

Universidade Federal do Rio Grande do Sul
INF01046 – Fundamentos de Processamento de Imagens

Software de Processamento de Imagens – Parte II

Jorge Wichrowski Krieger de Mello – 143283
jwkmhdr@hotmail.com

Site : www.inf.ufrgs.br/~jwkmello

www.inf.ufrgs.br/~jwkmello/tudo/trab_fpi.html

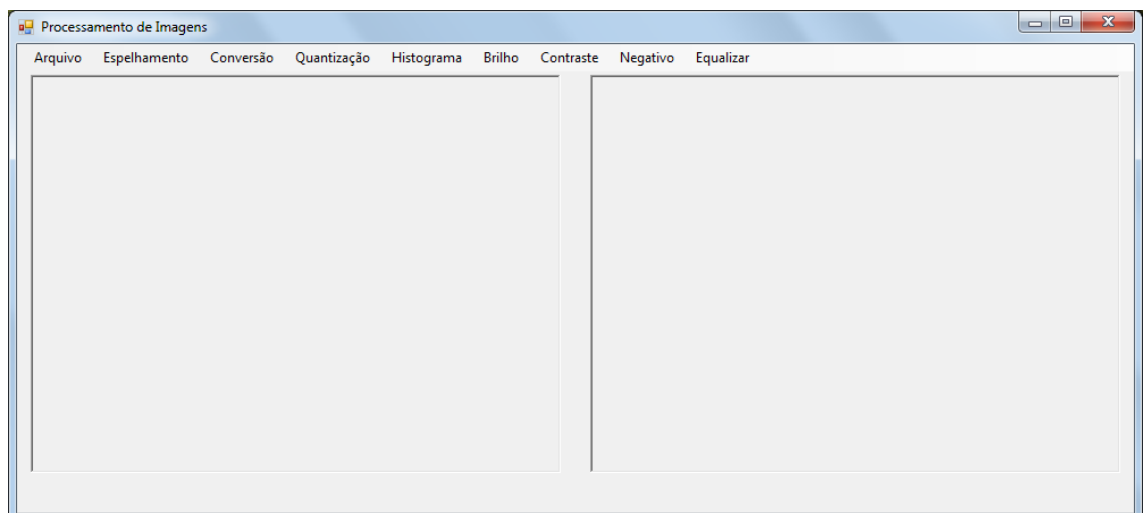
Prof. Manuel M. Oliveira

1. INTRODUÇÃO

Este trabalho me possibilitou uma ótima experiência de aprendizado, pude trabalhar um pouco com processamento de imagens, pude aprender uma linguagem que não havia tido muito contato C#, entendi um pouco mais sobre formatos das imagens, usei o Visual Studio uma das melhores ferramentas na minha opinião, mas que nem sempre é possível usar nos trabalhos escolares. Também estou tendo a oportunidade de melhorar um pouco meu site pessoal com mais conteúdo.

Neste relatório vou mostrar o desenvolvimento e o resultado dos 5 itens pedidos na definição do trabalho. Mas antes gostaria de explicar seu funcionamento e ressaltar que este programa é possivelmente o mais intuitivo de se usar já feito para esta disciplina. Tudo acontece em uma janela auto dimensionável, com todas as opções visíveis para o usuário. Quando você escolhe uma opção ele abre uma janela para que você selecione uma imagem, após selecionar uma imagem ele mostra ela na janela 1 e o resultado da opção escolhida na janela 2. Você pode salvar esta imagem digitando CTRL + S ou simplesmente Arquivo -> Salvar. Para a quantização existe um campo de texto no próprio menu com o número que você deseja escolher, o valor default é 8. O programa funciona com vários tipos de imagens .jpg, .png, .bmp, entretanto ele salva apenas no formato .jpg. O programa também mantém o valor de Alfa das imagens, ou seja, não modifica sua transparência. Ele irá funcionar em qualquer sistema operacional Windows que tenha o net.framework4.0 instalado.

Agora o programa também mostra o Histograma, Brilho, Contraste, Negativo e Equaliza a imagem. A interface continua intuitiva no entanto esta pouco prática, mas o objetivo é justamente ver os conceitos de processamento de imagens e não de se vender o software. Então considero que a intuitividade é mais importante que a prática.



2. Desenvolvimento

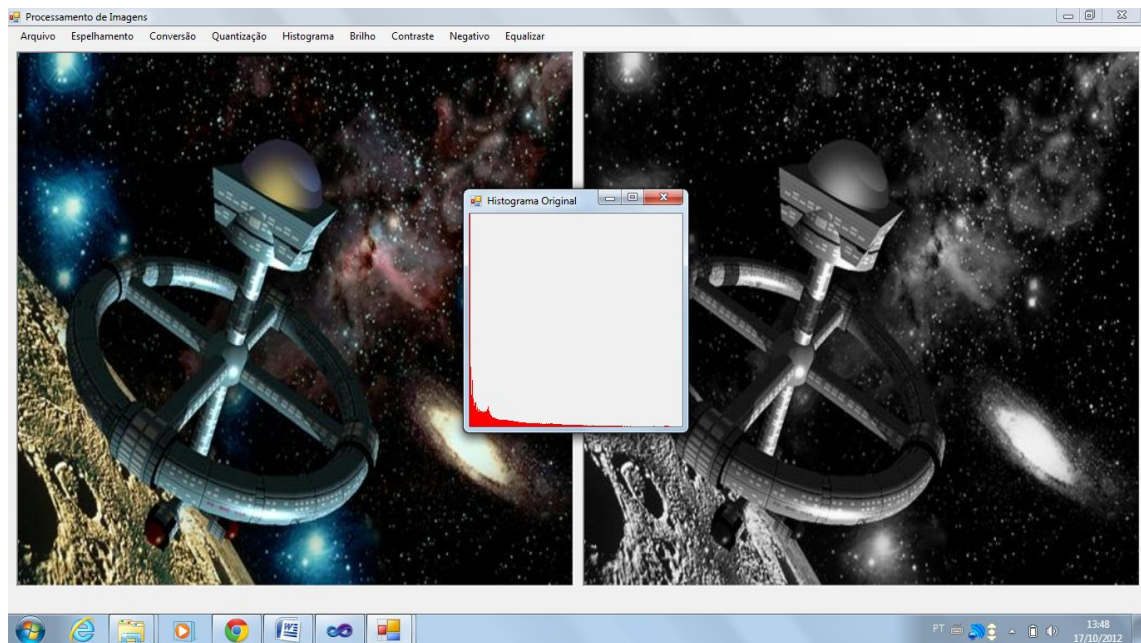
2.1 Exibir Histograma:

A imagem é convertida para tons de cinza usando a função do trabalho anterior só que agora o valor dos tons de cinza são postos em um vetor que é passado ao novo formulário que irá exibir o histograma.

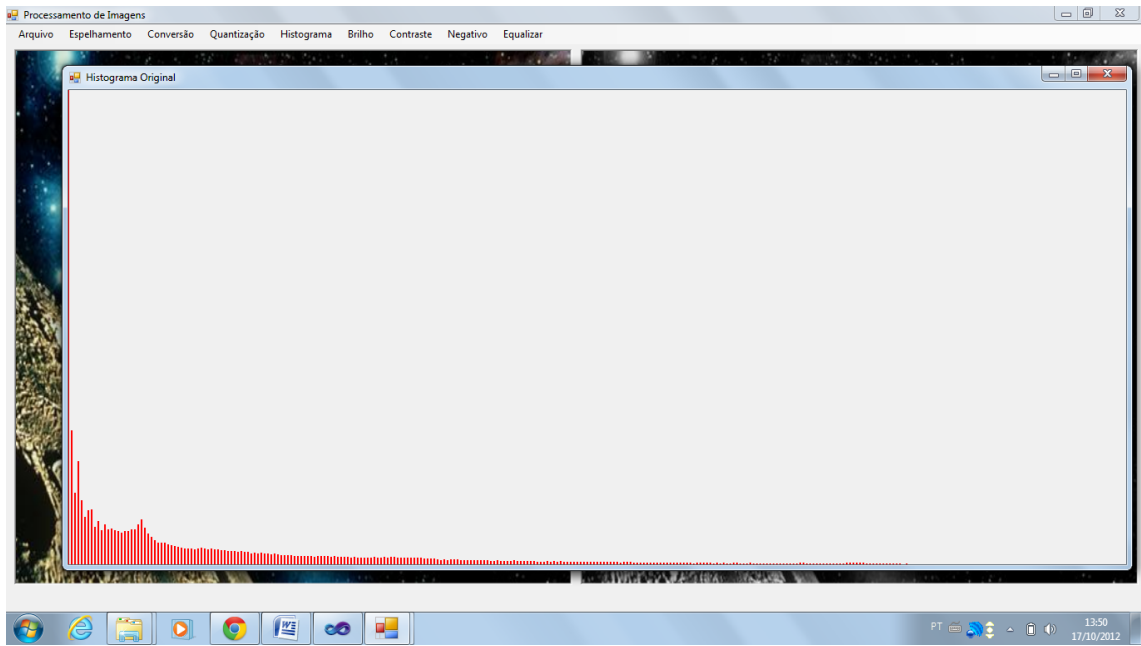
Código Fonte:

```
pictureBox1.Image = Image.FromFile(openFileDialog1.FileName);  
imagem = new Bitmap(Image.FromFile(openFileDialog1.FileName));  
nome = openFileDialog1.FileName;  
int h = imagem.Width;  
int v = imagem.Height;  
nova_imagem = GrayScaleFilter(imagem);  
  
pictureBox2.Image = nova_imagem;  
  
Form2 newForm2 = new Form2();  
newForm2.Vcin = Vcin;  
newForm2.Show();
```

Resultado:



Veja que o histograma também é dimensionável:



2.2 Alterar Brilho:

Código Fonte:

```
nova_imagem = new Bitmap(imagem.Width, imagem.Height);
int i, u;
for (i = 0; i < v; i++)
{
    for (u = 0; u < h; u++)
    {
        faixaR = imagem.GetPixel(u, i).R + briR;
        if (faixaR > 255)
            faixaR = 255;
        else if (faixaR < 0)
            faixaR = 0;

        faixaG = imagem.GetPixel(u, i).G + briG;
        if (faixaG > 255)
            faixaG = 255;
        else if (faixaG < 0)
            faixaG = 0;

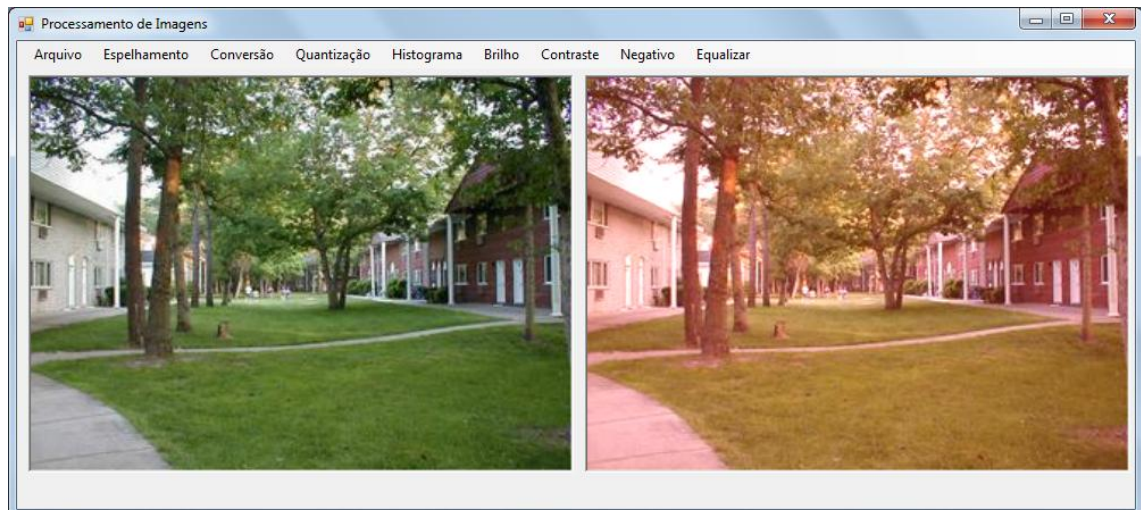
        faixaB = imagem.GetPixel(u, i).B + briB;
        if (faixaB > 255)
            faixaB = 255;
        else if (faixaB < 0)
            faixaB = 0;

        int trasn = imagem.GetPixel(u, i).A;

        nova_imagem.SetPixel(u, i, Color.FromArgb(trasn,
faixaR, faixaG, faixaB)); }}

```

Resultado: (Alterando o brilho do canal Vermelho)



2.3 Contraste:

Código Fonte:

```
for (i = 0; i < v; i++)
{
    for (u = 0; u < h; u++)
    {
        faixaR = Convert.ToInt32(imagem.GetPixel(u, i).R *
conR);

        if (faixaR > 255)
            faixaR = 255;
        else if (faixaR < 0)
            faixaR = 0;

        faixaG = Convert.ToInt32(imagem.GetPixel(u, i).G *
conG);

        if (faixaG > 255)
            faixaG = 255;
        else if (faixaG < 0)
            faixaG = 0;

        faixaB = Convert.ToInt32(imagem.GetPixel(u, i).B *
conB);

        if (faixaB > 255)
            faixaB = 255;
        else if (faixaB < 0)
            faixaB = 0;

        int trasn = imagem.GetPixel(u, i).A;

        nova_imagem.SetPixel(u, i, Color.FromArgb(trasn,
faixaR, faixaG, faixaB));}}}
```

Resultado:

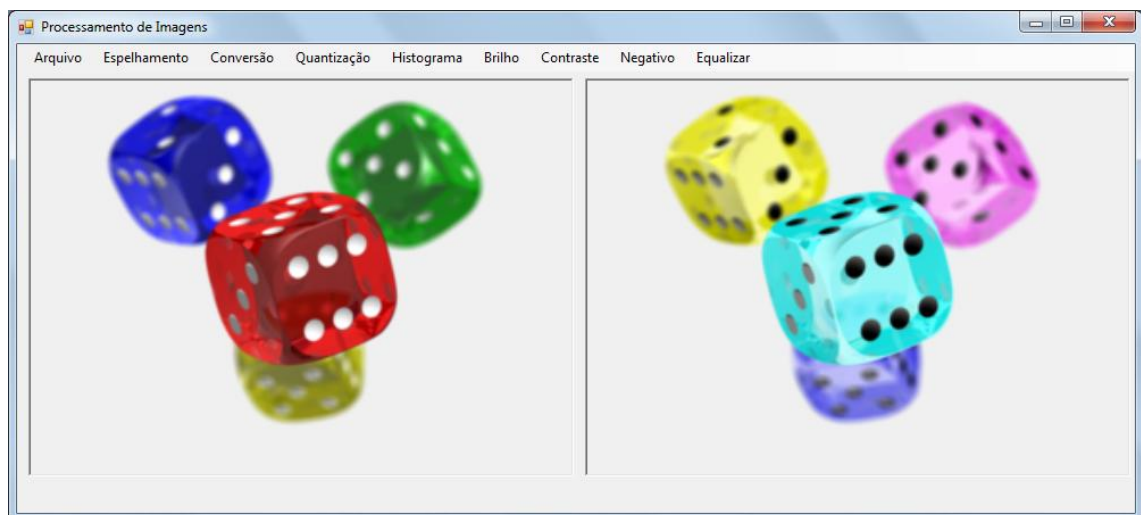


2.4 Negativo

Código Fonte:

```
for (i = 0; i < v; i++)
{
    for (u = 0; u < h; u++)
    {
        faixaR = 255 - imagem.GetPixel(u, i).R;
        faixaG = 255 - imagem.GetPixel(u, i).G;
        faixaB = 255 - imagem.GetPixel(u, i).B;
        int trasn = imagem.GetPixel(u, i).A;
        nova_imagem.SetPixel(u, i, Color.FromArgb(trasn,
faixaR, faixaG, faixaB));
    }
}
```

Resultado :



2.5 Equalizar Histograma

É realizado a contagem dos tons de red, blue e green para a imagem, encima deles é aplicado a fórmula fornecida em sala de aula, para gerar o novo histograma de cores red, blue e green. Nós gráficos o que é mostrado é o histograma de escalas de cinza apenas.

Código Fonte:

```
try {
    int h=0, v=0, i =0;

    if (openFileDialog1.ShowDialog() == System.Windows.Forms.DialogResult.OK)
    {
        pictureBox1.Image = Image.FromFile(openFileDialog1.FileName);
        imagem = new Bitmap(Image.FromFile(openFileDialog1.FileName));
        nome = openFileDialog1.FileName;
        h = imagem.Width;
        v = imagem.Height;
        nova_imagem = GrayScaleFilter(imagem);

        Form2 newForm2 = new Form2();
        newForm2.Vcin = Vcin;
        newForm2.Show();
    }

    double a = 255.0 / (Convert.ToDouble(h) * Convert.ToDouble(v));
    int [] Vcin_cumR = new int[256];
    int [] Vcin_cumG = new int[256];
    int [] Vcin_cumB = new int[256];

    int[] VcinR= new int[256];
    int[] VcinG = new int[256];
    int[] VcinB = new int[256];

    int x=0,y=0;
    for (y = 0; y < imagem.Height; y++)
        for (x = 0; x < imagem.Width; x++)
        {
            Color c = imagem.GetPixel(x, y);
            VcinR[c.R]++;
            VcinG[c.G]++;
            VcinB[c.B]++;
        }

    Vcin_cumR[0] = Convert.ToInt32(a * Convert.ToDouble(VcinR[0]));
    for (i = 1; i <= 255; i++)
    {
        Vcin_cumR[i] = Vcin_cumR[i - 1] + Convert.ToInt32(a *
Convert.ToDouble(VcinR[i]));
        if (Vcin_cumR[i] > 255)
            Vcin_cumR[i] = 255;
        else if (Vcin_cumR[i] < 0)
            Vcin_cumR[i] = 0;
    }

    Vcin_cumG[0]= Convert.ToInt32(a * Convert.ToDouble(VcinG[0]));
    for (i = 1; i <= 255; i++)
    {
        Vcin_cumG[i] = Vcin_cumG[i - 1] + Convert.ToInt32(a *
Convert.ToDouble(VcinG[i]));
        if (Vcin_cumG[i] > 255)
            Vcin_cumG[i] = 255;
        else if (Vcin_cumG[i] < 0)
            Vcin_cumG[i] = 0;
    }

    Vcin_cumB[0] = Convert.ToInt32(a * Convert.ToDouble(VcinB[0]));
```

```

    for (i = 1; i <= 255; i++)
    {
        Vcin_cumB[i] = Vcin_cumB[i - 1] + Convert.ToInt32(a *
Convert.ToDouble(VcinB[i]));
        if (Vcin_cumB[i] > 255)
            Vcin_cumB[i] = 255;
        else if (Vcin_cumB[i] < 0)
            Vcin_cumB[i] = 0;
    }

nova_imagem1 = new Bitmap(h, v);

for (x = 0; x < h; x++){
    for (y = 0; y < v; y++) {
        int trasn = imagem.GetPixel(x, y).A;
        int auxR = imagem.GetPixel(x, y).R;
        int auxG = imagem.GetPixel(x, y).G;
        int auxB = imagem.GetPixel(x, y).B;
        int r = Vcin_cumR[auxR];
        int g = Vcin_cumG[auxG];
        int b = Vcin_cumB[auxB];
        nova_imagem1.SetPixel(x, y, Color.FromArgb(trasn, r, g, b));}}

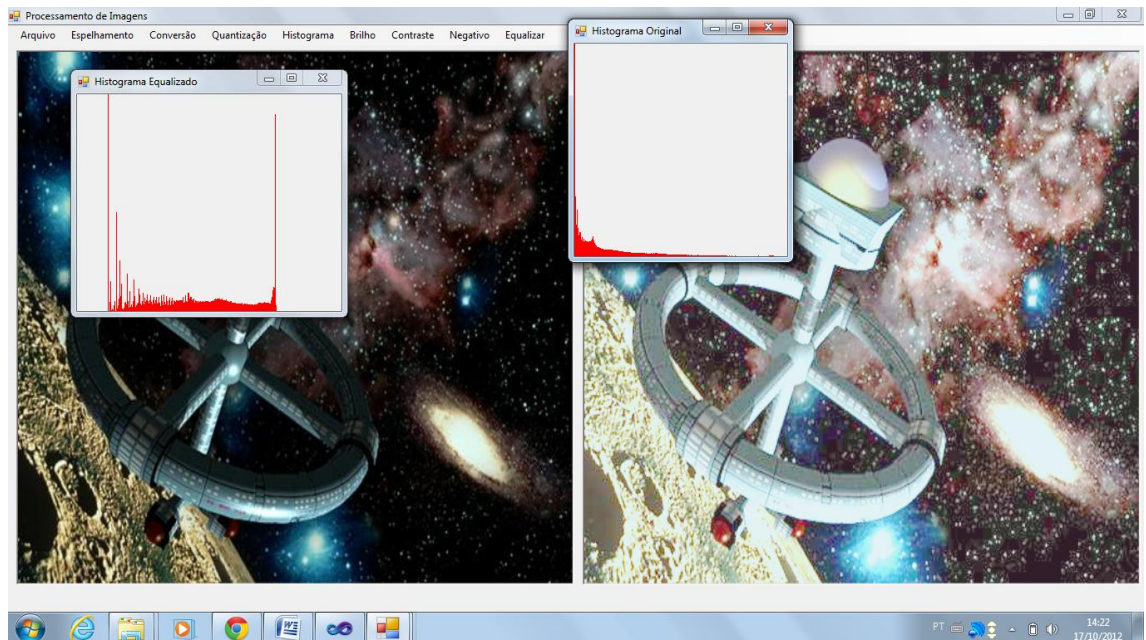
pictureBox2.Image = nova_imagem1;

GrayScaleFilter2(nova_imagem1);
Form3 newForm3 = new Form3();
newForm3.Vcin1 = Vcin2;
newForm3.Show();

    }
catch { }

```

Resultado:



3. Conclusão

Pude trabalhar vários conceitos, fazer um programa divertido, usar aquilo que eu julgo o melhor para trabalhar com software. Enfim usei muitos conhecimentos que aprendi durante a faculdade, foi realmente muito bom este trabalho e prazeroso.

Site : www.inf.ufrgs.br/~jwkmello

www.inf.ufrgs.br/~jwkmello/tudo/trab_fpi.html