

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

Software de Processamento de Imagens – Parte I

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

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

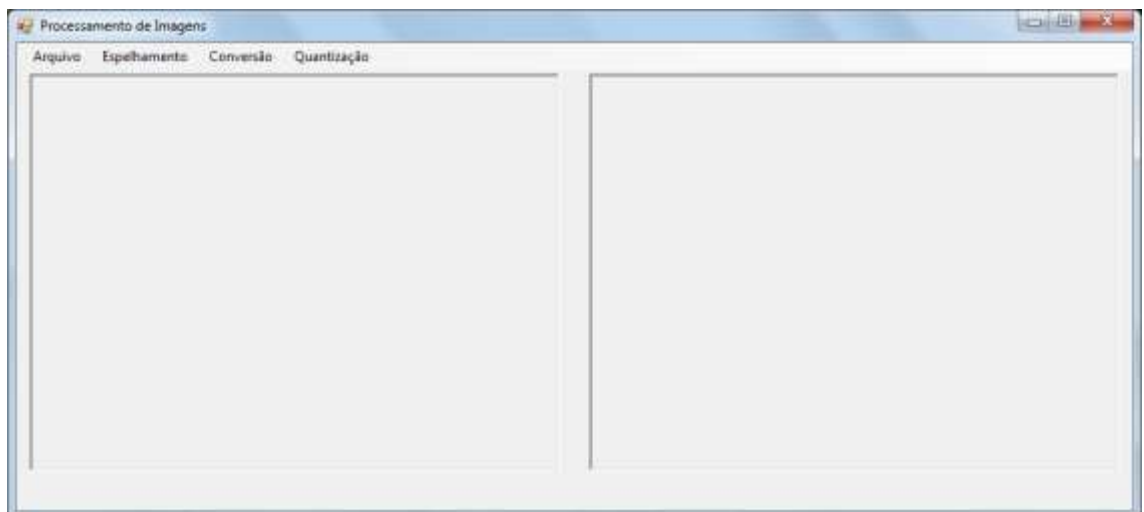
www.inf.ufrgs.br/~jwkmello/tudo/trab_fpi.html

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.

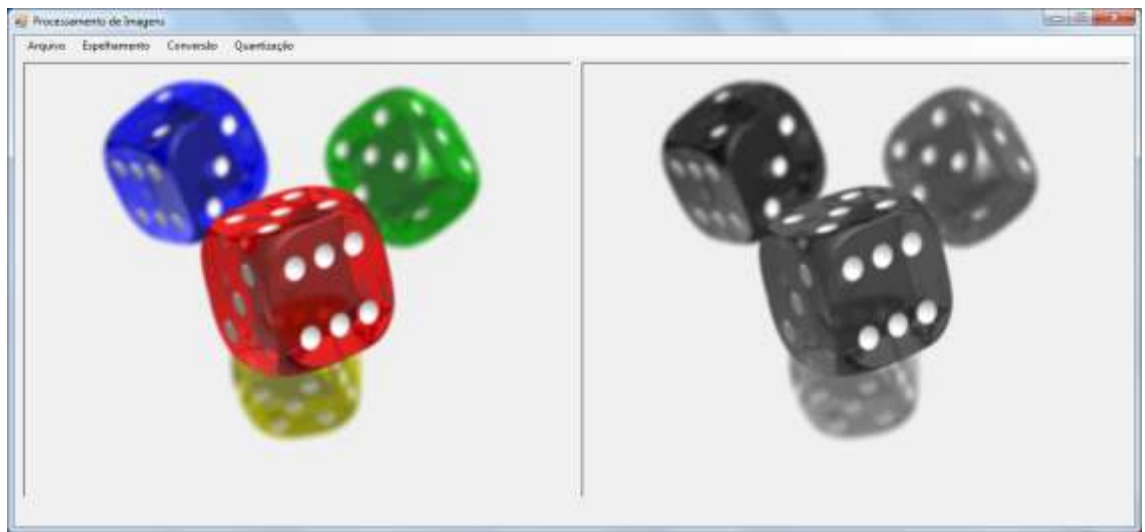
Neste relatório vou mostrar o desenvolvimento e o resultado dos 5 itens pedidos na definição do trabalho. Mas antes gostaria de explicar seu funcionamento e ressaltar que este programa é possivelmente o mais intuitivo de se usar já feito para esta disciplina. Tudo acontece em uma janela auto dimensionável, 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.



Vejamos abaixo a janela auto dimensionável:



Não alteração do fator alpha (imagem .png):



2. Desenvolvimento

2.1 Espelhamento Horizontal:

Ambos espelhamentos foram feitos baseados na troca pixel por pixel, pois a biblioteca usada não tinha uma função que nos desse a linha ou coluna inteira da matriz, se armazenássemos os pixels numa matriz antes de trocar as linhas e colunas seria mais demorado pois teríamos que percorrer todos pixels uma vez de qualquer jeito para poder armazená-los.

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 = new Bitmap(imagem.Width, imagem.Height);
int i, u;
for (i = 0; i < v; i++)
{
    for (u = 0; u < h; u++)
    {
        nova_imagem.SetPixel(h - u - 1, i, imagem.GetPixel(u, i));
    }
}
pictureBox2.Image = nova_imagem;
```

Resultado:



2.2 Espelhamento Vertical:

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 = new Bitmap(imagem.Width, imagem.Height);
int i, u;
for (u = 0; u < h; u++)
{
    for (i = 0; i < v; i++)
    {
        nova_imagem.SetPixel(u, v - i - 1, imagem.GetPixel(u, i));
    }
}
pictureBox2.Image = nova_imagem;
```

Resultado:



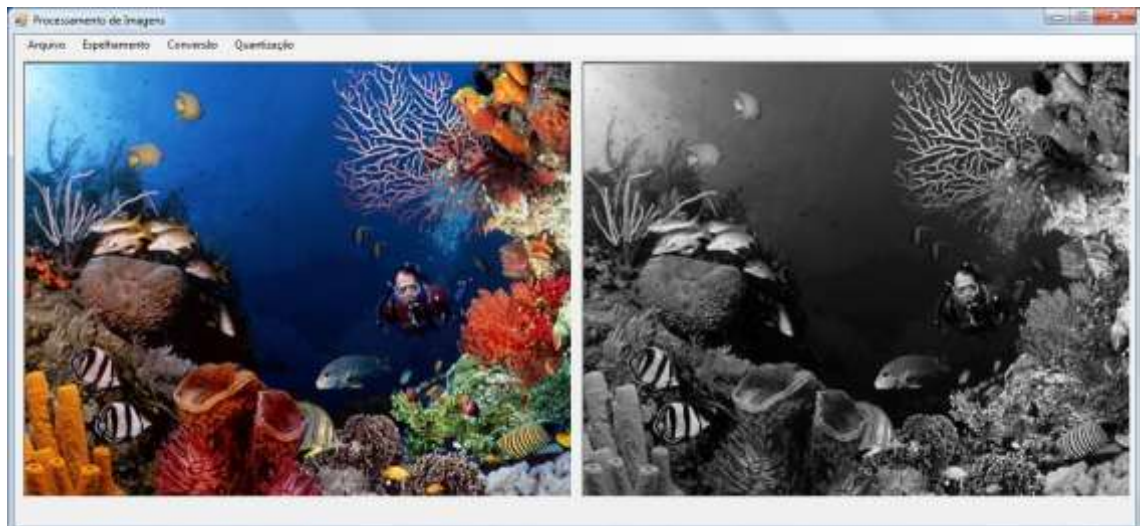
2.3 Imagem em Tons de Cinza:

Foi utilizado a fórmula para conversão RGB-YCbCr e o valor de Y foi colocado para todos os canais.

Código Fonte:

```
Bitmap grayScale = new Bitmap(image.Width, image.Height);
for (Int32 y = 0; y < grayScale.Height; y++)
    for (Int32 x = 0; x < grayScale.Width; x++)
    {
        Color c = image.GetPixel(x, y);
        Int32 gs = (Int32)(c.R * 0.3 + c.G * 0.59 + c.B * 0.11);
        int trasn = imagem.GetPixel(x, y).A;
        grayScale.SetPixel(x, y, Color.FromArgb(trasn,gs, gs, gs));
    }
return grayScale;
```

Resultado:



2.4 Quantização:

Dependendo do valor do pixel ele recebe o valor intermediário para o grupo ao qual ele pertence, o número de grupos depende do usuário que pode escolher de 1 a 256. Se ele escolher 1 a imagem vai ficar toda cinza se escolher 256 não haverá nenhuma alteração.

Código Fonte:

```
try
{
    int quan = Convert.ToInt32(valor.Text);

    if (quan > 256)
        quan = 256;
    else if (quan <= 0)
        quan = 1;

    int dist = 256 / quan;

    if (openFileDialog1.ShowDialog() ==
System.Windows.Forms.DialogResult.OK)
    {
        pictureBox1.Image = Image.FromFile(openFileDialog1.FileName);
        imagem = new
Bitmap(Image.FromFile(openFileDialog1.FileName));
        nome = openFileDialog1.FileName;
        int h = imagem.Width;
        int v = imagem.Height;
        int novo_valorR = 0;
        int novo_valorG = 0;
        int novo_valorB = 0;
        int faixaR = 0;
        int faixaG = 0;
        int faixaB = 0;
        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 / dist;
                faixaG = imagem.GetPixel(u, i).G / dist;
                faixaB = imagem.GetPixel(u, i).B / dist;
                int trasn = imagem.GetPixel(u, i).A;

                novo_valorR = faixaR * dist + (dist / 2);
                novo_valorG = faixaG * dist + (dist / 2);
                novo_valorB = faixaB * dist + (dist / 2);

                if (novo_valorR > 255)
                    novo_valorR = 255;

                if (novo_valorG > 255)
                    novo_valorG = 255;

                if (novo_valorB > 255)
                    novo_valorB = 255;
            }
        }
    }
}
```

```
        nova_imagem.SetPixel(u, i,  
Color.FromArgb(trasn,novo_valorR, novo_valorG, novo_valorB));  
    }  
}  
pictureBox2.Image = nova_imagem;  
}  
} catch { }
```

Resultado para 8 tons:



Resultado para 2 tons:



Resultado para 2 tons sobre imagem colorida:



2.4 Salvamento:

A opção salvar esta disponível e pelo que vi toda vez que salvamos um .jpg ele perde qualidade, pois é aplicada algum tipo de compressão. Mas no meu caso o como eu recupero todos os pixels e depois salvo de novo e provavelmente minha biblioteca não esta realizando a melhor compressão o tamanho das imagens aumenta. Num teste inicial eu apenas copieei o conteúdo de um arquivo de imagem para outro arquivo neste caso como era de se esperar o tamanho se manteve o mesmo.

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