
**UNIVERSIDADE FEDERAL DE ALAGOAS
INSTITUTO DE COMPUTAÇÃO**

**Visão Computacional
Professor: Thales Vieira**

2a lista de exercícios

19 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: 03/05/2022.

1. Na documentação da OpenCV é possível encontrar diversos detectores e descritores¹. Usando um conjunto com pelo menos 6 imagens (você pode tirar fotos com seu telefone, por exemplo), experimente três detectores distintos nestas imagens, incluindo o detector de Harris. Pesquise e descreva resumidamente com suas palavras (até 200 palavras para cada) como cada um funciona, incluindo seus parâmetros e quais suas principais diferenças. Exiba resultados nas suas imagens, e em quais situações cada um funciona bem ou mal. Discuta seus parâmetros. Qual deles você achou melhor? Justifique.

2. Repita o mesmo procedimento acima para 3 tipos de descritores, incluindo o SIFT.

3. Repita o mesmo procedimento acima para 3 algoritmos de geração de correspondências (*matching*).

4. Escolha e descreva uma metodologia completa para geração de correspondências. Experimente 3 quantidades distintas de correspondências retornadas, variando parâmetros dos algoritmos. Aplique esta metodologia em duas imagens, exiba e discuta a qualidade dos resultados (correspondências corretas vs. *outliers*). Estes resultados devem ser exibidos com segmentos de reta entre imagens representando suas correspondências.

¹https://docs.opencv.org/4.5.3/db/d27/tutorial_py_table_of_contents_feature2d.html

5. Explique em palavras o que significa unicidade de pontos de interesse e mostre exemplos visuais usando 1 imagem.

6. Aplique o filtro LoG em pelo menos 6 escalas distintas (você pode usar a OpenCV). Em seguida, implemente um algoritmo que determina se um pixel (x, y, s) é um máximo local no espaço de posição-escala. Aplique este algoritmo em todos os pixels das imagens resultantes para encontrar estes extremos, e pinte estes pixels de uma cor diferente. Exiba as imagens resultantes. Obs.: você não precisa aplicar na imagem de menor e maior escala, nem nos pixels do bordo da imagem, visto que eles não tem uma vizinhança completa.