

UNIVERSIDADE FEDERAL DE SANTA CATARINA
CAMPUS BLUMENAU

LABORATÓRIO 08

EDUARDO MAFRA PEREIRA (15102929)

PROFESSOR LEONARDO MEJIA RINCON
PROFESSOR MARCOS VINICIUS MATSUO
Visão Computacional em Robótica

BLUMENAU

2019

INTRODUÇÃO

Conforme solicitado na atividade do Laboratório 08 foi implementado no Matlab um algoritmo para realizar a correção da perspectiva da imagem da Figura 1 (a) a fim de obter o resultado da Figura 1(b). Para isto foram utilizados os conceitos de linhas de perspectiva de imagem, transformada Hough e matriz de homografia.

Figura 1



(a)



(b)

DESENVOLVIMENTO

Na implementação do algoritmo inicialmente foi extraída a imagem fora de perspectiva através da função “iread”, e transformada para escala de cinzas utilizando a função “rgb2gray”. Na sequência foi aplicada a função “iopen” a fim de suavizar a imagem eliminando possíveis ruídos. É aplicado um filtro de detecção de bordas, um preenchimento na imagem gerada após a detecção das bordas e uma nova detecção de bordas, este processo de aplicar detecção de bordas preencher a imagem e aplicar detecção de bordas novamente foi necessário para obter apenas as bordas externas e torná-las mais uniformes. O algoritmo apresenta-se da seguinte maneira:

```
clear;clc; close all;

I = imread('capa.jpg');
Im = rgb2gray(I); % Transforma a imagem para a escala Gray.
s = double(ones(5,5));
Im = iopen(Im, s); % elimina ruídos
edges = icanny(Im); % Filtro de detecção de bordas
Im = bwconvhull(edges); % realiza preenchimento da imagem
edges = icanny(double(Im)); % filtro de detecção de bordas
figure; idisp(edges);
```

Após a obtenção de uma nova imagem que contém apenas as bordas externas da imagem é aplicada a transformada Hough e obtidas as linhas geradas através desta transformada.

```
I2 = edges;
h = Hough(edges, 'suppress',10);
linhas = h.lines(); % linhas de perspectiva
hold on;
linhas.plot;
linhas2 = linhas.seglength(I2); % linhas de perspectiva (com informação adicional).
k = find(linhas2.length>80); % medida de precaução, não é necessária neste caso
figure();idisp(I);
linhas2(k).plot('g');
```

Com as linhas de perspectiva geradas é possível encontrar os pontos desejados para aplicação da matriz de homografia, estes serão portanto os pontos de interseção das retas e podem ser calculados pela equação da reta.

$$v = au + b ; \text{ onde } a \text{ pode ser descrito como } -\tan(\theta) \text{ e } b \text{ como } \frac{\rho}{\cos(\theta)}$$

Se na interseção, os valores de u e v para ambos os pontos são iguais é possível determinar as coordenadas através dos coeficientes de suas retas:

$$u = \frac{\left(\frac{\rho_1}{\cos(\theta_1)} - \frac{\rho_2}{\cos(\theta_2)}\right)}{\tan(\theta_1) - \tan(\theta_2)} ;$$

$$v = -u \cdot \tan(\theta_2) + \frac{\rho_2}{\cos(\theta_2)} .$$

O algoritmo proposto utiliza ambas as equações acima extraíndo θ e ρ das linhas obtidas através da transformada Hough para encontrar os pontos de interseção. Alguns pontos encontrados não correspondem aos pontos desejados da imagem, sendo assim foi desenvolvida uma condição para armazenar em vetores apenas os pontos correspondentes.

```
[m,n] = size(linhas2);
k = 0;
a = 0;
for i = 1:n
    for j = i+1:n
        k=k+1;
        ul(k) = ((linhas2(i).rho/cos(linhas2(i).theta)) - ...
            (linhas2(j).rho/cos(linhas2(j).theta)))/...
            (tan(linhas2(i).theta)-tan(linhas2(j).theta)));
        vl(k) = -ul(k)*tan(linhas2(j).theta) + ...
            (linhas2(j).rho/cos(linhas2(j).theta));
        if ul(k)>0 && vl(k)>0
            a=a+1;
            u(a)=ul(k); % Pontos de interseção
            v(a)=vl(k);
        end
    end
end
```

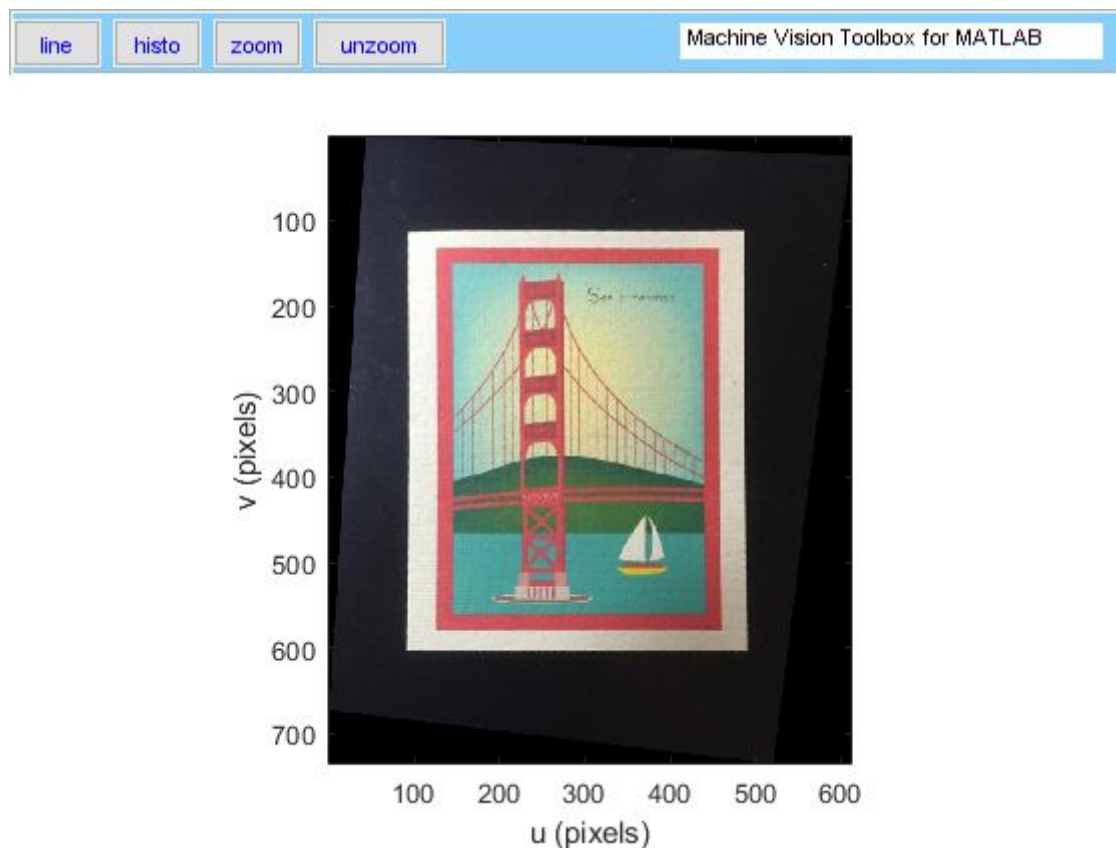
Por fim, com os pontos de interseção das linhas de perspectiva encontrados foi possível aplicar a função de homografia. Esta função foi fornecida pelos professores aos alunos no roteiro do laboratório 2.

```
x1=transpose(u);  
y1=transpose(v);  
x2=[min(u);min(u);max(u);max(u)];  
y2=[min(v);max(v);min(v);max(v)];  
T = maketform('projective',[x1 y1],[x2 y2]);  
% Realiza a homografia da imagem.  
T.tdata.T  
% Realiza a homografia da imagem.  
[Im2,xdata,ydata]=imtransform(I,T);  
idisp(Im2);
```

RESULTADO

O resultado é apresentado na Figura 2, e demonstra que a construção do algoritmo foi realizada com sucesso pelo aluno.

Figura 2 - Resultado obtido



REFERÊNCIAS

{Mejia} RINCON, Leonardo Mejia. Visão Computacional em Robótica

: Notas de aula. Blumenau, SC, 2019.

{Vinicius} MATSUO, Marcos Vinicius Matsuo. Visão Computacional em Robótica

: Notas de aula. Blumenau, SC, 2019.