

# Processamento Digital de Imagens - Trabalho Prático

Juliana Amorim Guimarães  
Email: guimaraesajuliana@gmail.com  
Centro de Ciências Exatas, Naturais e da Saúde  
Departamento de Computação  
Universidade Federal do Espírito Santo - Campus Alegre

17 de Julho de 2017

## 1 Introdução

Trabalho de programação da disciplina de Processamento Digital de Imagens do curso de Sistemas de Informação, semestre 2017/1. A estimativa é que o trabalho consuma de 6 a 10 horas de programação para ser concluído.

### 1.1 Trabalho Prático

Esta etapa consiste em preparar a solução, sobre o tema escolhido, de acordo com a bibliografia da disciplina e com os conceitos vistos em sala de aula. No início de todos os arquivos deverá haver um comentário identificando o arquivo, o trabalho e o autor. Por exemplo:

```
%=====
% \file solucao.mat
% \title Trabalho de PDI – 2017/1
% \author Fulano de Souza
% \date 2017-07-12
%
% Este arquivo implanta a solucao do problema
% proposto usando o algoritmo ABC. Para usar a
% solucao desenvolvida, o programa deve ser executado
% da seguinte forma:
% <comando de compilacao>
% <comando de execucao com lista de parametros>
%=====
```

Obs.: Se identificado plágio, o trabalho será **zerado**.

## 1.2 Informações Importantes

- Este documento pode sofrer alteração sempre que a professora julgar necessário.
- É responsabilidade do aluno verificar se possui a versão mais atualizada deste documento.
- A data de entrega do trabalho é 28 de julho de 2017, às 23:50h.
- O trabalho pode ser desenvolvido individualmente ou em dupla.
- Se desenvolvido em dupla, a professora dará a nota do trabalho que deverá ser distribuído pela dupla e informado à professora em prazo hábil para o fechamento das notas.
- A entrega do trabalho deverá ser feita pela sala da disciplina no Moodle.
- Um único arquivo ZIP deverá ser “carregado” no Moodle. Este ZIP deverá conter todos os arquivos relacionados ao trabalho.

## 2 Problema

Um recurso muito útil no processamento de imagens é a possibilidade de alterar a cor de determinado objeto a fim de destacá-lo, dentro de determinado contexto. Faça um programa que leia imagens e permita ao usuário trocar uma cor que desejar na imagem.

A Figura 1 mostra os modelos de cores HSI, e os passos a seguir descreve como alterar a cor de uma imagem de entrada de vermelho para azul.



Figura 1: Modelos de cores HSI

1. Ler a imagem de entrada.
2. Converter a imagem RGB para HSI.
3. Trocar apenas o valor da cor, no modelo HSI.  
 Por exemplo: A cor vermelha no HSI se encontra entre os ângulos  $-30^\circ$  e  $30^\circ$ . A cor azul se encontra exatamente na direção oposta, já que é uma cor complementar, como pode ser visto em na Figura 1.
4. Finalmente, chama-se a função de conversão do HSI para RGB passando como parâmetro esta nova imagem HSI alterada.

Os trechos de código a seguir mostram scripts de conversão de modelos de cores que podem ser usados para facilitar esse processo.

---

```
function hsi_image = rgb2hsi(rgb_image)
R = double(rgb_image(:, :, 1))/255.0;
G = double(rgb_image(:, :, 2))/255.0;
B = double(rgb_image(:, :, 3))/255.0;
I = (R + G + B)/3;
temp = (R - G).*(R - G) + (R - B).*(G - B);
temp(:, :) = sqrt(double(temp(:, :)));
[L, C] = size(temp);
S = zeros([L C]); S = double(S);
H = S;
for i=1:L
    for j = 1:C
        if (R(i,j) + G(i,j) + B(i,j)) ~= 0
            S(i,j)=1-(3*min(min(R(i,j),G(i,j)),B(i,j)))/(R(i,j)+G(i,j)+B(i,j));
        end;
        if (temp(i, j) > 0)
            H(i, j) = acos(0.5*(2*R(i,j) - G(i,j) - B(i,j))/temp(i, j));
            if (B(i,j) > G(i,j))
                H(i,j) = 2*pi - H(i, j);
            end;
        end;
    end;
end;
H = H/(2*pi);
figure, imshow(uint8(H*255))
figure, imshow(uint8(S*255))
figure, imshow(uint8(I*255))
hsi_image = zeros(size(rgb_image));
hsi_image(:, :, 1) = H;
hsi_image(:, :, 2) = S;
```

```
hsi_image(:, :, 3) = I;
```

---

---

```
function rgb_image = hsi2rgb(hsi_image)
H = double(hsi_image(:, :, 1)); H = 2*pi*H;
S = double(hsi_image(:, :, 2));
I = double(hsi_image(:, :, 3));
[L C] = size(H);
R = zeros([L C]);
G = R;
B = R;

for i = 1:L
    for j = 1:C
        if (H(i, j) >= 0) && (H(i, j) < (2*pi/3))
            B(i, j) = I(i, j) * (1 - S(i, j));
            R(i, j) = I(i, j) * (1 + S(i, j) * cos(H(i, j))/cos((2*pi/6) - ...
                H(i, j))); G(i, j) = 3 * I(i, j) - (R(i, j) + B(i, j));
        end;
        if (H(i, j) >= (2*pi/3)) && (H(i, j) < (4*pi/3))
            H(i, j) = H(i, j) - (2*pi/3);
            R(i, j) = I(i, j) * (1 - S(i, j));
            G(i, j) = I(i, j) * (1 + S(i, j) * cos(H(i, j))/cos((2*pi/6) - ...
                H(i, j)));
            B(i, j) = 3 * I(i, j) - (R(i, j) + G(i, j));
        end;
        if (H(i, j) >= (4*pi/3)) && (H(i, j) < (2*pi))
            H(i, j) = H(i, j) - (4*pi/3);
            G(i, j) = I(i, j) * (1 - S(i, j));
            B(i, j) = I(i, j) * (1 + S(i, j) * cos(H(i, j))/cos((2*pi/6) - ...
                H(i, j)));
            R(i, j) = 3 * I(i, j) - (G(i, j) + B(i, j));
        end;
    end;
end;
```

---