

Implementação Java do filtro de Marr e Hildreth para detecção de bordas

José Iguelmar Miranda¹

Uma introdução sobre bordas em imagens digitais é encontrada em Miranda & Camargo Neto (2006a), onde 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. Miranda & Camargo Neto (2007) apresentam mais um filtro, com base em equações diferenciais parciais ou modelos de difusão anisotrópica. Além desses filtros, podem ser encontrados os seguintes filtros, também implementados em Java: o filtro ótimo de Canny (Miranda & Camargo Neto, 2008a) e o filtro de Shen e Castan (Miranda & Camargo Neto, 2008b).

O objetivo desta publicação é apresentar a implementação Java do algoritmo de Marr e Hildreth (Marr & Hildreth, 1980) para a detecção de bordas. O objetivo maior das implementações é a construção de uma biblioteca de processamento de imagens em Java, como software livre, sob a licença GPL (General Public License) conforme publicada pela Free Software Foundation. Todos os fontes se encontram disponíveis no diretório da rede Agrolivre (<<http://repositorio.agrolivre.gov.br/projects/pid/>>).

Técnicas de detecção de bordas com derivadas de

segunda ordem empregam alguma forma de diferenciação desse tipo de ordem para acentuá-las. Uma borda é detectada se existe uma mudança espacial acentuada na segunda derivada. Dois tipos de derivada de segunda ordem existem: Laplaciano e derivada segunda direcional (Pratt, 2007). O presente caso é abordado na ótica do Laplaciano.

O filtro de Marr e Hildreth

No final da década de setenta, David Marr procurou combinar o que se sabia acerca da visão biológica em um modelo que pudesse ser usado em visão por máquinas. Com base nas observações feitas, de acordo com Parker (1997), a conclusão foi que um algoritmo de detecção de bordas poderia ser condicionado a:

- Convolver a imagem, I , com uma função Gaussiana bidimensional.
- Completar o Laplaciano da imagem convolvida, L .
- Os pixels das bordas seriam aqueles para os quais existe um cruzamento zero em L .

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)

Portanto, o problema da detecção de bordas deveria considerar uma função Gaussiana e o seu Laplaciano ou derivada parcial de segunda ordem.

Isso foi o que constataram Marr & Hildreth (1980), para os quais, onde quer que uma mudança de intensidade ocorra, existirá um pico na primeira derivada direcional ou, de forma equivalente, um cruzamento zero na segunda derivada direcional da intensidade. Portanto, a tarefa de detectar essas mudanças pode ser reduzida a encontrar os cruzamentos zeros da segunda derivada da intensidade, na direção apropriada, em:

$$f(x, y) = \nabla^2 [G(x, y) * I(x, y)] \quad (1)$$

Onde o operador nabla, ∇^2 , representa o Laplaciano (Pratt, 2007):

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

Sendo a segunda derivada, $*$ é o operador de convolução, $I(x, y)$ é a imagem e $G(x, y)$ é a função Gaussiana bidimensional, definida por Parker (1997):

$$G(x, y) = \exp\left(-\frac{x^2 + y^2}{2}\right) \quad (3)$$

A justificativa para o uso do Laplaciano pelos autores surgiu da consideração sobre a orientação da derivada direcional. No artigo, eles consideraram a orientação da derivada direcional aquela que coincide com a orientação formada localmente pelo seu cruzamento zero, na direção do eixo y . Segundo os autores, o único operador diferencial de segunda ordem, independentemente da orientação, é o Laplaciano, ∇^2 .

Assim, a detecção de mudanças na intensidade pode ser observada com base no filtro apresentado em (1), $\nabla^2 G$.

Para fazer a convolução na imagem digital, a Gaussiana deve ser amostrada, criando uma imagem bidimensional. Marr & Hildreth (1980) propuseram o operador para detecção de bordas denominado Laplaciano do Gaussiano (LOG – *Laplacian of Gaussian*), no qual uma suavização calcada na função Gaussiana é realizada antes da aplicação do Laplaciano.

A fórmula acima, (1), geralmente é adaptada para usar cálculo de diferenças. Como a operação é comutativa, pode-se calcular o Laplaciano do Gaussiano analiticamente e amostrar a função, criando uma máscara de convolução que pode ser aplicada à imagem, produzindo o mesmo resultado. De acordo com Parker (1997), o Laplaciano do Gaussiano (LOG) é calculado com:

$$\nabla^2 G = \left(\frac{r^2 - 2}{4}\right) \exp\left(-\frac{r^2}{2}\right) \quad (4)$$

Sendo $r = \sqrt{x^2 + y^2}$

Aspectos da implementação

A implementação do filtro de Marr e Hildreth 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:

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

O filtro exige apenas dois parâmetros, conforme chamada para execução:

```
C:[diretório de instalação]>java FiltroMarr <imagem> <dp>
```

Onde os parâmetros são:

- <imagem> – a imagem a ser filtrada;
- <dp> – desvio padrão.

O programa cria primeiro uma matriz bidimensional para armazenar os valores do Laplaciano do Gaussiano, conforme equação (4):

```
amostraLoG = new double[n][n];
for (int i = 0; i < n; i++)
    for (int j = 0; j < n; j++)
        amostraLoG[j][i] = funcLoG(distancia((double)i, (double)j,
            (double)width, (double)width), s);
```

Em seguida, chama o método para fazer a convolução dessa matriz com a imagem original:

```
// Convolução da imagem fonte com a Gaussiana nas direções X e Y
System.out.println("Convolucao com Laplaciano do Gaussiano:\n");
arrayLaplace = convolveImagem(src, amostraLoG);
```

O passo final é gerar uma imagem com os cruzamentos zero do Laplaciano:

```
// Localiza os cruzamentos zeros
System.out.println("Cruzamentos zero:\n");
cruzaZero(arrayLaplace, src);
```

Um cruzamento zero no pixel p implica valores diferentes dos dois pixels opostos e vizinhos em uma dada direção (Fig. 1). Por exemplo, se a borda no pixel p for vertical, então o pixel à esquerda de p terá um sinal diferente do pixel à direita de p . Existem quatro casos para teste: superior/inferior, esquerda/direita e duas diagonais. O teste é realizado para cada pixel do Laplaciano do Gaussiano pela função `cruzaZero`, conforme recomendação de Parker (1997).

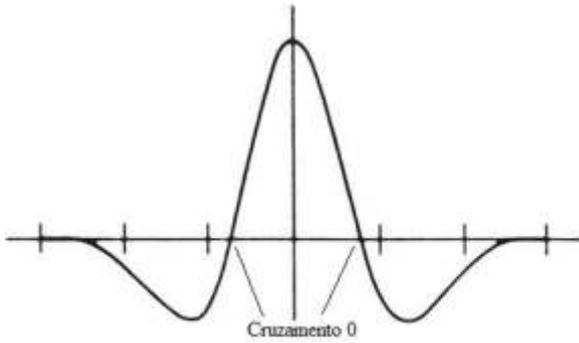


Fig.1. Segunda derivada da função Gaussiana em uma dimensão.

Estudo de caso

Na sequência de figuras a seguir, mostramos o resultado da aplicação do filtro de Marr e Hildreth para detecção de bordas. A Fig. 2 mostra a detecção de bordas em uma imagem do satélite Landsat, banda 5.

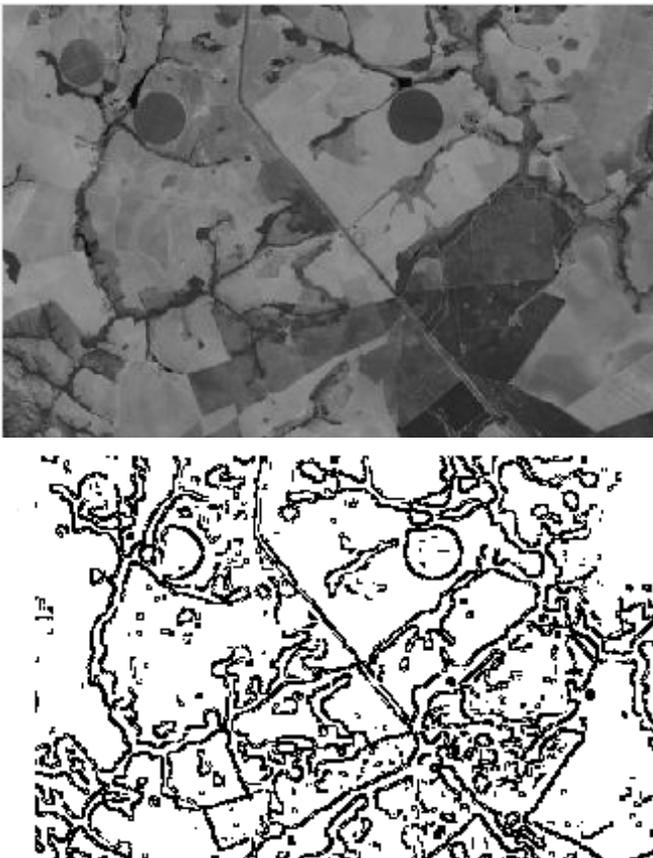


Fig. 2. (a) imagem original; (b) $\sigma = 1,4$.

A Fig. 2 (a) mostra a imagem original. Em (b), a imagem foi filtrada com um desvio padrão de 1,4. Percebe-se, na imagem filtrada, que o algoritmo conseguiu delinear algumas feições da imagem original. A identificação da estrada se confunde nas imediações das áreas com solo exposto (zona escura), por apresentarem respostas espectrais próximas. Próximo das áreas de vegetação

(zona clara), ela aparece mais delineada. Um pivô central ficou definido, com pequena distorção, pela proximidade com uma mata de galeria e os outros dois pivôs sofreram mais a influência dessas matas. No mais, as feições das matas galerias estão razoavelmente delineadas. Uma análise do filtro permite dizer que seu comportamento ficou dentro do esperado, pois o algoritmo conseguiu detectar as mudanças bruscas dos valores dos pixels de bordas (zonas claras e escuras).

A Fig. 3 mostra uma imagem com um conjunto de objetos em forma de fibras. Em (a), temos a imagem original e em (b), a imagem foi filtrada com um desvio padrão de 1,4. Como a imagem original é praticamente bicolor, a ação do filtro foi mais eficaz em detectar as bordas do tecido fibroso. Mesmo as fibras mais finas e tênues foram detectadas.

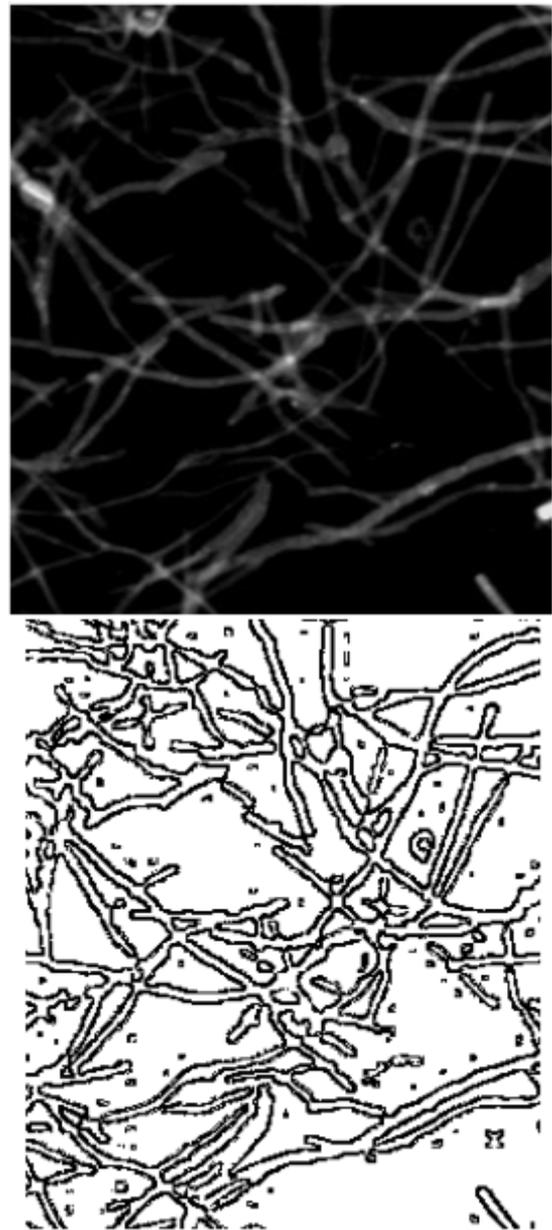


Fig. 3. (a) imagem original; (b) $\sigma = 1,4$.

Conclusões

- A utilização do filtro de Marr e Hildreth se mostrou eficiente para a detecção de bordas.
- Sobre o parâmetro desvio padrão, observamos que valores menores que 1,3 não apresentam bons resultados, dificultando a visualização das bordas.
- Da série de filtros desenvolvidos, conforme mencionado na Introdução, este é o que exige o menor número de parâmetros, apenas o desvio padrão.
- A implementação Java da versão do filtro se encontra disponível e operacional.
- O fonte, considerado software livre, está sob a licença GPL (General Public Licence) conforme publicada pela Free Software Foundation, e disponível no diretório da rede Agrolivre (<<http://repositorio.agrolivre.gov.br/projects/pid/>>).

Referências Bibliográficas

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.; 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, 2008a. (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.

MIRANDA, J. I.; CAMARGO NETO, J. *Filtro Shen-Castan (ISEF) para detecção de bordas*. Campinas: Embrapa Informática Agropecuária, 2008b. (Embrapa Informática Agropecuária. Comunicado Técnico). No prelo.

PARKER, J. R. *Algorithms for image processing and computer vision*. New York, NY: John Wiley, 1997. 417 p.
PRATT, W. K. *Digital image processing*. 4. ed. Hoboken, NJ: John Wiley, 2007. 786 p.

Comunicado Técnico, 90

Ministério da Agricultura, Pecuária e Abastecimento



Embrapa Informática Agropecuária
Área de Comunicação e Negócios (ACN)
Endereço: Caixa Postal 6041 - Barão Geraldo
13083-886 - 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