

PTC2892

Princípios da Formação e Processamento de Imagens Médicas

Ambiente computacional e filtros

Sérgio S Furuié

S. Furuié 30/7/2010 - 1

Plano da aula de hoje

- Exemplos de programação
 - ✓ Plugins e java
 - ✓ Eclipse como IDE integrado ao ImageJ
- Operadores
- Convolução e filtros
- As aulas estão no site:
 - ✓ <http://moodle.stoa.usp.br>

S. Furuié 30/7/2010 - 2

Algumas perguntas

- Como aumentar a produtividade na implementação de plugins?
 - ✓ Usar ambiente para desenvolvimento integrado
 - ✓ edição ⇔ compilação ⇔ teste ⇔ correção/edição ⇔ ...
 - ✓ => Eclipse



- Como despreocupar-se com bordas e tipos de dados?
 - ✓ Usar classe do Iplab (ImageAccess.java)

S. Furuié 30/7/2010 - 3

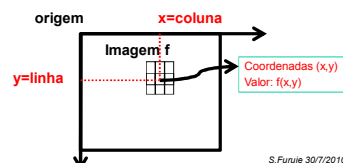
Eclipse

- Instalando ...
- Básico sobre o eclipse
 - ✓ Exemplos e Tutoriais p/ quem quiser se aprimorar
- Configurando para criar plugin no ImageJ
 - ✓ Criar projeto (ex. ptc) com saída em ptc/plugins
 - ✓ Libraries adicionais incluindo documentação
 - Add External Jars: ij.jar e vincular a documentação
 - ✓ Run/Main: ij.ImageJ
 - ✓ Linkar o source com a sua area de plugins no ij
 - Permite que voce use o mesmo src

S. Furuié 30/7/2010 - 4

Revisando alguns pontos

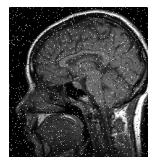
- Imagem f=> matriz
- Convenções:
 - ✓ canto superior esquerdo ⇔ matriz[linha=0,coluna=0]
 - ✓ Referência a matriz em java=>f[linha][coluna]
 - ✓ Função f(x=coluna,y=linha): elemento na abscissa x, ordenada y
 - ✓ Métodos em ImageJ: ip.getPixel(x,y)
 - ✓ Valor 0=> preto; valor 255=> branco
- plugins



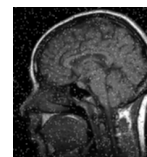
S. Furuié 30/7/2010 - 5

Conceito: filtro e convolução

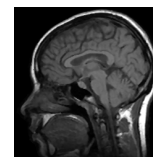
- Filtragem é uma operação feita sobre um sinal ou imagem para obter um outro sinal ou imagem
- Filtro linear e invariante pode ser implementado por convolução
- Convolução pode ser executado rapidamente no domínio da frequência por FFT



Ruído salt pepper



Filtro
Média, r=1



Filtro Mediana,
r=1

S. Furuié 30/7/2010 - 6

Como filtrar sinal?

□ **filtros digitais**

Séries numéricas:
 $S1 = [0 \ 2 \ 4 \ 4 \ 2 \ 0 \ -2 \ 12 \ 45 \ 166 \ 50 \ 33 \ -2 \ 0 \ \dots]^T$
 Filtragem da média entre os vizinhos $[1 \ 1 \ 1]/3$:
 $S2 = [0 \ 3 \ 3 \ 3 \ 3 \ 0 \ 3 \ 18 \ \dots]^T$

Grid intervalo: 0.2 sec, 0.5 mV

$$g(i) = \sum_{k=-N_w}^{N_w} w_k \cdot f(i-k)$$

$$\frac{-2.1 + 12.1 + 45.1}{3} = 18.3$$

S. Furule 30/7/2010 - 7

Convolução 1D: Lin Inv discreto

$x[n] = \sum_{k=-\infty}^{\infty} x[k] \cdot \delta[n-k]$
 $\delta[n-k] \rightarrow h_k[n]$

Linearidade $\Rightarrow y[n] = \sum_{k=-\infty}^{\infty} x[k] \cdot h_k[n]$

Invariante $\Rightarrow h_k[n] = h[n-k]$

$$y[n] = \sum_{k=-\infty}^{\infty} x[k] \cdot h[n-k] = \sum_{k=-\infty}^{\infty} h[k] \cdot x[n-k]$$

$$y[n] = x[n] * h[n]$$

Na prática, devido a finito e zeros $\Rightarrow y[n] = \sum_{k=k_1}^{k_2} h[k] \cdot x[n-k]$

S. Furule 30/7/2010 - 8

Convolução

- Formalmente e no caso contínuo: **convolução**
- Convolução: comutativo, associativo, distributivo

$$y(t) = \int_{-\infty}^{\infty} x(\tau) \cdot h(t-\tau) d\tau$$

S. Furule 30/7/2010 - 9

Formalizando convolução: 2D, contínuo

Definição para funções contínuas:

$$g(x, y) = h(x, y) * f(x, y)$$

$$g(x, y) \triangleq \iint_{-\infty}^{\infty} f(x', y') \cdot h(x-x', y-y') dx' dy'$$

Pode-se provar facilmente que equivale a:

$$g(x, y) = \iint_{-\infty}^{\infty} h(x', y') \cdot f(x-x', y-y') dx' dy'$$

e

$$G(u, v) = H(u, v) \cdot F(u, v)$$

Na qual G, H e F são as transformadas de Fourier 2D de g, h e f respectivamente

S. Furule 30/7/2010 - 10

Convolução 2D: discreto

$$g(x, y) = \sum_{x'=-a}^a \sum_{y'=-b}^b w(x', y') \cdot f(x-x', y-y')$$

S. Furule 30/7/2010 - 11

Filtragem de imagens

□ **Aplicando uma média (janela móvel de pesos)**

$$w = \frac{1}{9} \times \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

Algoritmo:

$$g(x, y) = \sum_{x'=-a}^a \sum_{y'=-b}^b w(x', y') \cdot f(x-x', y-y')$$

S. Furule 30/7/2010 - 12

Implementação em pseudo-código

$$g(x,y) = \sum_{x'=-a}^a \sum_{y'=-b}^b w(x',y') \cdot f(x-x',y-y')$$

Entradas:
f[R linhas][C colunas]; w[A linhas][B colunas] (A e B:ímpares)
Desconsidere as bordas

Algoritmo:

- Sejam: a=A/2; b=B/2
- for (x=a; x < (R-a); x++)
 for (y=b; y < (C-b); y++)
 g[x][y]=0;
 for (x1=-a; x1 <= a; x1++)
 for (y1=-b; y1 <= b; y1++)
 g[x][y]=g[x][y] + w[x1+a][y1+b] * f[x-x1][y-y1]
 end for
 end for
end for

S.Furue 30/7/2010 - 13

Implementação em java

$$g(x,y) = \sum_{x'=-a}^a \sum_{y'=-b}^b w(x',y') \cdot f(x-x',y-y')$$

Entradas:
f[R linhas][C colunas]; w[A linhas][B colunas] (A e B:ímpares)
Desconsidere as bordas

Algoritmo:

- int a=A/2; int b=B/2
- for (int x=a; x < (R-a); x++)
 for (int y=b; y < (C-b); y++)
 {g[x][y]=0;
 for (int x1=-a; x1 <= a; x1++)
 for (int y1=-b; y1 <= b; y1++)
 g[x][y]=g[x][y] + w[x1+a][y1+b] * f[x-x1][y-y1];
 }

S.Furue 30/7/2010 - 14

Evitando tipos

- P/ não ter de lidar com cada tipo (byte, short, int, float) vamos converter qualquer destes tipos em float
 - ✓ FloatProcessor ip_float=(FloatProcessor)ip.convertToFloat();
 - ✓ ImagePlus imp_float=new ImagePlus("Float", ip_float);
- Usar então apenas "ip_float" e "imp_float" de forma semelhante aos anteriores.

S.Furue 30/7/2010 - 15

```

10// Convolucao_media.java S Furue EPUSP
11// Realiza convolucao 2D geral, sem considerar bordas. Ex. com media. OK.
12
13import ij.*;
14import ij.process.*;
15import ij.plugin.filter.*;
16
17public class Convolucao_media implements PlugInFilter {
18    ImagePlus imp;
19
20    public int setup(String arg, ImagePlus imp) {
21        this.imp = imp;
22        return DOKS_ALL;
23    }
24
25    public void run(ImageProcessor ip) {
26        // transformando em float p/ float independente do tipo
27        FloatProcessor ip_float=(FloatProcessor)ip.convertToFloat();
28        ImagePlus imp_float=new ImagePlus("Float", ip_float);
29        // obtendo as dimensoes da imagem
30        int w = ip_float.getWidth();
31        int h=ip_float.getHeight();
32        int N=w/2; //deve ser IMPAR!
33        int m=N/2;
34        // criando novo array p/ nao interferir no original
35        float[] pixels_temp=new float[w*h];
36        // criando os pesos para as medias
37        float[] p; pesos=new float[N][N];
38        for(int r=0; r<N; r++)
39            for(int c=0; c<N; c++) pesos[r][c]=(float)1.0/(N*N);
40        // realizando a convolucao com o kernel da media
41        for(int ymc; y< h-m; y++) // nao considera bordas
42            for (int xmc; x<w-m; x++) // nao considera bordas
43                { int px = y*w + x; // posicao no array pixels[]
44                  float media=0;
45                  for(int c=-m; c<=m; c++)
46                      for(int o=-m; o<=m; o++)
47                          media=media + pesos[c+mc][o+mc]*ip_float.getPixelValue(x-o,y-c);
48                  pixels_temp[px]=media;
49                }
50        // atualizando o processador
51        ip_float.setPixels(pixels_temp);
52        // refresh na imagem
53        imp_float.show();
54    }
55}

```

V7/2010 - 16

Gradiente

$$\nabla f(x,y) = \frac{\partial f(x,y)}{\partial x} \hat{u}_x + \frac{\partial f(x,y)}{\partial y} \hat{u}_y, \quad (\text{Vetor !})$$

$$\left. \frac{\partial f(x,y)}{\partial x} \right|_x = \frac{f(x+1,y) - f(x-1,y)}{2}$$

$$\left. \frac{\partial f(x,y)}{\partial y} \right|_y = \frac{f(x,y+1) - f(x,y-1)}{2}$$

$\left. \frac{\partial f(x,y)}{\partial x} \right _x$	-1	0	1
$\left. \frac{\partial f(x,y)}{\partial x} \right _y$	-1	0	1
$\left. \frac{\partial f(x,y)}{\partial y} \right _x$	-1	0	1

Sobel	-1	0	1
$\left. \frac{\partial f(x,y)}{\partial x} \right _x$	-1	0	1
$\left. \frac{\partial f(x,y)}{\partial x} \right _y$	-2	0	2
$\left. \frac{\partial f(x,y)}{\partial y} \right _x$	-1	0	1
$\left. \frac{\partial f(x,y)}{\partial y} \right _y$	-1	-2	-1
	0	0	0
	1	2	1

Cuidado com representação computacional da imagem (índices)

S.Furue 30/7/2010 - 17

Módulo do vetor gradiente

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

$\left. \frac{\partial f(x,y)}{\partial x} \right _x$	-1	0	1
$\left. \frac{\partial f(x,y)}{\partial x} \right _y$	-1	0	1
$\left. \frac{\partial f(x,y)}{\partial y} \right _x$	-1	0	1
$\left. \frac{\partial f(x,y)}{\partial y} \right _y$	-1	-2	-1
	0	0	0
	1	2	1

Sobel

-1	0	1
-2	0	2
-1	0	1
-1	-2	-1
0	0	0
1	2	1

Segmentação de Imagens

S.Furue 30/7/2010 - 18

Exemplos de derivadas

ImageJ

- diferença entre elementos na horizontal (bordas verticais)
 - Derivada parcial em x
- diferença entre elementos na vertical (bordas horizontais)
 - Derivada parcial em y
- módulo do gradiente

95	80	80	80
95	90	90	105
150	20	100	130
220	150	160	150

$\left. \frac{\partial f(x,y)}{\partial x} \right|_{i,j} = \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix}$
 $\left. \frac{\partial f(x,y)}{\partial y} \right|_{i,j} = \begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix}$

-115

255

S.Furule 30/7/2010 - 19

Algoritmo p/ Laplaciano

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

$$\left. \frac{\partial f(x,y)}{\partial x} \right|_{i,j} = f(i + \frac{1}{2}, y) - f(i - \frac{1}{2}, y)$$

$$\therefore \left. \frac{\partial^2 f(x,y)}{\partial x^2} \right|_{i,j} = \frac{\partial f(i + \frac{1}{2}, y)}{\partial x} - \frac{\partial f(i - \frac{1}{2}, y)}{\partial x}$$

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

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

$$\therefore \left. \frac{\partial^2 f(x,y)}{\partial x^2} \right|_{i,j} = f(i + 1, y) - 2f(i, y) + f(i - 1, y)$$

idem em relação a y e temos :

$$\left. \frac{\partial^2 f(x,y)}{\partial y^2} \right|_{i,j} = f(x, j + 1) - 2f(x, j) + f(x, j - 1)$$

$$\therefore \nabla^2 f(x,y) = f(x + 1, y) + f(x - 1, y) + f(x, y + 1) + f(x, y - 1) - 4f(x, y)$$

S.Furule 30/7/2010 - 20

Gradiente, Laplace,...

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

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

0	1	0
1	-4	1
0	1	0

Filtro tipo laplaciano em 1D

$\begin{matrix} i-1 & i & i+1 \\ 1 & -2 & 1 \end{matrix}$

$$H(z) = (z^{-1} - 2 + z)$$

$$H(w) = -2(1 - \cos(wT))$$

S.Furule 30/7/2010 - 21

Laplaciano da Gaussiana (LoG)

- Edge detector
- ✓ Gauss=>Smooth
- ✓ Laplace=>Zero crossing

LoG(x)

$$Gauss(x,y) = \exp\left(-\frac{x^2 + y^2}{2\sigma^2}\right)$$

$$g(x,y) = \nabla^2 Gauss(x,y) * f(x,y)$$

S.Furule 30/7/2010 - 22

Exemplos de operadores

- Exemplos no ImageJ
 - ✓ convolução
 - ✓ gradiente
 - ✓ realce de bordas
 - ✓ Laplaciano
 - ✓ bordas

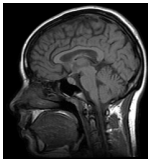
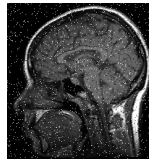
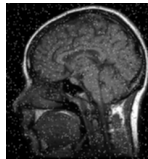
S.Furule 30/7/2010 - 23

Exemplo no ImageJ: filtro

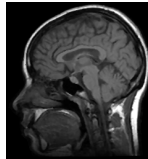
- Vamos implementar um plugin no ImageJ que:
 - ✓ Filtre os ruídos substituindo pela média dos 8-vizinhos (média entre 9 pixels)
 - ✓ Filtre os ruídos substituindo pela mediana (lista exercício 2)
- Como tratar as bordas?
 - ✓ Supondo valores zero fora da imagem
 - ✓ Valores espelhados p/ pixels fora da imagem
 - ✓ Tratar a exceção considerando apenas os pixels da imagem. Necessário considerar os sub-kernels adaptando os valores dos pesos, pois a normalização fica diferente

S.Furule 30/7/2010 - 24

Filtro adequado !

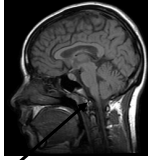
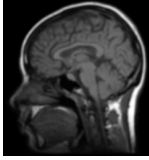
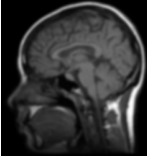
original Ruído salt pepper Filtro Média, $r=1$



Filtro Mediana, $R=1$

S.Furule 30/7/2010 - 25

Imagens: efeito de filtro (suavização)

Original
238 x 253, 8 bits
58kB

Filtrado, média, $r=2$ Filtrado, média, 2 x ($r=2$)

S.Furule 30/7/2010 - 26

SÍNTESE DA AULA

S.Furule 30/7/2010 - 27

Práticas

1. [P] Criar plugin para obter derivada parcial de $f(x,y)$ em x
2. [P] Criar plugin para obter o laplaciano de $f(x,y)$

S.Furule 30/7/2010 - 28

S.Furule 30/7/2010 - 29

S.Furule 30/7/2010 - 30



Lista de exercícios 2

- Entrega na próxima semana

S. Furule 30/7/2010 - 31