

UNIVERSIDADE FEDERAL DE SANTA CATARINA
CAMPUS BLUMENAU

LABORATÓRIO 07

EDUARDO MAFRA PEREIRA (15102929)

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

BLUMENAU

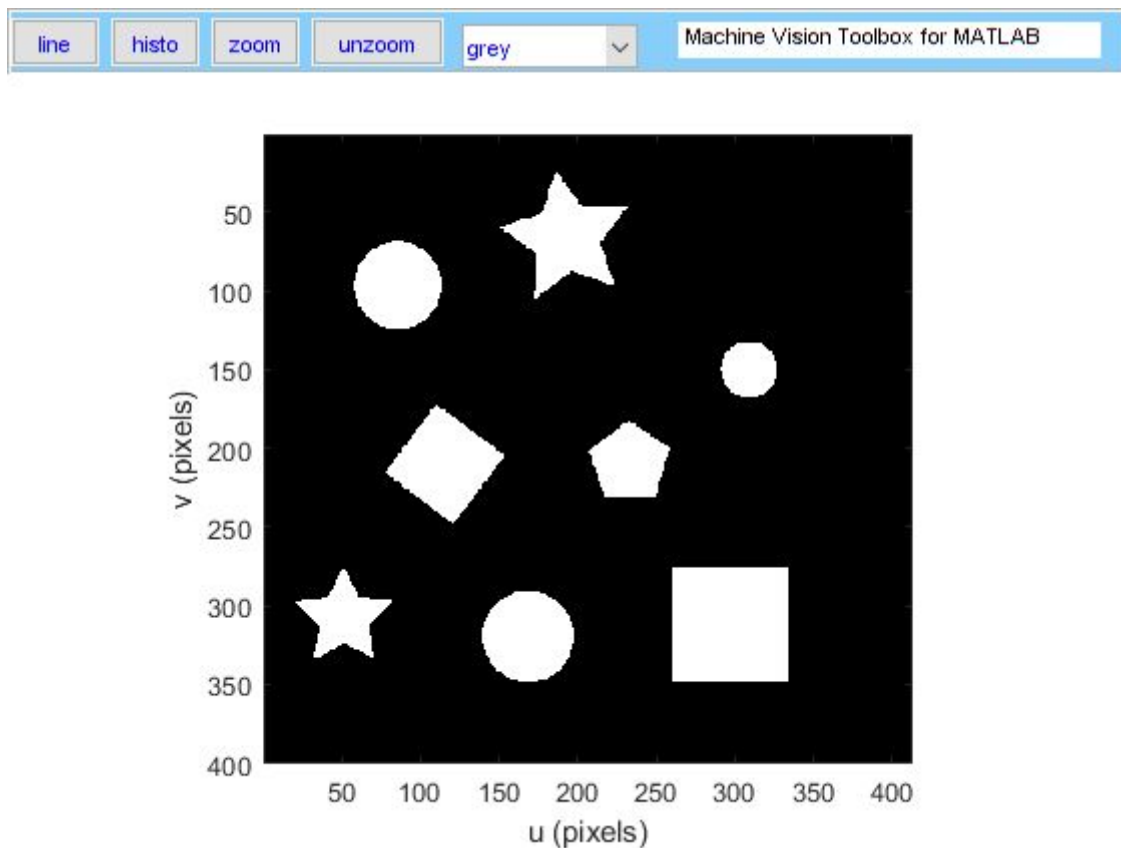
2019

INTRODUÇÃO

Conforme solicitado na atividade do Laboratório 07 foi desenvolvido no Matlab um algoritmo para identificar e classificar os objetos na Figura 1 através do cálculo de momentos de cada objeto.

A base lógica deste algoritmo foi fornecida no laboratório 7 da disciplina BLU3040 de visão computacional.

Figura 1



DESENVOLVIMENTO

O algoritmo desenvolvido é apresentado da seguinte maneira:

Inicialmente são processadas as imagens individuais fornecidas pelo professor Marcos Vinicius Matsuo, estas imagens funcionam como uma espécie de banco de dados dos quais os objetos da imagem principal será comparada para identificar objetos nesta imagem.

O processamento destas imagens individuais é dado da seguinte forma:

- Foi criado um laço de “for” que vai de 1 a 4 onde 1 representa o círculo, 2 representa estrela, 3 representa pentágono e 4 representa o quadrado.
- O cálculo dos momentos necessários para encontrar os vetores descritores propostos neste laboratório são realizados através de uma função desenvolvida (“momentos.m”). Os pontos centrais de cada objeto de comparação também são cálculos neste laço.
- Os momentos calculados são atribuídos a uma função que calcula os vetores descritores de cada objeto (“descritores.m”). Estes vetores são atribuídas a uma matriz que servirá de comparação para comparar os vetores descritores dos objetos que serão encontrados na imagem principal posteriormente.
- O Laço descrito se apresenta da seguinte forma:

```
clc
clear
close all
I = imread('LAB2_FIG.png'); % Captura a imagem a ser processada.
circulo = imread('referencia_circulo.png'); % Captura as formas a serem
% comparadas.
estrela = imread('referencia_estrela.png');
pentagono = imread('referencia_pentagono.png');
quadrado = imread('referencia_quadrado.png');
figure(); idisp(I);
% Calcula momentos centrais:
for i=1:4 % Neste laço serão calculados os momentos necessários de cada
    % imagem individual (circulo,estrela,pentagono,quadrado) e por fim seus
    % vetores descritores.
    if i==1
        im = circulo;
    end
    if i==2
        im = estrela;
    end
```

```

if i==3
im = pentagono;
end
if i==4
im = quadrado;
end
m00(i) = momentos(im, 0, 0, 0, 0);
m10(i) = momentos(im, 1, 0, 0, 0);
m01(i) = momentos(im, 0, 1, 0, 0);
uc(i) = m10(i)./m00(i);
vc(i) = m01(i)./m00(i);
m11(i) = momentos(im, 1, 1, uc(i), vc(i));
m20(i) = momentos(im, 2, 0, uc(i), vc(i));
m02(i) = momentos(im, 0, 2, uc(i), vc(i));
m21(i) = momentos(im, 2, 1, uc(i), vc(i));
m12(i) = momentos(im, 1, 2, uc(i), vc(i));
m30(i) = momentos(im, 3, 0, uc(i), vc(i));
m03(i) = momentos(im, 0, 3, uc(i), vc(i));
Matriz(i,:) = descritor(m00(i),m11(i),m20(i),m02(i),m21(i),m12(i),m30(i),m03(i));
end

```

A segunda parte do código serão comparados os vetores descritores atribuídos a matriz do laço anterior com os objetos encontrados na imagem principal da Figura 1. O restante do código é apresentado como:

- Foi criado um laço para percorrer toda a imagem e encontrar os objetos dentro desta imagem.
- Ao encontrar um objeto são encontrados os pontos mínimos e máximos de cada objeto para serem atribuídos a criação da bounding box posteriormente.
- Os momentos necessários para encontrar os vetores descritores de cada objeto são calculados da mesma maneira que ao laço anterior.
- O cálculo do vetor descritor também é realizado através da função “descritor.m”.
- São comparados estes vetores descritores encontrados com os vetores atribuídos a matriz do laço anterior, e dependendo da forma do objeto uma bounding box de cor diferente será criada neste objeto.
- O restante do código é apresentado como:

```

[lbl, m, parents, cls] = ilabel(I);
for i = 1:m % Este laço Calcula os momentos necessários de cada objeto
    % encontrado na imagem processada, encontra o vetor descritor de cada
    % objeto e por fim compara o vetor descritor encontrado com os valores
    % descritores atribuídos a Matriz do laço anterior através de uma
    % relação de módulo das somas dos 7 elementos de cada vetor.
    if cls(i) == 1
        im = (lbl == i);
        [v, u] = find(im);
        |
        umin = min(u);
        umax = max(u);
        vmin = min(v);
        vmax = max(v);

        m00 = momentos(im, 0, 0, 0, 0);
        m10 = momentos(im, 1, 0, 0, 0);
        m01 = momentos(im, 0, 1, 0, 0);
        uc = m10./m00;
        vc = m01./m00;

        m11 = momentos(im, 1, 1, uc, vc);
        m20 = momentos(im, 2, 0, uc, vc);
        m02 = momentos(im, 0, 2, uc, vc);
        m21 = momentos(im, 2, 1, uc, vc);
        m12 = momentos(im, 1, 2, uc, vc);
        m30 = momentos(im, 3, 0, uc, vc);
        m03 = momentos(im, 0, 3, uc, vc);

        vetor = descritor(m00,m11,m20,m02,m21,m12,m30,m03);
        compara = sqrt((sum(sum(vetor)))^2)

        if compara <= 1.01*(sqrt((sum(sum(Matriz(1,:))^2)))) && ...
            compara >= 0.99*(sqrt((sum(sum(Matriz(1,:))^2))))
            plot_box(umin,vmin,umax,vmax, 'g')
        end
        if compara <= 1.01*(sqrt((sum(sum(Matriz(2,:))^2)))) && ...
            compara >= 0.99*(sqrt((sum(sum(Matriz(2,:))^2))))
            plot_box(umin,vmin,umax,vmax, 'b')
        end
        if compara <= 1.01*(sqrt((sum(sum(Matriz(3,:))^2)))) && ...
            compara >= 0.99*(sqrt((sum(sum(Matriz(3,:))^2))))

```

```

        plot_box(umin,vmin,umax,vmax, 'r')
    end
    if compara <= 1.01*(sqrt((sum(sum(Matriz(4,:))^2)))) && ...
        compara >= 0.99*(sqrt((sum(sum(Matriz(4,:))^2))))
        plot_box(umin,vmin,umax,vmax, 'm')
    end
end
end
end

```

FUNÇÕES DESENVOLVIDAS

A função desenvolvida para o cálculo dos momentos é apresentada como:

```

function valor = momentos(im, p, q, uc, vc) % Função para calcular os
% momentos.
    valor = 0;
    i = (im == 1);
    for v = 1 : size(i,1)
        for u = 1 : size(i,2)
            valor = valor + ((u - uc)^p)*((v - vc)^q)*i(v,u);
        end
    end
end
end

```

Já a função para encontrar os vetores descritores é apresentada como:

```

function vetor = descritor(m00,m11,m20,m02,m21,m12,m30,m03) % Função para
% calcular os vetores descritores.

y1 = 0.5*(1+1)+1; % y11
y2 = 0.5*(2+0)+1; % y20 = y02
y3 = 0.5*(2+1)+1; % y21 = y12
y4 = 0.5*(3+0)+1; % y30 = y03

% Cálculo dos "n" necessários para encontrar o vetor descritor.
n11 = m11/(m00^y1);
n20 = m20/(m00^y2);
n02 = m02/(m00^y2);
n21 = m21/(m00^y3);
n12 = m12/(m00^y3);
n30 = m30/(m00^y4);
n03 = m03/(m00^y4);

% Aplicação das fórmulas apresentadas.
v(1) = n20 + n02;
v(2) = (n20 - n02)^2 + 4*n11^2;
v(3) = (n30 - 3*n12)^2 + (3*n21 - n03)^2;

```

RESULTADO

Resultado está presente na Figura 2 e demonstra que a construção do algoritmo foi realizada com sucesso pelo aluno.

Figura 2

