

パターン認識の大枠の構造を考えて行ってみたいと思います。画像とか、画像の中間解析結果をマップにして保存するとか、知識となっているパターンモデルを多数参照するため、そしてそれらを参照しながら、高度に並行処理をしていくため、多数の解析用CPUと、それらCPUが共有メモリを持つことは必須であると言えます。システムを設計するときは、リードオンリーで共有するのか、リードライトなメモリアクセスなのかといったことは重要な問題です。リードライトはロックする必要があるので、処理速度に影響が大であることと、システムを実現するのもリードオンリーよりも負担が大きいからです。

リードオンリーといっても、一度は書き込みが行われます。書き込みは追加書きは許すとして、パターン認識のアルゴリズムを考えて行きますと、多くは、元イメージデータをリードオンリーで読み込み、解析して、結果をマップとオブジェクトにまとめて、メモリに追加書きしていくことが分かります。更新は無いのです。ただ、曖昧性処理のため、重み付投票を行うところがリードライトの更新作業が入ります。そこで、この部分を画像データとは別にオブジェクトとして持ち、リードライトが行える共有メモリに置きます。そうして、ここから本体の画像解析結果データのオブジェクトを参照していくと良いです。

画像解析結果データは、2次元上の配置情報と点なのか線なのか領域なのかといった種類と曲率とか大きさなどの属性を持ちます。配置はマップにデータに名前付けして登録して行きます。名前で、本体のデータを参照できるようにしておけば、ハッシュによって高速に検索できます。重み付投票処理するニューロ部もデータ本体の参照は名前にしておく良いでしょう。2次元のマップは、また解析結果のデータの間の関係を表現するにも有利です。ネットワークで表現する必要もないからです。しかし、マップのある点がどんなデータに含まれるかを検索するのは高速に行われなくなると、マップは複数のCPUから同時に検索することになるでしょう。書き込みフェーズと検索オンリーフェーズを設けるのが良いかも知れません。

画像認識の手順を概観しておきましょう。

- (1) 特徴点、線、領域を解析で得て、それらをオブジェクトとマップで表現します。
- (2) 曲率、2線のなす角度の大きさと分布を解析する。
- (3) 対向線、輪郭線、幾何図形を得ます (重み付投票あり)
- (4) 部分オブジェクトの認識、輪郭線認識、重ね合わせ部分認識 (重み付投票オンリー)
- (5) 全体のシーン認識 (重み付投票オンリー)