

Visão Estéreo com OpenCV em Python

Matheus Barbosa de Miranda
matheusbmiranda@gmail.com

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

Abstract

Este trabalho consiste em estudar os métodos de reconstrução 3D para mapas de profundidade e disparidade a partir de um par de imagens estéreo da base de dados de Middlebury e Furukawa-Ponce.

1 Introdução

O problema de visão estéreo envolve a derivação das coordenadas do mundo -bem como a profundidade- da imagem real a partir de duas fotos simultâneas de câmeras diferentes. As câmeras devem necessariamente estar a uma distância horizontal da outra e podem estar rotacionadas. A álgebra por trás deste problema é conhecida como geometria epipolar.

1.1 Geometria epipolar

A figura 1 ilustra um exemplo de um par estéreo. A distância entre os centros das câmeras C e C' em 1 (a) é chamada *baseline*. Note que a projeção do ponto X na esquerda e na direita são os pontos x e x' , respectivamente. O ponto representado por e e e' em 1 (b) são chamados epípolos. A reta formada pelos pontos x' e e' é chamada a linha epipolar de x .

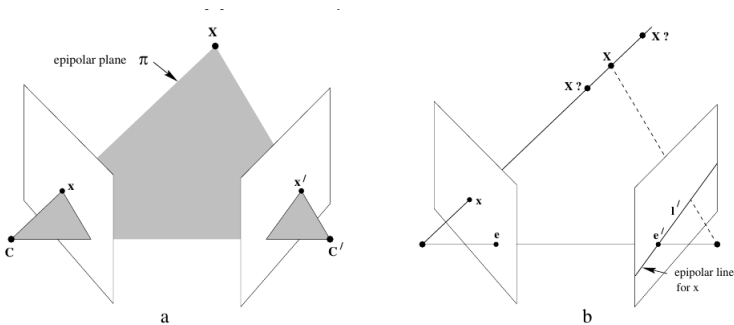


Figura 1: Par estéreo.[1]

O plano π , plano epipolar, é delimitado como mostrado na figura 1(a). Pontos em posições diferentes geram planos, que são sempre rotações em torno da *baseline*, formando o chamado pincel de planos epipolares[1].

1.2 Matriz fundamental

A matriz fundamental F obedece a condição $x'Fx = 0$, sendo x e x' os pontos da direita e da esquerda, respectivamente. F é de rank 2 e possui sete parâmetros: dois para cada epipolo e três da homografia dos planos das imagens [1]. Outra propriedade da matriz F é que para todo ponto x sua linha epipolar correspondente é dada por $l' = Fx$ [1]. De forma similar, o epipolo e' satisfaz $e'^T(Fx) = 0$ [1].

2 Metodologia

2.1 Correspondência entre pontos

A primeira parte do experimento consistiu em achar a correspondência de um ponto da imagem da esquerda, à escolha do usuário, e encontrar o ponto correspondente na imagem da direita. Foram utilizadas imagens da base de dados estéreo de Middlebury[2], que consiste em pares de fotos cujas câmeras estão alinhadas e com a mesma orientação, ou seja, as linhas epipolares são horizontais. As matrizes de calibração são dadas na própria base de dados, tendo sido obtidas conforme [2]. Foram consideradas apenas os pares *perfect*.

Para achar o ponto correspondente na imagem da direita foi utilizado o algoritmo de coeficientes normalizados do OpenCV, `cv2.TM_COEFF_NORMED`[3]. O usuário digita o tamanho da janela de template que será usada no algoritmo, instanciando uma classe *Window* que gera a janela. Como não foi aplicado nenhum *padding*, pontos muito próximos à borda são considerados inválidos. O resultado dado pelo algoritmo é uma nova coordenada na imagem da direita que corresponde à da esquerda. Um quadrado verde do tamanho da janela é desenhado e mostrado na tela, bem como as coordenadas da imagem da direita.

Tendo os dados da projeção de um mesmo ponto em duas imagens bem como a distância focal e a *baseline* das câmeras, é possível calcular a profundidade para todos os pontos que possuem correspondência, bem como as coordenadas X e Y do mundo real.

2.2 Mapa de disparidade e de profundidade

2.2.1 Base de dados da Middlebury

Criar o mapa de disparidade é uma tarefa que envolve uma manipulação algébrica para obter a matriz fundamental do par estéreo. Entretanto como a base de dados de Middlebury já trás imagens retificadas, tais contas serão simplificadas. O algoritmo do OpenCV utilizado foi o `stereoSGBM_matcher` para a correspondência estéreo e `cv2.createDisparityWLSFilter` para a criação do mapa de disparidade. O algoritmo SGBM (*Semi-Global Block Matching*) utiliza como principais parâmetros o tamanho de janela, disparidade mínima e número de disparidades. Pontos que não possuem correspondência são por padrão colocados como 0. Nesse caso, é fácil notar que apenas pontos na faixa de $([0, height], [0, baseline])$ não terão correspondência, já que as linhas epipolares são horizontais. Assim é possível identificar se um ponto que tem valor de disparidade nulo é um ponto de não correspondência ou se a disparidade é de fato zero naquele ponto.

3 Resultados e Análise

Clicando no ponto (434,337), o algoritmo de correspondência para uma janela de tamanho 11 px foi o ponto (389,337). Nota-se que a coordenada x permanece inalterada, o que indica que de fato não há alteração na altura correspondente. O algoritmo funciona bem com janelas maiores (aproximadamente 55 px). Um possível motivo para isso é que janelas menores têm maiores chances de encontrarem bordas ou outras features que destaquem a região. Áreas "lisas" como paredes ou com luz uniforme tendem a errar na correspondência. As imagens resultantes para um clique são mostradas nas figuras 2 e 3. As coordenadas do mundo para o mesmo ponto foram dadas por (618.44,411.06,153.1) mm.

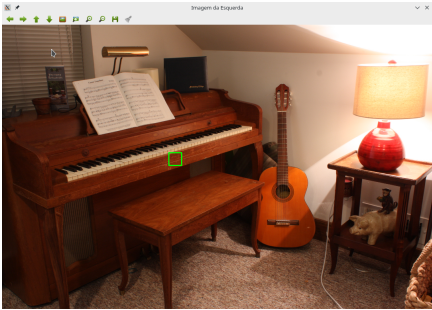


Figura 2: Imagem da esquerda

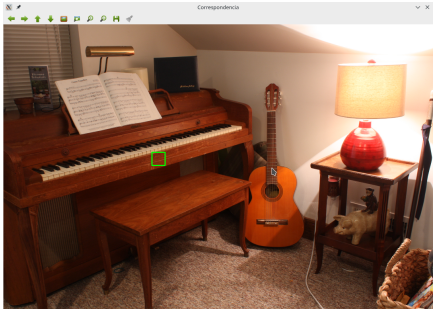


Figura 3: Imagem da direita

O mapa de disparidade para o cenário piano e playroom estão exibidos nas figuras 4 e 5, respectivamente. O mapa de profundidade foi obtido utilizando os pontos Z das coordenadas do mundo transformados pra unsigned int de 8 bits, resultando nas figuras 6 e 7. As distâncias foram normalizadas de acordo com a maior distância possível em cada imagem. Por ser uma imagem com pouca profundidade máxima muito baixa, a imagem playroom ficou muito escura e com muitos pontos marcados em branco, o que possivelmente indica que a normalização foi prejudicada.

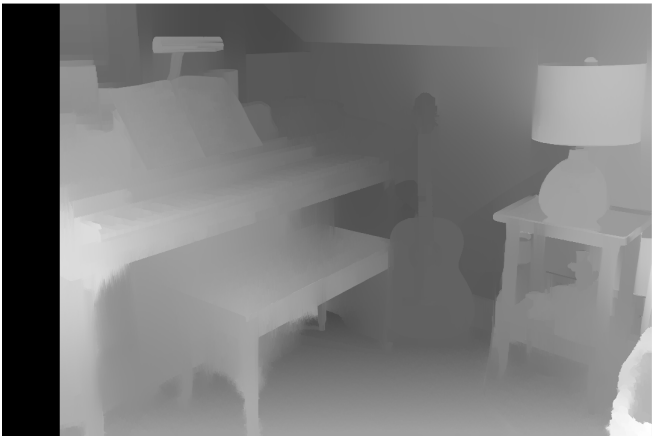


Figura 4: Mapa de disparidade - Piano (perfect)

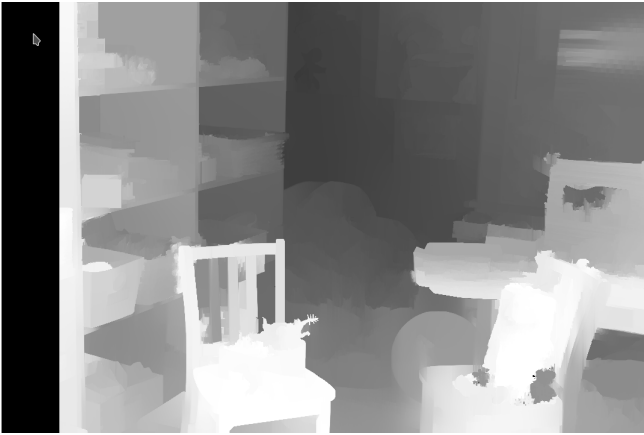


Figura 5: Mapa de disparidade - Playroom (perfect)



Figura 6: Mapa de profundidade - Piano (perfect)



Figura 7: Mapa de profundidade - Playroom (perfect)

4 Conclusão

O algoritmo de correspondência entre imagens estéreo funciona melhor com janelas menores, o que sugere que quanto mais informação a janela tiver melhor a chance de acerto. As disparidades tiveram muita distorção, o que sugere que possivelmente a calibração realizada pelos parâmetros não foi perfeita. A imagem do playroom é interessante, pois a profundidade de todos os pontos é relativamente parecida, o que tornou o mapa de profundidade muito escuro, dificilmente sendo possível identificar alguma forma.

Referências

- [1] G. Bradski and A. Kaehler. *Learning OpenCV*. O'Reilly Media, 2008.
- [2] R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, ISBN: 0521623049, 2000.
- [3] Middlebury. *Stereo datasets with ground truth (2014)*. Visitado em 3 de Outubro de 2019.
- [4] Daniel Scharstein. High-resolution stereo datasets with subpixel-accurate ground truth. 2014.
- [5] OpenCV Team. Opencv official documentation. https://opencv-python-tutroals.readthedocs.io/en/latest/py_tutorials/py_imgproc/py_template_matching/py_template_matching.html Data de visita: 3 de Outubro de 2019.