

Mapa de profundidade com câmeras estéreo

Mateus Berardo de Souza Terra
17/0018806
mateus.b.s.terra@gmail.com
Víctor Rodrigues Pacheco
17/0063879
victorrpacheco98@gmail.com

Departamento de Ciência da
Computação
Universidade de Brasília
Campus Darcy Ribeiro, Asa Norte
Brasília-DF, CEP 70910-900, Brazil,

Abstract

Mostrou-se o procedimento de criação de mapas de profundidade com base em imagens obtidas a partir de câmeras estéreo disponíveis em <http://vision.middlebury.edu/stereo/data/scenes2014/>. A criação do algoritmo fez uso da biblioteca multiplataforma OpenCV [1]. Considerando que as imagens fornecidas foram tiradas com câmeras previamente calibradas e que as imagens foram retificadas o primeiro passo foi a aplicado a técnica SumAll [2] para gerar o mapa de disparidade e logo em seguida permitir a síntese do mapa de profundidade.

1 Introdução

A soma das distâncias absolutas, também conhecida como $L1_norm$, é um algoritmo que compara duas imagens, ou procura uma certa imagem em outra. Para o cálculo, é utilizada uma imagem de referência de tamanho w que é comparada pixel a pixel com uma janela de mesmo tamanho na segunda imagem. A distância absoluta é calculada por meio da diferença entre os valores de cores.

Esse algoritmo pode ser utilizado para indicar a disparidade, demonstrada na figura 1, de imagens estéreo, realizadas por duas câmeras(a), embora não seja o melhor algoritmo, uma vez que não lida bem com oclusões ou mesmo deslocamentos.

1.1 Repositório

O código desenvolvido nesse projeto estão, assim como o histórico de desenvolvimento, disponíveis na plataforma GitHub em:

<https://github.com/MatTerra/StereoVision>.

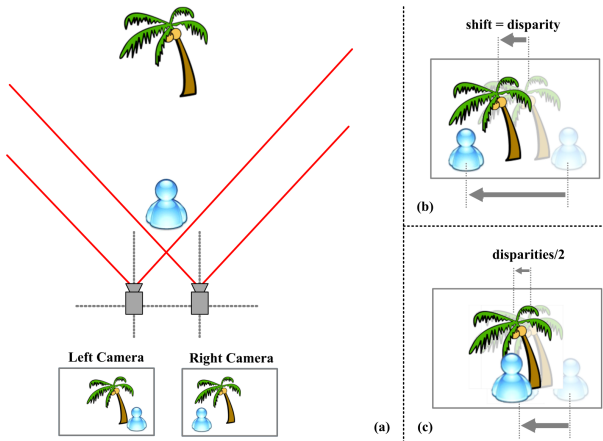


Figure 1: Disparidade.

https://cdn-images-1.medium.com/max/1600/1*wSyFon7Qm_DV165CZYMe8w.png

2 Desenvolvimento

2.1 Sistema

Todos os códigos foram desenvolvidos e testados em uma máquina virtual Oracle VirtualBox [1] com as seguintes características e em um computador com as configurações descritas em seguida:

- Versão VM 6.0.4
- Versão OS: Ubuntu 16.04 LTS 64-bits
- Versão Python: 3.5
- Versão OpenCV: 3.2.0

Segunda máquina:

- Versão OS: Manjaro 18.04
- Versão Python: 3.5.6
- Versão OpenCV: 3.2.0

3 Metodologia

3.1 Requisito 1

A execução deste requisito inicia pela implementação do algoritmo SAD. A implementação foi inicialmente testada na imagem Jade Plant, disponível em [2].

Além da localização, utilizamos a biblioteca OpenCV para montar um mapa de disparidade das imagens. Testamos esse algoritmo nas duas imagens, obtendo melhores resultados na Motocicleta, disponível em [3].

4 Resultados

4.1 SAD

Com o SAD, processamos a imagem Jade Plant. Nela, o casamento de pontos não obteve resultados satisfatórios utilizando o SAD, embora a introdução de um offset na busca para restringir a busca a uma certa janela específica na imagem destino tenha melhorado os resultados.

Este offset foi baseado na distância média obtida manualmente entre pontos da imagem. Podemos ver um caso em que a busca é bem sucedida na figura 2 e outro caso em que a busca não retornou a posição correta na figura 3.

4.2 Disparidade

A escolha do número de disparidades e o mínimo valor que ela pode assumir criam um offset na imagem, ocasionando um bloco não processado. Na figura 4, podemos ver o mapa obtido para a cena da motocicleta [5]. É evidente que uma parte considerável da imagem não foi corretamente processada, sendo apenas ignorada. No mapa, pixels mais claros indicam maior disparidade.

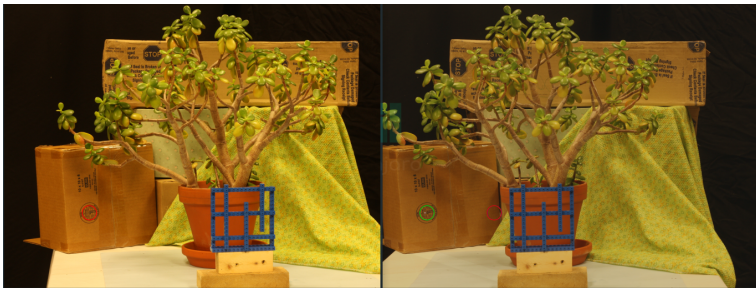


Figure 2: Resultado 1 correto.



Figure 3: Resultado 1 não correto.

5 Conclusão

Para a correta construção do mapa de profundidade são necessários 4 procedimentos:



Figure 4: Mapa de disparidade da motocicleta.

- Calibrar as câmeras;
- Retificar as imagens;
- Computar a Disparidade;
- Estimar a Profundidade.

Há dificuldade para a retificar as imagens e computar a disparidade principalmente por oscilações de brilho e pela presença de áreas de baixo contraste.

Foi notado também a impossibilidade de identificação da profundidade em áreas de oclusão (áreas capturadas apenas por uma câmera) pelo método adotado.

References

- [1] Loraof. Sum of absolute differences, 2019. URL "https://en.wikipedia.org/wiki/Sum_of_absolute_differences".
- [2] OpenCV.prg. Comunidade opencv, 2019. URL "<https://opencv.org/>".
- [3] Oracle. Oracle vm virtualbox, 2019. URL <https://www.virtualbox.org/>.
- [4] Middlebury University. Jadeplant - perfect, 2019. URL <http://vision.middlebury.edu/stereo/data/scenes2014/datasets/Jadeplant-perfect>.
- [5] Middlebury University. Motorcycle - perfect, 2019. URL <http://vision.middlebury.edu/stereo/data/scenes2014/datasets/Motorcycle-perfect>.