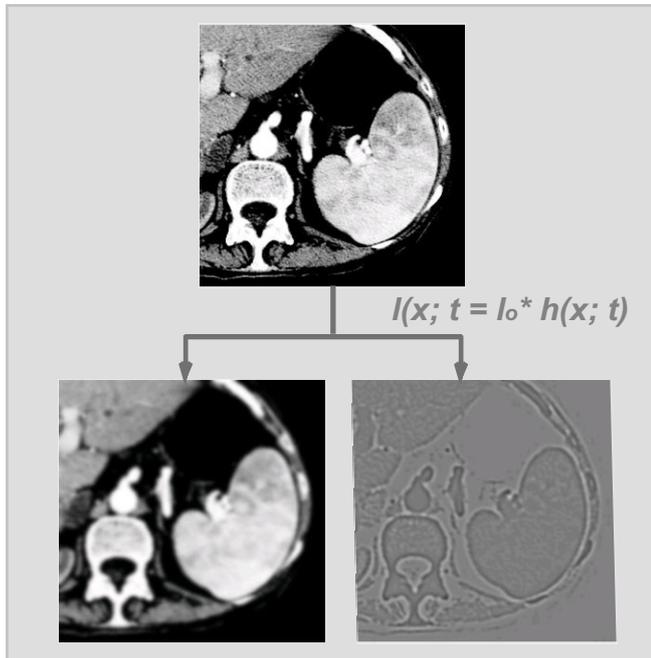


ISSN 1677-8464

Filtro de Difusão Linear Complexa para Detecção de Bordas: Implementação Java

José Iguelmar Miranda¹
João Camargo Neto²



O objetivo deste comunicado é mostrar a implementação Java™ do filtro de difusão linear complexa (Gilboa et al., 2004) para detecção de bordas. A necessidade dessa implementação se deve ao seu uso em projetos desenvolvidos na Embrapa Informática Agropecuária na área de processamento de imagens aplicado a agropecuária. Atualmente, esse filtro está sendo utilizado no projeto de contagem, não-supervisionada, de pés de laranja e caju.

A detecção de bordas é um dos processos mais comuns na análise de imagens digitais, contando com uma grande variedade de algoritmos. A razão é que as bordas definem o contorno dos objetos encontrados na imagem. As bordas são as regiões de transição numa imagem digital e geralmente definem fronteiras entre um objeto e o fundo, e entre contornos de objetos que se sobrepõem ou se tocam. Isso significa que se as bordas dos objetos contidos na imagem podem ser detectadas precisamente, os objetos podem ser localizados e suas propriedades básicas, como área, perímetro e forma, medidas. Portanto, o processo de detecção de bordas se qualifica como uma ferramenta essencial para a análise de imagens.

A detecção de bordas é parte de um processo conhecido como *segmentação* – a identificação de regiões dentro de uma imagem. Existem, no processamento de imagens, duas técnicas que trabalham com bordas: *detecção de bordas* e

realce de bordas. Tecnicamente, a *detecção de bordas* visa localizar os pixels de borda enquanto o *realce de bordas* incrementa o contraste entre as bordas e o fundo, tornando as bordas mais visíveis (Parker, 1997). Na prática, ambos os termos são usados com a mesma finalidade, pois a maioria dos algoritmos de detecção de bordas atribui aos pixels de borda valores que os tornam mais visíveis.

Aspectos Teóricos

Existem algumas definições possíveis para bordas, que se aplicam em diferentes circunstâncias. A definição mais comum e genérica é *borda de passo ideal* (Fig. 1a). Neste exemplo de uma dimensão, a borda representa a mudança de nível de cinza num local específico. Quanto maior a diferença de níveis de cinza no local da transição, mais fácil se torna detectar a borda. Porém, no mundo real, ocorrem complicações. Por exemplo, no processo de digitalização, a amostragem da cena nem sempre consegue fazer uma correspondência exata entre as bordas dos objetos reais e os pixels que lhes vão representar. O mais comum é que ocorra o processo mostrado na Fig. 1b.

Uma segunda complicação tem a ver com o problema ubíquo do *ruído*. Devido a muitos fatores, como intensidade de luz, tipo de câmera e lentes, movimento, temperatura,

¹ Ph.D. em Geoprocessamento, Pesquisador da Embrapa Informática Agropecuária, Caixa Postal 6041, Barão Geraldo - 13083-970 - Campinas, SP. (e-mail: miranda@cnptia.embrapa.br)

² Ph.D. em Processamento de Imagens, Pesquisador da Embrapa Informática Agropecuária, Caixa Postal 6041, Barão Geraldo - 13083-970 - Campinas, SP. (e-mail: camargo@cnptia.embrapa.br)

efeitos atmosféricos, entre outros, é improvável que dois pixels que correspondam precisamente ao mesmo nível de cinza na cena terão o mesmo nível na imagem digital. Os ruídos são do tipo aleatório e sistemático. Ruídos aleatórios só são caracterizados por uma distribuição estatística. Ruídos sistemáticos são mais fáceis de detectar e eliminar. O resultado de ruídos na imagem é produzir uma variação aleatória nos valores dos níveis de cinza entre os pixels, de maneira que a borda ideal mostrada na Fig. 1 não é encontrada em imagens reais.

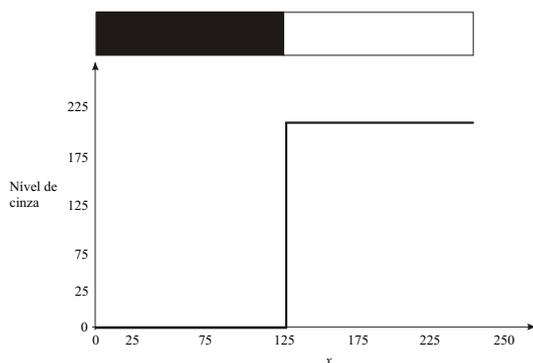


Fig. 1a. Exemplo de borda ideal.

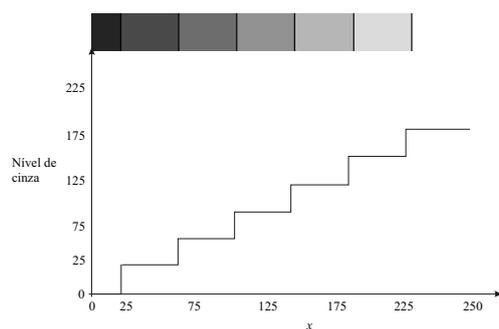


Fig. 1b. Exemplo de borda ideal.

Portanto, não se pode ignorar a presença do ruído aleatório nas imagens. O problema é que ele não pode ser identificado e medido precisamente, desde que não se pode diferenciar a sua contribuição nos valores de níveis de cinza dos pixels da imagem. Felizmente, algumas vezes, o ruído aleatório pode ser caracterizado pelo seu efeito na imagem, expressado como uma distribuição de probabilidade tendo média e desvio padrão específico (Parker, 1997). Assim, antes de se trabalhar com uma imagem, faz-se necessário um processo de filtragem desse tipo de ruído, normalmente através do processo de detecção de bordas.

Desde que uma borda é definida como uma mudança de nível de cinza, um operador que seja sensível a essa mudança tem a tarefa de detectar bordas. Geralmente, os operadores de borda podem ser classificados em três grupos: a) aqueles com base em derivadas parciais, aproximadas por *diferenças* para o caso discreto das imagens digitais, cuja função é identificar lugares onde existem grandes mudanças de intensidade; b) aqueles que modelam a borda com um filtro de pequenas dimensões mostrando propriedades abstratas de uma borda ideal; e c) operadores que usam modelos

matemáticos para as bordas, com base em equações diferenciais parciais, ou modelos de difusão, que procuram, dependendo do método adotado, por máximos ou mínimos da função. Esse último é o tipo de filtro da presente abordagem.

Filtro de Difusão Complexa

Gilboa et al. (2004) desenvolveram o filtro de difusão linear complexa com base na idéia de que em várias áreas da física e da engenharia podia-se estender a análise do domínio real para o domínio complexo, mesmo que os valores envolvidos fossem todos do domínio apenas real. Exemplos desse procedimento podem ser tirados do uso de transformadas de Fourier e de *wavelets* no processamento de imagens digitais, transformadas essas com valores complexos. Ressalta-se que as *wavelets* não são necessariamente transformadas complexas. Os autores adotaram a idéia de trabalhar com números complexos e generalizaram a idéia de modelar filtros com equações diferenciais parciais, usadas para construir modelos de difusão.

Esses modelos de difusão são conhecidos na matemática como *problemas com valores de contorno*, representado por uma equação diferencial parcial ou ordinária, dada com *condições de contorno*, para assegurar uma solução única. As condições de contorno especificam os valores da função ou suas derivadas nos contornos (bordas) da região na qual a equação é definida. Os modelos de difusão podem ser linear, ou isotrópico, e não linear, ou anisotrópico. O presente trabalho apresenta o modelo linear, conforme proposto por Gilboa et al. (2004). Portanto, toda referência ao modelo de difusão, será considerado o linear.

Existe um tipo de transformação no processamento de imagens chamado espaço-escala, que consiste na extração de informação espacial de uma imagem considerando um conjunto de escalas, indo de pequenos detalhes em áreas específicas a grandes atributos que se estendem sobre a imagem. Ela funciona como filtros aplicados repetidamente em diferentes escalas da imagem (Schowengerdt, 1997). Os autores generalizam o uso dessa transformação para o domínio complexo, combinando a equação diferencial parcial de difusão com o modelo livre da equação de Schrödinger.

A análise da difusão complexa mostra que a difusão generalizada apresenta propriedades da difusão direta e inversa. O ponto fundamental do estudo dos autores, suportado tanto teórica como numericamente, é que a parte imaginária serve como um detector de bordas, equivalendo à segunda derivada na escala temporal, cujo efeito é suavizar a imagem, quando o coeficiente de difusão complexa, c no presente trabalho, se aproxima do eixo real (Fig. 2).

A equação diferencial parcial canônica da espaço-escala linear que satisfaz a ambos métodos (máximo ou mínimo da função, conforme dito na introdução) tem a seguinte forma:

$$I_t = c\Delta I, \quad t > 0; x \in \mathfrak{R}.$$

$$I(x; 0) = I_0, \quad I_0 \in \mathfrak{R}; c, I \in \mathbb{C} \text{ (complexos).}$$

onde I_i é a imagem "perfeita" descrevendo a cena real; I_o é a imagem observada, ou inicial, apresentando uma degradação, devido ao ruído; c é o coeficiente de difusão complexo e Δ é o Laplaciano da função. Como a equação apresentada é uma diferencial parcial, a expressão $I_t = c\Delta I$ indica a condição de contorno e $I(x; 0) = I_o$ indica a condição inicial. O coeficiente de difusão complexa pode ser escrito como $c = re^{i\theta}$, sendo que c , no presente caso, se restringe a um valor positivo real ($\theta = 0$) (Fig. 2).

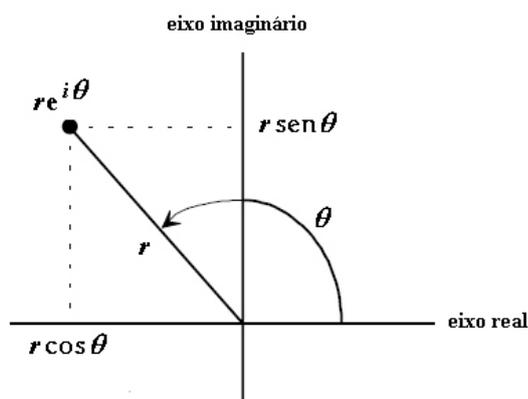


Fig. 2. Forma exponencial e retangular de um número complexo.

A abordagem espaço-escala é uma técnica múltipla resolução estabelecida para a análise de estrutura de imagens. A informação distribuída sobre todas as escalas é gerada como a solução $I(x, y; t)$ da equação linear do calor. Dessa maneira, pontos críticos e bordas são retirados de todas as escalas permitindo uma análise da cena como um todo. Um problema associado com essa abordagem é que importantes atributos estruturais, como as bordas, são suavizadas e borradas ao longo do fluxo, à medida que a imagem processada evolui no tempo. Como consequência, a trajetória dos cruzamentos-zero da segunda derivada, que indicam a localização de bordas, variam de escala para escala.

Para contornar esse problema, Perona & Malik (1990) propuseram um processo de difusão não linear, ou anisotrópico, onde a difusão acontece de acordo com uma variável adaptada, o coeficiente de difusão, para reduzir o efeito de suavização próximo das bordas. Gilboa et al. (2004) propuseram mais uma generalização para a espaço-escala linear e não linear, que passaram a ser casos especiais de uma teoria mais geral de processos de difusão complexo. Esses processos são encontrados na física quântica, eletromagnetismo e ótica. Gilboa et al. (2004) usaram como modelo a equação de Schrödinger³, dependente do tempo.

A equação de Schrödinger descreve o comportamento mecânico quântico. Também chamada de equação de onda de Schrödinger, é uma equação diferencial parcial que descreve como a função de onda de um sistema físico

evolui ao longo do tempo. No caso mais simples de uma partícula sem *spin* (rotação), sujeita a um campo externo, ela tem a forma:

$$i\hbar \frac{\partial \Psi}{\partial t} = -\frac{\hbar^2}{2m} \frac{\partial^2 \Psi}{\partial x^2} + V(x)\Psi(x,t) \quad (1)$$

onde $\Psi = \Psi(x, t)$ é a função de onda de uma partícula quântica dependente do tempo t , m é sua massa, \hbar é a constante de Planck, $V(x)$ é o potencial do campo externo e $i = \sqrt{-1}$. Maiores detalhes sobre o uso dessa equação são encontrados em Gilboa et al. (2004).

O lado direito de (1) é chamado de *operador de Schrödinger* e é interpretado como um operador de energia das partículas em consideração. O primeiro termo é a energia cinética e o segundo é a energia potencial. Uma conclusão dos autores é que uma solução básica da equação (1) é um plano de ondas, cujo comportamento é encontrado no fluxo complexo. No estudo, os autores usaram a equação com energia potencial zero (sem campo externo) mas com "energia cinética" complexa e não linear.

Para usar o operador de difusão complexa, os autores consideraram a seguinte equação diferencial parcial com os seguintes valores inicial e de contorno do problema:

$$I_t = cI_{xx}, \quad t > 0, x \in \mathbb{R}, c \in \mathbb{C} \quad (\text{condição de contorno}) \quad (2)$$

$$I(x; 0) = I_o, \quad I_o \in \mathbb{R}, I \in \mathbb{C}. \quad (\text{condição inicial})$$

Essa equação unifica a equação de difusão linear para $c \in \mathbb{R}$ e a equação livre de Schrödinger, isto é, $c \in \mathbb{C}$ e $V(x) = 0$. A solução fundamental complexa $h(x; t)$ deve satisfazer a relação:

$$I(x; t) = I_o * h(x; t) \quad (3)$$

Onde $*$ representa convolução. A equação (3) significa que a imagem "perfeita" (I) pode ser obtida através de uma série de convoluções no tempo (t) a partir da imagem inicial (I_o). A série de convoluções teria o objetivo de eliminar o problema dos ruídos. No mais, a equação (3) é a fórmula genérica para se filtrar uma imagem, sendo a detecção de bordas um desses objetivos. Existem várias maneiras de h ser definida. A contribuição dos autores foi apresentar uma solução fundamental complexa para $h(x; t)$ dada por:

$$h(x; t) = K g_\sigma(x; t) e^{i\alpha(x; t)} \quad (4)$$

sendo:

$$g_\sigma(x; t) = \frac{1}{\sqrt{2\pi\sigma(t)}} e^{-x^2/2\sigma^2(t)}, \quad K = \frac{1}{\sqrt{\cos\theta}}, \quad \alpha(x; t) = \frac{x^2 \sin\theta}{4tr} - \frac{\theta}{2}, \quad \sigma(t) = \sqrt{\frac{2tr}{\cos\theta}}$$

quando $\theta \rightarrow 0$, a parte imaginária pode ser considerada como a segunda derivada, com efeito de suavização da imagem inicial, fatorada por θ e o tempo t . A equação (4), com uma dimensão, pode ser generalizada para \mathbb{R}^n .

A aplicação dessa solução está relacionada ao detalhamento do comportamento de pequenos valores de θ (o ângulo do espaço de coordenadas polares) obtido pela separação entre as partes real e imaginária do sinal (ou da imagem), $I = I_r + iI_i$, e do coeficiente de difusão, $c = c_r + ic_i$, em um conjunto de duas equações:

³Erwin Rudolf Josef Alexander Schrödinger, matemático austríaco, publicou seu trabalho revolucionário sobre a mecânica das ondas numa série de seis artigos em 1926. A mecânica das ondas foi a Segunda formulação da teoria quântica, sendo a primeira a mecânica de matrizes de Heisenberg. Por suas contribuições à mecânica quântica, recebeu o prêmio Nobel em 1933.

$$I_{Rt} = c_R I_{R_{xx}} - c_I I_{I_{xx}}, \quad I_{R|t=0} = I_0 \quad (5)$$

$$I_{It} = c_I I_{R_{xx}} + c_R I_{I_{xx}}, \quad I_{I|t=0} = 0$$

onde $c_R = \cos\theta$, $c_I = \sin\theta$ (Fig. 2).

Os autores sugerem o valor de $\theta = \pi/1000$, para implementações da solução da equação, valor esse usado na implementação Java.

Alguns Aspectos da Implementação

A linguagem Java apresenta-se atualmente como uma alternativa para a implementação de aplicações não só para a web, mas para uso diverso. A facilidade de se executar aplicativos Java em qualquer ambiente operacional é um forte apelo para o uso dessa linguagem. Pelo menos essa vantagem motivou o desenvolvimento do presente trabalho.

O construtor do filtro de difusão complexa é definido usando quatro parâmetros:

```
public FiltroDifusaoComplexa(String aFile, int
interacao, int metodo,
    Int limiar) {
    ...
}
```

Onde:

1. *aFile*: nome do arquivo com a imagem digital para ser filtrada;
2. *interacao*: o número de interações, ou seja, o número de convoluções no tempo t (equação 3);
3. *metodo*: o método a ser usado para o processo de convolução. Pode ser linear ou rampa, ou não-linear. O presente documento trata apenas do método linear;
4. *limiar*: o limiar, valor a ser usado de acordo com o parâmetro anterior, o tipo de método. Para o método linear, o limiar é sempre um. Para o método rampa, ou não linear, esse valor pode ser mudado.

O construtor tem a responsabilidade de ler a imagem fornecida, definir as variáveis e chamar o algoritmo de difusão complexa. No retorno dessa chamada, ele mostra as imagens resultantes do processo de convolução. O usuário pode visualizar o resultado apenas da componente real, imaginária, ou ambos, dado que se está trabalhando com números complexos. O método que implementa efetivamente a difusão complexa está implementado com dois tipos de assinatura (polimorfismo):

```
private int[][] difusaoComplexa(int[][] in, int
interacoes,
    Complex limiar) {
    ...
}
e
private int[][] difusaoComplexa(int[][] in, int
interacoes,
    int limiar) {
    ...
}
```

O primeiro método retorna a parte imaginária do número complexo e o segundo método retorna a parte real. Esse procedimento permite o usuário visualizar uma ou ambas as partes do número complexo.

Em qualquer dos casos, o método recebe a imagem digital no formato de uma matriz de inteiros bidimensional, o número de interações que se deseja fazer na imagem e o valor do limiar, definido como um número complexo ou como um inteiro. Como resultado, devolve a imagem convolvida em uma outra matriz de inteiros, bidimensional. Os valores dos pixels da imagem retornada são calculados de acordo com a parte do número complexo desejado. Para retornar a parte complexa:

```
mat[col][lin] = (int) (128.0 + buferMatriz
[col][lin]/theta);
```

sendo *theta* definido como $\pi/1000$. Para retornar a parte real:

```
mat[col][lin] = (int) Math.abs(buferMatriz
[col][lin]);
```

O núcleo do método reside num ciclo de repetições, em conformidade com o parâmetro *interacoes*, que define quantas vezes a imagem vai ser filtrada. A equação 5 está implementada dentro desse ciclo.

A classe *FiltroDifusaoComplexa* depende de outra classe, *MatrizComplexa*, devidamente implementada para que a aplicação funcione normalmente.

Estudo de Caso

A Fig. 3 mostra o resultado da aplicação do filtro linear de difusão complexa. A parte real resultante da aplicação do filtro aparece na parte superior da figura. A primeira imagem representa zero interação, ou seja, é a própria imagem original. À medida que o número de interações aumenta, essa imagem se torna mais borrada, pelo fato dela ter suas bordas continuamente suavizadas pelo algoritmo. O efeito continuado dessa suavização é dificultar a identificação de bordas da imagem, como pode ser constatado na figura.

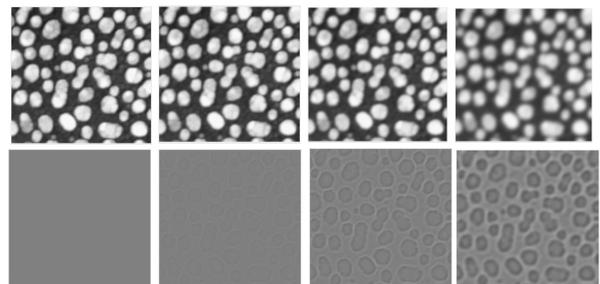


Fig. 3. Difusão complexa linear ($\theta = \pi/1000$). Valores da imagem depois de 0, 1, 5 e 25 interações. A partir da esquerda: parte real (superior) e imaginária (inferior).

Na parte inferior, aparecem as imagens equivalente à parte imaginária do filtro. Ao contrário do que acontece com a parte real, que é constantemente suavizada, perdendo as características das bordas, o filtro na parte imaginária mantém as características da imagem original, enquanto realça suas bordas, conforme o propósito da ação do filtro.

A aplicação do filtro no processo de identificação de pés de frutas tem se mostrado eficiente, possibilitando o processo de contagem.

Conclusões

- A utilização do filtro de difusão complexa se mostrou eficiente para uma primeira abordagem ao problema de contagem individual de plantas.
- A parte real da imagem original não tem utilidade para efeitos de detecção de bordas, sendo descartada no processo de contagem de pés de frutas.
- A implementação Java da versão do filtro se encontra disponível e operacional.
- Existe a intenção de se testar outros filtros com base em equações diferenciais parciais, como os de difusão anisotrópica, ou não linear.

Referências Bibliográficas

GILBOA, G.; SOCHEN, N.; ZEEVI, Y. Y. Image enhancement and denoising by complex diffusion processes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, v. 26, n. 8, p. 1020-1036, Aug. 2004.

PARKER, J. R. *Algorithms for image processing and computer vision*. New York: John Wiley, 1997. 417 p.

PERONA, P.; MALIK, J. Scale-space and edge detection using anisotropic diffusion. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, v. 12, n. 7, p. 629-639, July, 1990.

SCHOWENGERDT, R. A. *Remote sensing: models and methods for image processing*. San Diego: Academic Press, 1997. 522 p.

Comunicado Técnico, 75

Ministério da Agricultura, Pecuária e Abastecimento



Embrapa Informática Agropecuária
Área de Comunicação e Negócios (ACN)
Endereço: Caixa Postal 6041 - Barão Geraldo
13083-970 - Campinas, SP
Fone: (19) 3789-5743
Fax: (19) 3289-9594
e-mail: sac@cnptia.embrapa.com.br

1ª edição on-line - 2006

© Todos os direitos reservados.

Comitê de Publicações

Presidente: Kleber Xavier Sampaio de Souza.
Membros Efetivos: Adriana Farah Gonzalez (secretária), Ivanilde Dispatto, José Iguelmar Miranda, Marcia Izabel Fugisawa Souza, Silvio Roberto Medeiros Evangelista, Stanley Robson de Medeiros Oliveira.

Suplentes: Laurimar Gonçalves Vendrusculo, Maria Goretti Gurgel Praxedes.

Expediente

Supervisor editorial: Ivanilde Dispatto
Normalização bibliográfica: Marcia Izabel Fugisawa Souza
Editoração eletrônica: Área de Comunicação e Negócios