

Capítulo 2

Segmentação de Imagens

[Este texto corresponde ao capítulo 2 da dissertação de mestrado de Roberto Hirata Jr., intitulada “Segmentação de Imagens por Morfologia Matemática”, defendida em 1997 no IME-USP.]

Neste capítulo faremos um apanhado geral dos métodos mais utilizados em segmentação de imagens, que aqui chamamos de **clássicos**, em contraposição aos métodos de segmentação introduzidos pela Morfologia Matemática.

O objetivo disto é criar um vocabulário básico e introduzir as técnicas clássicas mais importantes no âmbito da segmentação de imagens. Não pretendemos com isso fazer um trabalho fechado, no sentido que dispense o estudo de outras referências. Nem pretendemos fazer uma análise comparativa das técnicas para concluir que certa técnica é melhor do que uma outra. Como veremos, a adequação de uma técnica é extremamente dependente do problema que se está resolvendo. Por fim, queremos com este pequeno resumo dar subsídios para que se perceba, ao final do trabalho, que a Morfologia Matemática é uma abordagem equivalente à abordagem clássica.

É raro o processo de extração automática de informações de imagens que não utilize segmentação. Assim, o resultado da segmentação de uma imagem pode determinar o sucesso, ou o fracasso, de toda a análise daquela imagem. É por isso que alguns cuidados gerais que devem ser tomados desde a aquisição da imagem, como por exemplo, utilização de luz com direção e intensidades controladas e “hardware” de aquisição de boa qualidade, tornam-se ainda mais importantes quando se pretende segmentar a imagem.

Existem diversas opiniões a respeito do que é segmentar uma imagem [HS85, CA79, Kan80, RD79, YB89, Hal79, GW92, Pra91], mas não existe nenhuma teoria sobre o assunto. Uma consequência disto é que existem muitas técnicas operacionais e computacionais para resolver os diversos problemas desta área. No entanto, não é difícil depararmos com casos em que a imagem não possui informações suficientes para ser segmentada sem o uso de um conhecimento prévio a respeito dos objetos que estão sendo segmentados [Mar82]. Um exemplo de uma imagem que contém objetos assim é a vista na fig. 2.1. As ampliações desta imagem (fig. 2.2 e fig. 2.3) mostram duas células na parte superior que estão juntas e só sabemos que são duas comparando com o formato e tamanho das outras células na imagem. Também, é fácil encontrar um problema de segmentação para o qual os métodos clássicos existentes até o momento não sejam satisfatórios para resolvê-lo.

Antes de apresentar os métodos de segmentação, vamos estabelecer algumas notações úteis que usaremos daqui para frente.

2.1 Notações Básicas

Seja \mathbf{E} um conjunto de pontos no plano¹, onde os pontos são dados pelas suas coordenadas. Por questões práticas, o conjunto \mathbf{E} é normalmente definido como um subconjunto retangular de $\mathbb{Z} \times \mathbb{Z}$, por exemplo, $[a, b] \times [c, d] \subset \mathbb{Z} \times \mathbb{Z}$.

Um **grafo de conectividade**² [Ser82, RP66, KR89] G é um par ordenado de conjuntos (\mathbf{E}, D) , onde \mathbf{E} é um conjunto de pontos e D é um subconjunto de pares não ordenados de pontos de \mathbf{E} , denominado conjunto de arestas.

¹Restringiremos nosso estudo aos conjuntos definidos em espaços de uma ou duas dimensões, denotados por 1D e 2D, respectivamente. A extensão de vários dos conceitos e definições encontrados neste trabalho para espaços de dimensão maior não é difícil.

²também conhecido como grade de conectividade

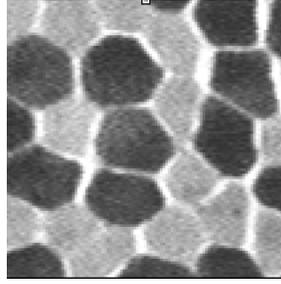


Figura 2.1: Fronteira duvidosa

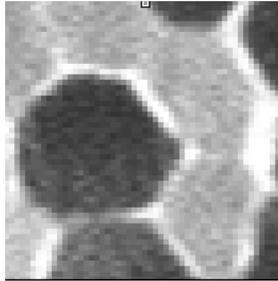


Figura 2.2: Primeira ampliação

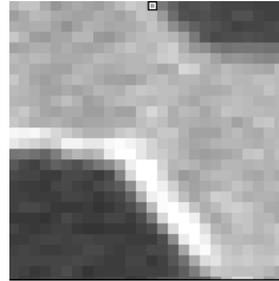


Figura 2.3: Segunda ampliação

Uma aresta $\{x, y\} \in D$ junta os pontos x e y e define uma relação que denominaremos de **vizinhança** entre x e y . Isto é, se $\{x, y\} \in D$, então x e y são pontos **vizinhos** [Ser82, RP66, KR89].

Se (g, h) são as coordenadas de um ponto qualquer $x \in \mathbb{E}$ e se, segundo G , x tem como vizinhos os pontos $(g + 1, h)$, $(g - 1, h)$, $(g, h + 1)$ e $(g, h - 1)$ (fig. 2.4) então \mathbb{E} é dito ser 4-adjacente (fig. 2.5).

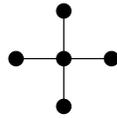


Figura 2.4: Um ponto e seus 4 vizinhos

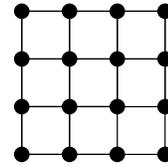


Figura 2.5: Espaço 4-adjacente

Se além desses vizinhos, x tem como vizinhos os pontos $(g + 1, h + 1)$, $(g + 1, h - 1)$, $(g - 1, h + 1)$ e $(g - 1, h - 1)$ (fig. 2.6), então \mathbb{E} é dito ser 8-adjacente (fig. 2.7).

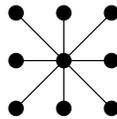


Figura 2.6: Um ponto e seus 8 vizinhos

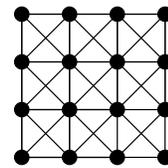


Figura 2.7: Espaço 8-adjacente

Dependendo de como \mathbb{E} for definido, nem todos os vizinhos de um ponto $x \in \mathbb{E}$ precisam existir. Por exemplo, se \mathbb{E} for definido como o subconjunto $[a, b] \times [c, d] \subset \mathbb{Z} \times \mathbb{Z}$, então nem todos os vizinhos de x existem se $x = (g, h)$, quando $g = a$ ou $g = b$ ou $h = c$ ou $h = d$.

Outros tipos de grades de conectividade podem ser definidas sobre os pontos de \mathbb{E} [KR89, Ser82]. Por exemplo, um tipo muito utilizado na “École Nationale Supérieure des Mines de Paris” é a grade 6-adjacente (fig. 2.9, onde cada ponto tem 6 vizinhos (fig. 2.8)). Neste trabalho vamos nos restringir apenas aos conjuntos \mathbb{E} 4-adjacentes ou

8-adjacentes.

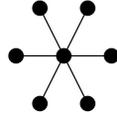


Figura 2.8: Um ponto e seus 6 vizinhos

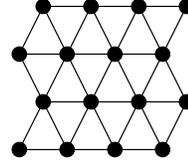


Figura 2.9: Espaço 6-adjacente

Sejam p e q dois pontos de \mathbf{E} . Um **caminho** C de p a q é uma seqüência de pontos de \mathbf{E} , $C = \{p_0, p_1, \dots, p_n\}$, tal que os extremos de C , p_0 e p_n são, respectivamente, os pontos p e q , além disso, p_i é vizinho de p_{i+1} , $\forall i \in [0, n-1]$.

O comprimento do caminho C de p a q é igual ao número de pontos da seqüência menos um, $|C| = n - 1$.

Uma maneira de definir a **distância** entre p e q , denotada $d(p, q)$, é o comprimento do menor caminho de p a q . Existem outras maneiras de definir a distância entre dois pontos de \mathbf{E} [Ser82], mas não as veremos aqui.

Um subconjunto, ou **região**, $X \subset \mathbf{E}$ é conexo³, segundo G , se $\forall p, q \in X$ existe um caminho de p a q consistindo inteiramente de pontos contidos em X .

Seja $x \in \mathbf{E}$, $X \subset \mathbf{E}$ e $d(x, y)$ uma distância. A distância entre o ponto x e um subconjunto X , sob d , é dado pelo valor:

$$d(x, X) = \min\{d(x, y) : y \in X\}$$

Um ponto de \mathbf{E} que esteja associado ao valor 0 na imagem é um **ponto preto** ou **ponto de fundo**. Um ponto de \mathbf{E} que esteja associado a um valor diferente de 0 na imagem é um **ponto branco**. O conjunto de todos os pontos pretos da imagem formam o fundo da imagem.

Define-se a **borda interior**, ∂X , de um conjunto $X \subset \mathbf{E}$, como sendo o conjunto dos pontos $p \in X$ tal que p tem pelo menos um vizinho que não pertence a X .

Duas regiões X e Y são **adjacentes** se existem $x \in X$ e $y \in Y$, tal que x e y são vizinhos.

O **complemento** de uma região $X \subset \mathbf{E}$ é denotado por \overline{X} .

Dada uma região conexa $X \subset \mathbf{E}$, um **buraco** na região X é uma região $Y \subset \mathbf{E}$, adjacente a X , tal que não existe um caminho $C \subset \mathbf{E}$, totalmente contido em \overline{X} , que ligue um ponto $p \in Y$ com um ponto $q \in \overline{X}/Y$.

Denota-se por $K^{\mathbf{E}}$ o conjunto de todas as funções $f : \mathbf{E} \rightarrow K$, onde $K = [0, k]$ é um subintervalo fechado do conjunto \mathbb{Z} .

Dessa forma, uma imagem em níveis de cinza pode ser representada por uma função⁴ $f \in [0, k]^{\mathbf{E}}$, onde $k \in \mathbb{Z}$, e $k > 1$.

Da mesma forma, uma imagem binária, i.e., uma imagem na qual cada ponto só pode assumir dois valores 0 e k pode ser representada por uma função $f \in \{0, k\}^{\mathbf{E}}$.

Se os conjuntos $K_1^{\mathbf{E}}$ e $K_2^{\mathbf{E}}$ representam conjuntos de imagens, um operador $\Psi : K_1^{\mathbf{E}} \rightarrow K_2^{\mathbf{E}}$ será chamado de um **operador entre imagens**.

Definição 2.1.1 *Seja $f \in K^{\mathbf{E}}$ uma imagem definida em \mathbf{E} e P um predicado lógico de segunda ordem [HDW94] sobre $X \subset \mathbf{E}$. Segmentar a imagem f [GW92, Har92] é encontrar uma partição $\Omega = \mathbf{E}_1, \mathbf{E}_2, \dots, \mathbf{E}_n$ de \mathbf{E} tal que:*

1. $\bigcup_{i=1}^n \mathbf{E}_i = \mathbf{E}$
2. $\mathbf{E}_i \cap \mathbf{E}_j = \emptyset \quad \forall i, j \in [1, n], i \neq j$
3. $P(\mathbf{E}_i)$ é verdadeiro $\forall i \in [1, n]$
4. $P(\mathbf{E}_i \cup \mathbf{E}_j)$ é falso, $\forall i, j \in [1, n], (i \neq j)$

³A conexidade por caminhos é uma possível definição de conexidade para subconjuntos de \mathbf{E} , mas existe uma outra definição topológica que não depende da definição de caminho, apenas da definição de adjacência, veja [KR89]

⁴A constante 0 provém do fato dela estar associada a um dos níveis de cinza da imagem, em geral, o preto.

As condições 1 e 2 definem uma partição usual de um conjunto. Em 1 assegura-se que todo ponto da imagem pertence a algum elemento da partição, o que é razoável pois uma segmentação deve ser completa. Em 2 garante-se que nenhum ponto da imagem pode estar em mais de um elemento da partição ao mesmo tempo, ou ainda, nenhum ponto pode satisfazer a critérios diferentes ao mesmo tempo.

As condições 3 e 4 especificam Ω no que tange à segmentação. Em 3 impõe-se que os pontos de cada elemento \mathbf{E}_i da partição Ω satisfaçam o predicado lógico P , i.e., que cada \mathbf{E}_i satisfaça a algum critério definido pelo predicado. Note que P é um predicado definido sobre conjuntos, mais especificamente sobre regiões conexas, e não sobre um certo elemento do conjunto, i.e., é um predicado de segunda ordem, como em Lógica costuma-se chamar. Em 4 impõe-se que se tomarmos a união de duas regiões distintas, $\mathbf{E}_i \cup \mathbf{E}_j$, então \mathbf{E}_i e \mathbf{E}_j têm de ser diferentes com relação ao predicado P . Por exemplo, tomemos uma imagem com objetos de diversas cores. Vamos supor que queremos segmentar os objetos azuis e os objetos vermelhos da imagem. Seja P o predicado: (Os pontos de \mathbf{E}_i são todos vermelhos) ou (Os pontos de \mathbf{E}_i são todos azuis) ou (Os pontos de \mathbf{E}_i são diferentes de vermelho e azul), onde Ω é uma partição de \mathbf{E} e $\mathbf{E}_i \in \Omega$. Uma partição $\mathbf{E}_1, \mathbf{E}_2$ e \mathbf{E}_3 , tal que \mathbf{E}_1 seja formado por todos os pontos vermelhos da imagem, \mathbf{E}_2 por todos os pontos azuis da imagem e \mathbf{E}_3 por todos os outros pontos da imagem, satisfaz 1 pois contém todos os pontos da imagem, satisfaz 2 pois nenhum ponto pode ter duas cores simultaneamente, satisfaz 3 pois o predicado é verdadeiro para cada \mathbf{E}_i e satisfaz 4 pois se unirmos quaisquer dos elementos da partição teremos um novo elemento com pontos de pelo menos uma das cores, azul ou vermelho, o que torna falso o predicado para esse elemento.

Pela definição 2.1.1, dada uma imagem e um certo problema de segmentação temos dois problemas a resolver:

- Formular um predicado lógico que possibilite através das informações disponíveis na imagem a solução do problema.
- Achar um operador entre imagens, que pode ser composto por vários outros, que particione o espaço \mathbf{E} segundo as regras do predicado, de preferência eficientemente.

Existem várias maneiras de encontrar esses operadores. A maior parte baseia-se nas descontinuidades, ou semelhanças, entre os níveis de cinza dos pontos da imagem. Outros baseiam-se em características como, por exemplo, cor, tamanho, forma, textura e etc. Vamos ver a seguir algumas técnicas que já se tornaram de grande importância em Processamento de Imagens e que, por isso, são conhecidas como clássicas.

Trataremos na maioria das vezes com imagens de duas dimensões, mas a teoria aplica-se muitas vezes a imagens de qualquer dimensão inteira.

2.2 Métodos Clássicos de Segmentação

Vamos mostrar a seguir um pequeno resumo de algumas das técnicas e métodos clássicos existentes, começando com as que se baseiam em similaridades, como o “threshold”⁵ e passando para as que se baseiam em dissimilaridades, como o gradiente.

2.2.1 Threshold

Um dos operadores mais utilizados para segmentação de objetos que possuem características semelhantes em termos de luminosidade é o **threshold**,

$$T : K_1^{\mathbf{E}} \longrightarrow \{0, k_2\}^{\mathbf{E}}.$$

Dada uma imagem $f \in K_1^{\mathbf{E}}$, o objetivo deste operador é atribuir o mesmo rótulo ($k_2 \in \mathbb{Z}$) para todos os pontos $p_i \in \mathbf{E}$ tais que $f(p_i)$ satisfaça a um certo critério.

Threshold Simples

Existem vários tipos de threshold, o mais simples, e de maior uso, depende de apenas um parâmetro h , normalmente chamado de **limiar** ou mesmo de **threshold**, definido por:

⁵Neste texto, todas as palavras em um idioma estrangeiro serão grafadas entre aspas “ ”, entretanto, esta palavra será utilizada daqui para frente sem as aspas, por ser muito comum.

$$T_h(f)(x) = \begin{cases} k & \text{se } f(x) \leq h \\ 0 & \text{c.c.} \end{cases}$$

É fácil ver que o operador threshold não é inversível e que neste caso seu resultado é uma imagem binária que vale k apenas nos pontos onde o valor da função f é menor que h .

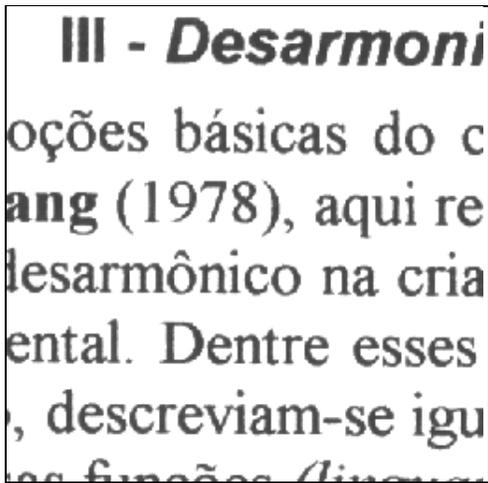


Figura 2.10: Imagem extraída de um texto

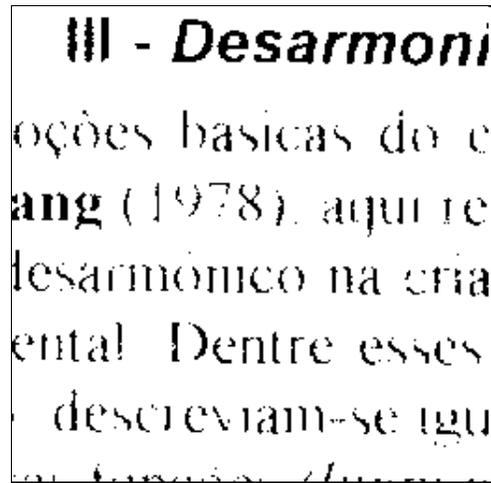


Figura 2.11: Parâmetro $h_2 = 85$

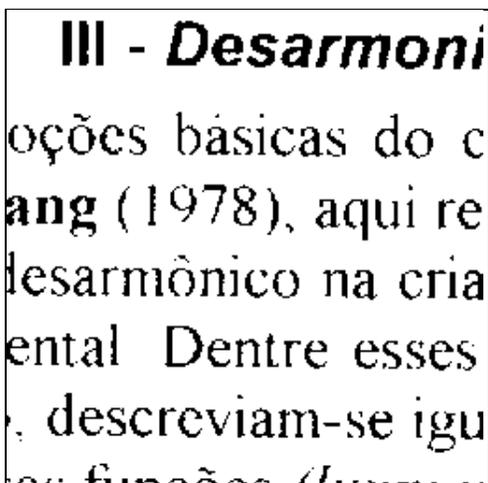


Figura 2.12: Parâmetro $h_2 = 100$

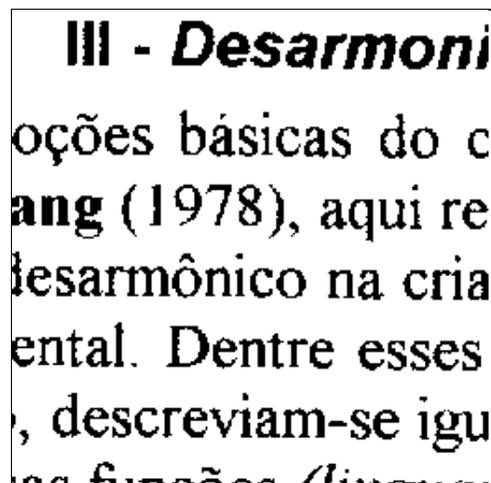


Figura 2.13: Parâmetro $h_2 = 150$

Threshold Múltiplo

Este operador é chamado de **threshold múltiplo** pois admite dois limiares h_1 e h_2 , e é definido por:

$$T_{h_1, h_2}(f)(x) = \begin{cases} k & \text{se } h_1 \leq f(x) \leq h_2 \\ 0 & \text{c.c.} \end{cases}$$

No caso que $h_1 = 0$, o threshold múltiplo é equivalente ao threshold simples.

A imagem da fig. 2.10, mostra um texto digitalizado e as imagens fig. 2.11, fig. 2.12, fig. 2.13, fig. 2.14 e fig. 2.15 mostram diversos exemplos do resultado do operador threshold múltiplo aplicado sobre esse texto⁶. Os parâmetros

⁶Devido ao tipo de impressão utilizado neste trabalho, não é fácil perceber que as letras na imagem em níveis de cinza têm o interior bem escuro e a intensidade dos níveis vai aumentando em direção às bordas, i.e., as bordas das letras são mais claras.

utilizados no threshold (h_1 e h_2) estão grafados nas legendas das figuras, com exceção das vezes que $h_1 = 0$; nestes casos ele não aparece.

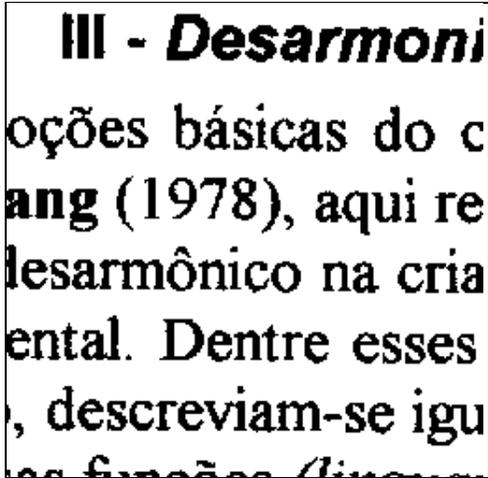


Figura 2.14: Parâmetro $h_2 = 200$

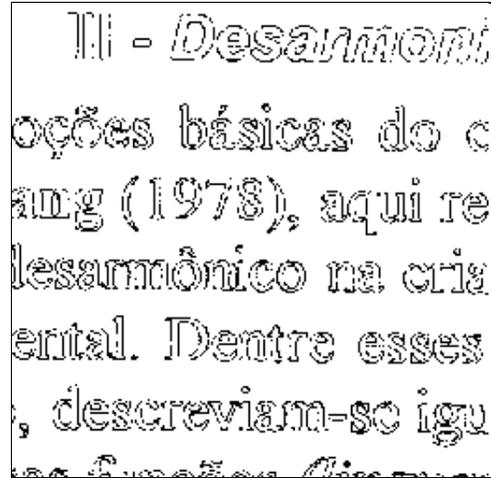


Figura 2.15: Parâmetros $h_1 = 100$ e $h_2 = 200$

Normalmente é difícil encontrar os parâmetros h_1 e h_2 que se ajustem corretamente para toda a imagem nessa variante. No entanto, o operador é útil quando queremos segmentar regiões cujos pontos estejam em uma determinada faixa de intensidades.

Threshold Adaptativo

O operador de **threshold adaptativo** (Adaptative Threshold [Wat72]) é uma generalização do anterior. A idéia aqui é achar parâmetros diferentes para cada parte da imagem, i.e., achar duas funções f_1 e $f_2 \in K^E$, $f_1(x) < f_2(x) \forall x \in E$, que sirvam de limiares. A fórmula geral é dada por:

$$T_{f_1, f_2}(f)(x) = \begin{cases} k & \text{se } f_1(x) \leq f(x) \leq f_2(x) \\ 0 & \text{c.c.} \end{cases}$$

Critérios para o estabelecimento de limiares

Uma maneira para estabelecer limiares para a segmentação de imagens é a análise visual dos níveis de cinza dos pontos dos objetos que queremos segmentar. Por exemplo, a maioria dos pontos da imagem da fig. 2.10 correspondentes às letras têm níveis de cinza abaixo de 150. Um threshold simples com limiar $h = 150$ é suficiente para segmentar todos os caracteres da imagem.

O uso criterioso de operadores de threshold requer, normalmente, o cálculo do histograma⁷ dos pontos da imagem de acordo com os seus níveis de cinza para o encontro dos limiares corretos. Isto torna a segmentação mais precisa e automática do que quando o mesmo processo é feito por uma amostragem a partir da inspeção visual da imagem.

Quando o histograma da imagem possui um vale como o da figura 2.16, pode-se escolher o limiar como sendo o valor mínimo da curva do histograma. Mas, existem técnicas mais elaboradas para o encontro de limiares [Rus91].

O histograma em conjunto com a limiarização não resolve todos os problemas de segmentação pois não leva em consideração, por exemplo, a forma dos objetos na imagem, i.e., dois objetos de formatos diferentes podem ser indistinguíveis usando-se esta técnica.

Os problemas do mundo real são bem mais complexos e, normalmente, o threshold é usado em conjunto com outros operadores. Além disso, ele é reservado como um dos últimos operadores a se aplicar na imagem, dada a grande perda de informações que acontece após a sua aplicação.

⁷O histograma dos níveis de cinza de uma imagem é um gráfico cujo eixo das abscissas representa os níveis de cinza da imagem e o eixo das ordenadas representa a frequência com que cada nível aparece na imagem.

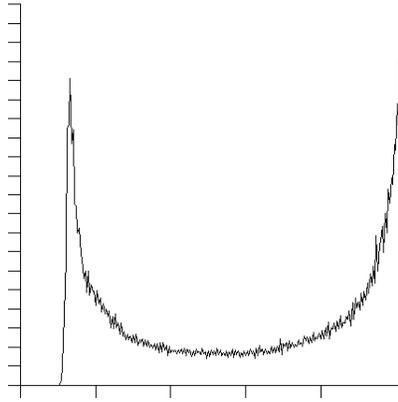


Figura 2.16: Histograma de uma imagem em níveis de cinza

Finalmente, é importante mencionar que os algoritmos que implementam tanto o operador de threshold como o cálculo do histograma têm complexidade linear com relação ao número de pontos da imagem.

2.3 Crescimento de Regiões

Seja $\{\mathbf{E}_i\}$ uma família de subconjuntos de \mathbf{E} cujos pontos são escolhidos de acordo com a imagem $f \in K^{\mathbf{E}}$ a ser segmentada. Inicialmente os subconjuntos são denominados **sementes**, pois é a partir deles que começa o processo. Após o início eles são chamados de **regiões**. Crescimento de regiões é o nome que se dá à técnica de aumentar o número de pontos de cada semente, unindo pontos a ela segundo algum critério de similaridade. O processo termina quando não houver mais pontos em \mathbf{E} que satisfaçam o critério de similaridade. O conjunto das regiões após o término do processo, mais o seu complemento em relação a \mathbf{E} , formam uma partição desse espaço.

O crescimento das regiões se faz a partir dos pontos vizinhos da borda das regiões e de um certo critério de semelhança cuja forma mais simples é a diferença absoluta entre os níveis de cinza dos pontos da borda das regiões e seus vizinhos [Har92, HS85].

Na literatura encontramos vários outros critérios para formar regiões e cada um deles resulta em um operador diferente entre funções. Os operadores simples, i.e., baseados apenas num critério de vizinhança, podem dar resultados ruins com maior frequência, pois uma região A pode ligar-se a uma outra região B devido apenas à diferença existente entre um ponto da borda da região A e seu vizinho na região B.

Uma forma de agrupamento que evita esse problema foi introduzida por Brice e Fenema [BF70]. A técnica pode ser resumida assim:

- Na primeira etapa do processo, pares de pontos vizinhos são ligados para formar um conjunto, que os autores chamaram de **regiões atômicas**, se eles têm o mesmo valor de cinza. Em outras palavras, se eles têm a mesma intensidade e são vizinhos, segundo a grade G , então eles irão formando regiões conexas.
- Na segunda etapa, que eles chamaram de **heurística da fagocitose**, regiões maiores devem absorver as regiões menores segundo o seguinte critério:

Sejam P_1 e P_2 os comprimentos das bordas das regiões $R_1 \subset \mathbf{E}$ e $R_2 \subset \mathbf{E}$, respectivamente (fig. 2.17). Seja $|I|$ o comprimento da fronteira I entre R_1 e R_2 . Seja $|W|$ o comprimento das partes “fracas” da fronteira, i.e., o comprimento de $W \subset I$ onde a diferença absoluta entre os níveis de cinza de R_1 e R_2 através da fronteira é menor do que um certo ϵ_1 . Desta forma, duas regiões, R_1 e R_2 , são ligadas se:

$$\frac{|W|}{\min\{|P_1|, |P_2|\}} > \epsilon_2 \text{ onde } \epsilon_2 = \frac{1}{2}, \text{ usualmente.}$$

Se ϵ_2 é pequeno, duas regiões podem juntar-se com facilidade. Se, ao contrário, ele é grande, só quando uma região praticamente envolver a outra é que elas serão agrupadas.

- Na terceira etapa, regiões adjacentes cuja fronteira é fraca, i.e., têm vários pontos cuja diferença absoluta entre os níveis de cinza através da fronteira é menor do que ϵ_1 , são ligadas segundo o seguinte critério:

$$\frac{|W|}{|I|} > \epsilon_3, \text{ onde } \epsilon_3 = \frac{3}{4}, \text{ usualmente.}$$

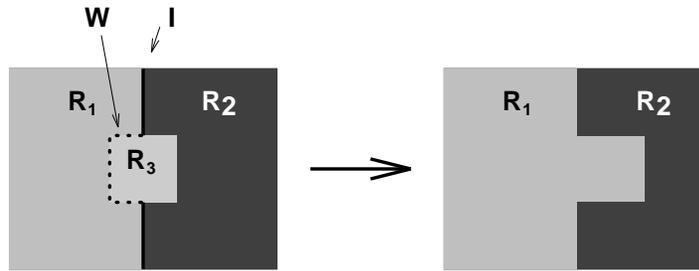


Figura 2.17: Dinâmica do Crescimento de Regiões

Esta técnica de Brice e Fenena produz resultados satisfatórios, segundo a literatura clássica, desde que as imagens não sejam muito complexas, i.e., não tenham muitos objetos a serem segmentados e possuam pouca textura.

2.4 “Split & Merge”

O método de “Split & Merge” (**S&M**) é a composição dos métodos de separar (“Split”) e de juntar (“Merge”). O primeiro consiste em, a partir da imagem inteira, ir separando os pedaços em quartos (quartas partes) se eles não forem suficientemente homogêneos. Este processo é feito até um certo número de regiões pré-estabelecido, ou até que não seja mais possível separá-las (devido à região ter ficado menor que um certo limite ou se tornado um ponto). O segundo consiste em, a partir de uma imagem dividida em pedaços, ir juntando as regiões que forem parecidas.

Para controlar eficientemente este processo, é usada uma estrutura de dados do tipo árvore “quadtree” (i.e., uma árvore na qual cada nó que não seja folha têm quatro filhos).

A imagem inteira é a raiz da árvore, os descendentes são obtidos separando (“Split”) a imagem em quatro partes, sucessivamente; os ascendentes são obtidos juntando-se (“Merge”) quatro nós da árvore (não necessariamente do mesmo pai) de acordo com os critérios de homogeneidade estabelecidos.

As técnicas de **S&M** podem ser computacionalmente “pesadas” se o processo de separação for iniciado pela imagem inteira. Por outro lado, se o processo começar com a árvore excessivamente separada (ramificada), tal que as folhas são os pontos da imagem, podem ocorrer erros devido às medidas da uniformidade de uma região estarem baseadas, inicialmente, apenas em pontos vizinhos. Assim, o **S&M** inicia, normalmente, num estágio intermediário entre estes dois extremos.

Vários autores relatam também que as imagens segmentadas por este método podem ter segmentos com formato ligeiramente retangulares, devido às regras de divisão e junção [Pra91, Har92, Hal79].

2.5 “Clustering” ou Aglomeração

Uma técnica semelhante à de crescimento de regiões é o “**clustering**”, ou **aglomeração** [DH73, Pra91, DV84]. A diferença principal entre essas técnicas é que aquela (crescimento de regiões) atua sobre o espaço de pontos da imagem, i.e., o espaço **E**, e esta atua sobre um espaço conhecido como **espaço de atributos**.

Atributo é o nome que se costuma dar a uma medida feita sobre uma imagem. Normalmente, ele é relativo às propriedades ou às medidas de determinados conjuntos de padrões. Essas medidas podem ser relativas a tamanho de objetos na imagem, desvio padrão ou média dos níveis de cinza da imagem ou de regiões da imagem, número de objetos em uma certa região, níveis de cinza de vários espectros de luz, componentes de um sistema de cores (RGB, HSB, etc) e etc.

A imagem da fig. 2.18 mostra um exemplo de como coletar um atributo usando uma janela de tamanho 1×3 pontos com centro no ponto do meio. Se p for o ponto do centro da janela, um exemplo de atributo no ponto p é a média dos níveis de cinza dos pontos vistos pela janela.

Um outro exemplo de atributos que podem ser tomados de maneira semelhante são os níveis de cinza dos pontos da imagem vistos por uma janela quando centrada no ponto p . No caso acima, para cada ponto p da imagem teremos 3 atributos (W_1, W_2 e W_3), cujos valores são os níveis de cinza dos pontos da janela.

Um espaço de atributos é um sistema cartesiano em que cada eixo representa um atributo diferente. Dada uma imagem, toma-se um vetor de atributos $\mathbf{u} = (u_1, u_2, u_3, \dots, u_n)$ para cada ponto da imagem.

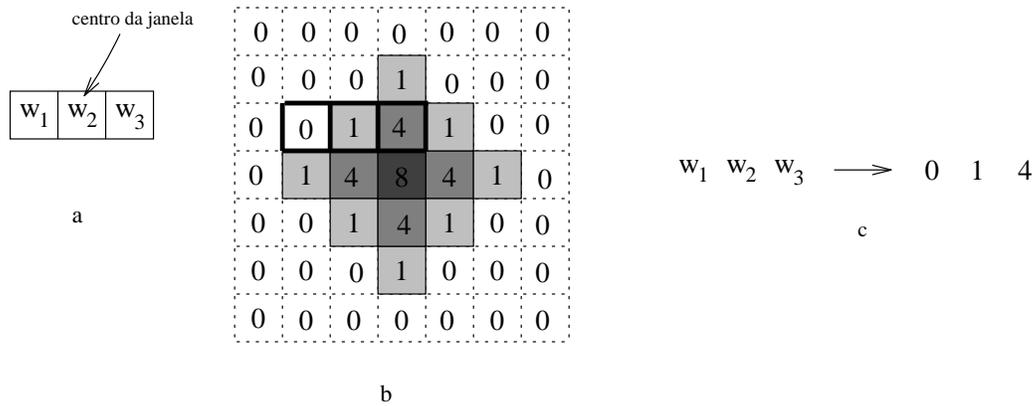


Figura 2.18: Dinâmica da coleta de configurações

No espaço dos atributos vão estar representados N pontos, X_1, X_2, \dots, X_N . O problema de aglomeração é um problema de particionamento do espaço de atributos em um número K de subconjuntos disjuntos. A idéia por detrás da técnica é que os pontos de um certo objeto ou padrão da imagem devem ter características semelhantes e o conjunto desses pontos formariam, então, um aglomerado (“cluster”), como mostrado na fig. 2.19. Há várias formas de fazer esta aglomeração dos pontos [DH73]; a mais usada parte de um número pequeno de “clusters” (cujos centros são escolhidos dentre os pontos da imagem) e a cada etapa o número de centros vai aumentando com o objetivo de diminuir as distâncias⁸ entre os pontos do aglomerado e os novos centros de aglomeração.

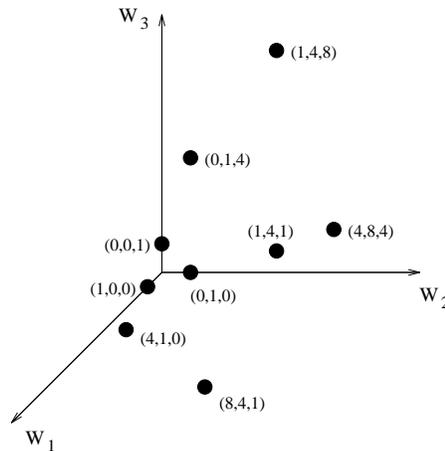


Figura 2.19: Representação de aglomerados em um espaço de atributos

Um dos principais problemas do método é saber o número ótimo de “clusters” para que o resultado da segmentação seja bom. Vamos apresentar a seguir uma técnica de dois estágios, desenvolvida por Coleman e Andrews [CA79], que resolve esse problema. No primeiro estágio são computados os vetores de atributos. No segundo estágio, é determinado o número ótimo de núcleos, seus centros e os pontos que os compõem, segundo um critério de proximidade dos centros de aglomeração.

O algoritmo usado para isso tenta, primeiramente, aglomerar todos os N vetores em torno de 2 centros (que podem ser escolhidos ao acaso dentre os N vetores), segundo a distância do vetor ao centro mais próximo dele. Após isso, o número de centros vai aumentando de 1 e um fator de qualidade β vai sendo computado a cada iteração até que

⁸A distância neste caso pode ser a distância usual do espaço cartesiano.

ele atinja um valor máximo (que será o número ótimo de aglomerados). A cada próxima etapa, o centro do novo aglomerado a ser criado será o ponto que possui a maior distância ao centro do aglomerado, dentre os aglomerados atuais.

O cálculo do fator de qualidade β é feito da seguinte forma. Sejam

- N o número de pontos a serem aglomerados
- K o número de aglomerados atual
- M_k o número de pontos do aglomerado A_k (k -ésimo aglomerado)
- \mathbf{u}_i um vetor de características, $1 \leq i \leq N$
- \mathbf{u}_0 o vetor médio de todos os vetores, dado por $\mathbf{u}_0 = \frac{1}{M} \sum_{i=1}^M \mathbf{u}_i$
- \mathbf{u}_k a média dos vetores de A_k
- S_k o conjunto dos elementos de A_k
- \mathbf{S}_B e \mathbf{S}_W , duas matrizes esparsas (externa e interna, respectivamente) dadas por:

$$\mathbf{S}_W = \frac{1}{K} \sum_{k=1}^K \frac{1}{M_k} \sum_{x_i \in S_k} [\mathbf{u}_i - \mathbf{u}_k][\mathbf{u}_i - \mathbf{u}_k]^T$$

$$\mathbf{S}_B = \frac{1}{K} \sum_{k=1}^K [\mathbf{u}_k - \mathbf{u}_0][\mathbf{u}_k - \mathbf{u}_0]^T,$$

onde u^T é o vetor transposto de u . Finalmente,

$$\beta = tr[\mathbf{S}_W]tr[\mathbf{S}_B],$$

onde tr é o traço da matriz.

Após aglomerados, os diversos “clusters” formam segmentos da imagem.

O “clustering” é uma conhecida técnica de reconhecimento de padrões, mais especificamente, uma técnica de “aprendizado” não supervisionado [DH73] que é usada para a solução de diversos problemas em Processamento de Imagens. A parte heurística de um processo de segmentação por “clustering” consiste na escolha das medidas que serão feitas na imagem. O restante do processo é, geralmente, estatístico.

Até agora vimos as técnicas clássicas baseadas em semelhanças entre os objetos ou padrões da imagem. A partir de agora veremos as técnicas clássicas que são baseadas nas dissimilaridades entre eles. Os objetos que podem ser localizados por estas técnicas podem ser de três tipos: pontos, linhas e arestas.

Os operadores usados neste caso são, normalmente, definidos por uma operação de vizinhança. Seja $\mathbf{N}_G(\mathbf{x})$ o conjunto dos pontos vizinhos do ponto x segundo o grafo de conectividade G . Vamos apresentar os operadores para o caso em que \mathbf{E} é 8-adjacente⁹ e vamos dar índices a esses pontos segundo a imagem da fig. 2.20. Os operadores são do tipo (seja $f \in K^E$):

$$\Psi(f)(x) = \sum_{i=1}^9 w_i f(x_i) \tag{2.1}$$

onde $x_5 = x$, $x_i \in \mathbf{N}_G(\mathbf{x})$ ($i \neq 5$) e w é um vetor de pesos dados aos pontos para diferenciá-los. O conjunto dos pesos w_i também é conhecido como **máscara detectora** e pode ser representado por uma matriz 3×3 do tipo:

⁹A adaptação desses operadores para outras conectividades pode ser realizada facilmente.

w_1	w_2	w_3
w_4	w_5	w_6
w_7	w_8	w_9

Figura 2.20: Máscara

2.6 Detecção de pontos isolados

O conceito de ponto isolado em uma imagem binária é bem simples e a sua segmentação, mais ainda. Chamamos de **ponto isolado** um ponto $p \in \mathbf{E}$ tal que $f(p) = 1$ e $f(q) = 0, \forall q \in \mathbf{N}_{\mathbf{G}}(\mathbf{p})$.

Para segmentar um ponto isolado, primeiramente usa-se um operador local que o realce em relação aos outros. O operador definido pela equação 2.1 com os pesos w_i dados por $w_5 = 8$ e $w_i = -1, 1 \leq i \leq 9, i \neq 5$, aplicado à imagem binária $f(x) \in [0, 1]^{\mathbf{E}}$ resulta em uma imagem em níveis de cinza onde todo ponto isolado vale 8 (i.e., $\Psi(f)(x) = 8$, caso x seja um ponto isolado) e $\Psi(f)(x) < 8$, caso contrário.

Para completar a segmentação destes pontos, basta aplicar um threshold que segmente os pontos cujo valor seja igual a 8.

Pontos isolados em imagens de níveis de cinza tem uma conceituação um pouco diferente; dizemos que um ponto é isolado quando seu valor difere consideravelmente de seus vizinhos e não, simplesmente, quando todos os seus vizinhos são nulos. O operador que será aplicado à imagem pode ser o mesmo que é aplicado para imagens binárias, mas o threshold dependerá da diferença de níveis de cinza entre o ponto e seus vizinhos que considerarmos suficiente para tratá-lo como isolado.

2.7 Detecção de retas

Da mesma forma que um ponto isolado pode ser segmentado por um operador do tipo definido pela equação 2.1, também pode ser um segmento de reta, desde que o vetor de pesos usado seja adequado para realçar as partes de uma reta. Para cada direção possível das retas que se deseja realçar na imagem deve-se considerar um vetor de pesos para realçá-la. Por exemplo, utilizando a representação de matrizes para os pesos w_i , se quisermos realçar uma reta horizontal, devemos usar a matriz da figura 2.21 e para realçar uma reta a 45° devemos usar a matriz da figura 2.22.

-1	-1	-1
2	2	2
-1	-1	-1

Figura 2.21: Matriz de pesos

-1	-1	2
-1	2	-1
2	-1	-1

Figura 2.22: Matriz de pesos

2.8 Detecção de arestas ou bordas

Dá-se o nome de **aresta** ou **borda** aos pontos de uma imagem pertencentes à fronteira entre regiões que tenham níveis de cinza diferentes entre si. Do Cálculo Diferencial sabemos que os operadores diferenciais, por exemplo o gradiente, são adequados para detectar mudanças na inclinação de curvas ou superfícies [Apo58].

Dada uma função $f : \mathbb{R}^2 \rightarrow \mathbb{R}^2$, o gradiente de f é definido pelo vetor:

$$\nabla f(x, y) = \left(\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y} \right)$$

Geometricamente, ele é um vetor que aponta na direção de maior crescimento de f no ponto (x, y) e seu módulo (que é a infomação importante na detecção de arestas) é proporcional à intensidade do crescimento de f .

Para uma imagem digital $f \in K^{\mathbf{E}}$, o módulo do gradiente pode ser aproximado pelo operador [GW92],

$$|\nabla f| \simeq \left| \frac{\partial f}{\partial x} \right| + \left| \frac{\partial f}{\partial y} \right| \simeq [(w_7 + 2w_8 + w_9) - (w_1 + 2w_2 + w_3)] + [(w_3 + 2w_6 + w_9) - (w_1 + 2w_4 + w_7)]$$

Os termos entre colchetes são denominados **Operadores de Sobel**. Eles podem ser vistos na forma de tabelas na fig. 2.1.

Existem outras formas de aproximar o cálculo do gradiente, mas os operadores de Sobel são os que dão os resultados visualmente mais suaves.

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

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

Tabela 2.1: Operadores de Sobel

2.9 Qualidade de uma Segmentação

Um sistema de extração automática de informações deve ter, também, um módulo automático de avaliação da qualidade da imagem segmentada. Encontrar e implementar um bom critério de avaliação do resultado de uma segmentação de imagens não é simples. No entanto, algumas regras já tornaram-se conhecidas e, em geral, são seguidas[Har92, HS85]:

1. Regiões de uma imagem segmentada devem ser uniformes e homogêneas com respeito a algum critério, tal como níveis de cinza ou textura.
2. O interior de uma região segmentada E_i deve ser simples (em geral côncavo) e não deve conter muitos buracos pequenos. As imagens das fig. 2.23 e fig 2.24 ilustram como deve, ou não, ser o interior de uma região segmentada, respectivamente.
3. Regiões adjacentes a uma região segmentada devem ser significativamente diferentes com respeito à característica de uniformidade dos pontos na região segmentada. Por exemplo, se a característica importante é a textura, esta deve ser significativamente diferente de uma região para outra região segmentada.
4. As fronteiras de cada região devem ser finas (em espessura), de preferência não regulares, e precisas em relação aos objetos que estão sendo segmentados.

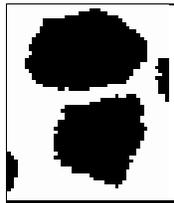


Figura 2.23: Segmentação boa



Figura 2.24: Segmentação ruim

Neste capítulo vimos um resumo de alguns operadores mais importantes para a segmentação de imagens, que já tornaram-se clássicos.

Nos capítulos seguintes vamos tratar o assunto de segmentação sob a abordagem de uma poderosa teoria algébrica, a Morfologia Matemática.

Referências Bibliográficas

- [Apo58] Tom Apostol. *Mathematical Analysis*. Addison-Wesley, 1958.
- [BF70] C. R. Brice and C. L. Fenema. Scene analysis using regions. *Artificial Intelligence*, 1:205–226, 1970.
- [CA79] Guy B. Coleman and Harry C. Andrews. Image segmentation by clustering. In *Proceedings of the IEEE*, volume 67, May 1979.
- [DH73] R. O. Duda and P. E. Hart. *Pattern Classification and Scene Analysis*. John Wiley and Sons, 1973.
- [DV84] Nelson Delfino D’Avila and Flavio R. D. Velasco. *Processamento Digital de Imagens*. IME-USP, 1984.
- [GW92] R. C. Gonzalez and R. E. Woods. *Digital Image Processing*. Addison-Wesley Publishing Company, 1992.
- [Hal79] Ernest L. Hall. *Computer Image Processing and Recognition*. Academic Press, 1979.
- [Har92] Robert Haralick. *Encyclopedia of Artificial Intelligence*, volume 2, chapter Segmentation, pages 1473–1491. John Wiley and Sons, Inc., 1992.
- [HDW94] J.Flum H.-D.Ebbinghaus and W.Thomas. *Mathematical Logic*. Springer Verlag, 1994.
- [HS85] Robert M. Haralick and Linda G. Shapiro. Image segmentation techniques. *Computer Vision, Graphics and Image Processing*, 29:100–132, 1985.
- [Kan80] Takeo Kanade. Region segmentation: Signal vs. semantics. *Computer Vision and Image Processing*, (13):279–297, 1980.
- [KR89] T. Y. Kong and A. Rosenfeld. Digital topology: Introduction and survey. *Computer Vision, Graphics and Image Processing*, (48):357–393, 1989.
- [Mar82] David Marr. *Vision*. W.H.Freeman, 1982.
- [Pra91] W. K. Pratt. *Digital Image Processing*. John Wiley and Sons, 1991.
- [RD79] Azriel Rosenfeld and Larry S. Davis. Image segmentation and image model. In *Proceedings of the IEEE*, volume 67, May 1979.
- [RP66] A. Rosenfeld and J. L. Pfaltz. Sequential Operations in Picture Processing. *Journal of the Association for Computing Machinery*, pages 471–494, 1966.
- [Rus91] John C. Russ. *Computer-Assisted Microscopy*. Plenum Press, New York, 1991.
- [Ser82] J. Serra. *Image Analysis and Mathematical Morphology*. Academic Press, London, 1982.
- [Wat72] S. Watanabe. *Frontiers of Pattern Recognition*, chapter Boundary Detection of Radiographic Images by a Thresholding Method. Academic Press, Inc, 1972.
- [YB89] S.D. Yankowitz and A.M. Bruckstein. A new method for image segmentation. *Computer Vision and Image Processing*, 46(1):82–95, April 1989.