

Relatório 1 – Visão Computacional.
Aluno: Eduardo Mafra Pereira.
Orientador: Leonardo Mejia Ricon.

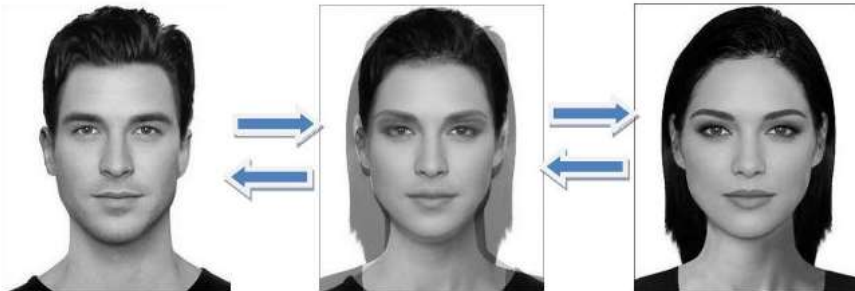
Etapa 1:

O objetivo desta etapa é realizar transição de uma imagem para outra através do software Matlab. Foram utilizadas duas imagens monocromáticas em tons de cinza chamadas de "ManGray.jpg" e "WomanGray.jpg". Para implementação do programa utilizamos uma soma ponderada das duas matrizes através de um laço for, onde a cada interação uma matriz "diminui" sua intensidade e outra "aumenta".

As variáveis Iman e Iwon armazenam as matrizes de pixels ManGray.jpg e WomanGray respectivamente.

```
clc
Iman = imread('ManGray.jpg');
Iwon = imread('WomanGray.jpg');
ilustrativa= imread('amostral.jpg');
%laço de interação entre as matrizes.
for i = 0:0.1:1
    % Soma ponderada das matrizes "Imag" e "Img", e criação de uma terceira
    % matriz "imagem".
    imagem = Iman*i + Iwon*(1-i);
    imshow(imagem);
    pause(0.3)
end
imshow(ilustrativa);
title('Imagem Ilustrativa da Transformação'); %Isto foi realizado apenas para proporcionar um melhor
%entendimento do relatório.
```

Imagem Ilustrativa da Transformação



Etapa 2:

A segunda etapa deste relatório teve o objetivo de construir um telescópio virtual por meio de interações entre as imagens "PanoramicGray.jpg" e "TelescopicGray.jpg", além disto foi solicitado uma inversão de cor da imagem "TelescopicGray.jpg" e a aplicação de transformações homogêneas na imagem gerada como resultado do telescópio virtual.

```
clc
%insere imagens
Ipan = imread('PanoramicGray.jpg');
Itel = imread('TelescopicGray.jpg');

[n,m] = size(Itel); %atribui o tamanho da imagem TelescopicGray
%laço para inverter as cores da TelescopicGray
for i = 1:m
    for j = 1:n
        if Itel(i,j) > 100
            Itel(i,j)=0;
        else
            Itel(i,j)=255;
        end
    end
end
%Set o ponto inicial do "telescópio" na imagem panorâmica:
imshow(Ipan);
title('Imagem Panorâmica Original');
% Captura o ponto inicial Pi da matriz de captura Itel na matriz a ser
%capturada Ipan.
[x] = ginput(1);
x = int64(x);
Pi=[x(2),x(1)];
% Icap é a matriz capturada
Icap(:, :) = Ipan(Pi(1):Pi(1)+n-1,Pi(2):Pi(2)+m-1);
% imshow(Ipan) mostrar a composição das duas imagens
Ipan(Pi(1):Pi(1)+n-1,Pi(2):Pi(2)+m-1) = Itel(:, :); %composição das matrizes
%rotacao
% Ângulo definido por pela variável theta
theta = 45;
% Matriz de rotação 2D
R = [cosd(theta) -sind(theta) 0;
     sind(theta) cos(theta) 0 0 1];
Tf = affine2d(R);
```

```

    % Realiza a rotação da matriz Icap
    rot = imwarp(Icap,Tf);
%espelhamento
    % Matriz de espelhamento M
    M = [1 0 0;0 -1 0; 0 0 1];
    Tf = affine2d(M);
    % Realiza o espalhamento da matriz Icap
    esp = imwarp(Icap,Tf);
% cisalhamento
    % Parâmetros de cisalhamento
    shx=0.4;
    shy=0.4;
    % Matriz de cisalhamento
    Sh = [1 shx 0;shy 1 0; 0 0 1];
    Tf = affine2d(Sh);
    % Realiza o cisalhamento da matriz Icap
    cis = imwarp(Icap,Tf);
% escalonamento
    % Parâmetros de escalonamento
    sx=7;
    sy=8;
    % Matriz de escalonamento
    s = [sx 0 0;0 sy 0; 0 0 1];
    Tf = affine2d(s);
    % Realiza o escalonamento da matriz Icap
    esc = imwarp(Icap,Tf);
figure(1), imshow(Ipan);
title('Imagem Panorâmica');
figure(2), imshow(Itel);
title('Imagem Telescópio Modificado');
figure(3), imshow(Icap);
title('Imagem Capturada');
figure(4), imshow(rot);
title('Rotação Aplicada na Imagem Capturada');
figure(5), imshow(esp);
title('Espelhamento Aplicado na Imagem Capturada');
figure(6), imshow(cis);
title('Cisalhamento Aplicado na Imagem Capturada');
figure(7), imshow(esc);
title('Escalação Aplicado na Imagem Capturada');

```

Warning: Image is too big to fit on screen; displaying at 50%
Warning: Image is too big to fit on screen; displaying at 50%
Warning: Image is too big to fit on screen; displaying at 67%

Imagem Panorâmica Original



Imagem Panorâmica



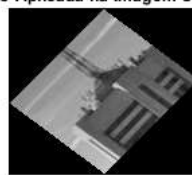
Imagem Telescópio Modificado



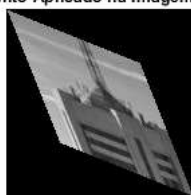
Imagem Capturada



Rotação Aplicada na Imagem Capturada



Cisalhamento Aplicado na Imagem Capturada



Espelhamento Aplicado na Imagem Capturada



Escalonamento Aplicado na Imagem Capturada



Etapa 3:

Nesta etapa será realizada uma transformação homográfica na imagem "UFSC.jpg". Afim de retirar as projeções de perspectiva presentes na imagem.

```
I=imread('UFSC.jpg');
%Definiu-se os pontos desejados para o resultado após a transformação.
p=[0,0,200,0,0,200,200,200];
%Realocou-se os pontos desejados na matriz li
li=zeros(8,1);
for i = 1:8
    li(i)=p(i);
end
%Criando a matriz de homografia
Am=zeros(8,8);
imshow(I);
title('Imagem Original');
%Capturando os pontos iniciais
[x y] = ginput(4);
%linha 1
Am(1,1)=x(1);
Am(1,2)=y(1);
Am(1,3)=1;
Am(1,7)=((-li(1))*x(1));
Am(1,8)=((-li(1))*y(1));
%linha 2
Am(2,4)=x(1);
Am(2,5)=y(1);
Am(2,6)=1;
Am(2,7)=((-li(2))*x(1));
Am(2,8)=((-li(2))*y(1));
%linha 3
Am(3,1)=x(2);
Am(3,2)=y(2);
Am(3,3)=1;
Am(1,7)=((-li(3))*x(2));
Am(1,8)=((-li(3))*y(2));
%linha 4
Am(4,4)=x(2);
Am(4,5)=y(2);
Am(4,6)=1;
Am(4,7)=((-li(4))*x(2));
Am(4,8)=((-li(4))*y(2));
%linha 5
Am(5,1)=x(3);
Am(5,2)=y(3);
Am(5,3)=1;
Am(5,7)=((-li(5))*x(3));
Am(5,8)=((-li(5))*y(3));
%linha 6
Am(6,4)=x(3);
Am(6,5)=y(3);
Am(6,6)=1;
Am(6,7)=((-li(6))*x(3));
Am(6,8)=((-li(6))*y(3));
%linha 7
```

```

Am(7,1)=x(4);
Am(7,2)=y(4);
Am(7,3)=1;
Am(7,7)=(-li(7))*x(4);
Am(7,8)=(-li(7))*y(4);
%linha 8
Am(8,4)=x(4);
Am(8,5)=y(4);
Am(8,6)=1;
Am(8,7)=(-li(8))*x(4);
Am(8,7)=(-li(8))*y(4);
%calculando um "vetor" de homografia
h = inv(Am)*li;
h =[h;1];
%Por fim, a matriz de homografia H1
H1 = reshape(h,3,3);
H1(1,3) = 0;
H1(2,3) = 0;
%calculo da nova matriz de saida
Tf = affine2d(H1);
out = imwarp(I,Tf);
figure(2),imshow(out);
title('Resultado Após Aplicação da Matriz de Homografia');

```

Imagem Original



Resultado Após Aplicação da Matriz de Homografia

