3.4 – DETECÇÃO DE CONTORNOS (EDGE DETECTION)

Um contorno (edge) em uma imagem é um lado ou bordo no qual tenha ocorrido uma mudança significativa em algum aspecto físico da imagem, como na reflectância, iluminação, ou na distância das superfícies visíveis para o observador.

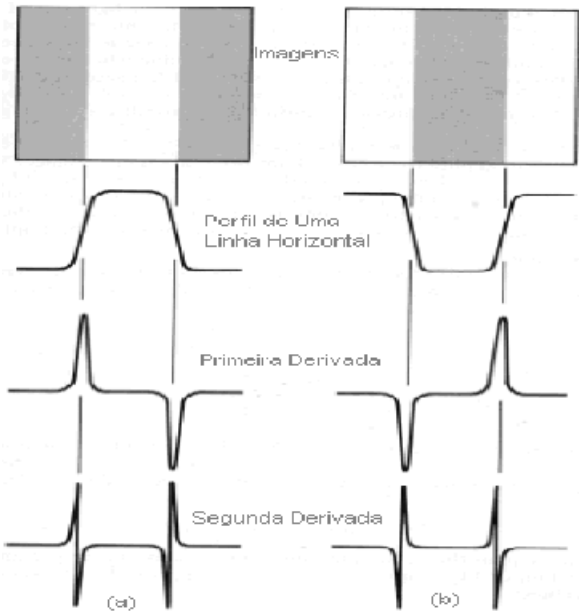
A detecção de contornos é útil em muitas situações, como no entendimento da imagem e na identificação de objetos. Na segmentação da imagem, a detecção de contornos é frequentemente o primeiro passo. Uma forma eficiente de codificação é registrar apenas os contornos detectados da imagem, pois apenas os contornos de uma imagem já permitem sua interpretação.

O significado de uma mudança física na imagem depende da aplicação, o que seria considerado contorno em uma aplicação pode não ser em outra. Para um sistema de identificação de objetos, apenas os contornos principais do objeto seriam suficientes para identificação, e detalhes adicionais dentro do objeto não serão considerados como contornos.

Um contorno só pode ser definido no contexto da aplicação. Apesar disso algoritmos para detecção de contornos úteis em uma ampla gama de aplicações têm sido desenvolvidos, os mais representativos serão discutidos a seguir.

3.4.1 – MÉTODOS BASEADOS NO GRADIENTE

Consideremos uma função $f(x)$ que represente um contorno em uma dimensão, no ponto x_o. Uma maneira de determinar x_o é calcular suas derivadas primeira, $f'(x)$, e segunda, $f''(x)$.



O valor de x_o pode ser determinado procurando os máximos e mínimos locais de $f''(x)$. Veremos nesta seção métodos que buscam x_o em $f'(x)$ e na próxima métodos que usam a derivada segunda.

Além de determinar a localização de um ponto do contorno, $f'(x)$ também pode ser usada para estimar a intensidade e a direção do contorno. Se $\mid f'(x) \mid$ for muito grande, $f(x)$ está variando muito rapidamente. Se $f'(x)$ for positiva, $f(x)$ está aumentando. Se $f'(x)$ for negativa, $f(x)$ está diminuindo.

Usando estas observações, uma forma de detectar contornos é usar o esquema:

$f(x)$
$\downarrow \frac{d.}{dx}$
$f'(x)$
$\downarrow \mid . \mid$
$\mid f'(x) \mid$
$\downarrow$
it is $>$ threshold em $x = x_o$? $\rightarrow x_o$ não é contorno
$\downarrow$ sim
$\mid f'(x) \mid$ é um máximo local ? $\rightarrow$ x_o não é contorno
$\downarrow$ sim
x_o é um ponto de um contorno

Se o módulo de $f'(x)$ for maior que determinado valor (thershold), x é um candidato a contorno. Se todos os valores de x forem maiores que o dado valor, o contorno será uma faixa e não um ponto. Para evitar isso usa-se o requerimento de $\mid f'(x) \mid$ ser um máximo local nos pontos dos contornos. A escolha do valor de threshold depende da aplicação. Se o threshold aumentar apenas os valores de x, em que $f(x)$ varia rapidamente serão registrados como candidatos a contornos. Geralmente o threshold ótimo é escolhido por tentativa e erro, também é possível uma escolha adaptativa.

A generalização de $f'(x)$ para uma função bidimensional, $f(x, y)$ é dada pelo gradiente:

$$\nabla f(x, y) = \frac{\partial f(x, y)}{\partial x}\vec{i}_x + \frac{\partial f(x, y)}{\partial y}\vec{i}_y$$

onde $\vec{i}_x$ é o vetor unitário na direção x e $\vec{i}_y$ é o vetor unitário na direção y. A generalização 2D do esquema anterior é:

$f(x,y)$
$\downarrow \frac{d.}{dx} + \frac{d.}{dy}$
$\nabla\ f(x,y)$
$\downarrow\ |\ .\ |$
$\downarrow\ |\ \nabla\ f(x,y)\ |$
it is $>$ threshold em $x = x_o$? $\rightarrow (x_o, y_o)$ não é contorno
$\downarrow\ sim$
se nescessário use um processo de redução de contornos
$\downarrow$
mapeamento dos contornos

O valor de $|\ \nabla f(x,y)\ |$ é calculado e comparado com um valor de threshold para determinar os candidatos a contornos. Se todos os valores de (x,y) tais que $|\ \nabla f(x,y)\ |$ sejam maiores que o valor de threshold forem detectados como contornos, os contornos aparecerão como faixas e não contornos. O processo de determinação de uma linha de contorno de uma lista de candidatos a pontos de contorno é chamado edge thinning. Um algoritmo simples de edge thinning é checar se $|\ \nabla f(x,y)\ |$ é um máximo local em pelo menos uma direção (esta propriedade é geralmente checada em algumas direções especificadas, ou pelo menos apenas na vertical e horizontal). Se $|\ \nabla f(x,y)\ |$ for um máximo local ao longo de uma determinada direção, o contorno em potencial será considerado um edge point. Uma desvantagem com estes algoritmos simples de edge thinning é que eles criam várias linhas de contorno menores falsas na vizinhança das linha de contorno maiores.

Um método simples de remover a maioria destas linhas de contorno falsos é impor como restrição adicional que:

(a) Se $|\ \nabla f(x,y)\ |$ tem um máximo local em (x_o, y_o) na direção horizontal, mas não na direção vertical, (x_o, y_o) é um ponto de contorno quando:
$\left|\frac{\partial f(x,y)}{\partial x}\right|_{em\ x_o, y_o} > K \left|\frac{\partial f(x,y)}{\partial y}\right|_{em\ x_o, y_o}$, K é uma constante.

(b) Se $|\ \nabla f(x,y)\ |$ tem um máximo local em (x_o, y_o) na direção vertical, mas não na direção horizontal, (x_o, y_o) é um ponto de contorno quando:
$\left|\frac{\partial f(x,y)}{\partial y}\right|_{em\ x_o, y_o} > K \left|\frac{\partial f(x,y)}{\partial x}\right|_{em\ x_o, y_o}$, onde K é escolhido geralmente em torno de 2.

Estas restrições requerem que a variação da razão da intensidade em uma das direções seja significativamente maior que na outra direção.

Um sistema de detecção de contornos, que usa uma função $\mid \nabla f(x,y) \mid$, é chamado nondirecional detector se a função não tiver tendência em uma direção particular. Quando $\mid \nabla f(x,y) \mid$ apresenta tendências é chamado directional detector. Por exemplo usando $\mid \frac{\partial f(x,y)}{\partial x} \mid$ ao invés de $\mid \nabla f(x,y) \mid$ o sistema detectará contornos na direção vertical mas não na horizontal.

Nos sistemas 2D as derivadas parciais $\frac{\partial f}{\partial x}(x,y)$ e $\frac{\partial f}{\partial y}(x,y)$ são substituidas por algumas formas de diferenças. Por exemplo geralmente substitui-se $\frac{\partial f}{\partial x}$ por

$$\frac{f(n_1,n_2) - f(n_1-1,n_2)}{T}$$

onde $f(n_1,n_2) - f(n_1-1,n_2)$ indica *o ponto* menos *o seu vizinho anterior horizontal* ou,

$$\frac{f(n_1+1,n_2) - f(n_1,n_2)}{T}$$

ou ainda:

$$\frac{f(n_1+1,n_2) - f(n_1-1,n_2)}{2T}$$

Como estas derivadas serão comparadas com um threshold que pode ser ajustado os fatores de escala, $1/T$ e $1/2T$, são omitidos. Usualmente diversas expressões são testados até se encontrar a mais adequada, na estimativa de $\partial f(x,y)/\partial x$. Para ter-se valores médios melhores geralmente usa-se:

$$f(n_1+1,n_2+1) - f(n_1-1,n_2+1) + f(n_1+1,n_2) - f(n_1-1,n_2) +$$
$$+ f(n_1+1,n_2-1) - f(n_1-1,n_2-1) = \frac{\partial f(x,y)}{\partial x}$$

ou

$$f(n_1+1,n_2+1) - f(n_1-1,n_2+1) + 2f(n_1+1,n_2) - f(n_1-1,n_2) +$$
$$+ f(n_1+1,n_2-1) - f(n_1-1,n_2-1) = \frac{\partial f(x,y)}{\partial x}$$

Estas operações podem ser vistas como convoluções de $f(n_1,n_2)$ com $h(n_1,n_2)$. Exemplo de filtros h direcionais que fazem estas operações são:

- para contornos verticais

-1		1
-1		1
-1		1

$\frac{\partial}{\partial x} f(x,y) = h^* f$

- para contornos horizontais

1	1	1
-1	-1	-1

$\frac{\partial f}{\partial y}(x,y) = h^* f$

- para contornos a + 45°

1	1	
1		-1
	-1	-1

- para contornos a - 45°

	1	1
-1		1
-1	-1	

Os filtros diagonais acima correspondem aos ∇f anteriores rotacionado de 45°.

Filtros não direcionais, podem ser desenvolvidos por aproximações da fórmula:

$$\mid \nabla f(x,y) \mid = \sqrt{\left(\frac{\partial f(x,y)}{\partial x}\right)^2 + \left(\frac{\partial f(x,y)}{\partial y}\right)^2}$$

que combinam os termos usados nos detectores direcionais. Um exemplo de aproximação discreta da fórmula acima é dado por:

$$\mid \nabla f(x,y) \mid \Rightarrow \sqrt{f_x(n_1,n_2)^2 + f_y(n_1,n_2)^2}$$

onde

$$f_x(n_1,n_2) = f(n_1,n_2) * h_x(n_1,n_2)$$

e

$$f_y(n_1,n_2) = f(n_1,n_2) * h_y(n_1,n_2)$$

onde * indica convolução, h_x e h_y são os filtros de Kernel:

-1		1
-2		2
-1		1

h_x

1	2	1
-1	-2	-1

h_y

O método que usa estes filtros é denominado Sobel. O método de Roberts usa como h_x e h_y os kernels:

	1
-1	

h_x

1	
	-1

h_y

Existem muitas variações possíveis destes métodos de detecção de contorno. Pode-se variar as expressões usadas para representar $\nabla f(x,y)$ e também usa-se diversas formas de

edge thinning. Os diversos métodos também podem ser "melhorados" de diversas maneiras. Geralmente as formas de obter o gradiente ou diferencial são muito sensíveis ao ruido sendo muito importante a redução de ruidos em muitos casos. Pontos detectados como contornos isoladamente podem ser removidos usando processos de mapeamento de contornos. Os métodos baseados no gradiente podem causar "descontinuidades" nos contornos detectados. Sendo importante o desenvolimento de formas que imponham continuidade aos contornos.

3.4.2 – MÉTODOS BASEADOS NO LAPLACIANO

O objetivo dos algoritmos de detecção de contornos é localizar as regiões onde a intensidade está variando rapidamente. Nas funções 1D, a busca a regiões de mudanças rápidas de intensidade corresponde a busca de regiões de $f'(x)$ altos. Nos métodos baseados no gradiente, $f'(x)$ é considerado *alto* quando sua magnitude, $\mid f'(x) \mid$, for maior que um certo threshold. Outra forma de assumir que $f'(x)$ é grande é procurar o máximo local, ou seja onde $f''(x) = 0$. Considerar $f''(x) = 0$ como ponto de contorno faz com que muitos pontos sejam declarados contorno. Desde que não existe chegagem na magnitude de $f'(x)$ qualquer ondulação em $f(x)$ é suficiente para gerar um contorno. Devido a sensibilidade ao ruido, a aplicação de um sistema de redução de ruido antes da utilização do esquema de detecção é muito importante no processamento de imagens com ruido de fundo.

Para generalizar a derivada segunda em 2D usa-se a função Laplaciano:

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

no caso de funções 2D discretas: $f(n_1, n_2)$, as derivadas segundas, $\partial^2 f/\partial x^2$ e $\partial^2 f/\partial y^2$, podem ser substituídas por diferenças de segunda ordem. Estas diferenças podem ser representadas pela convolução de $f(n_1, n_2)$ com um filtro $h(n_1, n_2)$ adequado. Exemplos destes filtros são:

	1	
1	-4	1
	1	

1	1	1
1	-8	1
1	1	1

-1	2	-1
2	-4	2
-1	2	-1

Considerando os pixels vizinhos de $f(n_1, n_2)$:

	n_1, n_{2-1}	
n_{1-1}, n_2	n_1, n_2	n_{1+1}, n_2
	n_1, n_{2+1}	

para mostrar que $f(n_1, n_2) * h(n_1, n_2)$ pode ser considerada como uma aproximação discreta de $\nabla^2 f(x, y)$, vamos considerar o primeiro destes filtros. Aproximando a primeira derivada pela diferença progressiva;

$$\frac{\partial f}{\partial x}(x, y) \;\rightarrow\; f_x(n_1, n_2) = f(n_{1+1}, n_2) - f(n_1, n_2)$$

e a segunda pela diferença regressiva

$$\frac{\partial^2 f}{\partial x^2}(x, y) \;\rightarrow\; f_{xx}(n_1, n_2) = f_x(n_1, n_2) - f_x(n_{1-1}, n_2)$$

temos, substituindo f_x na expressão acima:

$$\frac{\partial^2 f}{\partial x^2}(x, y) \;\rightarrow\; f_{xx}(n_1, n_2) = \Big[f(n_{1+1}, n_2) - f(n_1, n_2)] - [f(n_1, n_2) - f(n_{1-1}, n_2) \Big] = \\ = f(n_{1+1}, n_2) - 2f(n_1, n_2) + f(n_{1-1}, n_2)$$

De maneira similar:

$$\frac{\partial^2 f}{\partial y^2}(x, y) \;\rightarrow\; f_{yy}(n_1, n_2) = f(n_1, n_{2+1}) - 2f(n_1, n_2) + f(n_1, n_{2-1})$$

assim,

$$\nabla^2 f(x, y) \;\rightarrow\; \nabla^2 f(n_1, n_2) = f_{xx}(n_1, n_2) + f_{yy}(n_1, n_2) = f(n_{1+1}, n_2) + \\ f(n_{1-1}, n_2) - 4f(n_1, n_2) + f(n_1, n_{2-1}) + f(n_1, n_{2+1})$$

Os demais kernels dos filtros, dependem de como a derivada segunda é aproximada. Depois de encontrar-se $\nabla^2 f(n_1, n_2)$ procura-se pelos pontos em que $\nabla^2 f(n_1, n_2)$, for nula. Como estes contornos são limites de regiões, tendem a ser uma linha contínua. A busca por contornos contínuos, importante no caso de métodos baseados no gradiente, não é necessária no caso dos métodos baseados no Laplaciano. No entanto a consideração de todos os pontos em que $\nabla^2 f(n_1, n_2) = 0$ são contornos, tende a gerar um número muito grande de contornos. Estes falsos contornos aparecem em regiões de variações locais pequenas nas imagens. Como um exemplo, considere uma região de fundo uniforme tal que $f(n_1, n_2)$ seja constante, nesta região $\nabla^2 f(n_1, n_2)$ será zero. Se o contorno for obtido pela variação de sinal de $\nabla^2 f(n_1, n_2)$. Qualquer pequena variação de $f(n_1, n_2)$ poderá causar a detecção de um falso contorno. Uma forma de remover estes falsos contornos é requerer que a variância local seja suficientemente grande em um ponto de contorno.

A variância local $\sigma_f^2(n_1, n_2)$ pode ser estimada por:

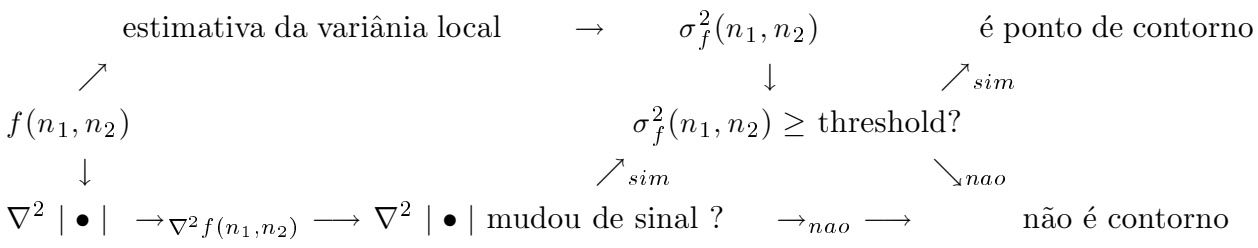
$$\sigma_f^2(n_1,n_2) = \frac{1}{(2M+1)^2} \sum_{k_1=n_1-M}^{n_1+M} \sum_{k_2=n_2-M}^{n_2+M} \Big[f(k_1,k_2) - m_f(k_1,k_2) \Big]^2$$

onde:

$$m_f(n_1,n_2) = \frac{1}{(2M+1)^2} \sum_{k_1=n_1-M}^{n_1+M} \sum_{k_2=n_2-M}^{n_2+M} f(k_1,k_2)$$

sendo M escolhida tipicamente em torno de 2.

O esquema abaixo representa um método de detecção contornos baseado no Laplaciano, usando variância local.



Como no cálculo da variância, $\sigma_f^2(n_1,n_2)$, será comparada com um valor de threshold o fator de escala $1/(2M+1)^2$ pode ser eliminado. Este cálculo pode ser apenas executado nos pontos onde $\nabla^2 f(n_1,n_2)$ for nulo.

3.4.3 – MÉTODO DE MARR E HILDRETH PARA DETECÇÃO DE CONTORNO

Nas 2 seções anteriores, discutimos algoritmos de detecção de contornos que mapeiam os contornos a partir de uma imagem de entrada. Marr e Hildreth (em "Theory of edge detection", Proc. Royal Soc., London, vol. B 207, 1980 pp. 187-217) observaram que ocorrem variações significativas na intensidade com as variações no grau de definição e resolução dos detalhes. Por exemplo, regiões com sombras fora de foco e regiões com foco preciso e perfeita definição de detalhes podem estar presentes na mesma imagem. Uma detecção ótima de contornos precisaria de operadores que funcionam em diferentes escalas de precisão. Marr e Hildreth sugerem que a imagem original tenha sua freqüência limitada em diferentes pontos de corte e que algoritmos de detecção de contornos sejam aplicados a estas imagens pré-processadas. O resultado será um mapeamento de

contornos em diferentes escalas. Estes mapas conteriam informações importantes sobre diferentes parâmetros. A imagem visual compõem-se de contornos, linhas e sombras, em localizações específicas. Ocorrem mudanças nestes elementos quando a iluminação e a reflectância variam. Se o mesmo contorno aparece quando diferentes aspectos são variados, corresponde à variação de intensidade devido a um único fenômeno físico independente das variáveis que produzem a variação de intensidade dentro da mesma região.

Para limitar uma imagem em diferentes freqüências de corte, é usado um filtro com forma Gaussiana, dado por:

$$h(x,y) = e^{\frac{-(x^2+y^2)}{2\pi\sigma^2}}$$

onde σ determina a freqüência de corte. σ alto corresponde a baixa freqüência de corte.

A forma Gaussiana é escolhida pela sua "suavidade", um filtro suave provavelmente não introduzirá mudanças na forma original; não modificando a localização dos contornos. Depois da imagem ser regulada pelo filtro, os contornos podem ser detectados usando um dos algoritmos de detecção de contorno das seções anteriores. A operação derivada parcial dos algoritmos das seções anteriores e o filtro acima podem ser combinados. Por exemplo, observando que $\nabla^2[\bullet]$ e a operação convolução são lineares, pode-se escrever:

$$\nabla^2(f(x,y) * h(x,y)) = f(x,y) * [\nabla^2(h(x,y)] = f(x,y) * [\frac{\partial^2 h(x,y)}{\partial x^2} + \frac{\partial^2 h(x,y)}{\partial y^2}]$$

Se a função $h(x,y)$ for a expressão Gaussiana anterior $\nabla^2 h(x,y)$ é dado por:

$$\nabla^2 h(x,y) = \frac{e^{\frac{-(x^2+y^2)}{2\pi\sigma^2}}}{\pi^2\sigma^4}(x^2+y^2-2\pi\sigma^2)$$

No trabalho citado de Marr e Hildreth, os contornos são encontrados, por simplicidade, considerando os pontos mudança de sinal de $\nabla^2 f(x,y)$. Limitar a banda de $f(x,y)$ tende a reduzir ruidos, reduzindo os problemas associados a sensibilidade ao ruido. O cálculo de $f(x,y) * [\nabla^2 h(x,y)]$ é equivalente a uma filtragem bandpass em $f(x,y)$ onde σ^2 é o parâmetro que controla a largura de banda que o filtro permitirá passar (valores muito comuns são 4,16,32). No caso discreto (x,y) nas expressões acima são simplesmente substituídos por (n_1,n_2).

3.4.4 – DETECÇÃO DE CONTORNOS BASEADA EM MODELOS

Os algorimos, discutidos até aqui são métodos gerais, desenvolvidos independentemente de qualquer aplicação. Um método alternativo consiste no desenvolvimento de algoritmos para uma aplicação particular. Se a forma do contorno for conhecida, esta pode

ser incorporada ao algoritmo. Para ilustrar o desenvolvimento da detecção de contornos para uma aplicação específica, vamos considerar o problema de detecção dos contornos das **artérias** coronárias de um **angiograma**.

As coronárias são artérias que circundam o coração e fornecem sangue ao músculo cardíaco. Estreitamento destas veias impedem que o volume adequado de sangue chegue ao coração, causando dor e dano à circulação. Este dano é chamado doença coronariana. Para determinar a severidade do estreitamento é usado um angiograma. Um angiograma é um raio x das artérias obtido depois da aplicação de um agente contrastante (geralmente iodo) injetado nas veias. O contraste é injetado diretamento nas artérias por um cateter a fim de alcançar altas concentrações. Diferentes observadores fazem avaliações visuais de um angiograma com variações bastante grande da severidade da doença. A medida mais comum da obstrução é a **percentagem da estenose**, que é definida como a percentagem máxima do estreitamento da artéria em determinado comprimento do vaso. Um método de **estimar esta estenose** começa por determinar o contorno dos vasos no angiograma.

Um modelo razoável de um angiograma $f(n_1, n_2)$ é dado por:

$f(n_1, n_2) = (v(n_1, n_2) + p(n_1, n_2))^*(g(n_1, n_2) + w(n_1, n_2))$

onde $v(n_1, n_2)$ representa o vaso, $p(n_1, n_2)$ representa o fundo, $g(n_1, n_2)$ o $blurring$ e $w(n_1, n_2)$ o ruído de fundo.

A função vaso, $v(n_1, n_2)$, é derivada de um modelo de cone generalizado em 3D que é contínuo e com seção transversal elíptica. É escolhida a forma elíptica por envolver poucos parâmetros e por fornecer boa estimativa da percentagem da estenose. A seção transversal 1D de $v(n_1, n_2)$, que consiste de um vaso sangüíneo é especificada por 3 parâmetros: 2 deles representam contorno do vaso e um $3^{\underline{o}}$ que está relacionado com o coeficiente de atenuação do raio x no iodo (contraste). A continuidade do vaso é garantida pela representação do vaso por função **spline cúbica**.

O fundo, $p(n_1, n_2)$, é modelado por um polinômio 2D de baixa ordem. Polinômios de ordens baixas são contínuos e sua escolha é motivada pela observação de que objetos no fundo, como tecidos e ossos são muito maiores que os vasos.

A função de nebulosidade, $g(n_1, n_2)$, é modelada por uma função contínua gaussiana 2D que considera a nebulosidade introduzida em vários estágios do processamento de imagens.

O ruido, $w(n_1, n_2)$, é randômico e suposto branco. Desta maneira o modelo é caracterizado por parâmetros do vaso, coeficientes de polinômio e variância do ruido de fundo.

Os parâmetros usados no modelo podem ser estimados por diversos procedimentos. Um destes é o método de estimativa de parâmetros por probabilidade máxima (discutido em LIM95 seção 6.1.5), neste método os parâmetros desconhecidos são estimados maxi-

mizando a função densidade de probabilidade da função $f(n_1, n_2)$ em relação ao parâmetro desconhecido.

O desenvolvimento de algoritmos de detecção de contorno para uma aplicação específica, como o deste exemplo, é mais complexo que os casos gerais discutidos, mas pode apresentar resultados muito melhores.

3.6 – INTERPOLAÇÃO DE IMAGENS

Novas imagens podem ser criadas a partir de imagens existentes por **interpolação temporal ou espacial**. Interpolação espacial é muito usada em ampliações (zooms) de imagens.

Interpolação temporal é usada em animações, filmes e sequencias com movimentos, para aplicações como restauração, análise de imagens, estimativa de movimento e outras. Esta requer muita área de armazenamento. Um novo frame é geralmente criado a partir de 2 frames adjacentes: frame anterior e próximo.

O método mais simples usado na prática, é a interpolação de ordem zero; onde um novo frame é criado pela repetição do frame mais próximo. Um exemplo disso é a passagem de animações de computador (24 frames/sec) para televisão (NTSC 60 fields/sec), onde são criados 3 fields idênticos consecutivos para cada frame anterior e 2 fields consecutivos são criados a partir do próximo frame.

Este processo é repetido para todo o filme, e conhecido como método 3.2 pull-down. Para a maioria das cenas onde não há muito movimento global o resultado é muito bom. Se houver muito movimento global esta interpolação de ordem zero produz "jerk" (choques).

Uma forma de melhorar a interpolação temporal é através da compensação do movimento. Uma retransmissão de TV ou uma cena em movimento é uma seqüência de frames parados apresentados em rápidas sucessões. A velocidade necessária para dar uma idéia adequada do movimento faz com que muitas informações sejam retundantes entre frames adjacentes. A maior parcela da variação da intensidade de um frame para o próximo é devida ao movimento de objetos. O processo de determinação do movimento dos objetos em uma seqüência de frames é chamado de estimativa de movimento.

3.7 – ESTIMATIVA DE MOVIMENTO

O processamento de imagens com consideração da presença do movimento é chamado de compensação do movimento (motion-compensated image processing), e tem diversas

aplicações. Uma destas aplicações é na interpolação de imagens. Estimando parâmetros de movimento, pode-se criar um novo frame entre 2 frames adjacentes. Na próxima seção as aplicações da compensação do movimento para interpolação de imagens são tratadas.

Aplicações importantes estão ligados à restauração de imagens: se os parâmetros de movimento puderem ser estimados e regiões idênticas puderem ser identificados em diferentes frames onde a intensidade das imagens era esperada ser semelhante, pode-se fazer uma **filtragem temporal destas regiões**.

A compensação do movimento também pode ser aplicado na codificação de imagens; a codificação pode ser limitada às diferenças de intensidades entre os frames. Pode-se também descartar alguns frames na codificação e depois reconstitui-los por interpolação.

Para exemplificar veremos como estimar o movimento de translação de objetos. Suponha que $f(x, y, t_{-1})$ e $f(x, y, t_o)$ denotam a intensidade de 2 imagens no tempo t_{-1} e t_o respectivamente. t_1 indica o frame anterior e t_o o frame atual. Vamos supor que:

$$f(x, y, t_o) = f(x \pm dx, y \pm dy, t_{-1}) \tag{0}$$

onde dx e dy são os deslocamentos horizontais e verticais entre t_{-1} e t_o.

Se assumirmos um movimento uniforme entre t_{-1} e t_0, com velocidade v_x e v_y na horizontal e vertical, respectivamente, podemos escrever para um tempo t intermediário.

$$f(x, y, t) = f\big(x \pm v_x(t - t_{-1}), y \pm v_y(t - t_{-1}), t\big)$$

onde $v_x = \frac{\partial f}{\partial x}(x, y, t)$, $v_y = \frac{\partial f}{\partial y}(x, y, t)$ é válida na região espaço/tempo na qual pudermos assumir movimento de translação uniforme. Para derivar uma relação que corresponde a existência de translação uniforme vamos usar a anotação:

$$S(x, y) = f(x, y, t_{-1})$$

assim

$$f(x, y, t) = S\big(\alpha(x, y, t), \beta(x, y, t)\big) \qquad t_{-1} \leq t \leq t_o$$

onde

$$\alpha(x, y, t) = x \pm v_x(t - t_{-1})$$
$$\beta(x, y, t) = y \pm v_y(t - t_{-1})$$

supondo que existam $\frac{\partial f(x,y,t)}{\partial x}$, $\frac{\partial f(x,y,t)}{\partial y}$ e $\frac{\partial f(x,y,t)}{\partial t}$ teremos:

$$\frac{\partial f(x,y,t)}{\partial x} = \frac{\partial S}{\partial \alpha}\frac{\partial \alpha}{\partial x} + \frac{\partial S}{\partial \beta}\frac{\partial \beta}{\partial x} = \frac{\partial S}{\partial \alpha} \tag{1}$$

$$\frac{\partial f(x,y,t)}{\partial y} = \frac{\partial S}{\partial \alpha}\frac{\partial \alpha}{\partial y} + \frac{\partial S}{\partial \beta}\frac{\partial \beta}{\partial y} = \frac{\partial S}{\partial \beta} \tag{2}$$

$$\frac{\partial f(x,y,t)}{\partial t} = \frac{\partial S}{\partial \alpha}\frac{\partial \alpha}{\partial t} + \frac{\partial S}{\partial \beta}\frac{\partial \beta}{\partial t} = \pm v_x \frac{\partial S}{\partial \alpha} \pm v_y \frac{\partial S}{\partial \beta} \tag{3}$$

assim substituindo (1) e (2) em (3):

$$\pm v_x \frac{\partial f}{\partial x}(x,y,t) \pm v_y \frac{\partial f}{\partial y}(x,y,t) = \frac{\partial f}{\partial t(x,y,t)}$$

A equação de constrangimento espaço/tempo (spatio-temporal constraint equation) pode ser escrita como:

$$v_x \frac{\partial f}{\partial x}(x,y,t) + v_y \frac{\partial f}{\partial y}(x,y,t) \pm \frac{\partial f}{\partial t}(x,y,t) = 0 \tag{4}$$

esta equação pode ser generalizada para incorporar outros tipos de movimento, como zoom.

A hipótese de translação simples que resultou na equação (0) e a hipótese adicional de translação com velocidade uniforme que resultou na equação (4) é bastante restritiva. O objeto não pode girar, se aproximar da camera (zoom), ou ter múltiplos objetos com diferentes v_x e v_y. No entanto assumindo movimento de translação uniforme, cubrimos uma parcela sinificativa do movimento típico em seqüência de imagens.

Os métodos de estimativa de movimento podem ser classificados em 2 grupos:
(1) Equiparação de Regiões – region matching methods e
(2) Constrangimento Espaço/Tempo, spatio-temporal constraint methods.
O primeiro baseia-se na equação (0) e o segundo na equação (4).

3.7.1.1 – MÉTODOS DE EQUIPARAÇÃO DE REGIÕES

Estes métodos envolvem a consideração de uma pequena região no frame e a procura por um deslocamento que produziria melhor "casamento" entre as possíveis regiões em frames adjacentes. Neste método o vetor deslocamento (d_x, d_y), é estimado pela minimalização da função:

$$\text{erro} = \int\int_{(x,y)\in R} C\Big[f(x,y,t_o), f(x-d_x, y-d_y, t_1)\Big]dxdy \tag{5}$$

onde R é a região espacial local usada para estimar (d_x, d_y) e $C[.,.]$ é uma métrica que índica a quantidade de diferenças entre os 2 argumentos.

A integral (5) pode ser substituída pelo somatório, se $f(x, y, t)$ for amostrada nas variáveis espaciais (x, y). Se for estimado (dx, dy) no tempo t_o, região R é a região espacial local que circunda uma posição espacial particular onde (dx, dy) é estimado.

O tamanho de R é ditado por diversas considerações. Se for muito grande, a hipótese de (dx, dy) aproximadamente constante na região R, pode não ser válida e o erro da expressão requererá mais cálculos. Se a região for muito pequena, a estimativa pode ser muito sensível a ruídos. Uma escolha razoável está entre 5×5 e 7×7 pixeis. Para a escolha de $C[.,.]$ existem muitas possibilidades; as escolhas mais comuns são a diferença em módulo dos 2 argumentos e o quadrado das diferenças. Estas escolhas reduzem (5) a:

$$\text{erro} = \int\int_{x,y \in R} \Big[f(x, y, to) - f(x - d_x, y - d_y, t_{-1}) \Big]^2 dxdy \tag{6}$$

e

$$\text{erro} = \int\int_{x,y \in R} \mid f(x, y, to) - f(x - d_x, y - d_y, t_{-1}) \mid dxdy \tag{7}$$

A função $f(x, y, t_o) - f(x - d_x, y - d_y, t_{-1})$ é chamada de DISPLACEMENT FRAME DIFFERENCE.

Em aplicações típicas da compensação do movimento a performance não é muito sensível a escolhe da métrica: $C[.,.]$. Enquanto a equação (0) for válida os erros (6) e (7) devem ser zero para d_x, d_y corretos. A minimalização deste erro é um problema não linear. Na solução deste problema diversas variações surgiram, que podem ser agrupadas em **block matching** e **métodos recursivos**.

A forma direta de resolver a minimalização é avaliar o erro para cada (d_x, d_y) possível em uma região e escolher o (d_x, d_y) com erro mínimo. Neste método um bloco de pixeis no tempo t_o é "casado" a um bloco no tempo $t - 1$. A expressão precisa ser avaliada muitas vezes sendo custoso em termos de computação. Alguns métodos foram desenvolvidos para reduzir estes cálculos. Em uma das simplificações, assumimos que (d_x, d_y) é constante em um bloco de (7×7) pixeis. Divide-se a imagens em blocos e estimamos (d_x, d_y) de cada bloco. Não é necessário que o tamanho de R e dos blocos seja os mesmo. Outra simplificação limita o espaço de busca a valores inteiros de (d_x, d_y). Esta simplificação reduz o espaço de busca de variáveis contínuas para valores discretos. A limitação a valores inteiros permite determinar $f(n_1 - d_x, n_2 - d_y, t - 1)$, necessária a avaliação da expressão erro sem interpolações. No entanto a estimativa de d_x, d_y se restringirá a valores discretos.

Pode-se reduzir a computação envolvida no **block matching method** ainda mais, usando procedimentos de busca mais eficientes que a força bruta. Um deste é chamado de three-step search. Em um primeiro passo do método, a expressão erro é avaliado em 9 valores de (d_x, d_y) que são marcados por "1". Ou seja supõem-se:

$$(d_x, d_y) = \{(-3,-3),(0,-3),(3,-3),(-3,0),(0,0),(0,3),(-3,+3),(0,3),(3,3)\}$$

Entre estes valores escolhemos (d_x, d_y) com menor erro. Suponha que o menor erro seja $(d_x = -3, d_y = -3)$. Em um segundo passo avaliamos o erro da expressão com oito valores adicionais de

$$(d_x, d_y) = \{(-2,-2),(0,-2),(2,-2),(-2,0),(2,0),(-2,+2),(0,2),(2,2)\}$$

Escolhemos (dx, dy) entre estes 8 valores. E repetimos o procedimento uma vez mais para escolha do (dx e dy) entre

{(-1, -1),(0, -1),(1, -1),(-1, 0),(0, 1),(0, 1),(-1, 1),(0, 1),(0, 1),(1, 1)}.

No final o valor de dx, dy será estimado, pela somatória dos dx e dy escolhido. Este procedimento pode ser generalizado para mais de 3 passos, para aumentar a possibilidade de variação de (d_x, d_y).

Um outro método de busca é primeiro estimar d_x, procurando melhor $(d_{x,0})$. Depois de d_x estimado, $\hat{d}_x, d_y$ é procurado obtendo-se $(\hat{d}_x, \hat{d}_y)$. Podemos melhorar d_x estimado obtendo-se um novo d_x, depois novo d_y. Neste método heurístico a quantidade de cálculo é reduzida , reduzindo-se o número de valores possíveis (d_x, d_y).

3.7.1.2 – MÉTODOS RECURSIVOS

No método anterior estimamos (d_x, d_y) calculando explicitamente o erro para alguns conjuntos de (d_x, d_y) especificados. Um método alternativo é usar algoritmos descendentes como os de **Newton-Raphson, Davidson-Fletcher-Powell** e *Steepest descen* para resolver o problema não linear de minimizar o erro em relação a (d_x, d_y). Nesta classe de algoritmos é usado um procedimento iterativo para melhorar a estimativa em cada iteração.

Suponha que ($\hat{d}_x(x)$ e $\hat{d}_y(x)$) seja a estimativa de movimento na k-ésima iteração, uma estimativa na próxima iteração será:

$$\begin{aligned} \hat{d}_x(k+1) &= \hat{d}_x(k) + u_x(k) \\ \hat{d}_y(k+1) &= \hat{d}_y(k) + u_y(k) \end{aligned} \tag{8}$$

onde u_y e u_x são termos de correção que variam dependendo do método usado. Se for usado o *Steepest descent* esta equação torna-se

$$\begin{aligned} \hat{d}_x(k+1) &= \hat{d}_x(k) - \epsilon \left. \frac{\partial \ erro \ (d_x, d_y)}{\partial dx} \right|_{d_x = \hat{d}_x(k) \quad \hat{d}_y = \hat{d}_y(k)} \\ \hat{d}_y(k+1) &= \hat{d}_y(k) - \epsilon \left. \frac{\partial \ erro \ (d_x, d_y)}{\partial dy} \right|_{d_x = \hat{d}_x(k) \quad \hat{d}_y = \hat{d}_y(k)} \end{aligned} \tag{9}$$

onde ϵ é o tamanho do passo que pode ser ajustado, e erro $(d_x,\ d_y)$ é o mesmo da equação (5) e, uma função de d_x e d_y para dada região R. Métodos recursivos envolvem tipicamente derivadas parciais e tendem a ser sensíveis a presensa de ruído ou detalhes finos na imagem. Suavizar a imagem antes do movimento frequentemente melhora a performance dos métodos recursivos.

Nos métodos recursivos $(d_x,\ d_y)$ não é limitada a valores inteiros e pode ser estimada com precisão em subpixeis. Os termos de correção envolvem tipicamente derivadas parciais do **erro** $\mathbf{(}d_x\mathbf{,}\ d_y\mathbf{)}$, que incluem avaliação de $f(x, y, t_{-1})$ e suas derivadas parciais em um ponto espacial arbitrário.

Na prática $f(x, y, t_{-1})$ é conhecida apenas em $x = n_1\ T_1$ e $y = n_2\ T_2$. Para avaliar as quantidades necessárias em pontos espaciais arbitrários, devemos usar técnicas de interpolação espacial já discutidas.

Nos métodos recursivos, $(d_x,\ d_y)$ é geralmente estimada em cada pixel (picture element) . Na utilização das relações de recursão (eq. 8), $(d\hat{x}(0), d\hat{y}(0))$ são geralmente obtidas de estimativa em pixeis adjacentes na mesma horizontal em linhas adjacentes ou em frames adjacentes. Este método é chamado de estimativa de pixel com recursividade horizontal, vertical ou temporal, respectivamente. Dado $(\hat{d}_x(x), \hat{d}_y(x))$, nós podemos usar relações recursivas em (8) apenas uma vez para um pixel e então passar para o próximo pixel. Alternativamente podemos usar recursão mais de uma vez para uma estimativa mais precisa de $(d_x,\ d_y)$ antes de passarmos para o próximo pixel.

Embora tenhamos classificado os métodos de equiparação de região em block matching e recursivos para ficarmos consistentes com a literatura, as fronteiras de ambos são nebulosas. Escolhendo um (grid) reticulado preciso no qual os erros podem ser avaliados, pode-se estimar $(d_x,\ d_y)$ nos subpixeis com precisão nos métodos block matching.

A procura em tree-step discutida podem também ser vista como um método recursivo já que a estimativa é melhorada iterativamente. A maior desvantagem destes métodos é a quantidade de cálculos nescessários.

3.7.2 – SPATIO-TEMPORAL CONSTRAINTS

Algoritmos desta classe são baseados nas equações de restrição espaço/tempo do movimento. Esta classe de algoritmos é baseado na equação (4):

$$v_x \frac{\partial f(x,y,t)}{\partial x} + v_y \frac{\partial f(x,y,t)}{\partial y} + \frac{\partial f(x,y,t)}{\partial t} = 0$$

que pode ser vista como uma equação linear de 2 parâmetros desconhecidos v_x e v_y. Supondo-se conhcidos $\frac{\partial f}{\partial x}$, $\frac{\partial f}{\partial y}$ e $\frac{\partial f}{\partial t}$ avalia-se $\frac{\partial f}{\partial x}$, $\frac{\partial f}{\partial y}$ e $\frac{\partial f}{\partial t}$ em diversos pontos (x_i, y_i, t_i) para $1 \leq i \leq N$ nos quais v_x e v_y são supostos constantes, obtendo-se o conjunto de equações lineares:

$$v_x \left.\frac{\partial f(x,y,t)}{\partial x}\right|_{(x_i,y_i,t_i)} + v_y \left.\frac{\partial f(x,y,t)}{\partial y}\right|_{(x_i,y_i,t_i)} + \left.\frac{\partial f(x,y,t)}{\partial t}\right|_{(x_i,y_i,t_i)} \cong 0$$

A estimativa da velocidade pode ser obtida minimizando

$$\text{erro} = \sum_{i=1}^{N} \left[v_x \left.\frac{\partial f(x,y,t)}{\partial x}\right|_{(x_i,y_i,t)} + v_y \left.\frac{\partial f(x,y,t)}{\partial y}\right|_{(x_i,v_i,t)} + \left.\frac{\partial f(x,y,t)}{\partial t}\right|_{x_i,y_i,t_i} \right]^2$$

3.7.3 – COMPENSAÇÃO DO MOVIMENTO POR INTERPOLAÇÃO TEMPORAL

Suponha que se tenha dois frames consecutivos $f(n_1, n_2, t_{-1})$ e $f(n_1, n_2, t_0)$ onde $t_{-1} < t < t_0$. A forma mais simples possível é escolher o frame conhecido (t_{-1} ou t_0) mais próximo, temporalmente, do tempo desejado. Neste método o resultado será uma seqüência cheia de "sacudidelas" (jerk) tanto mais notáveis quanto maiores forem o movimentos globais presentes.

Um método alternativo é fazer uma interpolação que compense o movimento usando os algoritmos de estimativa do movimento discutidos. Na interpolação com compensação do movimento, assume-se um modelo de movimento de translação uniforme dentro de

uma região espaço-tempo local. A partir de $f(n_1, n_2, t_{-1})$ e $f(n_1, n_2, t_0)$ nós computamos as velocidades em $f(n_1, n_2, t)$. As velocidades são então projetadas para o frame mais próximo de t, $-t$ ou t_0. Visto que o ponto espacial projetado geralmente não recai na posição original, é necessário interpolação espacial para determinar o frame interpolado. Se a velocidade estimada em um pixel em particular em $f(n_1, n_2, t_{-1})$ não foi considerada suficientemente precisa, a velocidade será suposta zero. Neste caso o valor do pixel interpolado será idêntico ao valor do pixel de mesma localização no quadro original que for mais próximo do tempo desejado t.

Não é possível ilustrar as características da previsão do movimento por motion-compensated frame interpolation usando apenas figuras paradas. A compensação do movimento por interpolação tem sido usado na modificação da velocidade dos frames. Em cenas usuais estas modificações podem produzir vídeos de qualidade comparável ao original, exceto por algumas taxas de velocidades não naturais em ações como caminhar e converçar.