

Filtro de Difusão Complexa não-linear para Atenuação de Ruídos: Implementação Java

José Iguelmar Miranda¹
João Camargo Neto²

O objetivo deste comunicado é apresentar uma implementação Java™ do filtro de difusão complexa não-linear (Gilboa et al., 2004) para atenuar ruídos, enquanto realça bordas de uma imagem digital. É um filtro utilizado em projetos desenvolvidos na Embrapa Informática Agropecuária na área de processamento de imagens aplicado a agropecuária.

Este documento complementa o trabalho de Miranda & Camargo Neto (2006a). O leitor deve consultar a obra citada para se inteirar dos aspectos teóricos relevantes.

Filtro de Difusão Complexa não-linear

Gilboa et al. (2004) derivaram um filtro de difusão complexa não-linear, ou anisotrópico, tomando por base as propriedades do filtro de difusão linear complexa, implementado em Java por Miranda & Camargo Neto (2006a). O objetivo do filtro é atenuar ruídos e realçar bordas da imagem. Os autores denominaram esse filtro de “preservação de rampa” para atenuação de ruídos, com base no fato de que funções de rampa podem ser usadas como modelo da estrutura básica das bordas das imagens.

Bordas singulares, ou em formato de escada (Fig. 1b em Miranda & Camargo Neto (2006a)), são um caso especial de funções de rampa. Bordas no formato de rampa são componentes fundamentais genéricos de imagens. Portanto, torna-se importante comparar a ação de uma equação de difusão complexa não-linear numa função de rampa. A equação básica de difusão não-linear para preservar rampas suavizadas, proposta pelos autores, é:

$$I_t = \frac{\partial}{\partial x} (c(\cdot) I_x) \quad (1)$$

Onde $c(\cdot)$ é o coeficiente de difusão. A questão é procurar um operador diferencial adequado para bordas tipo rampa. Uma proposta para esse operador é a segunda derivada (o Laplaciano em mais de uma dimensão), por apresentar uma alta magnitude próxima dos pontos extremos e baixa magnitude nos outros pontos, permitindo que o processo de difusão não-linear reduza o ruído sobre a rampa. Porém, dois problemas persistem com esse modelo: a) o ruído tem segundas derivadas altas; e b) o uso da terceira derivada parcial nesse modelo com estimativas de derivadas com mais ruído.

¹ Ph.D. em Geoprocessamento, Técnico 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)

Gilboa et al. (2004) propõem, então, usar a difusão complexa não-linear, seguindo os resultados da difusão complexa linear, implementando o valor imaginário do sinal para controlar o processo de difusão. A equação fundamental do processo é:

$$I_t = \nabla \cdot (c(\text{Im}(I))\nabla I)$$

$$c(\text{Im}(I)) = \frac{e^{i\theta}}{1 + \left(\frac{\text{Im}(I)}{k\theta}\right)^2} \quad (2)$$

sendo k um parâmetro de limiar. Como no procedimento da difusão complexa linear, o valor de θ , o ângulo de fase, é pequeno. Os detalhes sobre a implementação discreta se encontram no apêndice do referido artigo.

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 portabilidade de aplicativos Java para qualquer ambiente operacional é um forte apelo para o uso dessa linguagem. Pelo menos essa vantagem motivou o desenvolvimento do presente trabalho.

O construtor da classe do filtro de difusão complexa não-linear é definido com três parâmetros:

```
public FiltroDifusaoComplexaNaoLinear (
    String aFile,          // arquivo com imagem a ser filtrada.
    int iteracao,         // número de iterações; "t" (tempo) de (1)
    int limiar) {        // valor limiar; "k" na equação (2)
    ...
}
```

Onde:

1. `aFile`: nome do arquivo com a imagem digital para ser filtrada;
2. `iteracao`: o número de iterações, ou seja, o número de convoluções no tempo t (equação 1);
3. `limiar`: valor de "k" da equação (2). Sugere-se um valor de cinco.

O construtor tem a função de receber a imagem fornecida, definir algumas variáveis e chamar o algoritmo de difusão complexa não-linear. No retorno dessa chamada, ele mostra um painel (`JFrame`) com duas imagens: a original e o componente real da imagem complexa. O método que implementa a difusão complexa não-linear está definido como:

```
Private int[][] difusaoRampa(int[][] in, int iteracoes, int limiar) {
    ...
}
```

O método retorna a parte real ou imaginária do número complexo em uma matriz bidimensional (`int[][]`). No retorno ao construtor, antes de ser mostrada no painel, essa matriz é convertida em uma imagem do tipo `BufferedImage`, tarefa atribuída ao método `criaImagem`:

```
private BufferedImage criaImagem(int[][] in) {
    ...
}
```

que recebe como parâmetro uma matriz bidimensional e retorna uma `BufferedImage`, o resultado do filtro de difusão complexa não-linear.

O método principal, `difusaoRampa`, recebe a imagem digital no formato de uma matriz de inteiros bidimensional, o número de iterações que se deseja fazer na imagem (o parâmetro t) e o valor do limiar (o parâmetro k), ambos definidos como um inteiro. O núcleo do método reside num ciclo, em conformidade com o parâmetro `iteracoes`, que define quantas vezes a imagem vai ser filtrada, conforme a forma discreta da equação (2) e como resultado, retorna a imagem filtrada em uma matriz de inteiros, bidimensional.

Por definição, o método retorna a parte real da imagem complexa, pois neste algoritmo o que mais interessa é a parte real. Mas o componente imaginário pode também ser visualizado, sendo suficiente comentar o laço que define o `array` com a parte real da imagem e descomentar o laço que define o imaginário. No corpo do método o leitor encontra o local exato onde fazer essa alteração.

A classe `FiltroDifusaoComplexaNaoLinear` depende das classes `MatrizComplexa`, e `Complex`, Devidamente implementadas e disponíveis no diretório da rede agrolivre (<http://repositorio.agrolivre.gov.br/projects/pid/>) onde se encontra o presente programa, para que a aplicação funcione normalmente.

Estudo de Caso

Na seqüência de figuras a seguir, comparou-se o resultado da aplicação dos filtros de média e de difusão complexa anisotrópica para atenuação de ruídos e preservação de bordas. A Fig. 1 mostra uma área de agricultura em Guaíra, SP, com dois pivôs central, algumas áreas de solo nu (área branca), cana-de-açúcar e matas ciliares ao longo de córregos. A imagem da esquerda é a original, na qual simula-se um ruído uniforme, mostrado na imagem da direita.

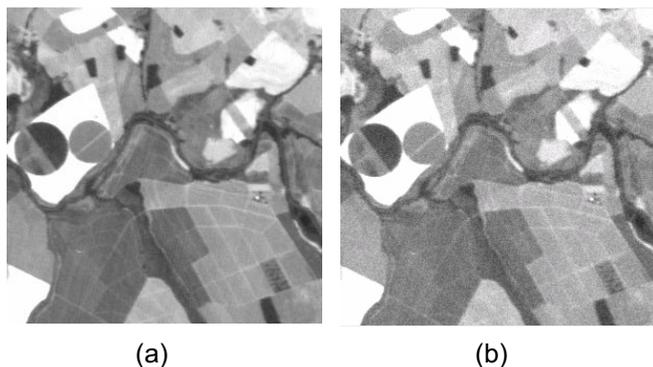


Fig. 1. A partir da esquerda: (a) imagem original e (b) imagem com ruído uniforme.

A Fig. 2 mostra, na esquerda, a imagem com ruído uniforme filtrada usando um filtro de média e na direita, uma linha de perfil na altura dos pivôs. A ideia é mostrar a variação dos valores de níveis de cinza ao longo dessa linha de perfil, comparando-a com as imagens filtradas com a difusão complexa.

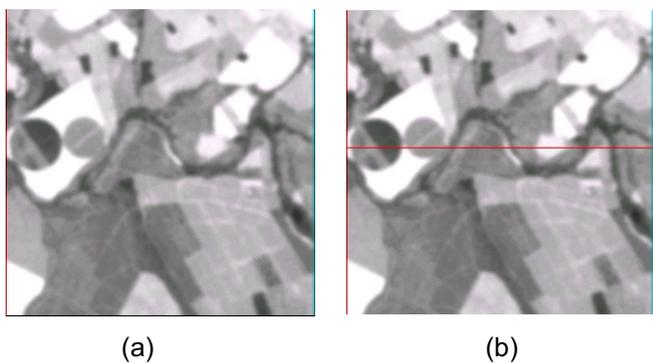


Fig. 2. Imagem filtrada (filtro de média) e linha de perfil.

A Fig. 3 mostra o resultado da aplicação do filtro de difusão complexa não-linear na imagem com ruído uniforme. O primeiro processo de filtragem usa cinco iterações (Fig. 3a) e o segundo, dez iterações (Fig. 3b). Essas imagens preservam mais detalhes da imagem original (Fig. 1a) do que a imagem da Fig. 2a (filtro de média).

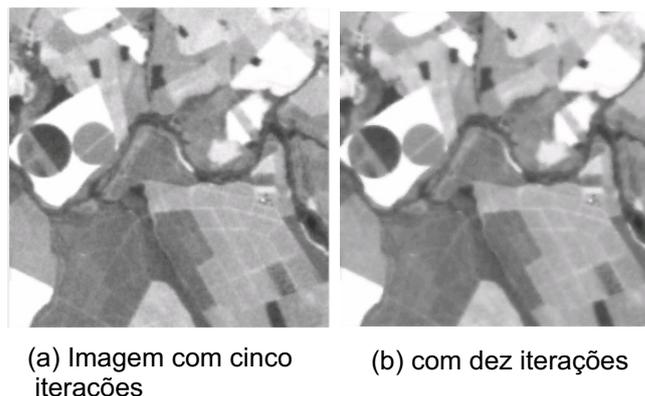


Fig. 3. Difusão complexa não-linear ($\theta = \pi/1000$). Imagem depois de 5 e 10 iterações, ambas com limiar = 5.

Visualmente, a imagem filtrada por difusão complexa anisotrópica com cinco iterações se mostra bem semelhante à imagem com filtro de média. A imagem com dez iterações mostra uma atenuação do ruído ainda maior, sem perda de bordas. As imagens filtradas com o filtro de difusão têm uma peculiaridade em relação ao filtro de média: as regiões se tornam mais homogêneas, perdendo granularidade. Para mostrar que o filtro de média atenua mais as bordas que o filtro de difusão complexa, mostra-se a seguir o histograma traçado ao longo da linha do perfil (Fig. 2). Embora a linha só tenha sido mostrada para a imagem com filtro de média, o programa que calcula o histograma do perfil foi executado para as imagens com filtro de difusão complexa, na mesma linha.

A Fig. 4 mostra o perfil da imagem com ruído uniforme. Os níveis de cinza são mostrados no eixo y e a linha do perfil ao longo do eixo x. Na figura são mostrados apenas os valores até o pixel cento e cinquenta e cinco. Picos correspondem a áreas com maiores valores de níveis de cinza. Por exemplo, o primeiro pico corresponde ao solo nu (branco), antes do pivô. Os níveis de cinza do primeiro pivô correspondem aos dois primeiros vales, aproximadamente entre os pixels dez e cinquenta e nove. O segundo pivô corresponde, aproximadamente, aos pixels entre sessenta e cem. O pico em cento e um corresponde a uma pequena área de solo exposto, entre esse pivô e a mata galeria. Os valores entre cento e um e cento e cinquenta e cinco, aproximadamente, correspondem a mata galeria e a cana-de-açúcar. O pico em cento e cinquenta e oito corresponde a estreita faixa de solo exposto, entre o canal e a mata galeria.

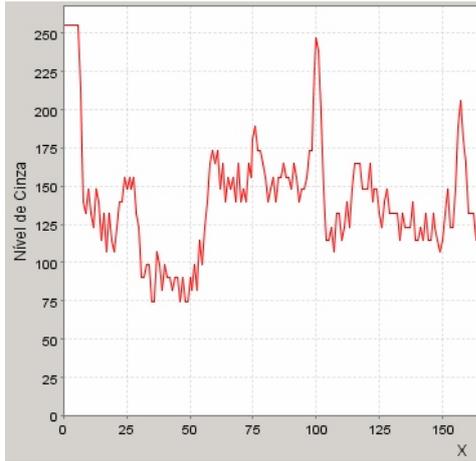


Fig. 4. Perfil da linha na imagem com ruído uniforme.

A Fig. 5 mostra o perfil da imagem com ruído uniforme filtrada usando os filtros de média (a) e difusão complexa (b). Comparando esses histogramas com o da Fig. 4, percebe-se que o processo de filtragem foi eficiente em ambos os casos. Praticamente, todo o ruído da imagem foi suavizado.

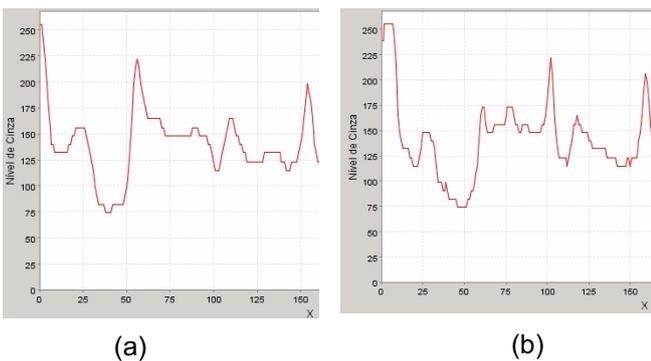


Fig. 5. Perfil da linha na imagem filtrada com os filtros de média (a) e difusão complexa (b).

A comparação mais interessante está entre os histogramas (a) e (b). Enquanto o filtro de média tende a perder informações das bordas da imagem, devido às suavizações, o filtro de difusão, após cinco iterações, permanece mais “fiel” ao histograma original, obviamente, sem os ruídos (Fig. 4). Na Fig. 5a, aproximadamente no pixel cinquenta e cinco, aparece um pico (nível de cinza = 225) que não existe, de forma tão acentuada, no histograma da imagem com ruído (nível de cinza = 175). O valor de 225 equivale a uma pequena área de solo exposto entre os dois pivôs, que no filtro de média foi bastante acentuada. Perceber que no filtro de difusão, o valor permanece o mesmo que da imagem com ruído (nível de cinza = 175). De uma maneira geral, o histograma da Fig. 5a nos mostra uma imagem bem mais suavizada que a imagem que gerou o histograma da Fig. 5b. E isso pode ser constatado

comparando as imagens das Fig. 2a e 3a.

Como uma última comparação, gerou-se duas imagens com as bordas destas imagens usando o algoritmo de Canny (1986). A Fig. 6a mostra as bordas da imagem gerada pelo filtro de média e a Fig. 6(b) mostra as bordas para a imagem gerada com o filtro de difusão.

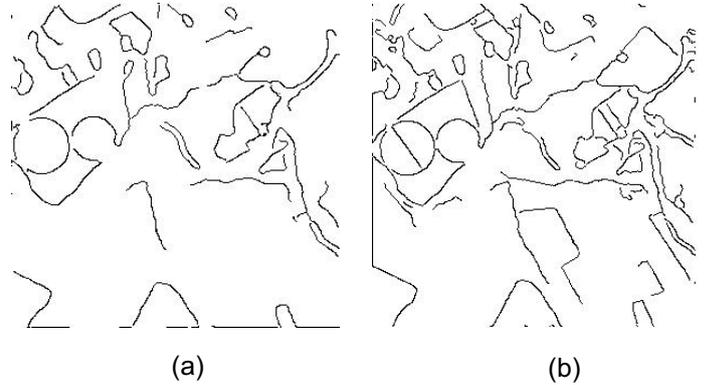


Fig. 6. Perfil da linha na imagem filtrada com os filtros de média (a) e difusão complexa (b).

Comparando as Fig. 6a e Fig. 6b, percebe-se que várias bordas foram simplificadas em (a) e preservadas em (b). Por exemplo, o primeiro pivô perdeu a divisão.

Conclusões

- A utilização do filtro de difusão complexa não-linear se mostrou eficiente, em comparação ao filtro de média, por exemplo, para a atenuação de ruídos na imagem, enquanto preserva as bordas dos objetos presentes. A implementação Java da versão do filtro se encontra operacional e disponível no endereço <<http://repositorio.agrolivre.gov.br/projects/pid/>>.
- Enquanto em Miranda & Camargo Neto (2006a) a parte real não tinha utilidade, na presente abordagem ocorre o inverso. A parte imaginária da imagem original não tem utilidade para efeitos de atenuação de ruídos no filtro de difusão complexa não-linear.
- O presente filtro faz parte da biblioteca Java de filtros de difusão não-linear, ou anisotrópico (Miranda & Camargo Neto, 2006b).

Referências Bibliográficas

- CANNY, J. A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, v. 8, n. 6, p. 679-698, 1986.
- GILBOA, G.; SOCHE, 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, 2004.

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: 12 fev. 2007.

MIRANDA, J. I.; CAMARGO NETO, J. *Modelo de difusão anisotrópica para detecção de bordas*. 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: 12 fev. 2007.

Comunicado Técnico, 83

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 - 2007

© Todos os direitos reservados.

Comitê de Publicações

Presidente: Kleber Xavier Sampaio de Souza.
Membros Efetivos: Adriana Farah Gonzalez (secretária), Ivanilde Dispatto, Marcia Izabel Fugisawa Souza, Martha Delphino Bambini, Sílvia Maria Fonseca Silveira Massruhá, Stanley Robson de Medeiros Oliveira.

Suplentes: Goran Neshich, Leandro Henrique Mendonça de Oliveira, Luiz Manuel Silva Cunha, Maria Goretti Gurgel Praxedes.

Expediente

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