

LAB 2 - OPERAÇÕES ESPACIAIS

Observação: Entregar (via Moodle) relatório técnico e códigos produzidos até as 23h59min do dia 03/10.

1 Atividade 1

Na primeira atividade deste laboratório cada aluno deve implementar um código para reduzir digitalmente os traços de uma tatuagem, conforme ilustrado na Figura 1. Uma das formas possíveis para alcançar um bom resultado é suavizar a imagem apenas em torno das regiões próximas ao contorno da tatuagem. Além disso, é permitido utilizar a função `ginput()` para selecionar uma região retangular na imagem de entrada, indicando onde a filtragem deve ser realizada. Tente propor um kernel/filtro que seja mais adequado para esta aplicação (ou seja, diferente do kernel normalizado ou gaussiano). Nesta atividade, cada aluno deve implementar a sua própria função de filtragem 2D. A qualidade do resultado obtido será utilizada como critério de correção.



Figura 1: (a) Imagem de entrada. (b) Imagem processada.

2 Atividade 2

Na segunda atividade deste laboratório cada aluno deve implementar, **no formato de função**, o algoritmo de detecção de borda de Canny (proposto em 1986 por John F. Canny). O algoritmo de Canny é, ainda hoje, amplamente utilizado em diversas aplicações de visão computacional devido a sua confiabilidade na detecção das bordas de uma imagem. Um exemplo de resultado que pode ser obtido através do detector de borda de Canny é ilustrado na Figura 2.

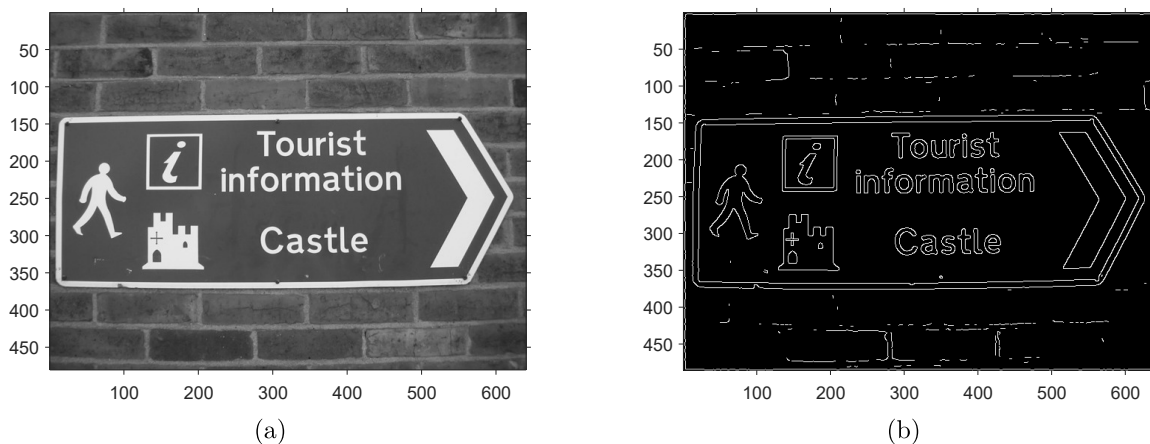


Figura 2: (a) Imagem de entrada. (b) Imagem binária com as bordas na cor branca, obtida através do detector de borda de Canny.

Especificamente, a execução do detector de borda de Canny pode ser descrita em 5 etapas, a saber:

1. A imagem de entrada \mathbf{I} é filtrada por um kernel gaussiano \mathbf{G}_σ (com variância σ) obtendo a imagem suavizada \mathbf{I}_s , conforme ilustrado na Figura 3.



Figura 3: (a) Imagem de entrada \mathbf{I} . (b) Imagem suavizada \mathbf{I}_s .

2. Computa-se a magnitude \mathbf{M} e a fase α do gradiente da imagem suavizada \mathbf{I}_s , obtendo resultados similares aos apresentados na Figura 4.

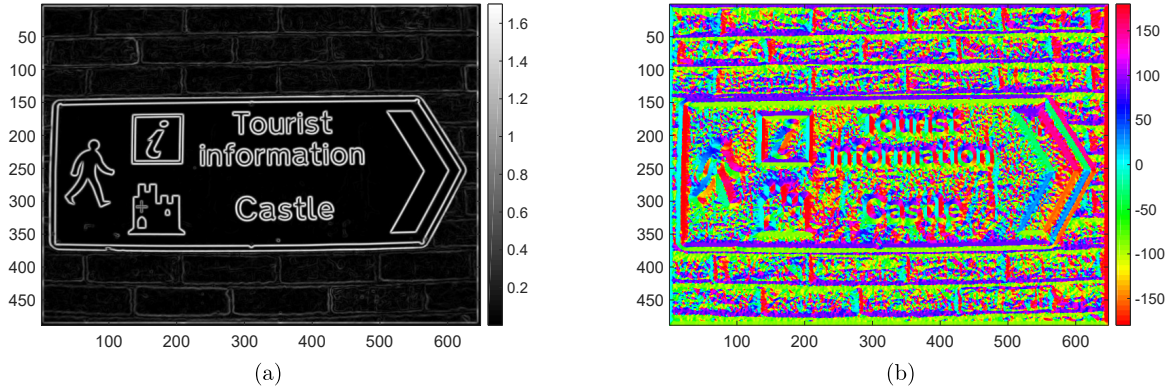


Figura 4: (a) Magnitude \mathbf{M} do gradiente da imagem suavizada \mathbf{I}_s . (b) Fase α do gradiente de \mathbf{I}_s .

3. Supressão de não máximos locais.

(3.a) Cria-se uma imagem \mathbf{G}_N (com pixels nulos), com a mesma dimensão de \mathbf{I}_s .

(3.b) Para cada pixel (u, v) de \mathbf{I}_s , verifica-se a direção do gradiente (ou seja, obtém-se $\theta = \alpha(u, v)$), conforme ilustrado na Figura 5.

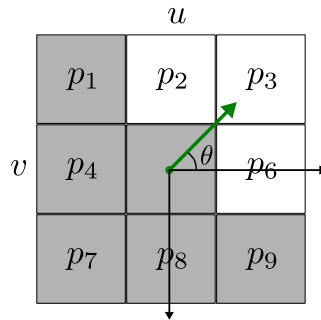


Figura 5: Obtenção da direção θ do gradiente do pixel (u, v) da imagem \mathbf{I}_s . O vetor de gradiente é representado pelo seta em verde.

- (3.c) Seleciona-se os pixels vizinhos adjacentes ao pixel (u, v) na direção do gradiente, conforme ilustrado na Figura 6.
- (3.d) Verifica-se se a **magnitude do gradiente** do pixel (u, v) (isto é, $\mathbf{M}(u, v)$) é maior do que a magnitude do gradiente dos **dois vizinhos** selecionados no passo anterior. Em caso afirmativo, tem-se que provavelmente o pixel (u, v) é um pixel de borda.

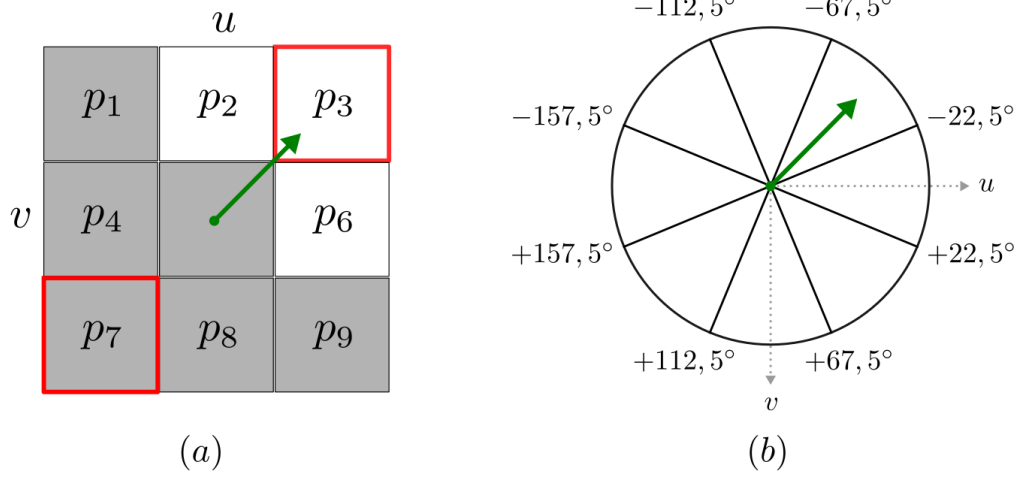


Figura 6: (a) Pixels selecionados (com borda em vermelho) localizados na direção do gradiente do pixel central. (b) Ângulos de referência utilizados para a seleção dos pixels vizinhos.

Assim, faz-se

$$\mathbf{G}_N(u, v) = \mathbf{M}(u, v)$$

caso contrário,

$$\mathbf{G}_N(u, v) = 0.$$

(3.e) Após realizar o procedimento anterior para todos os pixels, \mathbf{G}_N será uma imagem em escala de cinza (conforme a apresentada na Figura 7), com pixels não nulos representando possivelmente pixels de borda.

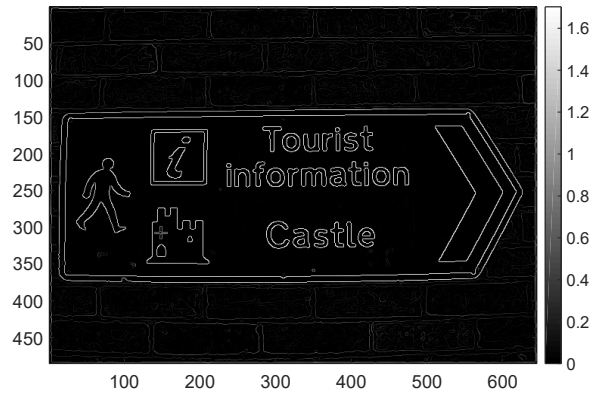


Figura 7: Imagem \mathbf{G}_N obtida ao final de etapa de supressão de não máximos locais.

4. Processo de limialização duplo.

(4.a) Escolhe-se dois limiares, um superior τ_H e outro inferior τ_L (isto é, com $\tau_H > \tau_L$).

(4.b) Na sequência, gera-se duas imagens binárias a partir das seguintes comparações:

$$\mathbf{G}_{NH} = \mathbf{G}_N \geq \tau_H$$

e

$$\mathbf{G}_{NL} = \mathbf{G}_N \geq \tau_L.$$

Um exemplo de resultado obtido nesta etapa é ilustrado na Figura 8. Note que \mathbf{G}_{NH} contém uma quantidade menor de pixels brancos do que \mathbf{G}_{NL} . Além disso, observe que até esta etapa, todos os pixels brancos de \mathbf{G}_{NH} também estão presentes em \mathbf{G}_{NL} .

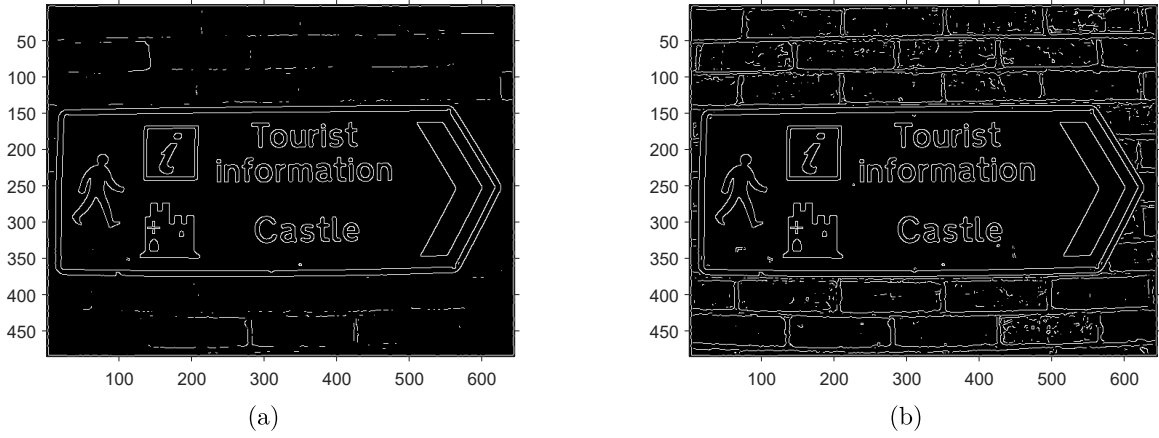


Figura 8: (a) Imagem \mathbf{G}_{NH} . (b) Imagem \mathbf{G}_{NL} .

(4.c) Na sequência, remove-se de \mathbf{G}_{NL} todos os pixels brancos que possui em comum com a imagem \mathbf{G}_{NH} . Para tal, \mathbf{G}_{NL} é atualizada através da seguinte operação diádica:

$$\mathbf{G}_{NL} = \mathbf{G}_{NL} - \mathbf{G}_{NH}.$$

Esta operação é realizada com o objetivo de obter uma imagem como a mostrada na Figura 9.

Neste ponto, diz-se que \mathbf{G}_{NH} e \mathbf{G}_{NL} contém pixels de borda **fortes** e **fracos**, respectivamente. Assim, considera-se que todos os pixels brancos de \mathbf{G}_{NH} são **válidos**, ou seja, são de fato pixels de borda. Assim, resta verificar quais os pixels brancos de \mathbf{G}_{NL} podem também ser considerados como pixels de borda válidos.

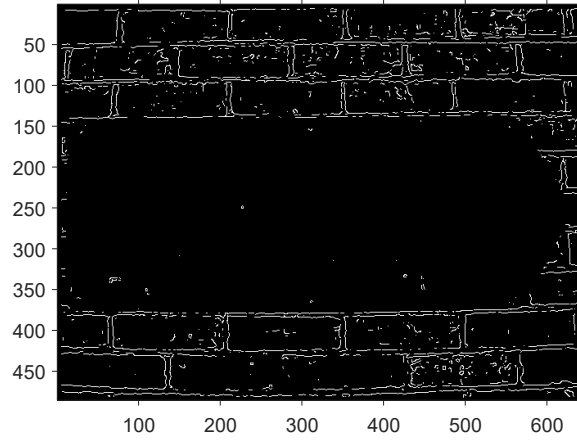


Figura 9: Imagem \mathbf{G}_{NL} atualizada.

5. Análise de conectividade.

- (5.a) Percorra todos os pixels de borda (ou seja, brancos) (u, v) da imagem \mathbf{G}_{NH} .
 - (5.b) Para cada pixel de borda (u, v) de \mathbf{G}_{NH} , selecione a região (janela 3×3) correspondente na imagem \mathbf{G}_{NL} .
 - (5.c) Na imagem \mathbf{G}_{NL} , marque como pixels de borda **válidos** todos os pixels branco na janela analisada.
 - (5.d) Após percorrer todos os pixels de borda de \mathbf{G}_{NH} , atribua valor nulo em todos os pixels que não foram validados na imagem \mathbf{G}_{NL} .
 - (5.e) A imagem de borda final será formada pelos pixels de borda válidos de \mathbf{G}_{NH} e \mathbf{G}_{NL} .
- Ou seja,

$$\mathbf{I}_b = \mathbf{G}_{NH} + \mathbf{G}_{NL}$$

onde \mathbf{I}_b é a imagem de borda final, similar a Figura 2 (b).

Especificamente, o algoritmo de detecção de borda de Canny deve ser implementado no formato de função, a qual deve possuir os seguintes argumentos de entrada: i) imagem de entrada \mathbf{I} em escala de cinza, ii) desvio padrão σ do filtro gaussiano (utilizado na primeira etapa de processamento), iii) limiar superior τ_H e iv) limiar inferior τ_L , utilizados no processo de limiarização duplo. Adicionalmente, como argumento de saída, a função deve fornecer a imagem de borda \mathbf{I}_b .

3 Orientações para elaboração do relatório técnico

A elaboração do relatório visa formalizar de forma textual as atividades realizadas, para tal é importante apresentar uma descrição do algoritmo desenvolvido e uma discussão acerca dos resultados obtidos. Para este laboratório, o relatório técnico deve conter as seguintes seções e informações:

1. **Identificação:** identificação do aluno e da atividade realizada.
2. **Descrição do problema:** nesta seção deve-se descrever o problema a ser solucionado. Para essa seção, é possível utilizar como base o enunciado do roteiro. Porém é importante não apresentar simplesmente uma cópia do roteiro, ou seja, descreva o problema com as suas palavras.
3. **Descrição do algoritmo proposto:** nesta seção, os principais pontos do algoritmo proposto devem ser descritos de forma detalhada e clara. Sugere-se que a descrição seja apresentada com o auxílio de diagramas e, também, através da apresentação de resultados parciais. Em outras palavras, evite descrever o algoritmo proposto apenas com texto. Em relação aos códigos, apresente apenas as seções/porções que considerar relevante. Ou seja, não é necessário apresentar uma cópia de todo código no relatório (lembre-se que o código deve ser enviado em arquivos separados).
4. **Resultados:** apresentar e discutir os resultados obtidos.