

PTC2892
Princípios da Formação e
Processamento de Imagens Médicas

Ambiente computacional: ImageJ e
plugins
Fundamentos da programação em
Proc. Imagens

Sérgio S Furuie

S. Furuie 30/7/2010 - 1

Plano da aula de hoje

- Exemplos de programação
 - Plugins e java
 - Operações simples
- Manipulando pixels e cores
- As aulas estão no site:
 - <http://moodle.stoa.usp.br>

S. Furuie 30/7/2010 - 2

Instalando ImageJ

- ImageJ
 - Download com java de <http://rsb.info.nih.gov/ij>
 - Por exemplo: Download ImageJ 1.42 bundled with 32-bit Java 1.6.0_14 (27MB)
 - Instale
 - Teste abrindo alguma imagem
 - Teste o compilador do plugin

S. Furuie 30/7/2010 - 3

Exemplo no ImageJ

- O que é plugin
- Como programar. Ver tutorial em:
 - ImageJ=>help => Documentation=> Developer Resources=> Writing ImageJ Plugins - A Tutorial (<http://www.imagingbook.com/index.php?id=102>)
- Regras básicas:
 - Nome do arquivo deve ter pelo menos um “_”
 - Arquivo deve ser depositado no subdiretório “plugins” ou em seu subdiretório
 - Nome do arquivo deve ser igual ao da classe acrescido de “.java”

S. Furuie 30/7/2010 - 4

Notação e representação de pixels

- Imagem f=> matriz
- Convenções:
 - canto superior esquerdo \Leftrightarrow 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), ip.putPixel(x,y,v)
 - Valor 0=> preto; valor 255=> branco

S. Furuie 30/7/2010 - 5

Notação no livro de Gonzalez

- Imagem => matriz => f[x][y]
- Convenções:
 - canto superior esquerdo \Leftrightarrow f[0][0] (linha=coluna=0)
 - Elemento na linha x, coluna y => f[x][y]
 - ip.getPixel(y,x)

S. Furuie 30/7/2010 - 6

Exemplo no ImageJ

- ❑ Vamos implementar um plugin simples no ImageJ que:
 - ✓ Inverta as intensidades da imagem, ou seja, os tons mais claros ficarão escuros e vice-versa
 - ✓ Por enquanto, trabalhar com imagem de 8bits (gray 8, 8G)

=> inverter_ptc2892_sf.java

S.Furue 30/7/2010 - 7

Pseudo-código

1. Obter a imagem: ip (imagem ativa no ij)
2. Obter o número de linhas e colunas da imagem
 - ✓ int nlin=ip.getHeight()
 - ✓ int ncol=ip.getWidth()
3. Para pixel em cada linha e cada coluna:
 - ✓ Ler pixel=ip.getPixel()
 - ✓ Modificar o valor do pixel
 - ✓ Armazenar novo valor
4. Refresh na imagem (display)

S.Furue 30/7/2010 - 8

Anatomia do programa em java

- ❑ // obtendo as dimensoes da imagem
- ❑ int nlin=ip.getHeight();
- ❑ int ncol=ip.getWidth();
- ❑ // realizando a inversao de intensidade
- ❑ int pixel;
- ❑ for(int y=0; y< nlin; y++)
- ❑ for(int x=0; x<ncol; x++)
- ❑ {
- ❑ pixel=ip.getPixel(x,y); // lendo pixel
- ❑ pixel= pixel & 0x000000ff; // AND: zera outros bytes
- ❑ pixel= pixel ^ 0x000000ff; // XOR: complemento
- ❑ ip.putPixel(x, y, pixel); // store pixel
- ❑ }
- ❑ imp.show(); // refresh display

S.Furue 30/7/2010 - 9

```

1 // Inverter Gray8.java S.Furue EPUSP
2 // Inverte a intensidade de imagens tipo cinza, 8 bits. OK.
3 import ij.*;
4 import ij.process.*;
5 import ij.plugin.filter.*;
6 public class Inverter_Gray8 implements PlugInFilter {
7     ImagePlus imp;
8     public int setup(String arg, ImagePlus imp) {
9         this.imp = imp;
10        return DOES_8G;
11    }
12    public void run(ImageProcessor ip) {
13        // obtendo as dimensoes da imagem
14        int nlin=ip.getHeight(); // obtendo dimensoes
15        int ncol=ip.getWidth();
16        int pixel;
17        // realizando a inversao de intensidade
18        for(int y=0; y< nlin; y++)
19            for(int x=0; x<ncol; x++)
20                {
21                    pixel=ip.getPixel(x,y); // lendo pixel
22                    pixel= pixel & 0x000000ff; // AND: zera os outros bytes
23                    pixel= pixel ^ 0x000000ff; // XOR: complemento
24                    ip.putPixel(x, y, pixel); // store pixel
25                }
26        imp.show();
27    }
28 }
  
```

S.Furue 30/7/2010 - 10

Invertendo a intensidade (geral)

```

❑ public class inverter_ptc2892_sf implements PlugInFilter {
❑     ImagePlus imp;
❑     public int setup(String arg, ImagePlus imp) {
❑         this.imp = imp;
❑         return DOES_ALL;
❑     }
❑     public void run(ImageProcessor ip) {
❑         // obtendo as dimensoes da imagem
❑         int nlin=ip.getHeight();
❑         int ncol=ip.getWidth();
❑
❑         // lendo a imagem
❑         byte[] pixels=(byte[]) ip.getPixels();
❑
❑         // realizando a inversao de intensidade
❑         for(int x=0; x< nlin; x++)
❑             for(int y=0; y<ncol; y++)
❑                 {
❑                     ip.set(x, y, -ip.get(x,y));
❑                 }
❑     }
❑ }
  
```

S.Furue 30/7/2010 - 11

Invertendo a intensidade (2)

```

❑ public class inverter_ptc2892_sf implements PlugInFilter {
❑     ImagePlus imp;
❑     public int setup(String arg, ImagePlus imp) {
❑         this.imp = imp;
❑         return DOES_ALL;
❑     }
❑     public void run(ImageProcessor ip) {
❑         // obtendo as dimensoes da imagem
❑         int nlin=ip.getHeight();
❑         int ncol=ip.getWidth();
❑
❑         // lendo a imagem
❑         byte[] pixels=(byte[]) ip.getPixels();
❑
❑         // realizando a inversao de intensidade
❑         for(int x=0; x< nlin; x++)
❑             for(int y=0; y<ncol; y++)
❑                 {
❑                     pixels[x*ncol + y]= (byte) (pixels[x*ncol+y] ^ 0xff); //xor
❑                 }
❑     }
❑ }
  
```

S.Furue 30/7/2010 - 12

Prática e programação em sala

- [P] Visualizar apenas a banda azul da imagem "Cardio.dcm" (RGB)
- [P] Visualizar apenas a banda verde da imagem "Cardio.dcm" (RGB)
- [P] Visualizar apenas a banda vermelha da imagem "Cardio.dcm" (RGB)
- [P] Transformar a imagem "Cell_Colony.jpg" (monocromática cinza) em monocromática vermelha. Compare usando LUT-Look Up Table

S.Furue 30/7/2010 - 13

```
1 // RGB_para_azul.java S.Furue EPUSP
2 // Converte imagem de RGB em azul. OK.
3 import ij.*;
4 import ij.process.*;
5 import ij.plugin.filter.*;
6 public class RGB_somente_azul implements PlugInFilter {
7     ImagePlus imp;
8     public int setup(String arg, ImagePlus imp) {
9         this.imp = imp;
10        return DOES_RGB;
11    }
12    public void run(ImageProcessor ip) {
13        // obtendo as dimensoes da imagem
14        int nlin=ip.getHeight();
15        int ncol=ip.getWidth();
16        // mantendo apenas a azul
17        int pixel;
18        for(int r=0; r< nlin; r++)
19            for(int c=0; c<ncol; c++)
20            {
21                pixel=ip.getPixel(c, r);
22                pixel = pixel & 0xff; // Deixar apenas o B (mascara)
23                ip.putPixel(c, r, pixel); // e atualizar o valor do pixel
24            }
25        imp.updateAndDraw();
26    }
27 }
```

S.Furue 30/7/2010 - 14

Anatomia do programa: Gray8_para_RGB

- // obtendo as dimensoes da imagem
- int nlin=ip.getHeight();
- int ncol=ip.getWidth();
- // transformando em vermelho
- int pixel;
- for(int r=0; r< nlin; r++)
- for(int c=0; c<ncol; c++)
- {
- pixel=ip_RGB.getPixel(c, r);
- pixel = pixel & 0xff0000; // Deixar apenas o Red
- ip_RGB.putPixel(c, r, pixel); // e atualizar o valor do pixel
- }
- imp_RGB.updateAndDraw();

S.Furue 30/7/2010 - 15

```
1 // Gray8_para_Red.java S.Furue EPUSP
2 // Converte imagem de 8bits em RGB (cinza) e depois em vermelho. OK.
3 import ij.*;
4 import ij.process.*;
5 import ij.plugin.filter.*;
6 public class Gray8_para_Red implements PlugInFilter {
7     ImagePlus imp;
8     public int setup(String arg, ImagePlus imp) {
9         this.imp = imp;
10        return DOES_RGB;
11    }
12    public void run(ImageProcessor ip) {
13        // realizando a conversao de 8b p/ RGB
14        ColorProcessor ip_RGB=(ColorProcessor)ip.convertToRGB(); // R=G=B=pixel
15        ImagePlus imp_RGB=new ImagePlus("RGB", ip_RGB);
16        imp_RGB.show();
17        IJ.showMessage("RGB em cinza. Click p/ continuar"); // R=G=B=pixel
18    }
19    // obtendo as dimensoes da imagem
20    int nlin=ip.getHeight();
21    int ncol=ip.getWidth();
22    // transformando em vermelho
23    int pixel;
24    for(int r=0; r< nlin; r++)
25        for(int c=0; c<ncol; c++)
26        {
27            pixel=ip_RGB.getPixel(c, r);
28            pixel = pixel & 0xff0000; // Deixar apenas o R (mascara)
29            ip_RGB.putPixel(c, r, pixel); // e atualizar o valor do pixel
30        }
31    imp_RGB.updateAndDraw();
32    }
33 }
```

S.Furue 30/7/2010 - 16