

Fundamentos de Visão Computacional (CMP197 e INF01030)

Relatório do trabalho 2

José Bruno da Silva Santos (00569622)
Rodrigo Lusa (00216668)

A estereoscopia, ou visão estéreo, é o caso particular de processamento de visão computacional que utiliza por base imagens bidimensionais, adquiridas por um sistema de duas ou mais câmeras para perceber a dimensão de profundidade em uma cena [1]. O objetivo deste trabalho é avaliar qualitativamente e quantitativamente os mapas de disparidade obtidos usando a técnica clássica de casamento de janelas.

O trabalho está dividido em 3 seções. Inicialmente a 1ª seção apresenta as técnicas utilizadas sobre a imagem e na 2ª seção é apresentada a descrição do algoritmo implementado. Por fim, na última seção é apresentado os resultados e discussões do trabalho desenvolvido.

1. Operações sobre a imagem

- Soma das diferenças quadradas (SSD) é a soma da diferença quadrada dos valores de pixel em dois patches. Esse custo de correspondência é medido em uma disparidade proposta. Se A , B são patches para comparar, separados por disparidade d então SSD é definido como:

$$SSD(A, B) = \sum_{i,j} (A_{ij} - B_{ij})^2 \quad [2]$$

- Soma das diferenças absolutas (SAD) é uma medida da similaridade entre os blocos de imagens, testa se dois patches são semelhantes pela medida de distância. A , B são definidos de forma idêntica como:

$$SAD(A, B) = \sum_{i,j} |A_{ij} - B_{ij}| \quad [2]$$

Usar SAD: Em geral, as diferenças absolutas são mais robustas e melhores contra ruídos/outliers do que as diferenças quadradas, pois os outliers têm menos efeito.

- Erro RMS (Root-Mean Square): é definido como a raiz quadrada do quadrado médio, onde o quadrado médio é a média aritmética dos quadrados dos números.

$$RMSE = \sqrt{\sum_{i=1}^n \frac{(\hat{y}_i - y_i)^2}{n}} \quad [3]$$

2. Implementação

- Nossa implementação consiste na execução de um funcionalidade que calcula o mapa de disparidade entre duas imagens estéreo (perfeitamente alinhadas). Recebe como entrada as duas imagens (esquerda e direita) e gera como saída o mapa de disparidade. Para cada pixel na imagem da esquerda, precisamos encontrar o pixel correspondente na imagem da direita. Como os valores de pixel podem ser ruidosos e são influenciados por muitos fatores, como ruído do sensor, iluminação, desalinhamento etc. podemos ter que confiar em um grupo de pixels ao redor para comparação. Para isso usamos predefinições para o tamanho do bloco e o tamanho do bloco de pesquisa. O **tamanho do bloco** refere-se ao tamanho da vizinhança que selecionamos para comparar pixels da imagem esquerda e da imagem direita especificada como número de pixels em altura e largura. O **tamanho do bloco de pesquisa** refere-se a um retângulo no qual procuraremos o melhor bloco correspondente.
- Para um pixel na imagem à esquerda, selecionamos os pixels em sua vizinhança especificados como tamanho de bloco definido da imagem à esquerda. Calculamos a pontuação de similaridade comparando cada bloco da imagem da esquerda e cada bloco selecionado do bloco de pesquisa na imagem da direita. Deslizamos o bloco na imagem à direita em um pixel dentro do bloco de pesquisa. Armazenamos todas as pontuações de similaridade. Assim, encontramos a maior pontuação de similaridade de pixel da etapa anterior. Todos este processo é executado a cada pixel. A métrica de similaridade utilizada foi a soma da diferença absoluta entre os valores de cada pixel.

3. Resultados e Discussões

- Abaixo segue uma tabela com os valores gerados e juntamente com o script que contém o algoritmo seguem as imagens utilizadas e os mapas de disparidade gerados, onde o nome de cada arquivo é imagem_tamBloco_tamBlocoPesq.png e os resultados gerados no arquivo results.txt.

Imagem	Tam Bloco	Tam Bloco Pesq	SSD	SAD	SAD %	RMSE	Bad Pixels
cones	7	21	2750886400	19730939	9.94%	127.68	80.95%
teddy	7	21	1727766144	15719067	13.20%	101.19	83.74%
cones	9	27	2613726720	19168134	12.51%	124.45	83.76%
teddy	9	27	1624439552	15233268	15.88%	98.11	85.48%
cones	7	28	2591532544	19080505	12.91%	123.92	68.48%
teddy	7	28	1602931968	15141420	16.39%	97.46	71.87%
cones	7	35	2415175936	18389284	16.06%	119.63	77.69%
teddy	7	35	1466196224	14520690	19.81%	93.21	79.26%

cones	7	56	2032305920	17049071	22.18%	109.74	87.26%
teddy	7	56	1383995776	14128618	21.98%	90.56	82.00%
cones	12	60	2074421632	17186461	21.55%	110.87	84.15%
teddy	12	60	1421173504	14263872	21.23%	91.77	79.35%
cones	12	36	2399463424	18329644	16.34%	119.24	77.15%
teddy	12	36	1480955904	14573068	19.53%	93.68	77.62%
cones	5	15	2913737728	20491163	6.47%	131.40	71.34%
teddy	5	15	1866935552	16471091	9.04%	105.18	75.65%
cones	5	10	3029163264	20947402	4.39%	133.98	59.75%
teddy	5	10	1977840128	17070004	5.74%	108.26	62.24%
cones	3	9	3058668800	21059678	3.87%	134.63	53.57%
teddy	3	9	2003265024	17199447	5.02%	108.96	56.08%

Referências

- [1] Marr, D. Vision: A Computational Investigation into the Human Representation and Processing of Visual Information, Henry Holt and Co., Inc., New York, NY, USA, (1982).
- [2] Stereo and Disparity. Disponível em: <<https://johnwlambert.github.io/stereo/>>. Acesso em: 25 fev. 2023.
- [3] Statistics - Root Mean Square. Disponível em: <https://www.tutorialspoint.com/statistics/root_mean_square.htm>. Acesso em: 26 fev. 2023.