

FUNDAMENTOS DE PROCESSAMENTO DE IMAGEM DIGITAL

1 Ementa e Bibliografia

OBSERVAÇÃO: Todo material apresentado em sala de aula se encontra em <http://www.dcc.unicamp.br/~cpg/material-didatico/mo815/9802/Welcome.html>

2 Motivação

O interesse em processamento de imagem digital surgiu de duas áreas:

- Interpretação humana usando melhoramento de imagens.
- Percepção de máquina através da extração de informações das imagens de forma tratável pelo computador.

Uma das primeiras aplicações na primeira categoria foi o melhoramento de imagens de jornal transmitidas via cabo submarino entre London e New York. Em 1920, o tempo requerido para esta transmissão foi reduzido de mais que 1 semana para menos que três horas graças ao sistema Bartlane. Melhoramentos continuaram, mas os avanços expressivos em processamento de imagem digital se iniciaram apenas em 1964 com os adventos do computador digital em larga escala e do programa espacial, através do processamento de imagens da lua. Hoje, processamento de imagem digital encontra diversas aplicações em: medicina, biologia, astronomia, automação industrial, engenharia, geologia, agronomia, etc.

3 Objetivos do curso

Introduzir conceitos sobre processamento de imagem digital que serão exercitados através do desenvolvimento de algumas aplicações envolvendo imagens médicas e vídeo digital.

4 Representação de imagem digital

Uma imagem monocromática é uma função de intensidade de luz bidimensional $f(x, y)$, onde x e y denotam coordenadas espaciais e o valor de f no ponto (x, y) é proporcional ao brilho (ou nível de cinza) da imagem neste ponto (ver Figura 1). Esta função também pode ser vista como uma superfície no espaço (x, y, z) , onde para cada ponto (x, y) plotase na coordenada z o valor de $f(x, y)$ (obs: abordagem muito utilizada em morfologia matemática.).



Figura 1: Imagem monocromática da Lenna.

4.1 Modelo de imagem

A intensidade de luz $f(x, y)$ pode ser modelada da seguinte forma:

- $f(x, y) = i(x, y)r(x, y)$ (iluminação do ambiente e reflectância dos objetos), onde
- $0 < f(x, y) < \infty$, $0 < i(x, y) < \infty$ e $0 < r(x, y) < 1$.

4.2 Amostragem e quantização

Para gerar uma imagem digital, $f(x, y)$ deve ser digitalizada ao longo de x e y , e na amplitude $z = f(x, y)$. Para tanto é feita uma amostragem (normalmente uniforme) de $f(x, y)$ nas direções x e y , gerando uma matriz de $N \times M$ amostras, seguida de uma

quantização do valor de $f(x,y)$ em L níveis inteiros de cinza (ver Figura 2). Nesta matriz, cada elemento $p(x,y), x = 0, 1, \dots, M - 1$ e $y = 0, 1, \dots, N - 1$, é chamado *pixel* (i.e. uma abreviação de *picture elements*). Dizemos então que a imagem tem dimensão M pixels na horizontal (eixo x) e N pixels na vertical (eixo y). O conceito de dimensão de um pixel ao longo do eixo x , ou do eixo y , está relacionado com o espaçamento físico entre as amostras. Cada pixel tem associado um valor $L_{min} \leq p(x,y) \leq L_{max}$, onde $L = L_{max} - L_{min} + 1$. Doravante, vamos assumir sem perda de generalidade que $L_{min} = 0$.

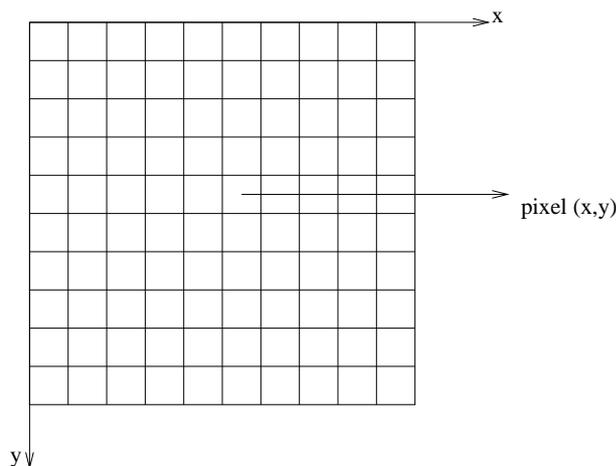


Figura 2: Representação de uma imagem monocromática digital.

A Figura 3 ilustra a representação matricial da imagem da Figura 1 em uma região de interesse de 10×10 pixels (à direita) em torno de um ponto indicado sobre o olho da Lenna (à esquerda).

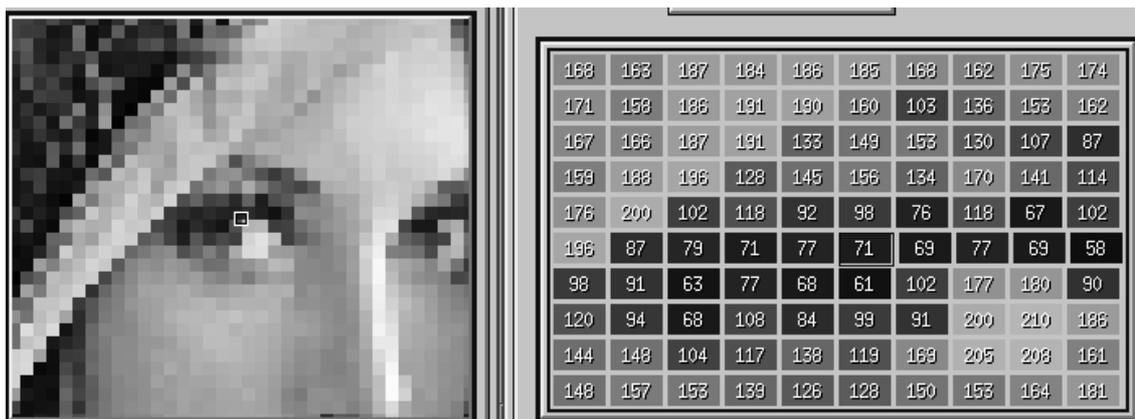


Figura 3: Ponto indicado sobre o olho da Lenna (à esquerda). Matriz de pixels em uma região de interesse de 10×10 pixels em torno do ponto indicado (à direita).

Considerando que o processo de digitalização envolve parâmetros de amostragem e quantização, uma pergunta natural é quantas amostras $N \times M$ e níveis de cinza L são

necessários para gerar uma boa imagem digital?

Isto depende fundamentalmente da quantidade de informação contida na imagem e do grau de detalhes desta informação que é perceptível pelo olho humano. Estes parâmetros nos levam aos conceitos de resolução espacial e profundidade da imagem.

4.2.1 Resolução espacial e Profundidade da imagem

Considere, por exemplo, uma imagem $f(x, y)$ contida em uma região retangular de 30cm em x por 20cm em y . Se obtivermos amostras uniformemente espaçadas a cada 1mm em x e em y (dimensões do pixel são $1\text{mm} \times 1\text{mm}$), teremos $N = 200 \times M = 300$ amostras, ou seja, 60000 pixels. Dizemos então que a *resolução espacial* da imagem é 200×300 pixels. O número L de níveis de quantização da função $f(x, y)$ é normalmente uma potência de 2 (i.e. $L = 256, 1024, 4096$). Digamos que neste exemplo $L = 256$. Isto significa que cada pixel pode ter associado um valor de cinza entre 0 e 255, que requer no máximo 8 bits para ser armazenado na memória do computador. Dizemos então que a *profundidade da imagem* é 8 bits por pixel (ou 1 byte por pixel). Podemos então observar que necessitamos $200 \times 300 \times 1 = 60\text{kbytes}$ de memória para armazenar esta imagem.

Figura 4 mostra um experimento de variação de resolução espacial com uma imagem digital da Lenna. A Figura 4a mostra uma imagem de 256×256 pixels e 256 níveis de cinza. Figuras 4b e 4c mostram o resultado de reduzir a resolução espacial de 256×256 para 128×128 e 64×64 pixels, respectivamente. Em todos os casos o número de níveis de cinza permanece 256. Para manter a mesma área de display da Figura 4a, os pixels das imagens de mais baixa resolução são replicados. Note a degradação quadriculada sofrida pelas imagens devido a perda de resolução espacial (i.e. o pixel começa a ser perceptível).

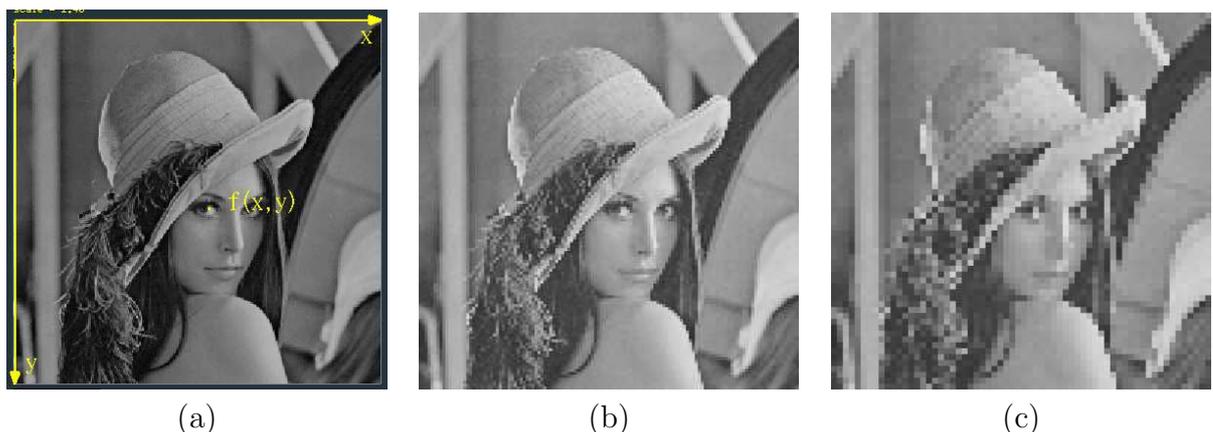


Figura 4: Imagens da Lenna de 256 níveis de cinza: (a) 256×256 pixels, (b) 128×128 pixels e (c) 64×64 pixels.

Figura 5 mostra um experimento de variação de profundidade com a mesma imagem da Lenna. Em todos os casos a resolução espacial é a mesma da imagem da Figura 4a,

256 × 256 pixels. As Figuras 5a, 5b e 5c mostram respectivamente uma redução de profundidade de 8 bits por pixel ($L = 256$ níveis de cinza) para 4 bits por pixel ($L = 16$ níveis de cinza), para 3 bits por pixel ($L = 8$ níveis de cinza) e para 2 bits por pixel ($L = 2$ níveis de cinza ou imagem binária). Note que os detalhes da imagem original vão dando lugar a regiões homogêneas de níveis de cinza.

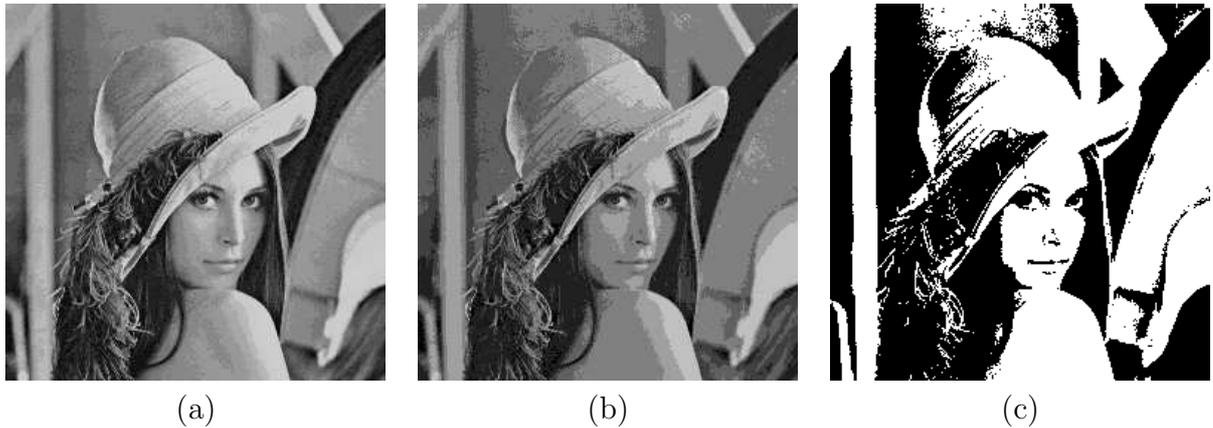


Figura 5: Imagens da Lenna de 256 × 256 pixels: (a) 16 níveis de cinza, (b) 8 níveis de cinza e (c) 2 níveis de cinza (imagem binária).

4.3 Imagem Digital Multibanda

Em uma imagem digital monocromática, o valor do pixel é um escalar entre 0 e L . Imagens multibandas podem ser vistas como imagens nas quais cada pixel tem associado um valor vetorial $p(x, y) = (l_1, l_2, \dots, l_n)$, onde $0 \leq l_i \leq L_i - 1$ e $i = 1, 2, \dots, n$. Aqui vale a observação que em algumas aplicações l_i pode ser um valor negativo, mas sem perda de generalidade vamos sempre assumir 0 como o valor mínimo de intensidade. Em geral, l_i pode representar grandezas diferentes, tais como, temperatura, pressão, frequência, amostradas em pontos (x, y) e com intervalos de valor completamente diferentes. No entanto, se $L_i, i = 1, 2, \dots, n$, for igual a 256, por exemplo, teremos uma imagem com profundidade n bytes por pixel (Figura 6a).

Uma outra forma de representar uma imagem multibanda é como uma sequência de imagens monocromáticas (i.e. as bandas) $p_i(x, y) = l_i, 0 \leq l_i \leq L_i - 1$ e $i = 1, 2, \dots, n$ (Figura 6b). Neste caso, se $L_i, i = 1, 2, \dots, n$, for igual a 256, teremos n bandas com profundidade 1 byte por pixel (Figura 6b).

Neste curso, vamos adotar a segunda forma de representação.

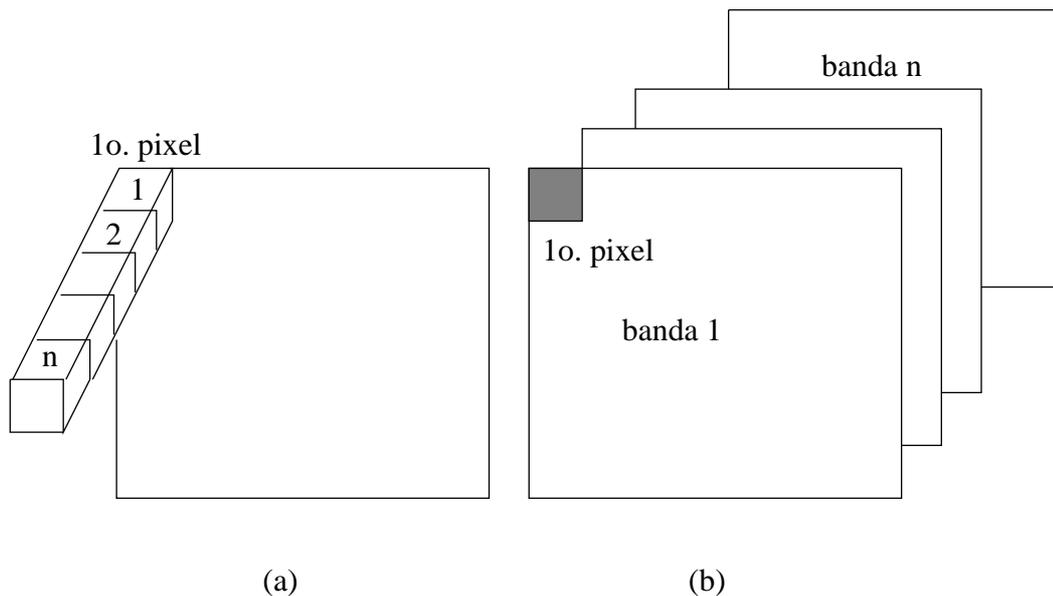


Figura 6: (a) uma imagem com n bytes por pixel (b) n bandas com um byte por pixel.

4.3.1 Imagem Colorida

Uma imagem colorida é uma imagem multibanda, onde a cor em cada ponto (x,y) é definida através de três grandezas: luminância, matiz e saturação. A luminância está associada com o brilho da luz, a matiz com o comprimento de onda dominante e a saturação com o grau de pureza (ou intensidade) da matiz. A maioria das cores visíveis pelo olho humano pode ser representada como uma combinação de três cores primárias: vermelho (R), verde (G) e azul (B). Assim, uma representação comum para uma imagem colorida utiliza três bandas R, G, e B com profundidade 1 byte por pixel (ou considerando a primeira forma de representação, temos uma imagem com profundidade 24 bits por pixel). A Figura 7a mostra uma imagem RGB da Lenna, e as Figuras 7b, 7c e 7d mostram as bandas R, G, e B, respectivamente.

Uma imagem colorida também pode ser armazenada usando uma imagem monocromática e um mapa de cores. Neste caso, o valor de cinza de cada pixel na imagem é um índice para uma célula do mapa de cores e a célula do mapa de cores contém o valor dos componentes R, G e B referentes a cor do pixel (Figura 8). A quantidade de células do mapa de cores determina o número de cores utilizado para representar a imagem. Monitores pseudocolor, por exemplo, utilizam 256 cores para mostrar as imagens coloridas.

Outras formas de representação de uma imagem colorida são discutidas na seção 8.



Figura 7: (a) Imagem RGB-24bits da Lenna (b) componente R (c) componente G e (d) componente B.

4.4 Imagem Digital Multidimensional

Até o momento, discutimos os casos de amostragem e quantização no espaço 2D. Portanto, o que comumente chamamos imagem é uma imagem digital bidimensional. Se a amostragem e a quantização envolvessem apenas uma direção do espaço teríamos o caso particular de uma imagem digital unidimensional com tratamento similar ao dado para sinais digitais. Neste curso, estamos interessados em imagens digitais tridimensionais, ou seja, em uma extensão dos conceitos de imagem digital monocromática e multibanda para uma terceira dimensão que pode ser espaço ou tempo. Isto é o mesmo que dizer que a amostragem e a quantização podem ocorrer em (x, y, z) ou (x, y, t) , onde x, y, z representam o espaço e t o tempo. Portanto, uma imagem digital 3D será representada como uma seqüência de imagens monocromáticas ou multibandas ao longo do eixo espacial z ou do

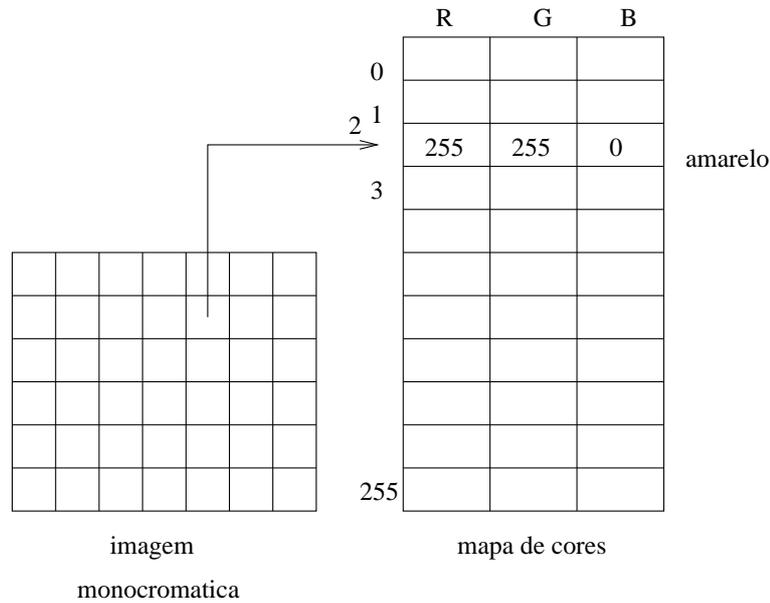


Figura 8: Uso de mapa de cores para imagem colorida.

eixo temporal t .

4.4.1 Imagem Tomográfica

Equipamentos tomográficos, (e.g. tomógrafos de raios-X, ressonância magnética) geram imagens monocromáticas de cortes (ou fatias) normalmente paralelos e uniformemente espaçados em uma dada região 3D (Figura 9a). Em medicina, por exemplo, estes cortes são normalmente longitudinais cobrindo uma dada região do corpo do paciente que contém um dado órgão/fenômeno em estudo. Considerando as dimensões $p \times p$ de um pixel nestas imagens e o espaçamento d entre os cortes, a extensão do pixel em 3D forma um pequeno paralelepípedo de dimensões $p \times p \times d$ que é chamado voxel (i.e. *volume element* - Figura 9b). O centro dos voxels coincide com o centro dos pixels. Os voxels representam pontos de amostragem de algum fenômeno físico e são usados para reconstruir no computador a forma ou função de estruturas tridimensionais. Neste caso, a imagem pode ser representada como uma matriz 3D como ilustrado na Figura 9c.

Imagens tomográficas possuem normalmente 512×512 ou 256×256 pixels e profundidade 1 ou 2 bytes por pixel. A Figura 10, por exemplo, mostra três fatias de ressonância magnética de resolução espacial 256×256 pixels e profundidade 2 bytes por pixel ($L = 1106$).

Como foi dito anteriormente, os voxels são usados para reconstruir a forma ou função de estruturas 3D. A Figura 11 ilustra uma reconstrução de um crânio a partir de imagens de tomografia de raios-X.

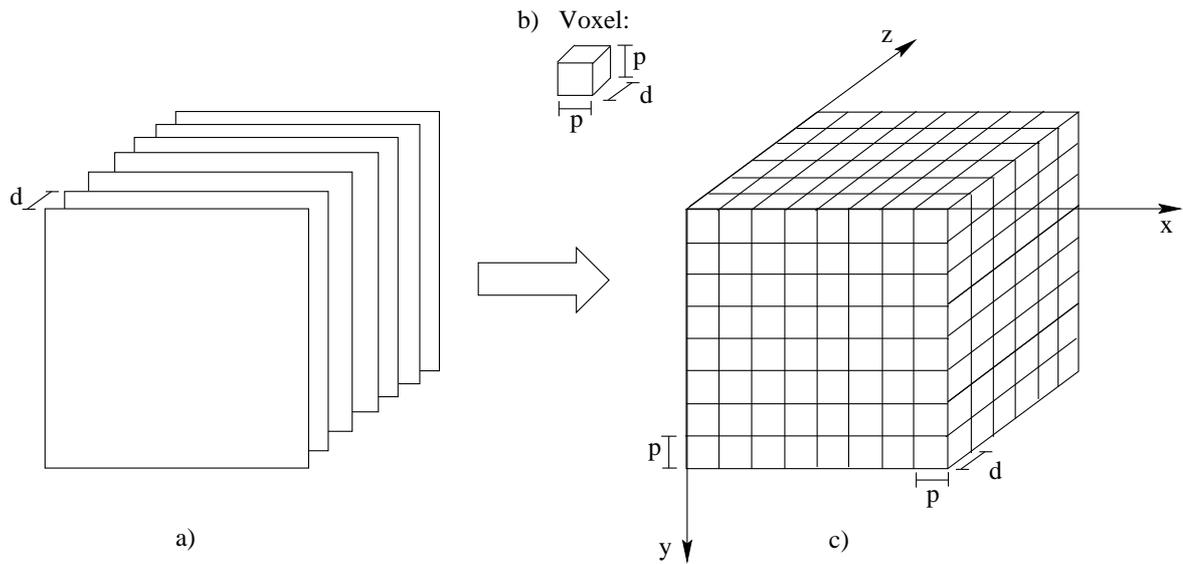


Figura 9: Representação da imagem através de voxels.

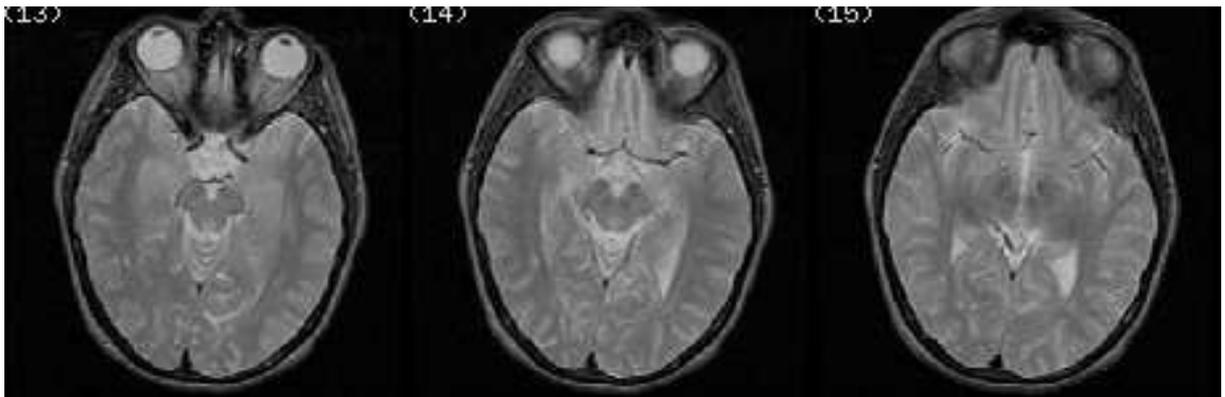


Figura 10: Ressonância magnética da cabeça de um paciente.

A título de curiosidade, tomógrafos também geram imagens digitais 4D, fazendo amostragem e quantização em (x, y, z, t) . Este é o caso de imagens tomográficas de um coração batendo, onde as três primeiras coordenadas são espaciais e a quarta coordenada é temporal.

4.4.2 Vídeo Digital

Um vídeo digital é uma seqüência de imagens, normalmente coloridas (multibandas), ao longo do eixo temporal. Portanto, pode ser definido como uma imagem digital 3D, onde as duas primeiras dimensões são espaciais (x, y) e a terceira dimensão é o tempo t . A Figura 12 mostra três quadros de uma seqüência de vídeo digital. Observe que neste caso, porém, em vez da informação 3D de objetos, temos objetos 2D se movimentando no



Figura 11: Reconstrução 3D de um crânio a partir de imagens tomográficas.

tempo.

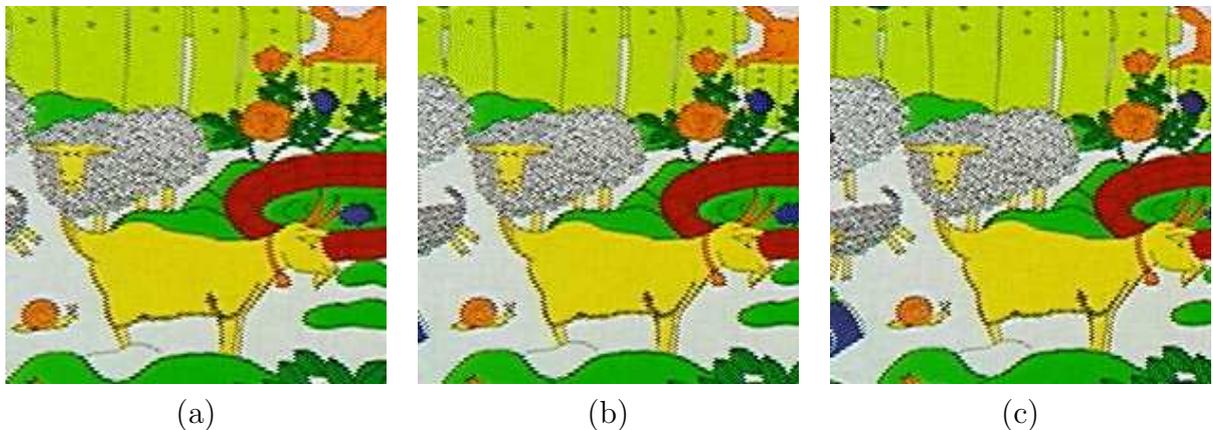


Figura 12: Sequência de vídeo digital com três quadros.

O vídeo digital pode ser encontrado com diferentes tipos de profundidade e resolução espacial. Em geral a profundidade é 24 bits por pixel, mas algumas variações serão discutidas com relação a TV digital. De acordo com o padrão MPEG-2, por exemplo, a TV digital deve ter 4 níveis de resolução espacial: baixa (288×352 pixels), principal (480×720 pixels), alta-1440 (1152×1440 pixels) e alta (1152×1920 pixels).

5 Elementos de um Sistema de Processamento de Imagens

Os elementos que compõem um sistema de processamento de imagem digital são aquisição, armazenamento, processamento, display e comunicação. As imagens podem ser adquiridas através de câmeras de vídeo, scanners de mesa, tomógrafos, satélites, etc. Estas imagens podem ser armazenadas em disco rígido ou em outros dispositivos de armazenamento, tais como, fitas magnéticas e discos ópticos. A imagem armazenada pode ser processada para cumprir os mais variados objetivos: compressão, melhoramento, reconhecimento de padrões, etc. A imagem adquirida ou processada pode ser visualizada em diversos dispositivos de display: filmes fotográficos, impressoras, tela do computador, etc. Estas imagens também podem ser transmitidas para outras localidades através de sistemas de comunicação.

6 Etapas do Processamento de Imagens

Podemos entender uma imagem como uma forma compacta de representar muitas informações. Em um sistema de processamento de imagens estas informações podem passar por diversas formas de representação. Portanto, as etapas do processamento de imagens descrevem o fluxo destas informações com um dado objetivo definido pela aplicação.

A Figura 13 ilustra este fluxo de forma geral. Após aquisição da imagem, o pré-processamento consiste na maior parte de transformações lineares e não-lineares aplicadas à imagem visando várias metas: melhoramento de contraste, remoção de ruído, regiões de interesse, decorrelação e codificação das informações para transmissão da imagem, reamostragem dos pixels em uma nova escala, treinamento e extração de características de imagem para segmentação, etc. Muitas aplicações requerem apenas operações de pré-processamento. As informações de interesse podem também ser extraídas das imagens e representadas de uma outra forma. Para tanto, a segmentação de imagens particiona a imagem em regiões disjuntas com algum significado para a aplicação. Por exemplo, podemos querer separar um objeto de interesse do resto dos pixels da imagem particionando-a em duas regiões. A saída da segmentação pode ser a fronteira do objeto com seu exterior ou os pontos de seu interior. Isto define duas formas de representação para o objeto. A representação consiste, portanto, das várias formas de armazenar a fronteira e o interior de objetos segmentados. Esta nova representação da imagem contém informações sobre a forma e a topologia dos objetos. A descrição quantitativa destas informações através da extração de características estruturais complementa o sentido de representação. Em seguida, com base na descrição, o reconhecimento associa um rótulo a cada objeto segmentado enquanto a interpretação associa um significado ao conjunto de objetos segmentados. Um exemplo em uma aplicação de leitura automática de endereços é o reconhecimento e a interpretação de um conjunto de caracteres como o código de endereçamento postal.

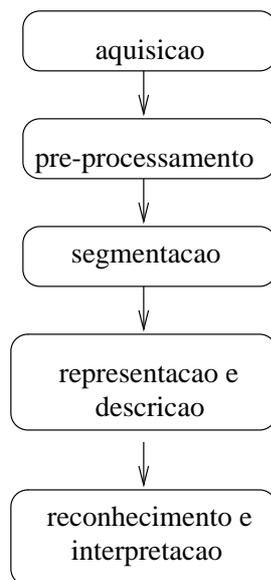


Figura 13: Etapas do processamento de imagens

7 Introdução à Topologia Digital

Topologia digital é o estudo de propriedades de objeto em imagem digital, as quais não são afetadas por transformações geométricas, exceto aquelas que envolvem junção ou separação de partes do objeto. Em processamento de imagem digital, um objeto em uma imagem 2D (ou 3D) é aproximado por um conjunto de pixels (ou voxels). Portanto, topologia digital estuda as propriedades deste conjunto de pixels (ou voxels) que correspondem às propriedades topológicas do objeto original. Nossa meta nesta seção é definir apenas os conceitos básicos usados em topologia digital para compreender o restante do curso.

7.1 Vizinhança

Em uma imagem digital 2D um pixel $p = (x, y)$ tem quatro vizinhos que compartilham uma aresta com p : $p = (x+1, y)$, $(x-1, y)$, $(x, y+1)$ e $(x, y-1)$ (Figura 14a). Este conjunto é chamado de vizinhança 4 de p ($N_4(p)$). Considerando os vizinhos que compartilham pelo menos um vértice com $p = (x, y)$ temos um conjunto vizinhança 8 de p ($N_8(p)$ - Figura 14b). Este conjunto é formado pelos pixels de $N_4(p)$ e os pixels diagonais $(x+1, y+1)$, $(x-1, y+1)$, $(x+1, y-1)$ e $(x-1, y-1)$. Um tratamento especial é normalmente dado aos pixels que pertencem às bordas da imagem, pois alguns de seus vizinhos vão estar fora da imagem.

Os conceitos de vizinhança são facilmente estendidos para imagens 3D, onde teremos vizinhanças 6, 18 e 26, considerando os voxels que compartilham com o voxel central uma face, pelo menos uma aresta e pelo menos um vértice, respectivamente. A Figura 14c,

por exemplo, ilustra o caso vizinhança 6. Neste caso os vizinhos do voxel $v = (x, y, z)$ são $(x - 1, y, z)$, $(x + 1, y, z)$, $(x, y - 1, z)$, $(x, y + 1, z)$, $(x, y, z - 1)$ e $(x, y, z + 1)$.

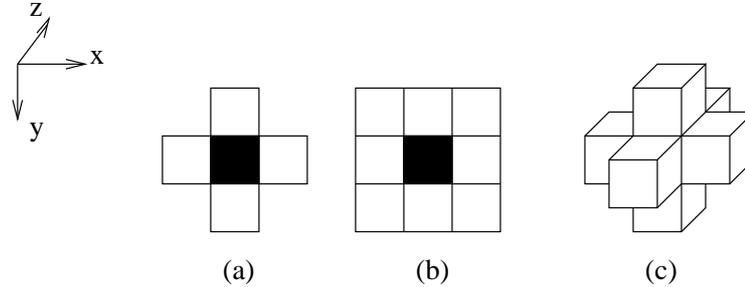


Figura 14: (a) Vizinhança 4 do pixel escuro (b) Vizinhança 8 do pixel escuro (c) Vizinhança 6 do voxel central.

De forma similar ao que foi feito para vizinhança, os conceitos definidos nas próximas seções também se estendem para uma imagem 3D. Para facilitar a explicação, entretanto, vamos doravante considerar apenas imagens 2D.

7.2 Relação Binária

Uma relação binária R aplicada a um conjunto X é um subconjunto do produto cartesiano $X \times X$.

Uma relação binária é dita reflexiva se $(a, a) \in R$, para todo $a \in X$, simétrica se $(a, b), (b, a) \in R$, para todo $a, b \in X$, e transitiva se $(a, b), (b, c) \in R$ implica que $(a, c) \in R$, para todo $a, b, c \in X$. Neste caso R é dita de equivalência.

7.3 Métrica

Uma função d de distância entre pixels é uma métrica se: $d(p, q) \geq 0$ ($d(p, q) = 0$, se $p = q$), $d(p, q) = d(q, p)$, $d(p, r) \leq d(p, q) + d(q, r)$, onde $p = (x_p, y_p)$, $q = (x_q, y_q)$, e $r = (x_r, y_r)$ são três pixels da imagem. As métricas mais usadas são:

- Euclideana: $d(p, q) = ((x_p - x_q)^2 + (y_p - y_q)^2)^{1/2}$,
- City-block: $d(p, q) = |x_p - x_q| + |y_p - y_q|$,
- Chessboard: $d(p, q) = \max\{|x_p - x_q|, |y_p - y_q|\}$.
- Chamfer: $d_{a,b}(p, q) = a \max\{|x_p - x_q|, |y_p - y_q|\} + (b - a) \min\{|x_p - x_q|, |y_p - y_q|\}$, onde a, b são constantes (e.g. $a = 5$ e $b = 7$).

7.4 Relação de Adjacência e Grafos

Uma relação de adjacência A é uma relação binária entre pixels, a qual depende de suas posições, e opcionalmente de outras propriedades locais da imagem. Dizemos que $A(p)$ é o conjunto dos pixels adjacentes ao pixel p de acordo com A . Isto é, $q \in A(p)$ é o mesmo que $(p, q) \in A$. Uma relação de adjacência leva, portanto, à definição de um grafo $G = (D, A)$ para a imagem, onde D é o conjunto de pixels. Neste grafo, um caminho π é uma seqüência de pixels adjacentes $\langle p_1, p_2, \dots, p_n \rangle$, onde $(p_i, p_{i+1}) \in A$, $i = 1, 2, \dots, n - 1$. Exemplos:

- $(p, q) \in A$ se $d(p, q) \leq \rho$, onde d é distância Euclideana e ρ é um escalar,
- $(p, q) \in A$ se $q - p \in \{(-1, -1), (1, -1)\}$,
- $(p, q) \in A$ se $|x_p - x_q| + |y_p - y_q| \leq 1$ e $|f(p) - f(q)| \leq l$, onde l é um limiar de brilho.

Observe que $\rho = 1$ é vizinhança-4, $\rho = \sqrt{2}$ é vizinhança-8, e $\rho = \sqrt{5}$ faz com que pixels mais distantes sejam vizinhos no grafo. Esta relação é simétrica e invariante à translação. Note também que o segundo exemplo está relacionado com a definição de elemento estruturante planar usada em morfologia matemática, e portanto uma relação de adjacência pode ser assimétrica.

7.5 Relação de Conexidade

Um pixel p é conexo a um pixel q se existir um caminho de p a q no grafo definido por A . Note que a conexidade pode ser assimétrica.

7.6 Componente Conexa

Um componente conexo de um conjunto X de pixels é um subconjunto $Y \subset X$, onde todos os pares (p, q) de pixels em Y são conexos (i.e. existe um caminho de p a q e um caminho de q a p , que não necessariamente são os mesmos).

Por exemplo, seja R uma relação binária "conexidade-4" aplicada sobre o conjunto $X = \{p_1, p_2, p_3, p_4\}$ da Figura 15; isto é $R = \{(p_1, p_2), (p_2, p_1), (p_2, p_3), (p_3, p_2), (p_1, p_3), (p_3, p_1)\}$. Note que p_4 não é conexo-4 com nenhum outro elemento de X .

7.7 Objeto

Anteriormente mencionamos que a segmentação de imagens pode ser utilizada para separar um objeto do resto dos pixels da imagem. Considerando o exemplo da Figura 16,

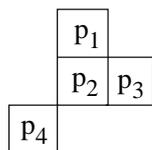


Figura 15: O conjunto X possui dois componentes conexos-4: $\{p_1, p_2, p_3\}$ e $\{p_4\}$.

podemos particionar uma imagem em componentes conexos disjuntos de acordo com diferentes critérios de classificação. Desta forma, o background da imagem é definido por um ou mais componentes conexos, C_1, C_2 e C_3 , que não têm significado de interesse para a aplicação. Cada objeto pode ser definido como um ou mais componentes conexos cuja vizinhança é composta por componentes do background ou por outros objetos. Neste caso temos dois objetos, O_1 e O_2 , onde o primeiro possui um componente conexo e o segundo dois.

Uma borda de objeto é um conjunto de pixels do seu interior que possui ao menos um pixel adjacente no exterior. As bordas do objeto O_2 , por exemplo, são B_2, B_3, B_4 , e B_5 . Um objeto pode ser representado por suas bordas ou pelos pixels que compõem seu interior.

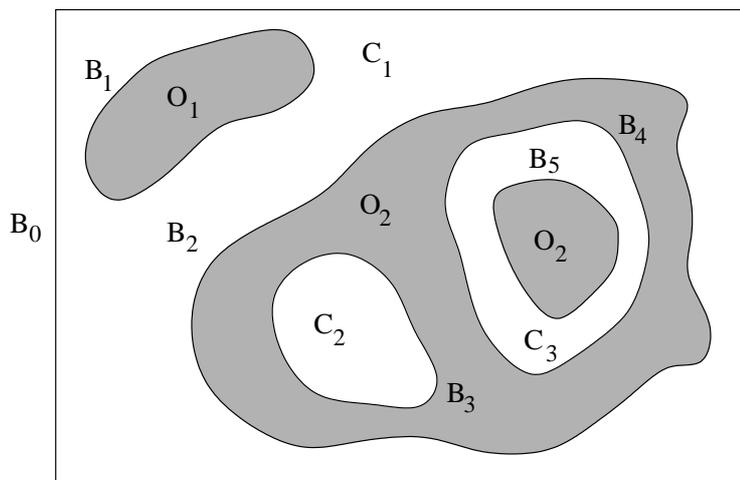


Figura 16: Uma imagem e seus componentes conexos.

O número de componentes conexos NC e o número de buracos NB são exemplos de propriedades topológicas de objeto. O número de buracos corresponde ao número de bordas internas (e.g. B_3 e B_4 na Figura 16) de todos os componentes do objeto. Essas propriedades podem ser usadas como descritores para análise. Um exemplo é o número de Euler definido por $NC - NB$.

8 Espaços de Cores

Um espaço Euclidiano 3D é gerado por uma origem e três vetores ortonormais. De forma similar, um espaço de cor é gerado por uma cor origem e três componentes primárias de cor. Dizemos então que uma imagem colorida usa um dado espaço de cor quando os três valores associados a cada pixel da imagem representam respectivamente as intensidades das três componentes primárias deste espaço. Uma imagem colorida pode ser representada em diferentes espaços de cores. O mais comum é o espaço RGB (red,green,blue). Este espaço é complementar do espaço CMY (cyan, magenta, yellow) como ilustra o cubo da Figura 17a. Note que os níveis de cinza de uma imagem monocromática caem na diagonal do cubo, que a origem do espaço RGB é a cor preta e que a origem do espaço CMY é a cor branca. A cor branca resulta da adição de máxima intensidade das componentes RGB e da ausência das componentes CMY, portanto RGB é dito aditivo e CMY é dito subtrativo. Sabemos também que a cor é definida por três grandezas: luminância, matiz e saturação. Os espaços HSV (hue, saturation, value) e HLS (hue, lightness, saturation) usam este tipo de representação. A Figura 17b mostra por exemplo a pirâmide do espaço HSV.

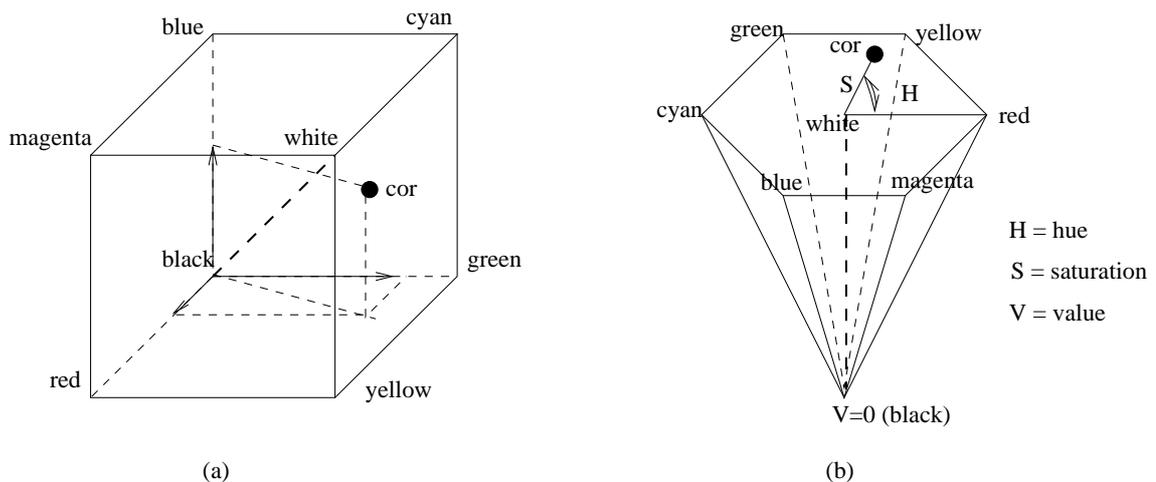


Figura 17: (a) Espaços RGB e CMY (b) Espaço HSV

Em vídeo digital, os espaços mais usados são RGB e YCbCr (TV digital). No espaço YCbCr, a luminância é representada pela componente Y enquanto a crominância (matiz e saturação) é representada pelas componentes Cb e Cr. Neste curso, estamos interessados nos espaços RGB e YCbCr, e nas transformações entre eles. Para transformar um quadro de vídeo de YCbCr para RGB e vice-versa, aplicamos as equações abaixo:

$$Y = 0.257R + 0.504G + 0.098B + 16 \quad (1)$$

$$Cr = 0.439R - 0.368G - 0.071B + 128 \quad (2)$$

$$Cb = -0.148R - 0.291G + 0.439B + 128 \quad (3)$$

$$R = 1.164(Y - 16) + 1.596(Cr - 128) \quad (4)$$

$$G = 1.164(Y - 16) - 0.813(Cr - 128) - 0.392(Cb - 128) \quad (5)$$

$$B = 1.164(Y - 16) + 2.017(Cb - 128) \quad (6)$$

9 Reamostragem

Uma imagem digital pode ser reamostrada visando reduzir ou aumentar sua resolução espacial. No caso de redução temos uma subamostragem e no caso de aumento de resolução temos uma interpolação.

Considere uma imagem 3D representada por $p(x, y, z)$, $x = 0, 1, \dots, M - 1$, $y = 0, 1, \dots, N - 1$, $z = 0, 1, \dots, K - 1$. Obviamente este exemplo também se aplica quando em vez de z temos o eixo temporal t . Para reamostragem, temos que considerar inicialmente uma representação intermediária na qual x' , y' e z' podem assumir valores reais positivos. Esta representação intermediária consiste de pixels $q(x', y', z')$ cujo valor é uma função linear/não-linear do valor dos pixels p da imagem original que estão em uma certa vizinhança de q , e $x' = mx$, $y' = ny$, $z' = kz$, m, n, k são valores reais e positivos. Em seguida, os valores de x' , y' e z' são reenumerados tal que $x' = 0, 1, \dots, M' - 1$, $y' = 0, 1, \dots, N' - 1$, $z' = 0, 1, \dots, K' - 1$. Na subamostragem temos $m, n, k > 1$ e na interpolação temos $0 < m, n, k < 1$. Observe que poderíamos subamostrar uma imagem em uma dada direção e interpolá-la em outra. Note que $N' < N$, $M' < M$ e $K' < K$ na subamostragem, e $N' > N$, $M' > M$ e $K' > K$ na interpolação.

O experimento apresentado na Figura 4 para uma imagem 2D consiste de subamostragem usando $m = 2$ e $n = 2$ seguida de uma interpolação com $m = 1/2$ e $n = 1/2$. Na subamostragem, o valor dos pixels $q(x', y') = p(2x, 2y)$, $x' = 0, 1, \dots, (M/2) - 1$, $y' = 0, 1, \dots, (N/2) - 1$, selecionando apenas as amostras de coordenadas pares da imagem original e reduzindo à metade suas dimensões. Na interpolação, o valor dos pixels $q(x', y') = p(x/2, y/2)$, $x' = 0, 1, \dots, M - 1$, $y' = 0, 1, \dots, N - 1$, é obtido por replicação do pixel mais próximo da imagem reduzida retornando às dimensões da imagem original. As reamostragens utilizadas neste experimento utilizam uma função linear de ordem zero para estimar o valor dos pixels q .

Existem várias situações que requerem reamostragem dos pixels durante o processamento. A seguir vamos ver dois exemplos envolvendo TV digital e imagem tomográfica.

9.1 TV Digital

Em TV digital, os formatos de vídeo levam em conta as limitações de banda de frequência para transmissão e a estratégia de exposição do vídeo na tela da televisão (30 quadros

por segundo, ou melhor, 60 campos por segundo). Com relação ao segundo aspecto, o vídeo digital é dito entrelaçado. Ou seja, cada quadro é composto por dois campos. O campo par, formado pelas linhas pares do quadro, representando a imagem no instante t e o campo ímpar, formado pelas linhas ímpares, representando a imagem no instante $t + 1$. Assim, um único quadro de vídeo já contém informação temporal e pode ser visto como uma imagem 3D em (x,y,t) . Na verdade, cada campo representa uma imagem subamostrada na vertical com $n = 2$. A Figura 18 mostra, por exemplo, uma subregião de um quadro de vídeo (Figura 18a) e o campo par desta subregião (Figura 18b). Observe que devido ao movimento entre os campos, as bordas da imagem da Figura 18a estão quebradas. Por outro lado, devido a subamostragem, a imagem da Figura 18b está achatada. Portanto, visando um processamento eficiente destas imagens, cada quadro de vídeo deve ser separado em dois campos e cada campo deve ser interpolado na vertical com $n = 1/2$.

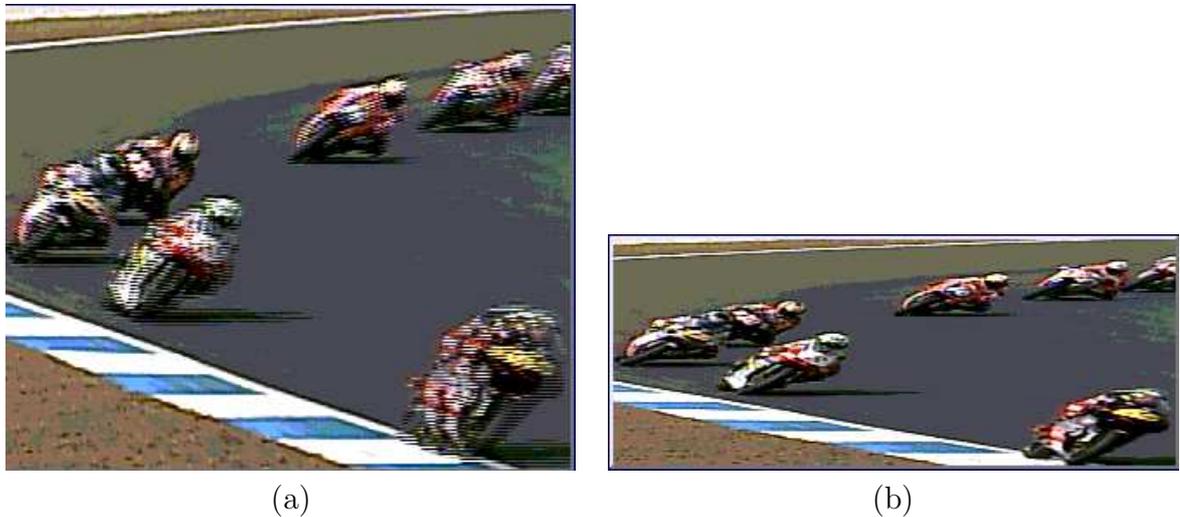


Figura 18: (a) Região de interesse de um quadro de vídeo (b) Campo par da imagem em (a).

O vídeo no espaço YCbCr pode ainda ser encontrado com diferentes formatos: 444, 422 e 420. Estes formatos estão relacionados com uma subamostragem nas componentes de croma, pois estas são menos relevantes para a qualidade do vídeo do que a informação de luminância. A Figura 19 ilustra o processo de conversão espacial entre estes formatos. Para passar de 444 para 422, as componentes de croma Cb e Cr são subamostradas na horizontal com $m = 2$. Para passar de 422 para 420, as componentes Cb e Cr são subamostradas na vertical com $n = 2$. Para fazer o caminho inverso é só substituir subamostragem por interpolação. A Figura 20a mostra, por exemplo, a banda de luminância de um quadro de vídeo e a Figura 20b mostra a banda de croma. Neste caso temos duas bandas por quadro e o vídeo está no formato YCbCr:422.

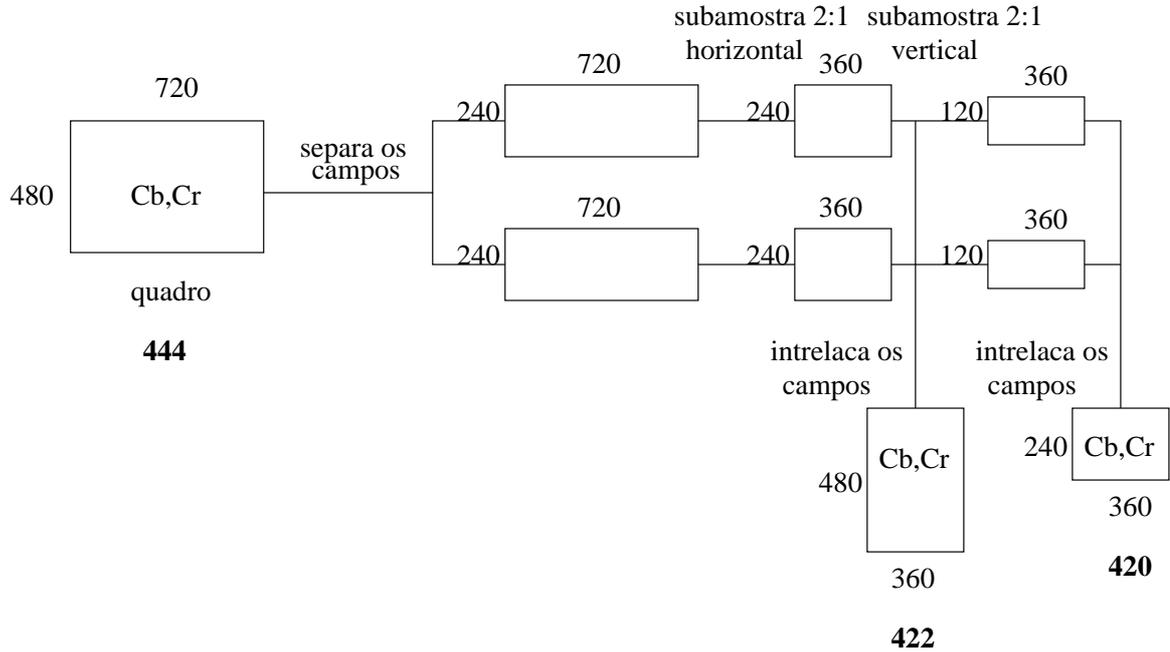


Figura 19: Conversão espacial de croma entre formatos 444, 422 e 420.

9.2 Imagem Tomográfica

Imagens tomográficas são normalmente subamostradas na direção z (i.e. $d > p$ - ver Figura 9). Para evitar distorções na reconstrução 3D de objetos, estas imagens devem possuir voxels cúbicos (i.e. $d = p$). Muito embora possamos interpolar/subamostrar os voxels em qualquer direção, nós normalmente interpolamos as imagens na direção z usando uma função linear ou uma função trilinear dependendo da vizinhança considerada. Note que o número de fatias a serem interpoladas vai depender da relação d/p .

Na interpolação linear assume-se que a variação de intensidade dos voxels é linear na direção z . A Figura 21 mostra como é feita a interpolação linear de um voxel v_m considerando seus vizinhos v_n e v_{n+1} na direção z . A intensidade associada ao novo voxel v_m é dada pela equação abaixo.

$$i(v_m) = i(v_n) + \frac{(i(v_{n+1}) - i(v_n))l_i}{l_s + l_i} \quad (7)$$

A interpolação trilinear explora a vizinhança em todas as direções x , y e z . A Figura 22 ilustra, por exemplo, um caso de interpolação trilinear que considera um cubo, onde os vértices são os 8 vizinhos mais próximos do voxel v_m a ser interpolado. O processo utiliza as distâncias d_1 , d_2 e d_3 mostradas na Figura 22b, e os valores de d e p para calcular a intensidade $i(v_m)$. Este cálculo consiste de sete operações que usam a equação 7. Calcula-se $i(p_1)$ a partir de $i(v_1)$ e $i(v_2)$, $i(p_2)$ a partir de $i(v_3)$ e $i(v_4)$, e $i(p_3)$ a partir de $i(p_1)$ e $i(p_2)$. De forma similar chega-se a $i(p_6)$ e finalmente calcula-se $i(v_m)$ a partir de $i(p_3)$



Figura 20: Quadro de vídeo no formato YCbCr:422. (a) Componente Y (b) Componentes Cb e Cr.

e $i(p_6)$. Note que podemos ainda considerar diferentes vizinhanças e diferentes métricas para medir a distância entre os voxels gerando diferentes tipos de interpolação.

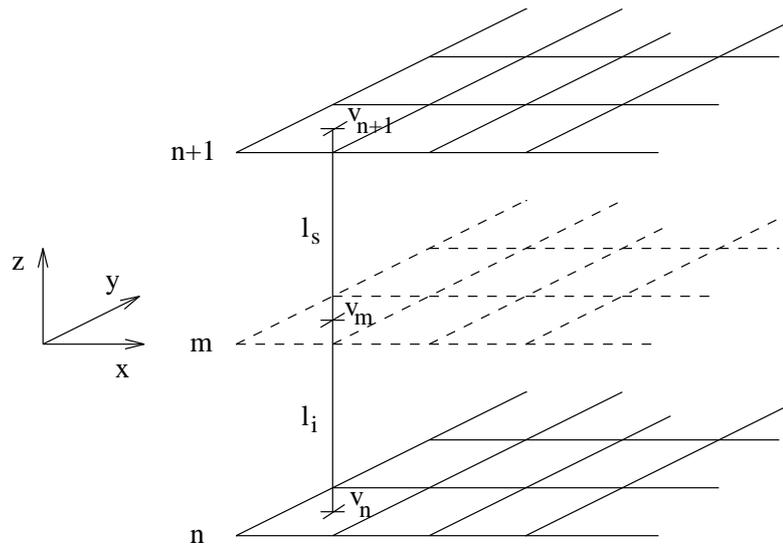


Figura 21: Interpolação Linear

Para concluir, a Figura 23 ilustra as distorções geradas na reconstrução 3D do objeto da Figura 11 quando subamostrado ou superamostrado na direção z . A Figura 23a mostra um caso de subamostragem ($d > p$) e a Figura 23b mostra um caso de superamostragem ($d < p$).

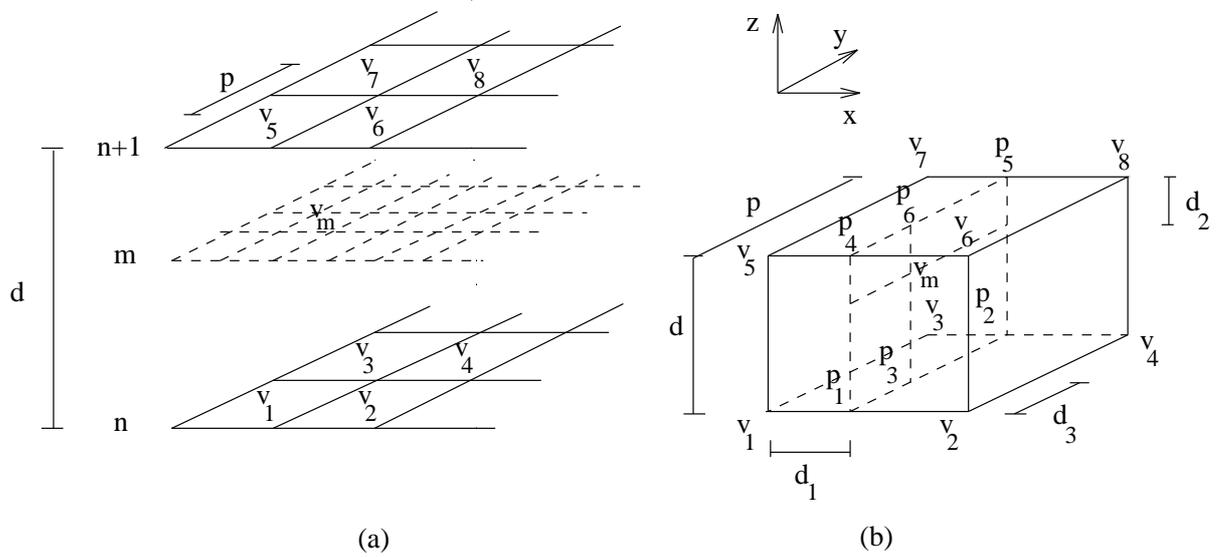
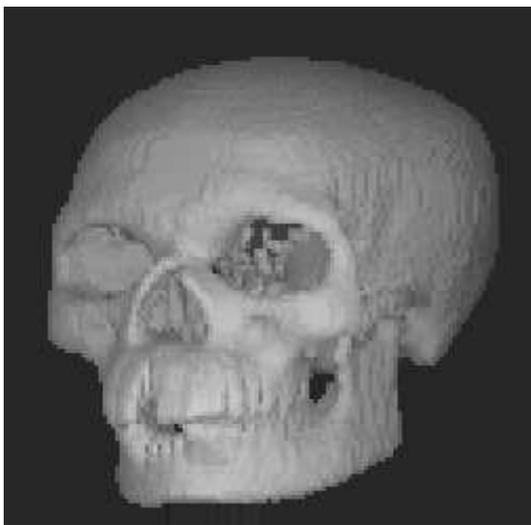
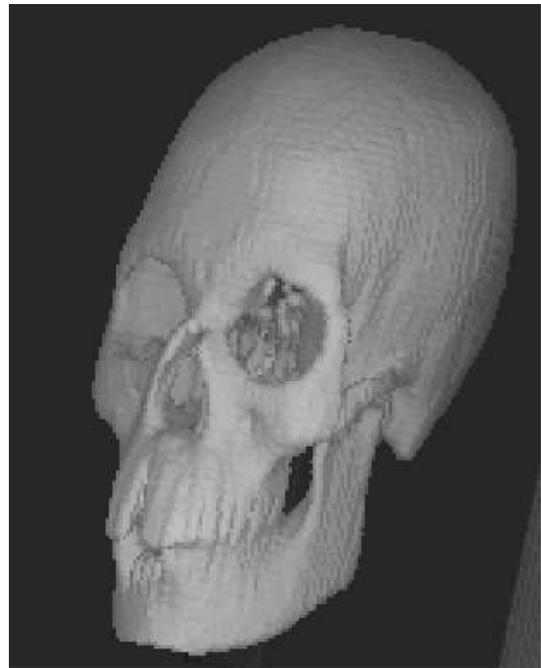


Figura 22: Interpolação Trilinear



(a)



(b)

Figura 23: Reconstrução 3D de um crânio. (a) Subamostragem (b) Superamostragem.

PRÉ-PROCESSAMENTO

10 Histograma

O histograma dos níveis de cinza de uma imagem f , $h_f(z)$, constitui uma *operação global* indicando a frequência de ocorrência dos níveis de cinza ou brilho z de f (função densidade de probabilidade). Assim, o histograma de uma imagem com L níveis de cinza é representado por um vetor de L elementos, cujos valores podem ser definidos a partir do seguinte algoritmo:

Início

$h(f(x,y)) = 0$ {zera contadores de níveis de cinza }

Para cada valor $f(x,y)$ **faça**

$h(f(x,y)) = h(f(x,y)) + 1$

Fim-Para

Fim

O histograma dá uma idéia global da *dinâmica* de uma cena e tem aplicações em inúmeras transformações de imagens, tais como filtragem, segmentação, reconhecimento de padrões etc. As figuras 24 e 25 apresentam imagens com seus respectivos histogramas.

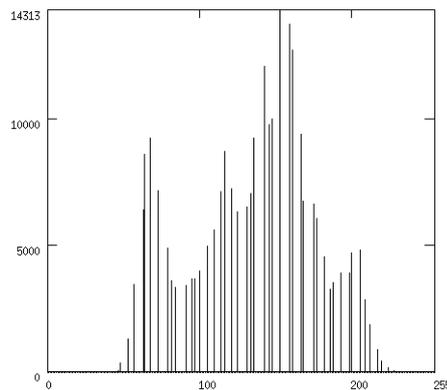


Figura 24:

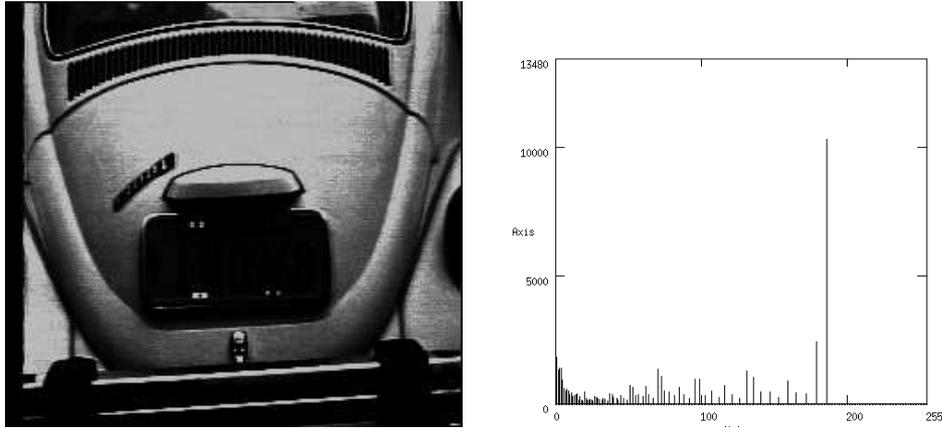


Figura 25:

Observe que um mesmo histograma pode estar associado a diferentes imagens e que a sua informação é invariante a operações de rotação e translação (objetos se movendo num mesmo background) da imagem original. Um histograma pode ser definido igualmente para imagens multiespectrais. No caso de imagens coloridas, por exemplo, podemos definir um histograma para cada banda espectral ou um histograma 3-D referente às componentes RGB da imagem. Neste caso, a intensidade dos voxels (pixels num volume) é proporcional à frequência de aparição dos níveis de cinza na cena original.

11 Transformações radiométricas ou de escala de cinza

As transformações radiométricas independem da localização dos pixels na imagem e podem, de maneira geral (e em termos de implementação), ser representadas por uma operação do tipo *look-up table* (LUT) que transforma um pixel de nível de cinza g_i em um nível de cinza g_f . Seja \mathbf{G}_i a escala de cinza da imagem inicial \mathbf{f} : $\mathbf{G}_i = [0,1,\dots,N_i]$, e \mathbf{G}_f a escala de cinza da imagem resultante \mathbf{g} : $\mathbf{G}_f = [0,1,\dots,N_f]$. Uma LUT é uma aplicação $\mathbf{f}(\mathbf{G}_i) \rightarrow \mathbf{g}(\mathbf{G}_f)$ tal que

$$\forall g_i \in \mathbf{G}_i, \quad \exists g_f \in \mathbf{G}_f, \quad g_f = r(g_i) \quad (8)$$

Alguns exemplos de funções r de transformações radiométricas são indicados na Fig. 26.

A função r_1 define o *negativo* ou *complemento* da imagem original; a função r_2 realça o *contraste* entre os pontos p_1 e p_2 , e r_3 corresponde a uma *limiarização* (thresholding) dos pixels de entrada, resultando numa imagem *binária* (preta e branca). A Fig. 27 ilustra estas transformações para r_1 , r_2 e r_3 , respectivamente.

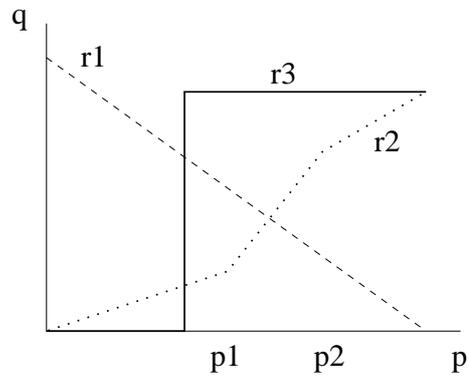


Figura 26:



Figura 27:

11.1 Equalização histogrâmica

A equalização ou modelagem histogrâmica é uma transformação radiométrica que visa aumentar a dinâmica dos níveis de cinza melhorando, por exemplo, o contraste de imagens digitalizadas sob péssimas condições de iluminação.

De modo geral, o que se procura é obter um mapeamento não-linear dos níveis de cinza da imagem de entrada de tal forma que a imagem resultante contenha uma distribuição mais uniforme dos seus níveis de cinza, ou seja, um histograma planar (Fig. 28).

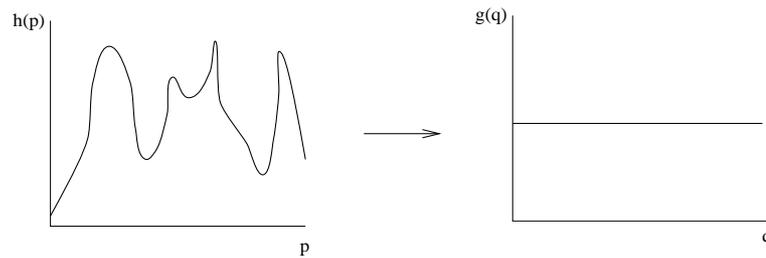


Figura 28:

Existem várias técnicas de equalização histogrâmica cujas abordagens consideram inicialmente funções contínuas e não discretas. A seguir, ilustraremos uma destas técnicas apresentada em [Woo93].

Seja f uma variável no intervalo $[0, 1]$ (f representa, por exemplo, a intensidade normalizada dos níveis de cinza de uma imagem de tal forma que $0 \leq f(x, y) \leq 1$). Uma transformação T , no intervalo $[0, 1]$, é tal que

$$g = T(f) \text{ (ou } g(x, y) = T(f(x, y)), \text{ para o caso de nossa imagem discreta)} \quad (9)$$

onde g é o valor resultante obtido a partir do nível de entrada f . Visando a monotonicidade e preservação da escala de cinza original, a transformação T deve satisfazer as seguintes condições (Fig. 29):

- T deve ser monotonicamente crescente no intervalo $[0, 1]$.
- $0 \leq T(f) \leq 1$ para $0 \leq f \leq 1$.

A primeira condição preserva a ordem na seqüência da escala de cinza, variando do preto ao branco, e a segunda garante que o intervalo desta escala continua o mesmo.

A transformação inversa, que também satisfaz as condições acima, é denotada por:

$$f = T^{-1}(g) \quad (10)$$

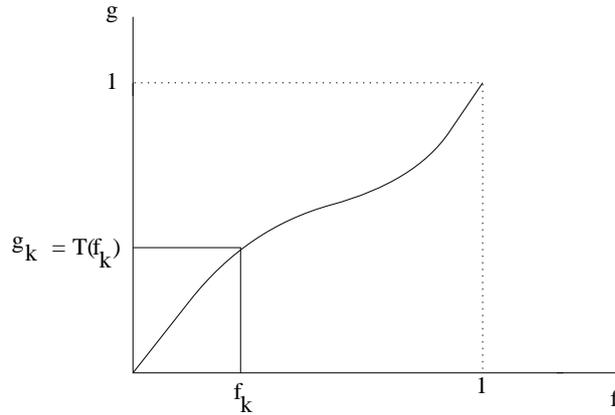


Figura 29:

Considerando f e g como variáveis contínuas, os níveis de cinza das imagens podem ser caracterizados por suas funções densidade de probabilidade (ou histogramas) denotadas por $p_f(f)$ e $p_g(g)$, respectivamente.

Consideremos, agora, a seguinte função de transformação

$$g = T(f) = \int_0^f p_f(w)dw \quad 0 \leq f \leq 1, \quad (11)$$

representando a função de distribuição cumulativa (FDC) de f . Esta função é monotonicamente crescente e varia de 0 a 1 em função de f .

Exercício: Mostre que a função densidade de probabilidade $p_g(g)$ resultante da transformação acima é, como desejamos, uniforme no intervalo $[0, 1]$.

A partir do exercício acima, concluímos que se $T(f)$ é uma FDC, então esta função pode ser empregada na definição de uma nova imagem com distribuição mais uniforme dos níveis de cinza. Como mencionado anteriormente, o efeito aqui será visível no que concerne o aumento do contraste dos níveis de cinza de uma imagem apresentando baixa dinâmica.

11.1.1 O caso discreto

Dada uma imagem contendo $n = M \times N$ pixels, assumindo valores discretos $k = 0, 1, \dots, L-1$, a extensão destes conceitos para o caso discreto se dá a partir das seguintes definições de probabilidades:

$$p_f(f_k) = \frac{n_k}{n} \quad 0 \leq f_k \leq 1, \quad (12)$$

onde n_k é o número de aparições do nível k , e $p_f(f_k)$ é a probabilidade deste nível. Finalmente, a forma discreta da Eq. 11 é

$$g_k = T(f_k) = \sum_{j=0}^k \frac{n_j}{n} = \sum_{j=0}^k p_f(f_j), \quad 0 \leq f_k \leq 1 \text{ e } k = 0, 1, \dots, L-1. \quad (13)$$

Esta equação define a função de transformação discreta de equalização histogrâmica. A Fig. 30 ilustra esta operação.

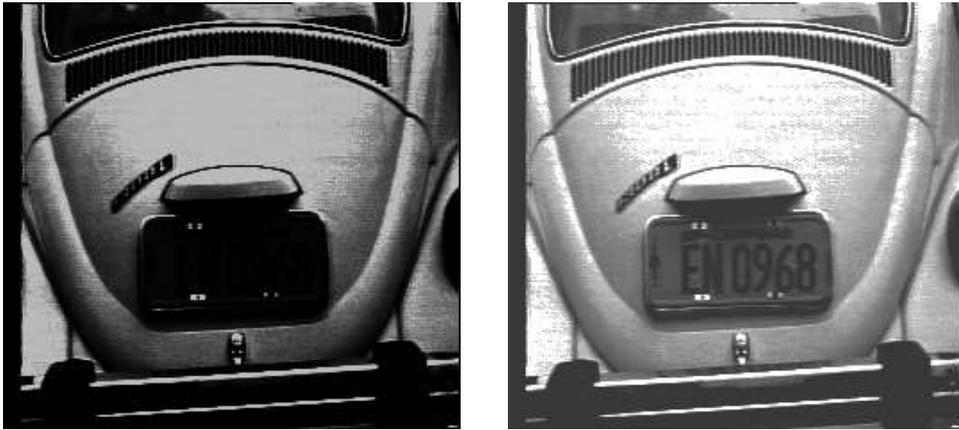


Figura 30:

12 Operações lineares: a Transformada de Fourier

A Transformada de Fourier (FT) é uma ferramenta largamente empregada em processamento de sinais, de modo geral, e em processamento de imagens, em particular. Denominada assim em homenagem ao físico francês Jean Baptiste Fourier (1768-1830), a FT decompõe um sinal em suas componentes elementares seno e cosseno. A FT aplicada a uma imagem no *domínio espacial* gera uma informação no *domínio da frequência*, em que cada ponto, definido por um vetor do tipo $(k.\text{cosseno}, k.\text{seno})$, representa uma dada frequência contida no domínio espacial da imagem.

As aplicações referentes à FT são inúmeras: filtragem, segmentação, reconhecimento de padrões, descrição de imagens, compressão e reconstrução constituem algumas delas.

12.1 Fundamentos matemáticos

As próximas seções apresentam alguns conceitos matemáticos básicos relacionados com a definição de transformadas lineares [Kak82].

12.1.1 Operações lineares em imagens

Seja Γ uma operação que transforma uma imagem \mathbf{f} em uma outra imagem \mathbf{g} , isto é, $\Gamma : \mathbf{f} \rightarrow \mathbf{g}$. Γ é um operador linear se:

$$\Gamma[a\mathbf{f} + b\mathbf{g}] = a\Gamma[\mathbf{f}] + b\Gamma[\mathbf{g}], \quad (14)$$

para \mathbf{f} e \mathbf{g} imagens e a, b constantes quaisquer.

Uma imagem 2-D pode ser vista como uma coleção de *fontes pontuais* concentradas no espaço bidimensional. No caso linear, a análise da saída de uma fonte pontual pode ser utilizada para definir a *função de espalhamento pontual* de Γ associada à *função de transferência* do sistema.

Seja a função definindo a seguinte seqüência de imagens

$$\text{rect}(x, y) = \begin{cases} 1, & \text{para } |x| \leq \frac{1}{2}, |y| \leq \frac{1}{2} \\ 0, & \text{caso contrário} \end{cases} \quad (15)$$

e seja

$$\delta_n = n^2 \text{rect}(nx, ny), \quad n = 1, 2, \dots, \quad (16)$$

isto é, δ_n é zero fora da área $1/n \times 1/n$ definida por $|x| \leq 1/2n, |y| \leq 1/2n$, e tem valor

constante n^2 dentro desta área. Assim,

$$\int \int_{-\infty}^{+\infty} \delta_n(x, y) dx dy = 1, \quad \text{para } n \text{ qualquer.} \quad (17)$$

A função definida pelo pseudo-limite da Eq. 17 acima, denotado por δ , quando $n \rightarrow \infty$, é conhecida como *função Delta de Dirac*. Neste caso, $\delta(x, y) = 0$ para todo (x, y) diferente de $(0, 0)$ e infinito neste domínio. Observe que $\delta(-x, -y) = \delta(x, y)$. Baseado em 17, definimos

$$\int \int_{-\infty}^{+\infty} \delta(x, y) dx dy = 1, \quad (18)$$

e podemos ver facilmente que

$$\int \int_{-\infty}^{+\infty} g(x, y) \delta(x, y) dx dy = g(0, 0), \quad (19)$$

ou ainda

$$\int \int_{-\infty}^{+\infty} g(x, y) \delta(x - \alpha, y - \beta) dx dy = g(\alpha, \beta) \quad (20)$$

que corresponde à *propriedade de deslocamento* da função. Outra propriedade interessante é dada por

$$\int \int_{-\infty}^{+\infty} e^{j2\pi(ux+vy)} dudv = \delta(x, y), \quad (21)$$

em que $j = \sqrt{-1}$ e $e^{-j\theta} = \cos\theta - j\sin\theta$, θ é a *variável de freqüência*.

A definição da função Delta de Dirac é considerada em várias aplicações como, por exemplo, na abordagem de problemas relacionados com quantização, filtragem (definição da função característica de um sistema), reconstrução a partir de projeções etc.

13 Invariância ao deslocamento

Seja a função de espalhamento $h(x, y)$ que define a saída de um ponto (x, y) de uma imagem. Uma operação linear é dita invariante ao deslocamento se a resposta à fonte pontual $\delta(x - \alpha, y - \beta)$, localizada no plano 2-D, é dada por $h(x - \alpha, y - \beta)$ (a saída é igualmente deslocada de α e β no plano).

Consideremos $f(x, y)$ uma imagem de entrada qualquer de um sistema. Pela Eq. 20, esta imagem pode ser vista como uma soma linear de fontes pontuais, isto é,

$$f(x, y) = \int \int_{-\infty}^{+\infty} f(\alpha, \beta) \delta(\alpha - x, \beta - y) d\alpha d\beta \quad (22)$$

Assumindo nosso operador linear Γ invariante ao deslocamento, obtemos

$$\Gamma[f(x, y)] = \Gamma\left[\int \int_{-\infty}^{+\infty} f(\alpha, \beta) \delta(\alpha - x, \beta - y) d\alpha d\beta\right] = \int \int_{-\infty}^{+\infty} f(\alpha, \beta) \Gamma[\delta(\alpha - x, \beta - y)] d\alpha d\beta, \quad (23)$$

o que significa que a resposta ao somatório de excitações é igual ao somatório da resposta de cada excitação.

Como a resposta a $\delta(\alpha - x, \beta - y) = \delta(x - \alpha, y - \beta)$, que representa uma fonte pontual em (α, β) , é dada por $h(x - \alpha, y - \beta)$, então

$$\Gamma[f(x, y)] = g(x, y) = \int \int_{-\infty}^{+\infty} f(\alpha, \beta) h(x - \alpha, y - \beta) d\alpha d\beta \quad (24)$$

define a **convolução** de \mathbf{f} e \mathbf{h} , $\mathbf{f} * \mathbf{h}$. Observe que $\mathbf{f} * \mathbf{h}$ é igual a $\mathbf{h} * \mathbf{f}$. A Fig. 31 ilustra esta operação.

14 Análise de Fourier

A Transformada de Fourier $F(u, v)$ de uma função 2-D, $f(x, y)$, é dada por

$$F(u, v) = \int \int_{-\infty}^{+\infty} f(x, y) e^{-j2\pi(ux+vy)} dx dy \quad (25)$$

Observe que a transformação apresenta valores complexos constituídos de uma parte real, R , e imaginária, I . Assim:

$$F(u, v) = R(u, v) + jI(u, v) \quad (26)$$

ou

$$F(u, v) = |F(u, v)| e^{-j\phi(u, v)} \quad (27)$$

em que

$$|F(u, v)| = \sqrt{R^2(u, v) + I^2(u, v)} \quad (28)$$

corresponde ao *espectro de Fourier*, e

$$\phi(u, v) = \text{tg}^{-1}\left[\frac{I(u, v)}{R(u, v)}\right] \quad (29)$$

define o *ângulo de fase*. A *energia* E de $f(x, y)$ é dada por

$$E(u, v) = |F(u, v)|^2 \quad (30)$$

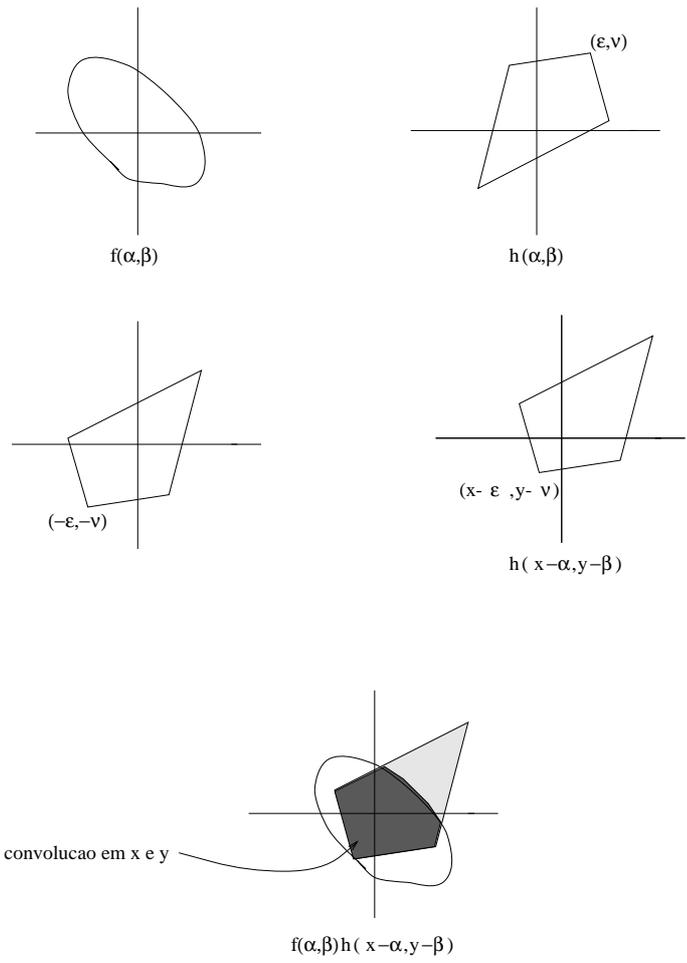


Figura 31: [Kak82]

Exercício 1: Mostre que a FT da função $rect(x, y)$ é a função $sinc(u, v)$ (Fig. 32), ou seja

$$F(u, v) = \frac{\text{sen}\pi u}{\pi u} \frac{\text{sen}\pi v}{\pi v} \quad (31)$$

Exercício 2: Generalize o resultado anterior mostrando que a FT de $rect(nx, ny)$ é

$$(1/n^2) \text{sinc}(u/n, v/n) \quad (32)$$

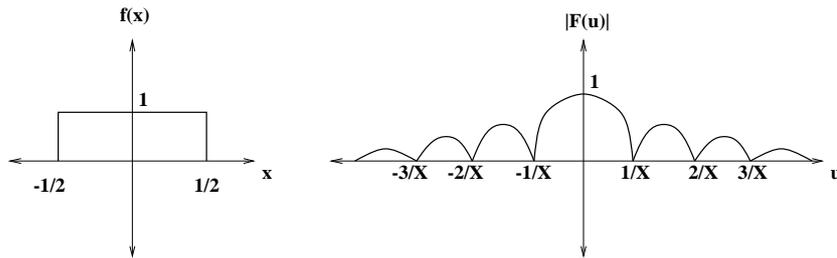


Figura 32:

Uma condição necessária para a existência de uma FT é que o sinal não tenha descontinuidades infinitas. Naturalmente, a função Delta de Dirac não obedece a esta condição. Na realidade, ela representa o limite da seqüência de funções $\delta_n = n^2 \text{rect}(nx, ny)$ bem comportada que, por sua vez, possui uma FT. Portanto, podemos definir o limite ($n \rightarrow \infty$) desta FT como sendo a FT da função Delta de Dirac, ou seja, quando

$$f(x, y) = \delta(x, y) \quad (33)$$

então

$$F[f(x, y)] = F(u, v) = \lim_{n \rightarrow \infty} \text{sinc}(u/n, v/n) = 1 \quad (34)$$

Multiplicando ambos os lados da Eq. 25 por $e^{j2\pi(ux+vy)}$, integrando em relação a u e v , e depois considerando a equação 21, obtemos:

$$\int \int_{-\infty}^{+\infty} F(u, v) e^{-j2\pi(ux+vy)} du dv = f(x, y) \quad (35)$$

ou

$$f(x, y) = \int \int_{-\infty}^{+\infty} F(u, v) e^{j2\pi(ux+vy)} du dv \quad (36)$$

Este último resultado define a *FT inversa* que expressa um sinal f como uma combinação linear de senos e cossenos de magnitudes e fases diferentes. Esta equação, juntamente com a *FT direta* (Eq. 25), constitui o par de transformadas de Fourier.

14.1 Algumas propriedades da FT

Seja $\mathcal{F}[f]$ a FT de uma função $f(x, y)$, isto é, $\mathcal{F}[f] = F(u, v)$. Algumas propriedades interessantes da FT são [Pap62], [Kor88]:

1. Linearidade:

$$\begin{aligned}\mathcal{F}[af_1(x, y) + bf_2(x, y)] &= a\mathcal{F}[f_1(x, y)] + b\mathcal{F}[f_2(x, y)] \\ &= aF_1(u, v) + bF_2(u, v)\end{aligned}$$

2. Escalonamento:

$$\mathcal{F}[f(\alpha x, \beta y)] = \frac{1}{\alpha\beta} F\left(\frac{u}{\alpha}, \frac{v}{\beta}\right)$$

3. Deslocamento no espaço:

$$\mathcal{F}[f(x - \alpha, y - \beta)] = F(u, v)e^{-j2\pi(u\alpha + v\beta)}$$

4. Deslocamento na frequência:

$$\mathcal{F}[e^{j2\pi(u_0x + v_0y)} f(x, y)] = F(u - u_0, v - v_0)$$

5. Rotação de 180°:

$$\mathcal{F}[\mathcal{F}[f(x, y)]] = f(-x, -y)$$

6. Convolução:

$$\begin{aligned}\mathcal{F}\left[\int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} f_1(\alpha, \beta) f_2(x - \alpha, y - \beta) d\alpha d\beta\right] &= \mathcal{F}[f_1(x, y)] \mathcal{F}[f_2(x, y)] \\ &= F_1(u, v) F_2(u, v)\end{aligned}$$

Este último resultado é de grande importância em processamento de sinais. Ele mostra que a convolução de duas funções no domínio espacial é equivalente à operação de multiplicação no domínio da frequência. A equivalência para a convolução no domínio da frequência pode ser expressa por

$$\mathcal{F}[f_1(x, y) f_2(x, y)] = \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} F_1(u - s, v - t) F_2(s, t) ds dt$$

15 Transformadas discretas: o caso geral

Seja \mathbf{f} uma imagem de dimensão $M \times N$ representada pela seguinte matriz de pixels:

$$[f]_{M \times N} = \begin{bmatrix} f(0, 0) & f(0, 1) & \cdots & f(0, N - 1) \\ f(1, 0) & f(1, 1) & \cdots & f(1, N - 1) \\ \vdots & \vdots & \cdots & \vdots \\ f(M, 0) & f(M, 1) & \cdots & f(M, N) \end{bmatrix}$$

Uma transformada discreta geral de $[f]_{M \times N}$ é uma matriz $[F]_{M \times N}$ dada pelo seguinte produto

$$[F] = [P][f][Q] \quad (37)$$

onde $[P]$ e $[Q]$ são matrizes não-singulares (possuem inversa, determinante $\neq 0$) de dimensão $M \times M$ e $N \times N$, respectivamente:

$$[F]_{M \times N} = [P]_{M \times M}[f]_{M \times N}[Q]_{N \times N} \quad (38)$$

A transformada $[F]_{M \times N}$ pode ser escrita, ainda, da seguinte forma

$$F(u, v) = \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} P(u, m) f(m, n) Q(n, v) \quad \text{para } u=0,1, \dots, M-1 \text{ e } v=0,1, \dots, N-1. \quad (39)$$

Sejam, agora, $[P]^{-1}$ e $[Q]^{-1}$ as matrizes inversas de $[P]$ e $[Q]$, respectivamente. É fácil mostrar que a transformada inversa de $[F]$ é dada por

$$[f] = [P]^{-1}[F][Q]^{-1} \quad (40)$$

Exercício: Verifique que se $[P]$ e $[Q]$ são matrizes reais, simétricas e ortogonais, então

$$[F] = [P][f][Q], \quad [f] = [P][F][Q] \quad (41)$$

15.1 Exemplo de uma caso particular: a Transformada Discreta de Fourier

Seja a matriz de transformação $[\Phi_{J \times J}]$ com um pixel na posição (m, n) dado por

$$\frac{1}{J} e^{-j(2\pi/J)mn}, \quad m, n = 0, 1, \dots, J-1 \quad (42)$$

e

$$[P] = [\Phi_{M \times M}] \quad e \quad [Q] = [\Phi_{N \times N}] \quad (43)$$

A Transformada Discreta de Fourier - DFT de uma matriz \mathbf{f} é dada por

$$[F] = [P][f][Q] \quad (44)$$

Considerando as definições de $[P]$ e $[Q]$ acima, e a expansão em 39, obtemos

$$F(u, v) = \frac{1}{MN} \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} f(m, n) e^{-j2\pi(\frac{mu}{M} + \frac{nv}{N})} \quad (45)$$

para $u = 0, 1, \dots, M - 1$ e $v = 0, 1, \dots, N - 1$.

Multiplicando-se $[\Phi_{J \times J}]$ pela matriz $[\Phi_{J \times J}]'$ definida por

$$e^{j(2\pi/J)mn}, \quad (46)$$

obtemos a matriz identidade, o que significa naturalmente que $[\Phi_{J \times J}]' = [\Phi_{J \times J}]^{-1}$.

Considerando-se $[P] = [\Phi_{M \times M}]$ e $[Q] = [\Phi_{N \times N}]$, então $[P]^{-1} = [\Phi_{M \times M}]^{-1} e [Q]^{-1} = [\Phi_{N \times N}]^{-1}$. Assim, a *Transformada Discreta Inversa de Fourier* é dada por

$$[f] = [P]^{-1}[F][Q]^{-1} \quad (47)$$

ou na sua forma expandida

$$f(m, n) = \sum_{u=0}^{M-1} \sum_{v=0}^{N-1} F(u, v) e^{j2\pi(\frac{mu}{M} + \frac{nv}{N})} \quad (48)$$

para $m = 0, 1, \dots, M - 1$ e $n = 0, 1, \dots, N - 1$.

15.2 DFT: visualização e implementação

Como vimos, a FT representa de forma bastante diferente a informação contida no domínio espacial. A Fig. 33 ilustra, para o caso 1-D, o modelo do processo reversível associado ao par de transformadas *direta* e *inversa* de Fourier.

A DFT é a versão digital da FT e, assim, não contém todas as freqüências constituintes da imagem mas um conjunto significativo destas capaz de descrevê-la.

Para o caso de imagens com dimensão $N \times N$, a DFT pode ser dada por:

$$F(u, v) = \frac{1}{N} \sum_{m=0}^{N-1} \sum_{n=0}^{N-1} f(m, n) e^{-j2\pi(\frac{mu}{N} + \frac{nv}{N})} \quad (49)$$

para $u, v = 0, 1, \dots, N - 1$. $f(m, n)$ é a imagem no domínio espacial e os termos exponenciais constituem as funções de base associadas a cada ponto $F(u, v)$ no domínio da freqüência. Estas funções são componentes seno e cosseno de freqüências crescentes. Assim, $F(0, 0)$ representa a menor freqüência (ou o nível DC do sinal) e $F(N - 1, N - 1)$, a maior destas freqüências.

Ainda para o caso de imagens de dimensão $N \times N$, a DFT inversa pode ser descrita por:

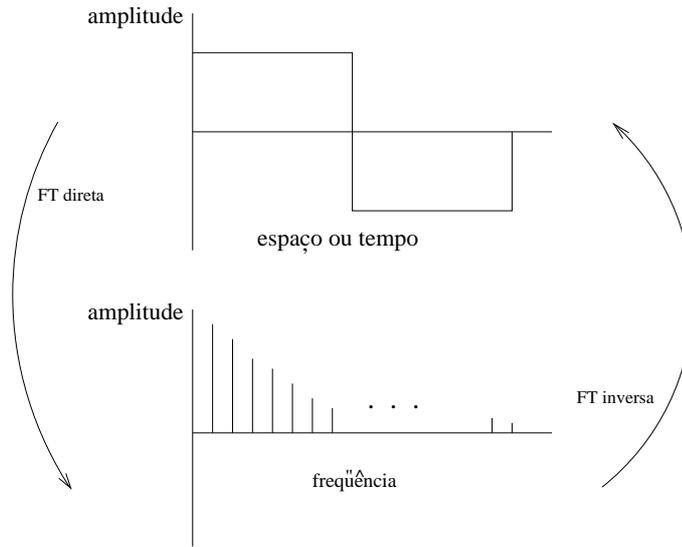


Figura 33:

$$f(x, y) = \frac{1}{N} \sum_{u=0}^{N-1} \sum_{v=0}^{N-1} F(u, v) e^{j2\pi(\frac{xu}{N} + \frac{yv}{N})} \quad (50)$$

para $x = 0, 1, \dots, N - 1$ e $y = 0, 1, \dots, N - 1$.

Observe a consideração do termo $1/N$ tanto na transformada direta como inversa (comente isto em função da linearidade da DFT).

Geralmente, o intervalo de valores representando o espectro da FT é muito grande (intervalo entre a representação das altas frequências, gerando valores de baixa intensidade, e a representação das baixas frequências que predominam no sinal). Para amenizar tal dificuldade na visualização (display) do espectro, podemos considerar a seguinte função de escalonamento

$$D(u, v) = c \log(1 + |F(u, v)|), \quad (51)$$

onde c é um fator de escala. Como podemos observar, a função acima realça o valor dos pixels associados às regiões de alta frequência, melhorando a dinâmica e, conseqüentemente, a qualidade da informação visual do espectro. Por exemplo, para um espectro variando na faixa $[0, 2.0 \times 10^5]$, $\log[1 + F(u, v)]$ define um novo valor s variando no intervalo $[0, 5.3]$. Um display deste intervalo num sistema de k bits deve considerar $c = \frac{s \cdot (2^k - 1)}{5.3}$. A Fig. 34 apresenta, respectivamente, uma imagem original, o display de $|F(u, v)|$ e uma melhor visualização deste espectro (numa escala de 0 a 255).

De modo geral, e como na Fig. 34, o display do espectro de Fourier considera o ponto $F(0, 0)$ centrado em $u_0 = v_0 = N/2$. Isto pode ser realizado facilmente a partir

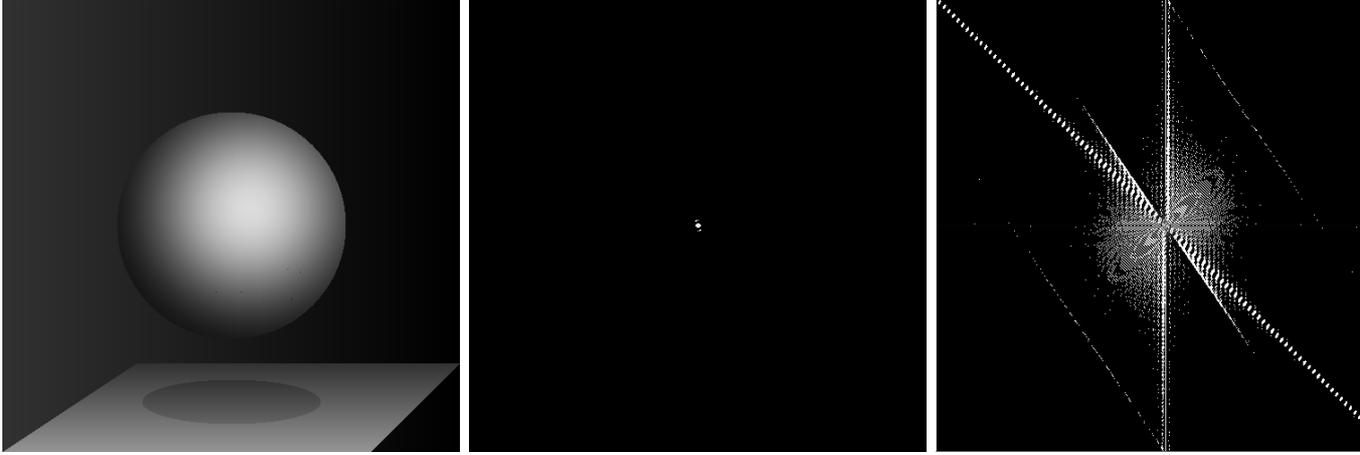


Figura 34:

das propriedades de translação ou deslocamento vistas anteriormente e que, para o caso discreto, podem ser indicadas por:

$$f(x, y)e^{j2\pi(u_0x+v_0y)/N} \Leftrightarrow F(u - u_0, v - v_0) \quad (52)$$

e

$$f(x - x_0, y - y_0) \Leftrightarrow F(u, v)e^{-j2\pi(ux_0+vy_0)/N}, \quad (53)$$

ou seja, multiplicando-se $f(x,y)$ pelo respectivo termo exponencial e calculando-se a DFT, obtemos um resultado com a origem deslocada para o ponto (u_0, v_0) , no plano da frequência. As considerações são semelhantes para a DFT inversa.

Observe que para o caso $u_0 = v_0 = N/2$, temos

$$e^{j2\pi(u_0x+v_0y)/N} = e^{j\pi(x+y)} = (-1)^{x+y} \quad (54)$$

e

$$f(x, y)(-1)^{x+y} \Leftrightarrow F(u - N/2, v - N/2), \quad (55)$$

o que fornece facilmente a translação do espectro. Observe, ainda, que o deslocamento de $f(x,y)$ não altera o valor do espectro já que:

$$|F(u, v)e^{-j2\pi(ux_0+vy_0)/N}| = |F(u, v)| \quad (56)$$

Em termos de uma primeira abordagem sobre a *implementação direta* da DFT, é importante considerar a propriedade de separabilidade da mesma. Neste caso, podemos expressar as equações 49 e 50 da seguinte forma:

$$F(u, v) = \frac{1}{N} \sum_{x=0}^{N-1} e^{-j2\pi ux/N} \sum_{y=0}^{N-1} f(x, y) e^{-j2\pi vy/N} \quad (57)$$

para $u, v = 0, 1, \dots, N-1$, e

$$f(x, y) = \frac{1}{N} \sum_{u=0}^{N-1} e^{j2\pi ux/N} \sum_{v=0}^{N-1} F(u, v) e^{j2\pi vy/N} \quad (58)$$

para $x, y = 0, 1, \dots, N-1$; ou ainda, para a Eq. 57:

$$F(u, v) = \frac{1}{N} \sum_{x=0}^{N-1} F(x, v) e^{-j2\pi ux/N}, \quad (59)$$

onde

$$F(x, v) = N \left[\frac{1}{N} \sum_{y=0}^{N-1} f(x, y) e^{-j2\pi vy/N} \right] \quad (60)$$

Este resultado mostra que $F(u, v)$ pode ser obtida considerando uma transformada 1-D, ao longo de cada linha de $f(x, y)$, definindo $F(x, v)$, seguida da mesma operação no sentido das colunas de $F(x, v)$. O mesmo desenvolvimento é válido para a Eq. 58.

Obs.: Consultar [Woo93], por exemplo, para outras propriedades interessantes da DFT.

Estudo dirigido: Verifique que o número de operações necessárias à implementação de $[F]_{N \times N} = [P]_{N \times N} [f]_{N \times N} [Q]_{N \times N}$ é proporcional a $2N^3$. Leia a seção 3.4 do livro [Woo93] para uma boa compreensão da conhecida *Fast Fourier Transform - FFT* que a partir de uma decomposição adequada da DFT define um método de implementação em que o número de operações é proporcional a $N \log_2 N$. Este é o método mais comum de implementação do par da DFT.

A seção 3.5 do mesmo livro aborda alguns aspectos relativos à propriedade de separabilidade e à definição geral de um transformada linear, vistas anteriormente. Leia esta seção para a análise de outros tipos de transformadas lineares. Em particular, a seção 3.5.3 aborda a *Transformada Discreta do Cosseno - DCT*. Após uma leitura sobre esta transformada, considere os comentários apresentados no exemplo da página 375 e aqueles abordados a partir do último parágrafo da página 378. Estes comentários estão relacionados com compressão de imagens e concernem diretamente a DCT.

16 Exemplo de aplicação: filtragem linear

Como mencionado anteriormente, a FT pode ser empregada em inúmeras aplicações de processamento de imagens, e um exemplo típico destas aplicações refere-se à *filtragem*. Com o objetivo de eliminar ou realçar determinadas componentes da imagem, tais como ruído ou contornos, a filtragem no domínio da frequência explora a característica reversível da FT e a informação contida neste domínio.

Os filtros *passa-baixas*, por exemplo, associados às regiões homogêneas da imagem, "deixam passar" as componentes de baixa frequência, atenuando as de alta frequência relacionadas com as transições bruscas da imagem. Estas transições são representadas por ruídos ou contornos, o que significa que ao eliminarmos o ruído indesejável da imagem, a partir de um filtro passa-baixas, estamos atenuando os seus contornos nas mesmas proporções, caso típico do processamento linear. Os filtros *passa-altas*, por sua vez, atenuam ou eliminam as componentes de baixas frequências. O efeito evidente desta filtragem é a obtenção de um realce nas zonas de alta frequência do sinal, isto é, dos seus contornos e – infelizmente – ruído.

Num projeto de filtros lineares podemos considerar, ainda, os filtros *passa-faixas* associados às regiões compreendidas entre as baixas e altas frequências. Estes filtros são empregados, por exemplo, em problemas de restauração de imagens. A Fig. 35 apresenta o modelo de filtros para o caso 1-D (uma rotação destas funções define os mesmos filtros para o caso 2-D).

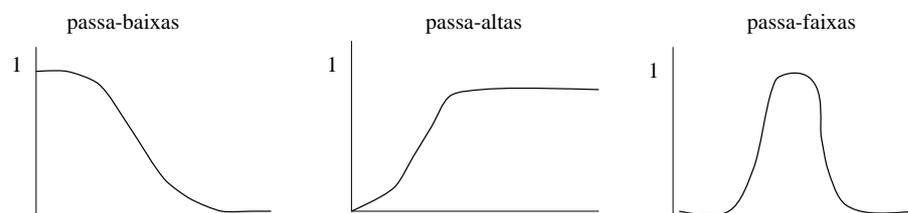


Figura 35:

O princípio geral da filtragem, no domínio da frequência, pode ser indicado pela seguinte transformação

$$G(u, v) = H(u, v)F(u, v), \quad (61)$$

onde $F(u, v)$ é a transformada de Fourier da imagem de entrada, $G(u, v)$ é a imagem de saída filtrada, e $H(u, v)$ é a respectiva *função de transferência* do filtro. O problema consiste, então, em definir a função $H(u, v)$ que conduza à imagem desejada $G(u, v)$. A transformada inversa, $\mathcal{F}^{-1}[G(u, v)]$, define a imagem filtrada no domínio espacial $g(x, y)$.

16.1 Filtros passa-baixas

Um *filtro ideal passa-baixas* 2-D pode ser representado pela seguinte função de transferência

$$H(u, v) = \begin{cases} 1 & \text{se } D(u, v) \leq D_0 \\ 0 & \text{se } D(u, v) > D_0 \end{cases} \quad (62)$$

onde $D_0 \geq 0$ é a *freqüência de corte* e $D(u, v)$ é a distância do ponto (u, v) até a origem do plano da freqüência, ou seja,

$$D(u, v) = (u^2 + v^2)^{\frac{1}{2}} \quad (63)$$

Observe que de acordo com a Eq. 62, as freqüências contidas no círculo de raio D_0 não sofrem atenuações, daí o termo filtro ideal. A Fig. 36 apresenta uma imagem original com ruído, uma "máscara" correspondente à função $H(u, v)$, e a imagem resultante do filtro passa-baixas.

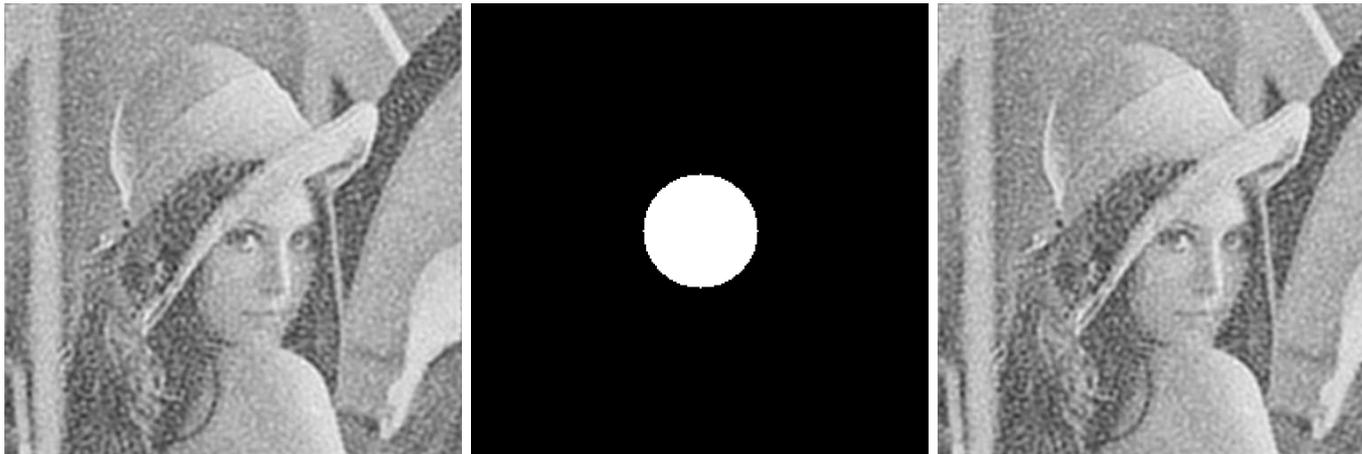


Figura 36:

Exercício: Verifique, em [Woo93], o exemplo apresentado na página 203 sobre filtro passa-baixas.

O *filtro passa-baixas de Butterworth* de ordem n é dado pela seguinte função de transferência:

$$H(u, v) = \frac{1}{1 + [D(u, v)/D_0]^{2n}} \quad (64)$$

Esta função define um filtro sem grande descontinuidade, não estabelecendo uma transição brusca na frequência de corte. Nestes casos, costuma-se definir um valor para D_0 correspondendo a uma porcentagem do máximo valor $H(u, v)$. Podemos ver facilmente que quando $D_0 = D(u, v)$ o valor de $H(u, v)$ cai para 50% do seu valor inicial. Se quisermos, por exemplo, que este valor atinja $1/\sqrt{2}$, então:

$$H(u, v) = \frac{1}{1 + [\sqrt{2} - 1][D(u, v)/D_0]^{2n}} = \frac{1}{1 + 0.414[D(u, v)/D_0]^{2n}} \quad (65)$$

16.2 Filtro passa-altas

Como mencionado anteriormente, as transições bruscas de um sinal estão associadas às componentes de alta frequência do espectro de Fourier. Assim, um realce da imagem, com ênfase nestas transições, pode ser obtido deixando-se passar estas altas frequências e atenuando-se as demais.

Um *filtro ideal passa-altas* 2-D é dado pela seguinte função de transferência:

$$H(u, v) = \begin{cases} 0 & \text{se } D(u, v) \leq D_0 \\ 1 & \text{se } D(u, v) > D_0 \end{cases} \quad (66)$$

D_0 é a frequência de corte medida a partir da origem, no plano da frequência, e $D(u, v)$ é definida como anteriormente. A Fig. 37 apresenta uma imagem original com ruído, uma "máscara" correspondente à função $H(u, v)$, e a imagem resultante do filtro passa-altas.

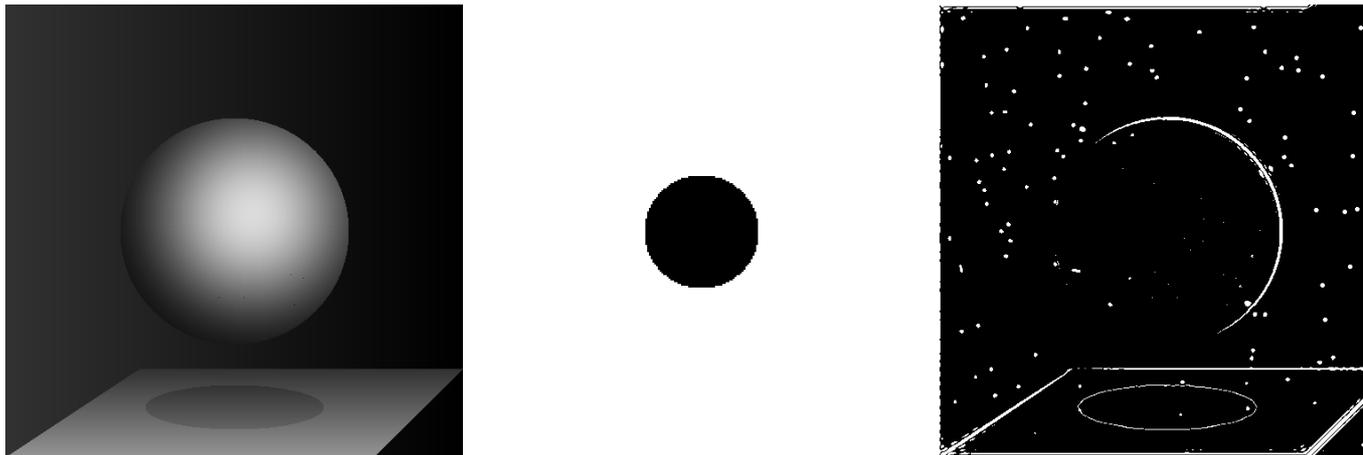


Figura 37:

O *filtro passa-altas de Butterworth* de ordem n é dado por

$$H(u, v) = \frac{1}{1 + [D_0/D(u, v)]^{2n}} \quad (67)$$

Novamente, quando $D_0 = D(u, v)$, $H(u, v)$ cai 50% do seu valor máximo.

Exercício: Verifique os resultados da filtragem apresentada na página 213 em [Woo93].

17 Filtragem linear no domínio espacial

A filtragem linear, no domínio espacial, pode ser entendida facilmente a partir da propriedade da convolução relacionada com a FT.

Para o caso discreto, a convolução descrita pela Eq. 24, pode ser expressa por

$$g(x, y) = \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} h(i, j) f(x - i, y - j), \quad (68)$$

onde h corresponde à representação espacial da função de transferência H (Eq. 61), f é a imagem de entrada $N \times N$ com ruído, e g é a imagem resultante da filtragem no domínio espacial equivalente a $\mathcal{F}^{-1}[G(u, v)]$.

De maneira geral, a função $h(x, y)$ pode ser obtida diretamente a partir da DFT inversa de $H(u, v)$, de tamanho $N \times N$. Na prática, o que se considera é uma representação reduzida de $h(x, y)$, caracterizando o tipo de filtragem desejado. Esta filtragem é definida por uma combinação linear dos pixels numa vizinhança de $f(x, y)$ através das operações indicadas pela Eq. 68.

Estudo dirigido: Leia a seção 4.5 em [Woo93] para uma abordagem sobre definição de uma máscara de convolução $h(x, y)$ a partir da DFT inversa de $H(u, v)$.

Sejam $\Gamma = (M + 1) \times (N + 1)$, M e N pares, uma vizinhança discreta centrada na origem, e uma imagem $f(x, y)$ com ruído. Uma *operação local* de filtragem linear, no domínio espacial, pode ser dada por

$$g(x, y) = \sum_{i=-m}^m \sum_{j=-n}^n h(i, j) f(x - i, y - j) \quad (69)$$

Esta equação representa uma convolução discreta da imagem f com uma *máscara* de coeficientes h . Assim, em termos de implementação, a operação de filtragem pode ser

obtida a partir de um deslocamento de h sobre as linhas de f (ou vice-versa), no sentido da *varredura direta* da imagem (da esquerda para a direita, e de cima para baixo). Os coeficientes de h definem o tipo da filtragem: suavização ou realce de contornos, por exemplo.

17.1 Alguns exemplos: filtros passa-baixas

O *filtro da média* é um exemplo típico de uma filtragem passa-baixas no domínio espacial. Para uma vizinhança elementar 3×3 , por exemplo, a máscara h pode ser dada por:

$$h = \frac{1}{9} \begin{vmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{vmatrix} \quad (70)$$

É fácil constatar o efeito de redução do ruído após tal filtragem, assim como o problema de suavização dos contornos da imagem que, naturalmente, se torna mais grave à medida que a máscara h aumenta de dimensão. A Fig. 38 ilustra a filtragem linear de uma imagem com uma máscara 11×11 .

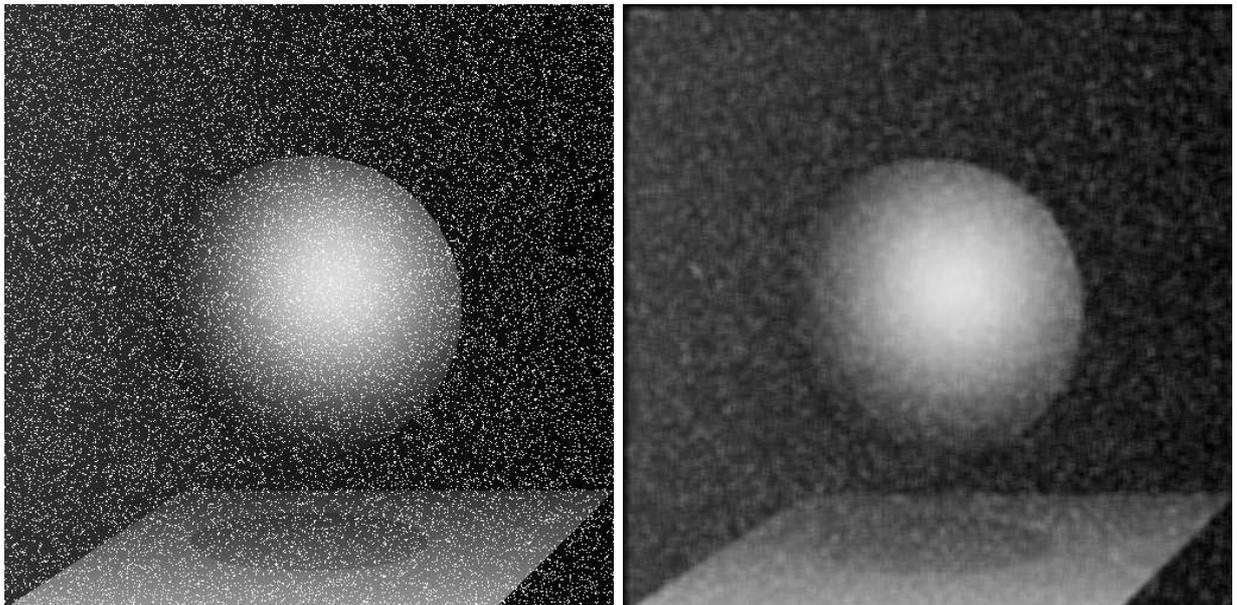


Figura 38:

Os valores de h podem ser definidos a partir da consideração do modelo de um determinado tipo de ruído, visando expressar aproximações deste modelo. As máscaras, a seguir, são definidas considerando o ruído com uma distribuição de probabilidade gaussiana, distribuição esta muito utilizada na prática.

$$h_1 = \frac{1}{16} \begin{vmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{vmatrix} \quad (71)$$

17.2 Detectores de contorno

Os detectores de contorno identificam transições bruscas na função $f(x, y)$. A máscara 3×3 , a seguir, é um exemplo de um detector de contorno espacial.

$$h = \begin{vmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{vmatrix} \quad (72)$$

Esta máscara, cujo somatório dos coeficientes é igual a zero, define valores nulos nas regiões homogêneas da imagem e valores mais elevados próximo aos seus contornos.

17.2.1 Operadores diferenciais

Os operadores diferenciais, tais como o *gradiente*, são largamente empregados na detecção de contornos [Kak82]. Para um sinal 2-D, identificamos mudanças da função em várias direções (derivadas parciais em torno dos eixos x e y , por exemplo). Neste caso, definimos um vetor gradiente do tipo:

$$\nabla f = \begin{bmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{bmatrix} \quad (73)$$

A magnitude deste vetor é

$$|\nabla f| = \left[\left(\frac{\partial f}{\partial x} \right)^2 + \left(\frac{\partial f}{\partial y} \right)^2 \right]^{1/2} \quad (74)$$

Esta equação define a intensidade do gradiente cuja direção, perpendicular à direção do contorno, é dada por

$$\theta = \text{tg}^{-1} \left[\frac{\nabla y}{\nabla x} \right] \quad (75)$$

17.2.2 Exemplos

Para o caso discreto, o operador acima é definido a partir de diferenças entre pixels, numa determinada vizinhança. O conhecido *gradiente de Roberts*, por exemplo, é representado

por (Fig. 39)

$$\nabla_{d1}f(x, y) = f(x, y) - f(x + 1, y + 1) \quad (76)$$

$$\nabla_{d2}f(x, y) = f(x, y + 1) - f(x + 1, y) \quad (77)$$

Este operador responde a mudanças nas direções diagonais (45° e 135°) da imagem, considerando pares de pontos centrados em $(x + 1/2, y + 1/2)$. O gradiente de Roberts pode ser representado, ainda, pelas seguintes máscaras de convolução:

$$\nabla_{d1} = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \quad \nabla_{d2} = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} \quad (78)$$

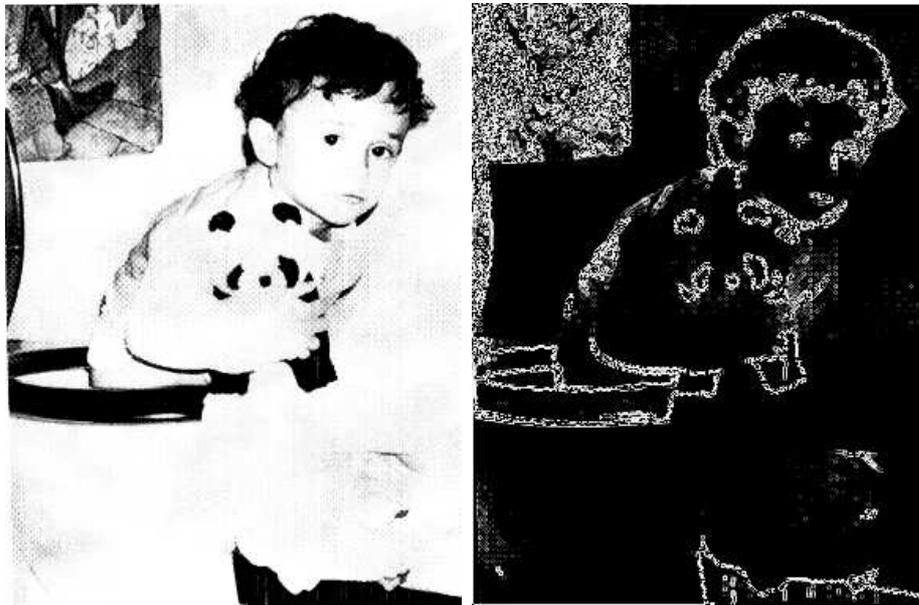


Figura 39:

Outras aproximações para a magnitude do gradiente, indicando descontinuidades nas direções principais x e y , podem ser obtidas a partir dos operadores de *Prewitt* e *Sobel* definidos, respectivamente, pelas seguintes máscaras de convolução:

$$\nabla_x = \begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix} \quad \nabla_y = \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix} \quad (79)$$

e

$$\nabla_x = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix} \quad \nabla_y = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \quad (80)$$

Para as diferenças nas direções ortogonais ∇_x e ∇_y , acima, a intensidade do contorno no ponto (x, y) é (Eq. 74):

$$|\nabla f(x, y)| = \sqrt{\nabla_x^2 + \nabla_y^2} \quad (81)$$

A magnitude do gradiente pode ser aproximada por

$$|\nabla f(x, y)| = \max(|\nabla_x|, |\nabla_y|) \quad (82)$$

ou

$$|\nabla f(x, y)| = |\nabla_x| + |\nabla_y| \quad (83)$$

Estas aproximações evitam o cálculo da raiz quadrada e, para o caso da Eq. 82, a introdução de novos valores de níveis de cinza fora do intervalo original. A Fig. 40 apresenta uma imagem seguida da detecção de seus contornos nas direções x e y , utilizando o operador de Prewitt e o cálculo da magnitude indicado pela Eq. 82.

O Laplaciano

O *Laplaciano* 2-D de uma função f é um operador diferencial linear dado por

$$\nabla^2 f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2} \quad (84)$$

Para o caso discreto, podemos mostrar que

$$\nabla^2 f(x, y) = \nabla_x^2 f(x, y) + \nabla_y^2 f(x, y) = [f(x+1, y) + f(x-1, y) + f(x, y-1) + f(x, y+1)] - 4f(x, y), \quad (85)$$

o que corresponde a uma convolução de f com a máscara

$$h = \begin{vmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{vmatrix} \quad (86)$$

Observe que, como esperado, o Laplaciano é zero para uma rampa linear e que o mesmo responde na entrada e na saída dos pontos de transição com valores positivos e negativos que se alternam. O módulo, $|\nabla^2 f|$, ou apenas a parte positiva, $|\nabla^2 f| \geq 0$, podem ser considerados como resultado do Laplaciano. No caso em que as respostas positivas e negativas são consideradas, um dado interessante associado a este operador é a *passagem por zero* (zero-crossing) da função.

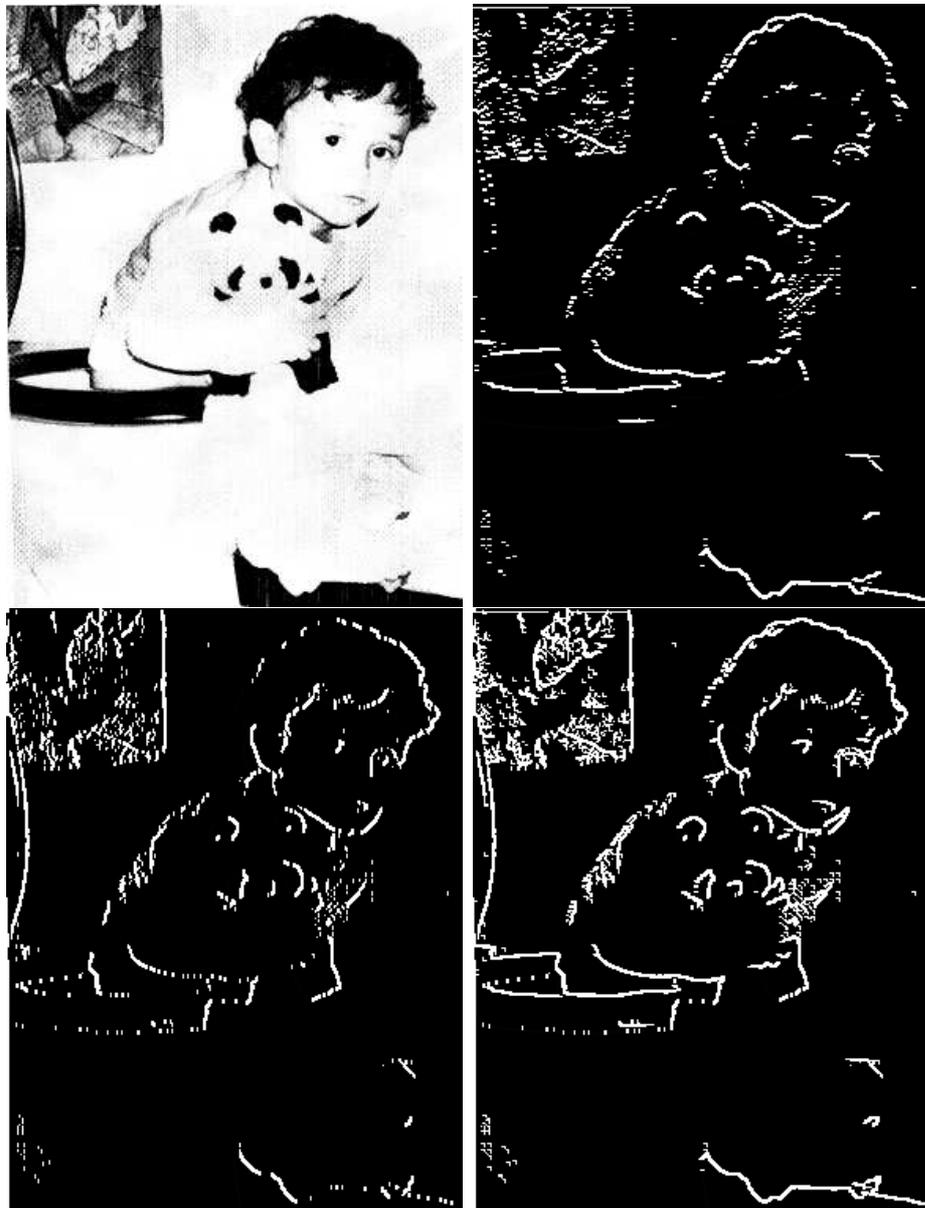


Figura 40:

Exercício 1: Mostre que o Laplaciano é proporcional à diferença entre uma imagem f e sua versão filtrada, \bar{f} , considerando os valores do ponto (x, y) e dos seus quatro vizinhos mais próximos.

Exercício 2: Verifique ???????

18 Filtragem não-linear

Como vimos, um dos grandes problemas relacionados com a eliminação do ruído de uma imagem, através de uma filtragem linear, refere-se à suavização dos seus contornos. O processamento não-linear aborda este problema tentando evitar uma filtragem homogênea ao longo das regiões próximas a estes contornos. Uma classe de filtros não-lineares bastante empregada na eliminação de ruídos, com preservação de contornos, são os *filtros estatísticos da ordem* [Kak82] [ea83]. Dentre estes, um dos mais importantes é o *filtro da mediana* que consiste em substituir o valor de um pixel m , centrado numa determinada vizinhança, pelo valor mediano dos pixels desta vizinhança ordenados de acordo com suas magnitudes [Hey82]. O filtro da mediana elimina eficientemente *ruído impulsivo*, do tipo *sal e pimenta*, representando descontinuidades abruptas e isoladas na imagem. Além disto, ele não introduz valores de níveis de cinza diferentes daqueles contidos na imagem original e, por afetar menos os contornos, pode ser aplicado iterativamente. A Fig. 41 apresenta uma imagem com ruído e uma iteração do filtro da mediana com uma vizinhança 5×5 .

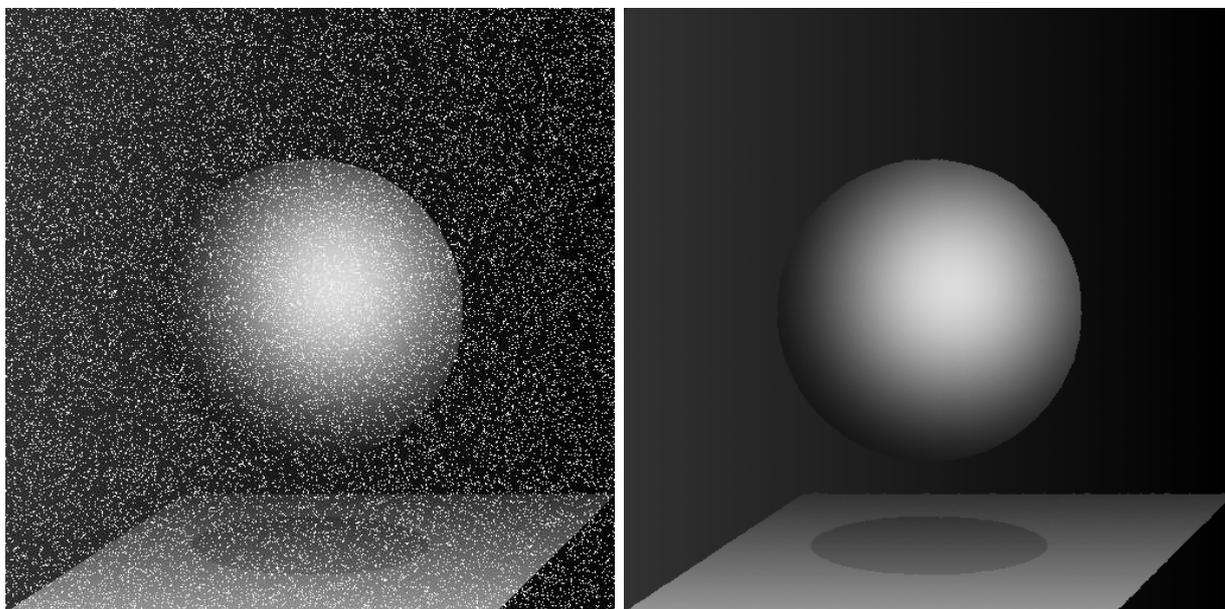


Figura 41:

Outros casos específicos dos filtros estatísticos da ordem são os valores *máximo* e

mínimo da seqüência ordenada de pixels. Como veremos, isto representa uma generalização das operações de *dilatação* e *erosão* morfológicas discutidas a seguir.

19 Morfologia Matemática

Um dos grandes problemas em processamento e análise de imagens refere-se à extração das componentes de interesse contidas na cena original. A análise do conteúdo desta cena depende de duas etapas principais. Uma relativa à *segmentação*, consistindo de transformações geralmente dependentes da aplicação, e outra referente à *análise quantitativa* dos objetos segmentados. Esta etapa visa a extração de características dos objetos a serem consideradas num processo de *classificação*.

Morfologia Matemática (MM) representa uma técnica não-linear de processamento de imagens e define um conjunto de ferramentas que auxiliam na realização destas etapas. Formalizada a partir dos anos 60 com os trabalhos de G. Matheron e J. Serra [Mat75], [Ser82], [Ser88], [Bar94], do Centro de Morfologia Matemática da Escola de Minas de Paris, ela tem como fundamento a teoria dos conjuntos e a geometria integral, e é utilizada atualmente no formalismo de diversos problemas práticos e teóricos de processamento e análise de imagens.

Como veremos, a idéia básica da MM consiste da comparação do conteúdo de uma imagem com outra menor e de forma conhecida, denominada *elemento estruturante*. De modo geral, este elemento contém características geométricas e/ou topológicas relacionadas com a informação que pretendemos extrair da imagem de interesse.

20 As transformações morfológicas

As transformações morfológicas podem ser consideradas sobre imagens binárias ou em níveis de cinza, vistas como conjuntos e funções, respectivamente. A estas imagens podemos associar os operadores booleanos de *união* \cup e *interseção* \cap , no caso de conjuntos, e os operadores *max* e *min*, no caso de funções.

As transformações morfológicas dispõem de algumas propriedades algébricas e topológicas que são importantes caracterizar. Seja ψ uma transformação agindo sobre imagens binárias ou em níveis de cinza. Mais distintamente, para imagens binárias, consideramos subconjuntos $\mathcal{P}(\mathbb{R}^2)$ de \mathbb{R}^2 e $\psi : \mathcal{P}(\mathbb{R}^2) \rightarrow \mathcal{P}(\mathbb{R}^2)$. Neste caso, a relação de ordem é a inclusão. Para imagens em níveis de cinza, ψ opera sobre funções de \mathbb{R}^2 em \mathbb{R} . Se \mathcal{F} designa o conjunto destas funções, ou seja, $\mathcal{F} = \{f : \mathbb{R}^2 \rightarrow \mathbb{R}\}$, então $\psi : \mathcal{F} \rightarrow \mathcal{F}$. Neste caso, a relação de ordem é:

$$\forall f, g \in \mathcal{F}, f \leq g \Leftrightarrow \forall x \in \mathbb{R}^2, f(x) \leq g(x) \quad (87)$$

É importante lembrar que, na prática, as imagens são processadas no plano discreto \mathcal{Z}^2 , com funções assumindo valores em \mathcal{Z} . Uma seqüência de imagens pode ser definida como uma função em \mathcal{Z}^3 em que a terceira variável representa o tempo, por exemplo.

A transformação morfológica ψ é dita *extensiva* se, e somente se

$$\forall X \in \mathcal{P}(\mathbb{R}^2), X \subseteq \psi(X) \quad \text{ou} \quad \forall f \in \mathcal{F}, f \leq \psi(f) \quad (88)$$

e *anti-extensiva* se, e somente se

$$\forall X \in \mathcal{P}(\mathbb{R}^2), \psi(X) \subseteq X \quad \text{ou} \quad \forall f \in \mathcal{F}, \psi(f) \leq f \quad (89)$$

A mesma transformação é dita *crescente* se ela preserva a ordem no espaço $\mathcal{P}(\mathbb{R}^2)$, ou seja,

$$\forall X, Y \in \mathcal{P}(\mathbb{R}^2), X \subseteq Y \Rightarrow \psi(X) \subseteq \psi(Y) \quad \text{ou} \quad (90)$$

$$\forall f, g \in \mathcal{F}, f \leq g \Rightarrow \psi(f) \leq \psi(g) \quad (91)$$

Em caso contrário ($X \subseteq Y \Rightarrow \psi(Y) \subseteq \psi(X)$ ou $f \leq g \Rightarrow \psi(f) \geq \psi(g)$), ela é dita *decrecente*. A operação ψ é *idempotente* se

$$\psi \circ \psi = \psi, \quad (92)$$

ou seja, o resultado de várias iterações do operador ψ sobre a imagem equivale ao resultado de uma única aplicação.

Duas transformações ψ_1 e ψ_2 são *duais* se, e somente se

$$\forall X \in \mathcal{P}(\mathbb{R}^2), \psi_1(X) = (\psi_2(X^c))^c \quad \text{ou} \quad \forall f \in \mathcal{F}, \psi_1(f) = -(\psi_2(-f)) \quad (93)$$

Finalmente, uma transformação é dita *homotópica* se ela não modifica o número de conexidade N_n de um conjunto X , isto é:

$$N_n(\psi(X)) = N_n(X) \quad (94)$$

21 Erosão e dilatação

Erosão e dilatação são as operações elementares da MM e formam a base para a construção das transformações mais complexas. Assim, numa cadeia morfológica de processamento de imagens, podemos encontrar um grande número de operadores encadeados, todos definidos a partir destas funções elementares.

Seja X o conjunto inicial dos pontos que constituem a imagem binária a ser analisada, e B uma pequena imagem representando um certo conhecimento geométrico e/ou topológico

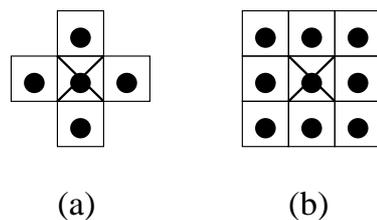


Figura 42: Exemplos de elementos estruturantes numa vizinhança 3×3 .

de X . Ao elemento estruturante B associamos um ponto *origem* representando, na imagem, a posição de referência para o resultado das transformações. A Fig. 42 apresenta dois exemplos de elementos estruturantes de dimensão 3×3 com origem no seu centro. A Fig. 42(a) define um elemento *4-conexo*, e a Fig. 42(b), um elemento *8-conexo*.

A dilatação de X por B , $\delta_B(X)$, é dada pela união de todos os pontos x de \mathfrak{R}^2 tal que o elemento estruturante B , centrado em x , B_x , intercepta X . Assim:

$$\delta_B(X) = \{x \in \mathfrak{R}^2, B_x \cap X \neq \emptyset\} \quad (95)$$

A Fig. 43 apresenta o resultado de duas iterações de dilatação de uma imagem binária com o elemento estruturante 3×3 indicado na Fig. 42(b). Aqui, as imagens contêm apenas dois valores (imagens binárias): o valor "1" representando o conjunto X de pontos claros da *forma* (foreground), e o valor "0" correspondente ao complemento de X , X^c , e representando os pontos escuros do *fundo* (background) da imagem.



Figura 43: Imagem original (a), e imagem dilatada (b).

A erosão de X por B , ϵ_B , é definida pelo conjunto dos pontos de x em \mathfrak{R}^2 tal que B , ao ser transladado para a posição x , esteja totalmente incluído no conjunto X . Assim:

$$\epsilon_B(X) = \{x \in \mathfrak{R}^2, B_x \subset X\} \quad (96)$$

A Fig. 44 apresenta o resultado de duas iterações de erosão de uma imagem binária com o elemento estruturante da Fig. 42(b).

No caso de funções, e considerando o exemplo prático de elementos estruturantes com valores iguais a zero no seu domínio $k \in \mathfrak{R}^2$ (*elemento estruturante planar*), a dilatação é dada simplesmente por:

$$\delta_B(f)(x) = \max\{x_k, k \in B\}, \quad (97)$$



Figura 44: Imagem original (a), e imagem erodida (b).

e a erosão:

$$\epsilon_B(f)(x) = \min\{x_k, k \in B\} \quad (98)$$

A Fig. 45 apresenta um exemplo de dilatação e erosão de uma imagem em níveis de cinza com um elemento estruturante de dimensão 5×5 . Como podemos observar, a dilatação amplia as zonas claras da imagem e a erosão realiza o processo inverso, aumentando o domínio das regiões escuras.

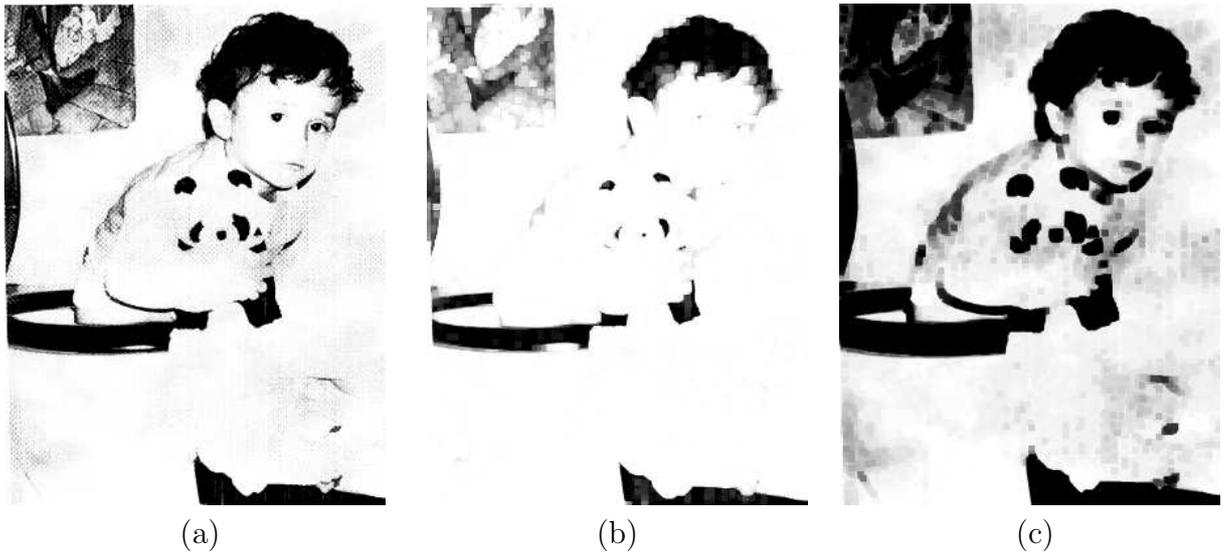


Figura 45: Imagem original (a), imagem dilatada (b), e imagem erodida (c).

É fácil constatar que erosão e dilatação são operações crescentes e duais, e que ambas não são nem idempotentes nem homotópicas. Observe, ainda, que para um elemento B de tamanho n , B_n , definido a partir de n iterações de dilatação com um elemento estruturante de tamanho unitário, B_1 (por exemplo, os elementos da Fig 42), então a erosão (dilatação) de um conjunto X com B_n pode ser decomposta em uma seqüência de n erosões (dilatações) de X com B_1 . Assim:

$$\delta_{B_1} \circ \delta_{B_2} = \delta_{\delta_{B_1}(B_2)} \quad e \quad (99)$$

$$\epsilon_{B_1} \circ \epsilon_{B_2} = \epsilon_{\delta_{B_1}(B_2)} \quad (100)$$

22 Abertura e fechamento morfológicos

As operações de *abertura* e *fechamento* com um elemento estruturante B , denotadas respectivamente por γ_B e φ_B , são obtidas combinando-se erosão e dilatação da seguinte forma:

$$\gamma_B = \delta_{\tilde{B}} \circ \epsilon_B, \quad (101)$$

e

$$\varphi_B = \epsilon_{\tilde{B}} \circ \delta_B, \quad (102)$$

onde \tilde{B} é o transposto de B , ou seja, o simétrico de B em relação a sua origem. Naturalmente, no caso de B simétrico, temos

$$\gamma_B = \delta_B \circ \epsilon_B$$

e

$$\varphi_B = \epsilon_B \circ \delta_B$$

Abertura e fechamento morfológicos são operações duais, crescentes e idempotentes. A abertura é anti-extensiva e o fechamento, extensivo.

Em termos visuais, a abertura regulariza os contornos e elimina pequenas "ilhas" e "cabos" estreitos de uma imagem binária. O fechamento suprime pequenos "lagos" e "canais" estreitos (Fig. 46). Da mesma forma, para imagens em níveis de cinza, a abertura elimina estruturas claras, enquanto o fechamento atua sobre as estruturas escuras. Tudo isto em função da forma e dimensão do elemento estruturante considerado. Observe, finalmente, que estas operações não são homotópicas.

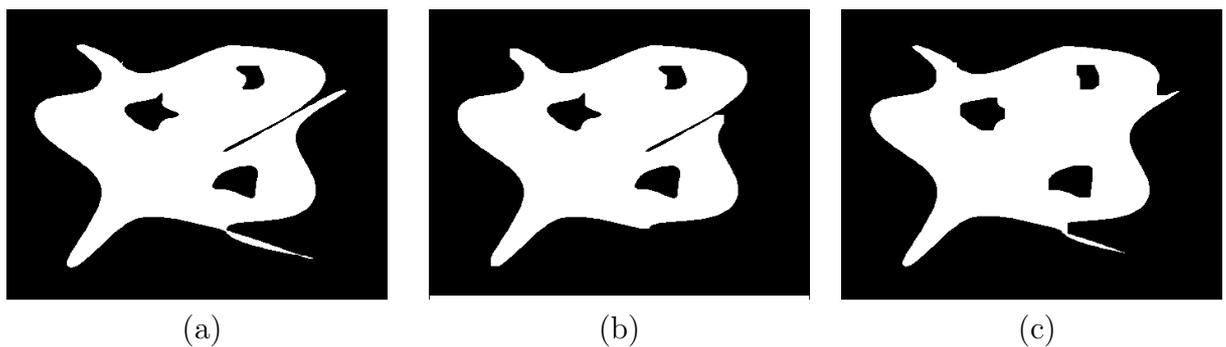


Figura 46: Imagem binária (a), abertura (b), e fechamento de tamanho 5 (c).

23 Filtros morfológicos

Os filtros morfológicos constituem alternativas interessantes à filtragem não-linear e são caracterizados por uma transformação ψ *crescente* e *idempotente* [Ser94]. Assim:

$$\forall (f, g) \in \mathcal{F} \times \mathcal{F}, f \leq g \Rightarrow \psi(f) \leq \psi(g) \quad (103)$$

$$\forall f \in \mathcal{F}, \psi(f) = \psi(\psi(f)) \quad (104)$$

Neste sentido, a erosão e a dilatação, por não serem idempotentes, não constituem filtros morfológicos. Por sua vez, a abertura e o fechamento são exemplos típicos de operadores de filtragem, cuja composição também define filtros. Esta composição pode ser considerada a partir de aberturas e fechamentos com elementos estruturantes de tamanho variável o que conduz, entre outros, à definição de filtros com comportamento simétrico em relação às estruturas claras e escuras da imagem [Ser88], [Che89]. A Fig. 47 ilustra uma filtragem por abertura.

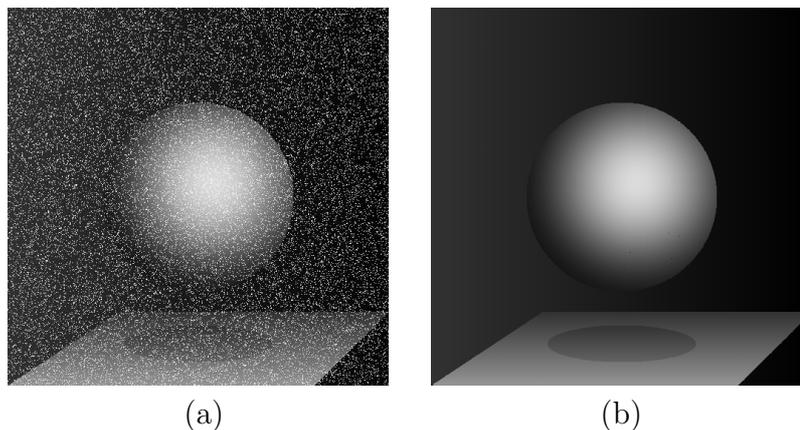


Figura 47: Imagem original com ruído (a), e exemplo de filtragem por abertura (b).

23.1 Outros exemplos de operações elementares

O conjunto das transformações morfológicas pode ser obtido a partir da combinação dos operadores apresentados anteriormente. Como primeiro exemplo destas transformações, podemos citar o resíduo da dilatação e erosão que define o *gradiente morfológico* simétrico (Fig. 48):

$$grad(f) = \delta_B(f) - \epsilon_B(f) \quad (105)$$

Da mesma forma, os gradientes interno e externo são dados, respectivamente, por

$$grad^-(f) = f - \epsilon_B(f) \quad (106)$$

$$grad^+(f) = \delta_B(f) - f \quad (107)$$

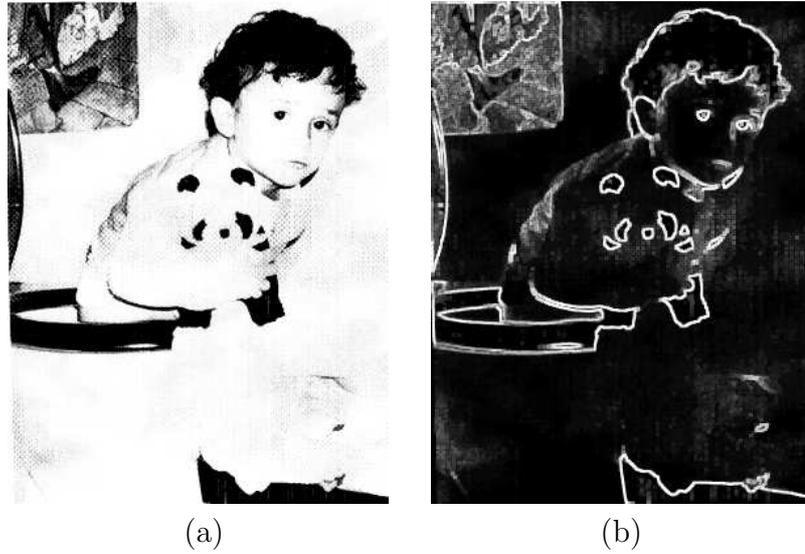


Figura 48: Imagem original (a), e o seu gradiente morfológico (b).

Outra transformação importante é o *chapéu mexicano claro* (white top-hat) que detecta as estruturas claras da imagem eliminadas por uma abertura de tamanho n (n iterações com um elemento estruturante elementar).

$$CM_n^+(f) = f - \gamma_n(f) \quad (108)$$

O *chapéu mexicano escuro* (black top-hat), associado às estruturas escuras da imagem, é dado por

$$CM_n^-(f) = \varphi_n(f) - f \quad (109)$$

A Fig. 49 ilustra estas transformações que são empregadas, entre outros, na definição do esqueleto morfológico de uma imagem [Ser88].

24 A transformação Tudo ou Nada

A transformação Tudo ou Nada (Hit or Miss) considera as duas fases (pontos "1" e pontos "0") do elemento estruturante B . Sejam B_x^1 e B_x^2 estas fases com a mesma origem centrada em $x \in X$. A transformação Tudo ou Nada de X com B , $X \otimes B$, é dada por

$$X \otimes B = \{x \in \mathfrak{R}^2 : B_x^1 \subset X, B_x^2 \subset X^c\} \quad (110)$$

ou ainda

$$X \otimes B = \epsilon_{B_x^1}(X) \setminus \delta_{B_x^2}(X) = \epsilon_{B_x^1}(X) \cap \epsilon_{B_x^2}(X^c) \quad (111)$$

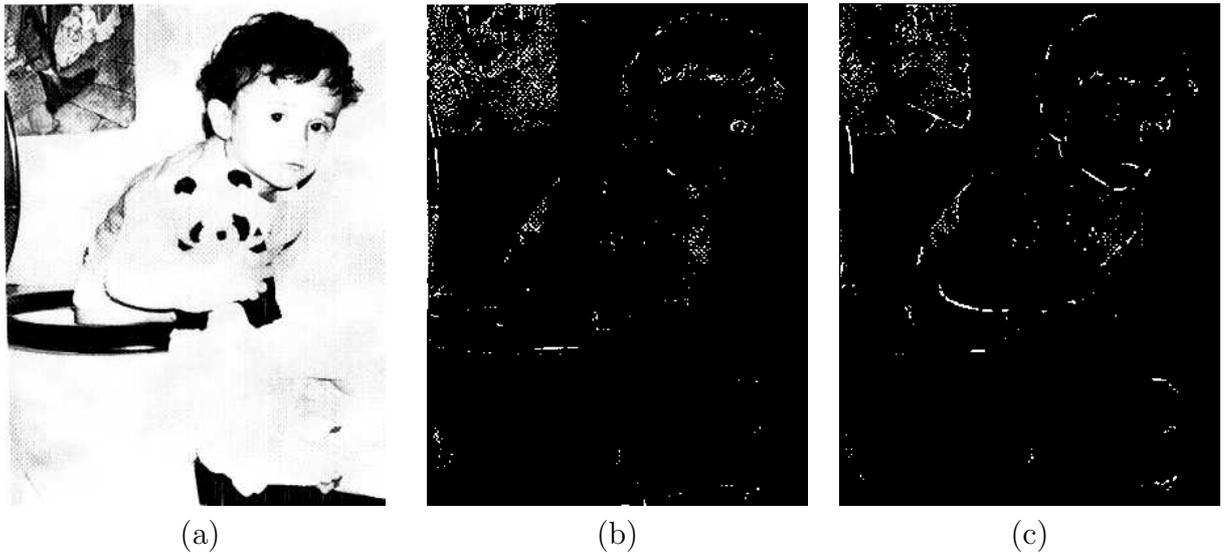


Figura 49: Imagem original (a), o chapéu mexicano claro (b), e o escuro (c).

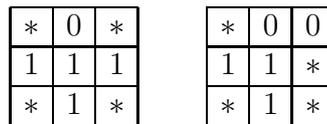


Figura 50: Exemplo de elementos estruturantes homotópicos de afinamento.

24.1 Afinamento e espessamento homotópicos

O afinamento e o espessamento homotópicos são exemplos de aplicações obtidas a partir da transformação do tipo Tudo ou Nada. O afinamento de uma imagem X com um elemento estruturante B , $X \ominus B$, define uma representação homotópica do esqueleto de X (Fig. 51), e é dado por

$$X \circ B = X \setminus (X \otimes B) \quad (112)$$

O afinamento elimina pontos da borda de X e deve ser aplicado iterativamente até a idempotência, considerando-se uma seqüência de elementos estruturantes que preservem a topologia da imagem. A Fig. 50 apresenta exemplos destes elementos que, juntamente com suas rotações de 90° , devem ser utilizados alternadamente no afinamento em uma malha retangular (o $*$ indica que o ponto pode assumir o valor "0" ou "1" na respectiva posição).

O espessamento é a operação dual do afinamento e é dado por

$$X \odot B = X \cup (X \otimes B) \quad (113)$$

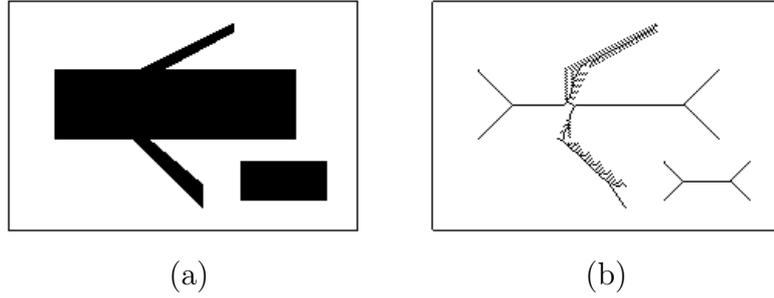


Figura 51: Imagem original (a), e o resultado do afinamento com as máscaras da Fig. 50 (b).

O espessamento homotópico adiciona pontos às componentes da imagem sem que a sua topologia seja alterada. É fácil verificar que os elementos estruturantes empregados aqui são o complemento daqueles utilizados na operação de afinamento e que, por dualidade, esta transformação define o esqueleto do fundo da imagem.

A abordagem anterior pode ser estendida a funções. Assim, seja f uma imagem em níveis de cinza e B_x^1 , B_x^2 os subconjuntos do elemento estruturante planar representando, respectivamente, as fases "1" e "0" de B . O afinamento de f por B , $f \circ B$, é dado por

$$(f \circ B) = \begin{cases} \delta_{B_x^2}(f), & \text{se } \delta_{B_x^2}(f) < f \leq \epsilon_{B_x^1}(f), \\ f, & \text{caso contrário} \end{cases} \quad (114)$$

O afinamento numérico define a zona de influência dos *mínimos regionais* de uma imagem, salientando os picos das "montanhas" que constituem o seu relevo. Esta operação pode ser associada à transformação que calcula a linha divisora de águas de uma função e que, como veremos posteriormente, constitui a base dos algoritmos morfológicos de segmentação.

25 Operações geodésicas e reconstrução

Sejam x e y dois pontos quaisquer de um conjunto X . O arco geodésico de extremidade xy é definido como o arco de menor distância $d_X(x, y)$ entre x e y , contido em X [Mai84]. Caso este arco não exista, esta distância é considerada infinita (Fig. 52). A função d_X é denominada distância geodésica.

Como discutido em [Mai84], as transformações morfológicas básicas podem ser definidas no espaço métrico (X, d_X) . Por exemplo, se $Y \subset X$, pontos x de X tal que $B_X(x, \lambda)$ constituem o conjunto λ -*dilatado* de Y em X . Este conjunto é denotado por

$$\delta_X^\lambda(Y) = \{x \in X : B_X(x, \lambda) \cap Y \neq \emptyset\} \quad (115)$$

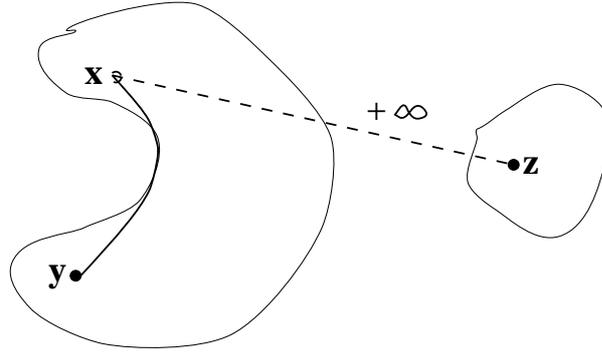


Figura 52: Noção de distância geodésica.

onde $B_X(x, \lambda) = \{x \in X: d_X(x, y) \leq \lambda\}$

Na prática, a dilatação geodésica de tamanho n , no espaço Z^2 , pode ser obtida a partir de n iterações da dilatação geodésica de tamanho 1 definida por

$$\delta_X^1(Y) = (Y \oplus B) \cap X, \quad (116)$$

e assim:

$$\delta_X^n(Y) = \underbrace{\delta_X^1(\delta_X^1(\dots\delta_X^1(Y)))}_{n \text{ vezes}} \quad (117)$$

O elemento estruturante B pode ser aqueles indicados na Fig. 42.

A erosão geodésica de Y em X é dada por

$$\epsilon_X^\lambda(Y) = \{x \in X : B_x(x, \lambda) \subseteq Y\} \quad (118)$$

As transformações geodésicas são empregadas na reconstrução de um conjunto conexo X marcado por outro conjunto de referência Y , não vazio, denominado *marcador*. O conjunto X é a máscara representando a componente binária que pretendemos preservar. Uma dilatação geodésica de Y em X , até a idempotência, permite a reconstrução total desta componente. Como exemplo, consideremos o problema de eliminação de partículas parcialmente incluídas numa imagem [Che89] (Fig. 53).

Se $\partial\Sigma$ representa o subconjunto de pontos pertencentes à fronteira de uma imagem Σ , então o novo subconjunto X' , contendo componentes totalmente incluídas em X , é dado por

$$X' = X \setminus \delta_X^\infty(\partial\Sigma) \quad (119)$$

$\delta^\infty(\cdot)$ indica a execução da dilatação até a idempotência.

A noção de reconstrução geodésica é muito importante para o processamento morfológico, e constitui a base de inúmeras transformações mais complexas relacionadas, por

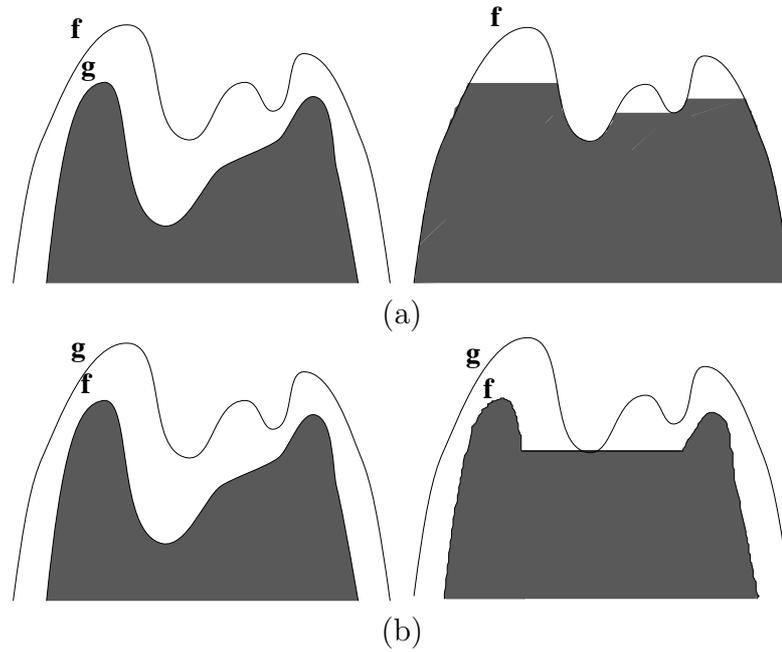


Figura 54: Exemplo de reconstrução geodésica (a), e reconstrução dual (b).

$$IZ_X(Y) = \bigcup_{i \in [1, k]} iz_X(Y_i) \quad (124)$$

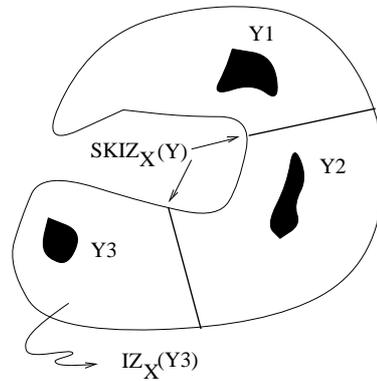


Figura 55: Esqueleto por zona de influência de Y em X .

27 Segmentação morfológica

A principal ferramenta de segmentação morfológica é baseada na transformação que define a linha divisora de águas - LDA ou *watershed* de uma função [Mey93], [Soi91]. Intuiti-

vamente, esta LDA pode ser obtida da seguinte forma. Considere uma imagem em níveis de cinza representada por uma superfície topográfica em que os níveis indicam a altitude do ponto no relevo. Suponha, agora, que os mínimos em cada uma das bacias (mínimos regionais) que constituem a superfície sejam perfurados, e que a imergimos em um lago com velocidade vertical constante. A água que penetra regularmente pelos orifícios preenche a superfície topográfica e, durante o preenchimento, dois ou mais fluxos vindos de mínimos diferentes podem se unir. Os diques construídos na superfície para evitar que tal junção de águas ocorra constituem a LDA da imagem.

Existem algoritmos eficientes que calculam a função LDA a partir da simulação deste processo de preenchimento de bacias [Mey93], [Soi91]. Um algoritmo iterativo baseado na definição do SKIZ considera os seguintes passos (ver [Vin93] para uma implementação do método). Seja o conjunto X_i obtido por limiarizações sucessivas de uma imagem f com N níveis de cinza:

$$X_i = \{x \in \mathcal{Z}^2, f(x) \leq i\} \quad i = 1, \dots, N \quad (125)$$

Se o conjunto $SKIZ_X(Y)$ indica, como anteriormente, o esqueleto por zona de influência de Y em X (seção 26), então o conjunto Z , representando a LDA de f , pode ser dado iterativamente por

$$\begin{aligned} Z_j &= SKIZ_{X_j}(X_{j-1} \setminus Z_{j-1}) \quad j = 1, \dots, N \\ Z &= \bigcup_j Z_j \end{aligned} \quad (126)$$

O algoritmo de LDA define a zona de influência de todos os mínimos regionais de uma imagem e, neste sentido, a sua aplicação resulta em uma super-segmentação da mesma. A Fig. 57(c) apresenta a imagem de LDA do gradiente, Fig. 57(b), calculado sobre a imagem de grãos de feijão na Fig. 57(a). Cada região da Fig. 57(c), delimitada por uma LDA, corresponde à zona de influência do mínimo mais profundo nesta região.

De modo geral, o modelo de segmentação utilizando o algoritmo de LDA é constituído de duas etapas principais: uma "inteligente" que consiste da obtenção de marcadores associados às regiões que pretendemos segmentar, e a outra, automática, representada pelo cálculo da LDA sobre uma imagem f que considera a informação destes marcadores. Esta imagem pode ser o chapéu mexicano, a própria imagem original ou o seu gradiente, por exemplo. O objetivo da primeira etapa, específica à aplicação, é eliminar mínimos não significativos da imagem diminuindo, assim, o efeito da super-segmentação. A segunda etapa define uma imagem de "relevo" adequado à segmentação das regiões. O gradiente, por exemplo, que atribui valores elevados aos pontos de contorno das regiões e mínimos no seu interior pode ser empregado na segmentação da Fig. 57(a).

De posse destas imagens, o próximo passo consiste em impor mínimos na imagem gradiente, na posição indicada pelos marcadores obtidos na primeira etapa. Esta imposição especifica as regiões da imagem a serem perfuradas no processo de inundação. As bacias que não forem marcadas não definirão zonas de influência e, conseqüentemente, não constituirão LDA.

Para a imposição dos mínimos, precisamos considerar uma imagem g definida da seguinte forma:

$$g(x) = \begin{cases} +\infty & \text{se } x \notin M \\ 0 & \text{se } x \in M \end{cases} \quad (127)$$

onde M representa o conjunto dos marcadores. Para o cálculo da LDA das regiões marcadas por M , efetuamos inicialmente uma reconstrução geodésica dual (secção 25) de g em $\min(f, g)$: $\epsilon_{\min(f,g)}^\infty(g)$. A Fig. 56 ilustra estas operações. Finalmente, o algoritmo de LDA pode ser aplicado nesta imagem reconstruída, originando uma segmentação mais adequada. Um algoritmo eficiente de segmentação por LDA considerando marcadores é apresentado em [Mey93].

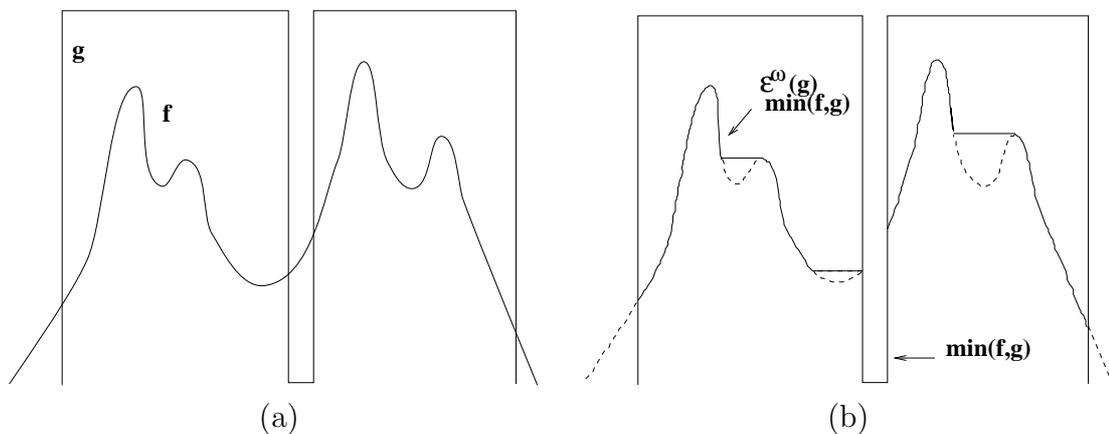


Figura 56: Imposição de mínimos (a), e reconstrução dual (b).

A Fig. 57 apresenta uma aplicação simples deste método. Os marcadores conexos M (Fig. 57(d)), associados a cada um dos grãos, foram obtidos facilmente por limiarização da imagem original [Kak82]. Por exemplo, todos os pixels com nível de cinza abaixo de um determinado valor (30, neste caso) fazem parte do conjunto M . A este conjunto, precisamos acrescentar um marcador para o fundo da imagem gradiente (por quê?). Observe a separação dos grãos que se tocam no resultado da segmentação (Fig. 57(e)).

28 Conclusão

Este artigo apresentou um conjunto de técnicas elementares de processamento de imagens, procurando ilustrar a diferença entre as transformações lineares que tratam a informação 2-D como uma entidade única, e as transformações não-lineares que, sob a ótica mais realista da Morfologia Matemática, a considera como um conjunto de objetos a ser transformado para posterior análise.

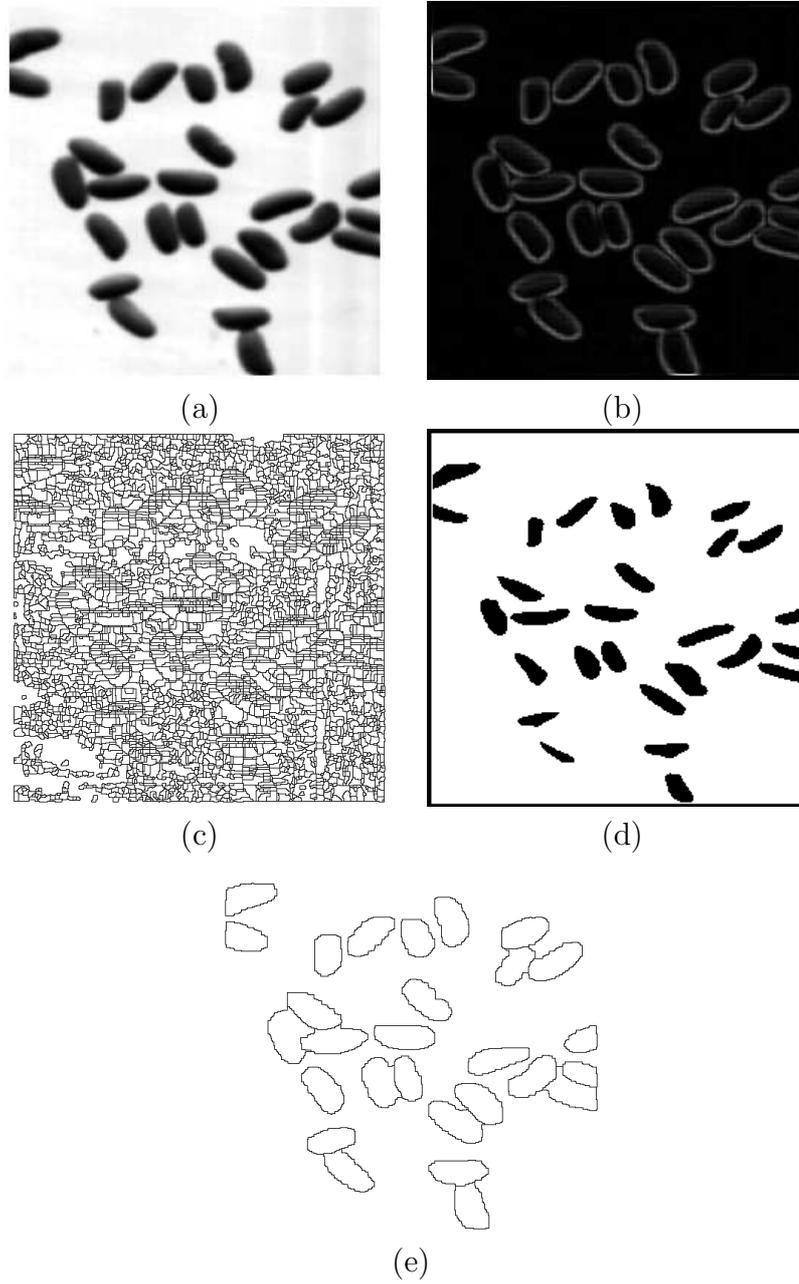


Figura 57: Imagem de grãos (a), imagem gradiente (b), e sua LDA (c). Marcadores obtidos por limiarização (d), e segmentação por LDA (e).

29 Visão Geral

Até o momento estudamos transformações de imagens sem nos preocupar com a análise de estruturas contidas nas imagens. A análise é a parte do processamento de imagens responsável pela interpretação semântica dessas estruturas com relação a uma dada aplicação. Conforme descrito em **Etapas do Processamento de Imagens**, a análise consiste das seguintes operações: segmentação, descrição, reconhecimento, e interpretação. A segmentação consiste da subdivisão de uma imagem em suas partes constituintes ou objetos, onde um objeto neste contexto refere-se a uma componente conexa. Após a segmentação, cada objeto é descrito com relação a suas propriedades geométricas e topológicas. Baseando-se nesta descrição, o reconhecimento de cada objeto resulta em uma imagem rotulada, onde pixels que pertencem a objetos diferentes têm associados rótulos distintos. A interpretação semântica de um conjunto de objetos rotulados no domínio de uma aplicação finaliza o processo de análise dando lugar a uma ação. Por exemplo, um sistema pode reconhecer dígitos em uma imagem de código de barras e interpretá-los como o número de acesso a um dado produto em um supermercado. A ação neste caso pode ser a contabilização do valor do produto na compra.

Nesta parte do curso vamos abordar as principais técnicas de segmentação de imagens. A segmentação é uma das principais e mais estudadas operações em processamento de imagens. A segmentação pode ser vista como um problema de classificar N elementos em K classes, tal que: $K \leq N$; $\cup_{i=1}^K K_i = \text{imagem}$; $K_i \cap K_j = \phi$, se $i \neq j$; e elementos em uma mesma classe K têm propriedades similares entre si e distintas das propriedades dos elementos de outras classes. Assim, a segmentação também pode ser modelada como um problema de otimização combinatória (i.e. programação inteira), no qual procura-se uma partição ótima de acordo com algum critério de similaridade entre os elementos de uma mesma classe. Neste contexto cada objeto de interesse na imagem é uma classe.

Métodos de segmentação podem ser classificados como **iterativos** [1, 2, 3, 4] ou **automáticos** [5, 6, 7, 8] levando em conta, respectivamente, a intervenção ou não intervenção do usuário na segmentação, ou podem ter sua classificação de acordo com a estratégia de representação dos objetos a serem segmentados: **métodos orientados a bordas** [2, 3, 4] (Seção 30) e **métodos orientados a regiões** [1, 9, 10] (Seção 31). Conforme descrito em **Introdução à Topologia Digital**, um objeto (ou componente conexa) pode ser representado por seu contorno envolvente (i.e. borda externa) - e no caso de "buracos" também pelos contornos internos (bordas internas) - ou pela região de pixels conexos que forma seu interior. Métodos orientados a bordas procuram detectar as

bordas de cada objeto de interesse na imagem. Enquanto métodos orientados a regiões procuram identificar todos os pixels que pertencem ao interior de cada objeto de interesse. Neste curso, nós adotaremos esta segunda forma de classificação para descrever as diversas estratégias de segmentação.

A Figura 58 mostra um diagrama das principais estratégias de segmentação. A abordagem clássica orientada a bordas detecta inicialmente segmentos de borda na imagem e depois une os segmentos que pertencem a uma mesma borda. Esta abordagem é denominada *edge-linking* (Seção 30.1). Uma outra abordagem, mais recente, parte de um modelo global para a borda que deve ser deformado durante a segmentação até aderir completamente à borda do objeto de interesse (Seção 30.2). Alternativamente, no caso dos métodos orientados a regiões, existem três abordagens mais relevantes: classificação de pixels (Seção 31.1), agregação de pixels (Seção 31.2) e *split-and-merge* (i.e. divisão e conquista - Seção 31.3). A classificação de pixels leva em conta características da imagem, tais como cor, gradiente, textura, para classificar cada pixel como pertencente a uma das K possíveis classes ou objetos da imagem. As outras duas abordagens levam em conta além de características de imagem, um critério de conectividade definido para os objetos. Na agregação de pixels, a idéia básica é o crescimento de regiões conexas a partir de pixels sementes e de um critério de parada no crescimento, tal que cada região final represente um objeto. As técnicas de divisão e conquista particionam a imagem em regiões conexas que podem ser fundidas ou subdivididas em novas regiões conexas, tal que ao final deste processo cada região represente um objeto.

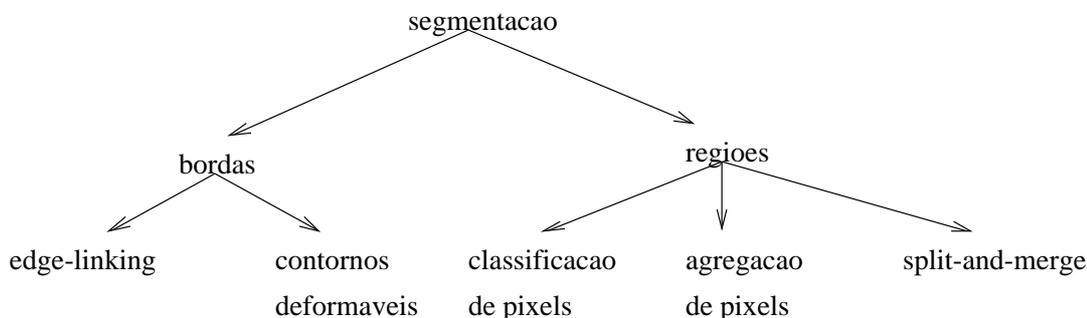


Figura 58: Estratégias de segmentação.

30 Segmentação Orientada a Bordas

30.1 Edge-Linking

Os segmentos de borda em uma imagem podem ser representados por linhas retas ou curvas com formas arbitrárias. A maioria dos métodos nesta abordagem utilizam o gradiente da imagem (ver **Filtragem Linear no Domínio Espacial**) para realçar os segmentos de borda. Depois comparam a magnitude dos pixels na imagem do gradiente com um

valor limiar T para gerar uma imagem binária, onde pixels com magnitude acima de T têm valor 1, indicando que pertencem a um segmento de borda, e o restante dos pixels têm valor 0 (Seção 30.1.1). Os segmentos de borda são então unidos baseados em critérios locais ou globais visando formar uma borda de objeto como um contorno fechado e conexo (Seção 30.1.2). Infelizmente esta estratégia é muito susceptível a erros, principalmente em imagens ruidosas onde o gradiente vai realçar ainda mais o ruído. Nestes casos, métodos de edge-linking baseados em grafos são considerados mais robustos (Seção 30.1.3).

30.1.1 Detecção de Segmentos de Borda

Uma analogia interessante para a detecção de segmentos de borda pode ser feita através de representação vetorial. Vimos em **Filtragem Linear no Domínio Espacial** que a filtragem linear no domínio espacial resulta da convolução de um filtro $h(i, j)$ com uma imagem $f(x, y)$. Podemos interpretar um conjunto de filtros $\{h_1, h_2, \dots, h_n\}$ como vetores do espaço R^n e a convolução da imagem f com cada filtro h_i como uma projeção vetorial de f na direção de h_i . Isto significa que se h_i for um filtro para detecção de segmentos horizontais, então as componentes de bordas horizontais da imagem devem ser mais realçadas do que as componentes em outras direções.

A Figura 59a, por exemplo, mostra a imagem de dois quadrados onde um segmento de borda (i.e. o lado de um quadrado) pode ter uma de quatro possíveis orientações: horizontal, vertical, 45° , e -45° . As Figuras 59b-e mostram, respectivamente, o resultado da convolução desta imagem com os quatro filtros seguintes:

$$\begin{array}{cccc}
 \begin{bmatrix} -1 & -1 & -1 \\ 2 & 2 & 2 \\ -1 & -1 & -1 \end{bmatrix} & \begin{bmatrix} -1 & 2 & -1 \\ -1 & 2 & -1 \\ -1 & 2 & -1 \end{bmatrix} & \begin{bmatrix} -1 & -1 & 2 \\ -1 & 2 & -1 \\ 2 & -1 & -1 \end{bmatrix} & \begin{bmatrix} 2 & -1 & -1 \\ -1 & 2 & -1 \\ -1 & -1 & 2 \end{bmatrix} \\
 \text{horizontal} & \text{vertical} & 45^\circ & -45^\circ
 \end{array}$$

Note que podemos facilmente classificar um pixel como pertencente a um segmento horizontal, por exemplo, se seu valor na imagem da Figura 59b for maior do que seu valor nas outras imagens 59c-e. Este processo é conhecido por detecção combinada. Note também que esta base de filtros é **ortogonal**, ou seja, o produto interno entre vetores h_i e h_j , onde $i \neq j$, é sempre 0.

Uma outra forma de entender as filtrações acima é visualizar as operações no domínio da frequência como filtros passa-altas. Neste caso, a convolução de $f(x, y)$ por $h(i, j)$ é equivalente ao produto das transformadas de Fourier de f e h . A Figura 60 mostra as respectivas transformadas (i.e. magnitude da transformada) de cada imagem da figura 59

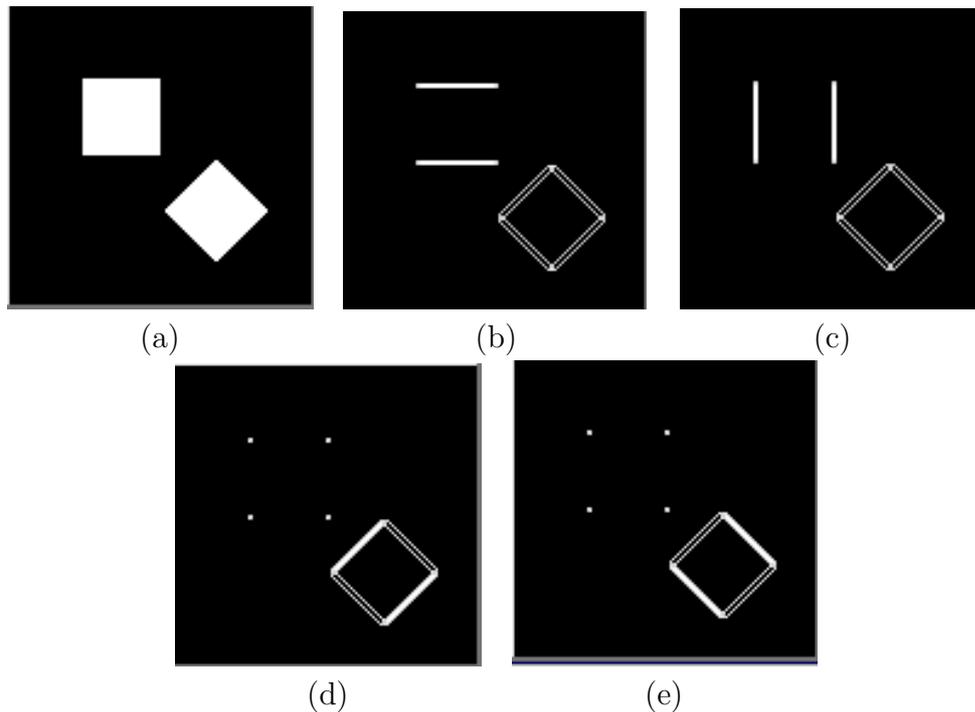


Figura 59: (a) Imagem original; (b) imagem filtrada pelo filtro horizontal; (c) imagem filtrada pelo filtro vertical; (d) imagem filtrada pelo filtro diagonal de 45° ; e (e) imagem filtrada pelo filtro diagonal de -45°

e a Figura 61 mostra a transformada de cada filtro h_i na ordem horizontal, vertical, 45° , e -45° , respectivamente. Observe que as imagens das Figuras 60b-e resultam do produto entre a imagem da Figura 60a e cada uma das imagens das Figuras 61a-d, respectivamente.

Em situações reais, as imagens são ruidosas e possuem segmentos de borda em diversas orientações. Portanto, os filtros de gradiente procuram realçar bordas em mais direções do que as citadas acima. Uma pré-filtragem de suavização (i.e. um filtro passa-baixas) também é utilizada para reduzir o ruído visando aumentar as chances de sucesso da detecção. A princípio, qualquer operador de gradiente linear ou não linear pode ser usado nesta etapa. Um exemplo comum é o operador de Sobel que já tem embutido um filtro de suavização (ver as respectivas máscaras de Sobel em **Filtragem Linear no Domínio Espacial**).

Exercício 1: Mostre que a convolução de uma imagem com as máscaras de Sobel pode ser implementada por uma convolução inicial com a máscara diferencial (i.e. passa-altas) $[-101]$ (ou sua contraparte vertical) seguida de uma convolução com a máscara de suavização $[121]$ (ou sua contraparte vertical).

Infelizmente, não existe nenhuma garantia que ao final deste processo todos os segmentos de interesse estarão presentes na imagem binária. A etapa de edge-linking propriamente dita é portanto fundamental para unir os segmentos de borda fechando os possíveis

gaps entre eles até formar o contorno final.

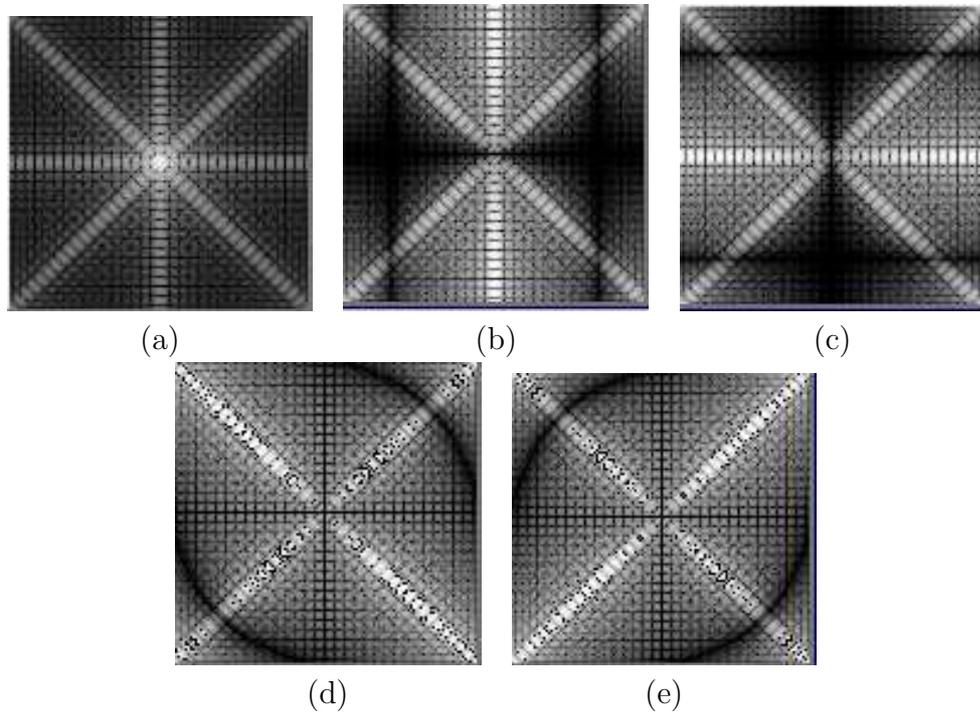


Figura 60: Magnitude das Transformadas de Fourier da: (a) imagem dos quadrados; (b) imagem filtrada pelo filtro horizontal; (c) imagem filtrada pelo filtro vertical; (d) imagem filtrada pelo filtro diagonal de 45° ; e (e) imagem filtrada pelo filtro diagonal de -45°

30.1.2 União de Segmentos Detectados

A união dos segmentos detectados pode ser feita por critérios locais ou globais. Os critérios locais são aqueles que levam em conta apenas uma vizinhança de pixels (e.g. 3×3 , 5×5) em torno dos extremos de cada segmento para avaliar se os segmentos satisfazem alguma propriedade de similaridade que permite a união. Suponha, por exemplo, dois pixels (x, y) e (x', y') vizinhos tal que (x, y) é um ponto extremo de um dado segmento e (x', y') é um ponto extremo de outro segmento. Um critério de similaridade para unir estes segmentos pode ser:

$$|mag[\vec{g}(x, y)] - mag[\vec{g}(x', y')]| \leq T \quad (128)$$

$$ang[\vec{g}(x, y), \vec{g}(x', y')] \leq \theta \quad (129)$$

onde $\vec{g}(x, y)$ é o vetor gradiente no ponto (x, y) , mag é a magnitude de um vetor, ang é o menor ângulo entre dois vetores, e T e θ são dois valores que definem um limiar superior de similaridade.

Os critérios globais são aqueles que analisam todos os pixels da imagem para decidir quais segmentos serão unidos. Uma das principais técnicas globais é a Transformada de

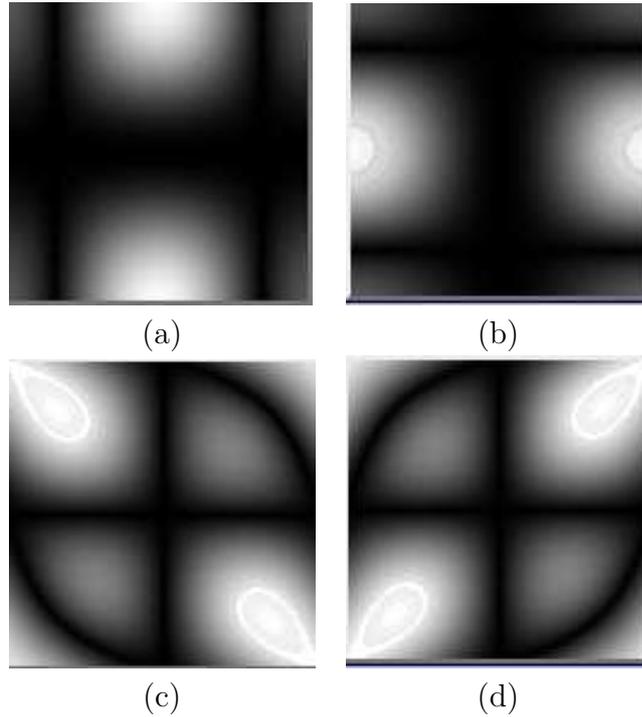


Figura 61: Magnitude das Transformadas de Fourier do: (a) filtro horizontal; (c) filtro vertical; (d) filtro diagonal de 45° ; e (e) filtro diagonal de -45°

Hough. Esta transformada pode ser usada, por exemplo, para selecionar os pixels da imagem binária de segmentos de borda que pertencem a segmentos de reta com uma mesma orientação. Dado um conjunto de pixels selecionados, a conexidade entre estes pixels e, conseqüentemente, a união dos segmentos pode levar em conta, por exemplo, a distância entre pixels vizinhos. Este método, descrito a seguir, representa uma estratégia robusta para fechar *gaps* nos segmentos de borda.

A Transformada de Hough consiste em levar qualquer função $g(v, p)$ de um espaço de coordenadas em R^n para um espaço de parâmetros em R^m , onde v é o vetor de coordenadas e p é o vetor de parâmetros. No caso mais simples, podemos representar todas as retas $y = ax + b$ do espaço xy que passam por um mesmo ponto (x_i, y_i) como uma única reta $b = -x_i a + y_i$ do espaço ab (ver Figura 62). Por outro lado, se duas retas $b = -x_i a + y_i$ e $b = -x_j a + y_j$ se interceptam no espaço ab no mesmo ponto (a_k, b_k) , então os pontos (x_i, y_i) e (x_j, y_j) pertencem a uma mesma reta $y = a_k x + b_k$ do espaço xy (ver Figura 63). As mesmas observações são válidas para qualquer curva. Por exemplo, poderíamos estar interessados em encontrar pontos que pertencem a círculos $(x - c_1)^2 + (y - c_2)^2 = c_3^2$. Neste caso, no entanto, o espaço de coordenadas é R^2 , mas o espaço de parâmetros é R^3 . Voltando ao nosso exemplo, o problema de trabalhar com a equação de retas na forma $y = ax + b$ é que os valores de a e b crescem infinitamente quando as retas se aproximam da direção vertical. Assim, a Transformada de Hough utiliza a representação normal para retas: $x \cos \theta + y \sin \theta = \rho$, onde $-90 \leq \theta \leq 90$ e $-\sqrt{2}D \leq \rho \leq \sqrt{2}D$, D sendo a

dimensão lateral da imagem (Figura 64a). Neste caso, as retas são senóides no espaço de parâmetros. A idéia principal, portanto, é subdividir o espaço de parâmetros em **células acumuladoras** $A(\theta_i, \rho_j)$ (Figura 64b). Cada célula representa uma reta do espaço xy com um determinado ângulo θ_i com o eixo x e distância ρ_j da origem. Inicialmente, todas as células são inicializadas com valor zero. Em seguida, para cada ponto (x_k, y_k) do espaço xy que pertence a um segmento de borda detectado, calcula-se para cada valor θ_i do espaço de parâmetros, o valor de ρ_j por arredondamento do valor de $x_k \cos \theta_i + y_k \sin \theta_i$ e incrementa de 1 o valor da célula $A(\theta_i, \rho_j)$. Ao final, o valor acumulado em cada célula $A(\theta_i, \rho_j)$ corresponde ao número de pontos do espaço xy que pertencem a mesma reta $x \cos \theta_i + y \sin \theta_i = \rho_j$. A precisão depende do número de células e o tempo computacional é linear e proporcional a este número.

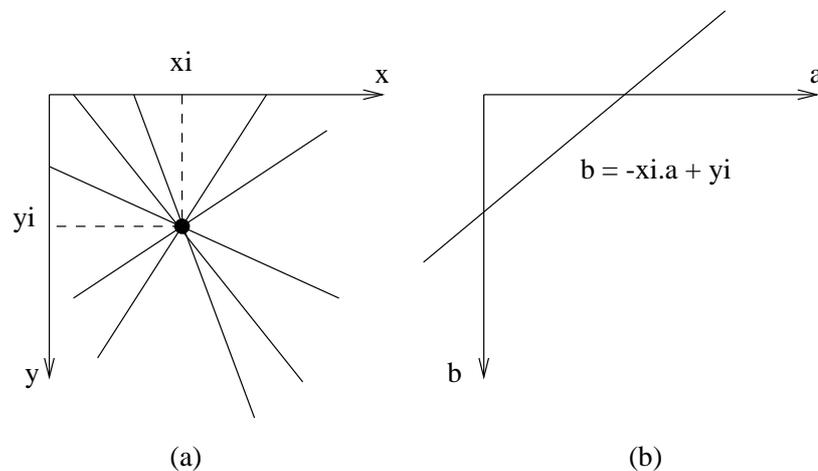


Figura 62: (a) Retas do espaço xy que se cruzam em um mesmo ponto (x_i, y_i) ; (b) Retas em (a) correspondem a uma única reta $b = -x_i a + y_i$ no espaço de parâmetros ab .

30.1.3 Edge-linking usando Grafos

A idéia de transformar o problema de detecção de segmentos de borda em um problema de encontrar o caminho de custo mínimo em um grafo ponderado e orientado foi inicialmente proposta por Martelli [11, 12]. Martelli propôs uma solução usando o algoritmo heurístico A^* [13] e mostrou que esta estratégia global de encontrar segmentos de borda é extremamente robusta em imagens ruidosas. Entretanto, na tentativa de reduzir o tempo de processamento do algoritmo A^* , variações do algoritmo de Martelli usando diferentes heurísticas foram subsequentemente propostas na literatura [14, 15]. Esses métodos comprometiam a otimalidade dos resultados e normalmente não garantiam nem que dados dois pontos sobre a borda, um segmento qualquer entre estes dois pontos seria encontrado. Outras abordagens globais para detecção ótima de segmentos de borda baseadas em programação dinâmica 1D também foram propostas [16, 17, 18]. Porém, impondo restrições ao formato da borda e limitando a solução do problema a alguns tipos de aplicação.

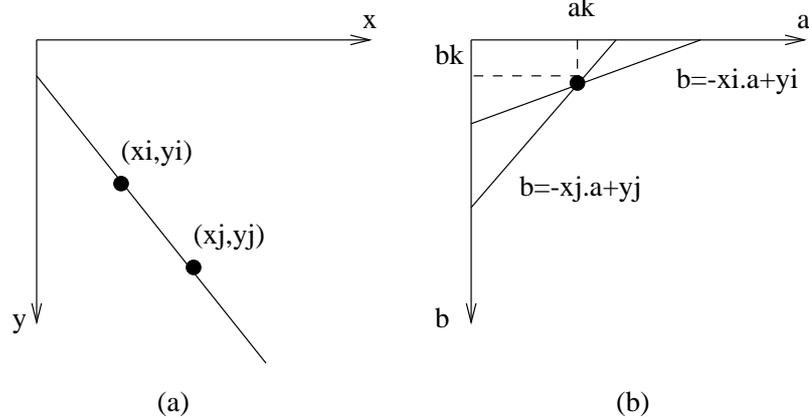


Figura 63: (a) Uma reta $y = a_k x + b_k$ do espaço xy que passa por dois pontos (x_i, y_i) e (x_j, y_j) ; (b) Duas retas $b = -x_i a + y_i$ e $b = -x_j a + y_j$ do espaço de parâmetros que se cruzam no ponto (a_k, b_k) correspondente aos parâmetros da reta em (a).

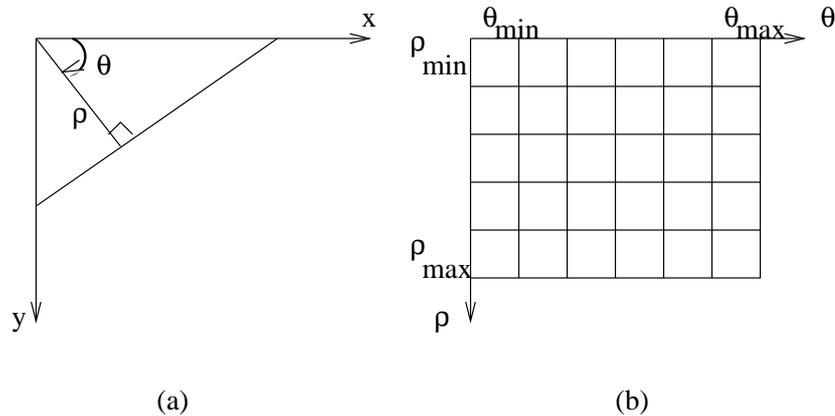


Figura 64: (a) Representação normal para retas: $x \cos \theta + y \sin \theta = \rho$, onde $\theta_{min} = -90$, $\theta_{max} = 90$, $\rho_{min} = -\sqrt{2}D$, $\rho_{max} = \sqrt{2}D$, e D é a dimensão lateral da imagem; (b) Células acumuladoras no espaço de parâmetros representando um número discreto de retas do espaço xy .

Considerando o aumento da capacidade computacional e o desenvolvimento de algoritmos rápidos para resolver o problema do caminho de custo mínimo, a idéia de Martelli resurge recentemente em um método iterativo de segmentação chamado *live-wire* [19, 4, 20]. A solução proposta nesse método pode ser vista como uma abordagem de programação dinâmica 2D que não impõe restrições à borda e não depende da aplicação, e como uma abordagem de grafos que sempre garante uma solução e que esta solução é ótima. O modelo de grafo utilizado em [4, 20] é apresentado na seção **A**. Na seção **B** mostramos o algoritmo utilizado para encontrar o caminho de custo mínimo e discutimos algumas propriedades deste algoritmo exploradas no método *live-wire* para garantir resposta em tempo real.

A - Imagem como um Grafo

Considere uma imagem vista como o grafo orientado $G = (V, E)$ apresentado na figura 65a, onde os vértices dos pixels são os vértices do grafo e cada aresta de pixel define dois arcos com orientações opostas no grafo. $G = (V, E)$ consiste de um conjunto finito V de $|V|$ vértices e um conjunto E de $|E|$ arcos. Cada vértice $v \in V$ pode ter até quatro vértices $u_k \in V$, $k = 1, 2, 3, 4$, distintos e vizinhos-4, formando o conjunto $A(v)$ dos vértices adjacentes ao vértice v em G (figura 65b). Um arco $e = (v, u)$ em E é um par ordenado de vértices adjacentes em G que inicia no vértice v , termina no vértice $u \in A(v)$ e tem associado um valor de comprimento $l(v, u)$, ou $l(e)$, no intervalo $[0, C_{max}]$, onde C_{max} é o maior custo associado a qualquer arco do grafo $G(V, E)$. Entre dois vértices adjacentes, $v, u \in V$, existem dois possíveis arcos, $(u, v), (v, u) \in E$, onde $l(u, v)$ pode ser diferente de $l(v, u)$. Existem várias formas de associar um valor $l(e)$ a um arco $e \in E$. Uma forma simples é fazer $l(e)$ inversamente proporcional à magnitude do gradiente da imagem calculado em e . Falcão et al [4] propõem um cálculo mais complexo baseado em treinamento supervisionado e em um conjunto de características de imagem, onde $l(u, v) \neq l(v, u)$ e a borda é um contorno orientado. Um caminho $P = (e_1, e_2, \dots, e_n)$ em G é formado por uma sequência de arcos conexos $e_i = (v_{i-1}, v_i)$, $i = 1, 2, \dots, n$, onde o vértice final de um arco é o vértice inicial do próximo arco (figura 65a, onde $n = 5$). P é dito um **caminho simples** em G quando não repete nenhum vértice, $v_i \neq v_j$ para todo $i \neq j$. O comprimento de um caminho P é definido como a soma do comprimento de seus arcos $d(P) = l(e_1) + l(e_2) + \dots + l(e_n)$. $P = \langle v, v' \rangle$ é dito o **caminho mais curto** de um vértice $v \in V$ para um vértice $v' \in V$ se, considerando o comprimento de todos os possíveis caminhos simples de v a v' , $d(P)$ é o comprimento mínimo.

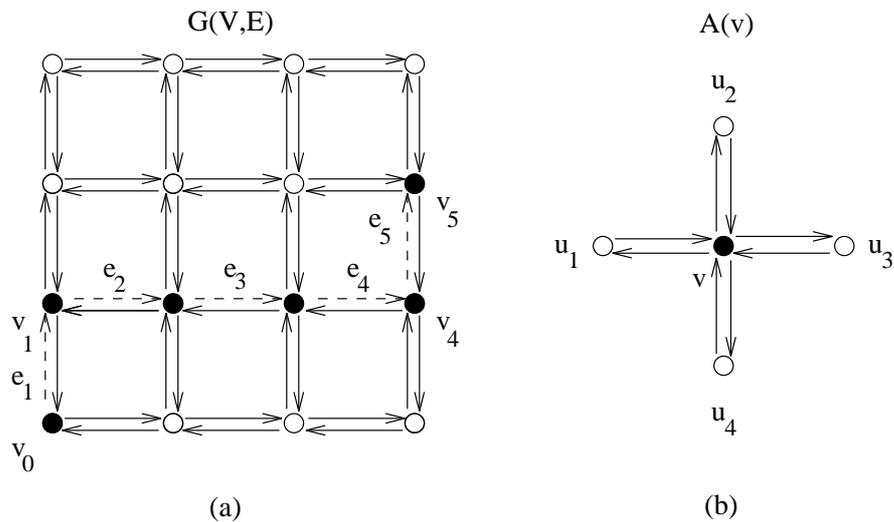


Figura 65: (a) Um grafo $G(V, E)$ definido para uma imagem. Um caminho $P = (e_1, e_2, \dots, e_5)$ em G entre os vértices v_0 e v_5 . (b) Cada vértice $v \in V$ define um conjunto $A(v) = u_1, u_2, u_3, u_4$ de no máximo 4 vértices vizinhos-4.

B - Algoritmo

Dados dois pontos v_s e v_e selecionados pelo usuário sobre a borda do objeto na imagem, o problema de encontrar o segmento de borda ótimo entre os dois pontos é transformado no problema de encontrar o caminho de custo mínimo entre o vértice v_s e o vértice v_e no grafo $G = (V, E)$. Uma das soluções ótimas mais eficientes para este problema é a implementação de Dial do algoritmo de Dijkstra [21]. Esta implementação utiliza uma fila de prioridade circular e tem complexidade computacional linear $O(|V| + |E|C_{max})$. Na verdade, o algoritmo de Dijkstra encontra todos os caminhos de custo mínimo a partir de v_s , mas como estamos interessados apenas no caminho mínimo de v_s a v_e , o algoritmo proposto em [20] é uma versão modificada da implementação de Dial. Este algoritmo funciona da seguinte forma:

Entrada: Uma função de custo $c(e)$, um vértice inicial v_s , e um vértice final v_e ;

Saída: Um caminho de custo mínimo de v_s a v_e ;

Estruturas Auxiliares: Um *array* $cc(v)$ de custos acumulados, representando o custo atual do caminho mínimo encontrado até o momento de v_s para cada vértice $v \in V$, um *array* $dir(v)$ indicando para cada vértice $v \in V$ qual é o vértice vizinho que está no caminho de custo mínimo partindo de v_s , uma fila circular Q de vértices com $C_{max} + 1$ buckets, e uma lista L de vértices já processados;

início

1. inicialize $cc(v)$ com ∞ and $dir(v)$ com *null* para todos os vértices $v \in V$;
 2. inicialize $cc(v_s)$ com 0 e coloque v_s em Q ;
 3. enquanto $v_e \notin L$ faça
 - a. remova um vértice v de Q tal que $cc(v) = \min_{v' \in Q} \{cc(v')\}$, e coloque v em L ;
 - b. para cada vértice v' tal que $v' \in A(v)$ e $v' \notin L$ faça
 - (i) calcule $cc_{tmp} = cc(v) + c(e')$ onde e' é o arco que vai de v' para v e $c(e')$ é o custo do arco e' ;
 - (ii) se $cc_{tmp} < cc(v')$ então
 - a. atualize $cc(v')$ com cc_{tmp} e $dir(v')$ para a direção de v' para v ;
 - b. se $v' \notin Q$ então insira v' em Q se não atualize a posição de v' em Q ;
- fim de para;*
- fim de enquanto;*
4. partindo de v_e , trace recursivamente o caminho ótimo seguindo a direção dos arcos em dir até encontrar v_s ;

Note que no algoritmo acima, a fila Q está sempre ordenada em ordem crescente de $cc(v)$. A Figura 66 ilustra um exemplo do algoritmo acima rodando para uma imagem inteira representada pelo grafo da Figura 66a. As Figuras de 66b a 66f mostram os *arrays* de custos acumulados cc e direções dir combinados em uma só representação. Nestas Figuras, o próximo vértice a sair da fila é sempre representado pelo círculo parcialmente escuro. A Figura 66b mostra a situação inicial antes da primeira iteração. As Figuras 66c e 66d mostram respectivamente a situação após a primeira e a segunda iteração. A Figura 66e mostra a situação após sete iterações e a Figura 66f mostra o resultado final após quinze iterações. Note que se o vértice v_e for previamente determinado como sendo, por exemplo, o vértice do canto superior esquerdo do grafo da Figura 66a, o caminho ótimo de v_s a v_e é encontrado logo após a sétima iteração (Figura 66e) e o algoritmo termina o processamento antes de visitar todos os vértices do grafo.

O algoritmo acima é usado de forma interativa no método *live-wire* [20] para segmentar bordas 2D. Neste método, o usuário especifica um ponto inicial sobre a borda usando o cursor. Para qualquer posição subsequente do cursor na imagem, dado que o usuário pode posicionar o cursor em qualquer lugar da imagem, uma curva - caminho ótimo (custo mínimo) - conectando o ponto inicial ao ponto que corresponde à posição atual do cursor é calculada e apresentada na tela do computador em tempo real, portanto o nome “*live-wire*”. Quando o cursor é posicionado próximo à borda, o segmento de *live-wire* adere à borda. Se este segmento descrever adequadamente a borda desejada, o usuário deposita o cursor cuja localização passa a ser a do novo ponto de partida. A borda completa (i.e. contorno fechado, conectado e orientado) é especificada via um conjunto de segmentos de *live-wire* desta maneira.

Para obter resposta em tempo real, o método explora as seguintes propriedades do algoritmo acima:

- Quando o algoritmo encontra um caminho de custo mínimo K de v_s até v_e , ele encontrou na verdade todos os caminhos que partem de v_s de custo mínimo menor do que K . Assim, quando o usuário movimenta o cursor para uma posição já calculada, o método apenas mostra o segmento de *live-wire* na tela.
- Se o algoritmo calculou todos os caminhos de custo mínimo menor do que K , este resultado é usado para calcular os caminhos de custo mínimo maior ou igual a K . Assim, o cálculo de segmentos de *live-wire*, quando o cursor está em movimento, é incremental com relação ao cálculo obtido na posição anterior.

O algoritmo acima pode ser visto como um caso particular do algoritmo A^* [13], no qual a heurística é nula. No algoritmo A^* , a fila Q é ordenada na ordem crescente de uma função $f(v) = cc(v) + h(v)$, onde $cc(v)$ é o nosso custo acumulado e $h(v)$ é uma estimativa do custo do caminho mínimo de v até v_e . Se esta estimativa for sempre menor do que o valor verdadeiro e f uma função monotônica, o algoritmo A^* garante o caminho ótimo. No nosso grafo isto seria possível usando a distância de *city-block*, por exemplo, como heurística e fazendo com que os valores de $c(v)$ estivessem no intervalo de $[1, C_{max}]$.

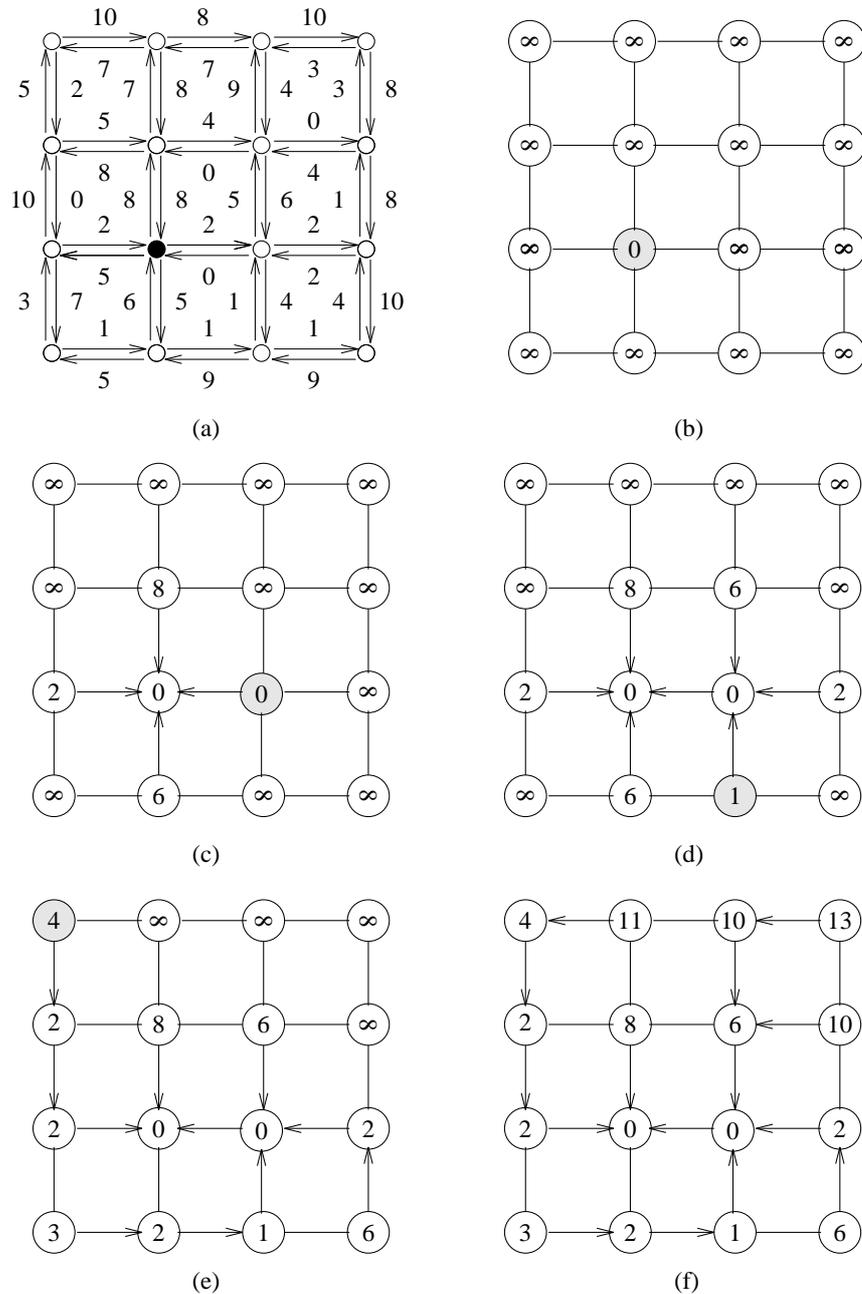


Figura 66: (a) Um grafo representando uma imagem, onde valores de custo de $[0,10]$ são associados às arestas do grafo e o vértice inicial é representado pelo círculo escuro. As Figuras de (b) a (f) ilustram respectivamente a situação do grafo de saída antes da primeira iteração, após a primeira iteração, após a segunda iteração, após a sétima iteração e após quinze iterações. Nestas Figuras, o próximo vértice a sair da fila é sempre representado pelo círculo parcialmente escuro.

A heurística faria com que o algoritmo ficasse ainda mais rápido. Infelizmente ela não ajuda muito no caso interativo, pois a cada nova posição do cursor na imagem (vértice v_e), temos que atualizar a função $h(v)$ e conseqüentemente a posição de todos os vértices v atualizados na fila.

Para concluir, veja o exemplo da Figura 67. A Figura 67a mostra uma fatia original de um joelho obtida por Tomografia de Raios-X. A Figura 67b mostra a imagem de custo c priorizando bordas de alto contraste, representadas por ossos e pele. Se o objetivo for segmentar a borda de menor contraste que está no meio da parte inferior da imagem, então para quaisquer dois pontos v_s e v_e selecionados sobre esta borda, espera-se que o caminho ótimo de v_s a v_e descreva a borda desejada. No entanto, como mostra a Figura 67c, este caminho toma uma rota alternativa sendo fortemente atraído pelas bordas de alto contraste. Após o usuário traçar típicos segmentos da borda de baixo contraste, como indicado na Figura 67d, o custo conjunto resultante do treinamento é mostrado na Figura 67e. Note na Figura 67e como todas as bordas de baixo contraste (indicadas pelas setas) obtêm custos significativamente mais baixos comparados com os custos da Figura 67b. O caminho ótimo de v_s a v_e é agora fortemente atraído pela borda de baixo contraste desejada (Figura 67f) e fortemente repelido pelas bordas de alto contraste como ilustrado na Figura 67g.

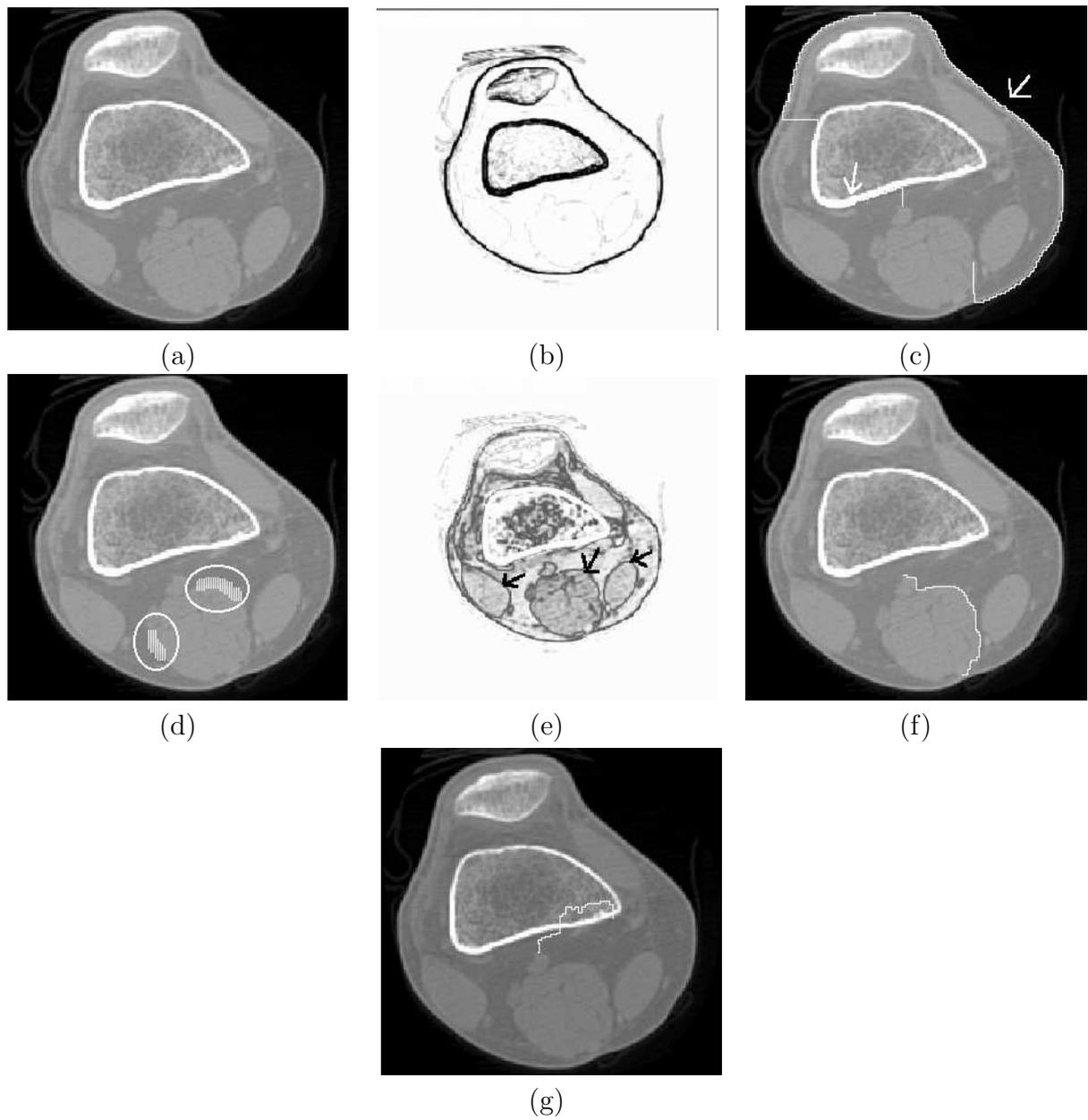


Figura 67: (a) Uma fatia do joelho obtida por CT; (b) Uma imagem de custos por aresta priorizando bordas de alto contraste; (c) Caminho ótimo sendo atraído pelas bordas de alto contraste; (d) Treinamento para as bordas de baixo contraste. (e) Nova função de custo após treinamento. (f) Caminho ótimo sendo atraído pelas bordas de baixo contraste. (g) Caminho ótimo sendo repelido pelas bordas de alto contraste.

30.2 Contornos Deformáveis

Contornos deformáveis têm se tornado bastante populares nos últimos 10 anos [22, 23, 24, 25, 26, 27, 28, 29, 30]. Entre estes modelos, o mais popular é conhecido por Snakes [31]. Snakes são curvas planares abertas ou fechadas (e.g. splines de continuidade controlada) que têm associado um funcional de energia e que se deformam sob a ação de forças internas e externas com o objetivo de minimizar este funcional. A idéia básica é que a partir de uma localização inicial na imagem, a snake se deforme até atingir um valor mínimo de energia que deve coincidir com a situação em que a snake adere à borda do objeto desejado na imagem. Um contorno ativo é um caso particular de snakes, onde a curva é fechada. Este modelo é usado para representar uma borda 2D de um objeto na imagem. Também existem extensões deste modelo para segmentar objetos 3D [32, 33, 24] a partir de um conjunto de fatias tomográficas.

Nesta seção, apresentamos a metodologia de segmentação baseada em contornos ativos.

30.2.1 Contorno Ativo: Modelo Contínuo

Um contorno ativo pode ser visto no plano como uma curva paramétrica fechada $\mathbf{v}(\mathbf{s}) = (\mathbf{x}(\mathbf{s}), \mathbf{y}(\mathbf{s}))$, onde $(\mathbf{x}, \mathbf{y}) \in \mathbb{R}^2$, $\mathbf{s} \in [0, 1]$ é o domínio paramétrico e $\mathbf{v}(0) = \mathbf{v}(1)$. O funcional de energia associado a este contorno pode ser descrito como:

$$\mathbf{E}_{\text{snake}} = \int \mathbf{E}_{\text{int}}(\mathbf{v}(\mathbf{s})) + \mathbf{E}_{\text{ext}}(\mathbf{v}(\mathbf{s})) d\mathbf{s} \quad (130)$$

onde,

$$\mathbf{E}_{\text{int}} = \frac{\alpha(\mathbf{s})|\mathbf{v}_s(\mathbf{s})|^2 + \beta(\mathbf{s})|\mathbf{v}_{ss}(\mathbf{s})|^2}{2} \quad (131)$$

e

$$\mathbf{E}_{\text{ext}}(\mathbf{v}(\mathbf{s})) = \mathbf{E}_{\text{image}}(\mathbf{v}(\mathbf{s})) + \mathbf{E}_{\text{user}}(\mathbf{v}(\mathbf{s})) \quad (132)$$

O termo \mathbf{E}_{int} representa a energia interna do contorno cuja diferencial gera a força interna que impõe regularidade ao contorno. Dois parâmetros desta função que controlam a simulação das características físicas do contorno, são: $\alpha(\mathbf{s})$, que controla a 'tensão' entre os pontos do contorno, e $\beta(\mathbf{s})$, responsável pela 'elasticidade' do contorno. Os termos \mathbf{v}_s e \mathbf{v}_{ss} significam as derivadas de primeira e segunda ordem, respectivamente, aplicadas sobre os pontos $\mathbf{v}(\mathbf{s})$. O termo \mathbf{E}_{ext} representa a energia externa cuja diferencial gera a força externa que atrai o contorno para a borda do objeto. Parte deste termo está diretamente ligada às características de borda na imagem, expressa através da componente $\mathbf{E}_{\text{image}}$

$$\mathbf{E}_{\text{image}}(\mathbf{v}(\mathbf{s})) = -c|\nabla[\mathbf{G}_\sigma * \mathbf{I}(\mathbf{x}, \mathbf{y})]| \quad (133)$$

onde \mathbf{c} controla a magnitude da energia, ∇ é o operador gradiente, e $\mathbf{G}_\sigma * \mathbf{I}$ denota a imagem trabalhada por um filtro de suavização (Gaussiano), cuja característica de largura σ controla a extensão do mínimo local de $\mathbf{E}_{\text{image}}\mathbf{v}(\mathbf{s})$. Note que $\mathbf{E}_{\text{image}}(\mathbf{v}(\mathbf{s}))$ deve ser baixa onde o gradiente é alto. A outra parte \mathbf{E}_{user} está relacionada com outras restrições externas impostas ao contorno pelo usuário. Por exemplo, Kass et al. [31] menciona uma energia elástica $-\mathbf{k}(\mathbf{x}_1 - \mathbf{x}_2)^2$ criada pelo usuário ao modelar uma mola com uma extremidade livre, representada por um ponto (x_1) selecionado sobre o contorno, e outra extremidade fixa, representada por um ponto (x_2) selecionado sobre a borda desejada. O diferencial desta energia gera uma força elástica que puxa o contorno para a borda desejada.

30.2.2 Contorno Ativo: Modelo Discreto

A discretização das equações acima é necessária para viabilizar a computação. Partindo da equação 130, o funcional discreto pode ser aproximado por:

$$\mathbf{E}_{\text{snake}} = \sum_{i=1}^n \mathbf{E}_{\text{int}}(\mathbf{i}) + \mathbf{E}_{\text{ext}}(\mathbf{i}) \quad (134)$$

onde o contorno é representado por um conjunto V de n pontos $v_i = (x_i, y_i)$, $v_0 = v_n$, $E_{\text{int}}(i)$ é a energia interna do contorno no ponto v_i e $E_{\text{ext}}(i)$ é a energia externa avaliada no mesmo ponto. De acordo com Euler, o valor mínimo do funcional da equação 134 resulta na seguinte equação:

$$\alpha_i(\mathbf{v}_i - \mathbf{v}_{i-1}) - \alpha_{i+1}(\mathbf{v}_{i+1} - \mathbf{v}_i) + \beta_{i-1}(\mathbf{v}_{i-2} - 2\mathbf{v}_{i-1} + \mathbf{v}_i) - 2\beta_i(\mathbf{v}_{i-1} - 2\mathbf{v}_i + \mathbf{v}_{i+1}) + \beta_{i+1}(\mathbf{v}_i - 2\mathbf{v}_{i+1} + \mathbf{v}_{i+2}) + (\mathbf{f}_x(\mathbf{i}), \mathbf{f}_y(\mathbf{i})) = 0 \quad (135)$$

onde $f_x(i) = \partial E_{\text{ext}}/\partial x_i$ e $f_y(i) = \partial E_{\text{ext}}/\partial y_i$, e o restante decorre das derivadas parciais de E_{int} .

Na Figura 68, observa-se como funciona a deformação deste modelo. Partindo de um contorno inicial que pode ser, por exemplo, selecionado pelo usuário. A energia em uma vizinhança de pixels (pixels p_{jk} na Figura 68) em torno de cada ponto v_i do contorno é avaliada em várias iterações para selecionar uma nova posição v'_i em que esta energia seja mínima. Assim o contorno é deformado em direção à borda desejada.

30.2.3 Problemas do Método

Uma das principais dificuldades dos modelos de contornos deformáveis é encontrar a melhor localização para o contorno inicial. Se este não estiver próximo ao objeto de interesse, ele não é atraído para a borda do objeto podendo se reduzir a um ponto na imagem. Em

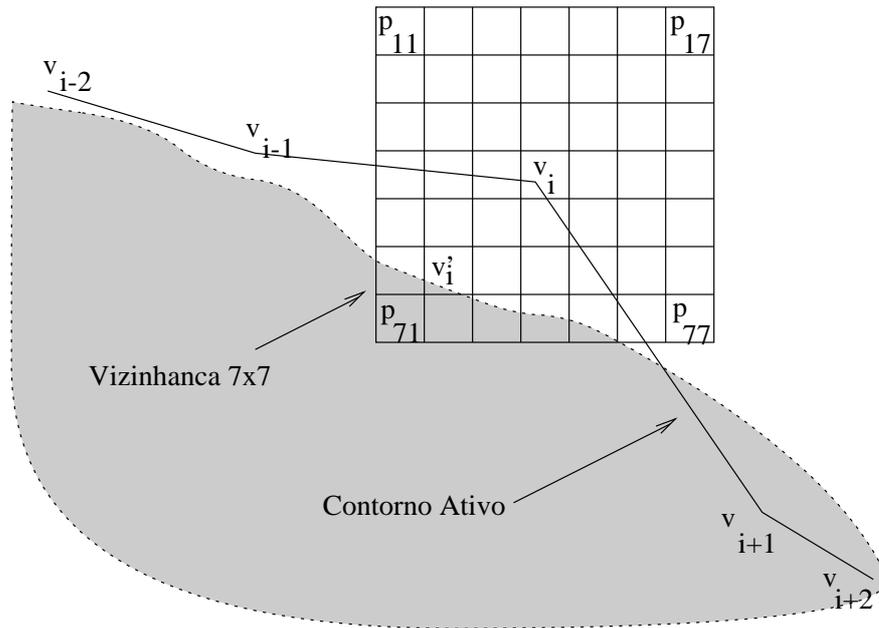


Figura 68: Exemplo de deformação do contorno ativo para um ponto v_i . Considerando uma vizinhança 7×7 , a próxima localização de v_i deve ser a do ponto v'_i onde o gradiente é mais alto.

[25], por exemplo, é proposto um procedimento automático baseado em Transformada de Hough para gerar o contorno inicial. Na prática, porém, a forma mais eficiente é a seleção manual feita pelo usuário. Outras dificuldades incluem: bordas fracas podem levar a comportamentos instáveis; o processo de minimizar a energia pode levar a oscilações desnecessárias; a escolha dos parâmetros do modelo tem um forte efeito sobre seu comportamento; o contorno fica preso em mínimos locais; o contorno tem problemas para aderir à borda do objeto, etc.

Estas dificuldades têm sido a principal motivação de muitos trabalhos sobre contornos deformáveis [34, 29]. A estabilidade da snake tem sido tratada pelo ajuste de parâmetros internos [35]. Leymarie e Levine [36] introduziram limites na força da imagem, novas regras para estabelecer parâmetros de rigidez e uma nova condição de término para a curva. Eles usaram snakes para representar a forma de seus objetos planos usando a transformada de distância para minimizar a função de energia [26], e para tratar objetos não rígidos, e.g. células [36]. Amini et al. [37] usou programação dinâmica para melhorar a estabilidade. Neunshwande et al. [30] apresentou um método no qual o usuário somente especifica os pontos finais da curva inicial. Zucker integrou evidência de borda local obtida através de um procedimento de relaxamento de rótulos, usando snakes que poderiam gerar estruturas construídas a partir destas bordas rotuladas [38]. Berger and Mohr [35] também usaram várias pequenas snakes crescentes para detectar comportamentos locais. Cooper et al. [39] apresentou um modelo probabilístico chamado "ripple filter", o qual age como um contorno deformável. Apesar destas melhorias, um número de dificuldades

consideráveis ainda permanecem, em particular, snakes sempre recaem na questão de uma propriedade de inicialização próxima da borda, além de requerer múltiplas inicializações, uma por objeto de interesse, etc.

30.2.4 Força Balão

Uma das propostas mais interessantes para melhorar o comportamento do contorno ativo é a utilização de uma força a mais, chamada força balão $E_{bal}(v(s))$ [34, 22, 28, 32, 33, 24]. A força balão é uma componente da E_{ext} e objetiva mover os pontos na direção normal da curva em cada ponto, evitando assim que o contorno se reduza a um único ponto ou fique preso em mínimos locais de energia (ver Figura 69).

A força Balão pode ser usada em um contorno deformável fechado para forçar o contorno a expandir ou reduzir, na ausência de influências externas. Um contorno inicializado dentro de uma imagem de objeto uniforme irá expandir sob a influência da força balão até que ele se aproxime da borda do objeto (i.e. até quando a função de energia externa afeta seu movimento). Uma das abordagens para esta força é adaptativa [23], sendo mais forte nas regiões homogêneas e fraca perto das bordas do objeto.

A idéia no caso discreto pode ser implementada calculando-se a energia do balão para cada ponto da vizinhança de v_i , segundo a equação abaixo:

$$\mathbf{c}_{jk}(\mathbf{v}_i) = \mathbf{n}_i \cdot (\mathbf{v}_i - \mathbf{p}_{jk}(\mathbf{v}_i)) \quad (136)$$

onde n_i é o vetor normal ao contorno no ponto v_i e $p_{jk}(v_i)$ é o pixel da vizinhança de v_i correspondente à entrada $c_{jk}(v_i)$ em matriz de energia do balão definida pela vizinhança de v_i . Portanto, a energia do balão é menor nos pontos mais longes de v_i na direção de n_i .

O vetor n_i pode ser encontrado pela rotação de 90° do vetor tangente t_i à curva no ponto v_i . O vetor t_i é facilmente computado por:

$$\mathbf{t}_i = \frac{(\mathbf{v}_i - \mathbf{v}_i - \mathbf{1})}{\|(\mathbf{v}_i - \mathbf{v}_i - \mathbf{1})\|} + \frac{(\mathbf{v}_i + \mathbf{1} - \mathbf{v}_i)}{\|(\mathbf{v}_i + \mathbf{1} - \mathbf{v}_i)\|} \quad (137)$$

31 Segmentação Orientada a Regiões

31.1 Classificação de Pixels

Na Seção 29 dissemos que a segmentação pode ser vista como um problema de classificação, onde dados N pixels de uma imagem e K classes: (a) cada pixel deve ser associado a uma e somente uma classe, (b) pixels com maior similaridade são associados à mesma classe,

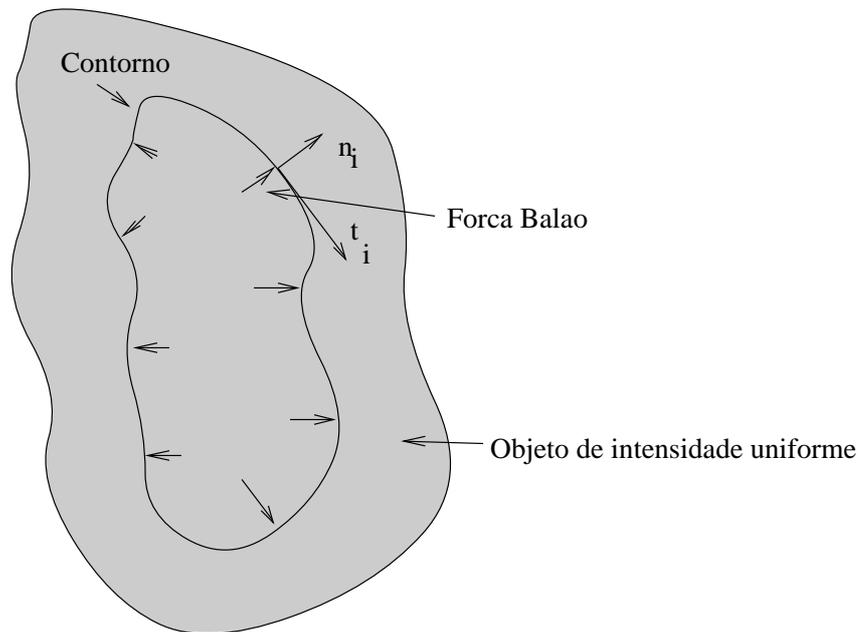


Figura 69: Um exemplo da força balão quando o contorno está sobre um objeto de intensidade uniforme. Neste caso, a força balão expande o contorno em direção à borda do objeto.

e (c) a união de todos os pixels de todas as classes é a imagem. Neste caso, porém, uma classe pode representar uma ou mais componentes conexas (i.e. objetos) com propriedades similares. Assim, após a segmentação por classificação de pixels, uma nova classificação baseada nas características estruturais das regiões segmentadas é normalmente necessária para reconhecer objetos (i.e. etapa de descrição e reconhecimento de um sistema de processamento de imagens).

Para classificar pixels, a técnica mais simples, mais antiga e ainda bastante utilizada é conhecida por *thresholding* (i.e. limiarização). Esta técnica consiste basicamente em subdividir o histograma da imagem em K intervalos disjuntos I_k , $k = 1, 2, \dots, K$, (i.e. K classes) usando $K - 1$ valores de limiar $T_1 < T_2 \dots < T_{K-1}$ e associar cada pixel da imagem com intensidade $f(x, y) \in I_k$ à classe k (ver Figura 70). No caso de 2 classes, por exemplo, c_1 representando o tecido ósseo em uma imagem 8-bits de Tomografia de Raios-X do joelho de um paciente e c_2 representando o background (i.e. o resto da imagem), podemos classificar os pixels com intensidade acima de $T_1 = 110$ como pertencentes à c_1 e abaixo deste valor como pertencentes à c_2 . O resultado desta classificação aplicada à imagem da Figura 71a é mostrado na Figura 71b, com valor 255 para os pixels em c_1 e 0 para os pixels em c_2 .

A técnica de limiarização recém introduzida é também denominada **limiarização global**. Quando a técnica de limiarização global é aplicada para uma propriedade da intensidade em uma vizinhança dos pixels, tal como a média de $f(x, y)$ em uma vizinhança

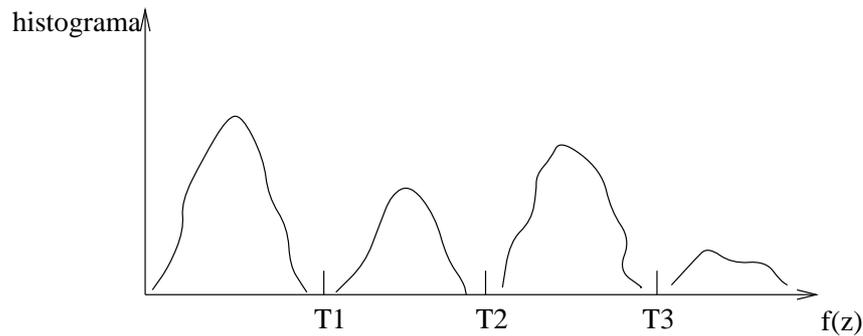


Figura 70: Limiarização no histograma de intensidade.

5×5 do pixel (x, y) , dizemos que a limiarização é local. E quando os valores de limiar variam em função da posição (x, y) dos pixels, dizemos que a limiarização é dinâmica.

Nesta seção estamos interessados em definir valores ótimos de limiar para a limiarização global (Seção 31.1.1) e na extensão desta técnica para múltiplas características de imagem (Seção 31.1.2). Também podemos ver este problema como um problema de *clustering* (i.e. agrupamento de pixels). Neste caso, descrevemos os métodos de agrupamento mais utilizados na Seção 31.1.3. A classificação baseada em múltiplas características também pode ser realizada via redes neurais [40]. Apesar de muito utilizada em reconhecimento de padrões, esta técnica não será abordada neste curso.

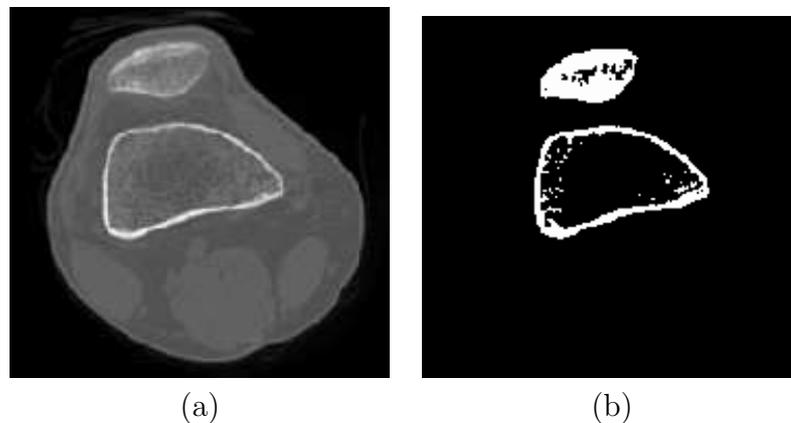


Figura 71: (a) Uma fatia do joelho obtida por CT; (b) Resultado da segmentação por limiarização global usando limiar 110.

31.1.1 Limiarização Ótima

Na prática, o histograma de uma imagem nem sempre apresenta uma separação bem definida entre as classes, como na Figura 70, dificultando a escolha do intervalo de cada classe. Neste caso, portanto, erros na classificação são inevitáveis e o limiar ótimo entre duas classes vizinhas é aquele que minimiza os erros.

Supondo, por exemplo, a imagem de um objeto com tons claros de cinza (classe c_2) sobre um background escuro (classe c_1). Seu histograma é como ilustrado na Figura 72, onde o pico mais alto representa os pixels do background e o pico mais baixo os do objeto. O histograma de uma imagem pode ser visto como uma densidade de probabilidade conjunta $\rho(z)$ dos pixels z de todas as classes presentes na imagem. Se separássemos os pixels de uma dada classe c_i , o histograma da imagem resultante seria equivalente à densidade de probabilidade $\rho(z|c_i)$ dos pixels z dada a classe c_i . Assim, a probabilidade de erro na classificação usando um limiar T entre as classes c_1 e c_2 é dada por:

$$P_{erro} = P(z \in c_1) * P(z \in c_2 | z \in c_1) + P(z \in c_2) * P(z \in c_1 | z \in c_2) \quad (138)$$

onde $P(z \in c_i)$ é a probabilidade de um pixel z pertencer a classe c_i e $P(z \in c_j | z \in c_i)$ é a probabilidade de classificarmos um pixel z que pertence à c_i como pertencente à classe c_j . P_{erro} é igual à área hachurada na Figura 72 e as probabilidades $P(z \in c_j | z \in c_i)$ são dadas por:

$$P(z \in c_1 | z \in c_2) = \int_{-\infty}^T \rho(z|c_2) \quad (139)$$

$$P(z \in c_2 | z \in c_1) = \int_T^{\infty} \rho(z|c_1) \quad (140)$$

Para achar o limiar ótimo T , para o qual P_{erro} é mínima, temos que conhecer todas as funções acima. Infelizmente, na prática, nós não dispomos de $\rho(z|c_i)$ e, portanto, assumimos alguma forma para $\rho(z|c_i)$. Supondo, por exemplo, que $\rho(z|c_i)$ é uma Gaussiana com média μ_i e variância σ_i^2 , o limiar ótimo T é dado por:

$$T = \frac{\mu_1 + \mu_2}{2} + \frac{\sigma^2}{\mu_1 - \mu_2} \log\left(\frac{P(z \in c_2)}{P(z \in c_1)}\right) \quad (141)$$

assumindo que $\sigma_1^2 = \sigma_2^2$. Esta prova é deixada como exercício para o leitor.

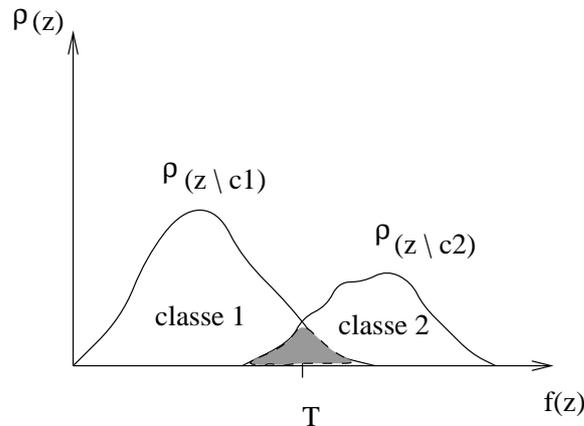


Figura 72: Limiarização Ótima para o caso de 2 classes.

31.1.2 Limiarização com Múltiplas Características

A generalização do problema do limiar ótimo apresentado na seção anterior para o caso de múltiplas classes e múltiplas características é discutida nesta seção. Para um conjunto de d características de imagem por pixel e K classes, este problema é resolvido no espaço R^d onde pixels de uma mesma classe se aglomeram em clusters e os limiares são transformados em funções discriminantes [41]. A Figura 73 mostra, por exemplo, um caso $d = 2$ com 2 classes onde as características são brilho e textura.

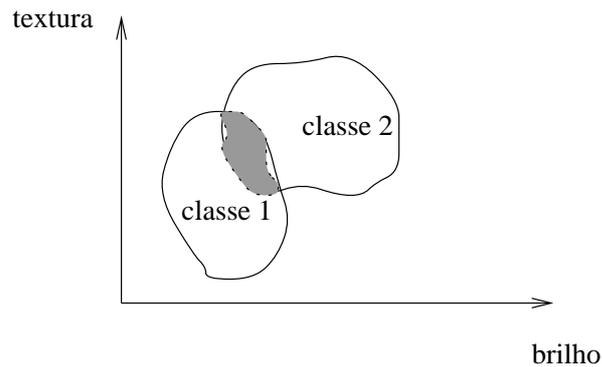


Figura 73: Histograma para 2 características.

Sendo \underline{z} um vetor de características associado a um pixel z , $\underline{\mu}$ o vetor média e Σ a matriz de covariância das características, e $\rho(\underline{z})$ e $\rho(\underline{z}|c_i)$ densidades de probabilidade Gaussianas de dimensão d , a função discriminante $g(z \in c_i)$ que associa um pixel z a uma classe c_i é dada por:

$$g(z \in c_i) = \frac{-1}{2}(\underline{z} - \underline{\mu}_i)^t \Sigma_i^{-1} (\underline{z} - \underline{\mu}_i) - \frac{d}{2} \log(2\pi) - \frac{1}{2} \log(|\Sigma_i|) \quad (142)$$

onde μ_i é o vetor média e Σ_i é a matriz de covariância das características para pixels da classe c_i . O classificador, portanto, associa o pixel z à classe c_i que apresenta o maior valor de $g(z \in c_i)$.

31.1.3 Métodos de Agrupamento

Dado um conjunto de d características por pixel, os métodos de agrupamento (ou *clustering*) se baseiam em um critério de distância (ou dissimilaridade) entre cada par de pixels no espaço de características R^d para agrupar os pixels mais próximos (i.e. aqueles com características mais similares) em uma mesma classe. A qualidade global do agrupamento é dada por uma função critério. Por exemplo, a soma do erro quadrático médio do agrupamento. Estes métodos dividem-se em **particionais** e **hierárquicos**. A diferença básica entre eles é que os métodos particionais geram uma única partição dos dados com um número fixo K de classes. Enquanto os métodos hierárquicos geram inicialmente um

conjunto de partições, percorrendo um intervalo de possíveis valores para o número de classes K . A partição final é escolhida posteriormente como um elemento deste conjunto. Obviamente, os métodos hierárquicos são mais complexos do que os métodos particionais e, portanto, estes são mais comuns na literatura.

31.1.4 Métodos Particionais

Em métodos particionais, os dois algoritmos mais utilizados são os algoritmos *k-means* e *fuzzy k-means*.

Algoritmo k-means:

No algoritmo *k-means*, dado um conjunto de n amostras $X = \{x_1, x_2, \dots, x_n\}$, onde x_i é um vetor de características de imagens calculadas para o pixel i , a idéia é encontrar a partição dos dados que minimize a seguinte função critério:

$$J(V) = \sum_{i=1}^{i=n} \quad (143)$$

Algoritmo fuzzy k-means:

31.1.5 Métodos Hierárquicos

Métodos aglomerativos:

Métodos divisivos:

Dendrograma:

Single-link e complete-link:

Single-link e minimum spanning tree:

31.2 Agregação de Pixels

31.3 Divisão e Conquista

Referências

- [Bar94] G. J. F. Banon & J. Barrera. *Bases da Morfologia Matemática para a Análise de Imagens Binárias*. IX Escola de Computação, Recife, PE, Julho 1994.
- [Che89] M. Coster & J. L. Chermant. *Précis d'Analyse d'Images*. Presses du CNRS, 1989.
- [ea83] A. C. Borik et alli. A Generalization of Median Filtering using Combination of Order Statistics. In *IEEE Proceedings*, 71(31), pages 1342–1350, 1983.
- [Hey82] G. Heygster. Rank filters in digital image processing. *Computer Graphics and Image Processing*, 17:148–164, 1982.
- [JC93] J. Serra & R. W. Schafer J. Crespo. Image Segmentation using Connected Filter. In *Int. Workshop on Mathematical Morphology and Its Application to Signal Processing*, pages 52–57, Barcelona, Spain, May 1993.
- [Kak82] A. Rosenfeld & A. Kak. *Digital Picture Processing*, volume 1. Academic Press, 1982.
- [Kor88] T. W. Korner. *Fourier Analysis*. Cambridge Univ., 1988.
- [Mai84] C. Lantuejoul & F. Maisonneuve. Geodesic Methods in Quantitative Image Analysis. *Pattern Recognition*, 17(2):177–187, 1984.
- [Mat75] G. Matheron. *Random Sets and Integral Geometry*. John Wiley & Sons, 1975.
- [Mey93] S. Beucher & F. Meyer. *Mathematical Morphology in Image Processing*, chapter 12. Marcel Dekker, Inc, 1993.
- [Pap62] A. Papoulis. *The Fourier Integral and its Applications*. McGraw-Hill, 1962.
- [Ser82] J. Serra. *Image Analysis and Mathematical Morphology*. Academic Press, 1982.
- [Ser88] J. Serra. *Image Analysis and Mathematical Morphology, volume 2: Theoretical Advances*. Academic Press, 1988.
- [Ser94] J. Serra. Morphological filtering: An overview. *Signal Processing*, (38):3–11, 1994.
- [Soi91] Luc Vincent & P. Soile. Watersheds in Digital Spaces: An Efficient Algorithm Based on Immersion Simulations. *IEEE Transactions on PAMI*, 13(6):583–598, 1991.
- [Vac95] Corinne Vachier. *Extraction de Caractéristiques, Segmentation d'Images et Morphologie Mathématique*. PhD thesis, ENSMP - Fontainebleau, França, 1995.
- [Vin93] Luc Vincent. Morphological Grayscale Reconstruction in Image Analysis: Applications and Efficient Algorithms. *IEEE Transactions on Image Processing*, 2(2):176–201, 1993.

- [Woo93] R. Gonzalez & R. R. Woods. *Digital Image Processing*. Addison-Wesley, 1993.
- [1] W.E. Higgins and E.J. Ojard, Interactive morphological watershed analysis for 3D medical images, *Computerized Medical Imaging and Graphics*, **17**(4/5), 1993, pp. 387–395.
- [2] I. Carlbom, D. Terzopoulos and K. Harris, Computer-assisted registration, segmentation, and 3D reconstruction from images of neuronal tissue sections, *IEEE Transactions on Medical Imaging*, **13**(2), 1994, pp. 351–362.
- [3] L. Cohen and R. Kimmel, Fast marching the global minimum of active contours, *Proceedings of the IEEE International Conference on Image Processing*, Lausanne, Sep, 1996, pp. 473–476.
- [4] A.X. Falcão, J.K. Udupa, S. Samarasekera, S. Sharma, B.E. Hirsch, and R.A. Lotufo, “User-steered image segmentation paradigms: live-wire and live-lane”, *Graphical Models and Image Processing*, **60**(4), Jul, 1998, pp 233–260.
- [5] A.E.O. Boudraa, J.J. Mallet, J.E. Besson, S.E. Bouyoucef and J. Champier, Left ventricle automated detection method in gated isotopic ventriculography using fuzzy clustering, *IEEE Transactions on Medical Imaging*, **12**(3), Sep, 1993, pp 451–465.
- [6] A.R. Cohen, B. Roysam and J.N. Turner, Automated tracing and volume measurements of neurons from 3D confocal fluorescence microscopy Data, *Journal of Microscopy*, **173**(2), Feb, 1994, pp. 103–114.
- [7] K.P. Philip, E.L. Dove, D.D. McPheron, N.L. Gotteiner, M.J. Vonesh, W. Stanford, J.E. Reed, J.A. Rumberg and K.B. Chandran, Automatic detection of myocardial contours in cine-computed tomographic images, *IEEE Transactions on Medical Imaging*, **13**(2), Jun, 1994, pp. 241–253.
- [8] A. Goshtasby and D.A. Turner, Segmentation of cardiac cine MR images for extraction of right and left ventricular chambers, *IEEE Transactions on Medical Imaging*, **14**(1), Mar, 1995, pp. 56–64.
- [9] R. Adams and L. Bischof, Seeded Region Growing, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **16**(6), Jun 1994, pp 641–647.
- [10] L. Najman and M. Schmitt, Geodesic Saliency of Watershed Contours and Hierarchical Segmentation, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **18**(12), Dec 1996, pp 1163–1173.
- [11] A. Martelli, Edge Detection Using Heuristic Search Methods, *Computer Graphics and Image Processing*, vol 1., no. 2, Aug 1972, pp 169–182.
- [12] A. Martelli, An Application of Heuristic Search Methods to Edge and Contour Detection, *ACM Communications Society*, vol. 19, no. 2, Feb 1976, pp. 73–83.

- [13] N.J. Nilsson, Principles of Artificial Intelligence, *Morgan Kauffman*, Palo Alto, CA, 1980.
- [14] D.H. Ballard and C.M. Brown, Computer Vision, *Prentice-Hall*, Englewood Cliffs, NJ, 1982.
- [15] M. Sonka, V. Hlavac and R. Boyle, Image Processing Analysis and Machine Vision, *Chapman & Hall Computing*, New York, NY, 1993.
- [16] A.C.M. Dumay, M.N.A.J. Claessens, C. Roos, J.J. Gerbrands and J.H.C. Reiber, Object Delineation in Noisy Images by a Modified Policy-Iteration Method, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 14, no. 9, Sep 1992, pp. 952-958.
- [17] U. Montanari, On the optimal detection of curves in noisy pictures, *ACM Communications Society*, vol. 14, no. 5, 1971, pp. 335-345.
- [18] D. Pope, D. Parker, D. Gustafson and P. Clayton, Dynamic Research Algorithms in Left Ventricular Border Recognition and Analysis of Coronary Arteries, *IEEE Proceedings of Computers in Cardiology*, 1984, pp. 71-75.
- [19] E.N. Mortensen and W.A. Barrett, Interactive segmentation with intelligent scissors. *Graphical Models and Image Processing*, vol. 60, no. 5, pp. 349-384, Sep, 1998.
- [20] A.X. Falcão, J.K. Udupa, and F.K. Miyazawa, "An ultra-fast user-steered segmentation paradigm: live-wire-on-the-fly", in *Proceedings of SPIE on Medical Imaging*, San Diego, CA, 1999, submitted.
- [21] R.K. Ahuja, T.L. Magnanti and J.B. Orlin, Network Flows: Theory, Algorithms and Applications, *Prentice-Hall*, Englewood Cliffs, NJ, 1993.
- [22] J.F. BRINKLEY. A flexible, generic model for anatomic shape: application to interactive two-dimensional medical image segmentation and mathing. *Computers on Biomedical Research*, 26:121-142, 1993.
- [23] Vikram CHALANA, Wendy COSTA, and Yongmin KIM. Integrating region growing and edge detection using regularization. *In Proceedings of the SPIE Conference on Medical Imaging*, 1995.
- [24] Laurent D. COHEN and Isaac COHEN. Finite- element methods for active contour models and balloons for 2-d and 3-d images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(11), November 1993.
- [25] Kok Fung LAI. *Deformable Contours: Modeling, Extraction, Detection and Classification*. PhD thesis, University of Wisconsin-Madison, 1994.

- [26] F. LEYMARIE and M.D. LEVINE. Tracking deformable objects in the plane using an active contour model. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 15:617–633, 1993.
- [27] Steven LOBREGT and Max. A. VIERGEVER. A discrete dynamic contour model. *IEEE Transactions on Medical Imaging*, 14(1), March 1995.
- [28] Blair MACKIEWICH. Active contour models (snakes). URL : <http://www.cs.sfu.ca/stella/papers/blairthesis/main/node29.htm>, 1995.
- [29] Tim McINERNEY and Demetri TERZOPOULOS. Deformable models in medical image analysis: a survey. *Medical Image Analysis*, 1(2):91–108, 1996.
- [30] W. NUENSCHWAANDER, P. FUA, and O KUBLER. Initializing snakes. *IEEE Conference on Computer Vision and Pattern Recognition*, pages 658–663, 1994.
- [31] Michael KASS, Andrew WITKIN, and Demetri TERZOPOULOS. Snakes: Active contour models. *International Journal of Computer Vision*, 1:321–331, 1987.
- [32] L.D. COHEN and I COHEN. Using a finite element method for active contour models and 3d reconstruction from cross sections. *Proc. Third International Conference on Computer Vision*, pages 587–591, 1990.
- [33] L.D. COHEN and I COHEN. Using deformable surfaces to segment 3d images and infer differential structures. *CVGIP: Image Understanding*, 56(2):242–263, September 1992.
- [34] J.D. BOISSONNAT. Shape reconstruction from planar cross sections. *Computer Vision, Graphics, and Image Processing*, pages 1–29, 1988.
- [35] MO.O BERGER and R. MOHR. Towards autonomy in active contour models. *Proceedings of the Tenth International Conference on Pattern Recognition*, pages 847–851, 1990.
- [36] F. LEYMARIE and M.D. LEVINE. Simulating the grassfire transform using an active contour model. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 14(1):56–75, 1992.
- [37] A. A. AMINI, T. WEYMOUTH, and R. JAIN. Using dynamic programming for solving variational problems in vision. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12:855–867, 1990.
- [38] S.W. ZUCKER, A. DOBBINS, and L. IVERSON. Two stages of curve suggest two styles of visual computation. *Neural Computation*, 1:68–81, 1989.
- [39] D. COOPER, H. ELLIOT, F. COHEN, L. REISS, and P SYMOSEK. Stochastic boundary estimation and object recognition. *Computer Graph. and Image Proc.*, 12:326–356, 1980.

- [40] R.J. Schalkoff. Pattern Recognition: Statistical, Structural, and Neural Approaches. *John Wiley & Sons, Inc.*, 1992.
- [41] R.O. Duda and P.E. Hart. Pattern Classification and Scene Analysis. *John Wiley & Sons, Inc.*, New York, NY, 1973.