UNIVERSIDADE FEDERAL DE SANTA CATARINA

Aluno: Aléxei Felipe Paim

Matrícula: 20250264

BLU3040-08754 (20222) - Visão Computacional em Robótica

Laboratório 2

2022

2. Descrição do problema:

Na atividade do laboratório dois, de Visão Computacional em Robótica foram, propostos dois problemas, aos quais deseja-se que sejam solucionados aplicando o conhecimento obtido em sala de aula. Assim os problemas foram os seguintes, desenvolver um código que seja capaz de reduzir e ofuscar digitalmente as linhas de uma tatuagem, de tal maneira que para isso seja aplicado um tipo de filtro de suavização e que fosse desenvolvido um matriz Kernel e então aplicar somente este filtro sobre os traços dessa tatuagem, como ilustra a figura 1.

(a) (b)

Figura 1 : (a) Imagem de entrada. (b) Imagem filtrada e processada .(MATSUO, 2022)

Na segunda atividade deste laboratório, foi proposto que fosse feito a implementação em forma de função para o algoritmo de detecção de borda de Canny. Sendo que este algoritmo é muito utilizado em diversos ramos da visão computacional e assim é essencial para que os alunos da disciplina compreendam seu funcionamento. Para isso o professor Matsuo, disponibilizou um passo a passo do que deveria ser feito para a obtenção do resultado final, assim como mostra a figura 2.

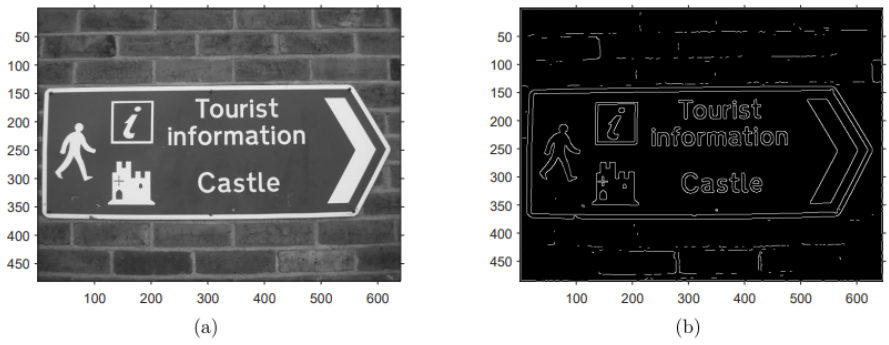


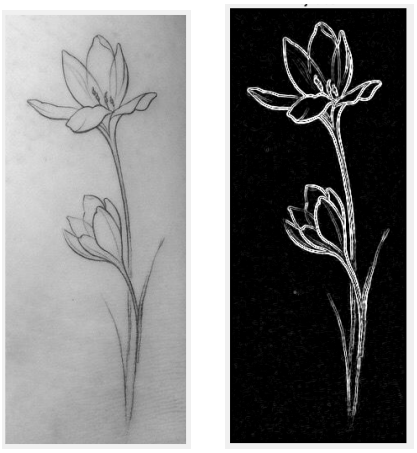
Figura 2 : (a) Imagem de entrada. (b) Imagem binária com as bordas na cor branca, obtida através do detector de borda de Canny. (MATSUO, 2022)

Descrição do algoritmo proposto:

Na atividade um do laboratório de visão computacional em robótica, inicia-se o algoritmo realizando a leitura da imagem da tatuagem que irá ser processada, e assim através da função *ginput,* selecionam-se quatro pontos ao redor da tatuagem a fim de montar uma imagem menor e com menos detalhes como ilustra a figura 3.

Figura 3 : seção da foto que será processada

Para dar sequência no algoritmo é realizado uma conversão da escala de cinza para que possa ser aplicado uma detecção de borda vertical e horizontal, com o intuito de conseguir encontrar os traços da tatuagem como mostra as imagens da figura 4.

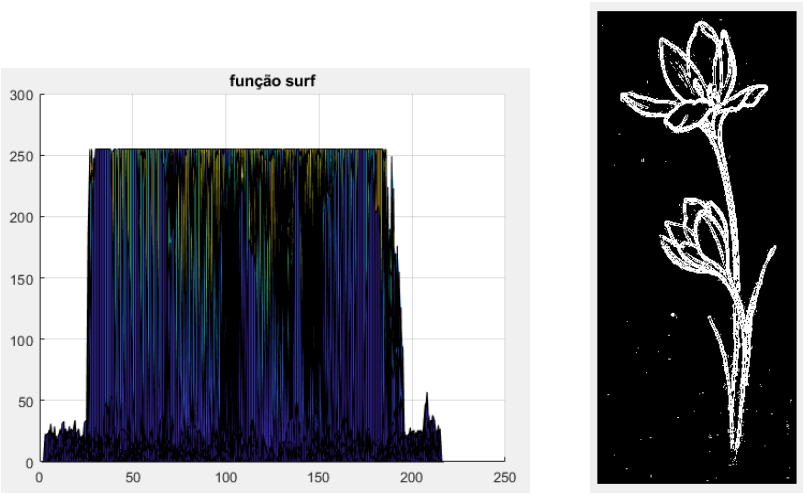


(a) (b)

Figura 4 : (a) Imagem em escala de cinza. (b) Imagem com detecção de borda

Agora, para que seja possível analisar melhor os pontos de borda, é feito uma limiarização sobre o item b da figura 4, a fim de transformar a imagem para o tipo *logical*. para isso é utilizado a função *surf* para que consiga-se aproximar um

valor de limiar. Portanto, nesta etapa os resultados ficam conforme ilustra as imagens da figura 5.



(a) (b)

Figura 5 : (a) Função surf para o limiar . (b) Imagem limiarizada do tipo logical

Então com a imagem do tipo *Logical* torna- se possível utilizar a função *Find,* que será responsável por encontrar as posições dos pixels com valor diferentes de 0, pois com estes pontos é realizado a substituição dos pixels na imagem sem a aplicação do filtro.

No entanto, para aplicar a filtragem de suavização na imagem, será utilizado a função *Filter2* com um *Kernel normalizado* e isso implica que não será possível utilizar a imagem no tipo RGB, assim para contornar esta limitação é realizado uma abertura em camadas RGB, e então aplicado o filtro, assim como mostra a figura 6.

```
%Abre a imagem cortada para as camada RGB
Rd = Is(:,:,1);
Gr = Is(:,:,2);
Bl = Is(:,:,3);

%% Kernel normalizado
w = 40;
K = 1/(w^2)* ones(w,w);

%% passa todas as camadas RGB pelo filtro Normalizado
R =filter2(K,Rd,'same');
R = uint8(R);

G =filter2(K,Gr,'same');
G= uint8(G);

B =filter2(K,Bl,'same');
B = uint8(B);

%Monta a imagem RGB com o filtro aplicado
I2(:,:,1)= R;
I2(:,:,2)=G;
I2(:,:,3)=B;
```

Figura 6: Imagem suavizada utilizando um kernel normalizado

Portanto, com a imagem filtrada e a imagem original cortada, é feito um laço de repetição para realizar a substituição dos pixels filtrados, na imagem não suavizada, de tal forma que é obtido o resultado a figura 7.

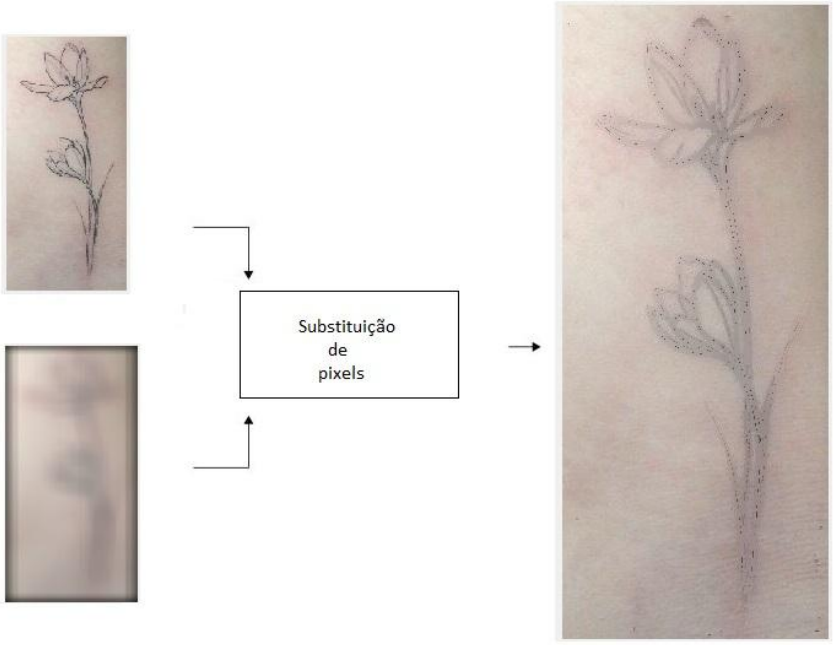


Figura 7: Processamento final da imagem

Por fim, é realizada operação de sobrepor a imagem processada na imagem de entrada e assim é obtido a imagem de saída figura 8, que corresponde ao resultado final deste algoritmo.

Figura 8: Imagem de saída

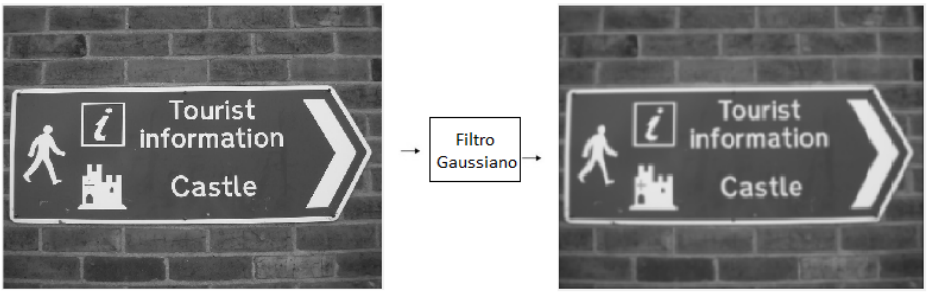
Na segunda atividade deste laboratório foi solicitado que fosse criado uma função no Matlab que realizasse o algoritmo de deteção borda de Canny. Para iniciar o código cria-se uma função chamada *f_detec_borda_canny,* que recebe como parâmetro uma imagem em escala de cinza, o valor de sigma para o filtro Gaussiano, o valor do limiar superior e valor do limiar inferior.

```
% função de algoritmo de detecção de borda

function Ib = f_detec_borda_canny(I, sig,Th, Tl);
```

O algoritmo de detecção de borda de Canny, pode ser implementado seguindo um processo de 5 etapas, que será descrito no desenvolver deste relatório.

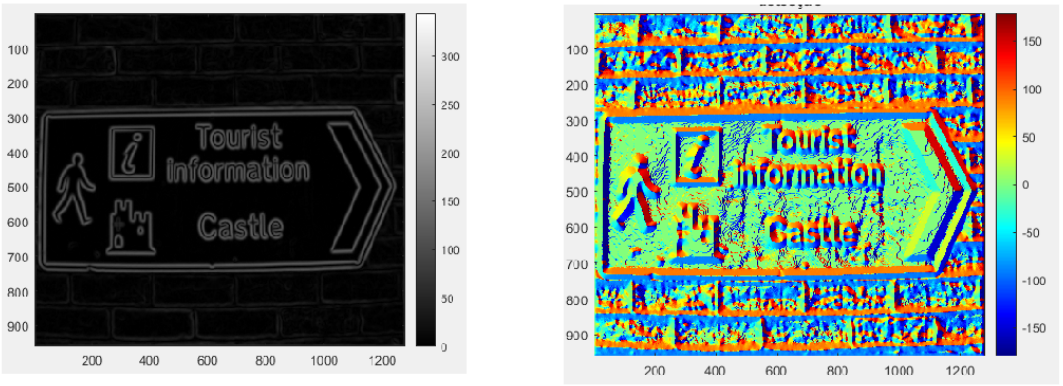
1.É pego a imagem de entrada da função e nela é aplicada um filtro de suavização, utilizando um filtro Kernel gaussiano com variância estabelecida por sigma, como mostra a figura 9.



(a) (b)

Figura 9 : (a) Imagem de entrada . (b) Imagem suavizada

2. Com a imagem suavizada realiza-se o processo de obtenção da imagem de Magnitude e da imagem de fase α, como ilustrado na figura 10



(a) (b)

Figura 10 : (a) Imagem da Magnitude . (b) Imagem da fase

3. Nesta etapa é realizado uma supressão de máximos locais, sendo que para isso primeiramente cria-se uma imagem com pixels nulos com o mesmo tamanho da imagem original, logo após verifica-se a direção do gradiente de cada posição da imagem de fase $\theta = \alpha(u, v)$.

Então é criado um laço de repetição que seleciona as posições vizinhas e adjacentes ao ângulo theta calculado no passo anterior. Assim é verificado se a magnitude do pixel central é maior que a de seus vizinhos selecionados. O processo é ilustrado na figura 11.

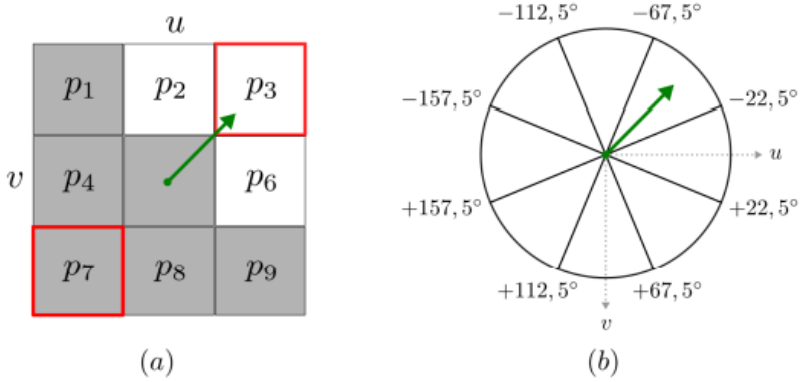


Figura 11 : (a) Pixels selecionados (com borda em vermelho) localizados na direção do gradiente do pixel central. (b) Ângulos de referência utilizados para a seleção dos pixels vizinhos.(MATSUO, 2022)

E em caso da magnitude do pixel central ser maior que a de seus vizinhos selecionados, é dito que este é um pixel de borda, então substituímos este ponto na imagem de pixels nulos, de tal forma que obtemos a imagem da figura 12.

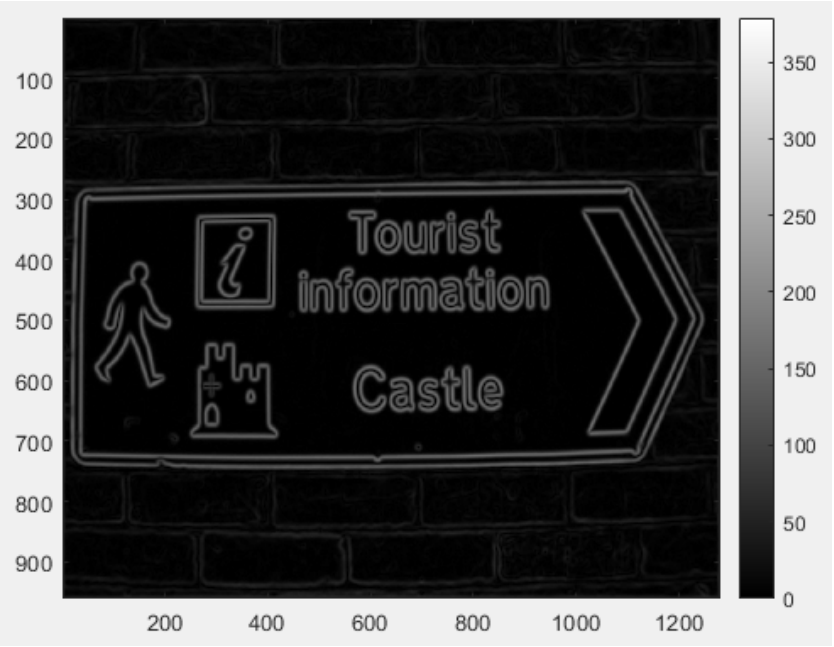


Figura 12 : (a) Imagem GN obtida ao final de etapa de supressão de não máximos locais

4. O Processo de limiarização duplo é feito através da definição de dois parâmetros de limiar, sendo um o Superior e outro o inferior que será utilizado para a composição da imagem de saída. Na sequência, gera-se duas imagens binárias a partir de comparações, como mostra a figura 13

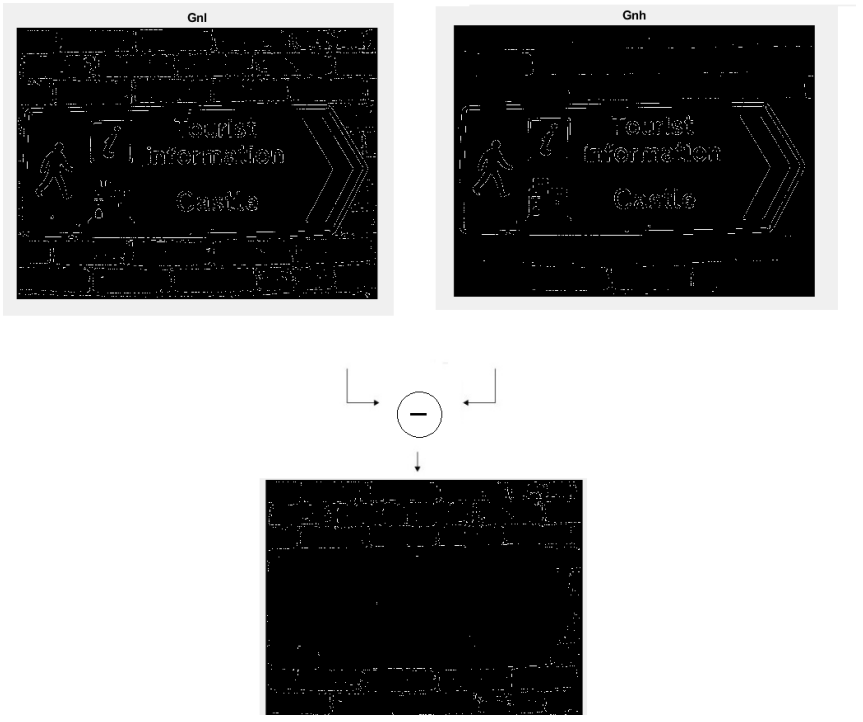


Figura 13 : Representação do processo de Limiarização duplo

5. Na etapa final é realizado a análise de conectividade, onde percorremos todos os pixels brancos da imagem de limiarização superior e para cada pixel branco desta imagem, seleciona-se uma janela 3x3, em uma matriz de zeros com mesmo tamanho e na mesma posição do pixel branco analiso, e assim se a janela possuir uns, marcamos este como válidos. A imagem de saída é representada na figura 14.

Figura 14 : Análise de conectividade

Por fim a imagem com a detecção de borda pelo algoritmo de Canny é formado pela soma da imagem da análise de conectividade mais a imagem do limiar superior, como mostra a figura 15.

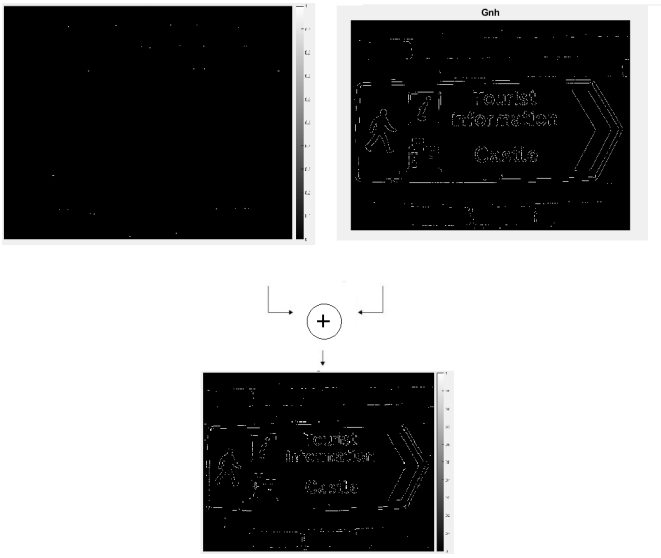


Figura 15 : Processo de composição final da imagem de borda

Resultados:

Com as duas atividades concluídas, nota-se que os resultados foram totalmente satisfatórios e cumprem o que foi solicitado, de tal forma que os resultados podem ser analisados através das imagens de saídas de cada um dos algoritmos. que estão representados na figura 16 e 17

.

Figura 16 :Imagem de saída da atividade 1

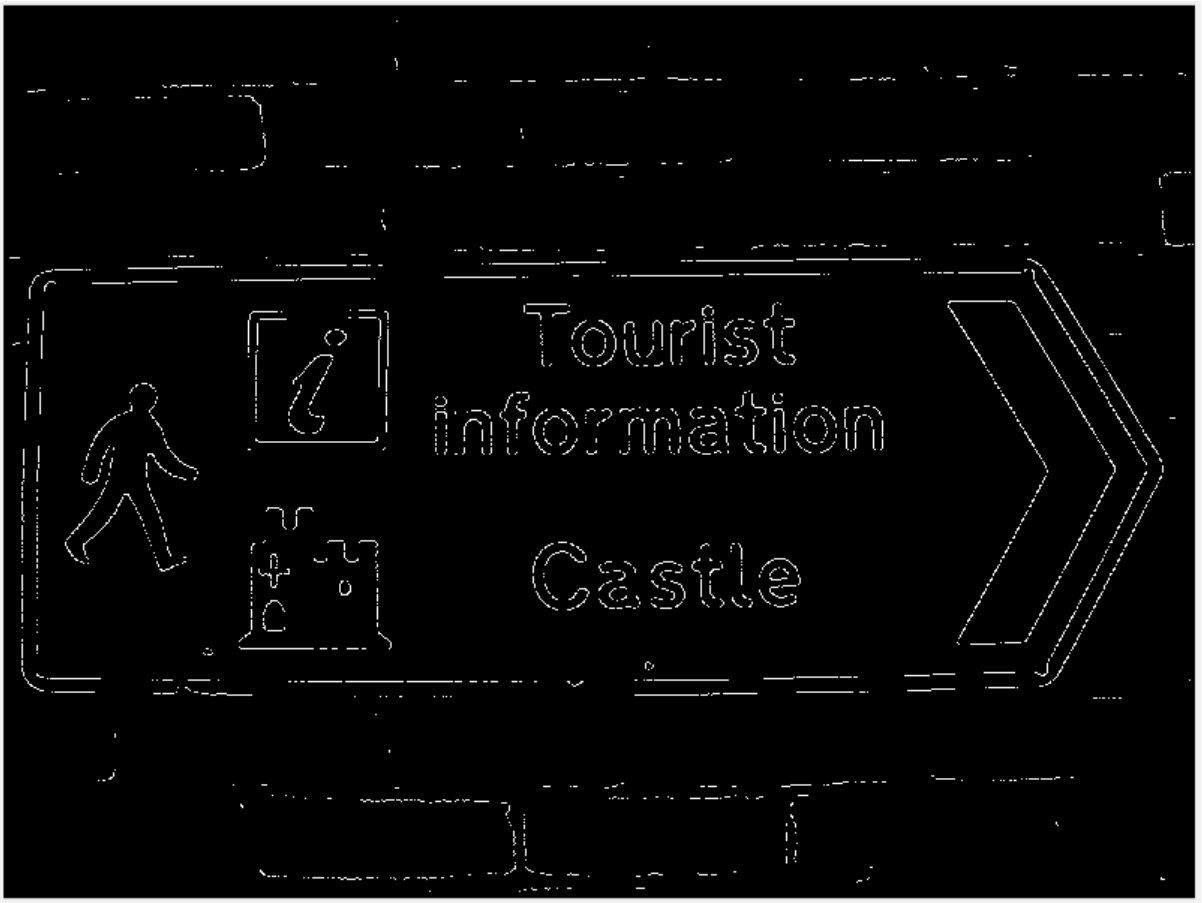


Figura 17 :Imagem de saída da atividade 2

Conclusão:

Por fim, concluímos que estas atividades de laboratório foram de suma importância para a fixação do conteúdo de processamento digital de imagem, de tal forma que após a realização desta atividade, nós alunos aprendemos um pouco mais sobre a aplicação operações espaciais

Referências:

MATSUO, Marcos. **LAB 2 - Operações Espaciais**. Disponível em: https://moodle.ufsc.br/pluginfile.php/5793118/mod_assign/introattachment/0/LAB1%20-%20Opera%C3%A7%C3%B5es%20di%C3%A1dicas%20e%20homografia%20planar.pdf?forcedownload=1. Acesso em: 16 out. 2022.