画像情報システム

第6回　パターン認識

木更津高専情報工学科　和崎

1. 類似度

- パターン認識では、形の類似度を用いて判断する
  - テンプレート（標準パターン）とどれだけ似ているか？＝類似度
  - 同じ区間（領域）のパターン同士を比較
- 連続関数の区間類似度
  - 区間内の距離（差）の積分をとる　$d=\int_a^b \left| f(x)-g(x) \right| dx$
    ⇒値は小さい方が類似度が高い
  - この方法では、直流成分の影響を受けやすい
    ⇒平均値を差し引くことで、直流成分を除去
    $F(x)=f(x)-\bar{f}, G(x)=g(x)-\bar{g}$として同様に計算
  - あるいは2乗誤差を使って

$$s=\int_a^b \left\{ F(x)-G(x) \right\}^2 dx$$

$$=\int_a^b F(x)^2 dx+\int_a^b G(x)^2 dx-2\int_a^b F(x)G(x)dx$$

類似度を表さない　　類似度に寄与

2. 相関係数

- 連続関数の類似度

$$s = \int_a^b F(x)G(x)\,dx$$

大きさで正規化
$F(x) = G(x)$
のときに1となる

$$R = \frac{\int_a^b F(x)G(x)\,dx}{\sqrt{\int_a^b F(x)^2\,dx}\sqrt{\int_a^b G(x)^2\,dx}}$$

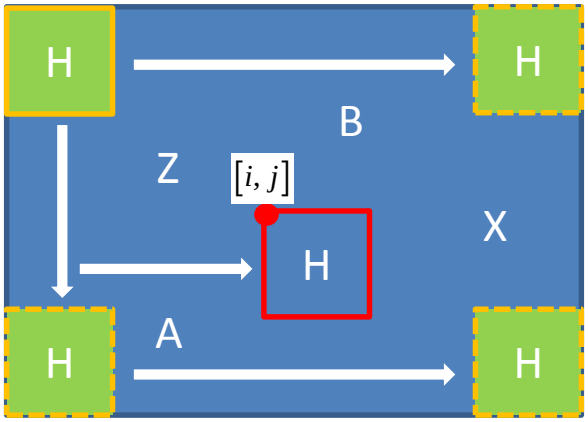
Rを相関係数という
（類似度が高いほど
1に近づく）

- 2つのベクトル（要素数：k）間の相関係数
  - ディジタル画像ではこれで計算する

$$R = \frac{\sum F_k G_k}{\sqrt{\sum F_k^2}\sqrt{\sum G_k^2}}$$

3. テンプレートマッチング

- 入力画像の中から、テンプレートに一致する位置を求める
  - 入力画像にテンプレートを置き、そのテンプレートをずらしながら一致する位置を探す
  - 一致をとるには、厳密には相関係数を用いる
  - 一致のみを検出するなら、単に類似度で判断
    - 計算速度の向上を目的とする



厳密

高速

$$R(i,j)=\frac{\sum_{l=0}^{n-1}\sum_{k=0}^{m-1}F[i+k,j+l]T[k,l]}{\sqrt{\sum_{l=0}^{n-1}\sum_{k=0}^{m-1}F[i+k,j+l]^2}\sqrt{\sum_{l=0}^{n-1}\sum_{k=0}^{m-1}T[k,l]^2}}$$: 相関係数（1に近い値の位置）

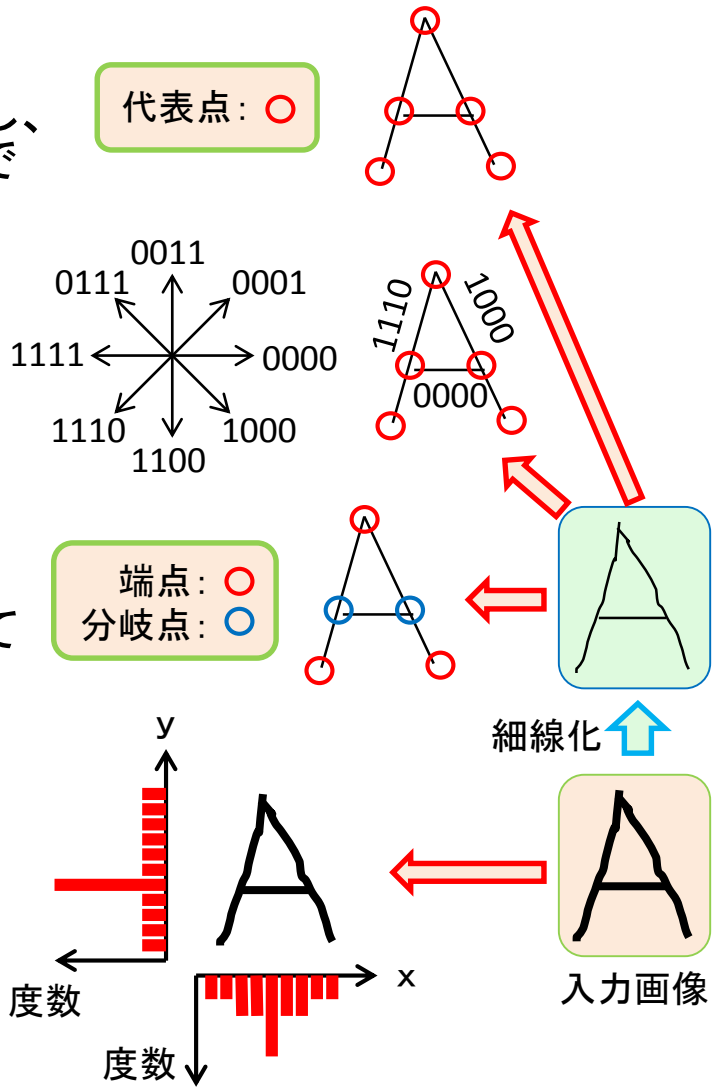
$$s(i,j)=\sum_{l=0}^{n-1}\sum_{k=0}^{m-1}f[i+k,j+l]t[k,l]$$: 類似度（最大値の位置）

$$d(i,j)=\sum_{l=0}^{n-1}\sum_{k=0}^{m-1}\left|f[i+k,j+l]-t[k,l]\right|$$: 距離（最小値の位置）

$f[i,j]$: 入力画像の画素値
$t[k,l]$: テンプレート画像の画素値
$m\times n$: テンプレート画像のサイズ

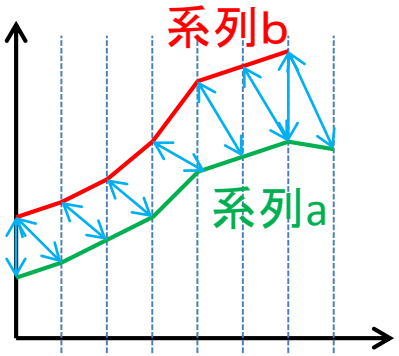
4. 一致検出の方法

- 特徴ベクトルによる方法
  - 位置座標系列による方法
    - 細線化した図形を複数の直線で近似し、折点を代表点として代表点間の距離ではかる
  - 方向コード列による方法
    - 直線近似した図形の各直線の方向をコード化して、コードの差ではかる
  - 特徴点による方法
    - 細線化した図形の端点・分岐点・交差点を検出し、種類や個数ではかる
    - 数字や文字認識の大まかな分類として有用
  - 周辺分布による方法
    - 図形のx方向やy方向の画素数分布を周辺分布として、その分布からはかる
    - 位置合わせによる誤差が大きくなる
      ⇒周波数解析でマッチング



一致検出の方法(2)

- DPマッチング
  - 動的計画法(Dynamic Programming)によるマッチング方法
  - 比較する要素数が異なったり、局所的なずれがあるような場合に有効
  - 要素間の対応付けを行いながら、距離を計算
    - 対応付けは、要素間距離が最も小さくなるように行う
    - 要素間距離は、物理的な要素間距離に重みをかけて算出
    - 重みの選び方が問題（2次元以上のデータでは難しくなる）



課題

- 課題21
  - テンプレート画像と入力画像をテンプレートマッチングで比較するプログラムを作成する
    - 入力画像：in21.bmp　テンプレート画像：in21-〇t.bmp（〇は数字）
    - 入力画像サイズが大きいので、sampleの下にある bmpfile550.o と def550.h を使用すること
    - If (R, G, B) = (255, 255, 255) then 値 = 0 else 値 = 1 とする前処理（2値化）を行うこと
  - 最も一致度が高い入力画像領域の座標を求め、その領域を切り出して解答画像を作成する
    - 切り出す領域の大きさはテンプレートの大きさに合わせること
    - テンプレート画像を複数用意するので、各々についてマッチングを行い、解答画像を作成すること（提出はプログラムリストと解答画像のみ）
    - マッチングの方法は以下の3種類で試すこと
      - 相関係数による方法
      - 類似度による方法
      - 距離による方法
    - テンプレート画像のファイル名がin21-1t.bmpであるとき、以下のように解答画像のファイル名をつけること
      - 相関係数による方法：ans21-1r.bmp
      - 類似度による方法：ans21-1s.bmp
      - 距離による方法：ans21-1d.bmp
    - 最も一致度の高い領域が複数あるときは、画像の左上→右上→…→左下→右下の順番で最初に見つかった領域を解答画像とすること

※注意：画像配列を拡張しており、計算機資源が不足して実行できないため、コマンドライン上から ulimit –s unlimited を実行すること（最初の1度だけでよい）