図形全体の解釈を決定づけ、無数の解釈フレームを持つようにするのは組み合わせ爆発が生じるので好ましくありません。要素イメージ毎に曖昧性を管理し、必要に応じて全体解釈を生成していくべきです。その時には知識などを動員して、意味ある解釈となることをこころみるので、解釈は一つとなります。無論一部では曖昧性が残り、揺らぎとして把握しておく必要が生じることはあります。

2.3 雑音、図形修飾の曖昧性

基本図形の解釈と雑音パターンの解析とは平行して行うべきです。雑音は基本的にドットパターン（バーストがあつたりします）であり、その傾向も一つの規則性があるわけです。図形修飾も規則性がありますから、その傾向の把握は重要です。雑音、図形修飾は影のパターン認識として、基本パターン認識に寄り添って存在するものといえます。文字ですと、明朝体、ゴシック体とかそんな情報もこの図形修飾解析の対象になるものでしょう。

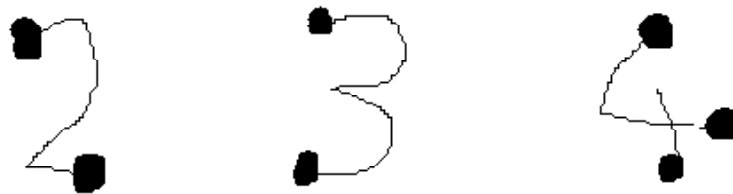


図 2.2 修飾パターン

2.4 変形による曖昧性

変形は剪断は別として、画素の順序は不変です。画素の間距離が短くなったり長くなったりするだけです。つまり、位相は不変です。だから、変形処理は基本的に次の2つを解析すればいいということになります。

- (1) 剪断
- (2) 位相不変変換

いずれの場合も各画素に対して対応をとっていけばいいのですが、それでは効率が悪い。特徴点を求めて、その選び抜かれた画象要素での対応を取っていくのが基本でしょう。順次特徴点の周辺の画素対応をとっていくという、粗から精密化へという方法も有効なものです。

2.5 パターン認識の大枠の流れ

パターン認識処理の基本的な流れをプロットしてみたいと思います。流れは、次のようになるのではないのでしょうか。

(1) 抽象プロットタイプとしての画像があって、その組み合わせとして把握する。

抽象プロットタイプとしては、点、線、円、三角、四角があり、それらの属性として、ふくらみ、傾き、曲率があると思います。

(2) その位相を保存した変形を各要素画像に施し、精密な情報を把握する。

(3) 雑音、修飾図形解析を行い、それらと有意パターンとを情報分離できるようにする。

(4) 最終判断を知識との組み合わせで下す。

それぞれのステップで、文脈処理が有意義に利用されていくはずであります。

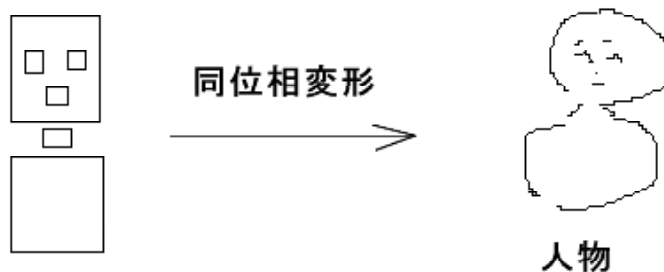


図 2.3 同位相変形

曲線は最終的に、ベジェ曲線近似されると今は直感しています。それで、綺麗に曲線の情報が把握できるからです。解析パラメータ化が容易にできるわけです。

基本的にパターン認識は、画像要素がそこそこにあることを認識し、しかる後、その要素の属性把握していく。最後に画像要素とその属性の組の配置関係を解析していき、最終的な認識結果を得る。そのような手順で進行していくものです。

第3章 マップシステムと曖昧性管理

曖昧性は画素ごとにあるわけです。従って画素が作るマップの各点に曖昧性フレームをコネクしていくべきでしょう。検索の高速化とか、処理の高速化とかの工夫が必要ですから、インデックステーブルを設けたり、配列を用いたマップを設けたりする必要もあります。

つまり、マップシステムの働きは次の通りです。

- (1) 図形情報の配置の保存
- (2) 情報の高速検索、高速処理
- (3) 情報の組織化

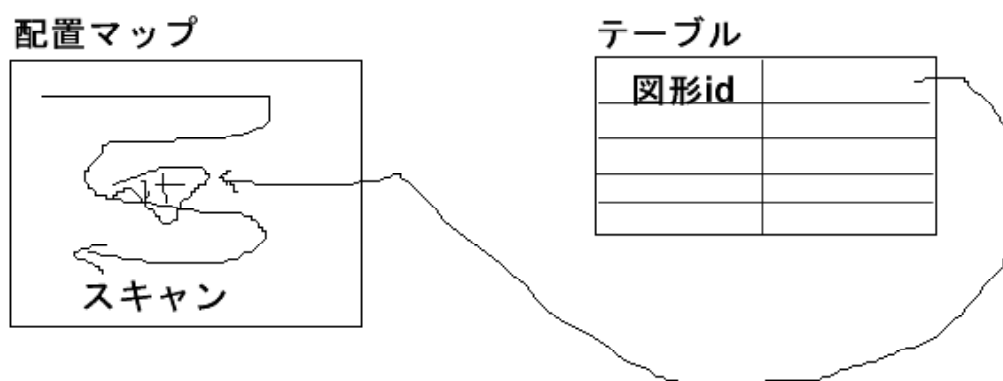


図3.1 マップシステム

3.1 基本的なデータエリア

マップシステムは基本的に次のものがあります。

- (1) ベクターマップ (Javaのベクター配列。情報をオブジェクトとして登録できる)
 - ・特徴図形情報オブジェクトの管理
 - ・曖昧性情報の管理
- (2) 配列マップ
 - ・輪郭線情報
 - ・特徴点配置情報
 - ・細線化情報

マップシステムを用いた曖昧性管理は次の2つを駆使して行います。これらは、ベクターマップと配列マップから作られるオブジェクトです。

(1) 曖昧性集約マップ

一つのマップシステムに曖昧性とか解析結果情報とかをまとめて保管するものです。作業データエリアです。

(2) 情報組織化マップ

図形要素から全体図形解釈まで、決定事項を管理するデータエリアです。

曖昧性集約マップの形式を次に示します。

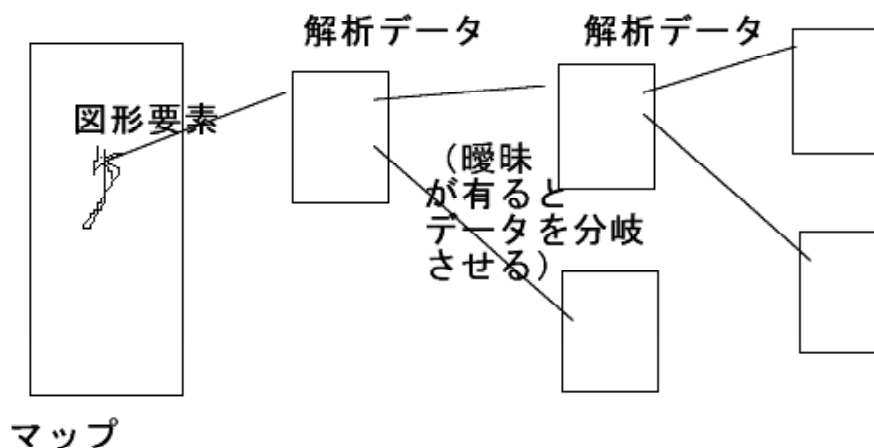
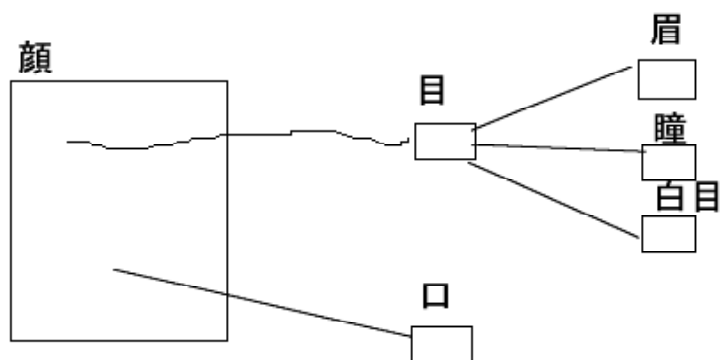


図3.2 曖昧性集約マップ

曖昧性を分岐で表します。結果に至るパラメータも情報として保存します。

情報組織化マップの形式を次に示します。



決定情報は入れ子になる

図3.3 情報組織化マップ

情報の入れ子構造は、図形要素の入れ子構造を反映しますから、それを素直に決定稿に反映させます。

後は文脈の記述となりますが、それは次のフレームで管理できるはずですが。

(1) パラメータセットフレーム

パラメータ項目とその値、あるいは起動するプロセスの識別子を格納するフレームです。パラメータの値は正規化した後のものとして図形の拡大縮小に影響されない汎用性のあるものとしします。

(2) ローカルパラメータフレーム

実際に解析するときを利用する値はデフォルトのものとは部分的に違います。この部分的に違うパラメータを管理するフレームです。

(3) プロダクションフレーム

文脈によって、適応するローカルパラメータフレームは異なってきます。それで、適用条件とその時のアクションを定義したフレームが必要になります。

このように、フレームでパターン解析システムを表現しておけば、自動学習が実現できるようになります。プログラムのメンテナンスも簡素化されます。

3.2 弛緩法

曖昧性処理には、図形要素の漸近的決定を行って行く弛緩法が有効です。途中矛盾する決定があっても、全体の情報とか、知識を補えば、順々に最終結論は集約していくでしょう。途中経過は曖昧性集約マップの細大漏らさず記載していき、その時々決定稿は情報組織化マップに行きます。思いがけない障害を発見したときには、バックトラッキングを掛けることになるでしょう。

弛緩法の記述はプロダクションシステムになるでしょう。

```
if((a:line,vertical#right_curb)(b:hole,round))then{set (letter,id="q",a:#b:)}  
.....
```

このコマンドのイメージは、次の通りです。

条件：線（属性が垂直で右ふくらみ）と輪（属性がまんまる）があったら

行動：文字"q"とする。それを線（識別子 a を振っている）と輪（識別子 b と振っている）のマップ要素に設定する。

第4章 曖昧性管理の効能

曖昧性とは悪いことばかりではありません。曖昧性管理を通して、類推ということが自然に行えるようになるのです。

つまり、ある典型パターンが複数あって、それらが雑音とか変形操作によって、一つの曖昧図形を生成したとすると、複数の典型パターンは曖昧図形をピボットとして、互いに対応付けられるわけです。知識とはそうしたものになります。さすれば、自然と相互に連想が実現される、類推が実現できるわけです。

曖昧性管理は、重要な特徴要素はなにかという判断をもちます。重要な特徴要素を共有するパターン間にも類推による連想が実現されることでもあるわけです。

第5章 学習システム

何が重要な特徴要素かという情報は沢山のパターン群を収集していかねば(コーパス)決定できないことです。パターンコーパスの収集とそのクラスタリングが重要な処理となってきました。特徴は記号で表現されなくては、計算機で処理できません。パターン認識と記号処理の合体を指向すべきところです。

記号化はアナログ量からデジタル量への変換であります。基本的にファジーを背景にもったものがパターンの記号でしょう。ファジーの量としては次のものがあります。

- (1) パラメータ、バリュースセットとファジー量との対応
- (2) パラメータとファジー量マップとの対応

データマイニング技術を極める問題に帰着されるわけです。

第6章 まとめ

曖昧性処理は無数の多階層の情報を扱わねばなりません。解析の途中結果がその後行程にも影響します。文脈処理、文脈管理が重要になります。文脈処理は並行処理での弛緩法が基盤アーキテクチャとなり、それにはマップシステムがデータベースとなるでしょう。並行処理にも弛緩法にも最適な情報表現だからです。今回は、その入り口を論じてみました。

以上