

Etiquetagem de componentes

Após segmentação, é comum produzir-se uma imagem binária em que diversos objectos disjuntos têm *pixels* de valor 1, e se situam sobre um fundo de valor 0.

Para efeitos de análise, é importante discriminar entre os diversos objectos, atribuindo a todos os *pixels* de um objecto uma etiqueta exclusiva.

Para se conseguir essa etiquetagem (*component labeling*) podem ser usados dois algoritmos básicos:

– Algoritmo 1

- Efectuar um varrimento normal da imagem até ser encontrado um *pixel* de valor 1
- Atribuir uma nova etiqueta ao *pixel* encontrado
- Partindo desse *pixel*, propagar a etiqueta para os *pixels* vizinhos de valor 1, em todas as direcções, até esgotar o componente
- Prosseguir o varrimento, a partir do ponto onde foi interrompido, e proceder como anteriormente para novos componentes
- Terminar quando se completar o varrimento

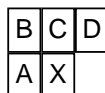
Este algoritmo tem o inconveniente de exigir o acesso aleatório aos *pixels* da imagem, podendo ser de execução bastante lenta.

Etiquetagem de componentes

– Algoritmo 2

- Efectuar um varrimento normal da imagem
- Para cada ponto 1 encontrado (X), analisar os vizinhos imediatos já visitados, como se mostra na figura

conectividade-8



conectividade-4

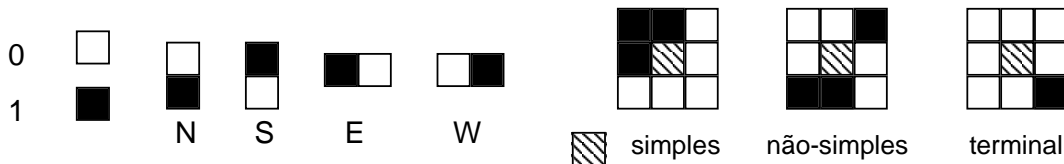


- Se todos esses vizinhos forem 0, atribuir a X uma etiqueta nova
- Caso contrário, lendo as etiquetas de A, B, C e D (ou apenas de A e B, para conectividade-4) por essa ordem, atribuir a X a primeira etiqueta diferente de 0
- Se, entre os pontos A, B, C e D houver mais do que uma etiqueta, assinalar a equivalência dessas etiquetas numa lista
- Terminado o varrimento, processar a lista de equivalências, definindo uma etiqueta única para cada grupo de etiquetas equivalentes
- Efectuar um segundo varrimento da imagem, atribuindo aos *pixels* as novas etiquetas.

Adelgaçamento e esqueletização

Como resultado da detecção de linhas ou de orlas (*thresholding* após realce dessas estruturas) é possível que estas resultem relativamente espessas. Para reduzir essas linhas a espessura unitária podem usar-se algoritmos de adelgaçamento, de que um exemplo é o seguinte:

- método NSEW: retirar sucessivamente pontos do contorno de uma região voltados (rotativamente) para N, S, E e W, sem destruir conectividade; ponto simples de um contorno é aquele para o qual o conjunto dos seus vizinhos imediatos pertencentes à região tem exactamente um componente; ponto terminal é um ponto simples com exactamente 1 vizinho na região.



- **algoritmo**: apagar todos os pontos do contorno da região a partir de uma dada direcção desde que sejam simples e não sejam terminais; repetir o processo alternando as quatro direcções NSEW; o algoritmo termina quando, num ciclo completo, nenhum ponto tiver sido apagado. (Obs.: para cada direcção, os pontos a apagar são primeiro assinalados e, depois de visitados todos os pontos, então são removidos).

Adelgaçamento e esqueletização

O adelgaçamento também pode ser aplicado a regiões, tendo em vista encontrar o seu esqueleto. A definição do esqueleto de um objecto pode fazer-se a partir da MAT (*medial axis transform*) de uma região: para cada ponto de uma região R com bordo (ou contorno) B , determinar o(s) ponto(s) mais próximo(s) em B , se houver mais do que 1 desses pontos, o ponto de R considerado pertence à MAT.

A definição anterior, válida para regiões definidas no plano contínuo, apresenta algumas dificuldades quando transposta para regiões definidas em coordenadas discretas. De entre os algoritmos para determinação da MAT de uma região, um dos mais usados utiliza a transformada de distância da região:

- **algoritmo** para cálculo da transformada de distância de uma região S (*pixels* de S com valor 1; *pixels* do complemento de S com valor 0): num primeiro passo, em rasto normal, atribuir ao *pixel* P (se pertencente a S) o valor do mínimo dos vizinhos já visitados acrescido de 1; num segundo passo, em rasto inverso, atribuir a P o mínimo entre o valor do 1º passo, e o mínimo dos vizinhos já visitados acrescido de 1; um algoritmo de reconstrução é idêntico ao anterior, mas substituindo 'mínimo dos vizinhos já visitados acrescido de 1' por 'máximo entre o valor de P e o dos vizinhos já visitados decrescido de 1'.

A MAT é o conjunto dos máximos locais (em sentido lato) da transformada de distância de uma região.

Adelgaçamento e esqueletização

Note-se que a transformada de distância, tal como definida anteriormente, pressupõe que se usam métricas de distância do tipo *cityblock* (ou *d4*) ou *chessboard* (ou *d8*):

métrica d8 (conectividade 8):



distância = 1

distância = 1

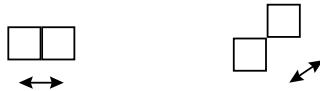
métrica d4 (conectividade 4):



distância = 1

distância = 2

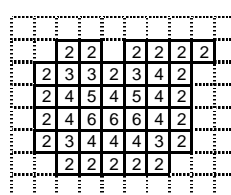
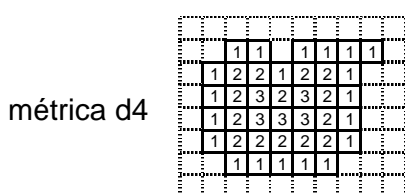
Muitas vezes usam-se métricas mais próximas da euclideana, como por exemplo a métrica 2-3:



distância = 2

distância = 3

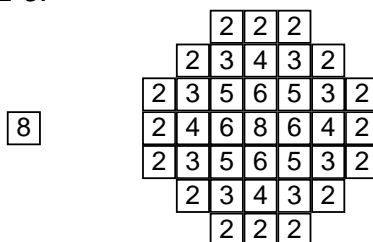
O algoritmo de cálculo da transformada de distância deverá ser adaptado para a métrica desejada; a adaptação é trivial.



Adelgaçamento e esqueletização

Geralmente requer-se que um esqueleto tenha as mesmas propriedades topológicas que a região original. Tal não acontece com a MAT. Para que a MAT possa dar origem a um esqueleto, deve proceder-se a um passo posterior à identificação dos máximos locais da transformada de distância, que consiste em acrescentar à MAT os pontos que se situam na subida mais íngreme entre os diversos segmentos que a podem constituir.

Note-se que o algoritmo de adelgaçamento NSEW produz um esqueleto com preservação da topologia da região, mas não permitindo a reconstrução da região original, o que é possível a partir da MAT (ou do esqueleto) obtido pela transformada de distância. Para isso, bastaria reconstruir cada um dos pontos que constituem a MAT e fazer a união dessas reconstruções. Como exemplo, mostra-se a reconstrução de um ponto de distância 8 na métrica 2-3.



Seguimento de contornos

As regiões de uma imagem também podem ser representadas pelo seu contorno, isto é, pelo subconjunto dos seus pontos que têm pelo menos um vizinho pertencente ao “fundo” (*background*). Para se detectar e, mais importante, para se estabelecer uma relação de ordem entre os pontos de um contorno, pode usar-se o seguinte algoritmo (*bug following*):

- **seguimento do contorno de uma região** \mathcal{S} (admitindo conectividade 8): sendo P e Q pontos adjacentes pertencentes, respectivamente, a \mathcal{S} e ao seu complemento, e definindo-se $Q=R_1, R_2, \dots, R_8$ como os vizinhos de P por ordem directa, começando em Q, as seguintes regras definem um algoritmo de seguimento:
 - 1º se for R_i o primeiro dos R's de valor 1 (isto é, de \mathcal{S}), então os novos P e Q serão $P'=R_i$ e $Q'=R_{i-1}$;
 - 2º o algoritmo termina quando se volta ao ponto inicial, desde que se encontre o ponto Q inicial (como um dos R's) antes de encontrar o próximo P'.

A sequência dos pontos P visitados pelo algoritmo acima constitui uma sequência ordenada dos pontos pertencentes ao contorno da região \mathcal{S} .

Codificação em cadeia

Quando a segmentação produz linhas (por detecção de linhas ou de orlas), estas podem ser adelgadas para representarem estruturas de espessura unitária; a simplificação pode ainda incluir a ligação (*linking*) ou a fusão (*merging*) de segmentos, de modo a reduzir o número de entidades presentes na imagem.

Por outro lado, a etiquetagem de componentes permite separar as diversas entidades (linhas constituídas por pontos contíguos).

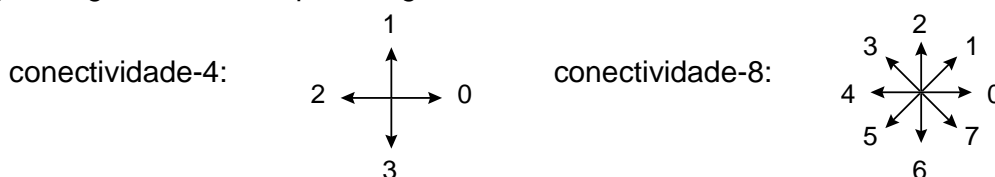
Para representar cada uma dessas entidades de uma forma compacta, pode recorrer-se ao código em cadeia (*chain-code* ou *Freeman code*). O seguimento dos pontos não envolve qualquer ambiguidade se as linhas forem delgadas; se o não forem, pode usar-se um algoritmo de seguimento de contornos como o antes referido para seguir todos os pontos exteriores da linha (neste caso codifica-se a “ida e volta”).

Para representar, de forma compacta, outras entidades espessas (como as resultantes de um algoritmo de segmentação de regiões), pode também optar-se por codificar apenas os seus contornos (exterior e interiores) por códigos de cadeia, usando-se um algoritmo de seguimento.

O código em cadeia é gerado ao mesmo tempo que se percorre a linha delgada ou se segue o contorno da região.

Codificação em cadeia

O código em cadeia representa a sequência de movimentos efectuados quando se transita de um pixel para o seguinte, necessitando de quatro símbolos quando se usa conectividade 4, e de oito símbolos quando se usa conectividade 8; os símbolos do código são geralmente os que a seguir se indicam:



Por exemplo, a linha aberta a seguir figurada é representada pelo código em cadeia (c.8) 7071075654, quando a codificação se inicia na extremidade sombreada.

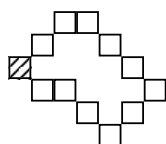


No caso de linhas fechadas, o código termina quando se retorna ao ponto inicial. A codificação descrita é absoluta, no sentido em que cada símbolo representa uma direcção constante no referencial-imagem.

Codificação em cadeia

Muitas vezes é preferível realizar-se a codificação em cadeia de modo relativo, isto é, cada símbolo representa a mudança de direcção em relação ao movimento anterior (no caso de linhas fechadas, o código pode ser realizado de forma circular, atribuindo ao primeiro símbolo a mudança relativa à transição do último ponto para o primeiro).

Ao código em cadeia expresso de forma relativa chama-se código em cadeia diferencial; para se gerar o código diferencial a partir do código absoluto, pode-se contar, no sentido directo, o número de direcções que separam dois elementos adjacentes do código absoluto, como se ilustra no exemplo seguinte (c.8):



110777553343	código absoluto
07700606017	código diferencial aberto
607700606017	código diferencial circular

Note-se que, contrariamente ao que acontece com o código absoluto, a representação por código diferencial é invariante à rotação; em ambos os casos, os códigos são invariantes à translação.

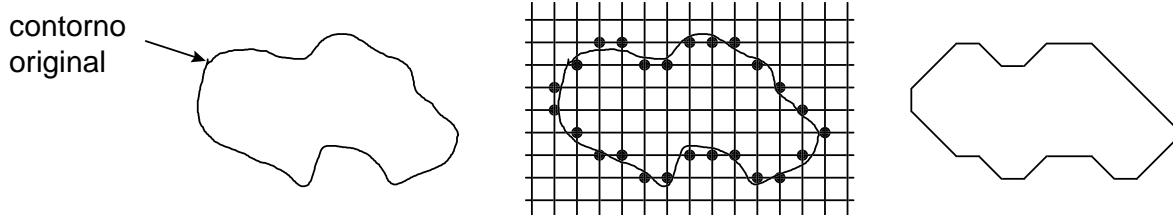
Codificação em cadeia

Tal como foram definidos, os códigos em cadeia dependem do ponto inicial escolhido (no caso de linhas abertas, há apenas ambiguidade quanto à extremidade inicial); para se conseguir invariância em relação a essa escolha, é comum normalizar-se o código em cadeia diferencial circular; a normalização consiste em rodar os símbolos do código de modo a formar-se o número inteiro menor. No exemplo anterior resultaria:

607700606017 código diferencial circular (antes da normalização)

006060176077 código diferencial circular (após normalização)

A independência do código em cadeia de um objecto relativamente à sua escala, e bem assim a maior insensibilidade ao ruído no contorno, pode conseguir-se por meio de uma sub-amostragem do contorno (ou da linha), com a vantagem adicional de o código resultar mais curto. Para o efeito, o contorno ou a linha são aproximados numa grelha de amostragem mais grosseira, como se mostra na figura seguinte:



Aproximação poligonal

A sub-amostragem antes referida tem como consequência a possibilidade de se descrever um contorno por meio de uma sequência de coordenadas de pontos (na grelha original) que são as extremidades de uma linha poligonal, a qual aproxima o contorno dado.

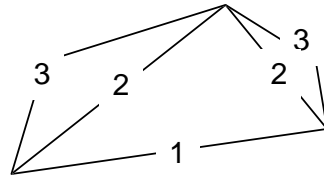
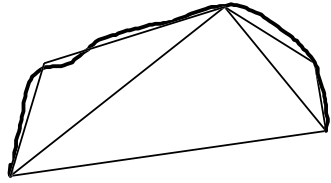
Outra forma de representação de linhas consiste justamente na sua aproximação poligonal com especificação do erro máximo tolerado para a aproximação (ϵ pixels).

Para se obter a aproximação poligonal, um dos métodos mais simples é o devido a Ramer, que se pode descrever, para linhas abertas, da seguinte forma:

- definir o segmento que une as extremidades da linha
- percorrer a linha e, para cada ponto, determinar a sua distância euclideana ao segmento
- se a máxima distância for inferior a ϵ , a aproximação é aceite; se não, selecciona-se o ponto da linha mais distante do segmento, e substitui-se o segmento pela concatenação dos sub-segmentos formados entre cada extremidade e esse ponto mais distante
- o processo anterior é repetido para cada um dos sub-segmentos determinados.

Na figura seguinte, ilustra-se o método.

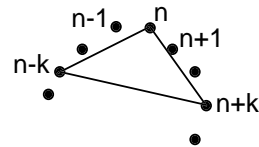
Aproximação poligonal



Muitos outros algoritmos permitem também obter aproximações poligonais de linhas, bem como aproximações por meio de representações polinomiais de grau mais elevado (representações de Bézier, *splines*, etc.).

Em qualquer tipo de aproximação, consegue-se uma representação mais eficiente quando se detectam, sobre o contorno, *pontos dominantes* (usados como extremidades nas aproximações poligonais, ou como pontos de controlo de Bézier ou de *splines*). Para extracção dos pontos dominantes pode-se determinar o valor da *curvatura* da linha em cada ponto, detectando-se aqueles que ultrapassem um certo valor limiar.

Um método simples de obter uma grandeza que aproxima a curvatura, consiste na determinação, para cada ponto n , da k -*curvatura* definida a partir do triângulo formado pelos pixels $n-k$, n e $n+k$, tal como se representa na figura ao lado (para o caso em que $k=3$).



Codificação em sequências de cor (*run-length coding*)

Já foi referido, a propósito da compressão de imagem, um método de representação que consiste na codificação dos *pixels* de uma imagem em sequências de cor uniforme, designado por *run-length*. De entre as diversas variantes possíveis, uma das mais usadas consiste na codificação separada de cada linha de imagem, que pode ser codificada por uma sequência de pares de valores: a cor (ou tom de cinzento) e o número de pixels consecutivos com a mesma cor (que se designa por *run*). Por exemplo, a linha de imagem seguinte:

12 15 15 18 18 18 12 12 20 20 20 20 24 25 25 25

pode ser codificada pela sequência de pares:

12,1;15,2;18,3;12,2;20,4;24,1;25,3

No caso de imagens binárias, como só há duas cores possíveis (0 e 1), pode-se prescindir do primeiro elemento de cada par, estabelecendo-se que cada linha começa sempre pela cor 0 (ou 1); evidentemente que, se uma linha começar pela cor 1 (ou 0), o primeiro elemento do código deverá representar o comprimento 0.

Codificação em sequências de cor (*run-length coding*)

Por exemplo, a linha de imagem seguinte:

1 1 1 0 0 0 0 0 0 1 1 1 1 0 1 1 1 1 1 1 1 0 0 0 0 0 1 1 1 1

pode ser codificada pela sequência de comprimentos (em que se convencionou que as linhas começam todas pela cor 0):

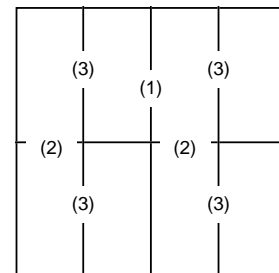
0,3,7,4,1,7,6,4

A representação por códigos *run-length* é especialmente apropriada para representar regiões em imagens binárias.

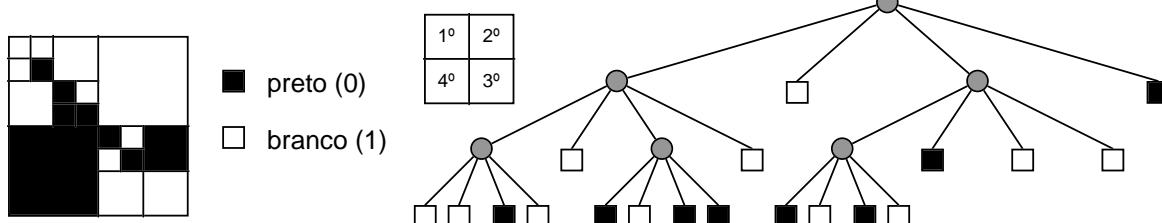
A partir da representação *run-length* é possível extrair directamente um grande número de medidas (como muitas das que são tratadas mais adiante), por meio de algoritmos designados por SRI (Stanford Research Institute). Esta é uma das mais fortes razões do sucesso que têm tido os *run-codes* em inúmeras aplicações práticas de análise de imagem; outra razão é o seu uso em transmissão de documentos, como em *facsimile*.

Árvores binárias e quaternárias

As *árvores binárias* também têm sido usadas para representação de regiões, no caso de imagens binárias; baseiam-se na subdivisão da imagem por meio de cortes alternadamente verticais e horizontais, registando-se em cada nó da árvore o seu valor (0 se a correspondente área é toda preta; 1, se for toda branca; C, se a área contiver *pixels* pretos e brancos). A figura ao lado mostra os três primeiros níveis de corte.



Muito mais frequentemente são usadas *árvores quaternárias* (*quadtrees*), que se baseiam na representação dos quadrantes e sub-quadrantes em que o espaço da imagem é sucessivamente subdividido, até ao nível em que todos os pixels de um sub-quadrante tenham a mesma cor. Mostra-se a seguir uma imagem binária, a sua subdivisão em quadrantes, e a respectiva árvore quaternária.

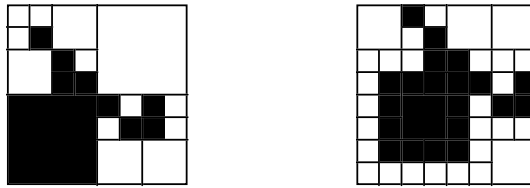


Árvores binárias e quaternárias

A representação de imagens binárias por meio de *quadrees* pode ser muito eficiente se acontecer, como é frequente, que a subdivisão em quadrantes termine a um nível ainda muito afastado dos *pixels* isolados (o que significa que um nó da árvore representa um grande número de *pixels*).

Por outro lado, muitas operações podem ser realizadas directamente sobre as *quadrees*; por exemplo, as operações lógicas entre duas imagens binárias ou a medida da área de objectos são facilmente realizadas, bastando levar-se em conta o nível da árvore em que se situa cada nó.

Contudo, a representação por quadrees é muito sensível a pequenas mudanças nas imagens; isto mesmo se pode constatar se se constroem as árvores quaternárias correspondentes às duas imagens seguintes, que apenas diferem de uma translação sofrida pelas regiões pretas.



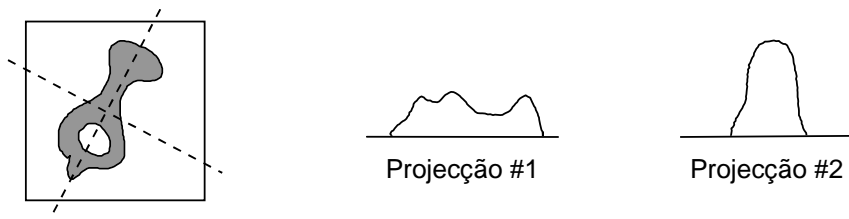
Projecções

No capítulo 2 já se referiu como é possível a reconstrução de imagem a partir de projecções, usando-se métodos tomográficos.

As projecções de uma imagem relativamente a um conjunto de direcções constituem, assim, uma alternativa para a representação de imagens; esta representação pode ser útil quando for possível usar um número reduzido de projecções (evidentemente, a reconstrução exacta da Imagem poderá ser impossível nesse caso).

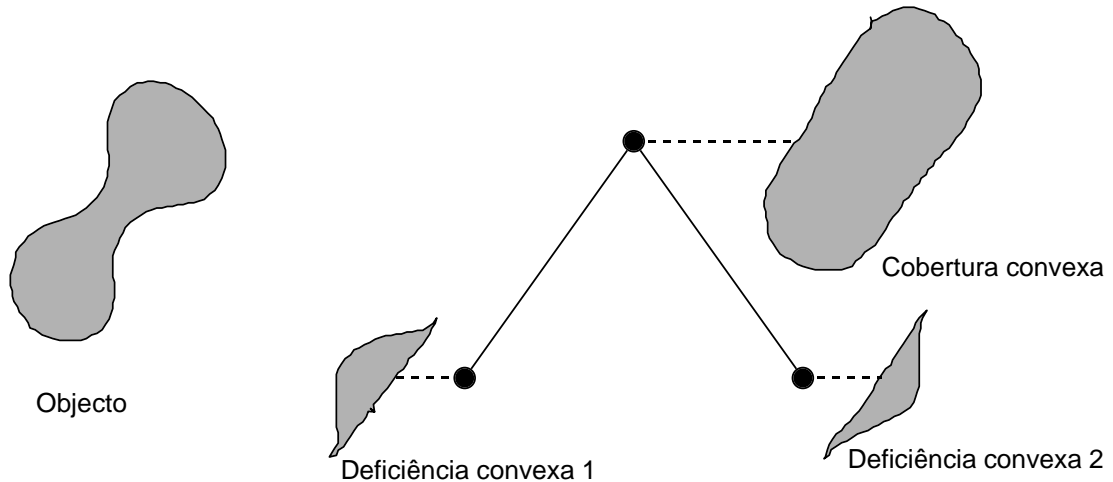
A substituição de uma representação bidimensional por um conjunto limitado de representações unidimensionais apresenta óbvias vantagens do ponto de vista da facilidade de acesso e de extracção de informação descritiva.

No caso de imagens binárias, o uso de apenas duas projecções ortogonais pode ser suficientemente descritivo dos objectos presentes. As direcções de projecção podem ser paralelas às linhas e colunas da imagem, ou então alinhadas com as direcções principais do(s) objecto(s).



Cobertura convexa / Árvore de concavidades

Muitos objectos podem ser convenientemente descritos por meio da sua *cobertura convexa* (menor figura convexa contendo integralmente o objecto) e das suas *deficiências convexas* (as partes da cobertura convexa que “sobram” quando se lhe subtrai o objecto inicial. As deficiências convexas, por sua vez, podem ser descritas de modo semelhante, construindo-se assim uma *árvore de concavidades* cuja representação pode ser de interesse para efeitos de identificação do objecto; cada nó da árvore pode ter associada a representação da figura convexa correspondente.



Extracção de características

Feita a segmentação de uma imagem, a etiquetagem e a representação dos segmentos, é necessário descrever, reconhecer, identificar ou classificar essas entidades. Para esse efeito, a extracção e a medida de características dos segmentos constitui uma fase indispensável.

De entre as inúmeras características que se podem extrair, enunciam-se de seguida algumas das mais comuns:

Área - geralmente tomada como o número de *pixels* que constituem um segmento; se tiver sido feita a calibração do sistema de aquisição de imagem, o número de *pixels* pode ser convertido para medidas de área no referencial-objecto (por exemplo, mm^2).

Perímetro - é o comprimento do(s) contorno(s) de um segmento. Pode ser medido pelo número de *pixels* que constituem o contorno, tomado quer com conectividade-4 quer com conectividade-8. Muitas vezes, para se conseguir uma melhor aproximação ao perímetro euclideano, toma-se o comprimento 1 para a distância entre *pixels* contíguos horizontalmente ou verticalmente, e o comprimento $\sqrt{2}$ para a distância diagonal. No caso de existir informação de calibração, o perímetro pode também ser convertido para unidades do referencial-objecto.

Extracção de características

“Redondez” (*roundness, compactness*) - medida baseada na relação entre área e perímetro de um objecto; trata-se de uma medida adimensional, calculada como

$$\frac{\text{Perímetro}^2}{4\pi \times \text{Área}}$$

O valor desta medida é mínimo (com o valor 1) para objectos perfeitamente circulares (em geometria contínua euclideana), tomando valores muito mais elevados para objectos alongados ou muito recortados.

Número de Euler - parâmetro topológico que, para uma imagem binária tem o valor

$$E = C - H$$

em que C é o número de segmentos conexos, e H é o número de furos nesses segmentos.

Rectângulo envolvente - é o rectângulo de lados horizontais e verticais que circunscreve o segmento; os lados e a área desse rectângulo podem ser usados como características, embora apresentem valores muito sensíveis à rotação dos objectos.

Extracção de características

Rectângulo “equivalente” - é o rectângulo de lados “*length*” e “*breadth*” que tem a mesma área e o mesmo perímetro que o objecto a que respeita. Note-se que, no caso de objectos compactos (“redondez” próxima de 1), pode não existir o rectângulo “equivalente”.

Raios máximo e mínimo - são as distâncias do centróide do objecto aos pontos mais afastado e mais próximo do contorno exterior do objecto, respectivamente. A razão entre estes dois parâmetros (raio máximo / raio mínimo) define uma característica descritiva também importante, designada por *excentricidade*.

Momentos absolutos - os momentos de uma imagem digital $f(i,j)$ são definidos por meio de

$$m_{pq} = \sum_i \sum_j i^p j^q f(i,j)$$

em que $p+q$ representa a *ordem* dos momentos.

Se a imagem for binária, os momentos podem ser calculados por meio de

$$m_{pq} = \sum_{(i,j) \in \mathcal{K}} i^p j^q$$

em que a soma só se efectua para pontos-objecto (de valor 1).

Extracção de características

Centróide - é o centro geométrico de um objecto, e as suas coordenadas podem ser calculadas a partir dos momentos absolutos (N é o número de pontos do objecto):

$$\bar{x} = \frac{1}{N} \sum_{(i,j) \in \mathcal{R}} i = \frac{m_{10}}{m_{00}} \quad \bar{y} = \frac{1}{N} \sum_{(i,j) \in \mathcal{R}} j = \frac{m_{01}}{m_{00}}$$

Momentos centrais - os momentos centrais têm uma definição idêntica aos momentos absolutos, mas usam coordenadas referidas ao centróide:

$$\mu_{pq} = \sum_i \sum_j (i - \bar{x})^p (j - \bar{y})^q f(i, j)$$

Orientação - a orientação é o ângulo formado entre a horizontal e a direcção do eixo do objecto com momento de inércia mínimo. A minimização do momento de inércia

$$I(\theta) = \sum_{(i,j) \in \mathcal{R}} D^2(i, j) = \sum_{(i,j) \in \mathcal{R}} [(j - \bar{y})\cos(\theta) - (i - \bar{x})\sin(\theta)]^2$$

permite obter o valor da orientação a partir de momentos centrais de 2ª ordem:

$$\theta = \frac{1}{2} \arctg \frac{2 \mu_{11}}{\mu_{20} - \mu_{02}}$$

Extracção de características

Rectângulo envolvente alinhado - é o rectângulo circunscrito ao objecto com o lado maior paralelo ao eixo de inércia mínima. Para o determinar pode-se percorrer o contorno exterior do objecto e, para cada ponto, determinar as suas coordenadas no referencial ortogonal centrado no centróide e com o eixo das abcissas paralelo à orientação do objecto; as coordenadas mínimas e máximas nos dois eixos do referencial determinam o rectângulo alinhado. Dois parâmetros derivados do rectângulo alinhado são também características importantes: o *alongamento* (cociente entre o lado maior e o lado menor do rectângulo), e a "*ocupação espacial*" (cociente entre a área do objecto e a área do rectângulo envolvente alinhado).

Elipse ajustada - é a elipse que tem os mesmos momentos de 2ª ordem que o objecto dado, estando portanto com o eixo maior alinhado com a orientação do objecto.