

Este documento contém parte do meu processo seletivo e propõe cumprir os dois dos objetivos propostos em MÓDULOS ESPECÍFICOS:

- COMPARAÇÕES DE PERFORMANCE
Foram implementados 3 *Features Descriptor* presentes no OpenCV, sendo eles BRISK, AKAZE e ORB.
- TRATAMENTO DA ENTRADA
O seguinte método de processamento de imagem foi implementado visando diminuir o ruído: Borramento Gaussiano.

Os seguintes aspectos foram considerados na avaliação: Diferença no consumo de tempo de processamento e de memória. A avaliação foi feita com auxílio da Ferramenta de Diagnóstico Do *Microsoft Visual Studio* em uma máquina com as seguintes especificações:

- SO: Windows 10
- Processador: AMD Ryzen 7 3700U, 2300 Mhz, 4 Núcleos, 8 Processadores Lógicos
- Placa de vídeo: integrada Radeon RX Vega 10

Os resultados obtidos estão presentes na Tabela 1 e no Gráfico 1:

Descriptor / Tratamento:	Processamento (%)	Memória (MB)
BRISK / -	39	122
BRISK / BLUR	35	98
AKAZE / -	59	82
AKAZE / BLUR	56	68
ORB / -	32	142
ORB / BLUR	28	128

Tabela 1 - Comparativo de Performance dos Métodos implementados

Processamento (%) and Memória (MB)

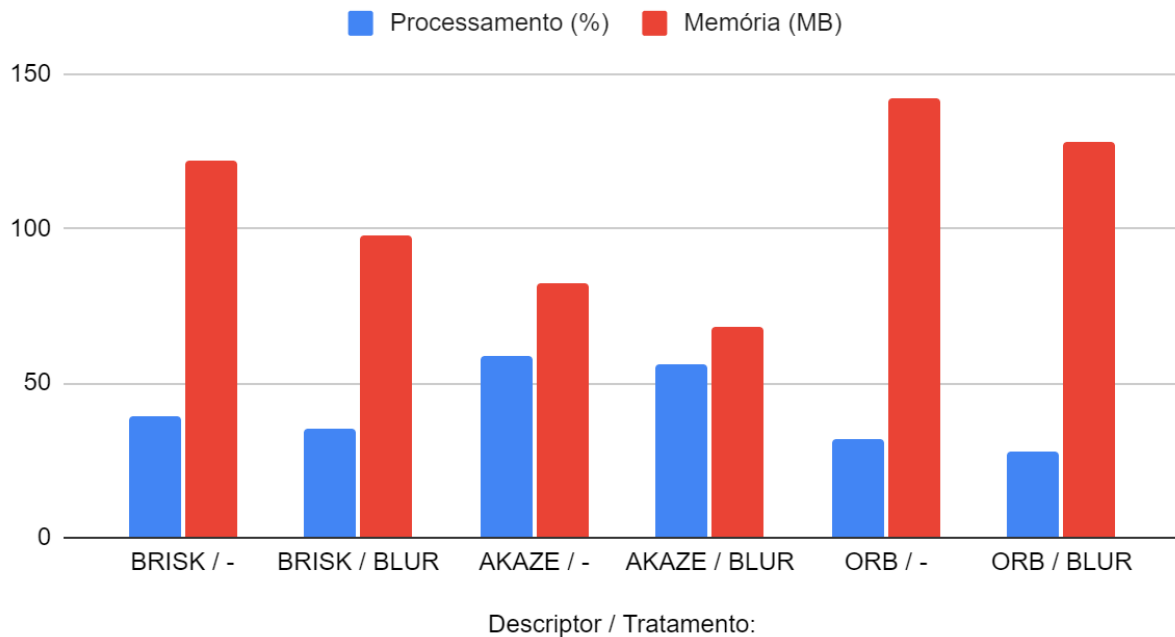


Gráfico referente aos dados da Tabela 1

Comentários finais e trabalhos futuros

Os 3 algoritmos analisados foram apresentados nos últimos 10 anos, mas o ORB se destaca por ser uma união de 2 algoritmos já muito eficientes (oriented FAST and Rotated BRIEF)[1]. O uso do borramento Gaussiano no tratamento da imagem mostrou uma ligeira melhora de processamento em todos os casos e uma redução expressiva de memória.

Os resultados obtidos estão de acordo com outros estudos comparativos[2] que concluiu que: "The feature-detector-descriptors can be rated for the speed of total image matching as: ORB(1000)>BRISK(1000)>AKAZE>KAZE>SURF(64D)>SIFT>ORB>BRISK>SURF(128D)".

Referências

[1]DOI: [10.1109/ICCV.2011.6126544](https://doi.org/10.1109/ICCV.2011.6126544)

[2]DOI: [10.1109/ICOMET.2018.8346440](https://doi.org/10.1109/ICOMET.2018.8346440)