UNIVERSIDADE FEDERAL DE SANTA CATARINA
CAMPUS BLUMENAU

LABORATÓRIO 06

EDUARDO MAFRA PEREIRA (15102929)

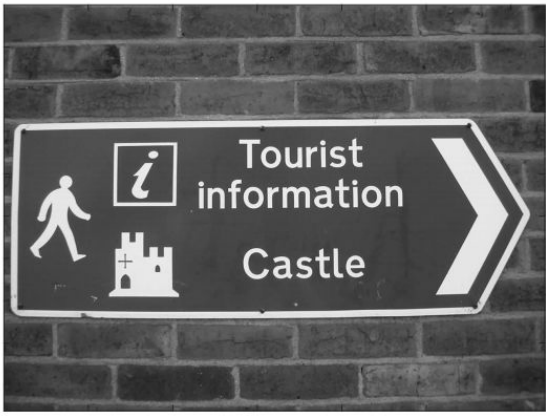
PROFESSOR LEONARDO MEJIA RINCON
PROFESSOR MARCOS VINICIUS MATSUO
Visão Computacional em Robótica

BLUMENAU
2019

INTRODUÇÃO

Conforme solicitado na atividade do Laboratório 06 foi implementado no Matlab um programa que indique as palavras que constam na Figura 1(a). Para isso foi criado um algoritmo que foi capaz indicar as palavras contidas na Figura 1(a) e as funções solicitadas nos tópicos de 1 a 3.

Figura 1 (a)



DESENVOLVIMENTO

O algoritmo desenvolvido é apresentado da seguinte maneira:

Inicialmente são carregadas as imagens requeridas, determinando um valor de área que abrange as letras presentes na imagem, e uma função que encontra e retorna as letras na imagem, aplicando a elas um *bounding box*. (Esta função será apresentada no final do documento).

```
clc
close all;
clear all;

global elementos;
global lista; % Variaveis declaradas como global

I = iread('castle.png'); % Captura a imagem
I_bw = im2bw(I,0.8); % Homogeniza a imagem a fim de retirar ruídos
[label, m, parents, cls] = ilabel(I_bw); % atribui índices aos objetos da imagem
amin = 40;
amax = 4500; % Restringe a área para identificar uma letra
idisp(I); % Mostra a Imagem
hold on


retornaletras(label,m,cls,amin,amax); % função que encontra e reotarna as


elementos1=[];
u=0; % variável ultilizada para atribuir valores da matrix elemento para uma
% nova matrix chamada de matrix elementol.
```

Com início nas *bounding box* encontradas através da função *"retornaletras"* foi implementado alguns laços de *"for"* para agrupar os *bounding box* de cada letra formando assim uma palavra. Algumas condições foram estipuladas para que as letras certas sejam agrupadas.

Elementos (:,:) é uma matriz construída na função *"retornaletras"*, nela estão presentes as coordenadas de cada *bounding box* (a primeira coluna refere-se às menores coordenadas em x de cada *bounding box*, a segunda coluna às maiores coordenadas em x, na terceira coluna as menores coordenadas em y, na quarta coluna maiores coordenadas y e a quinta coluna refere se ao *"label"* aplicada aquela letra).

```
for i = 1:size(lista,2)
    bbox = 0; % Se condição para saber se as box estão perto
    vetorXmax = [];
    vetorXmin = [];
    vetorYmin = [];
    vetorYmax = []; % Vetores que armazenam as posições das bounding box
    for j= i:size(lista,2)

        if elementos(i,2) >= 0.7*elementos(j,1) && elementos(i,2) <= 1*elementos(j,1) && ...
        elementos(j,2)>1 && elementos(i,3)>= 0.85*elementos(j,3) && elementos(i,3) <= 1.15*elementos(j,3)
        || elementos(i,1) >= 0.7*elementos(j,1) && elementos(i,1) <= elementos(j,1) && ...
        elementos(j,2)>1 && elementos(i,3)>= 0.85*elementos(j,3) && elementos(i,3) <= 1.15*elementos(j,3)
        || elementos(j,1) >= 0.7*elementos(i,2) && elementos(j,1) <= 1*elementos(i,2) && ...
        elementos(j,2)>1 && elementos(i,3)>= 0.85*elementos(j,3) && elementos(i,3) <= 1.15*elementos(j,3)
        % Condições que determina se as bounding box serão agrupadas ou não


        vetorXmax(j) = elementos(j,2);
        elementos(j,2) = 0;
```

```
        vetorXmax(j) = elementos(j,2);
        elementos(j,2) = 0;
        vetorXmin(j) = elementos(j,1);
        vetorYmin(j) = elementos(j,3);
        vetorYmax(j) = elementos(j,4); % Vetor onde são aplicadas as coordenadas
        % bounding box caso a condição anterior seja verdadeira.
        bbox = 1; % As bounding box estão próximas
        end

    end
```

Laços para determinar encontrar as *bounding box* resultantes e agrupar as letras para formar palavras:

```
    if bbox == 1 && elementos(i,1)>0

        for k = 1:size(vetorXmin,2)
            if vetorXmin(k) == 0
                vetorXmin(k) = 15000; % evitar que aconteça erros quando for
                % v=encotrar o valor mínimo do vetor "vetorXmin".
            end
        end
        for k = 1:size(vetorYmin,2)
            if vetorYmin(k) == 0

                vetorYmin(k) = 15000; % evitar que aconteça erros quando for
                % v=encotrar o valor mínimo do vetor "vetorXmin".

            end
        end

        xmin = min(vetorXmin);
        xmax = max(vetorXmax);
        ymin = min(vetorYmin);
        ymax = max(vetorYmax); % Encontra as posições para criar a bounding box
        u=u+1; % Altera linha de elementos1
        elementos1(u,:)=[xmin,xmax,ymin,ymax,u] % Nova matrix com as bounding box agrupadas
    end
```

Ainda não sendo possível encontrar as palavras finais, só pseudo palavras. Sendo assim, foi necessário aplicar e desenvolver uma segunda etapa do algoritmo para agrupar estas pseudo palavras de acordo com sua proximidade. A solução encontrada para isto foi relacionar estas pseudo palavras por índices para que finalmente sejam geradas as palavras.

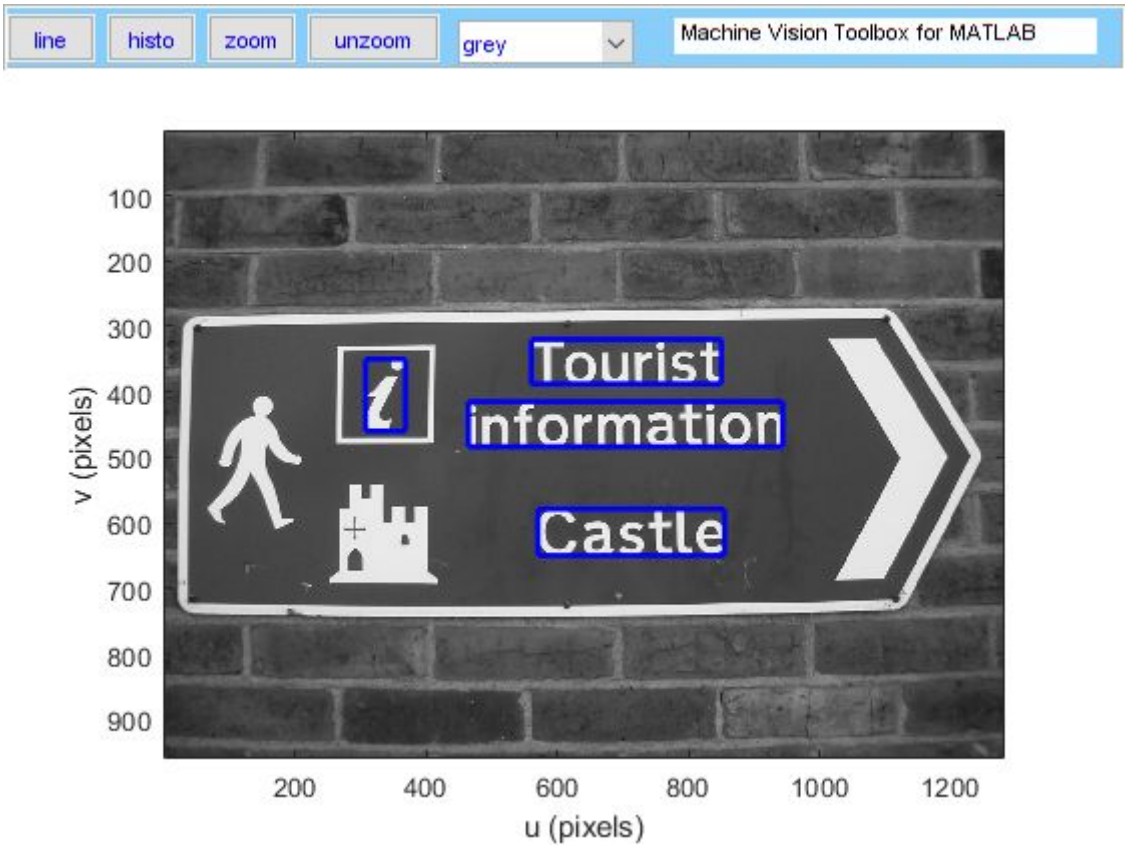
```
end
elementos=elementos1(:,:);
elementos = sortrows(elementos,3) % Ordena as matrizes do maior y para o menor

[n,m] = size(elementos) % Captura a quantidade de linas da matrix elementos

% Laço para relacionar bounding box próximas aplicando nelas um mesmo índice
for i= 1: n
    for j=i:n
        if elementos(i,2)>=0.98*elementos(j,1) && elementos(i,4)>= 0.9*elementos(j,4)...
                && elementos(i,4)<= 1.1*elementos(j,4)
          elementos(j,5) = elementos(i,5); % Relaciona os bounding box próximos aplicando um mesmo
          % índice
        end
    end
end

% Laço para fazer as bounding box finais agrupando as bounding box de mesmo índice
for i= 1:n
    vetorXmax = [];
    vetorXmin = [];
    vetorYmin = [];
    vetorYmax = [];
    for j = 1:n
```

```
    for j = 1:n
        if elementos(i,5) == elementos(j,5)
        vetorXmax(j) = elementos(j,2)
        vetorXmin(j) = elementos(j,1)
        vetorYmin(j) = elementos(j,3)
        vetorYmax(j) = elementos(j,4) % Atribui as coordenas de mesmo índice nos vetores de
        % posição
        for k = 1:size(vetorXmin,2)
            if vetorXmin(k) == 0
                vetorXmin(k) = 15000;
            end
        end
        for k = 1:size(vetorYmin,2)
            if vetorYmin(k) == 0
                vetorYmin(k) = 15000;
            end
        end
        end
    end
    xmin = min(vetorXmin);
    xmax = max(vetorXmax);
    ymin = min(vetorYmin);
    ymax = max(vetorYmax); % funções para encontrar os pontos extremos de
    % cada índice, podendo assim gerar o bounding box
```

Após as palavras finalmente encontradas, as coordenadas de suas *bounding box* são atribuídas a função desenha, está portanto desenha as *bounding box* na imagem.

```
    desenha(I,xmin,xmax,ymin,ymax); % função criada para desenhar o bounding box
end
```

RESULTADO



Funções desenvolvidas:

A função *"retornaletras"* encontra as letras dentro da imagem considerando que objetos que possuam sua área em um intervalo determinado e seu valor na matriz da imagem preto e branco igual a 1. As coordenadas de cada letra são atribuídas na matriz "elementos" para serem utilizadas no decorrer do algoritmo principal.

```
function retornaletras(label,m,cls,amin,amax)
a=1;
lista=[];
global lista;
for i = 1:m
    I2 = (label == i);

    [v,u] = find(I2); % encontra as posições dos objetos na imagem
    umin = min(u);
    umax = max(u);
    vmin = min(v);
    vmax = max(v);

    xmin(i) = umin; % atribui as posições a vetores
    xmax(i) = umax;
    ymin(i) = vmin;
    ymax(i) = vmax;
    %area

    area(i) = ((xmax(i)-xmin(i))*(ymax(i)-ymin(i))); % Calcula a área de cada objeto

    if area(i) >= amin && area(i) <= amax && cls(i,1)==1
       lista(a) = i; % Cria uma lista com os índices das letras
       a = a+1;
    end
end
hold on;
[a1,a] = size(lista)
elementos=zeros(a,5);
global elementos
for k = 1:a
    b = lista(k); % atribuí a variável "b" apenas os índices das letras

    elementos(k,:) = [xmin(b),xmax(b),ymin(b),ymax(b),b] % Cria uma matriz com todas as coordenadas
    % das bounding box de cada letra e a label atribuída a ela
end
elementos = sortrows(elementos)
end
```

Já a função “desenha” plot linhas na imagem de fundo após determinadas as coordenadas de cada linha. Para formar uma caixa retangular foram necessárias 4 linhas.

```
function desenha(I,xmin,xmax,ymin,ymax)
hold on
% Função criada para desenhar os bounding box finais
        plot([xmin xmax],[ymin, ymin], 'b','LineWidth',2);
        plot([xmin xmin],[ymin, ymax], 'b','LineWidth',2);
        plot([xmin xmax],[ymax, ymax], 'b','LineWidth',2);
        plot([xmax xmax],[ymin, ymax], 'b','LineWidth',2);
end
```