

Trabalho de implementação 2

Instruções: este trabalho deve ser realizado individualmente ou em duplas. O(s) aluno(s) tem liberdade para escolha da linguagem na qual irá implementar os exercícios propostos, mas sugere-se fortemente o uso da linguagem **Python**. Até o dia **02/05/2021**, cada aluno/dupla deverá fazer o *upload* no Moodle da disciplina de um arquivo .zip, contendo

- Os códigos fonte com as rotinas solicitadas, além dos scripts usados para gerar os resultados solicitados para o relatório
- Um breve relatório (até 2 páginas de texto, mais os resultados visuais – imagens – solicitados nos enunciados) explicando testes realizados e a interpretação dos resultados.

1. Considere um par de imagens estéreo, já retificadas. Considerando uma das imagens como referência, o objetivo é calcular a disparidade d para cada pixel (x, y) da imagem, definida como aquela para qual um custo $C_0(x, y, d)$ é mínimo. Teste os seguintes critérios de custo para o cálculo da disparidade:

- a) soma dos quadrados das distâncias (SSD, ou *sum of squared distances*) Euclidianas (usando espaço de cores RGB) em uma vizinhança quadrada ao redor do pixel em análise de tamanho $N \times N$. Varie o tamanho da vizinhança e avalie o resultado. Em particular, teste a vizinhança 1×1 , ou seja, considerando apenas o pixel em análise.
- b) soma de alguma medida de distância robusta aplicada aos vetores de cor RGB. Sugere-se usar $\rho(d) = d^2 / (d^2 + \epsilon^2)$, onde ϵ é um parâmetro que você deve ajustar. Assim como no item (a), varie o tamanho da vizinhança e avalie os resultados em seu relatório.

2. Refine os resultados do exercício 1 usando um processo de agregação. Nesse processo, gera-se inicialmente um volume de custos $C_0(x, y, d)$ conforme o exercício 1. Na agregação, se escolhe uma vizinhança espacial 2D para cada “fatia de disparidade” d através de $C(x, y, d) = F(C_0(x, y, d))$, onde F denota a estratégia de agregação. Para achar a disparidade final de cada pixel (x, y) , escolhe-se a disparidade d_* que minimiza $C(x, y, d)$. Para cada uma das métricas de casamento de blocos do ex. 1, implemente a etapa da agregação de custos usando duas diferentes estratégias para F :

- a) filtro da média*
- b) filtro da mediana

A agregação deve ser feita em uma janela $M \times M$, e você deve testar alguns valores de M em seu relatório e explicar o que ocorreu com a variação de M .

Para seus testes, use imagens da base de dados Middlebury (<http://vision.middlebury.edu/stereo/data/>). Em particular, use os datasets **Cones** e **Teddy** disponíveis em <https://vision.middlebury.edu/stereo/data/scenes2003/>.

*Alunos de CMP197 devem **obrigatoriamente** explorar imagens integrais para calcular a média. Você não precisa implementar a rotina que calcula a imagem integral, pode usar uma rotina pronta como `cv2.integral` em Python.