

Notas de Aula da Disciplina

# Introdução ao Processamento Digital de Imagens

Leonardo Vidal Batista

2005

# Conteúdo

	Capítulo 1 Introdução .....	
1.1	Processamento de Imagens e Áreas Correlatas .....	4
1.2	A Faixa Visível do Espectro Eletromagnético .....	13
1.3	A Estrutura do Olho Humano.....	14
1.4	Sistemas de Cores.....	16
1.5	Outros sistemas de Cores .....	18
	Capítulo 2 Um Modelo de Imagem Digital.....	
2.1	Imagens Monocromáticas.....	21
2.2	Amostragem e Quantização.....	23
2.3	Imagens Coloridas .....	27
2.4	Operações envolvendo imagens .....	28
	Capítulo 3 Dispositivos Gráficos .....	
3.1	Impressoras.....	31
3.2	Monitores de Vídeo e Aparelhos de TV.....	31
3.3	Memória de Vídeo.....	33
	Capítulo 4 Filtros Espaciais .....	
4.1	Filtros Lineares e Invariantes ao deslocamento.....	<b>Erro! Indicador não definido.</b>
4.2	Convolução.....	8
4.3	Operações Pontuais e Locais .....	36
4.4	Negativo .....	38
4.5	Controle de Contraste.....	41
4.6	Equalização e Expansão de Histograma.....	42
4.7	Filtros de Suavização.....	48
4.8	Filtros de Aguçamento .....	48
4.9	Operações Separáveis.....	52
	Capítulo 5 Operações Algébricas .....	
5.1	Dissolve Cruzado .....	54
5.2	Detecção de Movimento.....	55
5.3	Redução de Ruído por Média de Imagens.....	56
	Capítulo 6 Operações Topológicas .....	
6.1	Rotação.....	59
6.2	Expansão e Contração .....	59
6.3	<i>Pinch</i> na Vertical .....	61
6.4	Zoom .....	64
6.5	Warping Baseado em Campos .....	66
6.6	Morphing Baseado em Campos .....	69
	Capítulo 7 Transformadas.....	
7.1	Transformada Cosseno Discreta.....	73
7.2	Transformada de Fourier Discreta de Funções de Uma Variável.....	74
7.3	Transformada de Fourier Discreta de Funções de Duas Variáveis.....	75
7.4	Filtragem no Domínio da Frequência.....	80

# Capítulo 1

## Introdução

---

A utilização de imagens digitais tem experimentado um crescimento explosivo nas últimas décadas. É quase impossível criar uma lista exaustiva das aplicações modernas que envolvem imagens digitais, mas pode-se citar, a título de ilustração, algumas áreas de especial relevância:

- Sistemas de televisão digital;
- Câmeras fotográficas digitais e *scanners*;
- DVDs;
- Sistemas de teleconferência;
- Transmissões via fax;
- Monitoramento de regiões da superfície terrestre e previsão climática auxiliada por imagens de satélites;
- Imagens e vídeos digitais na Internet;
- Técnicas de diagnóstico médico por imagens, tais como ultrassonografia, angiografia, tomografia computadorizada e ressonância magnética;
- Realce e restauração de imagens por computador;
- Controle de acesso por identificação biométrica, utilizando, por exemplo, reconhecimento de faces, íris ou impressões digitais;
- Videogames;
- Realidade virtual;
- Simuladores para treinamento ou entretenimento;
- Robótica;
- Efeitos especiais em filmes
- Editoração eletrônica
- Impressoras
- Controle de qualidade
- Instrumentação, metrologia e controle

- Ciências Forenses
- Aplicações Militares

O processamento digital de imagens é a área do conhecimento que trata da modelagem matemática, análise, projeto e implementação em software e em hardware de sistemas digitais voltados ao tratamento de informação pictórica com o propósito de torná-la mais adequada à interpretação por seres humanos ou máquinas ou para obter maior eficiência de armazenamento e transmissão. As técnicas de processamento digital de imagens são altamente multidisciplinares, envolvendo diversos aspectos de ótica, eletrônica, matemática, fotografia e computação.

## **1.1 PROCESSAMENTO DE IMAGENS E ÁREAS CORRELATAS**

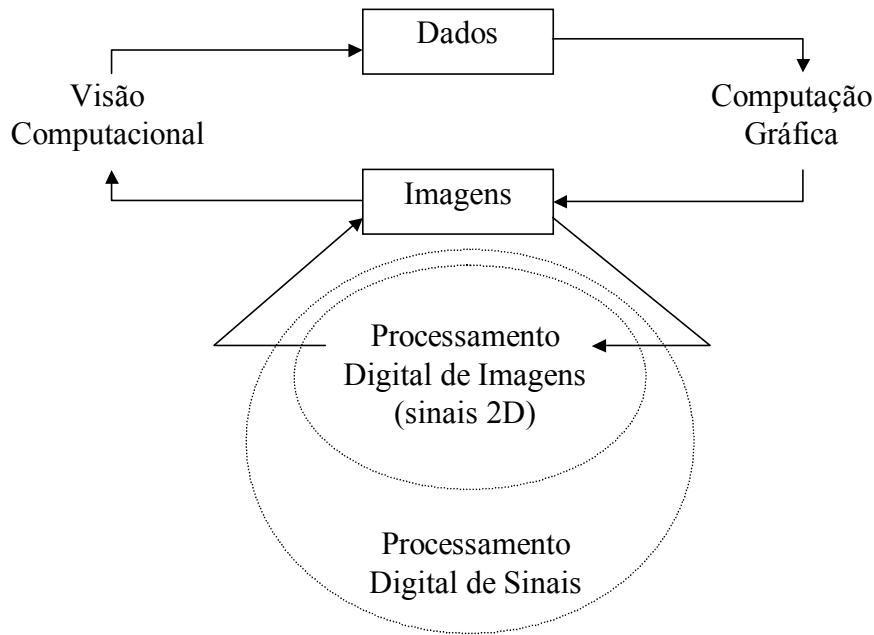
Muitos itens da lista anterior pertencem também ao domínio de dois campos intimamente relacionados com o de processamento digital de imagens: Computação Gráfica e Visão Computacional. Para definir mais precisamente o escopo deste texto, vejamos as diferenças mais relevantes entre estas áreas.

Na Computação Gráfica os dados de entrada são processados e utilizados para gerar imagens como produto final. Se os dados de entrada representam as coordenadas espaciais de objetos constituintes de um cenário, um sistema de computação gráfica poderá gerar uma imagem que permita visualizar o cenário descrito matematicamente. As aplicações da computação gráfica incluem programas de projeto auxiliado por computador (*computer aided design* - CAD), muito populares em engenharia; realidade virtual; visualização científica; efeitos especiais em filmes; interação humano-computador; etc.

A Visão Computacional percorre, de certa forma, um caminho inverso ao da computação gráfica. Agora, imagens de entrada são analisadas com vistas à obtenção de dados contendo informações geométricas ou físicas sobre o cenário. As aplicações incluem: robótica; sistemas automatizados de controle de acesso por imagens; reconhecimento de impressões digitais por computador; diagnóstico automatizado de enfermidades; etc.

No Processamento Digital de Imagens (PDI), importante subdivisão da área de Processamento Digital de Sinais e objetivo deste curso, imagens de entrada são processadas, gerando imagens como produto final. O PDI pode incluir etapas de visão computacional e de computação gráfica. Desde que a entrada e a saída do sistema envolvam imagens, o trabalho global é fundamentalmente de PDI. O PDI é altamente multidisciplinar, envolvendo diversos aspectos e conceitos de ótica, eletrônica, matemática, fotografia e computação.

O diagrama apresentado na Figura 1.1 ilustra as relações entre as áreas que constituem os fundamentos das técnicas computacionais relacionadas a imagens.



**Figura 1.1 Processamento digital de imagens e áreas correlatas**

Muitas vezes essas três áreas atuam de modo cooperativo e indissociável. Uma imagem de satélite de parte da Amazônia pode ser inicialmente segmentada, produzindo-se uma nova imagem com regiões claramente discerníveis. As diferentes regiões encontradas no processo de segmentação podem ser identificadas por cores bem distintas. Esta é uma operação típica de PDI. Quando informações geométricas, estatísticas, de textura, ou quaisquer outras são extraídas e utilizadas para rotular as diversas regiões como rios, florestas, queimadas, zonas urbanas, plantações e pastos, por exemplo, tem-se um procedimento de visão computacional, com as imagens sendo analisadas e gerando dados descritivos das regiões. As informações de descrição das regiões podem finalmente ser utilizadas por um programa de computação gráfica que gerará uma imagem mais agradável ou útil à apreciação humana. Assim, por exemplo, as regiões de florestas podem ser representadas na imagem final como áreas verdes com textura semelhante à das copas das árvores e as áreas urbanas por representações estilizadas de agrupamentos de prédios.

Perceba-se que as ações acima descritas, tomadas em conjunto, descrevem essencialmente uma aplicação de PDI, pois o sistema como um todo mapeia imagens em imagens.

Vê-se, portanto, que a visão computacional atua no escopo da análise de imagens, a computação gráfica na síntese de imagens, e o PDI pode envolver ambas as operações. Além de constituir um campo autônomo de pesquisa e aplicação, o PDI é comumente encontrado nas etapas de geração da imagem e de pós-processamento em computação gráfica e na fase de pré-processamento em visão computacional.

## 1.2 SISTEMAS NO TEMPO DISCRETO

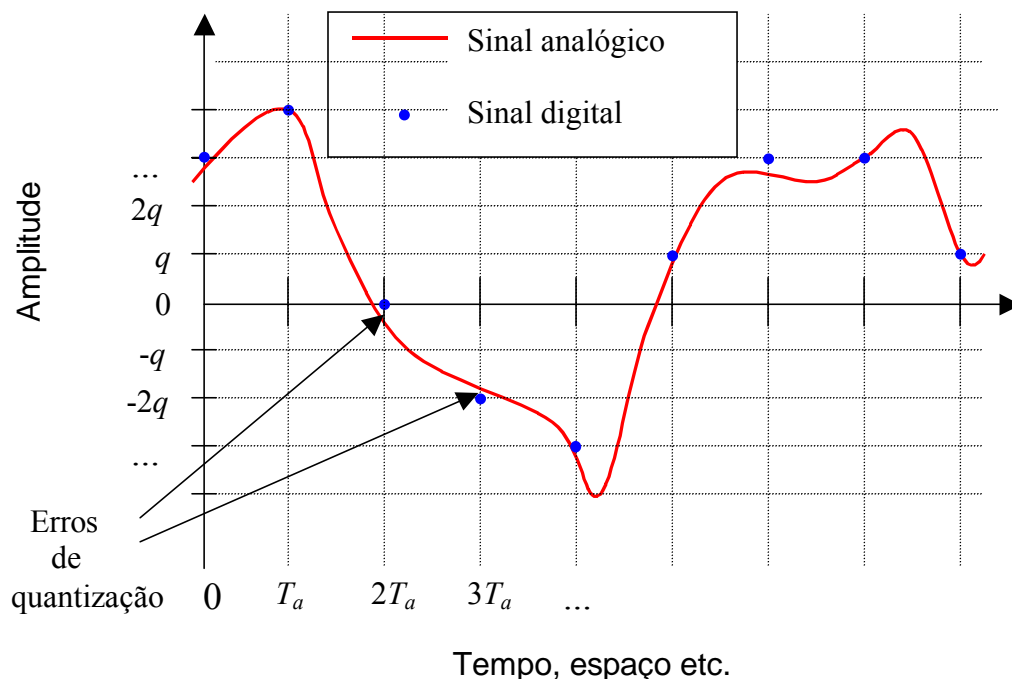
Entende-se por *senal* qualquer função de uma ou mais variáveis que conduz informação a respeito de um fenômeno de interesse. Como veremos no Capítulo 2, uma imagem é simplesmente um sinal bidimensional, ou seja, uma função de duas variáveis, que pode ser tratada em muitas situações como um conjunto de sinais unidimensionais. Para conservar a simplicidade de representação, é conveniente introduzir muitos dos conceitos fundamentais de processamento de imagens utilizando-se sinais unidimensionais. A transposição dos conceitos para o caso bidimensional é quase sempre imediata.

### 1.2.1 Amostragem e Quantização

Um sinal analógico ou contínuo pode ser expresso matematicamente como uma função contínua de uma variável contínua. Utilizaremos a notação usual de funções matemáticas para representar sinais contínuos. Por exemplo,  $x(t)$  é um sinal contínuo de uma variável contínua  $t$ .

Para ser utilizado em um computador, um sinal contínuo deve ser digitalizado. O processo de digitalização envolve discretização do domínio e do contradomínio do sinal. A discretização do domínio é denominada *amostragem*, e a discretização do contradomínio é denominada *quantização*.

A Figura 1.2 ilustra o processo mais comum de digitalização de um sinal unidimensional. Neste processo, efetua-se uma amostragem uniforme, que consiste em colher amostras do sinal a intervalos regulares. O espaçamento entre as amostras,  $T_a$ , denominado período de amostragem, é geralmente – mas não necessariamente – medido em unidades de espaço ou tempo. define-se a frequência de amostragem como sendo  $f_a = 1/T_a$  (amostras/unidade de espaço ou de tempo, por exemplo). Os valores das amostras colhidas são então aproximados para valores de um conjunto finito enumerável  $Q = \{-nq, \dots, -2q, -q, 0, q, 2q, \dots, mq\}$ , com  $m$  e  $n$  inteiros, em um processo denominado *quantização uniforme*. Há outras formas de quantização, mas a quantização uniforme é a mais utilizada na prática. Os valores  $iq$ ,  $i = -n, \dots, -2, -1, 0, 1, 2, \dots, m$ , são os *níveis de quantização*, e  $q$  define o *tamanho do passo de quantização*. A qualidade do sinal digital resultante é definida principalmente pelos parâmetros  $T_a$  e  $q$ . Observe-se na Figura 1.2 a ocorrência de *erros de quantização* causados pela aproximação dos valores originais para valores presentes em  $Q$ .



**Figura 1.2. Digitalização de um sinal unidimensional**

O processo de digitalização de um sinal unidimensional gera uma sequência de valores. Se  $x_a(t)$  é um sinal analógico, representaremos por  $x[n]$  o  $n$ -ésimo valor do sinal digital resultante do processo de amostragem e quantização de  $x_a(t)$ :

$$x[n] = \text{Quant}(x_a(nT_a)), n = 0, 1, 2, \dots$$

onde  $\text{Quant}(\cdot)$  representa a operação de quantização de uma amostra. No restante deste capítulo, utilizaremos  $n$  para representar uma variável discreta.

Nem todos os sinais digitais advêm da digitalização de sinais analógicos. Alguns sinais são digitais por natureza, e outros são matematicamente definidos na forma discreta. Por exemplo, o número de negócios fechados em cada pregão da BOVESPA é, inerentemente, um sinal digital. Alguns sinais matemáticos de especial interesse são:

Impulso unitário ou delta de Kronecker:

$$\delta[n] = \begin{cases} 1, n = 0 \\ 0, n \neq 0 \end{cases}$$

Degrau unitário:

$$u[n] = \begin{cases} 1, n \geq 0 \\ 0, n < 0 \end{cases}$$

Sinal senoidal:

$$s[n] = \text{sen}(2\pi fn)$$

Observe-se que esta última função apresenta domínio discreto, mas  $s[n]$  não é quantizado. Este tipo de sinal é discreto, mas não digital. Sinais discretos podem ser utilizados em desenvolvimentos teóricos e estabelecimento de conceitos; contudo, o processamento por computador exige a digitalização.

### 1.3 SISTEMAS DISCRETOS

Um sistema discreto mapeia um sinal discreto de entrada,  $x[n]$ , em um sinal discreto de saída,  $y[n]$ , através da aplicação de um operador  $T$  sobre  $x[n]$ :

$$y[n] = T\{x[n]\}$$

Um sistema discreto, representado pelo operador  $T$ , pode ser classificado de várias formas, algumas das quais descritas a seguir.

#### 1.3.1 Sistemas lineares.

Um sistema é linear se e somente se

$$T\{ax_1[n] + bx_2[n]\} = aT\{x_1[n]\} + bT\{x_2[n]\}$$

para quaisquer constantes  $a$  e  $b$  e quaisquer sinais  $x_1[n]$  e  $x_2[n]$ .

### 1.3.2 Sistemas invariantes ao deslocamento

Um sistema é invariante ao deslocamento se e somente se

$$T\{x_1[n-a]\} = y[n-a],$$

com  $y[n] = T\{x[n]\}$ , para qualquer constante  $a$  e qualquer sinal  $x[n]$ .

Quando  $n$  representa coordenadas espaciais ou temporais, estes sistemas são denominados, respectivamente, invariantes no espaço ou invariantes no tempo.

### 1.3.3 Sistemas Causais

Um sistema é causal se e somente se sua saída  $y[a]$  não depende de entradas  $x[n]$  com  $n > a$ .

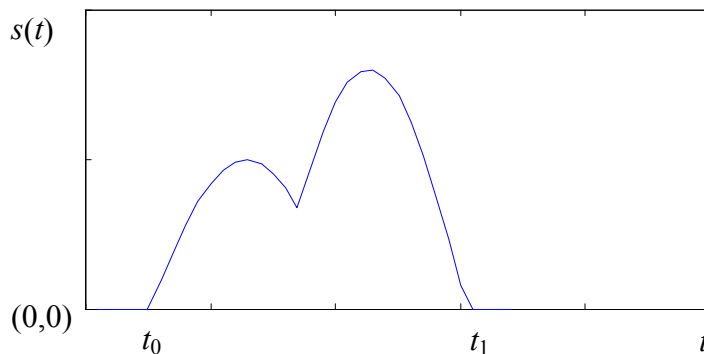
## 1.4 CONVOLUÇÃO

Sistemas lineares invariantes ao deslocamento podem ser matematicamente descritos por uma operação conhecida como *convolução*. Para introduzir este conceito fundamental em processamento de sinais, consideremos inicialmente o caso contínuo unidimensional. A *convolução* entre duas funções  $s(t)$  e  $h(t)$ , representada por  $s(t) * h(t)$ , gera uma função  $g(t)$  dada por

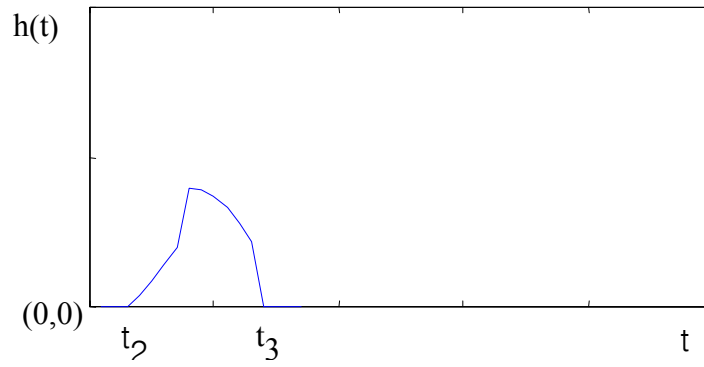
$$g(t) = s(t) * h(t) = \int_{-\infty}^{\infty} s(\tau)h(t - \tau)d\tau$$

A operação de convolução não é de visualização trivial. Um exemplo gráfico pode ajudar a esclarecer o mecanismo. Para simplificar, tomemos duas funções, esboçadas na Figura 1.3, com as seguintes características:

$$\begin{aligned} s(t) &= 0 \text{ para } t \notin [t_0, t_1] \\ h(t) &= 0 \text{ para } t \notin [t_2, t_3] \end{aligned}$$

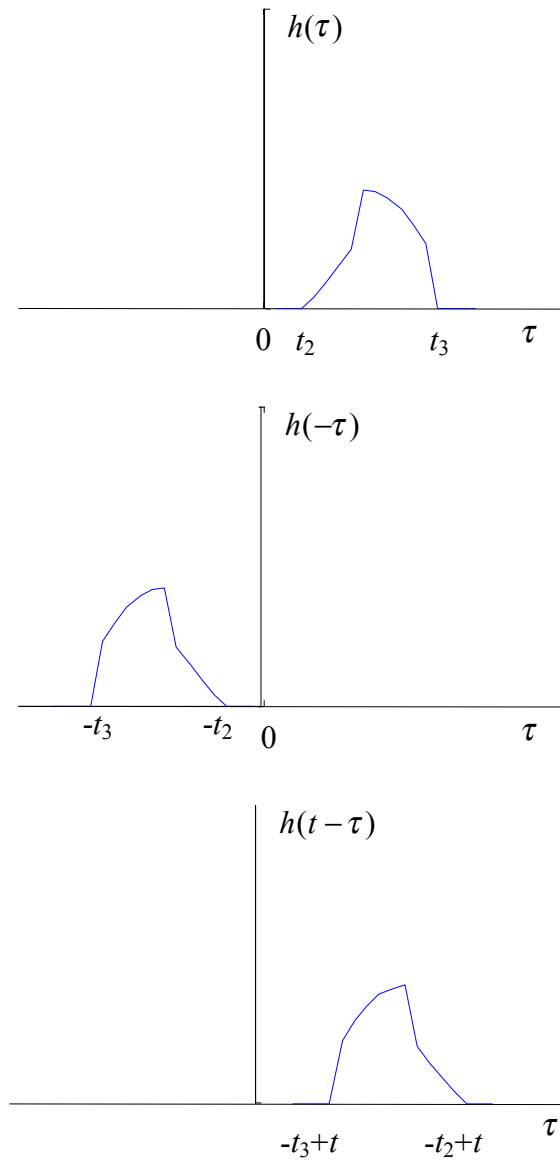






**Figura 1.3.** Dois sinais unidimensionais: (a)  $s(t)$  e (b)  $h(t)$

De acordo com a equação de convolução, para obter o valor de  $g(t)$  para um dado  $t$ , deve-se efetuar a integração do produto de duas funções da variável  $\tau$ ,  $s(\tau)$  e  $h(t-\tau)$ . A função  $h(t-\tau)$  é obtida efetuando-se o rebatimento de  $h(\tau)$  em relação à origem e um deslocamento da função resultante por  $t$ , conforme ilustra a Figura 1.4.



**Figura 1.4.** Rebatimento e deslocamento: (a)  $h(\tau)$ ; (b)  $h(-\tau)$ ; (c)  $h(t-\tau)$

O produto  $s(\tau) h(t-\tau)$  é, conseqüentemente, a convolução  $g(t) = s(t) * h(t)$ , será nulo para qualquer valor de  $t$  tal que não haja interseção entre os intervalos em que  $s(\tau)$  e  $h(t-\tau)$  assumem valores não nulos, ou seja,  $[t_0, t_1]$  e  $[-t_3 + t, -t_2 + t]$ . Mais especificamente,  $g(t) = 0$  para todo  $t$  tal que

$$\begin{aligned} -t_3 + t > t_1 &\Leftrightarrow t > t_1 + t_3 \text{ ou} \\ -t_2 + t < t_0 &\Leftrightarrow t < t_0 + t_2 \end{aligned}$$

Em resumo,  $g(t) = 0$  para  $t \notin [t_0 + t_2, t_1 + t_3]$ .

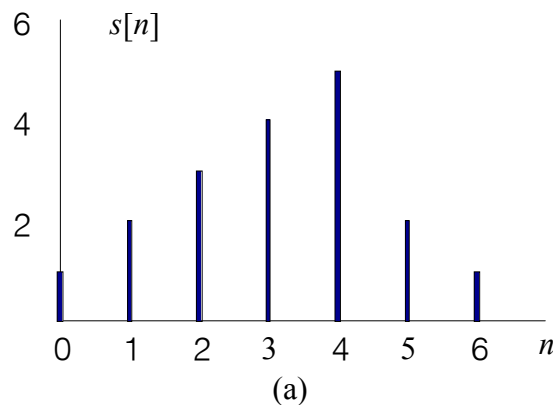
A convolução discreta é uma extensão imediata do conceito de convolução contínua. Se  $s[n]$  e  $h[n]$  são seqüências discretas definidas para todo inteiro  $n$ , a *convolução linear*  $g[n] = s[n] * h[n]$  é dada por:

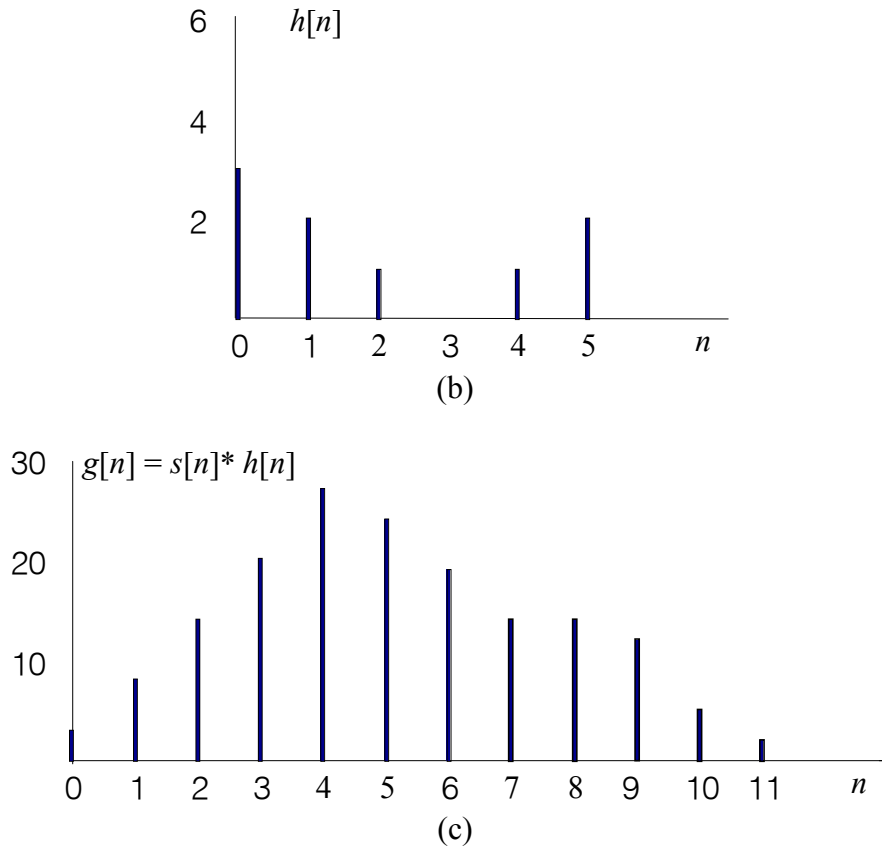
$$g[n] = s[n] * h[n] = \sum_{\tau=-\infty}^{\infty} s[\tau] h[n-\tau]$$

Nas situações práticas mais comuns, as seqüências de entrada são finitas. Se  $s[n]$  e  $h[n]$  representam as seqüências  $s[0], s[1], \dots, s[N_0-1]$  e  $h[0], h[1], \dots, h[N_1-1]$ , respectivamente, os valores de  $s[n]$  para  $n \notin [0, N_0-1]$  e de  $h[n]$  para  $n \notin [0, N_1-1]$  não estão definidos e, portanto, a equação de convolução linear não se aplica diretamente. Neste caso,  $s[n]$  e  $h[n]$  devem ser estendidos para todo inteiro  $x$  antes da convolução. Uma possibilidade é considerar  $s[n] = 0$  para  $n \notin [0, N_0-1]$ , e  $h[n] = 0$  para  $n \notin [0, N_1-1]$ . Por um raciocínio análogo àquele que levou à definição do intervalo no qual a convolução contínua podia assumir valores não nulos, pode-se mostrar facilmente que a convolução discreta  $g[n]$  com extensão por zeros pode apresentar valores não nulos apenas para  $n \in [0, N_0 + N_1 - 1]$ . Para um dado valor de  $n$  neste intervalo, o produto  $s[\tau] h[n-\tau]$  poderá assumir valores não nulos apenas para  $\tau \in [0, n]$ . A expressão de convolução pode então ser particularizada:

$$g[n] = s[n] * h[n] = \sum_{\tau=0}^{N-1} s[\tau] h[n-\tau]$$

com  $N = N_0 + N_1 - 1$ . A Figura 1.5 mostra a convolução de duas seqüências  $s[n]$  e  $h[n]$ , com 7 e 6 elementos, respectivamente, resultando em uma seqüência  $g[n]$  com  $7+6-1=12$  elementos.





**Figura 1.5 Convolução.** (a)  $s = [1, 2, 3, 4, 5, 2, 1]$ ; (b)  $h = [3, 2, 1, 0, 1, 2]$ ; (c)  $g = s * h = [3, 8, 14, 20, 27, 24, 19, 14, 14, 12, 5, 2]$

Um conceito fundamental em processamento digital de sinais é o de *convolução circular*. A convolução circular fornece uma ponte entre o domínio original e o domínio da frequência, apresentando grande interesse teórico e prático, conforme será visto no Capítulo 7. Para a definição da convolução circular, inicialmente as seqüências  $s[n]$  e  $h[n]$ , que contêm  $N_0$  e  $N_1$  elementos, respectivamente, são expandidas para  $N$  elementos, com  $N \geq N_0$  e  $N \geq N_1$ , empregando-se extensão por zeros, gerando as seqüências  $s_e[n]$  e  $h_e[n]$ :

$$s_e[n] = \begin{cases} s[n], & 0 \leq n < N_0 \\ 0, & N_0 \leq n < N \end{cases}$$

$$h_e[n] = \begin{cases} h[n], & 0 \leq n < N_1 \\ 0, & N_1 \leq n < N \end{cases}$$

Efetua-se então uma *extensão periódica* das seqüências  $s_e[n]$  e  $h_e[n]$ , considerando-se que  $s_e[n]$  constitui um período de uma seqüência periódica  $s_p[n]$  e que  $h_e[n]$  constitui um período de uma seqüência periódica  $h_p[n]$ . Define-se a *convolução circular*  $g_p = s[n] \otimes h[n]$  como

$$g_p = s[n] \otimes h[n] = \sum_{\tau=0}^{N-1} s_p[\tau] h_p[n - \tau]$$

Considere as seqüências finitas  $s_1[0], s_1[1], \dots, s_1[N_0-1]$  e  $h_1[0], h_1[1], \dots, h_1[N_1-1]$ , e as seguintes seqüências  $s_2[n]$  e  $h_2[n]$ , definidas para todo inteiro  $n$ :

$$s_2[n] = \begin{cases} s_1[n], & 0 \leq n < N_0 \\ 0, & \text{caso contrário} \end{cases}$$

$$h_2[n] = \begin{cases} h_1[n], & 0 \leq n < N_1 \\ 0, & \text{caso contrário} \end{cases}$$

Fazendo-se  $N = N_0 + N_1 - 1$ , pode-se mostrar que  $s_1[n] \otimes h_1[n] = s_2[n] * h_2[n]$ ,  $0 \leq n < N$ . Assim sendo, pode-se calcular a convolução circular por intermédio de uma convolução linear.

Os conceitos acima apresentados podem ser diretamente estendidos para matrizes. A convolução linear  $g[i, j] = f[i, j] * h[i, j]$ , com  $f$  e  $g$  definidas para todo  $i$  e para todo  $j$  inteiros, é dada por

$$g[i, j] = f[i, j] * h[i, j] = \sum_{\alpha=-\infty}^{\infty} \sum_{\beta=-\infty}^{\infty} f[\alpha, \beta] h[i - \alpha, j - \beta]$$

Se  $f[i, j]$  está definida apenas para  $0 \leq i < R_0$  e  $0 \leq j < C_0$ , e  $h[i, j]$  está definida apenas para  $0 \leq i < R_1$  e  $0 \leq j < C_1$ , pode-se considerar que  $f[i, j]$  são nulas para  $i$  e  $j$  fora destes intervalos, e calcular a convolução como:

$$g[i, j] = f[i, j] * h[i, j] = \sum_{\alpha=0}^{R-1} \sum_{\beta=0}^{C-1} f[\alpha, \beta] h[i - \alpha, j - \beta]$$

com  $R = R_0 + R_1 - 1$  e  $C = C_0 + C_1 - 1$ .

Para o cálculo da convolução circular, as matrizes  $f[i, j]$  ( $0 \leq i < R_0$  e  $0 \leq j < C_0$ ) e  $h[i, j]$  ( $0 \leq i < R_1$  e  $0 \leq j < C_1$ ), são expandidas para  $R \times C$  elementos, com  $R \geq R_0$  e  $R \geq R_1$ ,  $C \geq C_0$  e  $C \geq C_1$ , empregando-se extensão por zeros, gerando as matrizes  $f_e[i, j]$  e  $h_e[i, j]$ :

$$f_e[i, j] = \begin{cases} f[i, j], & 0 \leq i < R_0 \text{ e } 0 \leq j < C_0 \\ 0, & R_0 \leq i < R \text{ ou } C_0 \leq j < C \end{cases}$$

$$h_e[i, j] = \begin{cases} h[i, j], & 0 \leq i < R_1 \text{ e } 0 \leq j < C_1 \\ 0, & R_1 \leq i < R \text{ ou } C_1 \leq j < C \end{cases}$$

Efetua-se então uma *extensão periódica* das matrizes  $f_e[i, j]$  e  $h_e[i, j]$ , considerando-se que  $f_e[i, j]$  constitui um período de uma função bidimensional periódica  $f_p[i, j]$  e que  $h_e[n]$  constitui um período de uma função bidimensional periódica  $h_p[i, j]$ . Define-se então a *convolução circular*  $g_p[i, j] = f[i, j] \otimes h[i, j]$  como

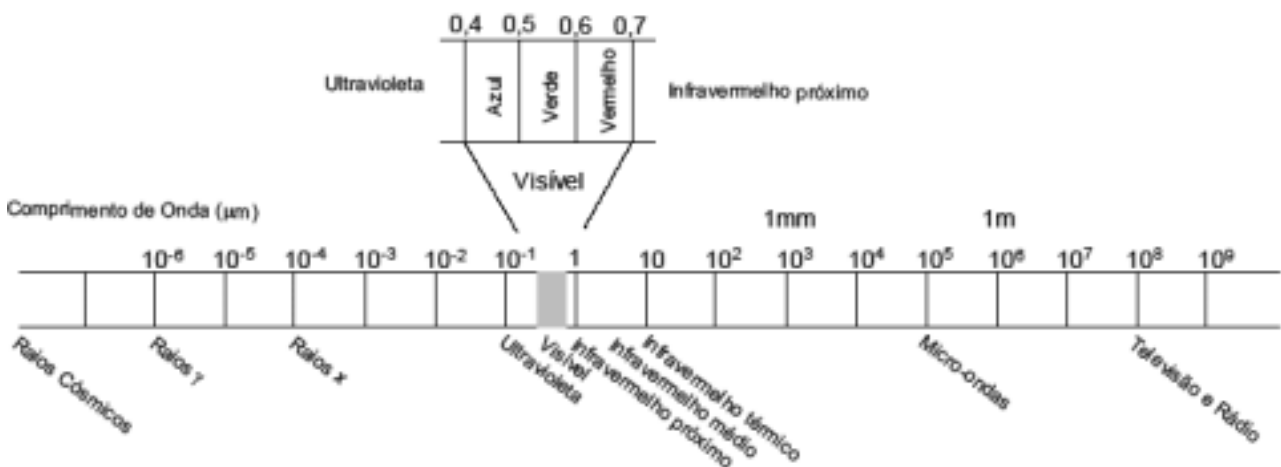
$$g_p[i, j] = f[i, j] \otimes h[i, j] = \sum_{\alpha=0}^{R-1} \sum_{\beta=0}^{C-1} f_p[\alpha, \beta] h_p[i - \alpha, j - \beta]$$

Como no caso unidimensional, pode-se calcular a convolução circular por intermédio de uma convolução linear limitada a  $0 \leq i < R$  e  $0 \leq j < C$ , desde que  $R = R_0 + R_1 - 1$  e  $C = C_0 + C_1 - 1$ .

Operações de filtragem lineares e invariantes ao deslocamento correspondem a uma convolução linear de uma imagem de entrada  $f[i, j]$  com uma matriz  $h[i, j]$  que define o filtro. Esta matriz recebe diversas denominações, tais como máscara convolucional, núcleo do filtro e resposta ao impulso.

## 1.5 A FAIXA VISÍVEL DO ESPECTRO ELETROMAGNÉTICO

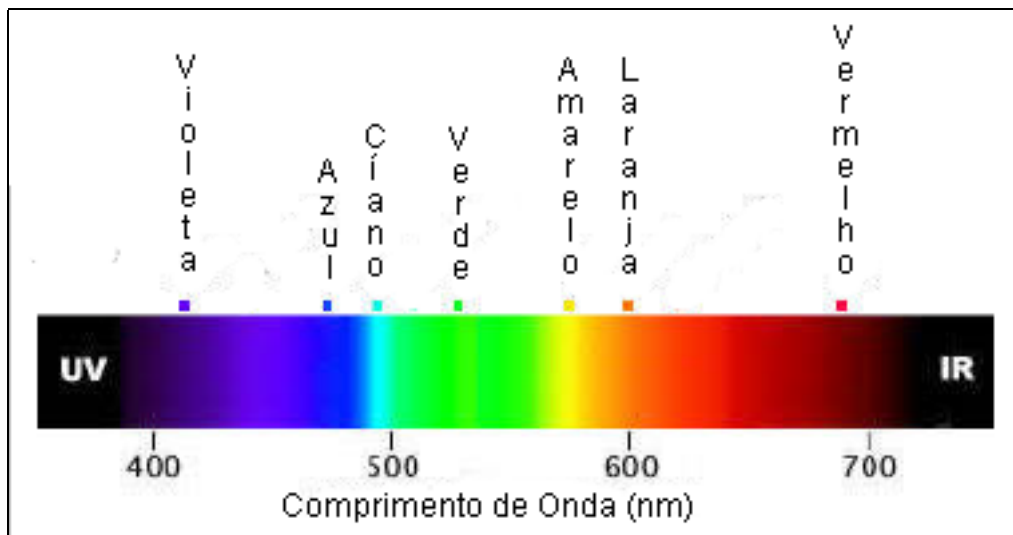
A luz é uma radiação eletromagnética e, assim sendo, apresenta um comportamento ondulatório com frequência e comprimento de onda característicos. A faixa visível do espectro eletromagnético se estende aproximadamente de 380 nm a 780 nm. A Figura 1.6 apresenta a denominação usual dada a diversas regiões dos espectro eletromagnético.



**Figura 1.6 O espectro eletromagnético. Fonte: Integração de Dados Estatísticos na Classificação de Imagens / Fernando Rocha, 2002**

Nenhuma radiação eletromagnética com comprimento de onda fora da faixa visível é percebida pelo olho humano. Dentro da faixa visível, o olho humano percebe comprimentos de onda diferentes como *cores* diferentes. Uma fonte que emite radiação com um comprimento de onda único é chamada de *monocromática*, e a cor da radiação é denominada *cor espectral pura* se a radiação estiver na faixa visível.

A Figura 1.7 e a Tabela 1.1 apresentam a faixa visível e as denominações usuais das cores associadas a cada região do espectro.



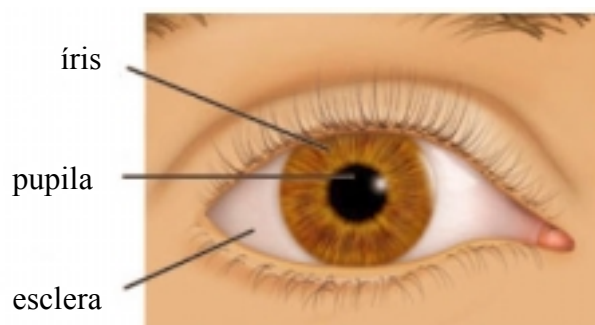
**Figura 1.7 O espectro eletromagnético visível. Obs: nm = nanômetro =  $10^{-9}$  m**

**Tabela 1.1 Denominação usual das cores**

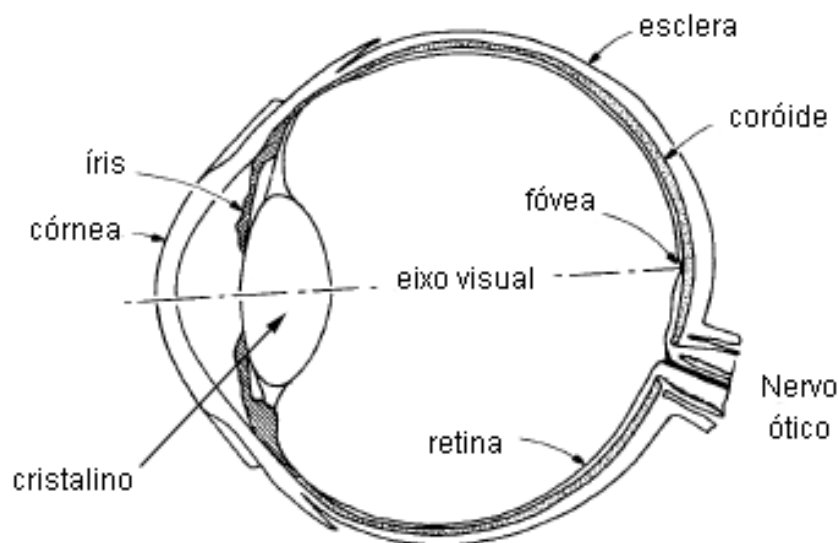
Denominação Usual da Cor	Faixa do Espectro (nm)
Violeta	380 – 440
Azul	440 – 490
Verde	490 – 565
Amarelo	565 – 590
Laranja	590 – 630
Vermelho	630 – 780

## 1.6 A ESTRUTURA DO OLHO HUMANO

O olho humano é aproximadamente esférico, com diâmetro médio em torno de dois centímetros. Algumas estruturas externas e internas do olho humano podem ser observadas na Figura 1.8 e na Figura 1.9. Os raios luminosos provenientes dos objetos externos penetram no olho através de uma abertura frontal na íris, a *pupila*, e de uma lente denominada cristalino, atingindo a *retina*, que constitui a camada interna posterior do globo ocular. Quando o olho está apropriadamente focalizado, forma-se uma imagem nítida invertida do cenário externo sobre a retina.



**Figura 1.8 Aspecto externo do olho humano**



**Figura 1.9 A estrutura interna do olho humano. Fonte: Digital Image Processing: PIKS inside / W. Prat, - 3rd Ed., 2001**

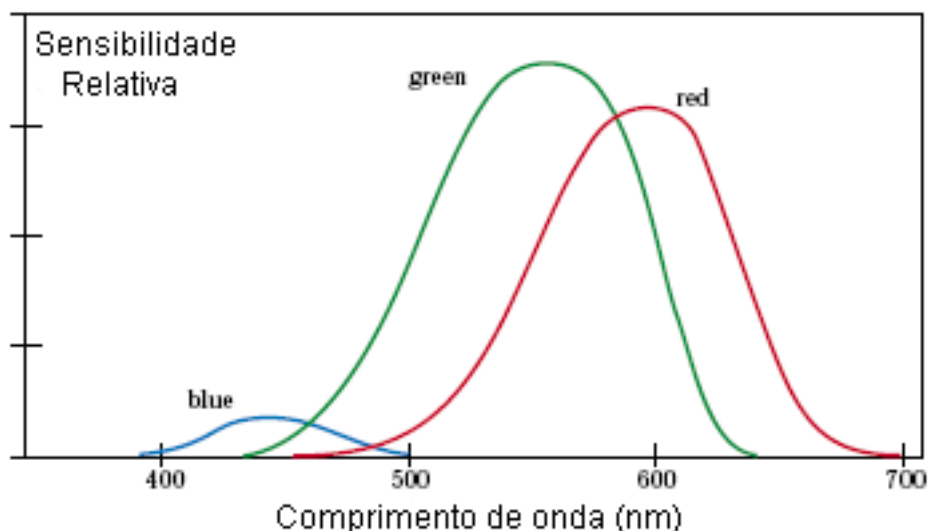
Sobre a retina distribuem-se dois tipos de elementos fotossensíveis, os *cones* e os *bastonetes*, cuja função é converter energia luminosa em impulsos elétricos que são transmitidos até o cérebro para serem interpretados.

Há de 75 a 150 milhões de bastonetes em cada olho. Os bastonetes são relativamente bem distribuídos sobre toda a retina, não são sensíveis às cores, e não permitem boa resolução de detalhes, principalmente por operarem em grupo, sendo cada grupo conectado a um terminal nervoso. Os bastonetes proporcionam uma visão geral e, por serem sensíveis a baixos níveis de iluminação, proporcionam a chamada *visão de baixa luminosidade*. Objetos coloridos à luz do dia parecem acinzentados sob o luar porque apenas os bastonetes são estimulados.

Há de seis a sete milhões de cones em cada olho, localizados principalmente na região central da retina denominada fóvea. Os cones são sensíveis às cores, e permitem alta resolução de detalhes, principalmente porque cada um deles está conectado ao seu próprio terminal nervoso. Quando olhamos para um objeto, o olho é posicionado pelos músculos que o controlam de forma que a imagem do objeto recaia sobre a fóvea. Por atuarem mais eficientemente sob boas condições de iluminação, os cones proporcionam a chamada *visão de alta luminosidade*.

O olho humano possui três tipos de cones, um mais sensível à faixa do espectro correspondente ao violeta e ao azul; outro ao verde; e outro ao amarelo, ao laranja e ao vermelho. Diz-se que o olho apresenta cones “azuis”, “verdes” e “vermelhos”. Devido a esta característica do olho humano, radiações luminosas de cores diversas podem ser obtidas por combinações destas três *cores primárias*, vermelho, verde e azul.

A Figura 1.10 apresenta a sensibilidade relativa dos cones às radiações eletromagnéticas.



**Figura 1.10 Sensibilidade relativa dos três tipos de cones presentes na retina. Fonte: The image processing handbook / John C. Russ. - 3rd Ed., 1998**

O olho humano é capaz de perceber muitas outras cores além das cores espectrais puras. A cor púrpura, por exemplo, é uma combinação de vermelho e azul. No entanto, qualquer radiação monocromática capaz de ativar simultaneamente os cones azuis e vermelhos ativará os cones verdes com maior intensidade, de forma que a sensação predominante será de cor verde.

## 1.7 SISTEMAS DE CORES

A cor de um objeto que emite radiação na faixa visível é definida pela soma das cores espectrais emitidas. O processo de formação de cores, neste caso, é aditivo. Dado o mecanismo de formação de cores do olho humano, o processo aditivo pode ser entendido como uma combinação de radiações monocromáticas nas faixas verde, vermelho e azul, em proporções variáveis. Por este motivo, estas cores são denominadas *cores primárias* da luz. Este processo de geração de cores caracteriza um *sistema RGB* (*Red, Green, e Blue*). A Comissão Internacional de Iluminação (CIE) estabeleceu os seguintes valores para os comprimentos de onda das primárias da luz:

Azul: 435,8 nm  
 Verde: 546,1 nm  
 Vermelho: 700 nm

As cores primárias combinadas duas a duas em igual intensidade produzem as **cores secundárias** da luz, magenta, ciano e amarelo:

Magenta = Vermelho + Azul  
 Ciano = Azul + Verde  
 Amarelo = Verde + Vermelho

Vê-se, então, que o processo de formação de cores para objetos que emitem radiação é aditivo. Define-se como *cor oposta* a uma dada cor secundária como a cor primária que não entra em sua composição. Assim, o verde é oposto ao magenta, o vermelho é oposto ao ciano, e o azul é oposto ao amarelo.



A cor branca é gerada pela combinação balanceada de radiações nas faixas do vermelho, do verde e do azul. Pode-se dizer também que a cor branca é uma combinação de qualquer cor secundária com sua oposta.

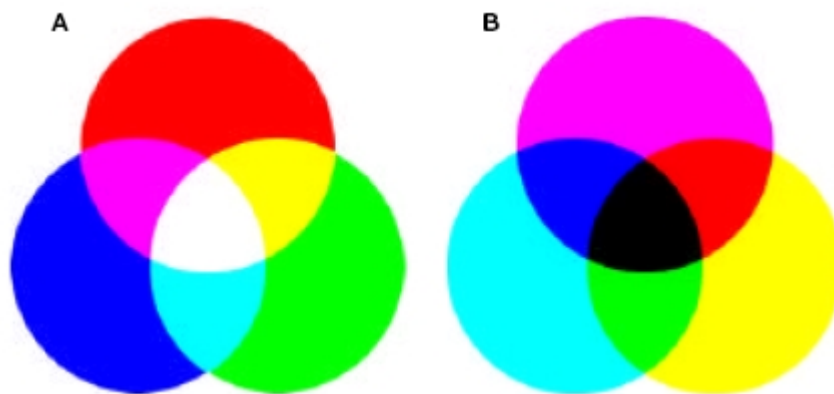
A cor de um objeto que não emite radiação própria, por outro lado, é definida pelos seus pigmentos, que absorvem radiações em determinadas faixas de comprimento de onda e refletem outras. Novamente, devido à natureza do olho humano, o processo de formação de cores pode ser entendido como a absorção ou reflexão, em proporções variáveis, das componentes verde, vermelho e azul da radiação incidente. Um pigmento ciano, por exemplo, reflete as componentes azul e verde da luz incidente e absorve ou subtrai desta a componente vermelha. Em outros termos, o ciano pode ser visto como o resultado da subtração do vermelho da cor branca. O processo de formação de cores pigmentares é, portanto, subtrativo. Tem-se agora:

$$\begin{aligned}\text{Magenta} &= \text{Branco} - \text{Verde} = \text{Vermelho} + \text{Azul} \\ \text{Ciano} &= \text{Branco} - \text{Vermelho} = \text{Azul} + \text{Verde} \\ \text{Amarelo} &= \text{Branco} - \text{Azul} = \text{Verde} + \text{Vermelho} \\ \text{Amarelo} + \text{Magenta} + \text{Ciano} &= \text{Branco} - \text{Azul} - \text{Vermelho} - \text{Verde} = \text{Preto}\end{aligned}$$

As cores primárias dos pigmentos são definidas como aquelas que absorvem uma cor primária da luz e refletem as outras duas. Assim, as cores primárias dos pigmentos são as secundárias da luz, ou seja, magenta, ciano, e amarelo.

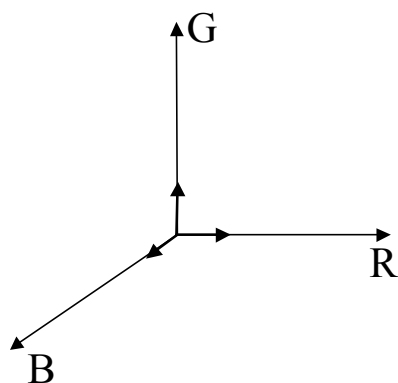
As cores pigmentares primárias combinadas duas a duas produzem as cores secundárias dos pigmentos. Combinando-se magenta, que não reflete a componente verde da luz incidente, com ciano, que não reflete a componente vermelha, tem-se que cor resultante é azul, a única componente refletida tanto pelo pigmento magenta quanto pelo pigmento ciano. Por um raciocínio análogo, magenta e amarelo produzem vermelho, enquanto que ciano e amarelo produzem verde. Vê-se então que as cores secundárias dos pigmentos são as primárias da luz, e as secundárias. Finalmente, a mistura balanceada de pigmentos magenta, ciano e amarelo absorve todas as componentes da luz incidente, produzindo, portanto, preto.

Os processos aditivo e subtrativo de formação de cores são ilustrados na Figura 1.11.



**Figura 1.11. Processos de formação de cores: (A) aditivo e (B) subtrativo**

Um monitor de vídeo forma imagens por emissão de radiação luminosa, em um processo envolvendo combinação de verde, vermelho e azul. Na prática, é necessário limitar a intensidade máxima admissível para as componentes de cor, de forma que um subconjunto do sistema RGB foi adotado. Qualquer cor neste sistema pode ser representada por suas componentes de vermelho, verde e azul. Assim, uma cor no sistema RGB é um vetor no espaço tridimensional apresentado na Figura 1.12, e pode ser representada por suas coordenadas (r, g, b).



**Figura 1.12 O espaço de cores RGB**

No espaço RGB, a reta definida pelo conjunto de pontos  $(i, i, i)$  é chamada de reta acromática. Pontos sobre a reta acromática representam tonalidades de cinza, ou *níveis de cinza*, variando continuamente do preto ao branco, à medida que percorremos a reta a partir da origem. O ponto  $(0, 0, 0)$  representa o preto (ausência de luz). Um ponto  $(M, M, M)$ , onde  $M$  é a intensidade máxima que pode assumir qualquer componente de cor, representa o branco.

As impressoras coloridas empregam o sistema CMY (Ciano, Magenta e Amarelo). Raramente os pigmentos adotados e as proporções nas misturas são perfeitos, de forma que a combinação balanceada das três cores pode resultar em algo próximo ao marrom, quando o desejado seria o preto. Por este motivo, muitas vezes acrescenta-se o preto como 4ª cor, e o sistema é dito CMYK, com o K final representado a cor preta (*black*).

## 1.8 OUTROS SISTEMAS DE CORES

O sistema YIQ é utilizado em transmissão de TV em cores pelo padrão NTSC (*National Television System Committee*), mantendo compatibilidade com os padrões de TV monocromática (P & B). No sistema YIQ, o Y representa a componente de luminância (intensidade percebida, ou brilho), enquanto que o I (*intermodulation*) e o Q (*quadrature*) representam as componentes de cromaticidade (cor). A conversão entre os sistemas RGB e YIQ é dada por:

$$Y = 0.299R + 0.587G + 0.114B$$

$$I = 0.596R - 0.274G - 0.322B$$

$$Q = 0.211R - 0.523G + 0.312B$$

$$R = 1.000 Y + 0.956 I + 0.621 Q$$

$$G = 1.000 Y - 0.272 I - 0.647 Q$$

$$B = 1.000 Y - 1.106 I + 1.703 Q$$

O sistema YUV é atualmente utilizado em transmissão de TV em cores pelos padrões PAL e NTSC, dentre outros, e também preserva compatibilidade com os padrões de TV P & B. No sistema YUV, o Y representa a componente de luminância (intensidade percebida, ou brilho), enquanto que as componentes U e V representam a cromaticidade. A conversão do sistema RGB para YUV é dada por:

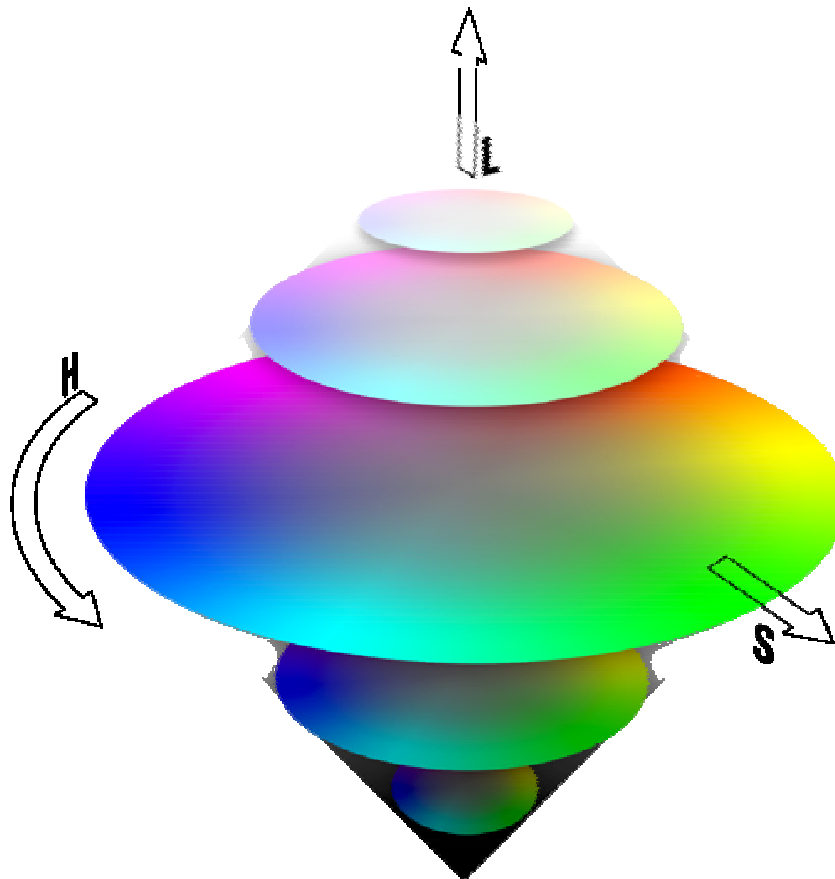
$$Y = 0.299R + 0.587G + 0.114B$$

$$U = -0.147R - 0.289G + 0.436B$$

$$V = 0.615R - 0.515G - 0.100B$$

Apesar da retina humana operar fisicamente de acordo com o sistema RGB, o ser humano não percebe as cores como uma combinação de vermelho, verde e azul. Os atributos perceptivos mais comumente utilizados na caracterização das cores são a intensidade, matiz (*hue*) ou tonalidade, e saturação. A matiz de uma cor é determinada pelo comprimento de onda dominante da cor, ou seja, é a cor espectral mais próxima a ela, de acordo com a percepção de um observador. Quando dizemos que um objeto é verde ou amarelo, estamos especificando a matiz. A saturação expressa o grau de pureza relativa da cor, com relação à adição de branco. As cores espectrais puras são completamente saturadas. Quanto mais branco houver em uma cor, menos saturada é ela.

No sistema HSI (*hue, saturation, intensity*), a matiz é especificada como um ângulo, com 0° representando a matiz vermelha, 120° representando verde e 240° representando azul; a saturação é um valor real entre 0 e 1, e a intensidade é um valor não negativo. Por reproduzir fielmente a maneira como o ser humano percebe cores, o sistema HSI é muito usado por artistas e em problemas de visão computacional. Uma representação esquemática do espaço de cores HSI pode ser vista na Figura 1.13.



**Figura 1.13 O sistema de cores HSI**

A conversão entre os sistemas HSI e RGB é efetuada conforme especificado nos seguintes pseudo-códigos [Fonte: adaptado de EasyRGB - <http://www.easyrgb.com>]:

De RGB para HSI:

```
var_R = (R / 255)           // Valores RGB entre 0 e 255
var_G = (G / 255)
var_B = (B / 255)
```

```

var_Min = min(var_R, var_G, var_B)           // Valor mínimo das componentes RGB
var_Max = max(var_R, var_G, var_B)           // Valor máximo das componentes RGB
del_Max = var_Max - var_Min

L = (var_Max + var_Min) / 2

if (del_Max == 0)                             // Nível de cinza, sem componentes cromáticas
{
    H = 0
    S = 0
}
else                                           // Componentes cromáticas
{
    if (L < 0.5) S = del_Max / (var_Max + var_Min)
    else      S = del_Max / ( 2 - var_Max - var_Min )

    del_R = (((var_Max - var_R) / 6) + (del_Max / 2)) / del_Max
    del_G = (((var_Max - var_G) / 6) + (del_Max / 2)) / del_Max
    del_B = (((var_Max - var_B) / 6) + (del_Max / 2)) / del_Max

    if (var_R == var_Max) H = del_B - del_G
    else if (var_G == var_Max) H = (1 / 3) + del_R - del_B
    else if (var_B == var_Max) H = (2 / 3) + del_G - del_R

    if ( H < 0 ) ; H += 1
    if ( H > 1 ) ; H -= 1
}

```

De HSI para RGB:

```

if (S == 0)
{
    R = L * 255
    G = L * 255
    B = L * 255
}
else
{
    if (L < 0.5) var_2 = L * (1 + S)
    else      var_2 = (L + S) - (S * L)

    var_1 = 2 * L - var_2

    R = 255 * Hue_2_RGB(var_1, var_2, H + (1 / 3))
    G = 255 * Hue_2_RGB(var_1, var_2, H)
    B = 255 * Hue_2_RGB(var_1, var_2, H - (1 / 3))
}

Hue_2_RGB(v1, v2, vH)           // Função Hue_2_RGB
{
    if (vH < 0) vH += 1
    if (vH > 1) vH -= 1
    if ((6 * vH) < 1) return (v1 + (v2 - v1) * 6 * vH)
    if ((2 * vH) < 1) return (v2)
    if ((3 * vH) < 2) return (v1 + (v2 - v1) * ((2 / 3) - vH) * 6)
    return (v1)
}

```

# Capítulo 2

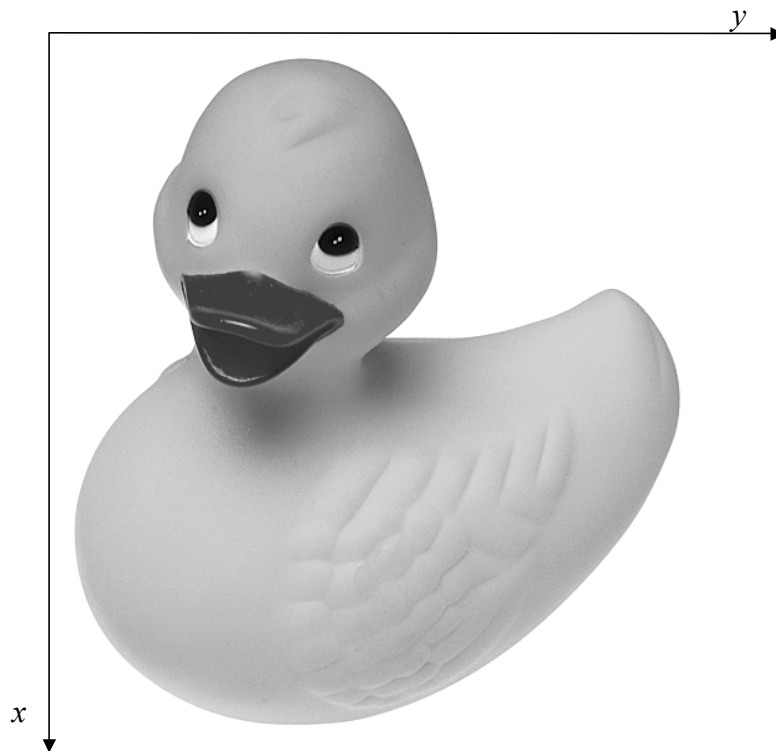
## Um Modelo de Imagem Digital

---

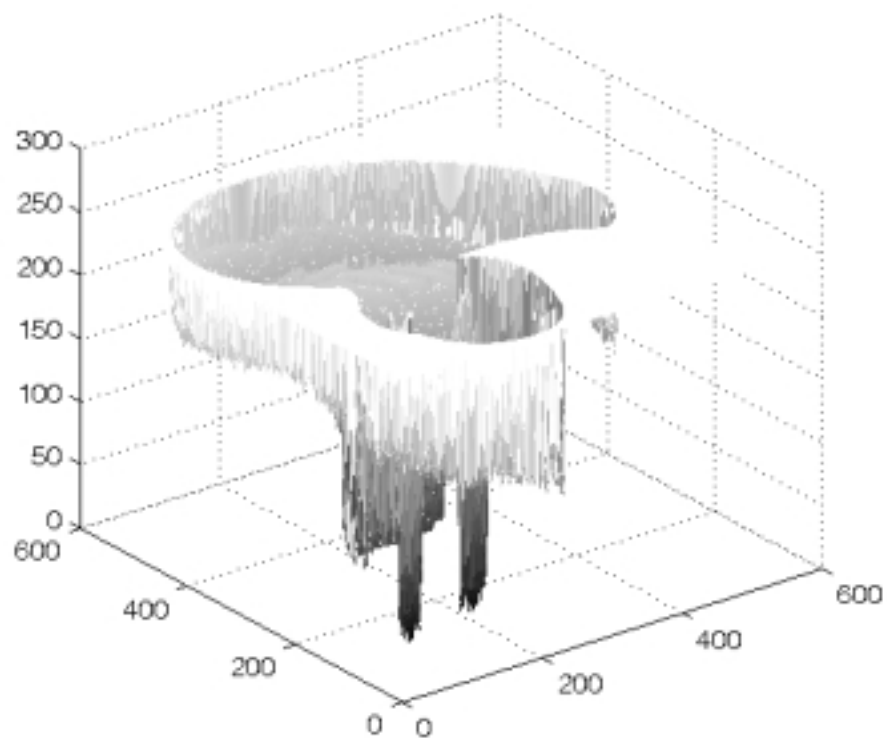
### 2.1 IMAGENS MONOCROMÁTICAS

Uma imagem monocromática é uma função de duas variáveis  $f_a(x,y)$ , onde as variáveis independentes  $x$  e  $y$  são em geral coordenadas espaciais e o valor de  $f_a(x,y)$  define a intensidade luminosa (ou brilho, ou *nível de cinza*) da imagem em  $(x,y)$ . A Figura 2.1 mostra a convenção aqui adotada para o posicionamento dos eixos nas imagens.

Para se enfatizar que uma imagem é essencialmente uma função de duas variáveis, é conveniente empregar um sistema de três eixos perpendiculares  $x, y, z$ , com  $z = f_a(x,y)$ , como na



**Figura 2.1 Convenção adotada para os eixos. [Imagem ducky.bmp – Fonte: Adobe ® Photoshop ® 7.0]**



**Figura 2.2 A imagem da Figura 2.1 vista como uma função de duas variáveis**

A geração das imagens é afetada por dois componentes: a iluminação,  $i(x,y)$ , que representa a quantidade de luz incidente sobre o ponto  $(x,y)$  e a reflectância,  $r(x,y)$ , que representa a quantidade de luz refletida pelo ponto  $(x,y)$ . A multiplicação de  $i(x,y)$  e  $r(x,y)$  formam  $f_a(x,y)$ :

$$f_a(x,y) = i(x,y)r(x,y)$$

com

$$0 < i(x,y) < \infty$$

e

$$0 < r(x,y) < 1$$

Destaque-se que  $i(x,y)$  é definida pela fonte de iluminação, enquanto que  $r(x,y)$  é uma característica dos objetos da cena. A Tabela 2.1 mostra os valores de iluminação sob diferentes condições ambientais e a Tabela 2.2 apresenta a reflectância associada a diferentes objetos.

**Tabela 2.1 Iluminação sob diferentes condições ambientais**

<b>Condição</b>	<b><math>i(x,y)</math> (foot-candles)</b>
Dia de sol	9000
Dia nublado	1000
Escritório comercial	100
Lua cheia	0,01

**Tabela 2.2 Reflectância de diferentes objetos**

<b>Material</b>	<b><math>r(x,y)</math></b>
Neve	0,93
Lâmina de prata	0,90
Parede Branca	0,80
Aço Inoxidável	0,65
Veludo preto	0,01

## 2.2 AMOSTRAGEM E QUANTIZAÇÃO

Para ser utilizada em um computador uma imagem deve ser digitalizada, ou seja, discretizada tanto espacialmente quanto em intensidade, por intermédio dos processos de amostragem e quantização descritos no Capítulo 1 para sinais unidimensionais. No caso de imagens, pode-se estender facilmente o processo, colhendo-se amostras a intervalos regulares na horizontal e na vertical. Se  $R$  amostras são colhidas na vertical e  $C$  na horizontal, obtem-se uma matriz  $R \times C$ . Para imagens, normalmente, faz-se  $Q = \{0, 1, \dots, L-1\}$ , de forma que a precisão da quantização é especificada pelo número de níveis de quantização  $L$ , e não diretamente por  $q$ .

A matriz resultante do processo de discretização espacial e de brilho representa uma *imagem digital monocromática*, e cada elemento  $f(i, j)$  é conhecido por *pixel* (contração de *picture element*).

$$\begin{bmatrix} f(0,0) & f(0,1) & \dots & f(0,C-1) \\ f(1,0) & f(1,1) & \dots & f(1,C-1) \\ \dots & \dots & \dots & \dots \\ f(R-1,0) & f(R-1,1) & \dots & f(R-1,C-1) \end{bmatrix}$$

Uma imagem monocromática apresenta unicamente gradações de cinzas, com pixels mais claros representando valores mais elevados; pixels pretos têm valor zero, enquanto que pixels brancos têm valor  $L-1$ . A mostra a imagem monocromática de dimensões  $R = 399$ ,  $C = 533$ , com  $L = 256$ , e a representação matricial correspondente.



(a)

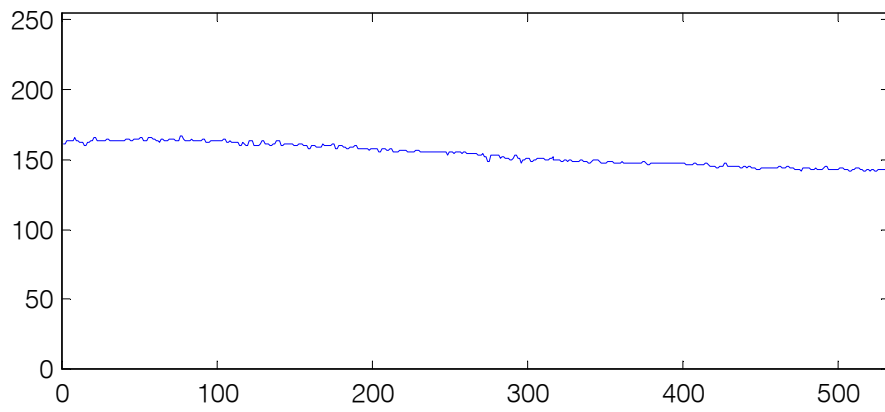
$$\begin{bmatrix} 161 & 161 & \dots & 142 \\ 161 & 161 & \dots & 142 \\ \dots & \dots & \dots & \dots \\ 163 & 163 & \dots & 95 \end{bmatrix}$$

(b)

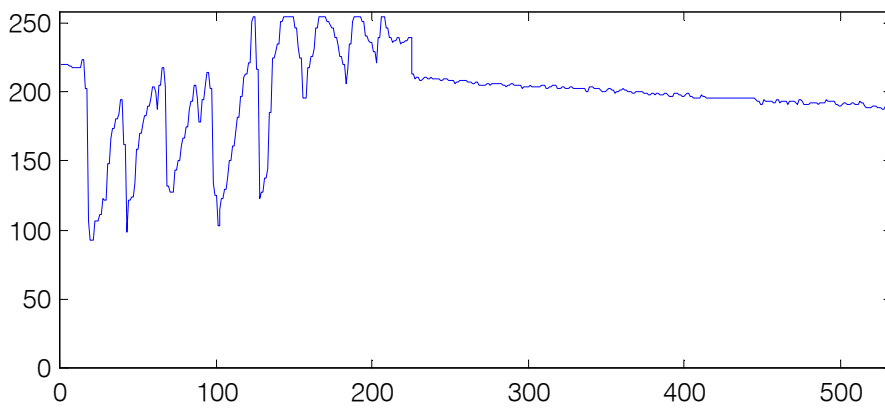
**Figura 2.3 Uma imagem digital monocromática: (a) representação gráfica usual; (b) representação matricial**

O enfoque matricial permite considerar uma imagem digital de dimensões  $R \times C$  como uma coleção de  $R$  sinais unidimensionais com  $C$  amostras cada, dados pelas linhas da matriz ou, alternativamente, como uma coleção de  $C$  sinais unidimensionais com  $R$  amostras cada, dados pelas colunas da matriz. Esta visão traz diversas implicações importantes, notadamente na construção de filtros e transformadas separáveis, que serão vistos em capítulos posteriores, conduzindo a implementações computacionalmente mais eficientes. A Figura 2.4 apresenta os sinais unidimensionais que constituem a primeira linha ( $i = 0$ ), uma linha intermediária ( $i = 200$ ) e a uma coluna intermediária ( $j = 266$ ) da imagem da Figura 2.3 (a).

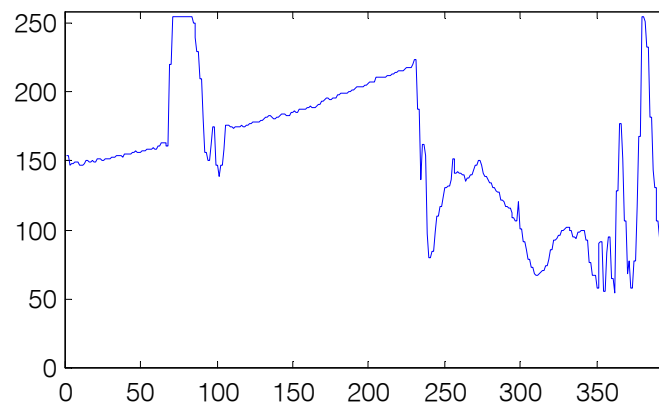




(a)



(b)



(c)

**Figura 2.4 Sinais unidimensionais extraídos da imagem da Figura 2.3: (a) linha  $i = 0$ ; (b) linha  $i = 200$ ; (c) coluna  $j = 266$ .**

Devido à eliminação de informação inerente à amostragem e à quantização, uma imagem digital é uma aproximação da imagem analógica a partir da qual ela foi obtida. A qualidade da aproximação depende essencialmente dos valores de  $R$ ,  $C$  e  $L$ . Em geral, quantos maiores estes parâmetros, melhor a qualidade da digitalização. É comum restringir-se o parâmetro  $L$  a potências de dois:

$$L = 2^l$$

com  $l$  inteiro. O número de bits necessário para representar uma imagem digital é, neste caso

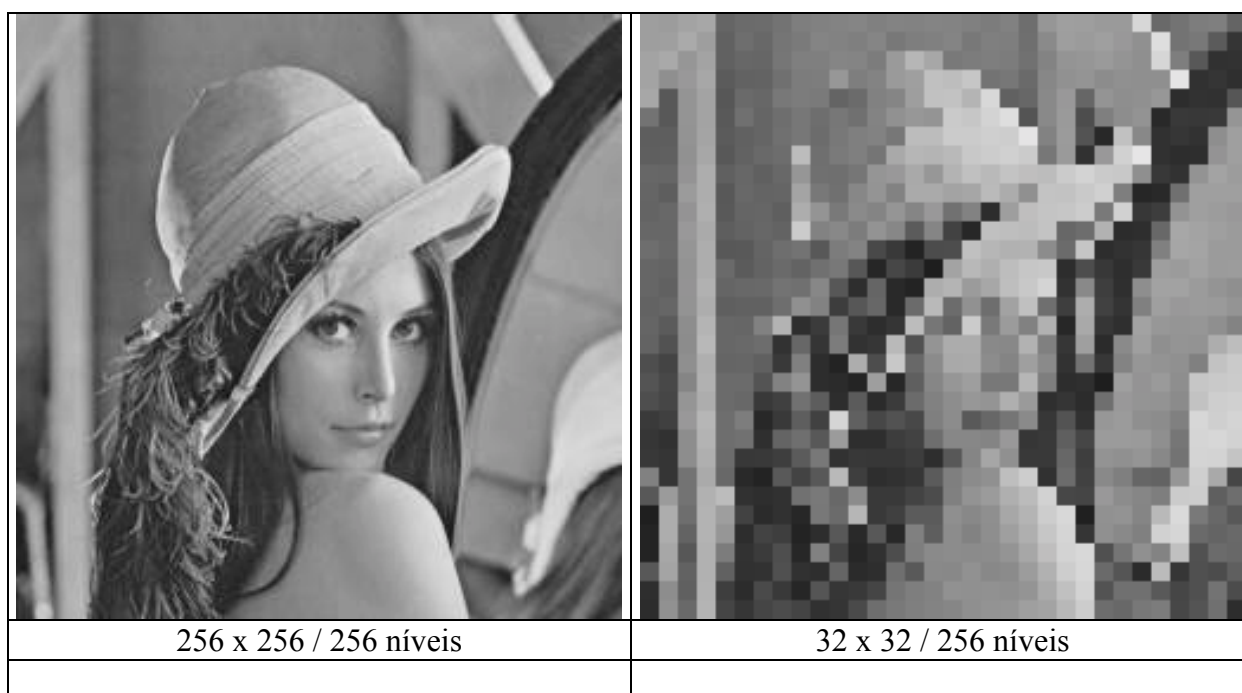
$$b = R \times C \times I$$

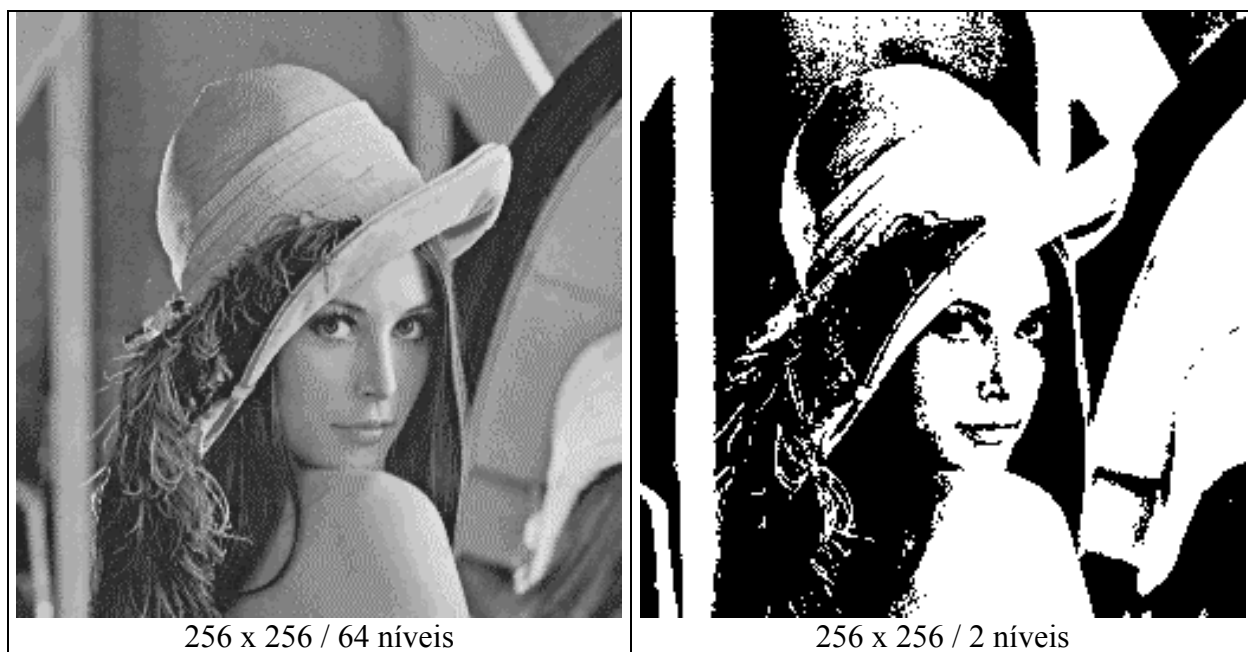
Aumentar  $R$ ,  $C$  e  $I$  melhora a qualidade da imagem, mas aumenta o número de bits requerido para a codificação binária da imagem, o que pode ocasionar problemas de armazenamento e transmissão. A Tabela 2.3 mostra o número de bytes de uma imagem digital monocromática para diversos valores de  $R$ ,  $C$  e  $L$ .

**Tabela 2.3 Número de Bytes em Uma Imagem Digital Monocromática**

R = C	L					
	2	64	256	32K	64K	16M
32	128	768	1024	1920	2048	3072
256	8192	49152	65536	122880	131072	196608
512	32768	196608	262144	491520	524288	786432
1024	131072	786432	1048576	1966080	2097152	3145728

O número de amostras e níveis de cinza para obtenção de uma imagem digital de qualidade adequada depende das características da imagem (quão complexa ela é, qual o seu tamanho, etc) e da aplicação que se pretende dar a ela. A Figura 2.5 mostra o efeito dos parâmetros de digitalização na qualidade visual de uma imagem monocromática.





**Figura 2.5** Qualidade de uma imagem monocromática para alguns valores de  $R$ ,  $C$  e  $L$

### 2.3 IMAGENS COLORIDAS

Em uma imagem digital colorida no sistema RGB, pode-se considerar um pixel como um vetor  $\mathbf{f}(i,j) = (f_r(x,y), f_g(x,y), f_b(x,y))$ , cujas componentes representam as intensidades de vermelho, verde e azul da cor. Pode-se considerar, portanto, que uma imagem colorida é a composição de três imagens monocromáticas,  $f_r(x,y)$ ,  $f_g(x,y)$ ,  $f_b(x,y)$ , denominadas, respectivamente, de *banda vermelha* (ou *banda R*), *banda verde* (ou *banda G*), e *banda azul* (ou *banda B*) da imagem colorida, conforme ilustra a Figura 2.6. Esta figura pretende destacar que os valores dos pixels nas bandas R, G e B representam de fato níveis de vermelho, verde e azul, respectivamente.

Para cada banda, valem os mesmos conceitos apresentados para a imagem digital monocromática. A amostragem é quase sempre efetuada de forma idêntica em todas as bandas.

O número de cores que um pixel pode assumir em uma imagem RGB com  $L_r$  níveis de quantização na banda R,  $L_g$  na banda G e  $L_b$  na banda B é  $L_r \times L_g \times L_b$ . Se

$$\begin{aligned} l_r &= \log_2(L_r) \\ l_g &= \log_2(L_g) \\ l_b &= \log_2(L_b), \end{aligned}$$

o número de bits por pixel necessários para representar as cores é  $l_r + l_g + l_b$ , e o número de bits necessário para representar uma imagem digital de dimensões  $R \times C$  é

$$b = R \times C \times (l_r + l_g + l_b)$$

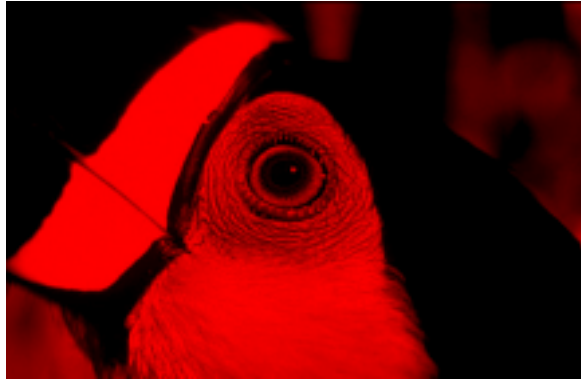
Exemplificando, se  $L_r = L_g = L_b = 2^8 = 256$ , ou seja, se há 256 níveis possíveis em cada banda, a imagem colorida pode apresentar até 16.777.216 cores diferentes. Neste caso, cada pixel será representado por  $3 \times 8 = 24$  bits. A Tabela 2.4 mostra o número de bytes necessários para representar uma imagem RGB, para diversos valores dos parâmetros de digitalização.

**Tabela 2.4 Número de bytes de uma imagem RGB**

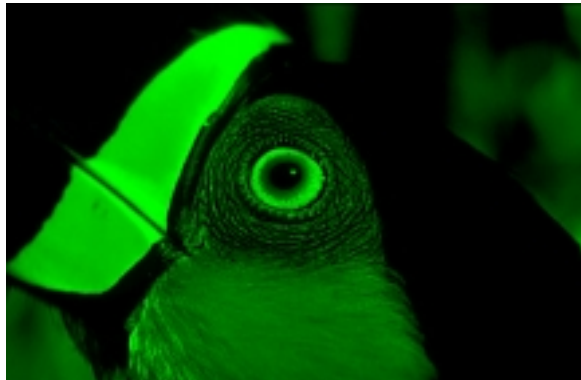
$R = C$	$L_r = L_g = L_b$		
	2	64	256
32	384	2304	3072
256	24576	147456	196608
512	98304	589824	786432
1024	393216	2359296	3145728

## 2.4 OPERAÇÕES ENVOLVENDO IMAGENS

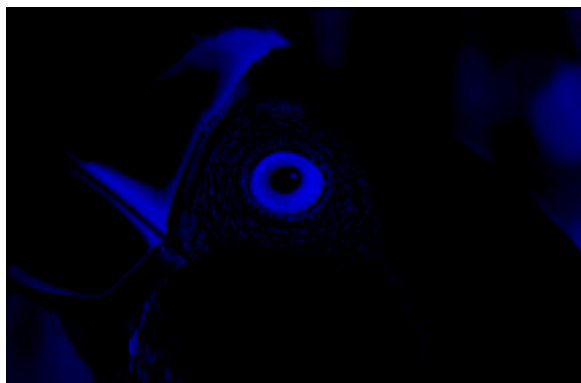
Em geral, as técnicas de processamento de imagens podem ser aplicadas no domínio do espaço ou no domínio da frequência. Nas operações no domínio do espaço trabalha-se diretamente com os pixels da imagem. O Capítulo 4, o Capítulo 5 e o Capítulo 6 tratam do domínio do espaço. Uma outra abordagem consiste em aplicar uma operação matemática genericamente denominada de transformada sobre a imagem a ser processada, convertendo-a do domínio do espaço para o domínio da frequência, e processar a imagem neste domínio. O Capítulo 7 trata de técnicas de processamento no domínio da frequência.



Banda R



Banda G



Banda B



Imagem RGB

**Figura 2.6** Uma imagem RGB e sua bandas R, G e B [Fonte (Imagem RGB): Araquém Alcântara, “Tucano-bico-preto”, <http://www.araquem.com.br/>]

Pode-se representar uma operação  $T$  sobre  $n$  imagens,  $f_1, f_2, \dots, f_n$ , produzindo uma imagem de saída  $g$ , como

$$g = T[f_1, f_2, \dots, f_n]$$

Em geral, a equação acima descreve o que se conhece como operação  $n$ -ária, com a imagem de saída sendo produzida a partir de uma ou mais imagens de entrada. Quando  $n = 2$ , tem-se uma operação *binária* e, quando  $n = 1$ , uma operação *unária*, onde uma única imagem de entrada é processada produzindo uma imagem de saída. As operações unárias são também chamadas de *filtros*. As operações de filtragem no domínio do espaço são abordadas no Capítulo 4, enquanto que as operações  $n$ -árias, com  $n > 1$ , são estudadas no Capítulo 5.

# Capítulo 3

## Dispositivos Gráficos

---

### 3.1 IMPRESSORAS

Nas impressoras coloridas modernas, cada ponto é constituído por uma combinação ponderada dos três pigmentos primários (CMY), com ou sem a adição de preto. A *resolução* de uma impressora proporciona uma medida do grau de detalhes que esta pode apresentar. Quanto maior a resolução, menor o ponto e, conseqüentemente, maior a qualidade da saída da impressora.

A resolução das impressoras é normalmente especificada em pontos por polegada (1 polegada = 2,54 cm), ou dpi (*dots per inch*). Uma impressora com resolução de 720 x 720 dpi pode imprimir 720 pontos por polegada na horizontal e 720 pontos por polegada na vertical.

### 3.2 MONITORES DE VÍDEO E APARELHOS DE TV

Muitos monitores de vídeo e aparelhos de TV possuem um dispositivo denominado *tubo de raios catódicos* (CRT). Um CRT consiste em um canhão que dispara feixes de elétrons em direção a uma tela revestida com pontos de fósforo. Bobinas defletoras produzem campos elétricos que direcionam os feixes de forma a fazê-los atingir pontos específicos da tela.

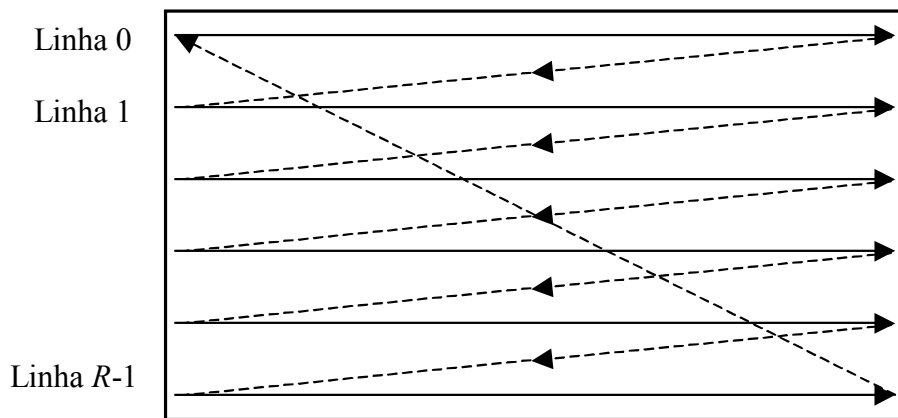
A proporção entre largura e a altura de uma tela de monitor de vídeo ou aparelho de TV convencional é de 3:4 (vertical:horizontal). Se uma diagonal da tela mede  $p$  polegadas, tem-se um monitor ou aparelho de  $p$  polegadas. Um monitor de 14 polegadas, por exemplo, tem uma largura  $w = 11,2$  e uma altura  $h = 8,4$  polegadas.

Nos monitores de vídeo de varredura fixa, muito semelhantes aos aparelhos de TV, feixes eletrônicos varrem a tela, linha a linha, da esquerda para a direita e de cima para baixo. Ao serem atingidos pelos elétrons, os pontos de fósforo que revestem a parede interna da tela absorvem a energia do feixe. Parte da energia absorvida é progressivamente liberada na faixa do espectro visível, de forma que os pontos brilham por um período que caracteriza a chamada *persistência* do fósforo. No caso de monitores de vídeo, a persistência do fósforo utilizado é baixa, de forma que os pontos atingidos pelo feixe brilham apenas por um breve período.

Nos monitores coloridos, a tela é revestida por agrupamentos de três tipos de fósforo, que emitem luz vermelha, verde ou azul. Cada um destes agrupamentos é denominado *triade*. Os pontos de fósforo em uma triade se encontram tão próximos entre si que parecem ao olho humano um único ponto colorido. Um canhão emite três feixes de elétrons, cada um deles direcionado para um ponto específico da triade. A intensidade de cada feixe é independentemente controlada, o que permite definir a intensidade da radiação emitida por cada ponto da triade e, conseqüentemente, a cor resultante.

O tamanho das triades é denominado *dot pitch*. Um *dot pitch* pequeno significa pontos bem definidos, ou seja, boa qualidade visual. De forma análoga às imagens digitais, a tela é uma matriz bidimensional de pixels, e cada pixel pode ser composto por várias triades. Uma tela com resolução de  $C \times R$  é capaz de apresentar  $C$  pixels na horizontal e  $R$  na vertical.

A figura seguinte mostra esquematicamente o trajeto dos feixes de elétrons.



Quando o feixe termina a varredura de todas as linhas da tela, diz-se que foi formado um quadro. A *freqüência de sincronismo horizontal (HSync)* é o número de linhas varridas por segundo durante a produção de um quadro, enquanto que a *freqüência de sincronismo vertical (VSync)* é número de quadros gerados por segundo.

Seja  $R$  o número de linhas em um quadro;  $t_r$  o tempo decorrido desde o início da varredura de uma linha até o início da varredura da próxima linha do mesmo quadro;  $t_v$  o tempo de retraço vertical; e  $t_q = R.t_r + t_v$  o tempo decorrido entre o início da varredura de um quadro e o início da varredura do próximo quadro.  $HSync$  e  $VSync$  são dadas por:

$$HSync = \frac{1}{t_r}$$

$$VSync = \frac{1}{t_q} = \frac{1}{R.t_r + t_v}$$

Para que a imagem na tela não apresente um efeito desagradável de cintilação, mais conhecido como *flicker*, é necessário que  $VSync$  seja igual ou superior a 60 quadros/s, ou seja, a duração de um quadro não deve ultrapassar 1/60 s. Assim, para não haver *flicker*, temos:



$$t_q \leq \frac{1}{60} \Leftrightarrow R.t_r + t_v \leq \frac{1}{60}$$

O tempo de retraço vertical pode ser expresso como um múltiplo de  $t_r$ , ou seja,  $t_v = n.t_r$ , e a equação anterior pode ser reescrita como:

$$(R + n)t_r \leq \frac{1}{60} \Leftrightarrow t_r \leq \frac{1}{60(R + n)} \Leftrightarrow HSync \geq 60(R + n)$$

Assim,  $HSync_{min} = 60(R + n)$  é o menor valor de  $HSync$  necessário para não ocorrer *flicker*. Em alguns modelos de monitor, na resolução de 640 x 480 o retraço vertical dura aproximadamente  $45t_r$ . Neste caso,  $HSync_{min} = 31500$  linhas/s = 31,5 KHz. Na resolução de 800 x 600,  $t_v \approx 30t_r$ , resultando em  $HSync_{min} = 37,8$  KHz. Na resolução de 1024 x 768,  $t_v \approx 20t_r$ , e  $HSync_{min} = 47,3$  KHz.

O modelo Samsung SyncMaster 3, um dos monitores de vídeo mais populares da última década, possui  $HSync = 35,5$  KHz e operam com uma resolução máxima de 800 x 600, o que resultaria em  $VSync = 56$  quadros/s, com a ocorrência de *flicker*.

Um paliativo muito comum nestes casos é o uso de *entrelaçamento*. Na formação de um quadro com entrelaçamento, o feixe eletrônico percorre primeiramente apenas as linhas ímpares e, no quadro seguinte, apenas as pares; no próximo, novamente as ímpares, e assim por diante. Assim, no modo entrelaçado, cada quadro possui a metade de linhas de um quadro no modo não-entrelaçado, o que permite dobrar a frequência vertical, minimizando o *flicker* ao preço de uma menor definição de detalhes nas imagens.

O Samsung SyncMaster 3 opera no modo entrelaçado para a resolução de 800 x 600, apresentando assim uma  $VSync = 112$  quadros/seg.

O VESA definiu uma  $VSync$  mínima de aproximadamente 70 Hz para evitar *flicker* e também o efeito estroboscópico de 60 Hz, causado pela cintilação da tela de vídeo na mesma frequência que a cintilação das lâmpadas ligadas à rede elétrica.

### 3.3 MEMÓRIA DE VÍDEO

A imagem a ser exibida na tela de um monitor deve ser armazenada na chamada *memória de vídeo*, localizada na placa controladora de vídeo. A memória de vídeo é repetidamente lida pela controladora, que utiliza as informações ali encontradas para ajustar a intensidade dos feixes eletrônicos do CRT, regenerando adequadamente a imagem na tela.

A cada pixel da tela corresponde uma locação na memória de vídeo, cujo conteúdo pode ser interpretado de duas maneiras. No modo direto, cada locação contém a informação sobre a intensidade das componentes RGB do pixel correspondente. Exemplificando, um byte na memória de vídeo pode reservar dois bits para controlar a intensidade da componente azul, três bits para a componente verde e três para a componente vermelha do pixel a ela associado.

Alternativamente, cada locação na memória de vídeo pode conter um índice para uma linha de uma *tabela de cores*, também localizada na placa controladora de vídeo e manipulável por *software*. Neste caso, a linha indexada contém informação sobre as componentes RGB do pixel associado àquela locação na tabela de vídeo. Uma *paleta de cores* é um preenchimento específico de uma tabela de cores em um dado momento.

Como ilustração, suponhamos que a cada pixel corresponda quatro bits na memória de vídeo. No modo direto, há 16 cores possíveis fixas, que podem ser obtidas alocando-se dois bits para o verde, um para o vermelho e um para o azul, por exemplo.

O controlador leria o dado correspondente a próxima posição do feixe na tela, e passaria os bits de controle de cada componente de cor a três conversores digital/analógico (DAC), que gerariam as tensões a serem aplicadas aos canhões de elétrons, determinando assim as intensidades das componentes.

No modo de tabela de cores, os quatro bits permitiriam indexar 16 linhas. Com 24 bits por linha, oito bits poderiam representar cada componente de cor. Duas possíveis paletas são apresentadas na Tabela 3.1 e na Tabela 3.2. A paleta da Tabela 3.2 contém 16 níveis de cinza diferentes, incluído o preto e o branco. Vê-se que a tabela de cores permite uma maior flexibilidade. Uma tabela de cores com 16 linhas contendo 24 bits/linha para especificação das componentes R, G e B das cores permite a seleção de 16 cores quaisquer de um total de  $256 \times 256 \times 256 = 16,7$  milhões de cores. Há um total de  $256^{16 \times 3} = 256^{48} = 2^{384}$  paletas diferentes neste caso, o que permite ampla flexibilidade na seleção de paletas adequadas para imagens específicas.

Com a tabela de cores da Tabela 3.2 são necessários três DACs de oito bits. O controlador lê o dado da memória de vídeo, usa-o para indexar uma linha da tabela de cores e envia os valores de cada componente, obtidos nesta linha, para cada um dos DACs.

Sistemas de exibição que operam com até 256 cores simultâneas normalmente empregam paletas. Acima disso, o modo direto é mais comum. Como a cada pixel correspondem  $n$  bits na memória de vídeo, com  $n = \log_2(\text{número máximo de cores simultâneas})$ , quanto maiores a resolução da tela e o número máximo de cores simultâneas, maior a memória de vídeo requerida, conforme se vê na Tabela 3.3.

**Tabela 3.1 Uma paleta com 16 cores**

Índice	R	G	B
0000	0	0	0
0001	0	0	255
0010	0	84	0
0011	0	84	255
0100	0	169	0
0101	0	169	255
0110	0	255	0
0111	0	255	255
1000	255	0	0
1001	255	0	255
1010	255	84	0
1011	255	84	255
1100	255	169	0
1101	255	169	255
1110	255	255	0
1111	255	255	255

**Tabela 3.2 Uma paleta com 16 níveis de cinza**

Índice	R	G	B
0000	0	0	0
0001	16	16	16
0010	33	33	33
0011	50	50	50
0100	67	67	67
0101	84	84	84
0110	101	101	101
0111	118	118	118
1000	135	135	135
1001	152	152	152
1010	169	169	169
1011	186	186	186
1100	203	203	203
1101	220	220	220
1110	237	237	237
1111	255	255	255

**Tabela 3.3 Tamanho da memória de vídeo para diversos modos de vídeo**

Resolução	Nº máximo de cores simultâneas	Memória de vídeo (em bytes)
640 x 480	16	150 K
640 x 480	256	300 K
640 x 480	16M	750K
800 x 600	256	468,75 K
800 x 600	64K	937,5 K
800 x 600	16M	1,37 M
1024 x 768	256	768 K
1024 x 768	64K	1,5 M
1024 x 768	16M	2,25 M

# Capítulo 4

## Filtros Espaciais

---

Os operadores *unários*, também denominados *filtros*, processam uma única imagem de entrada produzindo uma imagem de saída. Um filtro pode ser representado de forma simplificada como uma operação  $T$  que mapeia uma imagem de entrada  $f$  em uma imagem de saída  $g$ :

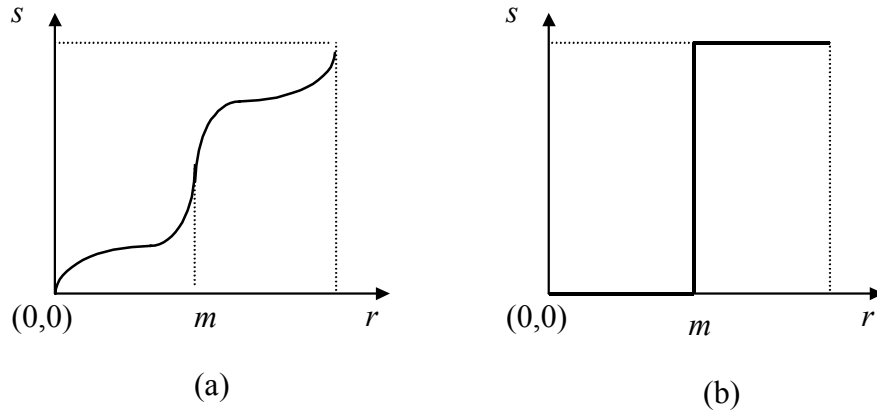
$$g = T[f]$$

### 4.1 OPERAÇÕES PONTUAIS E LOCAIS

Quanto ao escopo de atuação, filtros espaciais podem ser locais ou pontuais. Em uma *operação unária pontual*,  $g(i, j)$  depende do valor do pixel na locação  $(i, j)$  em cada uma das imagens de entrada. Por simplicidade de notação, representaremos um filtro pontual por

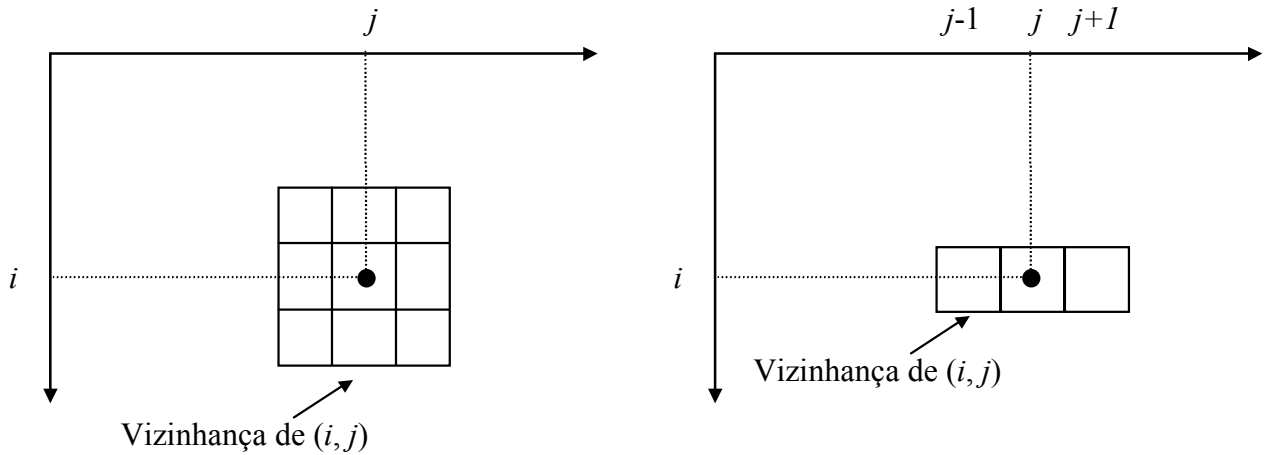
$$s = T(r)$$

onde  $r$  e  $s$  denotam, respectivamente, o nível de cinza de  $f$  e  $g$  em um ponto  $(i, j)$  qualquer. A Figura 1.1 ilustra o filtro de controle de contraste. Na parte (a) da Figura 4.1,  $T(s)$  tende a produzir uma imagem com contraste maior que a original, escurecendo os níveis abaixo de  $m$  e aumentando o brilho dos níveis acima de  $m$ . Em um caso extremo da transformação de aumento de contraste, apresentado na parte (b) da Figura 4.1,  $T(r)$  produz uma imagem com apenas dois níveis de cinza.



**Figura 4.1 Representação gráfica de um filtro pontual: (a) aumento de contraste e (b) caso extremo de aumento de contraste**

Em uma *operação unária local*,  $g(i, j)$  depende dos valores dos pixels das imagens de entrada em uma *vizinhança* de  $(i, j)$ . Comumente, a vizinhança de  $(i, j)$  é definida como sendo uma área retangular ou quadrada centrada em  $(i, j)$ , como indica a Figura 4.2.



**Figura 4.2 Duas possíveis vizinhanças do pixel nas coordenadas  $(i, j)$**

Um exemplo de um filtro local utilizando a vizinhança quadrada da Figura 4.2 consiste em definir  $g(i, j)$  como a media dos pixels da imagem de entrada  $f$  contidos na vizinhança:

$$g(i, j) = \frac{1}{9} [f(i-1, j-1) + f(i-1, j) + f(i-1, j+1) + \\ + f(i, j-1) + f(i, j) + f(i, j+1) + \\ + f(i+1, j-1) + f(i+1, j) + f(i+1, j+1)]$$

Se a operação for repetida para todas as coordenadas  $(i, j)$  da imagem de entrada, será obtida uma nova imagem  $g(i, j)$  onde cada pixel será a média dos pixels da imagem de entrada na vizinhança especificada. Há, no entanto, um problema para calcular  $g(i, j)$  na primeira e na última coluna, uma vez que a imagem de entrada não está definida para  $j = -1$  e  $j = C$ . Há várias maneiras de contornar esse problema; pode-se, por exemplo, redefinir a operação na primeira coluna como a média entre  $f(i, j)$  e  $f(i, j+1)$  e, na última coluna, como a média entre  $f(i, j-1)$  e  $f(i, j)$ .

As seções seguintes apresentam vários exemplos de filtros espaciais. Por simplicidade de notação, os conceitos serão apresentados predominantemente para imagens monocromáticas. Para imagens coloridas, dependendo dos objetivos específicos, há duas opções muitas empregadas:

1. Aplicação do filtro de maneira independente em cada banda da imagem. Esta modalidade pode causar fortes alterações de matizes, uma vez que as informações de cromaticidade são afetadas.
2. Aplicação do filtro apenas na componente de luminância. Pode-se, por exemplo, transformar a imagem original de RGB para YIQ, aplicar o filtro sobre a banda Y, e converter as bandas Y, I e Q da imagem filtrada para RGB. Esta opção não causa alteração de matizes.

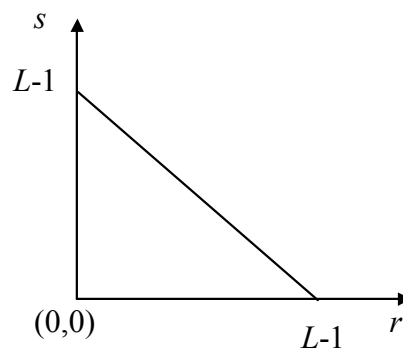
## 4.2 NEGATIVO

A obtenção do negativo de uma imagem digital tem várias aplicações. Por exemplo, se o negativo de uma imagem na tela é fotografado utilizando uma câmera fotográfica com filme convencional, o negativo deste filme, gerado no processo de revelação em laboratório, será uma imagem positiva, que poderá ser usada como *slide*. É também comum que o negativo de uma imagem constitua uma forma de visualização mais conveniente de determinados tipos de imagens.

O negativo de uma imagem digital é obtido por intermédio da operação pontual

$$s = T(r) = L - 1 - r$$

A representação gráfica desta operação pode ser vista na Figura 4.3



**Figura 4.3 Representação gráfica da operação de negativo**

Pode-se observar o resultado da aplicação do filtro negativo sobre uma imagem monocromática na Figura 4.4. A Figura 4.5 mostra os resultados da aplicação do filtro negativo seguindo as opções citadas no final da Seção 4.1. Como se pode notar na Figura 4.5 (b), a aplicação do negativo nas bandas R, G e B da imagem introduz alterações intensas de matizes, com a substituição das cores por suas complementares (verde/magenta, azul/amarelo, vermelho/ciano, branco/preto). Por outro lado, a aplicação do negativo exclusivamente na componente de luminância Y, com a conversão posterior para o sistema RGB original — Figura 4.5 (c) — preserva perfeitamente as matizes originais.



(a)



(b)

**Figura 4.4 Filtro negativo: (a) Imagem original [Fonte: Sebastião Salgado, exposição fotográfica “Terra – Movimento dos Sem-Terra”, *The New York Times On the Web*]; (b) negativo da imagem.**



(a)



(b)



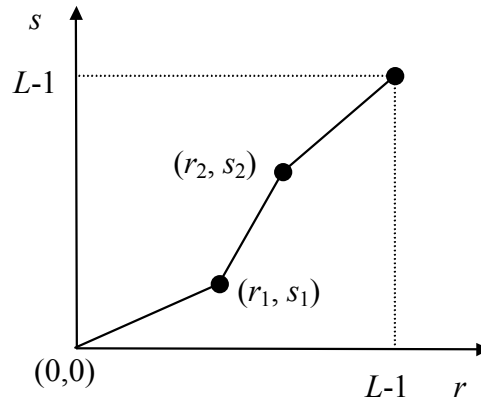
(c)

**Figura 4.5 Filtro negativo: (a) imagem original [Fonte: Denise Greco, “Lorikeet”, galeria de fotos do *site* <http://wwwbr.kodak.com>]; (b) negativo aplicado sobre as bandas R, G e B; (c) negativo aplicado sobre a luminância Y.**



### 4.3 CONTROLE DE CONTRASTE

Imagens de baixo contraste podem resultar de iluminação deficiente, abertura insuficiente do diafragma da câmera, tempo de exposição demasiadamente curto, problemas eletrônicos no equipamento de digitalização, etc. Um contraste reduzido dificulta o discernimento dos objetos presentes na cena. O objetivo do aumento de contraste é aumentar a faixa de níveis de cinza utilizada. Em algumas situações, no entanto, pode ser esteticamente interessante reduzir o contraste em uma cena. A operação de controle de contraste pode ser definida de muitas maneiras. Uma forma simples está graficamente representada na Figura 4.6.



**Figura 4.6 Operação de controle de contraste**

Se  $r_1 = s_1$  e  $r_2 = s_2$ , não há nenhuma alteração nos níveis de cinza. Se  $r_1 = (L-1)/2$ ,  $s_1 = 0$ ,  $r_2 = L/2$ , e  $s_2 = L-1$ , a operação produz uma imagem de dois níveis, 0 e  $L-1$ , com os níveis de entrada menores do que  $L/2$  sendo mapeados para 0 e os demais sendo mapeados para  $L-1$ . Se a declividade do primeiro e do último segmento de reta é menor que um e a do segmento central maior que um, tem-se aumento do contraste. Por outro lado, se a declividade do primeiro e do último segmento de reta é maior que um e a do segmento central menor que um, tem-se redução do contraste.

O filtro de *aumento de contraste adaptativo* é especificado por:

$$g(i, j) = \begin{cases} \mu(i, j) + \frac{c}{\sigma(i, j)}[f(i, j) - \mu(i, j)]; & \sigma(i, j) \neq 0 \\ f(i, j); & \sigma(i, j) = 0 \end{cases}$$

onde  $\mu(i, j)$  é a média em uma vizinhança  $n \times n$  de  $(i, j)$ ;  $\sigma(i, j)$  é o desvio padrão em uma vizinhança  $n \times n$  de  $(i, j)$ ; e  $c$  é um parâmetro não-negativo que controla a intensidade do aumento de contraste. A Figura 4.7 apresenta os resultados obtidos pelo filtro de aumento de contraste adaptativo. Note-se que, se o valor de  $c/\sigma(i, j)$  for reduzido, o termo  $\mu(i, j)$  pode se tornar dominante, e o filtro assume um comportamento semelhante ao do filtro *média*, descrito na Seção 4.5, produzindo um efeito de suavização das fronteiras entre os objetos presentes na imagem. Este efeito pode ser observado na Figura 4.7 (b).



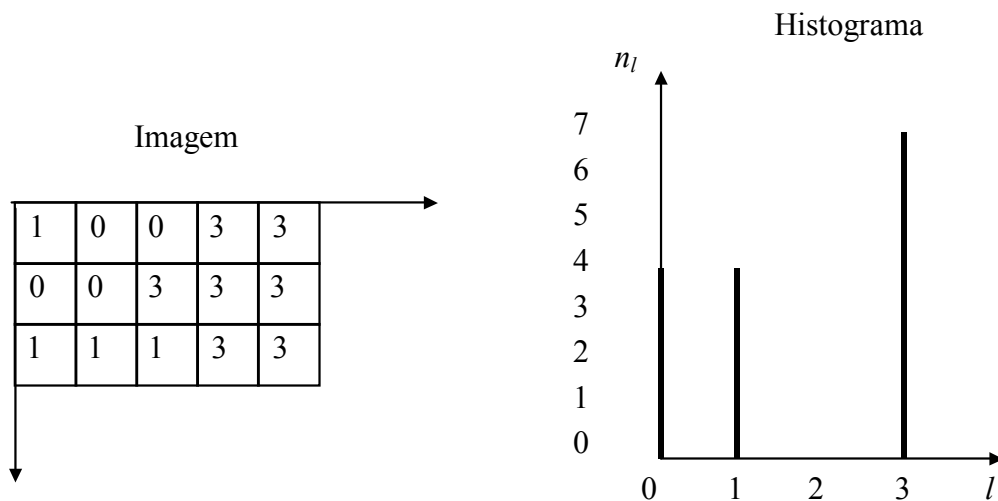
**Figura 4.7 Filtro de contraste adaptativo:** (a) imagem original [Fonte: Vistex Database (detalhe da imagem food.0000.pgm)]; contraste adaptativo com (b)  $c = 10$  e  $n = 5$ ; (c)  $c = 60$  e  $n = 3$ ; (d)  $c = 60$  e  $n = 5$ .

#### 4.4 EQUALIZAÇÃO E EXPANSÃO DE HISTOGRAMA

Se o nível de cinza  $l$  ocorre  $n_l$  vezes em uma imagem com  $n$  pixels, podemos estimar a probabilidade de ocorrência do nível  $l$  na imagem por intermédio de sua frequência relativa:

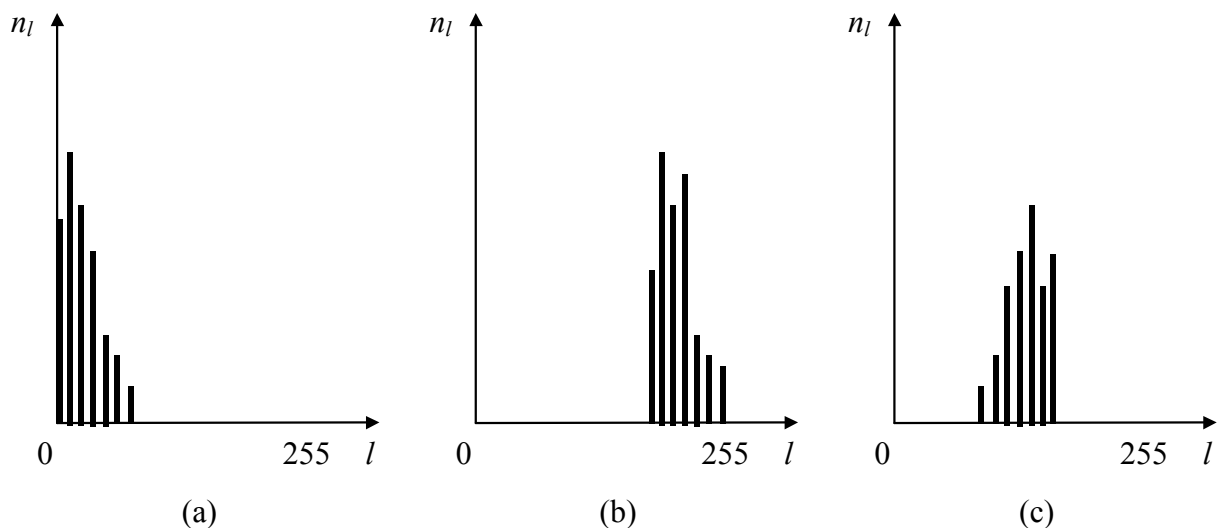
$$P(l) = \frac{n_l}{n}$$

O histograma da imagem é uma representação gráfica de  $n_l$  ou de  $P(l)$  em função de  $l$ , conforme ilustra a Figura 4.8.



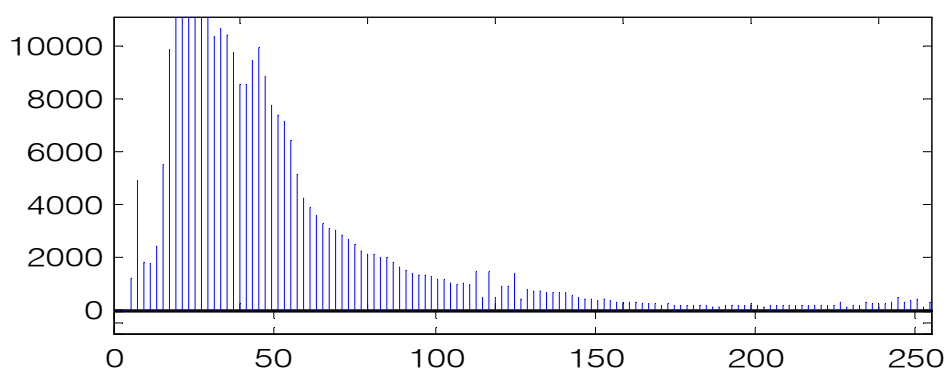
**Figura 4.8** Uma imagem 3 x 5 ( $L = 4$ ) e seu histograma

A análise do histograma permite tecer considerações a respeito da distribuição de níveis de cinza de uma imagem. A Figura 4.9 apresenta histogramas típicos de imagens escuras, claras, e de níveis intermediários.

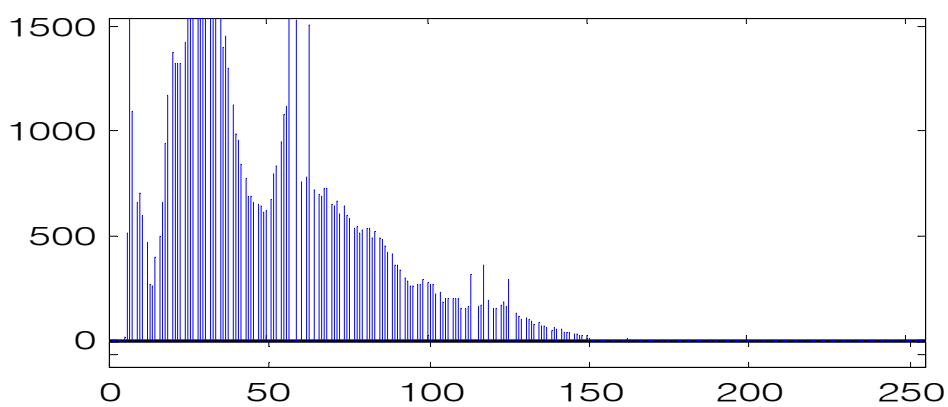


**Figura 4.9** Histogramas: (a) imagens escuras; (b) imagens claras; c) imagens de brilho intermediário

A Figura 4.10 apresenta uma imagem predominantemente escura contendo uma região de alta luminosidade, bem como o histograma correspondente. A Figura 4.11 apresenta uma subimagem extraída de uma região particularmente escura da imagem da Figura 4.10, e o histograma resultante, onde se pode observar a ausência de níveis de cinza elevados.



**Figura 4.10** Imagem “Jacare” e o histograma correspondente

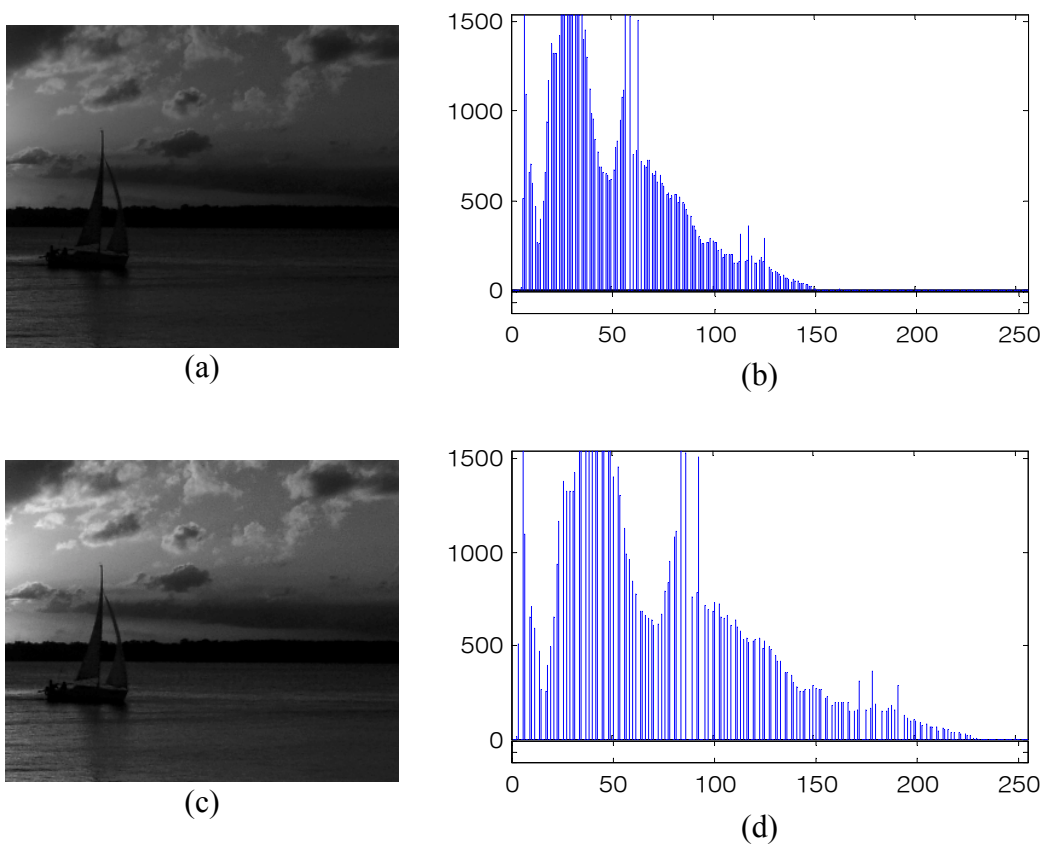


**Figura 4.11** Imagem extraída da imagem “Jacare” e o histograma correspondente

Nos casos em que uma faixa reduzida de níveis de cinza é utilizada, como na imagem da Figura 4.11, a técnica de *expansão de histograma* pode produzir uma imagem mais rica em detalhes. A expansão de histograma consiste na transformação pontual:

$$s = T(r) = \text{round}\left(\frac{r - r_{\min}}{r_{\max} - r_{\min}}(L - 1)\right)$$

onde  $r_{\min}$  e  $r_{\max}$  representam, respectivamente, o menor e o maior nível de cinza presentes na imagem de entrada, com  $r_{\min} < r_{\max}$ ,  $L$  é o limite superior da escala de cinzas e *round* significa arredondamento para o inteiro mais próximo. A operação atribui valor zero aos pixels com nível de cinza  $r_{\min}$  e valor  $L-1$  aos pixels com nível de cinza  $r_{\max}$ , e distribui proporcionalmente os níveis intermediários entre 0 e  $L-1$ . Assim, essa operação leva à utilização de toda a escala de cinzas disponível. A Figura 4.12 apresenta o efeito produzido pela operação.



**Figura 4.12 Expansão de histograma: (a) imagem original; (b) histograma da imagem original; (c) imagem resultante; (d) histograma da imagem resultante.**

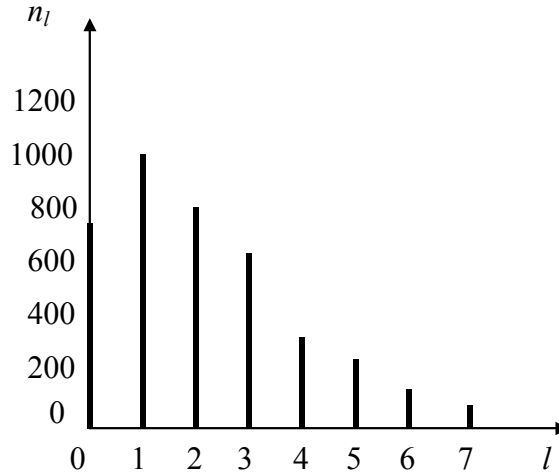
Quando a imagem de entrada apresenta tanto pixels de valor 0 quanto  $L-1$  (ou valores próximos a esses extremos) a expansão de histograma é ineficaz (ou praticamente ineficaz). Nestas situações, se a imagem de entrada apresenta uma distribuição pobre de níveis de cinza, a técnica de *equalização de histograma* pode produzir bons resultados. O objetivo da equalização de histograma é gerar uma imagem com uma distribuição de níveis de cinza uniforme. A técnica é útil para realçar diferenças sutis entre níveis de cinza próximos e leva, em muitos casos, a um aumento substancial no nível de detalhes perceptíveis.

A operação de equalização de histograma de uma imagem de dimensões  $R \times C$  é proporcionada pela transformação:

$$s = T(r) = \text{round}\left(\frac{L-1}{RC} \sum_{l=0}^r n_l\right)$$

Como exemplo, consideremos uma imagem 64 x 64, com  $L = 8$ , com o histograma apresentado na Figura 4.13.

$l$	$n_l$
0	790
1	1023
2	850
3	656
4	329
5	245
6	122
7	81

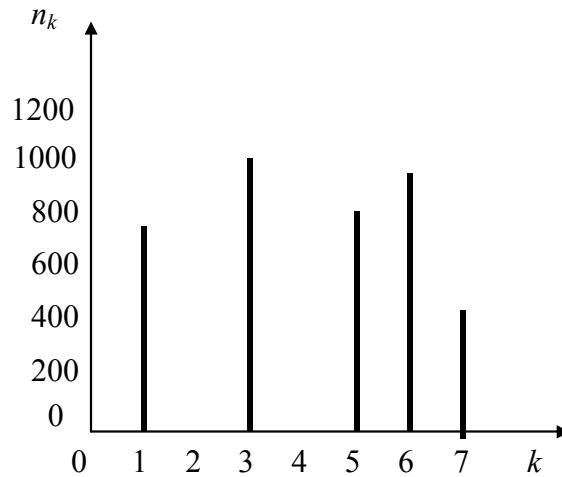


**Figura 4.13** O histograma de uma imagem 64 x 64,  $L = 8$

A equalização de histograma produz:

$$\begin{aligned}
 r = 0 &\rightarrow s = \text{round}(790 \times 7 / 4096) = 1 \\
 r = 1 &\rightarrow s = \text{round}(1813 \times 7 / 4096) = 3 \\
 r = 2 &\rightarrow s = \text{round}(2663 \times 7 / 4096) = 5 \\
 r = 3 &\rightarrow s = \text{round}(3319 \times 7 / 4096) = 6 \\
 r = 4 &\rightarrow s = \text{round}(3648 \times 7 / 4096) = 6 \\
 r = 5 &\rightarrow s = \text{round}(3893 \times 7 / 4096) = 7 \\
 r = 6 &\rightarrow s = \text{round}(4015 \times 7 / 4096) = 7 \\
 r = 7 &\rightarrow s = \text{round}(4096 \times 7 / 4096) = 7
 \end{aligned}$$

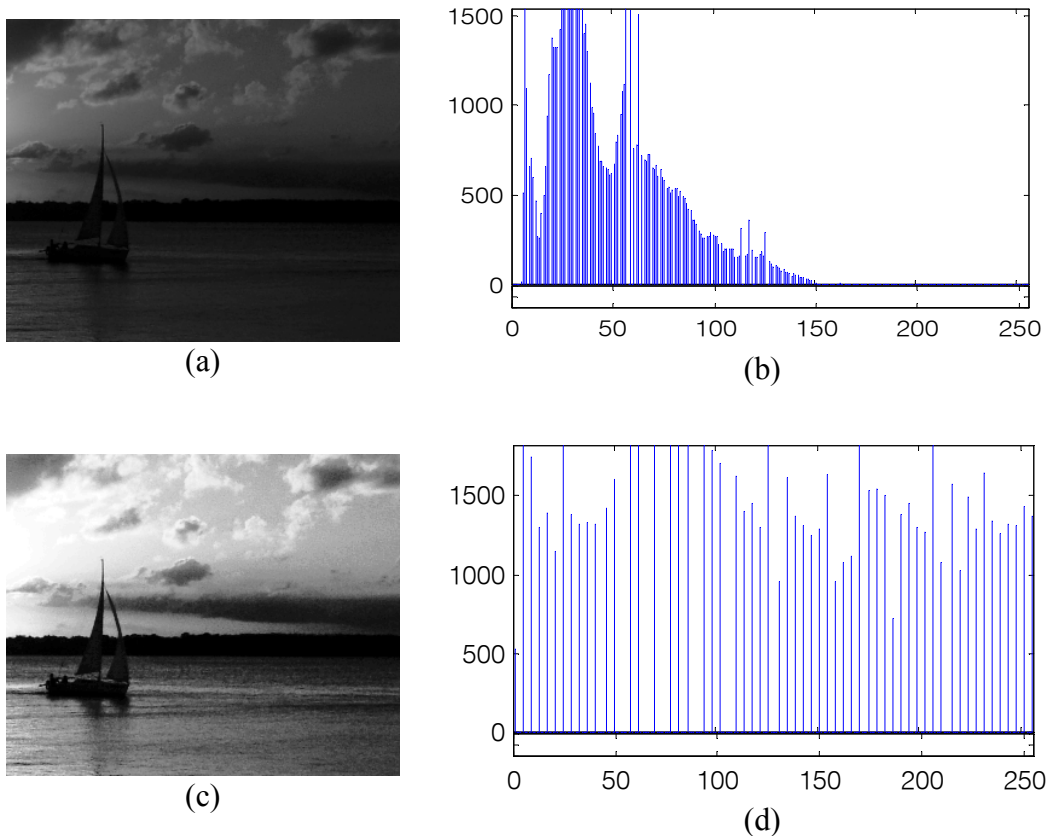
Vê-se que o nível 0 é transformado em 1; o 1 torna-se 3; o 2 torna-se 5; os níveis 3 e 4 são transformados em 6 (que conterà, portanto, 656 + 329 = 985 pixels); e o 5, o 6 e o 7 são transformados em 7 (que conterà 448 pixels). O histograma equalizado é apresentado na Figura 4.14.



**Figura 4.14 Resultado da equalização do histograma da Figura 4.13**

Observe-se que a os níveis de cinza não ocorrem todos com a mesma frequência na imagem resultante, o que caracterizaria uma equalização perfeita, mas o histograma original, característico de uma imagem preponderantemente escura, foi transformado em um histograma característico de uma imagem com níveis bem distribuídos.

A imagem resultante da equalização de histograma sempre terá pixels de valor  $L-1$ . No entanto, como ilustra a Figura 4.14, o valor mínimo presente na imagem resultante pode ser maior que zero. Por este motivo, para garantir que toda a escala de cinzas disponível será utilizada, é comum efetuar-se expansão de histograma após equalização de histograma. A Figura 4.15 ilustra o efeito obtido pela operação de equalização seguida por expansão de histograma.



**Figura 4.15 Equalização seguida de expansão de histograma: (a) imagem original; (b) histograma da imagem original; (c) imagem resultante; (d) histograma da imagem resultante.**



## 4.5 FILTROS DE SUAVIZAÇÃO

Os filtros *média*, *mediana* e *moda* produzem uma suavização da imagem, reduzindo variações bruscas de níveis de cinza entre pixels adjacentes, e podem ser usados para redução de ruídos ou interferências de origens diversas.

No filtro *média* o valor do pixel  $g(i, j)$  é a média dos valores dos pixels de  $f$  em uma vizinhança de  $(i, j)$  contendo  $n$  pixels. A suavização produzida é proporcional a  $n$ , e o filtro apresenta o inconveniente de reduzir fortemente a definição das bordas dos objetos na imagem. A operação pode ser efetuada por intermédio de uma convolução de  $f$  com uma máscara convolucional apropriada. Para uma vizinhança de dimensões  $3 \times 3$ , por exemplo, o núcleo do filtro *média* seria:

$$h = \frac{1}{9} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

No filtro *mediana*, o valor do pixel  $g(i, j)$  é a mediana dos valores dos pixels de  $f$  em uma vizinhança de  $(i, j)$  contendo  $n$  pixels. A *mediana* de um conjunto de  $n$  pixels ordenados por valor é o valor do pixel na posição central da lista ordenada, se  $n$  for ímpar, ou a média dos valores dos dois pixels nas posições centrais, se  $n$  for par de valores. Como exemplo, consideremos uma vizinhança  $3 \times 3$  com os seguintes valores, já ordenados: (10, 15, 20, 20, 20, 20, 20, 25, 100). A mediana é o valor central, ou seja, o valor do quinto elemento da lista (20). Consideremos agora uma vizinhança  $2 \times 2$  com valores (20, 20, 25, 100). A mediana é a média entre 20 e 25, ou seja, 22,5.

O filtro *mediana* também produz uma suavização proporcional a  $n$ , mas com uma melhor preservação das bordas dos objetos na imagem quando comparado com o filtro da *média*.

No filtro *moda*, o valor do pixel  $g(i, j)$  é a moda dos valores dos pixels de  $f$  em uma vizinhança de  $(i, j)$  contendo  $n$  pixels. A *moda* de um conjunto de valores é o valor mais freqüente do conjunto. Para o filtro da *moda*, se houver dois ou mais valores igualmente freqüentes, pode-se definir como valor de  $g(i, j)$  a média ou mediana destes valores.

A Figura 4.16 ilustra a eficácia da aplicação dos filtros *média*, *moda* e *mediana* sobre uma imagem contaminada por ruído aditivo uniforme. Destaque-se o resultado obtido pelo filtro *mediana* com vizinhança  $3 \times 3$ , eliminando praticamente todo o ruído e, ao mesmo tempo, preservando os detalhes visualmente mais importantes da imagem original.

## 4.6 FILTROS DE AGUÇAMENTO

Os filtros de aguçamento produzem efeito contrário ao dos filtros de suavização, acentuando variações de níveis de cinza entre pixels adjacentes. Os filtros de aguçamento são geralmente baseados no conceito de gradiente de funções bidimensionais. O *vetor gradiente* de uma função  $f$  contínua em  $(x, y)$  é assim definido:

$$\mathbf{G}[f(x, y)] = \begin{bmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{bmatrix}$$

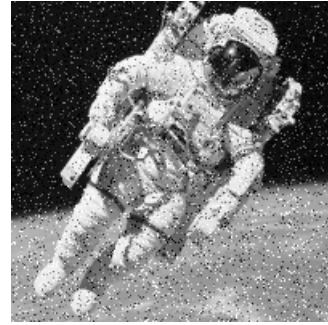


O vetor gradiente  $\mathbf{G}[f(x, y)]$  aponta no sentido da maior taxa de variação de  $f(x, y)$ , e seu módulo  $G[f(x, y)]$ , que representa a taxa de variação de  $f(x, y)$  por unidade de distância no sentido de  $\mathbf{G}$ , é dado por

$$G[f(x, y)] = \left[ \left( \frac{\partial f}{\partial x} \right)^2 + \left( \frac{\partial f}{\partial y} \right)^2 \right]^{1/2}$$



(a)



(b)



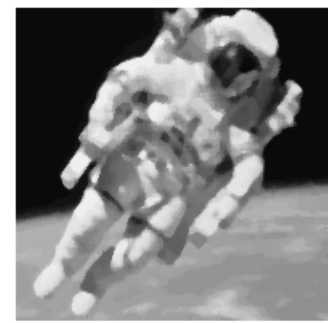
(c)



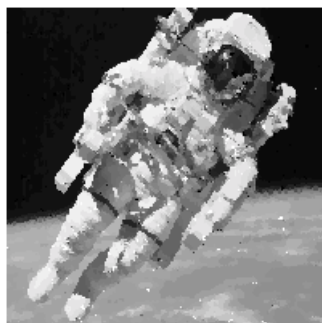
(d)



(e)



(f)



(g)



(h)

**Figura 4.16 Filtros média, mediana e moda com vizinhança 3 x 3 e 5 x 5: (a) imagem original; (b) imagem com ruído; imagens resultantes após aplicação sobre a imagem ruidosa dos filtros (c) média, 3 x 3; (d) média 5 x 5; (e) mediana 3 x 3; (f) mediana 5 x 5; (g) moda 3 x 3; (h) moda 5 x 5.**

Nos filtros de gradiente, o valor do pixel  $g(i, j)$  da imagem de saída é uma aproximação discreta do módulo do vetor gradiente nas coordenadas  $(i, j)$  da imagem  $f$  de entrada. Algumas aproximações usuais incluem:

$$g(i, j) = \{[f(i, j) - f(i+1, j)]^2 + [f(i, j) - f(i, j+1)]^2\}^{1/2}$$

$$g(i, j) = |f(i, j) - f(i+1, j)| + |f(i, j) - f(i, j+1)|$$

Gradiente de Roberts:

$$g(i, j) = \{[f(i, j) - f(i+1, j+1)]^2 + [f(i+1, j) - f(i, j+1)]^2\}^{1/2}$$

$$g(i, j) = |f(i, j) - f(i+1, j+1)| + |f(i+1, j) - f(i, j+1)|$$

Gradiente de Prewitt:

$$g(i, j) = |f(i+1, j-1) + f(i+1, j) + f(i+1, j+1) - f(i-1, j-1) - f(i-1, j) - f(i-1, j+1)| +$$

$$|f(i-1, j+1) + f(i, j+1) + f(i+1, j+1) - f(i-1, j-1) - f(i, j-1) - f(i+1, j-1)|$$

Gradiente de Sobel:

$$g(i, j) = |f(i+1, j-1) + 2f(i+1, j) + f(i+1, j+1) - f(i-1, j-1) - 2f(i-1, j) - f(i-1, j+1)| +$$

$$|f(i-1, j+1) + f(i, j+1) + f(i+1, j+1) - f(i-1, j-1) - 2f(i, j-1) - f(i+1, j-1)|$$

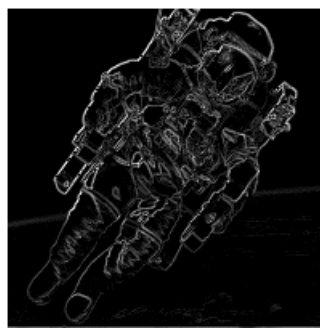
Em todas as aproximações, há uma relação direta entre o valor do filtro gradiente e a diferença entre pixels próximos. Assim, o valor resultante é maior nas transições mais acentuadas da imagem e menor em regiões mais uniformes, sendo zero em regiões de intensidade constante.

É muito comum efetuar uma *binarização* na imagem resultante da filtragem por gradiente, definindo-se um limiar e mapeando-se para  $L$  todos os pixels com módulo de gradiente maior ou igual ao limiar, e para 0 os demais pixels.

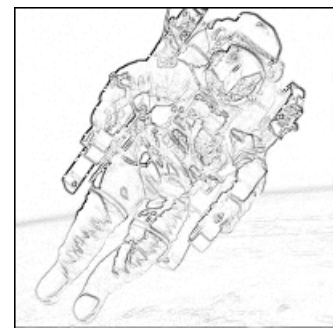
Na Figura mostra o efeito da aplicação do Gradiente de Roberts, com e sem binarização.



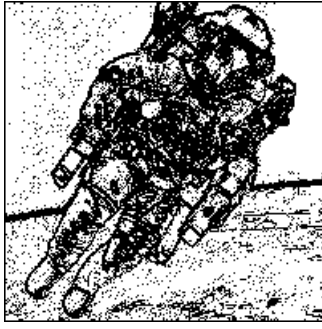
(a)



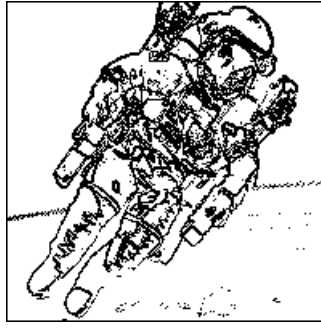
(b)



(c)



(d)



(e)



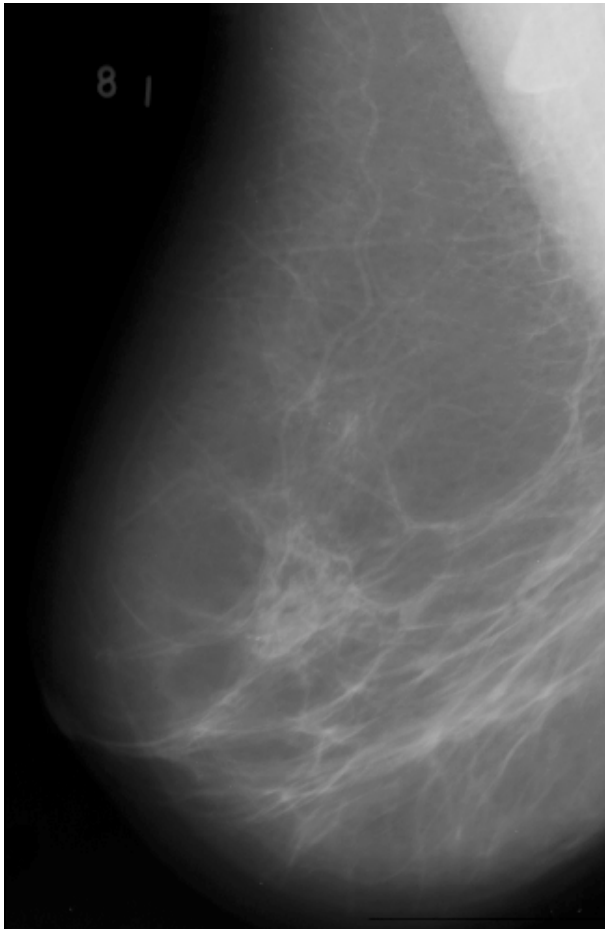
(f)

**Figura 4.17 Filtro gradiente de Roberts:** (a) imagem original; (b) imagem resultante da aplicação do gradiente de Roberts; (c) negativo da imagem em (b); (d) binarização da imagem em (c), com limiar 15; (e) binarização da imagem em (c), com limiar 30; (f) binarização da imagem em (c), com limiar 60.

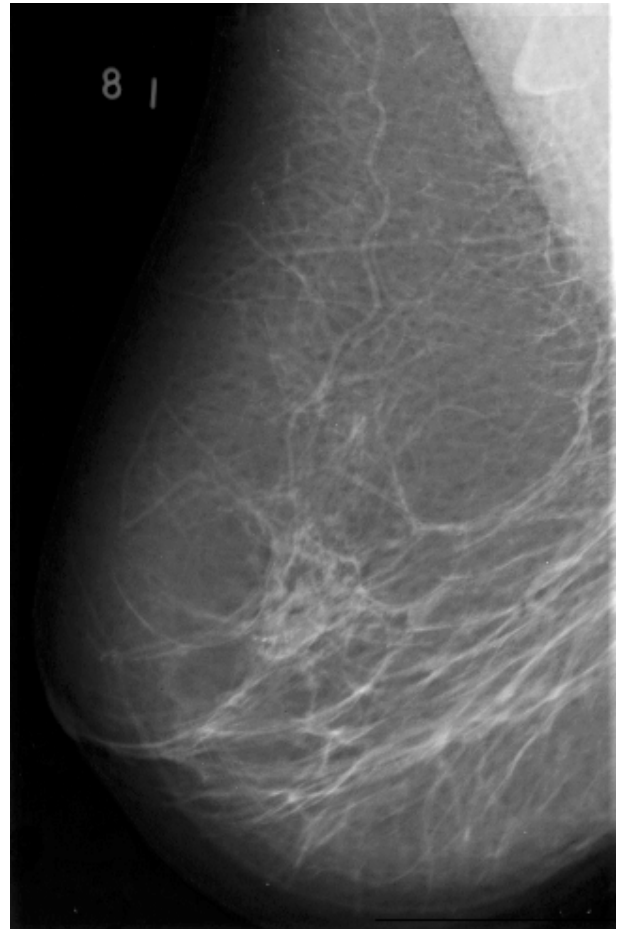
O filtro de reforço de altas frequências (*high-boost* ou *unsharp filter*) é definido como:

$$g(i, j) = [1+c]f(i, j) - cf_{LP}(i, j)$$

onde  $f(i, j)$  é a imagem de entrada,  $f_{LP}(i, j)$  é a imagem resultante da aplicação de um filtro de suavização sobre  $f(i, j)$  e  $c$  é um parâmetro não-negativo, geralmente menor que 1 que controla a intensidade do efeito, que torna-se mais intenso à medida que  $c$  cresce. O resultado da aplicação do filtro de reforço de altas frequências sobre uma imagem mamográfica pode ser visto na Figura 4.18. A imagem  $f_{LP}$  foi obtida aplicando-se o filtro da média com vizinhança  $7 \times 7$ .



(a)



(b)

**Figura 4.18** Filtro de reforço de altas frequências: (a) imagem mamográfica [Fonte: The Mini-MIAS Database of Mammograms, <http://peipa.essex.ac.uk/info/mias.html>]; (b) imagem com reforço de altas frequências, para  $c = 1,4$ .

#### 4.7 OPERAÇÕES SEPARÁVEIS

Uma operação sobre um objeto  $n$ -dimensional é separável quando pode ser realizada pela aplicação sucessiva de operações unidimensionais a cada uma das dimensões do objeto. Uma operação bidimensional separável  $T[f(i,j)]$  pode ser efetuada em dois passos:

1. Aplica-se uma operação unidimensional sobre as linhas de  $f(i,j)$ , produzindo uma imagem  $g_1(i,j)$
2. Aplica-se uma operação unidimensional sobre as colunas de  $g_1(i,j)$ , produzindo a imagem final  $g(i,j)$

A ordem de aplicação (primeiramente nas linhas ou nas colunas) é irrelevante. A decomposição de operações  $n$ -dimensionais em uma sucessão de operações unidimensionais permite implementações computacionalmente muito mais eficientes. Deixa-se ao leitor a tarefa de verificar quais são as operações separáveis apresentadas neste texto.

# Capítulo 5

## Operações Algébricas

---

As operações  $n$ -árias produzem uma imagem de saída aplicando-se operações soma, diferença, multiplicação ou divisão pixel a pixel das imagens de entrada, com possível envolvimento de escalares, como ilustram os exemplos a seguir:

$$g(i, j) = f(i, j) + c \text{ (ou, simplificadamente, } g = f + c)$$

$$g = f - c$$

$$g = f \cdot c$$

$$g = f / c$$

$$g = f_1 + f_2$$

$$g = f_1 - f_2$$

$$g = f_1 \cdot f_2$$

$$g = f_1 / f_2$$

$$g = f_1 + f_2 + \dots + f_n$$

$$g = c_1 f_1 + f_2 / f_3 - f_1 / (f_4 f_5) \dots + c_5 f_n f_n$$

onde  $c$  e  $c_i$  representam escalares. Na convenção aqui adotada, a operação  $g = f + c$  expressa a adição do escalar  $c$  a cada pixel da imagem  $f$ , produzindo a imagem de saída  $g$ . Efeito idêntico é obtido se considerarmos que a operação envolve a adição pixel a pixel da imagem  $f$  com uma outra imagem de intensidade constante e igual a  $c$ . A mesma observação aplica-se às demais operações envolvendo escalares. Vistas desta forma, as operações algébricas podem ser consideradas  $n$ -árias, com  $n \geq 2$ , de acordo com a classificação apresentada no início do Capítulo 4. Apresenta-se a seguir algumas aplicações envolvendo operações algébricas com imagens.

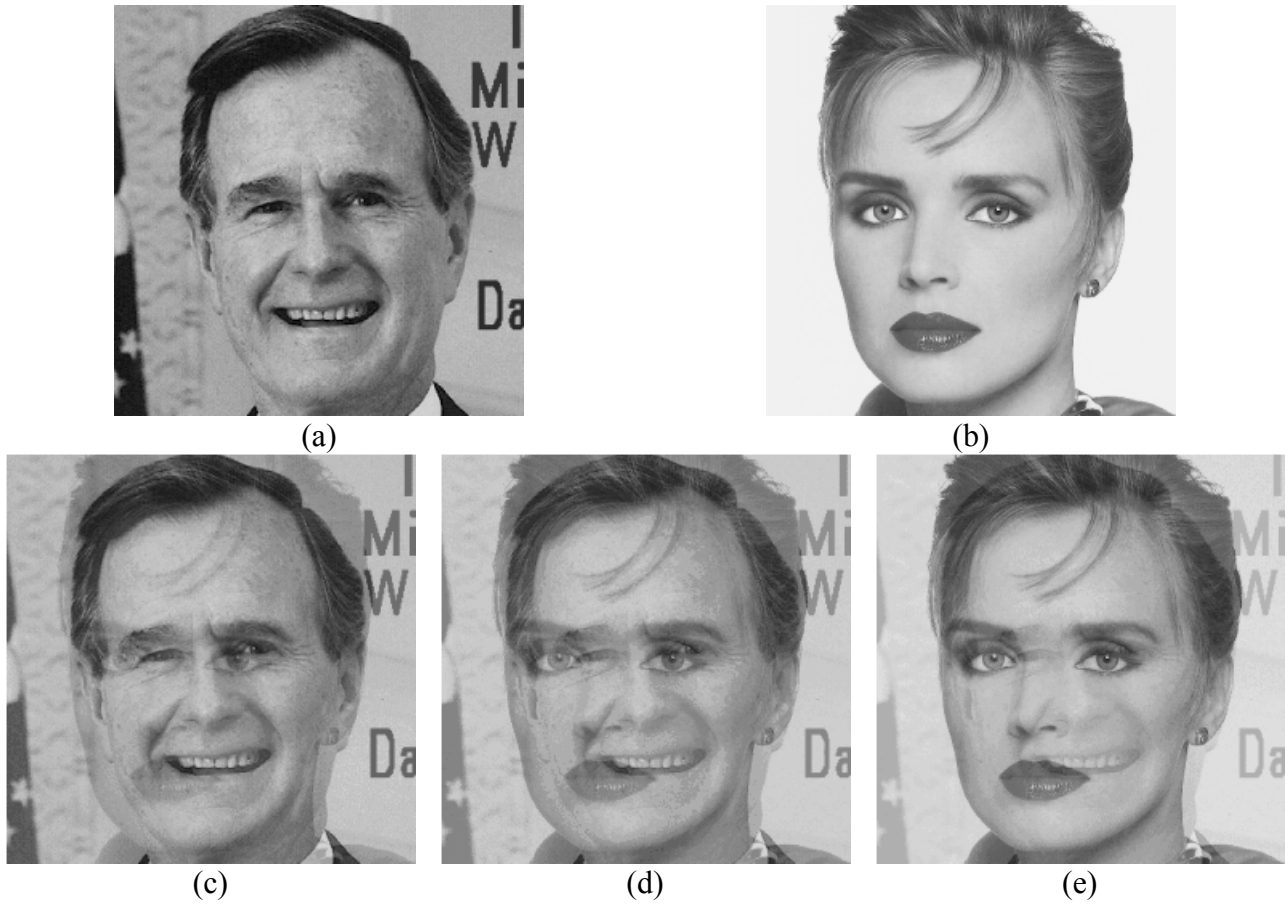
## 5.1 DISSOLVE CRUZADO

O *dissolve cruzado uniforme* de duas imagens  $f$  e  $g$  é uma nova imagem  $h_t$  dada por

$$h_t(i, j) = (1 - t)f(i, j) + t g(i, j)$$

onde o parâmetro  $t$  é um escalar no intervalo  $[0, 1]$ , constante para cada dissolve.

Note-se que a imagem  $h_t$  apresenta em cada ponto uma média ponderada das intensidades dos pontos correspondentes em  $f$  e  $g$ . O parâmetro  $t$  representa a proporção com que  $g$  contribui para formar cada ponto de  $h_t$ . Em particular,  $h_0 = f$  e  $h_1 = g$ . A Figura 1.1 ilustra o efeito para  $t = 0,3$ ;  $t = 0,5$ ; e  $t = 0,7$ .



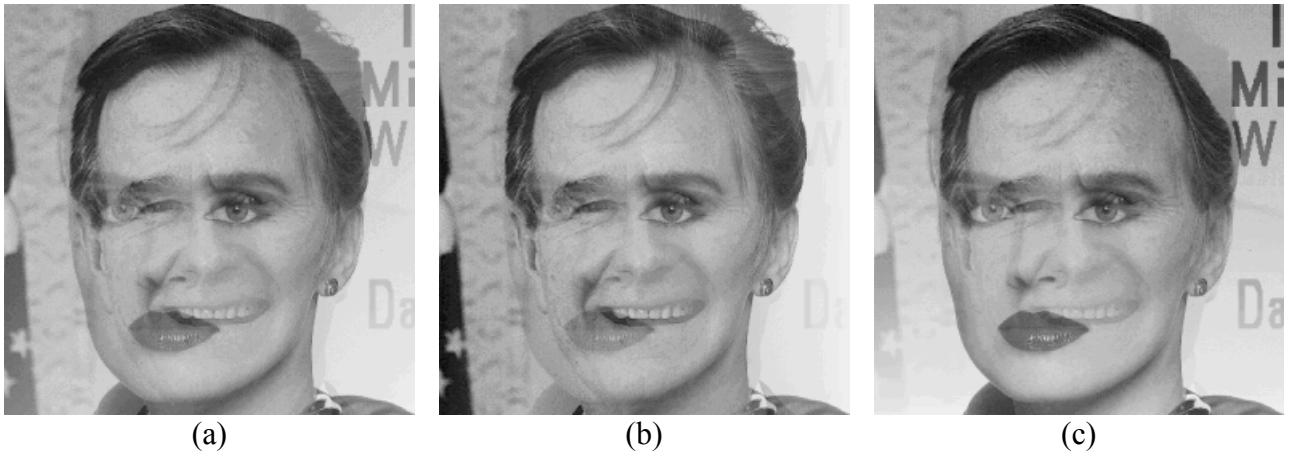
**Figura 5.1** Dissolve cruzado uniforme: (a) imagem  $f$ ; (b) imagem  $g$ ; (c) imagem  $h_{0,3}$ ; (d) imagem  $h_{0,5}$ ; (e) imagem  $h_{0,7}$ .

Em um *dissolve cruzado não-uniforme*  $t$  é uma matriz com as mesmas dimensões de  $f$  e  $g$  cujos elementos  $t(i, j)$  assumem valores no intervalo  $[0, 1]$ :

$$\tilde{h}_t(i, j) = [1 - t(i, j)]f(i, j) + t(i, j) g(i, j)$$

Alguns exemplos de dissolve não uniforme podem ser vistos na Figura 5.2.





**Figura 5.2** Dissolve cruzado não-uniforme sobre as imagens  $f$  e  $g$  da Figura 5.1. Imagem  $h_t$  para (a)  $t(i, j) = (i+j) / (R+C-2)$ ; (b)  $t(i, j) = j / (C-1)$ ; (c)  $t(i, j) = i / (R-1)$ .

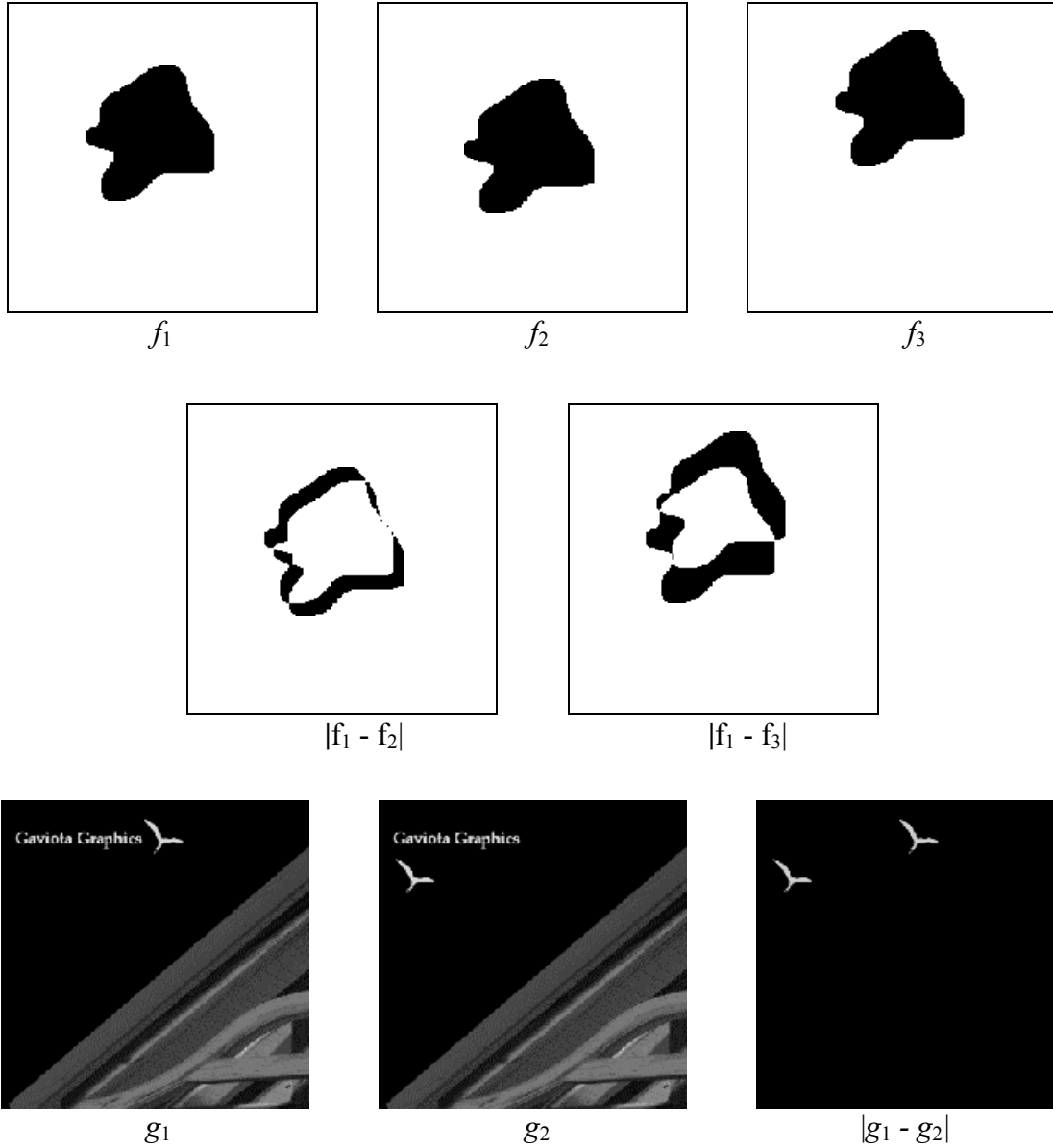
## 5.2 DETECÇÃO DE MOVIMENTO

Um vídeo é uma sucessão temporal de imagens denominadas *quadros*. Assim, todas as técnicas descritas neste texto podem ser diretamente aplicadas aos quadros individuais de um vídeo. Esta seção analisa o problema de detectar os objetos que se moveram entre dois quadros sucessivos de um vídeo apresentando um cenário estacionário. A detecção de movimento encontra aplicações variadas, incluindo visão computacional e compressão de vídeo, sendo de fato um ponto fundamental no esquema de compressão adotado no MPGE2, padrão de vídeo mais usado na atualidade.

Um problema mais genérico e de mesma natureza é o de detectar objetos em movimento entre duas fotografias ou imagens de um mesmo cenário. Ao contrário do vídeo, o intervalo de tempo entre as duas imagens pode ser da ordem de segundos, minutos, horas, dias, meses, ou anos. Em aplicações astronômicas, por exemplo, o intervalo poderia em princípio ultrapassar algumas décadas.

Um método simples para detectar objetos em movimento contra um fundo estacionário entre duas imagens  $f_1$  e  $f_2$  envolve a obtenção da imagem diferença entre  $f_1$  e  $f_2$ . Para realçar as regiões que se deslocam e atenuar o ruído fotográfico e de digitalização, pode-se estabelecer um limiar  $L_t$ :

$$g = \begin{cases} L - 1, & \text{se } |f_1 - f_2| \geq L_t \\ 0, & \text{caso contrario} \end{cases}$$



### 5.3 REDUÇÃO DE RUÍDO POR MÉDIA DE IMAGENS

Em muitas aplicações, é possível obter múltiplas imagens de cenas estacionárias. Se estas imagens são contaminadas em todos os pontos por um ruído aleatório aditivo de média  $\bar{n}$ , com  $0 \leq \bar{n} \leq L-1$ , pode-se encontrar a imagem média com o objetivo de reduzir o ruído.

Considere-se uma imagem  $g_k(i,j)$  formada pela adição de ruído  $n_k(i,j)$  de média zero à imagem original  $f(i,j)$ , ou seja

$$g_k(i,j) = f(i,j) + n_k(i,j)$$

A imagem obtida fazendo-se a média de  $M$  destas imagens ruidosas é

$$\bar{g}(i,j) = \frac{1}{M} \sum_{k=1}^M g_k(i,j)$$



$$\bar{g}(i, j) = \frac{1}{M} \sum_{k=1}^M [f(i, j) + n_k(i, j)]$$

$$\bar{g}(i, j) = f(i, j) + \frac{1}{M} \sum_{k=1}^M n_k(i, j)$$

$$\bar{g}(i, j) = f(i, j) + \bar{n}_M(i, j)$$

onde  $\bar{n}_M(i, j)$  é a média do ruído no ponto  $(i, j)$  nas  $M$  amostras tomadas. A medida que  $M$  aumenta,  $\bar{n}_M(i, j)$  se aproxima de  $\bar{n}$  em todos os pontos da imagem, de forma que o ruído, neste exemplo caracterizado como uma alteração aleatória da intensidade de cada pixel, é substituído por um acréscimo quase constante na intensidade dos pixels. Fazendo-se uma média de muitas imagens obtém-se uma imagem praticamente limpa (sem ruídos), que talvez necessite apenas de uma simples redução no brilho.

# Capítulo 6

## Operações Topológicas

---

As operações topológicas permitem efetuar alterações nas coordenadas dos pixels de uma imagem. Quando a operação introduz deformações, o processo é denominado de *warping*. Combinadas com filtros de amplitude, que permitem efetuar alterações nas intensidades dos pixels, as operações topológicas podem produzir sofisticados efeitos de transição entre duas imagens, tais como a transformação gradual (na forma e na cor) de um objeto de uma imagem em um objeto da outra imagem. Este efeito de metamorfose é denominado *morphing*.

Operações topológicas efetuam um mapeamento das coordenadas  $(i, j)$  dos pixels da imagem de entrada para uma coordenada  $(i', j')$  na imagem de saída, com

$$i' = r(i, j), \quad j' = s(i, j)$$

Por exemplo, a transformação

$$i' = j, \quad j' = i$$

causa um rebatimento da imagem em torno da diagonal principal:

Imagem original

2	0	0
0	0	3
1	1	1

Imagem rebatida  
pela diagonal

2	0	1
0	0	1
0	3	1

Uma rotação de  $-90^\circ$  de uma imagem de dimensões  $R \times C$  pode ser efetuada pela transformação

$$j' = R - 1 - i; \quad i' = j$$

Imagem original

2	0	2	2
0	0	3	3
1	1	1	2

Imagem rotacionada  
por  $90^\circ$

1	0	2
1	0	0
1	3	2
2	3	2

### Rotação e Rebatimento

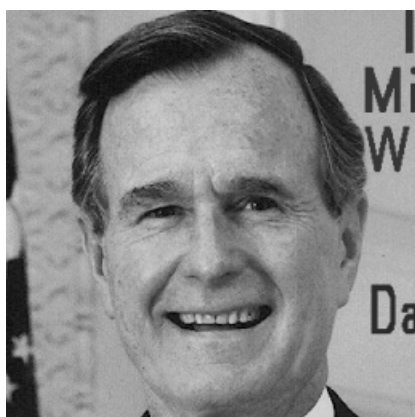
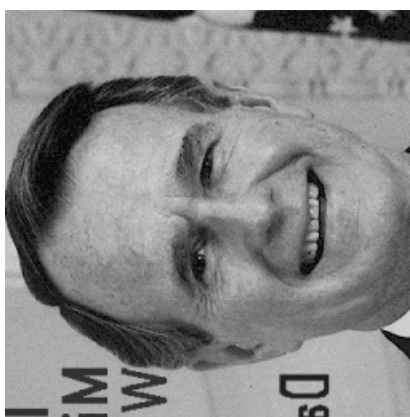
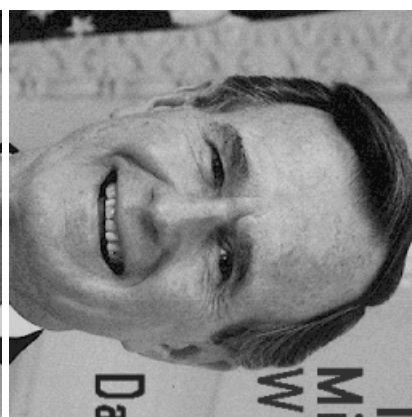


Imagem Original



Rebatimento pela diagonal



Rotação de  $-90^\circ$

## 6.1 ROTAÇÃO

A operação de rotação de uma imagem por um ângulo  $\theta$  em torno de um ponto  $(i_c, j_c)$  é dada por:

$$\begin{aligned} i' &= (i - i_c) \cos \theta - (j - j_c) \sin \theta + i_c \\ j' &= (i - i_c) \sin \theta + (j - j_c) \cos \theta + j_c \end{aligned}$$

## 6.2 EXPANSÃO E CONTRAÇÃO

Se  $p_1$  e  $p_2$  são pontos localizados em uma região  $A$  de uma imagem contínua, uma transformação topológica  $T$  sobre  $A$  é uma *expansão* se

$$|T(p_1) - T(p_2)| > |p_1 - p_2|, \quad \forall p_1, p_2 \in A, \quad p_1 \neq p_2$$

onde  $|a - b|$  denota a distância entre  $a$  e  $b$ .

Uma transformação topológica  $T$  sobre  $A$  é uma *contração* se

$$|(T(p_1) - T(p_2))| < |p_1 - p_2|, \quad \forall p_1, p_2 \in A, \quad p_1 \neq p_2$$

Em termos simples, uma expansão aumenta e uma contração reduz a distância entre os pixels de  $A$ . As definições de contração e de expansão para imagens discretas são análogas, mas os arredondamentos nas coordenadas espaciais podem fazer com que a distância entre alguns pares de pixels não se alterem.

Uma transformação de contração de uma linha de uma imagem contínua pode ser descrita como:

$$j' = s(j) = \frac{j - j_r}{F} + j_r$$

onde  $0 \leq j_r \leq C-1$ , com  $C$  representando o valor máximo da coordenada  $j$ , e  $F \geq 1$  é o fator de contração. Para entender a operação, representemos por  $j_1$  e  $j_2$ , com  $j_1 \leq j_2$ , as coordenadas de dois pontos arbitrários da linha antes da contração. A distância entre eles é dada então por  $j_1 - j_2$ . Após a contração, o ponto na coordenada  $j_1$  é levado para a coordenada  $(j_1 - j_r)/F + j_r$ , e o ponto na coordenada  $j_2$  é levado para a coordenada  $(j_2 - j_r)/F + j_r$ , de forma que a distância entre os pontos se torna  $(j_2 - j_1)/F$ . Como  $F \geq 1$ , a distância se reduz, caracterizando a contração. Uma análise mais cuidadosa da equação de contração dada mostra que, após a operação, todos os pontos da linha se aproximam do ponto de coordenada  $j_r$ .

Para imagens discretas, a contração pode ser descrita como:

$$j' = s(j) = \text{round}\left(\frac{j - j_r}{F} + j_r\right)$$

Agora, a contração pode levar dois ou mais pixels à mesma localização. Neste caso, pode-se atribuir a esta localização um valor igual à média dos pixels levados a ela.

Uma transformação de expansão de uma coluna de uma imagem contínua pode ser descrita como:

$$i' = r(i) = F \cdot (i - i_r) + i_r$$

onde  $0 \leq i_r \leq R-1$ , com  $R$  representando o valor máximo da coordenada  $i$ , e  $F \geq 1$  é o fator de expansão. Após a expansão, todos os pontos da coluna se afastam do ponto de coordenada  $i_r$ .

Para imagens discretas, a expansão pode ser descrita como:

$$i' = r(i) = \text{round}[F \cdot (i - i_r) + i_r]$$

A expansão em imagens discretas pode criar espaços entre dois pixels que eram adjacentes na imagem original. Estes espaços devem ser preenchidos por replicação de pixels ou por interpolação.

Para ilustrar os conceitos de contração e expansão, considere-se a figura a seguir representa os valores dos pixels de uma linha de uma imagem discreta.

20	0	10	20	30	20	0	0
----	---	----	----	----	----	---	---

Calculemos as coordenadas horizontais dos pixels após a contração com  $j_r = 3,5$  e  $F = 2$ :

$$s(0) = \text{round}\left(\frac{0-3,5}{2} + 3,5\right) = 2$$

$$s(1) = \text{round}\left(\frac{1-3,5}{2} + 3,5\right) = 2$$

$$s(2) = \text{round}\left(\frac{2-3,5}{2} + 3,5\right) = 3$$

$$s(3) = \text{round}\left(\frac{3-3,5}{2} + 3,5\right) = 3$$

$$s(4) = \text{round}\left(\frac{4-3,5}{2} + 3,5\right) = 4$$

$$s(5) = \text{round}\left(\frac{5-3,5}{2} + 3,5\right) = 4$$

$$s(6) = \text{round}\left(\frac{6-3,5}{2} + 3,5\right) = 5$$

$$s(7) = \text{round}\left(\frac{7-3,5}{2} + 3,5\right) = 5$$

Os pixels nas colunas 0 e 1 são levados pela contração para a coluna 2, de forma que o valor do pixel na coluna 2 na linha contraída é a média dos valores destes dois pixels na linha original, ou seja,  $(20 + 0)/2$ . Todos os demais mapeamentos múltiplos para uma mesma coluna podem ser resolvidos da mesma maneira, e a linha contraída será:

x	x	10	15	25	0	x	x
---	---	----	----	----	---	---	---

Os pixels marcados com x têm valor indefinido. Na prática, poderia ser atribuído um valor pré-definido a estes pixels, tal como o nível correspondente ao preto ou ao branco. Alternativamente, os valores indefinidos poderiam ser calculados a partir dos valores definidos. Por exemplo, o primeiro valor definido da linha, 10, poderia ser atribuído aos pixels que o antecedem, e o último valor definido da linha, 0, aos pixels que o sucedem.

A expansão na linha original, com os mesmos parâmetros  $i_r$  e  $F$  resulta em:

$$s(0) = \text{round}[2 \cdot (0 - 3,5) + 3,5] = -4$$

$$s(1) = -2$$

$$s(2) = 1$$

$$s(3) = 3$$

$$s(4) = 5$$

$$s(5) = 7$$

$$s(6) = 9$$

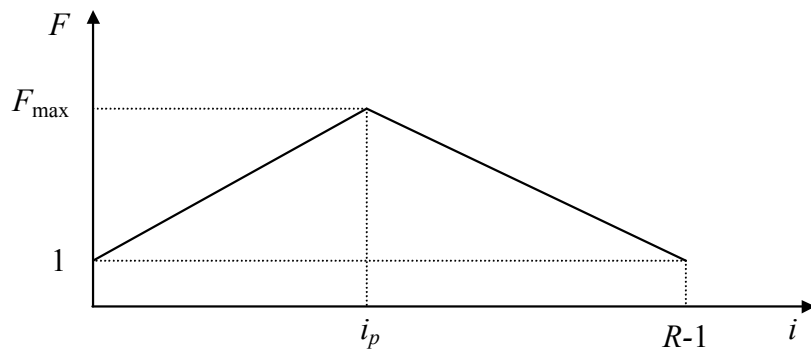
$$s(7) = 11$$

Se a linha expandida é restrita à mesma dimensão da linha original, os pontos mapeados para as coordenadas horizontais -4, -2, 9 e 11 não aparecerão na linha expandida, mas alguns poderão ser utilizados na interpolação destinada a preencher as descontinuidades causadas pela expansão. A linha expandida, com os pontos interpolados mostrados em **negrito**, ficaria assim:

<b>7</b>	10	<b>15</b>	20	<b>25</b>	30	<b>25</b>	20
----------	----	-----------	----	-----------	----	-----------	----

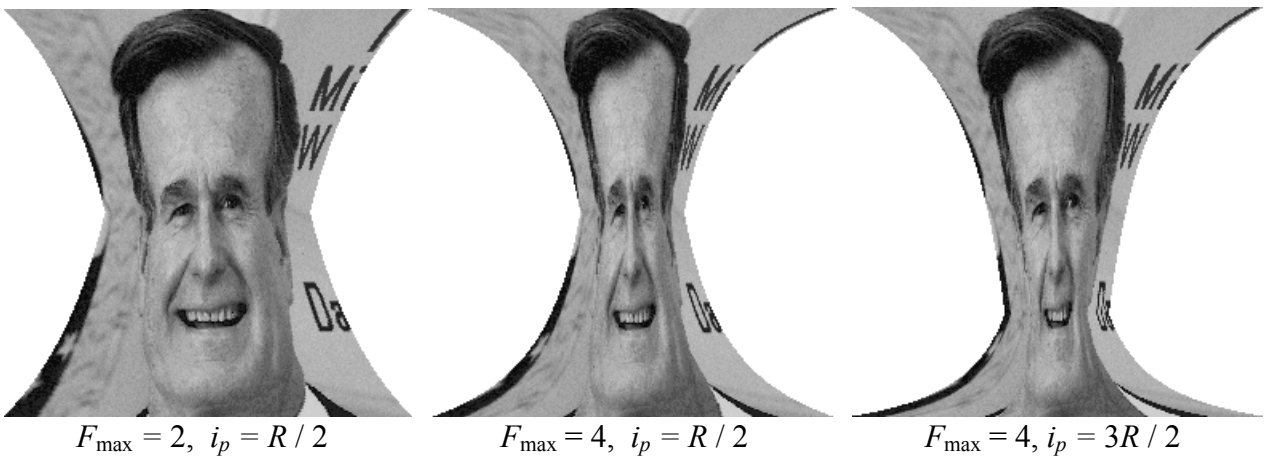
### 6.3 PINCH NA VERTICAL

Uma operação de *warping* conhecida como *pinch* pode ser obtida realizando uma contração na linha  $i = 0, 1, \dots, R - 1$ , com  $F = F(i)$  dado por:

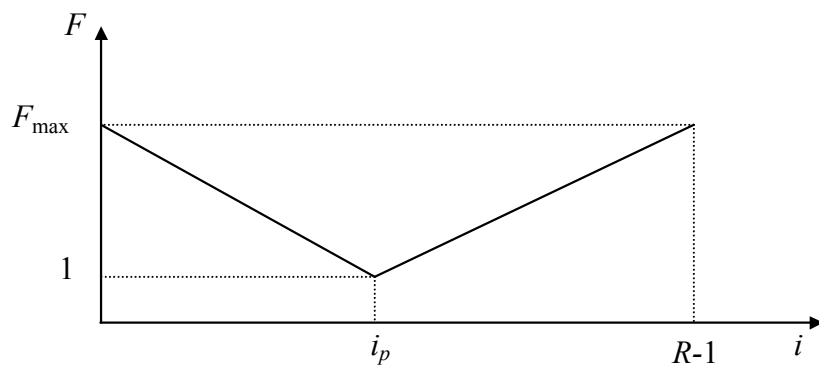


$$F = \begin{cases} \frac{F_{\max} - 1}{i_p} i + 1, & \text{se } i \leq i_p \\ -\frac{F_{\max} - 1}{R - 1 - i_p} (i - i_p) + F_{\max}, & \text{se } i > i_p \end{cases}$$

### ***Pinch Vertical***



Um outro efeito interessante de *pinch*, que denominaremos *pinch* vertical nas bordas, é obtido efetuando uma contração nas linhas da imagem com  $F = F(i)$  dado por:



$$F = \begin{cases} -\frac{F_{\max} - 1}{i_p} i + F_{\max}, & \text{se } i \leq i_p \\ \frac{F_{\max} - 1}{R - 1 - i_p} (i - i_p) + 1, & \text{se } i > i_p \end{cases}$$

### Pinch Vertical nas Bordas



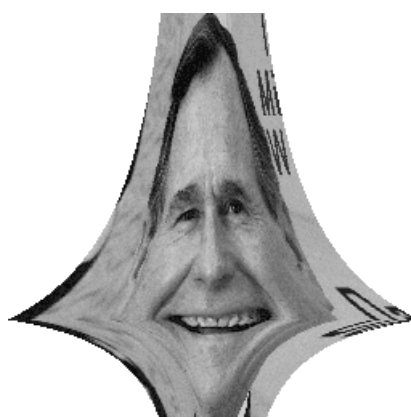
$$F_{\max} = 2, i_p = R / 2$$



$$F_{\max} = 4, i_p = R / 2$$



$$F_{\max} = 2, i_p = 3R / 2$$



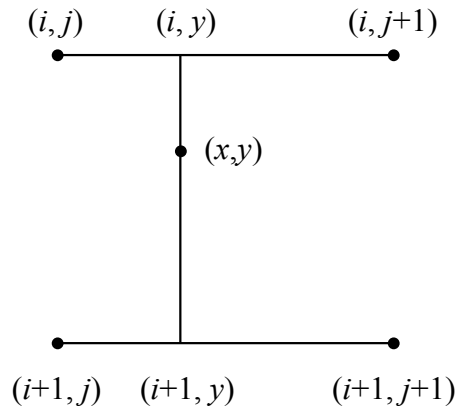
$$F_{\max} = 4, i_p = 3R / 2$$

## 6.4 ZOOM

A operação de *zoom* permite ampliar uma imagem ou uma região de uma imagem. O efeito pode ser obtido de forma simples por replicação de *pixels*. Para obter uma ampliação de  $n$  vezes, o pixel nas coordenadas  $(i, j)$  da imagem original é replicado na imagem ampliada em uma área quadrada com vértices em  $(i*n, j*n)$ ,  $(i*n + n - 1, j*n)$ ,  $(i*n, j*n + n - 1)$  e  $(i*n + n - 1, j*n + n - 1)$ . Por sua própria natureza, o zoom por replicação de pixels produz o aparecimento de blocos claramente discerníveis na imagem ampliada, especialmente para valores elevados de  $n$ .

O *zoom* por interpolação bilinear evita o surgimento de blocos, gerando uma ampliação com aspecto mais natural. O método pode ser entendido observando-se o esquema abaixo, que representa quatro pixels adjacentes na imagem original, situados nas coordenadas  $(i, j)$ ,  $(i, j+1)$ ,  $(i+1, j)$  e  $(i+1, j+1)$ .





Os níveis de cinza dos pixels nas coordenada  $(i, y)$  e  $(i+1, y)$ , com  $y$  no intervalo  $[j, j+1]$ , podem ser calculado por interpolação linear:

$$\begin{aligned} f(i, y) &= f(i, j) + (y - j) [f(i, j+1) - f(i, j)] \\ f(i+1, y) &= f(i+1, j) + (y - j) [f(i+1, j+1) - f(i+1, j)] \end{aligned}$$

Agora, o nível de cinza do pixel em  $(x, y)$ , com  $x$  no intervalo  $[i, i+1]$  e  $y$  no intervalo  $[j, j+1]$ , pode ser encontrado por interpolação linear entre níveis calculados nas coordenadas  $(i, y)$  e  $(i+1, y)$ :

$$f(x, y) = f(i, y) + (x - i) [f(i+1, y) - f(i, y)]$$

O processo completo acima descrito para cálculo do nível de cinza na coordenada fracionária  $(x, y)$  é denominado interpolação bilinear. Observe-se que as três equações de interpolação definem um valor real para o nível de cinza de um ponto situado em qualquer coordenada real dentro dos limites da imagem. Em outros termos, as equações acima permitem construir uma imagem analógica a partir de uma imagem digital. Este tipo de operação é comumente conhecido por *reconstrução de imagem*.

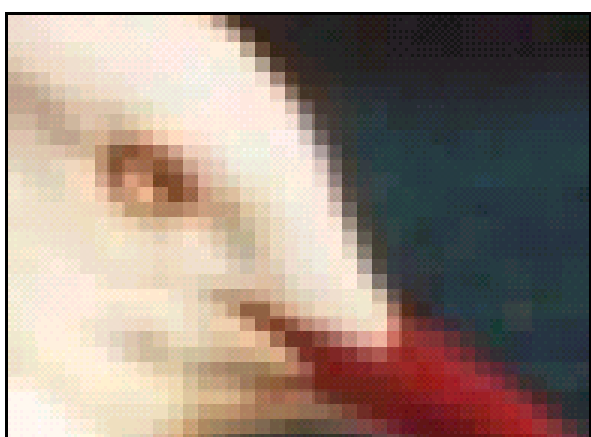
A partir da interpolação bilinear, pode-se efetuar a operação de *zoom* atribuindo-se aos *pixels* criados pela ampliação os valores na imagem analógica reconstruída, arredondados para o inteiro mais próximo.

O zoom de ampliação ou aproximação, descrito acima, é muitas vezes chamado de *zoom in*. O zoom de redução ou afastamento, *zoom out*, por um fator  $n$ , é obtido particionando-se a imagem em quadrados adjacentes de  $n \times n$  pixels, e substituindo-se estes quadrados por um único pixel na imagem reduzida. O valor deste pixel pode ser a média dos valores dos pixels presentes no quadrado, ou o valor de um único pixel em uma dada posição do quadrado.

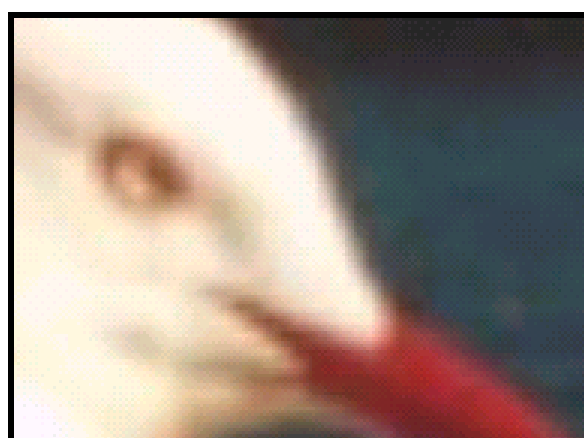
## Zoom



Imagem original



Zoom por replicação de pixels



Zoom por interpolação bilinear

## 6.5 WARPING BASEADO EM CAMPOS

A operação de *warping* (deformação) permite alterar livremente a localização dos pixels de uma imagem. O *warping* encontra larga aplicação na indústria do entretenimento e na publicidade, especialmente na criação de seqüências de animação e de efeitos especiais. A técnica é também empregada na área de realidade virtual; na correção de distorções óticas ou projetivas; no alinhamento de elementos da imagem, para fins de comparação; na operação de *morphing*, descrita na Seção 6.6; na ampliação de imagens com preservação de fronteiras nítidas entre os objetos; no registro de imagens (onde uma imagem é distorcida de forma que ; etc.

No *warping* baseado em campos, características consideradas importantes (por exemplo, nariz, boca, contorno da face) na imagem são marcadas por segmentos de retas orientados, que denominaremos vetores-referência. Para cada vetor-referência, um vetor-alvo é especificado. Se um único par de vetores referência-alvo é criado, o problema consiste em definir o ponto  $X'$  na imagem distorcida para onde um dado ponto  $X$  na imagem original será levado, de forma que as relações espaciais entre  $X'$  e o vetor-alvo seja idêntica àquelas entre  $X$  e o vetor-referência. No *warping* tratado nesta seção, as relações espaciais que devem ser preservadas são mostradas na próxima figura.

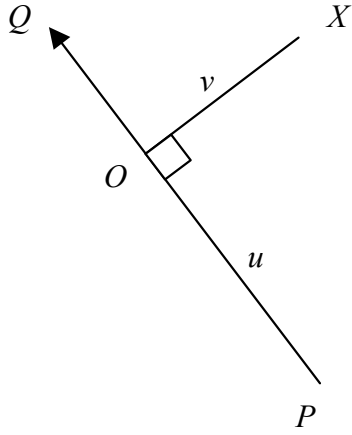


Imagem original

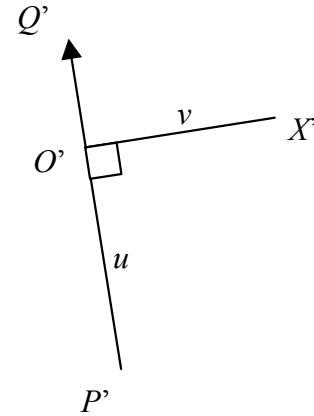


Imagem distorcida

Os parâmetros  $u$  e  $v$ , precisamente definidos nos próximos parágrafos, estabelecem as relações espaciais a serem preservadas. O parâmetro  $u$  representa o deslocamento de  $P$  até  $O$  no sentido do vetor  $PQ$ , dividido por  $\|P - Q\|$ . Assim, se  $O = P$ ,  $u = 0$ ; se  $O = Q$ ,  $u = 1$ ; se  $O$  está situado entre  $P$  e  $Q$ ,  $0 < u < 1$ ; se  $O$  está situado após o ponto  $Q$ ,  $u > 1$  e, se está situado antes do ponto  $P$ ,  $u < 0$ . O parâmetro  $v$  representa a distância entre o ponto  $X$  e a reta que contem o segmento  $PQ$ .

Dado o valor de  $u$ , a especificação de um valor não nulo para  $v$  define dois pontos, localizados à direita e à esquerda do segmento  $PQ$ . A ambigüidade pode ser eliminada convencionando-se atribuir a  $v$  um sinal positivo se o ponto  $X$  estiver localizado à direita do segmento  $PQ$ ; caso contrário, o sinal atribuído é negativo.

O ponto  $X'$  pode ser encontrado aplicando-se as fórmulas:

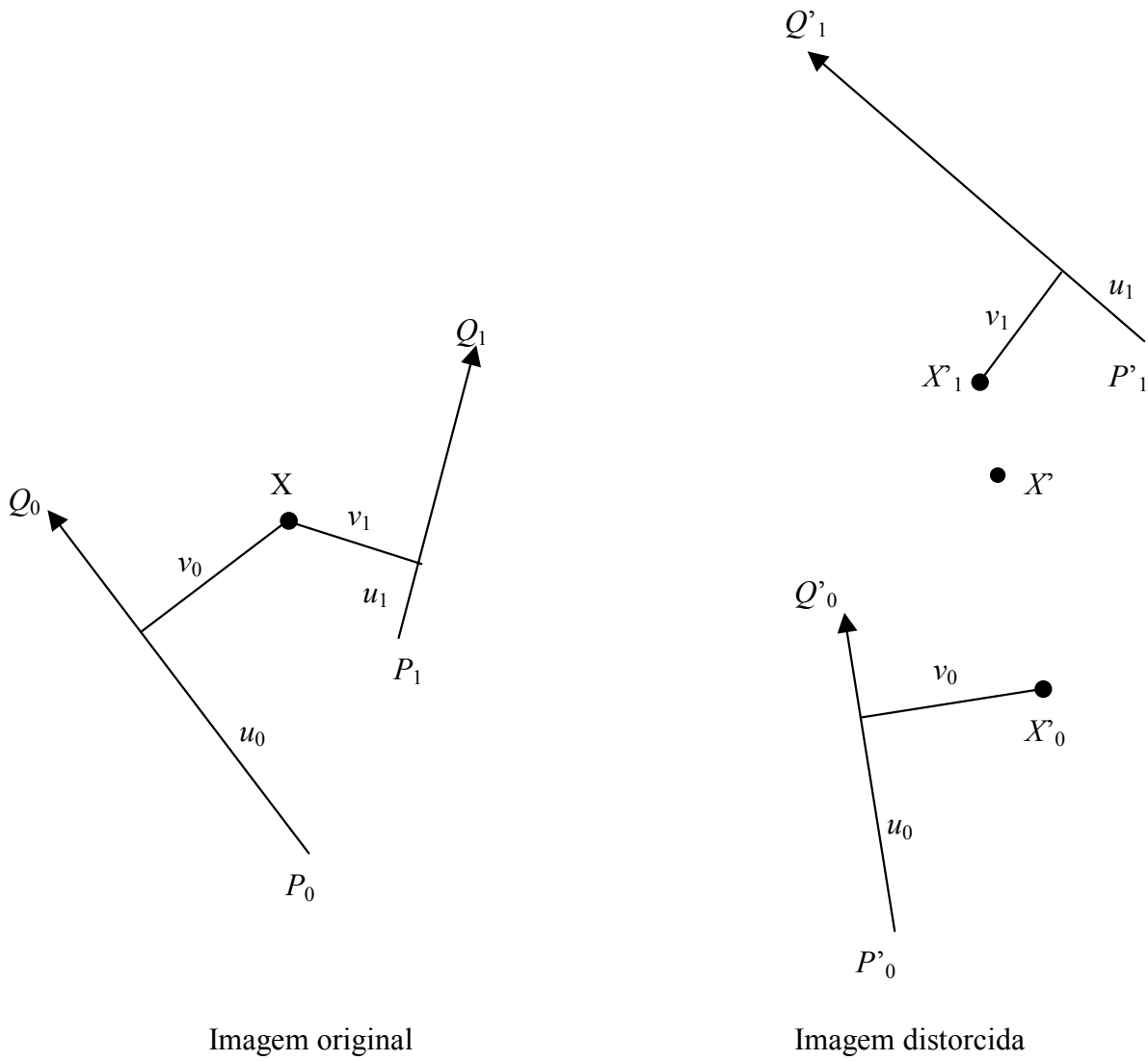
$$u = \frac{(X - P) \cdot (Q - P)}{\|Q - P\|^2}$$

$$v = \frac{(X - P) \cdot \perp(Q - P)}{\|Q - P\|}$$

$$X' = P' + u \cdot (Q' - P') + \frac{v \cdot \perp(Q' - P')}{\|Q' - P'\|}$$

onde  $\perp V$  representa um vetor perpendicular a  $V$  e de mesmo módulo que este. A ambigüidade no cálculo de  $v$  encontra-se exatamente na especificação de  $\perp V$ , uma vez que há dois vetores perpendiculares a  $V$  e de mesma dimensão que este, que diferem apenas no sentido. A convenção de considerar positivo o sentido “à direita do segmento  $PQ$ ” elimina a ambigüidade.

Quando há mais de um par de vetores referência-alvo, o problema é um pouco mais complexo. Nesta situação, cada ponto da imagem sofrerá a influência de todos os pares de vetores. Neste caso, será em geral encontrado um ponto diferente para cada par de vetores referência-alvo. A próxima figura ilustra o problema para dois pares de vetores,  $(P_0Q_0, P'_0Q'_0)$  e  $(P_1Q_1, P'_1Q'_1)$ .



Os diferentes pontos para os quais o ponto  $X$  da imagem original é levado por cada par de vetores referência-alvo são combinados por intermédio de uma média ponderada, produzindo o ponto  $X'$  para onde  $X$  será efetivamente levado. O peso  $w_i$  com que o segmento  $P_iQ_i$  contribuirá para a produção de  $X'$  mantém uma relação inversa com a distância entre a reta e o ponto  $X$ :

$$w_i = \left( \frac{l_i^p}{a + d_i} \right)^b$$

onde  $l_i = \|P_i - Q_i\|$ ;  $d$  é a distância entre  $X$  e o segmento  $P_iQ_i$ ; e  $a$ ,  $b$  e  $p$  são parâmetros não negativos descritos a seguir.

O parâmetro  $a$  define a aderência do segmento: valores de  $a$  próximos a zero fazem com que o segmento tenha um peso muito elevado sobre pontos próximos a ele e, conseqüentemente, exerça uma forte influência sobre estes pontos. Se  $a = 0$ , o segmento terá peso infinito na determinação das alterações de coordenadas dos pontos localizados sobre ele, o que significa estes pontos permanecerão sobre o segmento após a transformação. Valores elevados de  $a$  deixam os pontos próximos ao segmento, e até mesmo sobre o próprio segmento, mais susceptíveis à ação dos outros segmentos.

O parâmetro  $p$  controla a importância do tamanho do segmento nas alterações de coordenadas. Se  $p = 0$ , o peso independe do tamanho do segmento. Se  $p \neq 0$ , quanto maior o

segmento maior o peso a ele associado. Quanto maior o valor de  $p$ , maior a influência do tamanho do segmento nas deformações.

Finalmente,  $b$  controla a forma com que a influência do segmento sobre um dado ponto decresce em função da distância. Se  $b = 0$ , o peso independe de  $d$  e, quanto maior  $b$ , mais drástica é a taxa de decrescimento em função de  $d$ .

Experimentos com os parâmetros mostram que bons resultados, em termos de qualidade visual e tempo de processamento, são obtidos com  $a = 0$  ou  $a = 1$ ,  $b = 2$  e  $p = 0$  ou  $p = 1$ .

Apresenta-se a seguir o pseudo-código apresentado no artigo "Feature-Based Image Metamorphosis", publicado no *Siggraph '92* por Thaddeus Beier and Shawn Neely, para o cálculo de  $X'$ . Observe-se que o pseudo-código apresentado utiliza mapeamento reverso.

```

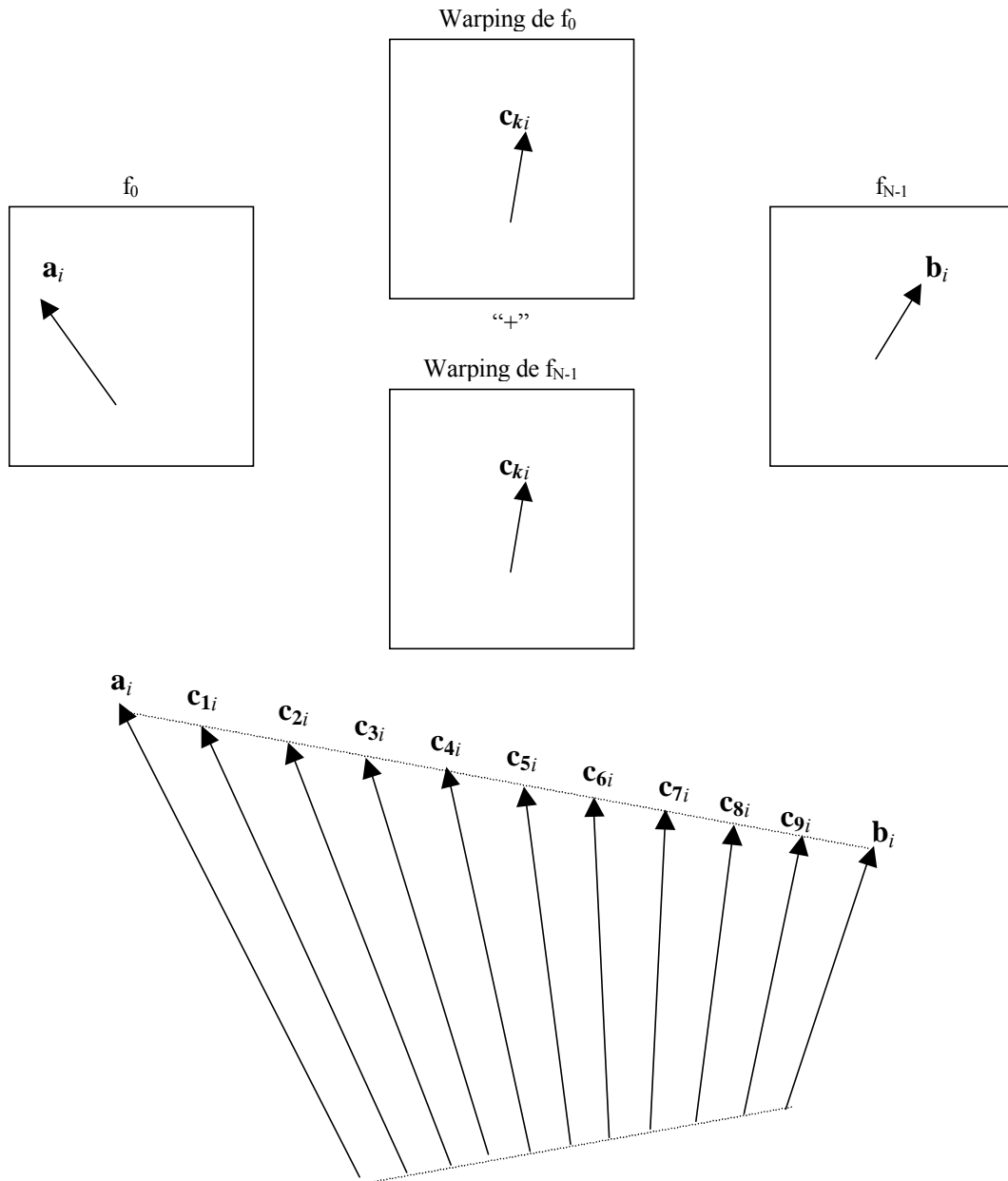
For each pixel X in the destination image
    DSUM = (0,0)
    weightsum = 0
    for each line  $P_iQ_i$ 
        calculate  $u, v$  based on  $P_iQ_i$ 
        calculate  $X'_i$  based on  $u, v$  and  $P_i'Q_i'$ 
        calculate displacement  $D_i = X'_i - X_i$  for this line
        dist = shortest distance from X to  $P_iQ_i$ 
        weight =  $[\text{length}^p / (a + \text{dist})]^b$ 
        DSUM +=  $D_i * \text{weight}$ 
        weightsum += weight
     $X' = X + \text{DSUM} / \text{weightsum}$ 
    destinationImage(X) = sourceImage( $X'$ )

```

## 6.6 MORPHING BASEADO EM CAMPOS

O *morphing* (metamorfose) é essencialmente uma técnica de interpolação de formas e cores entre duas imagens distintas, aqui identificadas por  $f_0$  e  $f_{N-1}$ . O objetivo é encontrar as imagens intermediárias  $f_1, f_2, \dots, f_{N-2}$ , de maneira a gerar uma transição gradual entre  $f_0$  e  $f_{N-1}$ . Assim, a imagem interpolada  $f_k$ ,  $k = 1, 2, \dots, N-2$ , afasta-se progressivamente de  $f_0$  e aproxima-se progressivamente de  $f_{N-1}$  à medida que  $k$  aumenta de 1 até  $N-2$ . O *morphing* é utilizado na criação de efeitos especiais na publicidade e na indústria cinematográfica; em realidade virtual; em compressão de vídeo; etc.

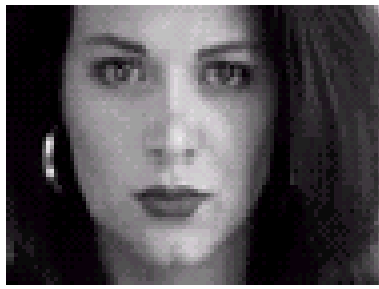
Da forma análoga à operação efetuada no *warping* baseado em campos, no *morphing* baseado em campos as características consideradas importantes na imagem  $f_0$  são marcadas por  $v$  vetores de referência, que representaremos por  $\mathbf{a}_i$ ,  $i = 1, 2, \dots, v$ . Para cada vetor-referência em  $f_0$ , um vetor-referência  $\mathbf{b}_i$  é especificado em  $f_{N-1}$ . Para cada imagem interpolada  $f_k$ ,  $k = 1, 2, \dots, N-2$ , e para cada par de vetores  $\mathbf{a}_i - \mathbf{b}_i$ , um novo vetor, que denominaremos vetor-alvo, representado por  $\mathbf{c}_{ki}$ , é criado por interpolação entre os vetores do par  $\mathbf{a}_i - \mathbf{b}_i$ . Efetua-se então um *warping* sobre  $f_0$  usando os pares  $\mathbf{a}_i - \mathbf{c}_{ki}$ , e um *warping* sobre  $f_{N-1}$  usando os pares  $\mathbf{b}_i - \mathbf{c}_{ki}$ . Se desejarmos gerar, por exemplo, nove imagens interpoladas, deveremos gerar nove vetores-alvo interpolados para cada par de vetores  $\mathbf{a}_i - \mathbf{b}_i$  especificados nas imagens  $f_0$  e  $f_{10}$ . A figura seguinte ilustra graficamente o procedimento.



As duas imagens geradas pelo *warping* duplo para um dado  $k$  são combinadas com a operação de dissolve cruzado uniforme para gerar a imagem interpolada  $f_k$ ,  $k = 1, 2, \dots, N-2$ . Para uma metamorfose realista, o peso de  $f_0$  no dissolve deve decrescer (linearmente, por exemplo) à medida que  $k$  aumenta de 1 até  $N-2$ .

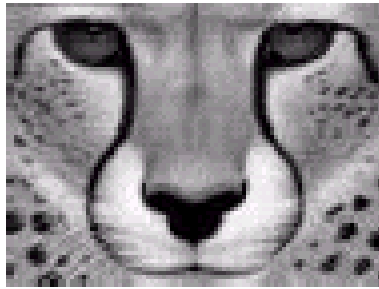
O parâmetro peso do dissolve seria  $t = 0,1$  para  $f_1$ ;  $t = 0,2$  para  $f_2$ ;  $t = 0,3$  para  $f_3$  e assim por diante.

**Morphing = Warping + Dissolve**



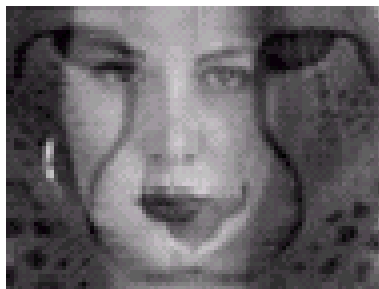
+

+

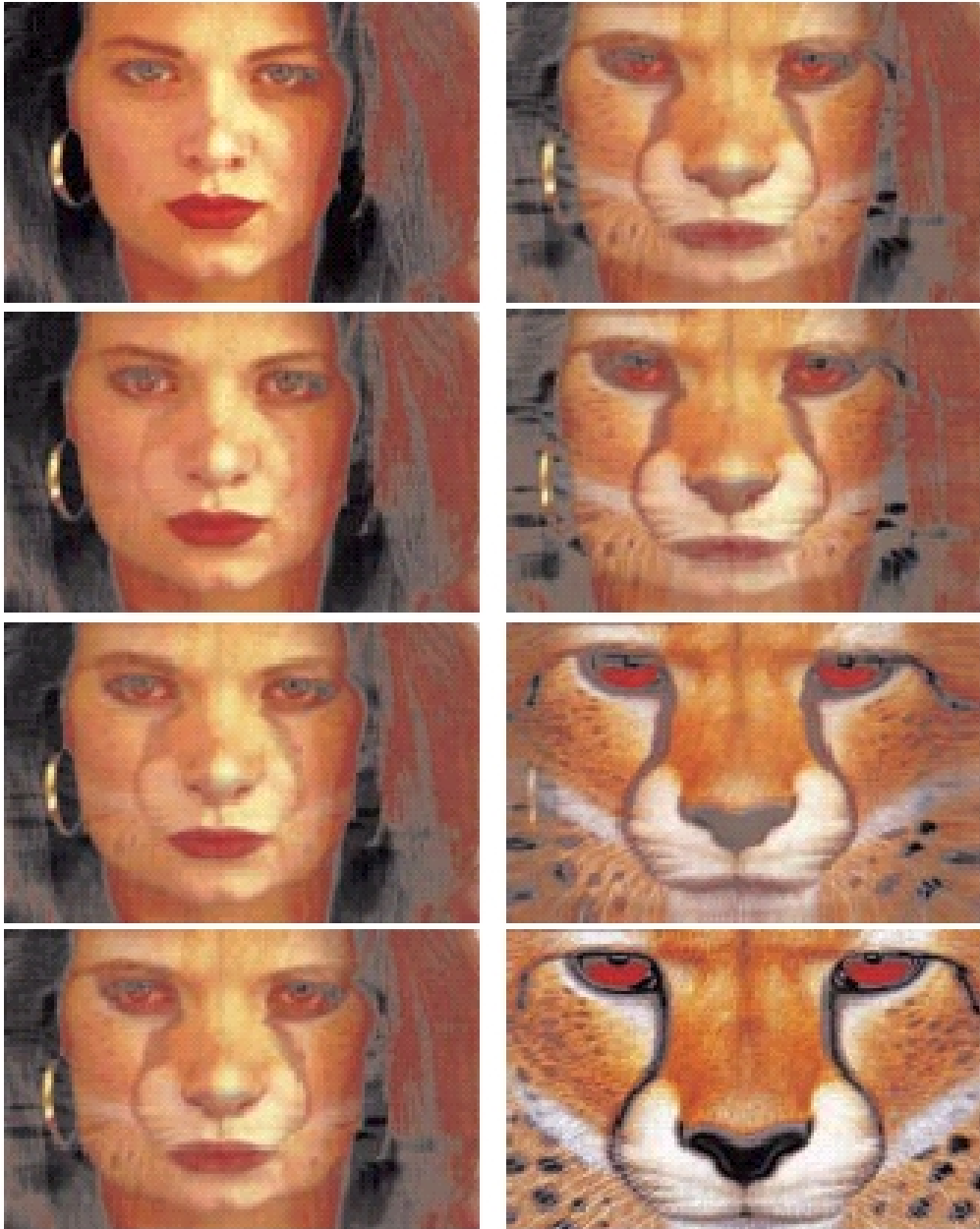


=

=



## Morphing





# Capítulo 7

## Transformadas

---

Em algumas situações é mais conveniente aplicar-se uma operação matemática genericamente denominada de transformada sobre um sinal a ser processado, convertendo-o para o domínio da frequência, efetuar o processamento do sinal neste domínio, e finalmente converter o sinal processado para o domínio original.

A teoria das transformadas representa um papel dos mais importantes na área de processamento de sinais e imagens. As transformadas geram um conjunto de *coeficientes* a partir dos quais é possível restaurar as amostras originais do sinal. Neste capítulo serão apresentadas duas transformadas muito utilizadas para conversão de sinais  $n$ -dimensionais para o domínio da frequência.

### 7.1 TRANSFORMADA COSSENO DISCRETA

Se  $x[n]$ ,  $n = 0, 1, \dots, N-1$ , são os elementos de uma seqüência  $\mathbf{x}$ , a DCT unidimensional de  $\mathbf{x}$  gera uma seqüência  $\mathbf{X}$  cujos coeficientes  $X[k]$   $k = 0, 1, \dots, N-1$ , são dados por:

$$X[k] = \left(\frac{2}{N}\right)^{1/2} c_k \sum_{n=0}^{N-1} x[n] \cos\left[\frac{(2n+1)k\pi}{2N}\right], \quad k = 0, 1, \dots, N-1$$

onde

$$c_k = \begin{cases} (1/2)^{1/2} & \text{para } k = 0 \\ 1 & \text{para } k = 1, 2, \dots, N-1 \end{cases}$$

Esta operação leva vetores de um determinado domínio (comumente, do domínio do tempo ou do espaço) para o chamado domínio da frequência. O coeficiente  $X[0]$ , que representa a componente de frequência zero e está relacionado diretamente com o valor médio de  $\mathbf{x}$ , é conhecido como *coeficiente DC* (de *direct current*); os demais são os *coeficientes AC* (de *alternating current*).

A sequência  $\mathbf{x}$  pode ser recuperada aplicando-se a  $\mathbf{X}$  a DCT inversa:

$$x[n] = \left(\frac{2}{N}\right)^{1/2} \sum_{k=0}^{N-1} c_k X[k] \cos\left[\frac{(2n+1)k\pi}{2N}\right], \quad n = 0, 1, \dots, N-1$$

Há diversos algoritmos para o cálculo eficiente da DCT, mas, ainda assim, o custo computacional quando a sequência  $\mathbf{x}$  contém muitos elementos pode ser elevado.

## 7.2 TRANSFORMADA DE FOURIER DISCRETA DE FUNÇÕES DE UMA VARIÁVEL

A Transformada de Fourier Discreta (DFT),  $F[u]$ ,  $u = 0, 1, \dots, N-1$ , de uma função discreta unidimensional  $s[n]$ ,  $n = 0, 1, \dots, N-1$  é dada por:

$$F[u] = \frac{1}{N} \sum_{n=0}^{N-1} s[n] e^{-\frac{j2\pi un}{N}}, \quad u = 0, 1, \dots, N-1$$

com

$$j = \sqrt{-1}$$

$$e^{\pm j\theta} = \cos\theta \pm j \sin\theta$$

A operação transforma os  $N$  elementos da função discreta em  $N$  elementos denominados *coeficientes*. Note-se que os coeficientes da DFT são, em geral, complexos, mesmo quando  $s[n]$  é real. A operação inversa, que permite reconstruir a função  $s[n]$  a partir dos coeficientes  $F[u]$ , é conhecida como transformada de Fourier discreta inversa (IDFT):

$$s[n] = \sum_{u=0}^{N-1} F[u] e^{\frac{j2\pi un}{N}}, \quad n = 0, 1, \dots, N-1$$

O módulo do coeficiente  $F[u]$  indica a importância com que o termo de frequência  $u/N$  contribui para a formação de  $s[n]$ . Quanto maior o módulo de  $F[u]$ , maior a importância da componente de frequência  $f_u = u/N$  em  $s[n]$ . O coeficiente  $F[0]$  representa a contribuição da componente de frequência zero, ou seja, representa um nível constante, e está diretamente relacionado com a média de  $s[n]$ . Por analogia com os circuitos de corrente direta e alternada,  $F[0]$  é denominado componente DC (de *direct current*), e os demais coeficientes são os componentes AC (de *alternating current*). Observe-se também que todas as frequências das exponenciais complexas que constituem os termos do somatório da IDFT são múltiplas da *frequência fundamental*  $f_1 = 1/N$ . O módulo da transformada de Fourier é denominado *espectro de Fourier*.

Na análise que se segue, consideraremos que o sinal original é função do tempo. A extensão dos resultados para outros domínios é imediata.

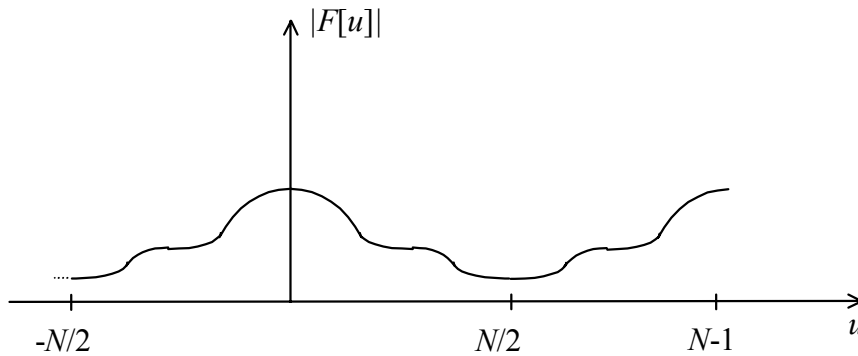
Seja  $\Delta x$  o período de amostragem, ou seja, o tempo, medido em segundos, entre duas amostras consecutivas de  $f[n]$ . A taxa de amostragem é, portanto,  $f_a = 1/\Delta x$  Hertz. As  $N$  amostras do sinal equivalem a  $N\Delta x$  segundos e, assim, a frequência fundamental  $f_1 = 1/N$  pode ser expressa em segundos substituindo-se  $N$  por  $N\Delta x$ :

$$f_1 = \frac{1}{N\Delta x} = \frac{f_a}{N} s$$

A DFT é periódica de período  $N$ :

$$F[u + N] = \frac{1}{N} \sum_{n=0}^{N-1} f[n] e^{-j2\pi(u+N)n/N} = \frac{1}{N} \sum_{n=0}^{N-1} s[n] e^{-j2\pi un/N} = F(u)$$

Se  $f[n]$  é real, pode-se demonstrar facilmente que  $|F[u]| = |F[-u]|$ , ou seja, o espectro de Fourier é simétrico com relação ao eixo  $|F[u]|$ . Da periodicidade e da simetria conclui-se que o gráfico de  $|F[u]|$  assemelha-se ao gráfico abaixo:



Assim, na DFT, as amostras  $0, \pm N, \pm 2N, \dots$ , correspondem à frequência zero, enquanto que  $\pm N/2, \pm 3N/2, \dots$ , correspondem à maior frequência possível. Esta frequência máxima é, portanto,  $N/2$  vezes maior que o valor da frequência fundamental  $1/(N\Delta x)$ . Assim, dada a frequência de amostragem  $f_a = 1/\Delta x$ , a frequência máxima em Hz é dada por:

$$f_{max} = \frac{N}{2} \frac{1}{N\Delta x} = \frac{f_a}{2}$$

O fato de o eixo  $u$  representar frequências crescentes até  $N/2$  e decrescentes de  $N/2$  a  $N-1$  dificulta a análise visual do espectro, de forma que é mais comum exibir o gráfico de  $|F[u]|$  no intervalo  $[-N/2, N/2]$ , interpretando-se as frequências negativas como uma mera conveniência matemática.

### 7.3 TRANSFORMADA DE FOURIER DISCRETA DE FUNÇÕES DE DUAS VARIÁVEIS

A DFT pode ser facilmente estendida para o caso de funções bidimensionais:

$$F[u, v] = \frac{1}{RC} \sum_{m=0}^{C-1} \sum_{n=0}^{R-1} s[m, n] e^{-j2\pi(um/C + vn/R)},$$

com

$$u = 0, 1, \dots, R-1$$

$$v = 0, 1, \dots, C - 1$$

A IDFT bidimensional é dada por:

$$s[m, n] = \sum_{u=0}^{C-1} \sum_{v=0}^{R-1} F[u, v] e^{j2\pi(um/C + vn/R)},$$

$$m = 0, 1, \dots, R - 1$$

$$n = 0, 1, \dots, C - 1$$

A DFT é uma operação separável, e considerações análogas àsquelas feitas para a DFT de funções de uma variável continuam válidas, com as devidas adaptações para o caso bidimensional. Agora, por exemplo, a DFT é periódica com período  $R$  na horizontal e com período  $C$  na vertical e, se  $s(m, n)$  é real,  $|F[u, v]| = |F[u, -v]| = |F[-u, v]| = |F[-u, -v]|$ .

Para inspeção visual, é interessante exibir o espectro de Fourier de funções bidimensionais como uma imagem, com coeficientes de valores mais elevados aparecendo como pontos mais claros. A conversão dos valores do espectro para valores permitidos de níveis de cinza pode ser efetuada como abaixo:

$$F_{im}[u, v] = \text{round}[(L - 1)|F[u, v]| / F_{max}]$$

onde  $L$  é o número de níveis de cinza permitido e  $F_{max}$  é o valor máximo de  $|F[u, v]|$ .

Comumente, os espectros de Fourier decaem rapidamente com a frequência e, assim, as componentes de alta frequência são obscurecidas na exibição, aparecendo como pontos pretos. Para melhorar a visualização destas componentes e obter valores adequados a uma imagem digital, pode-se criar a imagem

$$F_{log}[u, v] = \text{round}[(L - 1) \log(1 + |F[u, v]|) / F_{max2}]$$

onde  $F_{max2}$  é o maior valor presente em  $\log(1 + |F[u, v]|)$ . Observe-se que esta fórmula preserva os zeros de  $|F[u, v]|$ , pois  $F_{log}[u, v] = 0$  quando  $|F[u, v]| = 0$ .

Dadas as características de simetria e periodicidade da DFT bidimensional, é normalmente mais fácil analisar o espectro deslocando-se  $F_{im}[u, v]$  ou  $F_{log}[u, v]$  de forma a posicionar o ponto de frequência zero no centro da imagem.

## Exibição do Espectro de Fourier

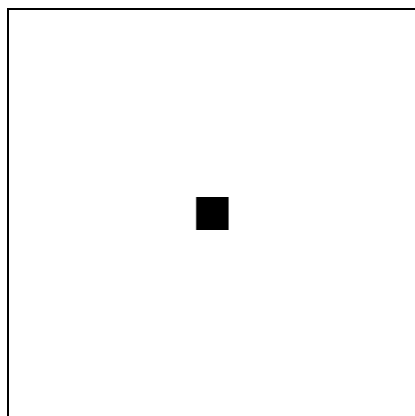
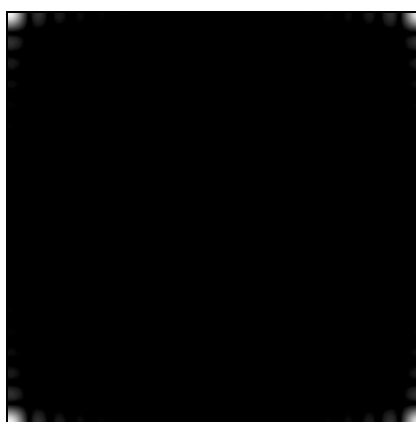
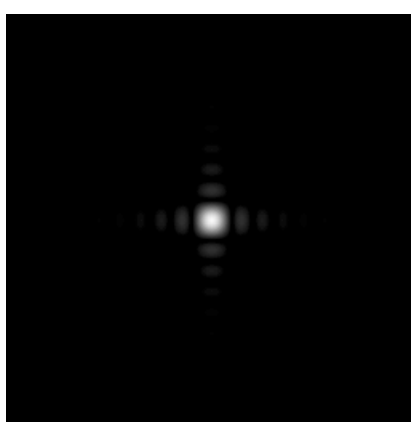


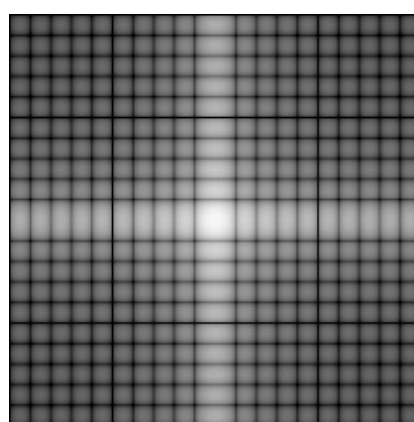
Imagem Original



Espectro  $F_{im}$



Espectro  $F_{im}$  deslocado



Espectro  $F_{log}$  deslocado

## Alguns Espectros de Fourier (deslocados)

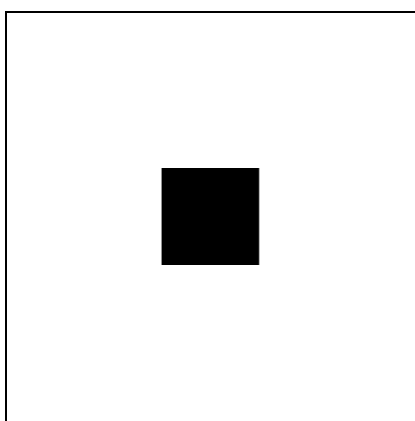
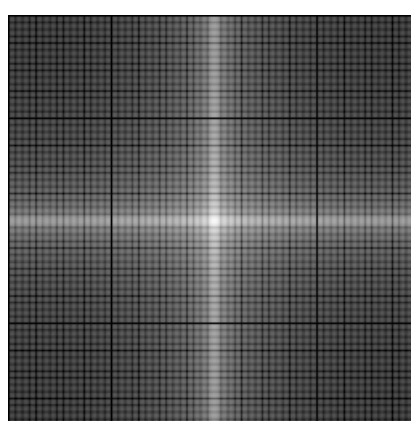


Imagem Quad60x60



Espectro  $F_{log}$  de Quad60x60

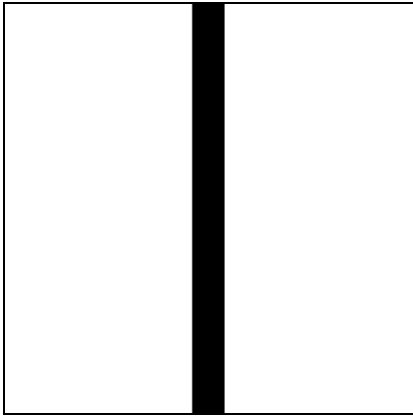


Imagem Retan20x256



Espectro  $F_{log}$  de Retan20x256

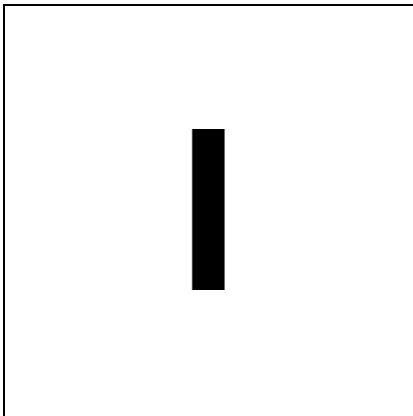
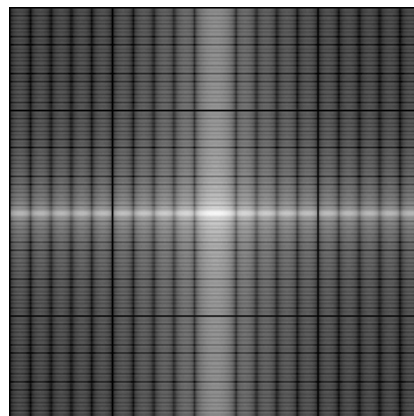


Imagem Retan20x100



Espectro  $F_{log}$  de Retan20x100

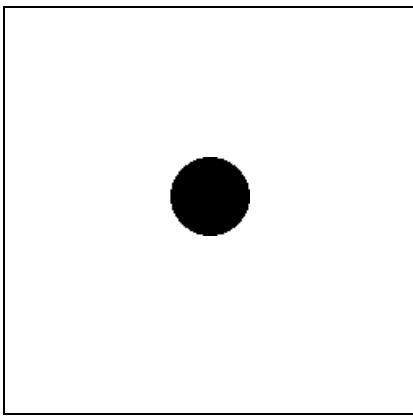
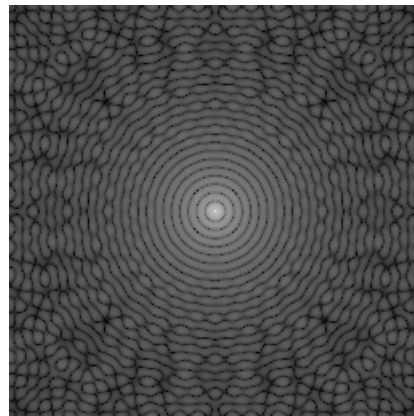


Imagem Bola1



Espectro  $F_{log}$  de Bola1

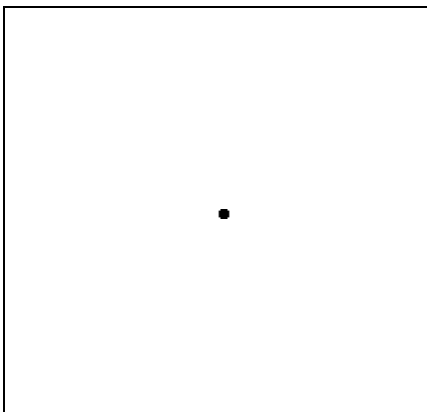
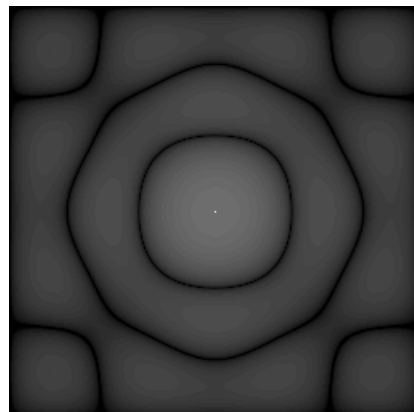


Imagem Bola2



Espectro  $F_{log}$  de Bola2

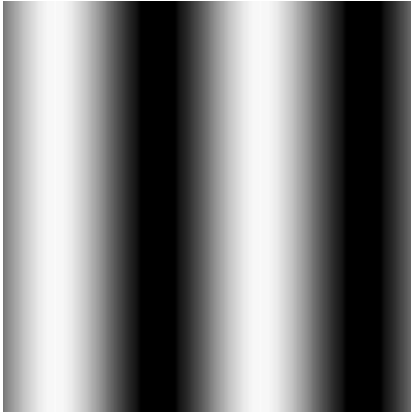
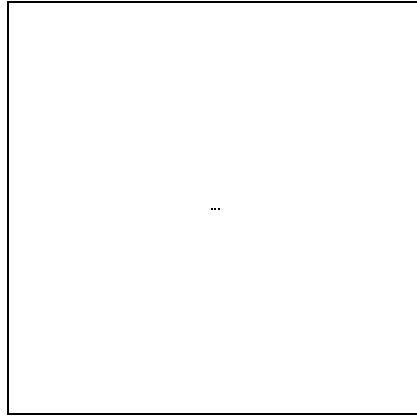


Imagem  $\sin 4\pi$



Negativo de espectro  $F_{log}$  de  $\sin 4\pi$

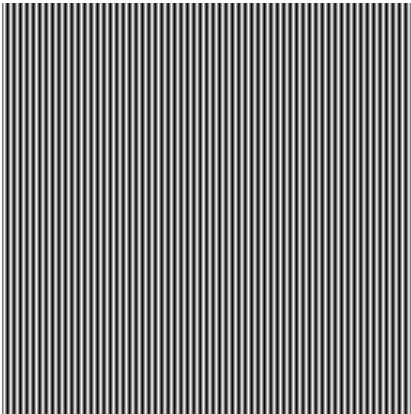
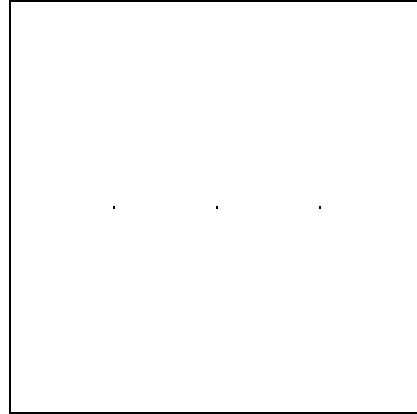


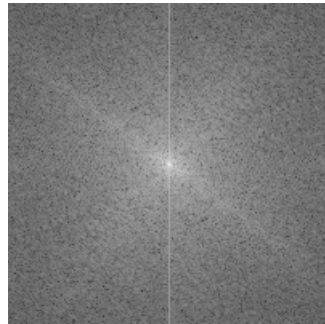
Imagem  $\sin 128\pi$



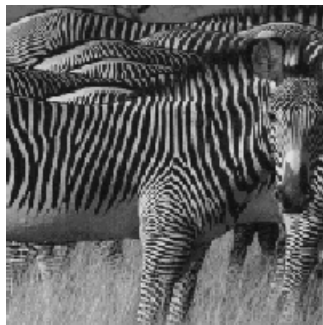
Negativo de espectro  $F_{log}$  de  $\sin 4\pi$



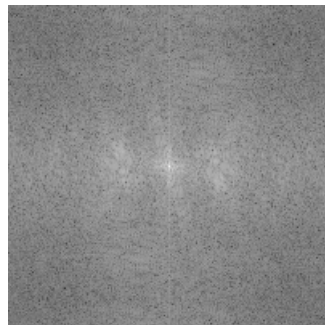
Space



Espectro  $F_{log}$  de Space



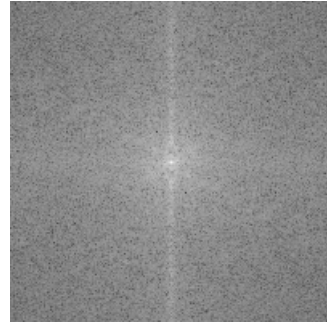
Zebra



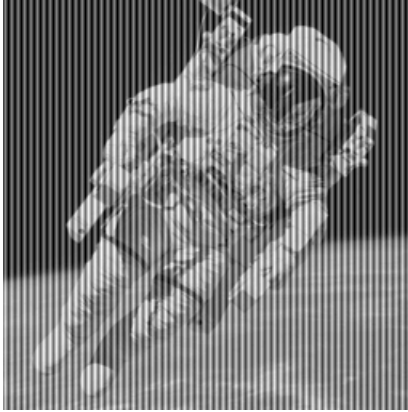
Espectro  $F_{log}$  de Zebra



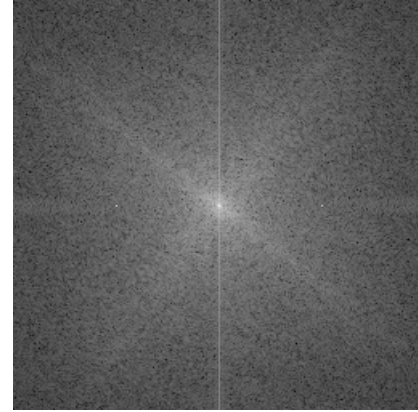
KingTut



Espectro de KingTut



SpaceSen



Espectro de SpaceSen

## 7.4 FILTRAGEM NO DOMÍNIO DA FREQUÊNCIA

Conforme visto em capítulos anteriores, as operações de filtragem no domínio do espaço podem ser implementadas com o uso de vizinhanças deslizantes, também denominadas máscaras convolucionais ou resposta ao impulso. Se  $G(u,v)$  é a DFT de  $g(m,n)$ ,  $F(u,v)$  é a DFT de  $s[m,n]$  e  $H[u,v]$  é a DFT de  $h[m,n]$ , pode-se mostrar que, se  $g[m,n] = s[m,n] \otimes h[m,n]$ , então  $G[u,v] = H[u,v]F[u,v]$ . Este resultado é conhecido como *teorema da convolução*, e a transformada  $H[u,v]$  é denominada função de transferência do filtro. A partir do teorema da convolução, definem-se todos os tipos fundamentais de filtros no domínio da frequência, apresentados a seguir.

### 7.4.1 Filtros Passa-Baixas

Um filtro passa-baixas apresenta uma função de transferência  $H[u,v]$  que atenua as componentes de alta frequência de  $F[u,v]$  e preserva suas componentes de baixa frequência. Seja

$$D(u,v) = \sqrt{u^2 + v^2}$$

Um filtro passa-baixas bidimensional ideal (ILPF) possui a função de transferência.

$$H[u,v] = \begin{cases} 1 & \text{se } D(u,v) \leq D_0 \\ 0 & \text{caso contrário} \end{cases}$$

$D_0$  corresponde à *frequência de corte* do filtro.



A filtragem por um ILPF seguida pela transformada inversa resulta em uma imagem suavizada que freqüentemente apresenta um efeito indesejável denominado *ringing*. O *ringing* é um efeito oscilatório causado pela descontinuidade da função de transferência do filtro. Este efeito é praticamente eliminado com o uso de filtros suaves, como o de Butterworth. A função de transferência do filtro passa-baixas de Butterworth (BLPF) de ordem  $n$  e frequência de corte a uma distância  $D_0$  da origem é:

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

Da fórmula acima, percebe-se que  $H(0,0) = 1$  é o maior valor de  $H(u, v)$  e, quando  $D(u, v) = D_0$ ,  $H(u, v) = 0,5$ . Assim,  $D_0$  é a distância da origem onde  $H(u, v)$  apresenta a metade do seu valor máximo.

#### 7.4.2 Filtros Passa-Altas

Um filtro passa-altas apresenta uma função de transferência  $H(u, v)$  que atenua as componentes de baixa frequência de  $F(u, v)$  e preserva suas componentes de alta frequência. Um filtro passa-altas bidimensional ideal (IHPF) possui a função de transferência

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

A função de transferência do filtro passa-altas de Butterworth (BHPF) de ordem  $n$  e frequência de corte a uma distância  $D_0$  da origem é:

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

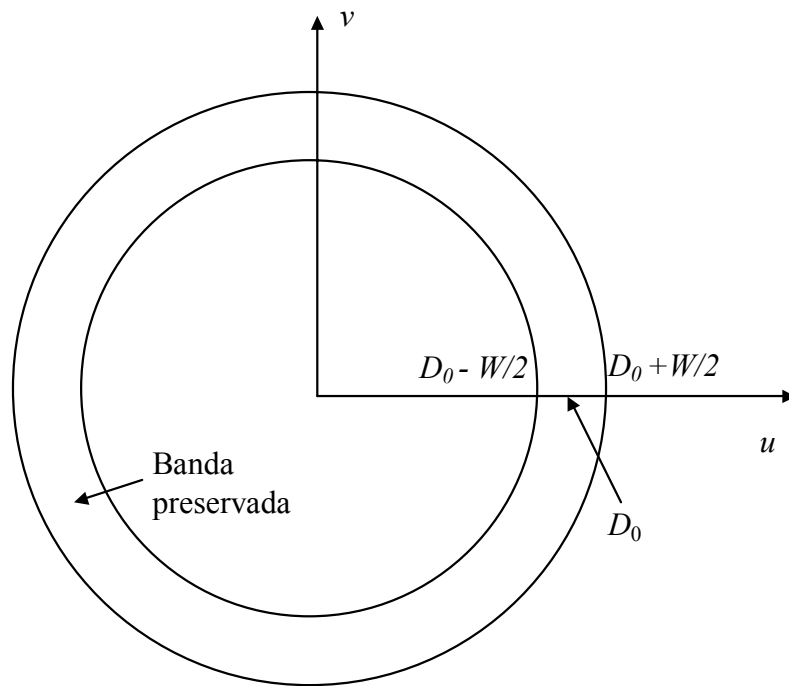
#### 7.4.3 Filtros Passa-Faixa

Um filtro passa-faixa apresenta uma função de transferência  $H(u, v)$  que preserva as componentes de uma faixa de frequências intermediárias de  $F(u, v)$  e atenua as demais. Um filtro passa-faixa bidimensional ideal (IBPF) que elimina todas as componentes de frequência exceto aquelas situadas em uma vizinhança de raio  $D_0$  centrada em um ponto  $(u_0, v_0)$  possui a função de transferência

$$H(u, v) = \begin{cases} 1 & \text{se } \sqrt{(u - u_0)^2 + (v - v_0)^2} \leq D_0 \quad \text{ou} \quad \sqrt{(u + u_0)^2 + (v + v_0)^2} \leq D_0 \\ 0 & \text{caso contrário} \end{cases}$$

Um IBPF bidimensional que elimina todas as componentes de frequência exceto aquelas situadas em uma faixa circular centrada na origem, análoga à da figura seguinte, possui a função de transferência

$$H(u, v) = \begin{cases} 0 & \text{se } \sqrt{u^2 + v^2} < D_0 - W / 2 \\ 1 & \text{se } D_0 - W / 2 \leq \sqrt{u^2 + v^2} \leq D_0 + W / 2 \\ 0 & \text{se } \sqrt{u^2 + v^2} > D_0 + W / 2 \end{cases}$$



#### 7.4.4 Filtros Corta-Faixa

Um filtro corta-faixa ou rejeita-faixa apresenta uma função de transferência  $H(u,v)$  que atenua as componentes de uma faixa de frequências de  $F(u, v)$  e preserva as demais. Os filtros corta-faixa ideais (IBRF) em torno da origem e em torno de um ponto arbitrário são o inverso dos IBPF, e podem ser facilmente obtidos da função de transferência destes.

## Filtros No Domínio da Frequência

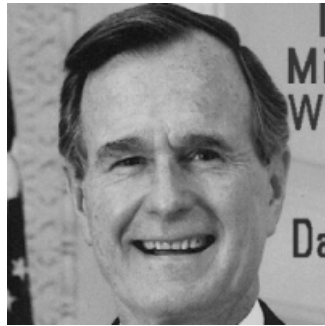
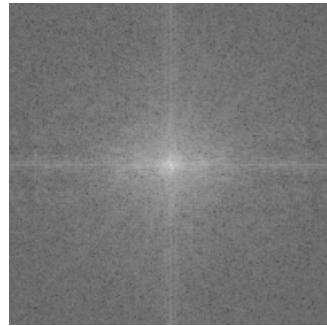
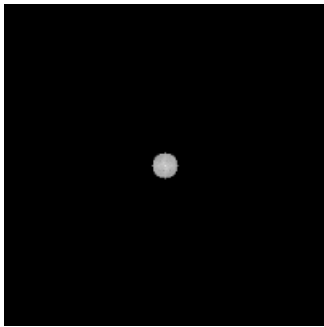


Imagem original



Espectro



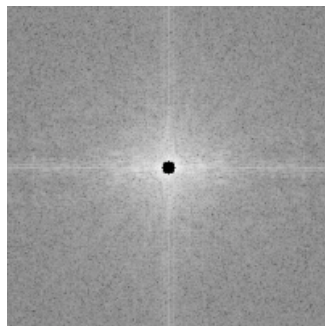
Espectro filtrado, ILPF,  $D_0 = 8$



Imagem filtrada, ILPF,  $D_0 = 8$



Imagem filtrada, ILPF,  $D_0 = 16$



Espectro filtrado, IHPF,  $D_0 = 4$



Imagem filtrada, IHPF,  $D_0 = 4$

## Filtros No Domínio da Frequência

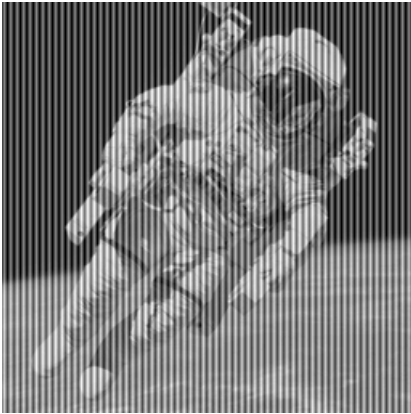
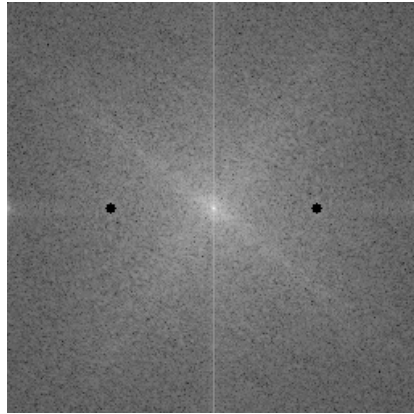


Imagem original



Espectro filtrado, IBRF,  $D_0 = 3$

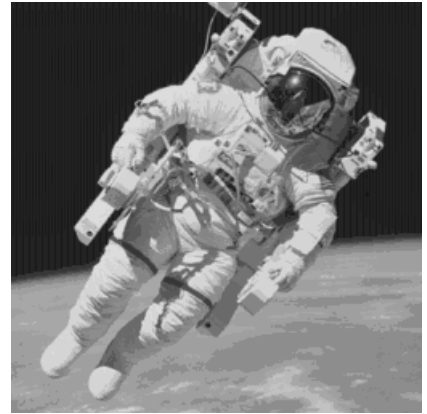
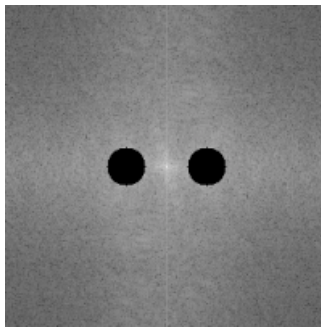


Imagem filtrada, IBRF,  $D_0 = 3$



Imagem original



Espectro filtrado, IBRF

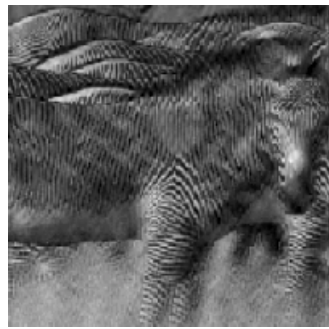


Imagem filtrada