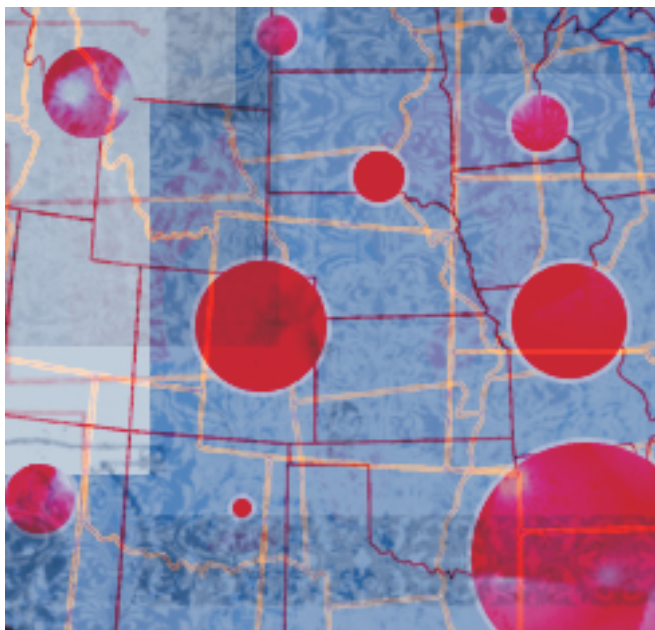




Filtro Shen-Castan (ISEF) para detecção de bordas

José Iguelmar Miranda¹
João Camargo Neto²

O filtro de Shen-Castan (Shen & Castan, 1992), também chamado de Infinite Symmetric Exponential Filter - ISEF, é considerado um filtro ótimo de detecção de bordas, a exemplo do filtro de Canny (Canny, 1986). Para incrementar a biblioteca de programas da Embrapa Informática Agropecuária na área de processamento de imagens, este documento apresenta a implementação do filtro ISEF em Java.

Uma introdução sobre o significado de bordas em imagens, bem como sua utilidade, é encontrada em Miranda & Camargo Neto (2006a), em que os autores apresentam a implementação Java do filtro de difusão linear complexa. Em Miranda & Camargo Neto (2006b), é apresentado um filtro de difusão complexa não linear, ou anisotrópico. Miranda & Camargo Neto (2007) apresentam mais um filtro, com base em equações diferenciais parciais ou modelos de difusão anisotrópica (não linear). O filtro ótimo de Canny (1986) também foi implementado em Java por Miranda & Camargo Neto (2008).

Com esta publicação, dá-se prosseguimento na implementação de algoritmos úteis para a detecção de bordas, atividade importante também no campo de visão por computador. O objetivo maior dessas

implementações é a constituição de uma biblioteca de processamento de imagens em Java, como software livre, sob a *General Public License* - GNU, conforme publicada pela *Free Software Foundation*. Todas as implementações encontram-se disponíveis no diretório da Rede Agrolivre (<http://repositorio.agrolivre.gov.br/projects/pid/>).

O filtro de Shen-Castan

Operação de diferença é uma das maneiras de se detectar bordas em imagens digitais, porém, essa abordagem é sensível a ruídos. Nesses casos, um processamento prévio, como a suavização, é recomendado para atenuar os ruídos. Muitos métodos de detecção de bordas em imagens digitais com ruídos têm sido propostos, como o gradiente de Robert, operadores de Sobel e de Prewitt, e o Laplaciano (Miranda, 2006). Em geral, esses métodos utilizam filtros de tamanho reduzido, como 3'3 ou 5'5. O trabalho de Shen e Castan tomou por base a teoria sobre detecção de bordas desenvolvida por Marr & Hildreth (1980), os quais mostraram que filtros com uma máscara de passa baixa de grande largura são menos sensíveis a ruídos. Partindo dessa ideia, Shen & Castan (1992)

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

imaginaram que o uso de um filtro com núcleo (kernel) de tamanho infinito resultaria em resultados ainda melhores.

Além da hipótese anterior, eles também consideraram uma melhor maneira de localizar, com precisão, os pixels da borda, e também a ação de um segundo filtro que apresentasse uma curva de distribuição com pequeno desvio padrão no centro, em vez de um filtro Gaussiano. No artigo, os autores analisam o processo de filtragem do ponto de vista do processamento de sinais, mas generalizam o resultado unidimensional para bi e multidimensional (Shen & Castan, 1992).

A definição de Shen e Castan para um detetor de bordas tem por base um conjunto de filtros de passa baixa e operações de diferenciação. Eles usaram um processo de otimização com base em três fatores: (1) a maximização da resposta do detetor a uma borda; (2) a minimização da resposta do filtro de suavização a ruídos; e (3) a minimização da resposta do detetor ao ruído (Pinho & Almeida, 1997).

O filtro de Canny (1986) produziu um resultado ótimo com relação a um conjunto de critérios. Segundo Parker (1997), enquanto esses critérios parecem razoáveis, não existe razão para serem os únicos, devido ao fato de o conceito de ótimo ser relativo, dando oportunidade de aparecerem outros critérios ótimos, em certas circunstâncias. Basicamente, Shen e Castan concordaram com os critérios de Canny: uma convolução com um filtro de suavização, seguido por uma busca dos pixels da borda. diferença nas abordagens está na função a ser otimizada. Shen & Castan (1992) sugerem a minimização do critério infinito normalizado, que em uma dimensão é:

A figura 1 indica que a função $f(x)$ que minimiza C_n é um filtro de suavização ótimo para um detetor de bordas. O filtro ótimo que os autores encontraram é o filtro

$$C_N = \frac{\sqrt{\int_{-\infty}^{\infty} f^2(x) dx \cdot \int_{-\infty}^{\infty} f'^2(x) dx}}{f^2(0)} \quad (1)$$

exponencial simétrico infinito, *Infinite Symmetric Exponential Filter – ISEF* (Shen & Castan, 1992), dado por:

$$f(x) = a \cdot b^{|x|} \quad (2)$$

Em 2, $a = (-\ln b)/2$ e $0 < b < 1$. No caso discreto, $a = (1 - b)/(1 + b)$ e $0 < b < 1$, que implica $0 < a < 1$. Os autores afirmaram que esse filtro produz melhor resultado na relação sinal ruído, do que o filtro de Canny, fornecendo ainda melhor localização das bordas. Por outro lado,

Shen e Castan não endereçaram o critério de múltiplas respostas das bordas, podendo criá-las com defeitos. Em duas dimensões, (2) toma a seguinte forma (Shen & Castan, 1992):

$$f(x,y) = a \cdot e^{-p(|x|+|y|)} \quad (3)$$

Sendo a como anteriormente mostrado, $0 < a < 1$ e $p > 0$. O valor do expoente representa a distância, em magnitude, entre x e y . Esse filtro pode ser aplicado a imagens de maneira semelhante ao filtro da derivada da função Gaussiana bidimensional (Miranda & Camargo Neto, 2008), decompondo-o em dois filtros de uma dimensão: um na direção x e outro na direção y . Observe que a função em (3) é real e contínua. Na implementação, ela é tornada discreta, mas precisa ter seus valores em números reais.

O trabalho de Shen e Castan contribuiu com a introdução de filtros recursivos de uma dimensão e seu uso acelera a convolução. Até esse ponto, a imagem só foi filtrada, as bordas são localizadas encontrando o cruzamento zero do Laplaciano (Marr & Hildreth, 1980). As derivadas expressam bem a mudança nos valores de cinza de uma imagem, possibilitando a detecção de bordas pelos valores máximos do gradiente ou o cruzamento zero da segunda derivada, incluso o Laplaciano, calculado por operações de diferença. Uma aproximação do Laplaciano pode ser obtida subtraindo a imagem original da imagem suavizada. Se S for a imagem suavizada e I a imagem original (Parker, 1997), então:

$$S[i,j] - I[i,j] \approx \frac{1}{4a^2} I[i,j] * \nabla^2 f(i,j) \quad (4)$$

É a imagem resultado e $D = S - I$ é a Laplaciana limitada por banda. A partir dessa imagem, uma imagem binária do Laplaciano (IBL) é obtida atribuindo o valor 1 para todos os pixels com valores positivo e 0 para os demais. Os pixels candidatos à borda estão na região de fronteira de IBL, que correspondem ao cruzamento zero.

Aspectos da implementação

A implementação do filtro de Shen e Castan é uma aplicação Java pura, contendo apenas um construtor e uma chamada pelo método `main()`. O uso de ambiente de desenvolvimento integrado, como NetBeans, ou uma interface mais rebuscada foram evitados. O objetivo principal é disponibilizar o filtro estanque, podendo ser inserido em outras aplicações de maneira direta ou, talvez, com um mínimo de adaptação. O programa deve ser compilado e executado em uma janela DOS, ou um TERM do Linux, mas também pode ser executado a partir de um ambiente integrado. Para compilar o programa basta executar o comando:

```
C:[diretório de instalação]>javac -
deprecation FiltroShenCastan.java
```

O filtro exige três parâmetros, conforme a chamada para execução:

```
C:[diretório de instalação]>java
FiltroShenCastan <imagem> <smoothing>
<windowSize>
```

Esses parâmetros são:

- <imagem> – a imagem a ser filtrada, em qualquer formato aceito pelo Java (.BMP, .JPEG, .JPG, .PNG, .GIF, etc.);
- <smoothing> – fator de suavização; o parâmetro b, conforme equação (2). Um bom valor a ser usado é 0,7;
- <windowSize> – largura da janela (kernel) do filtro do gradiente adaptativo. Pode ser 5, 7, 9, 11, etc. Um número ímpar é aconselhável. Quanto maior a dimensão do filtro, mais tempo de processamento será necessário.

O recurso de filtragem recursiva torna o processo de convolução, usado no algoritmo, mais rápido. A filtragem é feita pelo método ISEF, implementado como:

```
// Suaviza a imagem de entrada usando,
recursivamente, o filtro
// ISEF
smoothedImg = computeISEF(imgFloat,
b);
```

Em vez da imagem original, composta por valores inteiros de pixels, ela foi transformada em uma imagem de ponto flutuante, *imgFloat*, conforme explicado na equação (3). Esse método, por sua vez, chama dois outros, com o objetivo de realizar filtragem nas componentes *x* e *y*. Primeiro, ele chama o método:

```
// Primeiro, aplica o filtro na direção
vertical (nas linhas).
// O resultado do filtro é armazenado na
variável y.
y = aplicaISEFVertical(imgFloat,
causalComp, antiCausalComp, b);
```

cujo objetivo é filtrar as linhas. Em seguida, ele chama o segundo método:

```
// Segundo, aplica o filtro na direção
horizontal (nas colunas),
// usando a imagem filtrada nas linhas,
y.
y = aplicaISEFHorizontal(y, causalComp,
antiCausalComp, b);
```

para filtrar as colunas. O parâmetro *b* é o fator de suavização.

O trabalho realizado até agora foi de filtrar a imagem. O

próximo passo é gerar uma imagem binária do Laplaciano:

```
// Calcula imagem binária do Laplaciano
(ibl) a partir da imagem
// suavizada. imgBinLaplace é uma imagem
binária {0, 1}.
imgBinLaplace = computeIBL(smoothedImg,
imgFloat);
```

As bordas são localizadas nessa imagem encontrando os cruzamentos zero do Laplaciano, conforme equação (4).

```
// Faz detecção de bordas usando ibl com
limiar de gradiente
// Essa função altera apenas os valores de
'imgFloat', que
// armazena os valores das bordas
(gradiente).
imgFloat =
locateZeroCrossings(smoothedImg,
imgBinLaplace,
windowSize);
```

Em seguida, o método que implementa o filtro de Shen e Castan transforma de volta a imagem de ponto flutuante, *imgFloat*, em uma imagem inteira, *imgInt*, para que o resultado das bordas seja mostrado.

Estudo de caso

Na sequência de figuras a seguir, mostramos o resultado da aplicação do filtro de Shen e Castan para detecção de bordas. A Fig. 1 mostra a detecção de bordas em uma imagem do satélite Landsat, banda 5.

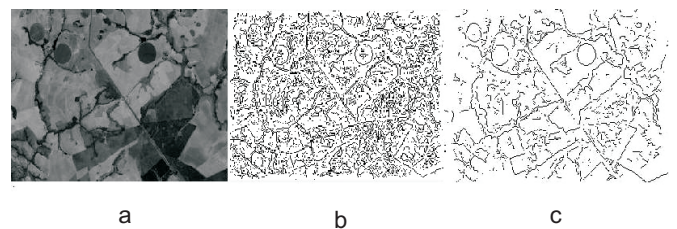


Fig. 1. (a) imagem original; (b) $b = 0,5$ e $w = 7$; (c) $b = 0,85$ e $w = 7$.

A Figura 1(a) mostra a imagem original. Em (b), passamos o filtro com $b = 0,5$ e $w = 7$; e em (c), $b = 0,85$ e $w = 7$. O primeiro valor de suavização permite que o filtro detecte as bordas dos objetos, mas mostra um efeito colateral, com a passagem das altas frequências nas áreas mais homogêneas. À medida que esse valor cresce, as altas frequências são filtradas, deixando transparecer melhor as bordas da imagem. No entanto, conforme mostrado anteriormente, b não pode ser maior que 1. O valor w representa a largura da janela (kernel)

do filtro do gradiente adaptativo.

A Figura 2 mostra outra sequência de imagens, agora com um conjunto de objetos curvilíneos.

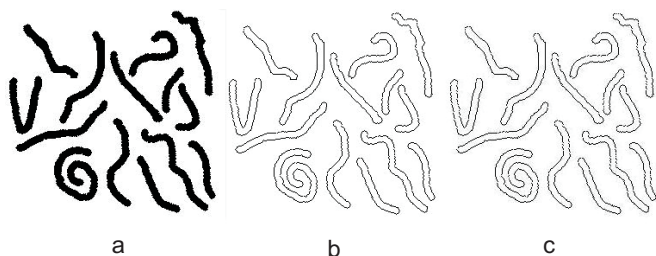


Fig. 2. (a) imagem original; (b) $b = 0,5$ e $w = 5$; (c) $b = 0,85$ e $w = 5$.

Nesse caso, a diferença entre os fatores de suavização não alterou em nada o processo de detecção de bordas. O menor valor já mostrou um bom desempenho do algoritmo. O valor da largura da janela foi menor que o do exemplo anterior, 5.

Por fim, na Figura 3, mostramos o filtro em ação com uma imagem colorida de um tomate. Nesse caso, a exemplo do que aconteceu na Fig. 1, os diferentes valores de suavização causaram efeitos diferentes na imagem com as bordas.

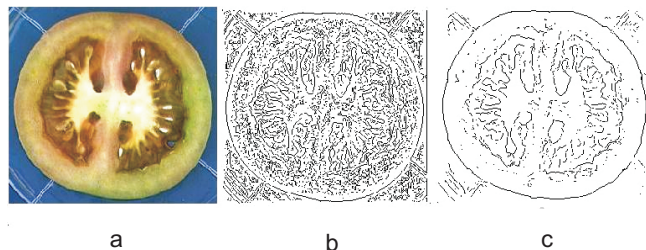


Fig. 3. (a) imagem original; (b) $b = 0,5$ e $w = 5$; (c) $b = 0,85$ e $w = 5$.

O ganho da distinção das bordas com o valor do parâmetro de suavização mais próximo de 1 resultou na filtragem mais eficiente de altas frequências em determinadas áreas da imagem.

Conclusões

1. A utilização do filtro de Shen e Castan se mostrou eficiente para a detecção de bordas;
2. Sobre o parâmetro de suavização b , da equação (2), observamos que valores menores que 0,7 permitem a passagem de altas frequências, com determinados tipos de imagens, dificultando a visualização das bordas;

3. Não percebemos vantagens significativas na variação da largura da janela (kernel) do gradiente adaptativo;
4. A implementação Java da versão do filtro se encontra disponível e operacional. A fonte, considerado software livre, está sob a licença *General Public License - GNU* conforme publicada pela *Free Software Foundation*, e disponível no diretório da Rede Agrolivre (<http://repositorio.agrolivre.gov.br/projects/pid/>). Dada a importância do uso desse tipo de aplicação, consideramos de utilidade sua disseminação de forma livre na comunidade científica.

Referências Bibliográficas

- CANNY, J. A computational approach to edge detection. *IEEE Transactions on Patterns Analysis and Machine Intelligence*, v. 8, n. 6, p. 679-698, 1986.
- MARR, D.; HILDRETH, E. Theory of edge detection. *Proceedings of the Royal Society of London. Series B*, v. 207, p. 187-217, 1980.
- MIRANDA, J. I. *Processamento de imagens digitais: prática usando Java*. Campinas, SP: Embrapa Informática Agropecuária, 2006. 322 p.
- MIRANDA, J. I.; CAMARGO NETO, J. Detecção de bordas com o modelo de difusão anisotrópica. In: SIMPÓSIO BRASILEIRO DE SENSORIAMENTO REMOTO (SBSR), 13, 2007, Florianópolis. *Anais...* São José dos Campos: INPE, 2007. p. 5957-5964. CD-ROM. Disponível em: <http://www.dsr.inpe.br/sbsr2007/>. Acesso em: 15 set. 2008.
- MIRANDA, J. I.; CAMARGO NETO, J. *Filtro de Canny para detecção de bordas: implementação Java*. Campinas: Embrapa Informática Agropecuária, 2008. (Embrapa Informática Agropecuária. Comunicado Técnico). No prelo.
- MIRANDA, J. I.; CAMARGO NETO, J. Filtro de difusão linear complexa para detecção de bordas: implementação Java. Campinas: Embrapa Informática Agropecuária, 2006a. 5 p. (Embrapa Informática Agropecuária. Comunicado Técnico, 75). Disponível em: <http://www.cnptia.embrapa.br/modules/tinycontent3/content/2006/ct75.pdf>. Acesso em: 15 set. 2008.
- MIRANDA, J. I.; CAMARGO NETO, J. Filtro de difusão complexa não linear para atenuação de ruídos:

implementação Java. Campinas: Embrapa Informática Agropecuária, 2006b. 4 p. (Embrapa Informática Agropecuária. Comunicado Técnico, 72). Disponível em:

<<http://www.cnptia.embrapa.br/modules/tinycontent3/content/2006/ct72.pdf>>. Acesso em: 15 set. 2008.

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

PINHO, A. J.; ALMEIDA, L. B. A review on edge detection based on filtering and differentiation. *Revista do DETUA*, v. 2, n. 1, p. 113-126, 1997.

SHEN, J.; CASTAN, S. An optimal linear operator for step edge detection. *Computer Vision, Graphics, and Image Processing: Graphical Models and Understanding*, v. 54, n. 2, p. 112-133, 1992.

Comunicado Técnico, 89

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) 3211-5743
Fax: (19) 3211-5754
URL: <http://www.cnptia.embrapa.br>
e-mail: sac@cnptia.embrapa.com.br

1ª edição on-line - 2008

Todos os direitos reservados.

Comitê de Publicações

Presidente: Kleber Xavier Sampaio de Souza.
Membros Efetivos: Leandro Henrique Mendonça de Oliveira, Marcia Izabel Fugisawa Souza, Martha Delphino Bambini, Sílvia Maria Fonseca Silveira Massruhá, Stanley Robson de Medeiros Oliveira, Suzilei Carneiro (secretária).

Suplentes: Goran Neshich, Maria Goretti Gurgel Praxedes.

Expediente

Supervisor editorial: Suzilei Carneiro
Normalização bibliográfica: Marcia Izabel Fugisawa Souza
Revisão de texto: Adriana Farah Gonzalez
Editoração eletrônica: Área de Comunicação e Negócios