Aluno: Eduardo Mafra Pereira
Email: emafra0@gmail.com

Trabalho 4

Visão computacional

Questão 1

Nesta questão foi preciso realizar procura de imagens através de outras imagens por meio de sua similaridade. Sendo assim foram propostos 3 algoritmos diferentes, cada algoritmo utilizando uma das funções de similaridades propostas em aula (SAD, SSD e ZNCC). O desenvolvimento do algoritmo se apresenta da seguinte maneira:

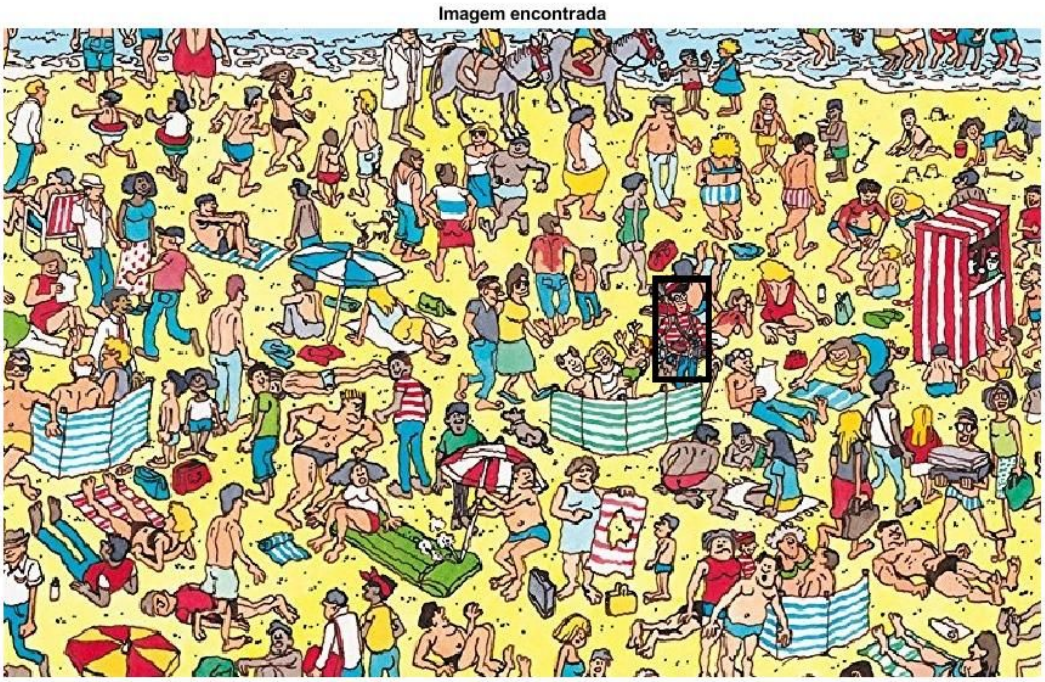
1.1 Função de similaridade SAD:

```
%SAD
clear all
clc

% Imagem processada:
Im = imread('Wally.jpg');
figure(1),imshow(Im);
title ("Imagem processada");
% Imagens a serem procuradas (neste relatório será feita a procura da
% imagem "Search2").
Search1 = imread('Search1.jpg');
Search2 = imread('Search2.jpg');
Search3 = imread('Search3.jpg');
Search4 = imread('Search4.jpg');

figure(2), imshow(Search2);
title("Imagem procurada");

%  Transformações da imagem para escala de cinzas e variável tipo double.
% (Estas transformações são muito  úteis para quando há necessidade de
% realizar operações algébricas nos elementos das matrizes):
Iml = double(rgb2gray(Im));
```

```
Search = double(rgb2gray(Search2));

% Captura as dimensões das imagens:
[n,m,c]= size(Iml);
[k,e,cl] = size(Search);

% Aplicação da função de similaridade SAD:
for i = 1:1:n-k
    for j = 1:1:m-e
        Proc(i,j) = sum(sum(sum(abs(Iml(i:i+k-1, j:j+e-1) - Search(:,:)))));

        loading = 100*(i/(n-k))
    end
end

% Realiza a procura da menor diferença de valores gerados através da
% aplicação da função SAD, encontrando portanto a imagem desejada:
[rows,cols] = find(Proc==min(min(Proc)));

% Aplica um contorno nos obejtos encontrados:
Im(rows:rows+4,cols:cols+e,:)=0;
Im(rows:rows+k,cols:cols+4,:)=0;
Im(rows-4+k:rows+k,cols:cols+e,:)=0;
Im(rows:rows+k,cols-4+e:cols+e,:)=0;
```

```
figure(3),imshow(Im);
title("Imagem encontrada");
```

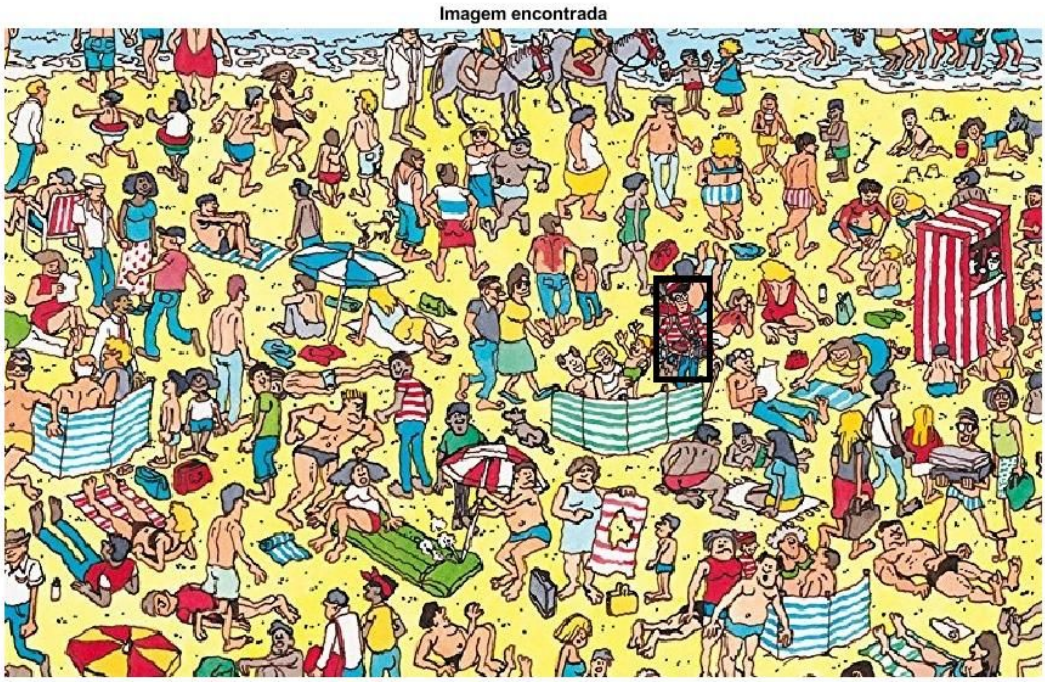
Resultado:



1.2 Função de similaridade SSD. Será utilizado o mesmo algoritmo do SAD porém aplicando a função de similaridade SSD:

```
for i = 1:1:n-k
    for j = 1:1:m-e
        Proc(i,j) = sum((sum(sum((Iml(i:i+k-1, j:j+e-1) - Search(:,:)).^2))));
        loading = 100*(i/(n-k))
    end
end
```

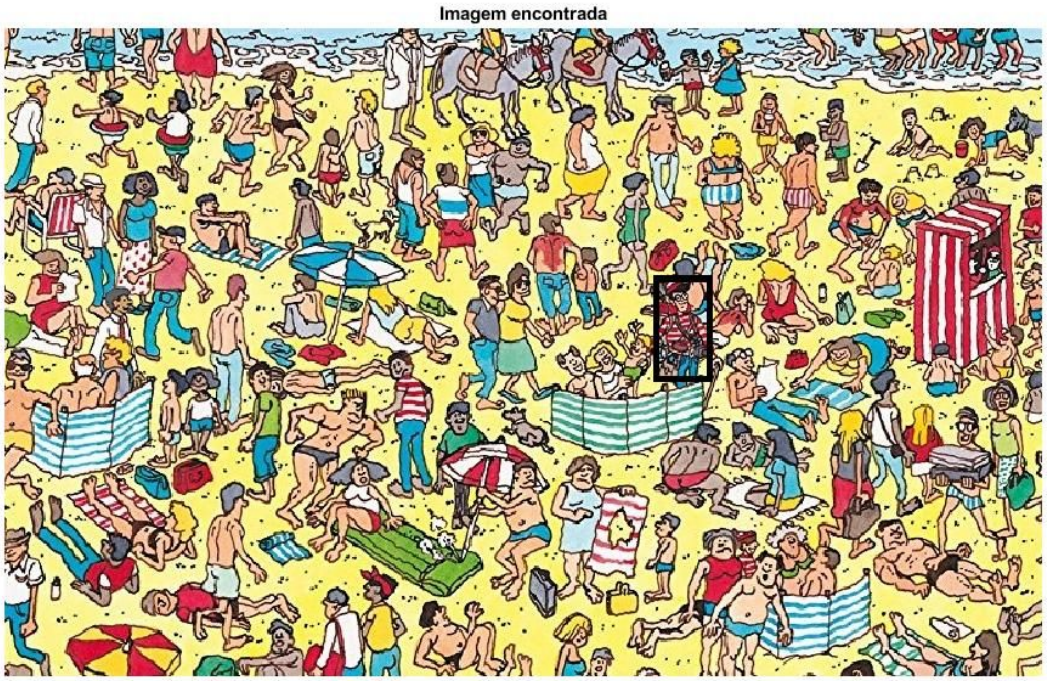
Resultado:



1.3 O processo é similar às questões anteriores porém utiliza a função de similaridade ZNCC e é encontrando agora o valor máximo gerado através da aplicação desta função de similaridade (ao contrário dos algoritmos anteriores que realizam a procurado do valor mínimo gerado através de suas funções de similaridade):

```
for i = 1:1:n-k
    for j = 1:1:m-e
        Proc(i,j) = sum((sum(sum((Iml(i:i+k-1, j:j+e-1) .* Search(:,:))))));
        Procl(i,j) = sqrt((sum((sum(sum((Iml(i:i+k-1, j:j+e-1)).^2)))))*(sum((sum(sum((Search(:,:)).^2))))));
        result(i,j) = Proc(i,j)/Procl(i,j);
        loading = 100*(i/(n-k))
    end
end

% Para a função de similaridade ZNCC deve-se utilizar o ponto de mais valor
% sobre a matrix "result" e nao o menor como nas questões anteriores.
[rows,cols] = find(result==max(max(result)));
```

Resultado:



1.4 Os mesmos algoritmos podem ser utilizados para encontrar as outras imagens solicitadas no laboratório 4, exercício 1.

Questão 2

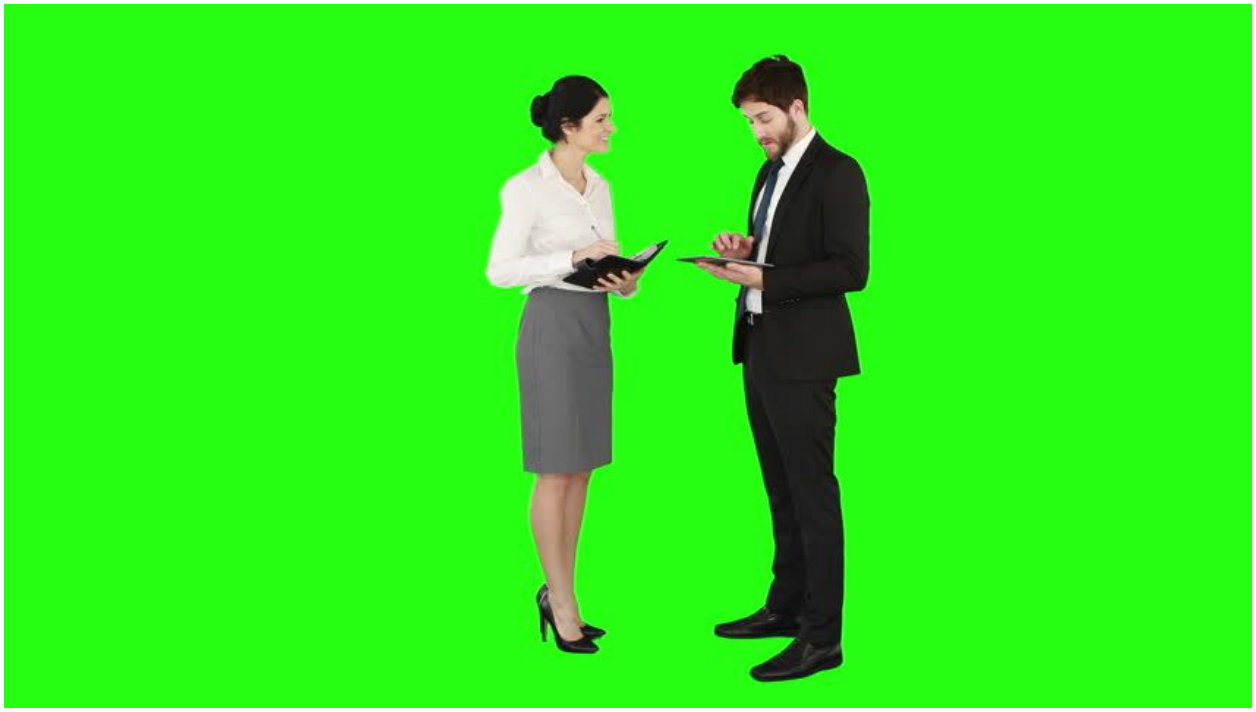
O problema solicita incrementar uma imagem de fundo verde em outra imagem fundo. Esta imagem terá o intuito de mostrar que as roupas utilizadas pelas pessoas na imagem tem um tecido que isola altas temperaturas deixando as pessoas mais confortáveis, sendo assim será proposta a imagem mais impactante possível. O algoritmo desenvolvido é apresentado da seguinte maneira:

```
clear all
clc

%  Captura as imagens a serem processadas:
Im = imread('Green.jpg');
fundo = imread('Inferno.jpg');

figure(4), imshow(Im);
title("Imagem pessoas");

%  Redimensiona a imagem para ser compatível com a imagem fundo:
croma = imresize(Im,0.5);
%  Transforma a imagem para uma matriz tipo double, isto permite realizar
% algumas operações necessárias para o processo do problema proposto:
im=double(croma);

%  Captura as dimensões das imagens:
[n,m,c] = size(croma);
[n1,m1,c1] = size(fundo);
```

```
%  Retira a parte verde da imagem "Green.jpg":
for i = 1:n
    for j = 1:m
        if ((im(i,j,1) >= 10  && im(i,j,1) <= 192) && (im(i,j,2) >= 190 &&
            im(i,j,2) <= 255) && (im(i,j,3) >= 0 && im(i,j,3) <= 218))
            im(i,j,1) = 255;
            im(i,j,2) = 255;
            im(i,j,3) = 255;
        end
    end
end
% Imagem com verde removido:
im = uint8(im);
% Transforma a imagem para escala de cinzas
im = rgb2gray(im);
% Equaliza a imagem para uma melhor transformação para uma matriz em preto
% e branco:
im = im-125;
%  Imagens de operação lógica para combinação da imagem "Green.jpg" com a
% imagem "Inferno.jpg":
escura = im2bw(im);
clara = not(escura);
```

```
figure(7),imshow(fundo);
title("Resultado final");
```

Imagens utilizadas:



Resultado:

Resultado final