

---

**UNIVERSIDADE FEDERAL DE ALAGOAS**  
**INSTITUTO DE COMPUTAÇÃO**

**Visão Computacional**  
**Professor: Thales Vieira**

---

**1a lista de exercícios**

**5 de abril de 2022**

---

**Instruções:**

A lista deve ser respondida por grupos de até 2 pessoas (graduação) e individualmente (mestrado).

Resoluções idênticas de grupos distintos serão desconsideradas.

O código e as imagens devem ser anexadas a cada questão.

Data limite para entrega: 19/04/2022.

1. Implemente uma função que receba o tamanho  $n$  de um filtro espacial com dimensões  $n \times n$ , e a matriz com os pesos do filtro, e aplique o filtro numa imagem, gerando e salvando a imagem resultante no disco.
2. Implemente dois tipos de filtro para remoção de ruídos e aplique-os em três imagens distintas.
3. Pesquise um filtro passa-alta e aplique-o em 3 imagens distintas, variando o valor de  $n$  em 3 unidades em cada imagem. O que acontece quando  $n$  cresce?
4. Pesquise um filtro passa-baixa e aplique-o em 3 imagens distintas, variando o valor de  $n$  em 3 unidades em cada imagem. O que acontece quando  $n$  cresce?
5. Aplique o detector de canny usando a função *Canny()* do OpenCV em uma ou mais imagens de sua escolha. Experimente variar os parâmetros *threshold1*, *threshold2* e *apertureSize*. Explique com suas palavras a influência de cada parâmetro no resultado.
6. Um filtro passa-baixa bastante conhecido é o filtro Gaussiano. Através de uma máscara determinada por um desvio padrão  $\sigma$ , é possível fazer o que chamamos de suavização gaussiana de uma imagem. Considere

o filtro 5x5 de desvio padrão unitário abaixo:

$$g = \begin{bmatrix} 1/273 & 4/273 & 7/273 & 4/273 & 1/273 \\ 4/273 & 16/273 & 26/273 & 16/273 & 4/273 \\ 7/273 & 26/273 & 41/273 & 26/273 & 7/273 \\ 4/273 & 16/273 & 26/273 & 16/273 & 4/273 \\ 1/273 & 4/273 & 7/273 & 4/273 & 1/273 \end{bmatrix}$$

Crie uma imagem  $I$  de dimensões  $100 \times 100$  que contém um único valor não nulo localizado no centro da imagem. Utilizando o filtro gaussiano acima, calcule e exiba  $I * g$ , onde  $*$  é a notação para o operador de convolução.

**8.** Implemente uma função que reduza um ou aumente a resolução da imagem por fatores  $(sx, sy)$ . Você só pode usar o OpenCV para carregar a imagem. Se  $s < 1$ , implemente o algoritmo do slide 4. Se  $s > 1$ , use interpolação por vizinho mais próximo (slide 23). Aplique em três imagens distintas.

**9.** Escolha uma imagem interessante e gere uma pirâmide aplicando sucessivamente downsampling 3 vezes com fator  $1/2$ . Faça o mesmo com uma pirâmide gaussiana e compare os resultados. Mostre os resultados de cada pirâmide em uma única imagem lado a lado, como nos slides da aula (vai exigir algumas operações de translação de imagens). Obs.: aqui você pode usar funções do OpenCV.

**10.** Usando as funções que você implementou, crie funções que apliquem filtros para calcular as imagens representando  $\frac{\partial f}{\partial x}$ ,  $\frac{\partial f}{\partial y}$  e  $|\nabla f|$ . Implemente também as mesmas usando o operador de Sobel. Aplique-as diretamente em algumas imagens de sua escolha, e compare o efeito do ruído.