

Projeto Demonstrativo 3

Raphael Soares 14/0160299
raphael.soares.1996@gmail.com

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

Abstract

Todos nós estamos familiares com a capacidade de imageamento estéreo que nossos olhos nos fornecem. Em qual grau podemos simular essa capacidade em sistemas computacionais? Os computadores realizaram essa tarefa achando correspondências entre pontos que são vistos pelas duas câmeras. Com essa correspondência e com a distância de separação entre as duas câmeras conhecida é possível calcular a localização tridimensional dos pontos. Esse segundo projeto tem como objetivo principal explorar e desenvolver algoritmos para extração de mapas de profundidade a partir de pares estéreo de imagens. Esses mapas foram obtidos a partir do mapa de disparidade, que contém informação de disparidade dos pontos correspondentes vistos pelas duas câmeras. Para as imagens obtidas de câmeras que não estavam alinhadas em paralelo foi necessário retificá-las antes. Além disso, medidas de um objeto de uma imagem foram estimadas calculando a distância da localização tridimensional dos pontos, assim como no projeto demonstrativo 2.

1 Introdução

Nós achamos pontos correspondentes em nossos olhos esquerdos e direitos e usamos isso para trabalhar o quão longe algum objeto está de nós. Com apenas um olho nós temos algumas pistas monoculares que podemos usar para estimar profundidade, entretanto o verdadeiro “3D”, a verdadeira percepção de profundidade só existe quando temos dois olhos. Com um único olho é possível obter apenas deduções, como saber a distância de um objeto observando o tamanho dele em dois instantes de tempo diferentes. Os computadores realizam essa tarefa de imageamento estéreo dos nossos olhos achando correspondências entre pontos que são vistos por duas câmeras. Para computadores, apesar da busca de pontos correspondentes ser computacionalmente cara, é usado o conhecimento de geometria do sistema para limitar a busca o máximo possível [4]. Na prática, o imageamento estéreo feito nesse projeto envolveu 3 passos, já que as imagens usadas foram obtidas a partir de duas câmeras:

1. Ajuste dos ângulos e das distâncias entre as câmeras, que é conhecido como retificação. A saída desse passo são as imagens retificadas e alinhadas por linha¹.

© 2018. The copyright of this document resides with its authors.
It may be distributed unchanged freely in print or electronic forms.

¹A principal informação que o computador precisa para fazer imageamento estéreo é saber onde estão nossas câmeras. Note que no caso dos nossos olhos, o cérebro já sabe onde estão nossos olhos e eles já estão “alinhados por linha”

- Busca das mesmas características na visão das duas câmeras (que também poderiam estar orientadas verticalmente, mudando assim as disparidades), um processo conhecido como correspondência. A saída desse passo é o mapa de disparidade, onde as disparidades são as diferenças no eixo x nos planos da imagens das mesmas características vistas na câmera da esquerda e da direita: $x_l - x_r$.
- Sabendo o arranjo geométrico das câmeras, é possível transformar o mapa de disparidade em profundidade, usando triangulação. Esse passo é chamado de reprojeção e a saída é o mapa de profundidade.

Normalmente, seria necessário um passo adicional para remover as distorções radiais e tangenciais da lente antes da retificação. Entretanto, as imagens usadas tanto no requisito 1 quanto no 2 já estão sem distorção.

2 Metodologia

Nesta seção são apresentados os métodos e procedimentos utilizados em cada um dos requisitos para obter os resultados pedidos.

2.1 Requisito 1

No Requisito 1 foi necessário fazer a correspondência estéreo (casamento de pontos tridimensionais em visões diferentes da câmera) entre as duas imagens. A título de comparação, dois algoritmos foram utilizados para fazer a correspondência estéreo. Ambos algoritmos de casamento estéreo servem ao mesmo propósito: converter duas imagens, uma esquerda e uma direita, em uma única imagem de profundidade. Esta imagem basicamente irá associar com cada pixel uma distância das câmeras para o objeto que esse pixel representa.

O primeiro, denominado *block matching* (*BM*) é um algoritmo rápido e efetivo que é similar ao desenvolvido por Kurt Konolige [1]. Ele funciona usando pequenas janelas de “somadas de diferenças absolutas” (*SAD*) para encontrar pontos correspondentes entre as imagens estéreo retificadas da esquerda e da direita. Este algoritmo encontra somente pontos com alta correspondência entre as duas imagens (alta textura). Assim, em uma cena altamente texturizada todos os pixels vão ter profundidade computada. Em uma cena com pouca textura, como um corredor, poucos pontos devem registrar profundidade.

O segundo é conhecido como *semi-global matching* (*SGBM*) *algorithm*. SGBM, uma variação do SGM introduzido em [2], difere do BM em dois aspectos. O primeiro é que o casamento é feito em nível de subpixel usando a métrica Birchfield-Tomasi [3]. A segunda diferença é que o SGBM tenta impor uma limitação global de suavidade na informação de profundidade computada. Esses dois métodos são complementares, no sentido que o BM é mais rápido, mas não fornece a confiança e acurácia do SGBM.

Ambos os algoritmos que são implementados pela OpenCV [4] são melhor detalhados e explicados em [5].

2.1.1 Block Matching

O algoritmo estéreo BM implementando na OpenCV é uma versão modificada do que se tornou uma das técnicas canônicas para computação estéreo. O mecanismo básico é retificar e alinhar as imagens de tal forma que as comparações precisem ser feitas apenas em linhas

individuais, e então ter um algoritmo que procura linhas nas duas imagens para grupos correspondentes de pixels. O resultado é um algoritmo confiável que é vastamente usado. Existem três estágios para o algoritmo estéreo bm, que funciona em pares de imagens retificadas e sem distorção:

1. Pré-filtragem para normalizar o brilho da imagem e realçar a textura.
2. Busca por correspondência ao longo das linhas horizontais epipolares usando uma janela SAD.
3. Pós-filtragem para eliminar correspondências ruins de casamentos.

2.1.2 Semi-global block matching

O algoritmo SGM, que deriva o SGBM implementado pela OpenCV, possui várias novas ideias, mas um custo computacional bem maior que o BM. As mais importantes novas ideias introduzidas no SGM são o uso de informação mútua como uma medida superior de correspondência local e o reforço de restrições consistentes ao longo de outras direções além da linha (epipolar) horizontal. De um modo geral, os efeitos dessas adições são fornecer uma maior robustez para iluminação e outras variações entre as imagens da esquerda e da direita, e ajudar a eliminar erros impondo restrições geométricas mais fortes através da imagem. O ponto principal do algoritmo é como atribuir um custo para cada pixel para todos as possíveis disparidades. Essencialmente, isso é análogo ao que é feito no *block matching*, mas há alguns novos passos. O primeiro novo passo é que é usado as métricas de Birchfield-Tomasi para comparar pixels, em vez de usar soma das diferenças absolutas. O segundo novo passo é que é usado uma importante suposição para a continuidade de disparidade (pixels vizinhos provavelmente tem a mesma ou disparidade similar) e ao mesmo tempo é usado um bloco de tamanho menor. Inclusive, no BM, janelas grandes tendem a ser um problema perto de discontinuidades (borda de algum elemento da imagem).

2.2 Requisito 2

2.3 Requisito 4

3 Resultados

Nesta seção são apresentados em forma de figuras e tabelas os resultados da aplicação para cada um dos requisitos.

3.1 Requisito 1

3.2 Requisito 2

3.3 Requisitos 3 e 4

4 Discussões e Conclusões

Para distorções radiais, a distorção é 0 no centro óptico do aparelho e cresce assim que movemos para a periferia. Na prática, esta distorção é pequena e pode ser caracterizada por

poucos termos de uma expansão da série de Taylor em torno de $r = 0$. Isso explica porque as medidas do objeto tiradas próximo ao centro da imagem se aproximam mais da medida real l , onde $l = 2.5$ cm . Além disso, algumas medidas da largura poderiam ser mais precisas se uma linha exatamente reta pudesse ser desenhada. Por exemplo, em $I_{raw,centre}$ para d_{min} a coordenada y , que corresponde a altura, não deveria ser alterada. Entretanto, podemos observar que as medidas estimadas foram bastante próximas da medida (largura l) real. Os resultados foram satisfatórios.

References

[1] Stan Birchfield and Carlo Tomasi. Depth discontinuities by pixel-to-pixel stereo. *International Journal of Computer Vision*, 35(3):269–293, 1999.

[2] R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, ISBN 0521540518, 2003.

[3] Heiko Hirschmuller. Stereo processing by semiglobal matching and mutual information. *IEEE Transactions on pattern analysis and machine intelligence*, 30(2):328–341, 2008.

[4] Itseez. Open source computer vision library. <https://github.com/itseez/opencv>, 2018.

[5] Adrian Kaehler and Gary Bradski. *Learning OpenCV 3: Computer Vision in C++ with the OpenCV Library*. O’Reilly Media, Inc., ISBN 978-1-4919-3800-3, 2016.

[6] Konolige Kurt. Small vision system: Hardware and implementation. In *The title of the book*, pages 111–116. Proceedings of the International Symposium on Robotics Research, Japan, 1997.