Capítulo 15

Segmentação de Imagens

Vimos que, com uso de ferramentas de **Computação Gráfica**, podemos sintetizar concepções e idéias em imagens discretas (Capítulos 2–10). **Processamento de imagens** nos provê funções de transformação entre estas imagens discretas com o objetivo de torná-las mais apropriadas para uma aplicação específica. No contexto de síntese de imagens, **pós-processamento** é muito utilizado para produzir efeitos como campo de profundidade e eliminar os efeitos de *aliasing* (Seção 9.5). Quando se trata de imagens capturadas, **pré-processamento** é amplamente aplicado com o objetivo de remover ruídos e corrigir distorções inerentes ao processo de aquisição, de realçar certos detalhes, ou de comprimir os dados para facilitar armazenamento, como vimos nos Capítulos 11–14. Melhorar a qualidade das imagens reais aumenta as chances para o sucesso de extração de unidades de informação contidas nela. Esta extração constitui a base da **Visão Computacional**. Na seção 15.1 uma noção de aquisição de **imagens de intensidade** é apresentada.

A visão computacional procura emular o processo da visão humana por meio de máquinas autônomas. Para a visão humana, uma **imagem clara** é uma imagem que permite o sistema de organização da visão agrupar rapidamente os sinais recebidos em **unidades semânticas**, inferir a estrutura lógica entre elas e capturar o seu significado. De acordo com os princípios de *Gestalt*, as regras básicas de organização da visão humana são: **similaridade**, **proximidade**, **fechamento** e **continuidade**.

(Ver Fig. 9.15, 9.16 do livro-texto de Foley.)

Na **Visão Computacional** as unidades semânticas que podem ser distinguíveis por um algoritmo são conhecidas por *features* e o processo de agrupamento de *pixels* de uma imagem digital em *features* é conhecido por

segmentação. Em geral, a segmentação automática é uma das etapas mais difíceis e mais importantes para reconhecimento (atribuição de rótulos aos objetos presentes na imagem) e interpretação dependente da aplicação (atribuição do significado aos objetos rotulados). Neste capítulo darems uma visão introdutória do processo de segementação.

Essencilamente, distinguem-se dois tipos de segmentos: **fronteira** e **regiões**. A segmentação por fronteira é adequada quando as características da fronteira, como os cantos e arestas, são de interesse. A segmentação por região, por sua vez, é aplicada quando se deseja extrair as propriedades internas do objeto, como o seu esqueleto e a sua textura. Utilizaremos os segmentos de fronteira para ilustrar o processo de segmentação.

A fronteira de um objeto aparece, usualmente, como descontinuidade em intensidades numa imagem em níveis de cinza, ou seja, como **borda de descontinuidade** de intensidade. Experimentos tem comprovado que o contorno de um objeto é um *feature* muito utilizado pela visão humana para reconhecer um objeto. Como os contornos podem ter formas variadas e muitas vezes dependente do contexto, a extração de uma borda a partir dos níveis de cinza de uma imagem é subdividida em duas etapas: deteção de arestas (Seção 15.2) e agrupamento destas arestas em fronteira (Seção 15.3). O resultado desta composição é descrito em um esquema de representação apropriado para processamentos subseqüentes. Tal esquema deve ser, preferencialmente, invariante a mudança de tamanho e a movimentos rígidos (rotação e translação). Na seção 15.4 apresentamos um exemplo de esquema de representação para a fronteira: **código da cadeia**.

Consideramos neste capítulo somente as imagens em níveis de cinza, ou seja monocromáticas, para as quais os algoritmos de segmentação mais adequados são baseados na regra de organização de fechamento/descontinuidade dos valores de níveis de cinza de seus *pixels*.

15.1 Aquisição de Imagens

Tipicamente, um sistema de aquisição de imagens digitais consiste de três módulos: uma câmera fotosensível, baseada principalmente em dispositivos de carga acoplada (*charge-coupled devices*, CCD), um digitalizador e um processador.

(Ver Fig. 1.7 do livro-texto de Gonzalez)

A entrada de uma câmera são feixes luminosos, que atravessam as suas lentes e excitam o plano de imagem. No caso de cêmeras CCD, este plano de imagem é um arranjo de $m \times n$ "fotossítios", capazes de gerar uma tensão de

saída proporcional à intensidade da luz incidente. A saída de cada fotossítio é um sinal elétrico contínuo, conhecido como **sinal de vídeo**, que pode ser lido ao escanearmos, linha por linha, o arranjo de CCD periodicamente.

(Ver Fig. 1.8 do livro-texto de Gonzalez)

O sinal de vídeo é enviado para o digitalizador, também conhecido como *frame grabber*, que digitaliza o sinal analógico num arranjo de $m \times n$ valores inteiros e os armazena na memória do computador. Observe que o resultado é equivalente ao resultado que obtivemos com o processo de síntese de imagens digitais – um arranjo de $m \times n$ valores de luminância/brilhância, representável por uma função discreta $I(u, v)$.

15.2 Detecção de Arestas

Uma **aresta** numa imagem em níveis de cinza é um pequeno conjunto de *pixels* na imagem $f(u, v)$ para o qual os níveis de cinza variam abruptamente e monotonicamente. Ela pode ser caracterizada por quatro parâmetros:

posição (u, v) do *pixel* em que a aresta está localizada.

gradiente , ou o módulo do vetor gradiante da função de intensidade $I(u, v)$,

$$\nabla I(u, v) = (\frac{\partial I}{\partial u}, \frac{\partial I}{\partial v}), \tag{15.1}$$

que é dado por

$$|\nabla I(u, v)| = \sqrt{\frac{\partial I}{\partial u}^2 + \frac{\partial I}{\partial v}^2} \tag{15.2}$$

e aproximado em

$$|\nabla I(u, v)| \approx |\frac{\partial I}{\partial u}| + |\frac{\partial I}{\partial v}|. \tag{15.3}$$

normal ou a direção do gradiente

$$\phi = tg^{-1}(\frac{\frac{\partial I}{\partial v}}{\frac{\partial I}{\partial v}}),$$

e

direção da aresta , que é perpencicular à normal ϕ.

Exercício 15.1 *Dada uma imagem de níveis de cinza em 8 bits*

20	*20*	*20*	*20*	*20*	*20*	*185*	*20*	*20*	*20*	*20*	*20*	*20*
185	*20*	*20*	*20*	*20*	*20*	*20*	*20*	*20*	*20*	*20*	*20*	*20*
20	*20*	*20*	*20*	*250*	*20*	*20*	*20*	*20*	*20*	*20*	*20*	*20*
20	*20*	*20*	*134*	*20*	*20*	*20*	*20*	*255*	*20*	*20*	*20*	*200*
20	*20*	*20*	*134*	*20*	*20*	*163*	*255*	*255*	*255*	*20*	*200*	*200*
20	*20*	*134*	*134*	*134*	*20*	*255*	*255*	*255*	*255*	*255*	*200*	*200*
20	*20*	*134*	*134*	*134*	*255*	*255*	*0*	*255*	*255*	*255*	*255*	*200*
20	*134*	*134*	*0*	*134*	*134*	*255*	*255*	*255*	*255*	*255*	*200*	*20*
20	*20*	*134*	*134*	*134*	*20*	*20*	*255*	*255*	*255*	*200*	*200*	*20*
20	*20*	*134*	*134*	*134*	*20*	*20*	*20*	*255*	*200*	*200*	*20*	*20*
20	*20*	*20*	*134*	*20*	*20*	*20*	*20*	*200*	*200*	*200*	*20*	*20*

Identifique as arestas da imagem. Determine a posição, o gradiente, a normal e a direção destas arestas.

Um detector ideal de uma aresta deve satisfazer os seguintes requisitos:

- minimizar a probabilidade de falsas detecções, confundindo os ruídos com as arestas;
- minimizar a distância entre as arestas detectadas e as arestas corretas.
- maximizar a precisão na detecção, ignorando os máximos locais falsos em torno de uma aresta correta.

Projetar um detector que satisfaça simultaneamente estes requisitos não é trivial. Os primeiros detectores de borda são baseados na magnitude do vetor dado pela Eq. 15.3. Como existem diferentes formas para discretizar as derivadas de uma função (diferenças ascendentes, diferenças descendentes, diferenças centradas e diferenças "cruzadas"), há diferentes propostas de máscaras para aproximar $|\nabla I(u,v)|$, almejando a simplicidade e a fidelidade dos resultados. Supondo uma máscara 3×3 centrada no *pixel* (u,v), cujo valor é z_5. Os valores dos seus *pixels* vizinhos-de-8 $(u-1,v-1)$, $(u-1,v)$, $(u-1,v+1)$, $(u,v-1)$, $(u,v+1)$, $(u+1,v-1)$, $(u+1,v)$ e $(u+1,v+1)$ são, respectivamente, z_1, z_4, z_7, z_2, z_8, z_3, z_6 e z_9.

z_1	z_2	z_3
z_4	z_5	z_6
z_7	z_8	z_9

O detector de arestas mais antigo é a **máscara de Roberts**, que aproxima Eq. 15.3 por soma de valores absolutos das diferenças entre as intensidades dos *pixels* vizinhos e os diferenciais por diferenças cruzadas

$$|\nabla I(u,v)| = |z_5 - z_9| + |z_6 - z_8|.$$

Os dois valores absolutos podem ser implementados com duas máscaras de convolução

1	0
0	-1

0	1
-1	0

com a imagem original, gerando duas imagens $I_1(u,v)$ e $I_2(u,v)$. A imagem de magnitude de gradiente é aproximada pelo resultado da soma destas duas imagens.

Um *pixel* (u,v) é considerado como um *pixel* de aresta, se

$$|\nabla I(u,v)| \approx I_1(u,v) + I_2(u,v) > l, \tag{15.4}$$

onde l é um valor limiar pré-definido. Para minimizar o número de falsas arestas, recomenda-se pré-processar a imagem ruidosa com um filtro passa-baixa, como o filtro gaussiano (Seção 11.3). Outra prática comum é aumentar o contraste entre os *pixels* (Capítulo 14).

O **operador de Prewitt** adota o esquema de diferenças centradas e considera todos os *pixels* vizinhos

$$|\nabla I(u,v)| = |(z_7+z_8+z_9)-(z_1+z_2+z_3)| + |(z_3+z_6+z_9)-(z_1+z_4+z_7)|.$$

Pode também ser implementado com uso de duas máscaras espaciais

$G_u =$

-1	0	1
-1	0	1
-1	0	1

$G_v =$

-1	-1	-1
0	0	0
1	1	1

que devem ser convoluídas com a imagem original para obter duas imagens $I_1(u,v)$ e $I_2(u,v)$, a partir das quais gera-se a imagem de gradiente com uso da Eq. 15.4.

O operador mais conhecido é o **operador de Sobel** que é uma variante de Prewitt com a vantagem de "suavizar"as diferenças entre as intensidades, ao invés de "suavizar"os valores das intensidades. Com isso, os *pixels* de vizinhança-de-4 tem um peso maior no cômputo do gradiente em cada *pixel*

$$\begin{aligned} |\nabla I(u,v)| &= |((z_7+z_8)-(z_1+z_2)) + ((z_8+z_9)-(z_2+z_3))| + |((z_3+z_6)-(z_1+z_4)) + ((z_6+z_9) \\ &= |(z_7+2z_8+z_9)-(z_1+2z_2+z_3)| + |(z_3+2z_6+z_9)-(z_1+2z_4+z_7)|. \end{aligned}$$

De forma análoga aos operadores anteriores, ele pode ser implementado com duas máscaras

$G_u =$

-1	0	1
-2	0	2
-1	0	1

$G_v =$

-1	-2	-1
0	0	0
1	2	1

(Ver Figs. 4.28, 4.29 e 7.6 do livro-texto de Gonzalez)

Exercício 15.2 *Determine a imagem de magnitude da imagem do Exercício 15.1 utilizando*

1. *máscaras de Roberts*
2. *operador de Prewitt*
3. *operador de Sobel*

Compare as imagens obtidas em termos da sua robustez em relação aos ruídos "sal-e-pimenta" (salt-and-pepper noise). Se pré-processarmos a imagem com um filtro Gaussiano, como seriam as imagens de magnitude do gradiente?

Observe que as operações de máscara nas bordas de uma imagem são implementadas utilizando-se as vizinhanças parciais apropriadas. Não há regras rígidas para o tratamento dos *pixels* da borda, como já comentamos oportunamente na seção 11.3.

15.3 Agrupamento de Arestas

Idealmente, as técnicas de detecção de arestas devem gerar somente *pixels* da fronteira. Na prática, o conjunto de *pixels* raramente caracteriza de forma não-ambígua a fronteira de um objeto devido aos ruídos e falhas durante o processo de aquisição. Portanto, técnicas adicionais, baseadas no princípio de similaridade, são aplicadas para conectar estes *pixels* detectados construindo uma "curva da borda". A técnica mais simples e difundida é a **transformada de Hough**.

A transformada de Hough foi originalmente proposta para detectar padrões complexos de *pixels* numa imagem binária. A idéia básica consiste em reduzir o problema de detecção de um padrão complexo em um problema de agrupamento de *pixels* da imagem que pertencem a uma curva pré-definida, transformando os pontos num sistema cartesiano para um **espaço de parâmetros** desta curva. Neste capítulo, ilustramos a técnica com as retas.

Seja a curva representada pela equação

$$y = a_A x + b_A \tag{15.6}$$

no sistema cartesiano. No plano xy, todos os pontos (x_i, y_i) desta reta são colineares e a cada ponto (x_i, y_i) tem uma reta correspondente no plano de parâmetros ab

$$b = -ax_i + y_i.$$

Estas retas são concorrentes no ponto (a_A, b_A) do plano ab. A transformação da função no espaço xy para o espaço de parâmetros ab é denominada a **transformada de Hough**. Através desta transformada, podemos agrupar num mesmo ponto do espaço de parâmetros de todos os pontos colineares no espaço cartesiano.

(Ver Figs. 7.15 e 7.16 do livro-texto de Gonzalez)

Para Eq. 15.6 a transformada de Hough pode resultar em valores de a tendendo para infinito, quando a reta é paralela ao eixo y. Para evitar este caso particular, utiliza-se as coordenadas polares na parametrização da reta

$$\rho = x\cos\theta + y\sin\theta,$$

onde ρ é a distância entre o ponto (x, y) da reta e a origem do sistema de referência e θ o ângulo entre o eixo x e o vetor (x, y).

(Ver Figs. 7.17–7.18 do livro-texto de Gonzalez)

O resultado da transformada de Hough de uma imagem $m \times n$ pode ser representado por uma imagem com os valores no domínio $[0, \sqrt{2}D] \times [-\frac{\pi}{2}, \frac{\pi}{2}]$, onde D é a distância máxima entre os vértices dos segmentos identificados. O valor de cada ponto desta imagem representa a quantidade de pontos colineares. Os máximos locais desta imagem "acumuladora" de quantidades correspondem às retas extraídas.

Exercício 15.3 *Considere as arestas identificadas no Exercício 15.2.*

1. *Determine a transformada de Hough para todas as arestas*
2. *Esboce a imagem discreta do resultado no plano $\rho\theta$.*
3. *Escreva a equação das retas identificadas.*
4. *Esboce a imagem de fronteiras extraídas.*

Vale ressaltar que a colinearidade dos pontos é determinada pela resolução do espaço de parâmetros.

15.4 Descrição de Fronteiras

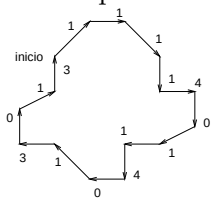
Identificadas as fronteiras dos objetos, é comum utilizar um esquema de representação conciso para descrever as informações extraídas. Nesta seção apresentamos dois esquemas de representação simples para descrever a sequência de *pixels* que pertencem a uma fronteira: **código da cadeia**, proposto por Freeman em 1974 e a representação por **assinatura**. Dentre as características desejáveis para um esquema de descrição é que ele seja **invariante** aos movimentos rígidos (translação e rotação).

15.4.1 Código da Cadeia

O código de cadeia consiste de uma sequência de dígitos, que indica a direção do *pixel* sucessor considerando a direção do *pixel* antecessor. O código do *pixel* sucessor varia com o tipo de conectividade adotado: na conectividade-de-4 são 4 dígitos (0 – para direita, 1 – para frente, 2 – para esquerda e 3 – para baixo) e na conectividade-de-8, 8 dígitos (0 – para Direita, 1 – para FD, 2 – para Frente, 3 – para FE, 4 – para Esquerda, 5 – para BE, 6 – para Baixo e 7 – para BD).

(Ver Fig. 8.1 do livro-texto de Gonzalez)

Quando a fronteira é uma curva fechada, a escolha da direção do primeiro par de *pixels* está condicionado à direção do último segmento da cadeia circular. Uma solução é adiar o preenchimento do código do primeiro segmento para o final, depois de percorrer complementamente a cadeia.



Exercício 15.4 *Escolha outros dois pixels distintos como pontos iniciais na figura anterior e determine o código da cadeia para estes dois pontos. Compare os códigos da cadeia associados aos três pontos iniciais distintos. Se rotacionarmos ou transladarmos a figura, o código da cadeia alterará? Por quê?*

Exercício 15.5 *Descreva as fronteiras extraídas no Exercício 15.3 com uso do código da cadeia.*

Observação 15.1 *Uma boa interodução ao código da cadeia pode ser encontrada em* `http://chaos.mind.ilstu.edu/curriculum/perception/chaincode1.html`.

15.4.2 Descrição por Assinatura

Segundo Gonzalez e Woods, uma **assinatura** é uma representação funcional unidimensional de uma fronteira, podendo ser gerada de diversas maneiras. Uma das mais simples é dada pelo gráfico da distância r da fronteira ao centróide em função do ângulo θ. A função $r(\theta)$ é uma assinatura de uma fronteira.

(Ver Fig. 8.5 do livro-texto de Gonzalez)