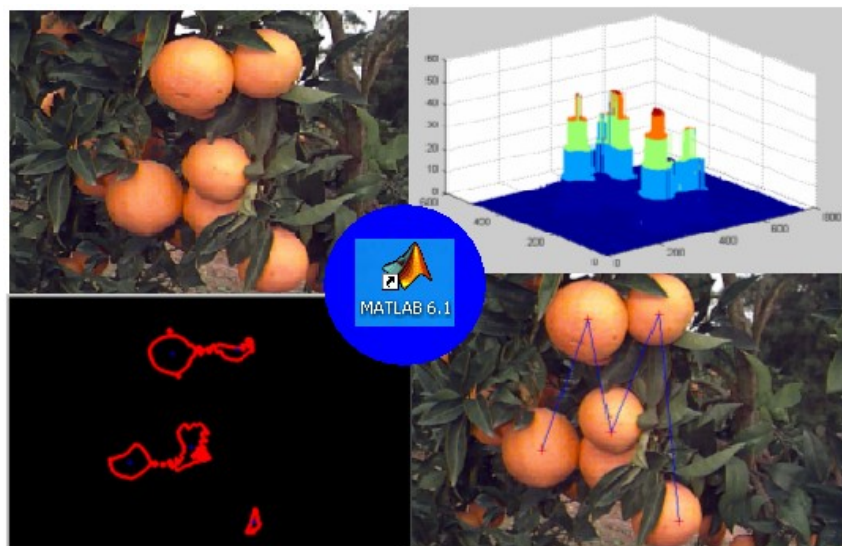


CONTROLO DE SISTEMAS 2002/2003

“Detecção de Laranjas”



Realizado por:

Ana Luísa Martins nº 9902004
Carla Sofia Flores nº 9902008



Este trabalho tem como objectivo fazer com que através do Matlab sejam detectadas as laranjas de uma fotografia de uma laranjeira. Para isso vimos que o que faz distinguir uma laranja ao olho humano, é a sua forma e a sua cor. Foram essencialmente estes dois aspectos (entre outros pormenores) que permitiram ao Matlab detectar a maioria das laranjas e também o facto de programa para programa não alterarmos as dimensões da imagem original.

Subdividimos o nosso trabalho em 5 programas porque se fosse só num, o Matlab demorava muito tempo a processar tanta informação e provavelmente bloqueava.

Assim no Matlab, deve-se correr um programa de cada vez. Pois ao executar o outro programa é carregado o resultado final do programa anterior.

Imagem Original:





Programa 1

Este programa tem como objectivo a ordenação de cores por tonalidade da imagem, para se poder definir uma gama de cor laranja para poder detectar as laranjas.

```
I=imread('laranjas.tif');

[image256,map256]=rgb2ind(I,256);%passa a imagem a indexada

[map256ord mapant]=sortrows(map256,1);%ordena o colormap por indexação a 1ª coluna
tam_fig=size(image256);%tira o tamanho da imagem
i=1:1:256;
mapaux=[mapant i'];%e criado um mapa auxiliar para poder ordenar as cores

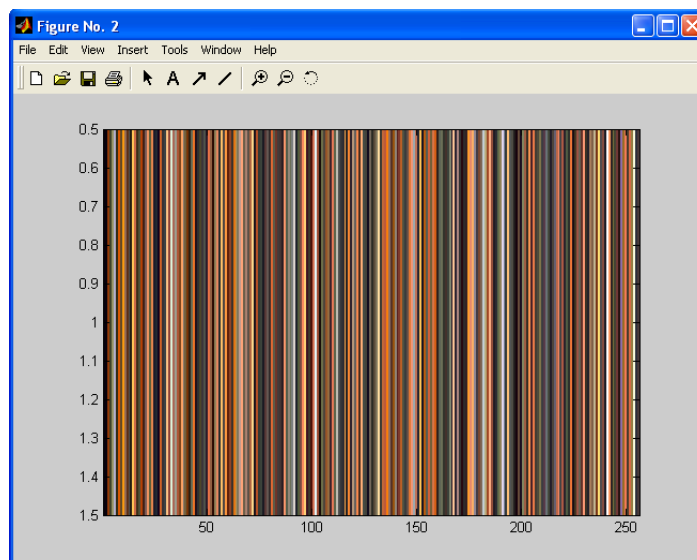
mapindex=sortrows(mapaux,1); %ordenação de cores por tonalidade no mapa auxiliar
for y=1:1:tam_fig(1)
    for x=1:1:tam_fig(2)
        image256ord(y,x)=uint8(double(mapindex(double(image256(y,x))+1,2))-1);
    end
end
End
f=figure(1)
set(f,'DoubleBuffer','ON')%estabiliza as imagens
imshow(image256ord,map256ord);
hold on

Figure
i=0:1:255;

colormap(map256);
image(i);
Figure
colormap(map256ord);
image(i);

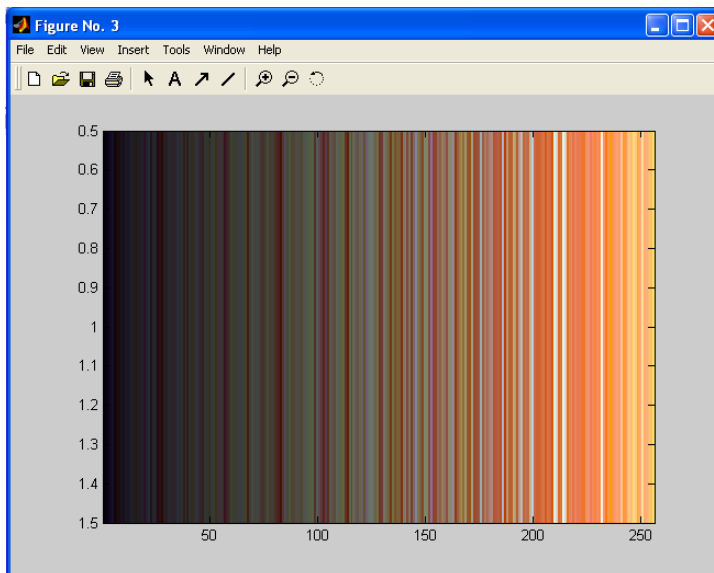
save salvo1 image256ord map256ord
```

Assim primeiramente obtemos o colormap da nossa fotografia desordenado:

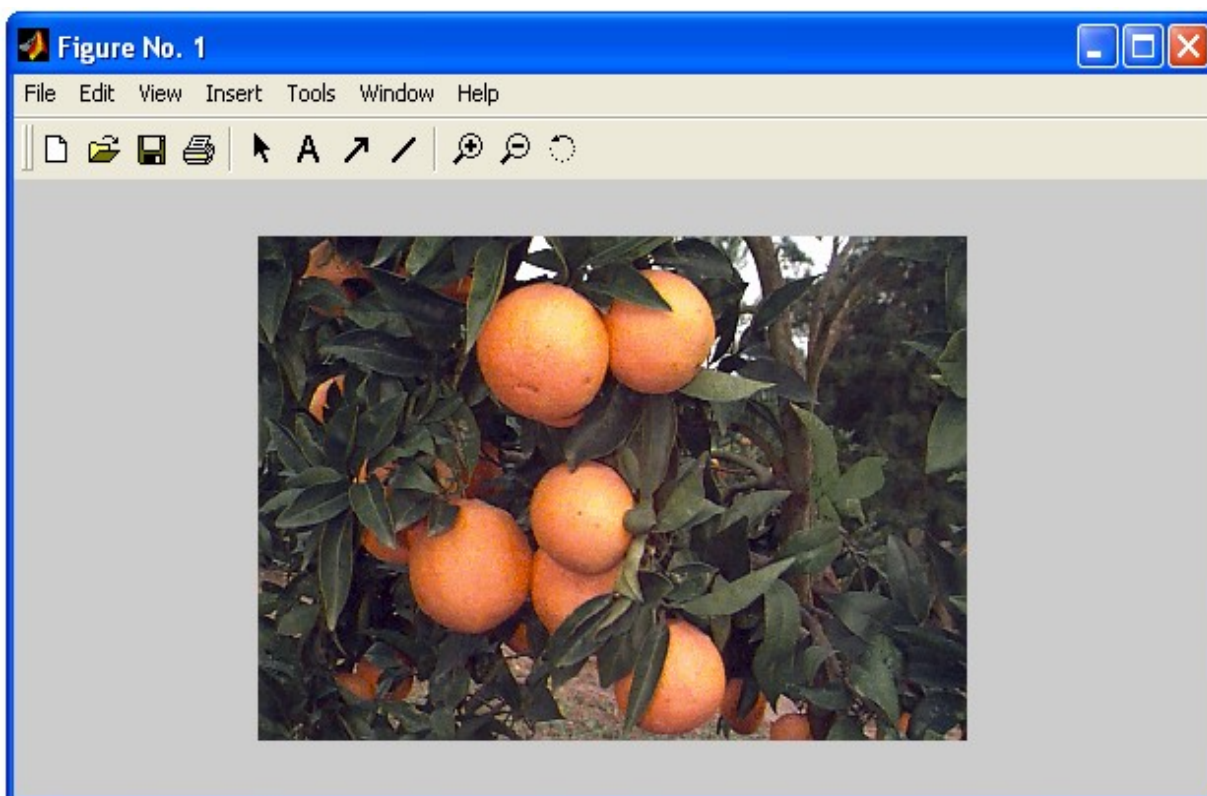




Depois através de um mapa auxiliar conseguimos obter o colormap da fotografia ordenado sendo possível definir uma gama de cores para laranja:



E finalmente obtemos a nossa fotografia idêntica à original, mas com a diferença de ter o colormap ordenado:





Programa 2

Este programa tem como objectivo passar a imagem para preto e branco. Assim, primeiramente é aplicado um filtro a imagem para retirar eventuais ruídos. Depois através do programa anterior definimos uma gama para a cor laranja que será de 130 até 255. Os pixeis detectados com estes valores passam a branco (=1) e os restantes ficam a preto (=0).

```
load salvo1%carrega o resultado do programa anterior

tam_fig=size(image256ord);
mgaussiana=fspecial('gaussian',5,5);%aplicação do filtro a imagem
image256ord=imfilter(image256ord,mgaussiana);%filtragem da imagem
k=1;
for y=1:1:tam_fig(1)
    for x=1:1:tam_fig(2)
        if (image256ord(y,x)>130)&(image256ord(y,x)<255)
            pontos(k,:)=[x y];
            imageBW(y,x)=1;%se o pixel estiver entre 130 e 255 entao e laranja e passa a branco
            k=k+1;
        Else
            imageBW(y,x)=0;%se o pixel for menor que 130 o pixel passa a preto
        end
    End
End

plot(pontos(:,1),pontos(:,2),'b.');//%plot da imagem em que os pixeis de 130 a 255 estao a azul

f=figure(2);
set(f,'DoubleBuffer','ON')
imshow(imageBW);
hold on
save salvo2 imageBW
```



Plot da imagem em os pontos entre 130 e 255 passam a azul:

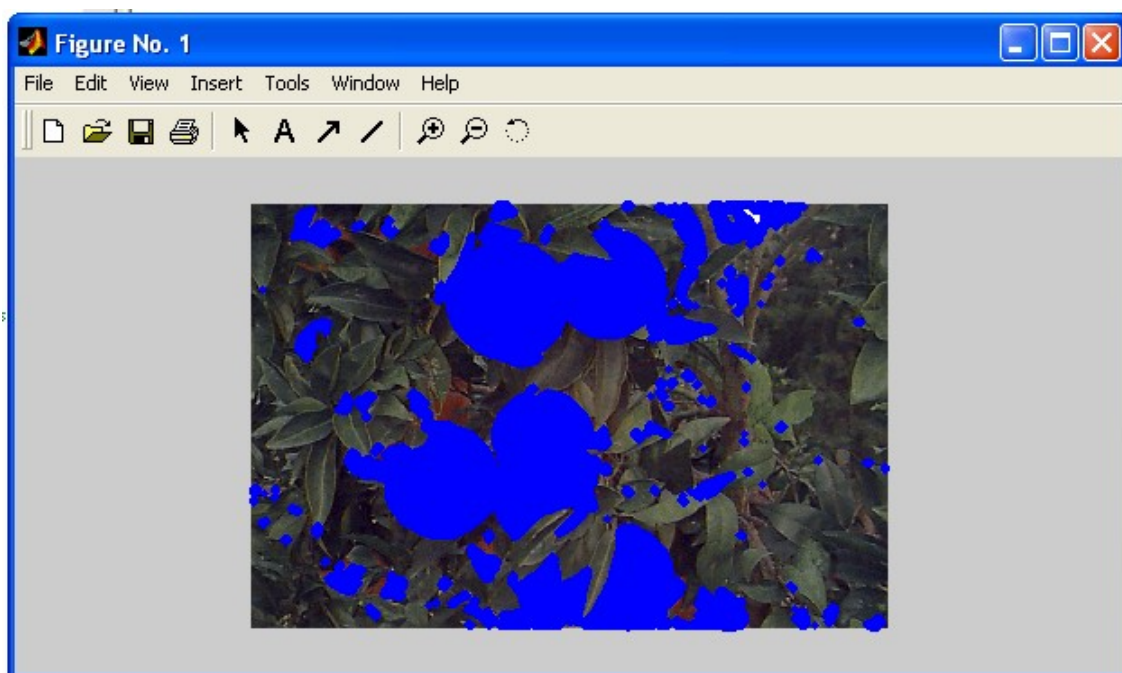
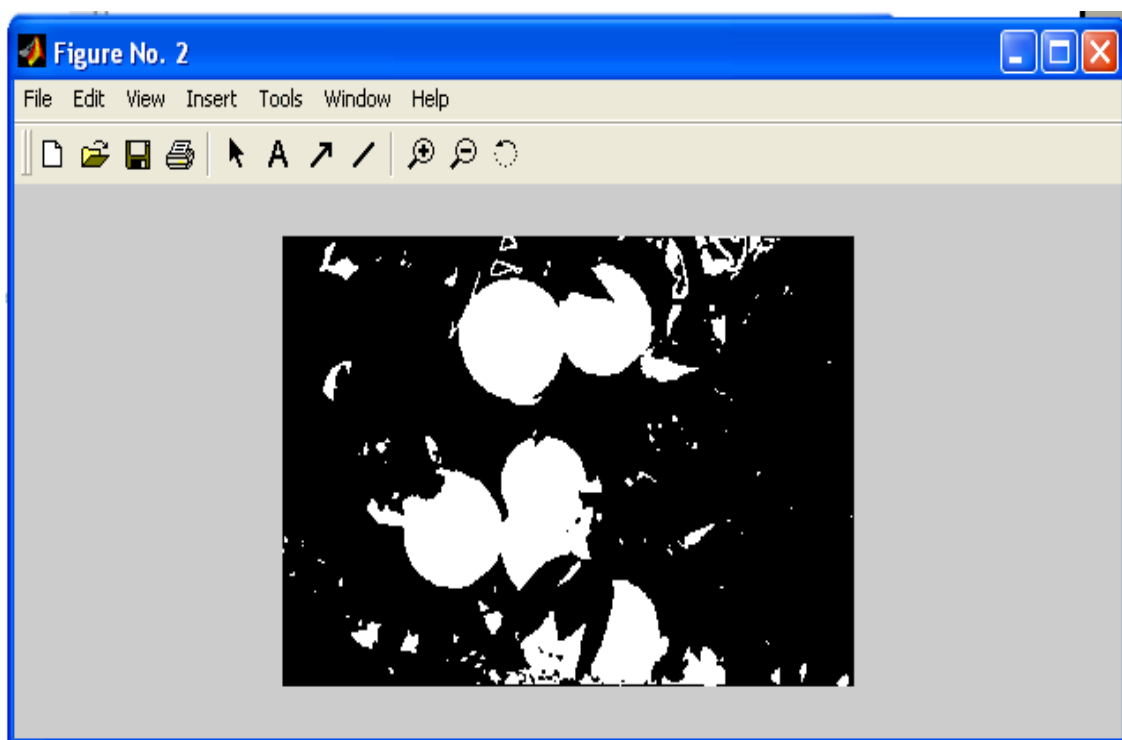


Imagem BW em os pontos azul da figura anterior ficam brancos e os restantes passam a preto:





Antes de passarmos ao programa 3 criámos uma função círculos para podermos definir vários círculos para poder detectar as laranjas pois as suas formas assemelham-se a círculos de diferentes raios em que existem uma margem de ser ou não ser círculo:

```
function saida=circulos(imageori,xr,yr,dim,per,imagecir)
[x y]=size(imageori);
tam=size(xr);
imageaux=imageori;
dim_a=dim-10;
for j=dim:1:y-dim
    for i=dim:1:x-dim
        valor = 0;
        for k=1:1:tam(2)
            valor=valor+imageori(i+round(yr(dim_a,k)),j+round(xr(dim_a,k)));
        end
        if(valor > per)
            imageaux(i,j)=valor+imagecir(i,j);
        End
    End
End
saida=imageaux;
```

Programa 3

```
load salvo2

raio=10:1:60;
w=-pi:0.5:pi;
xr=raio*cos(w);
yr=raio*sin(w);

[x y]=size(imageBW);
Image_circulo=zeros(x,y);

Image_circulo=circulos(imageBW,xr,yr,61,10,image_circulo);
Image_circulo=circulos(imageBW,xr,yr,41,12,image_circulo);
Image_circulo=circulos(imageBW,xr,yr,31,12,image_circulo);
Image_circulo=circulos(imageBW,xr,yr,21,12,image_circulo);

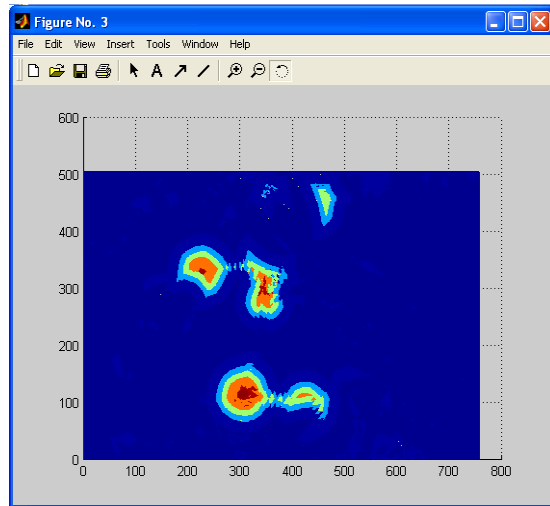
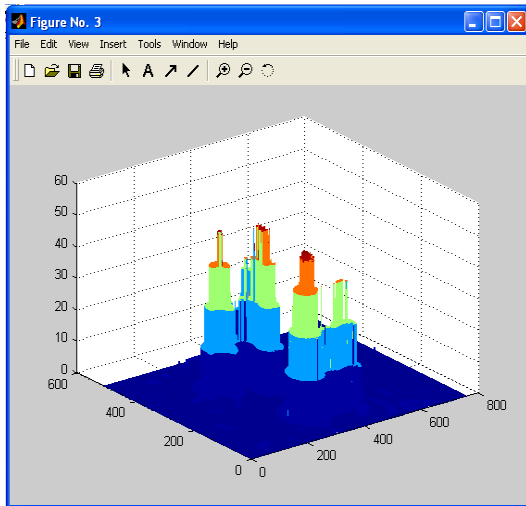
%sao criados 4 niveis de circulos de forma a abranger todas as laranjas
%entre um raio minimo e um raio maximo com uma determinada margem de ser laranja ou nao

save salvo3 image_circulo;

f=figure;
mesh(image_circulo)%pode-se ver os 4 niveis de circulos perfeitamente distintos
```



Figuras que apresentam o "Mesh" da imagem em dois ângulos diferentes em que se distinguem os 4 níveis de círculos:



Programa 4

Este programa dá uma melhor percepção das laranjas em que se vêem as coisas de forma mais clara. Então pegamos no programa anterior e aplicamo-lo à nossa imagem BW. Então se a soma de vários círculos for maior que 25 (valor obtido por afinização) temos uma laranja.

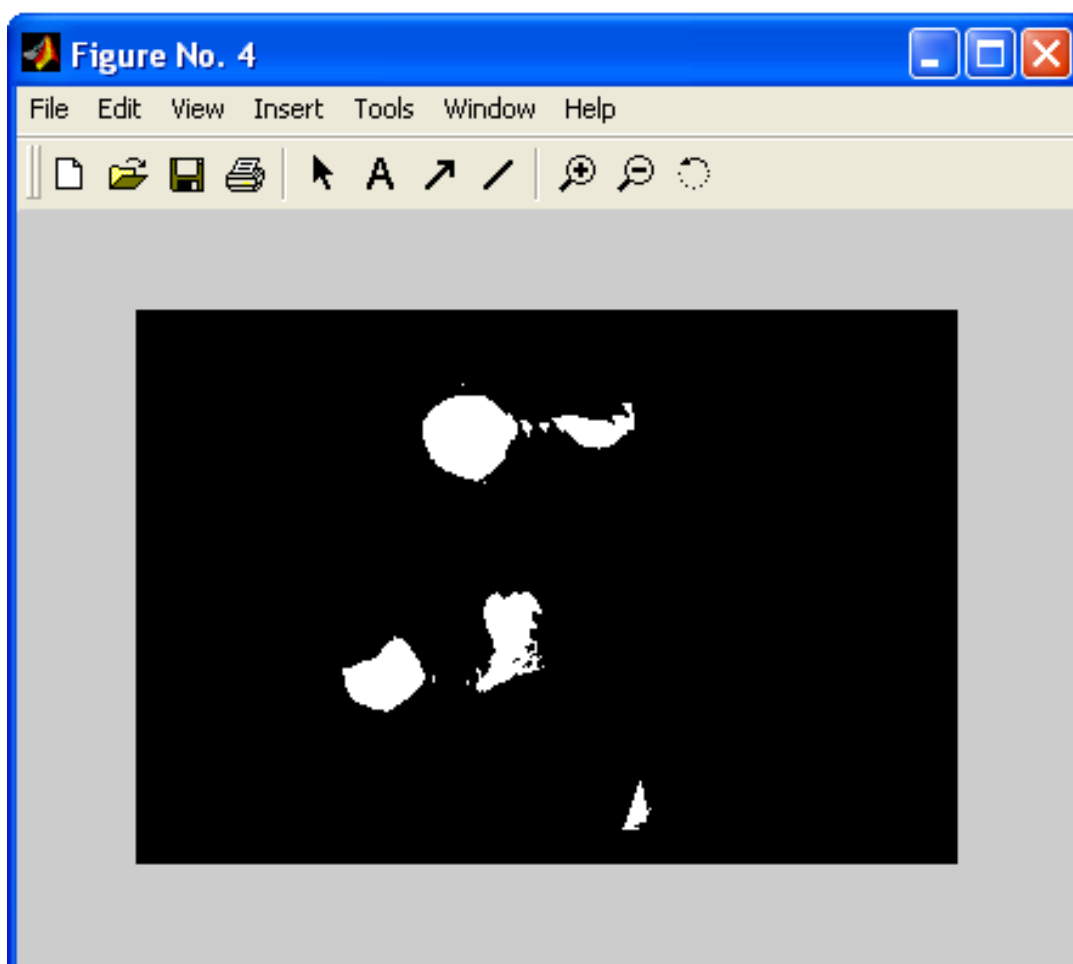
```
load salvo3
laranjas=image_circulo;
[x y]=size(image_circulo);
for j=1:1:y
    for i=1:1:x
        if image_circulo(i,j)>25 %se a soma de varios circulos for maior que 25
            laranjas(i,j)=1; %e uma laranja
        else
            laranjas(i,j)=0; %nao e laranja
        end
    end
end
end

f=figure(4)
set(f,'DoubleBuffer','ON');
imshow(laranjas)

save salvo4 laranjas
```




Laranjas perfeitamente distintas:





Programa 5

Neste programa fomos capturar os contornos "edge" das laranjas, para calcularmos os centros de massa das laranjas e através disso detectarmos finalmente as nossas laranjas.

```
load salvo4

f=figure(5);
set(f,'DoubleBuffer','ON')

mgaus=fspecial('gaussian',5,5);
laranjas=imfilter(laranjas,mgaus);

laranjas_edge = edge(laranjas,'log'); %calcula os contornos

imshow(laranjas_edge);

[x y]=size(laranjas_edge);

%percorre a imagem por linhas
cont=1;
last_orient=4;

%codigos de freeman na forma -1
orientacao=[-1 0
            -1 1
             0 1
             1 1
             1 0
             1 -1
             0 -1
            -1 -1];

som_x=0;
som_y=0;

numero_pixeis=0;
hold on
centro=1;
for j=1:1:y
    for i=1:1:x

        if laranjas_edge(i,j)==1

            x_init=i;
            y_init=j;

            while 1,
                new_found=0;
                h=8+last_orient-3;
                res=double(int8(h/8)); %resto da divisao
                new_orient=h-8*res+1;
                erro=0;
                while new_found~=1
```



```
a=i+orientacao(new_oriente,1);
b=j+orientacao(new_oriente,2);
if laranjas_edge(a,b)==1
    som_x=som_x+a;%atualiza somatorio de coordenadas
    som_y=som_y+b;
    laranjas_edge(a,b)=0;
    i=a;
    j=b;
    last_oriente=new_oriente;
    new_found=1;
    erro=0;
    plot(j,i,'r.');
```

```
else
    if(new_oriente==8)
        new_oriente=1;
        erro=erro+1;
    else
        new_oriente=new_oriente+1;
        erro=erro+1;
    end
end
if erro==9
    break;
end

end%end while
if erro==9
    som_x=0;
    som_y=0;
    numero_pixeis=0;
    break;
end
numero_pixeis=numero_pixeis+1;

if (x_init==i) & (y_init==j)
    if (numero_pixeis>50) %nº minimo de pixeis para ser laranja
        plot(som_y/numero_pixeis,som_x/numero_pixeis,'b+'); %da o centro de massa
        pause(0.1);
        centro_laranjas(centro,:)= [som_y/numero_pixeis som_x/numero_pixeis];
        centro=centro+1;
    end
    som_x=0;
    som_y=0;
    numero_pixeis=0;
    break;
end %end if
end % end while 1

end

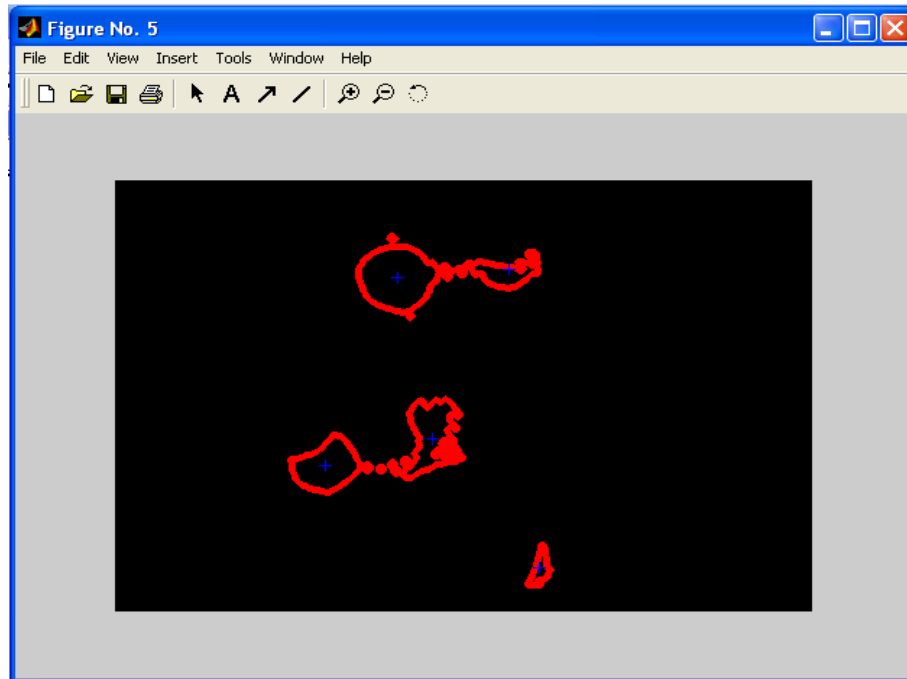
end
end
load salvo1

f=figure(5);
set(f,'DoubleBuffer','ON')
imshow(image256ord,map256ord)
hold on
```



```
plot(centro_laranjas(:,1),centro_laranjas(:,2),'r+') %centro das laranjas a vermelho  
plot(centro_laranjas(:,1),centro_laranjas(:,2),'b') %uniao dos centros das laranjas a azul
```

Detecção dos contornos e centros de massa das laranjas:



E finalmente temos as laranjas detectadas:

