

Projeto Demonstrativo 3 - Múltiplas vistas (02/05/2018)

Tiago de Carvalho Gallo Pereira (14/0164138)
Universidade de Brasília - UnB
Princípios da visão computacional - Turma A
tiagodecarvalhopereira@gmail.com

Abstract

Projeto demonstrativo para aprender a utilizar aplicações com múltiplas câmeras e com isso conseguir adquirir informações sobre as coordenadas do mundo real. Nesse projeto em específico vamos restringir o desenvolvimento para o uso de apenas 2 câmeras, simulando uma visão estéreo.

1. Introdução

A partir de uma imagem, nós humanos conseguimos inferir se um objeto está mais próximo ou mais longe da câmera. No entanto nós só conseguimos fazer isso porque conhecemos ou temos ideia do tamanho real de determinados objetos em uma imagem. Por exemplo, se eu vejo uma foto onde aparecem uma pessoa e um prédio e ambos tem o mesmo tamanho em pixels eu consigo concluir que o prédio está muito mais longe da câmera do que a pessoa, pois sabemos que um prédio é muito maior que uma pessoa, logo eles não podem estar lado a lado e terem o mesmo tamanho na imagem.

Contudo, mesmo nós humanos tendo essa habilidade de inferir a profundidade a partir de imagens bidimensionais, ainda há 2 problemas. O primeiro deles é que consigo inferir tamanhos e profundidades relativas, mas não valores absolutos do tipo "aquela pessoa da imagem mede 1,80m", portanto eu só tenho ideia da informação do mundo real, mas não tenho nada muito concreto. O segundo problema é que nós conseguimos fazer esse tipo de estimativa, pois temos muito conhecimento prévio sobre as imagens, mas o computador não tem esse conhecimento prévio e por isso não consegue fazer uma estimativa igual a nós.

Com isso, entramos no desafio que é como fazer que o computador consiga ter essa noção de tamanhos e profundidade igual nós fazemos naturalmente. Para resolver esse problema, nós podemos utilizar sistemas com múltiplas câ-

meras observando a mesma cena, onde nós sabemos a posição de uma câmera em relação a outra, com esses dados e um embasamento matemático que será explicado na metodologia conseguimos fazer essa estimativa com o computador. Para esse projeto, iremos utilizar câmeras estéreo para simplificar um pouco o problema e porque é a geometria de câmeras que mais se assimila com a visão humana.

2. Metodologia

2.1. Requisito 1

O primeiro passo para gerar uma imagem de profundidade a partir de duas imagens estéreo, é calcular o mapa de disparidade entre as duas imagens. Para isso, nós retificamos as duas imagens em relação ao eixo y de forma que as linhas epipolares de ambas as imagens fiquem alinhadas. Após alinhar as linhas epipolares, sabemos que um ponto que se encontra na linha y de uma imagem também estará na linha y da outra imagem, portanto podemos fazer a busca desse pixel apenas em uma linha da outra imagem e não nela inteira. Para fazer essa busca, utilizamos o método de SAD (*sum of absolute differences*) como indicado em 4.

O mapa de disparidade vai indicar a diferença da posição de um certo pixel entre as imagens da esquerda e da direita, por exemplo, digamos que um pixel é encontrado na imagem da esquerda no ponto $X_l = 2$ e na imagem da esquerda em um ponto $X_r = 1$, logo para esse ponto o mapa de disparidade vai indicar que a disparidade é $d = X_l - X_r = 1$. Portanto, podemos concluir que o mapa de disparidade vai ser uma matriz com as mesmas dimensões das imagens analisadas. Dois parâmetros que mudam completamente o resultado do mapa de disparidade são, o tamanho da janela W que é utilizada pelo SAD e o valor máximo de disparidade que você pode assumir (a maior distância entre os pixels das duas imagens), empiricamente foi visto que o melhor resultado obtido foi utilizando $W = 13 \times 13$ e $\text{maxDisparidade} = 128$ pixels.

O esforço computacional para calcular o mapa de disparidade é grande e isso traz uma dificuldade de implementação otimizada desse método, mesmo utilizando programa-

ção em várias threads, o programa executava na ordem de grandeza de minutos, logo foi utilizada a classe *StereoBM* do *OpenCV*, pois ela é bastante otimizada para a aplicação desse método, retornando resultados na ordem de grandeza de segundos.

Após obter os mapas de disparidade, precisamos das informações da posição de uma câmera em relação a outra, para podermos calcular o *baseline* (distância entre o ponto focal das duas câmeras) e a distância focal da câmera. Para esse primeiro requisito, utilizando as imagens fornecidas pelo professor, o trabalho foi bem menor pois as linhas epipolares já estavam alinhadas e foi fornecido *baseline* = 120 e distância focal = 25. Com esses dados, podemos utilizar as equações 1, 2 e 3 conforme estudadas em 2, para adquirir informações sobre as coordenadas do mundo real.

$$X = \frac{b(x_l + x_r)}{2(x_l - x_r)} \quad (1)$$

$$Y = \frac{b(y_l + y_r)}{2(x_l - x_r)} \quad (2)$$

$$Z = \frac{bf}{(x_l - x_r)} \quad (3)$$

Após utilizar as equações 1, 2 e 3 para calcular as coordenadas do mundo real, podemos pegar as informações das coordenadas Z para calcular um mapa de profundidade das imagens estéreo. Para isso, normalizamos as coordenadas Z de 1 a 255, deixando o valor 0 para aqueles pixels que não puderam ser calculados a disparidade (não foram encontrados em ambas imagens), após isso colocamos essas informações do Z em uma imagem e mostramos ela na tela.

2.2. Requisito 2

Para obter imagens estéreo novas foram utilizadas duas *Webcams* criando a geometria de uma câmera estéreo, conforme podemos ver na figura 1. Utilizando essas duas câmeras foi criado um script em python que abre as duas câmeras e captura imagens delas ao mesmo tempo, com isso podemos garantir que a cena observada pelas câmeras não mudou da imagem da esquerda para a da direita.

Após obter as imagens próprias, foi necessário calibrar os parâmetros intrínsecos das câmeras para retirar distorção radial e poder calcular a distância focal da câmeras, como a matriz de parâmetros intrínsecos da câmera é dada pela equação 4 e temos duas câmeras diferentes, foi utilizada uma abordagem de retirar a média aritmética da distância focal x e y das duas câmeras para estimar a distância focal da geometria como um todo, conforme a equação 5. Para calibrar ambas as câmeras foram utilizados os mesmos métodos do projeto demonstrativo 2.

$$K = \begin{pmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{pmatrix} \quad (4)$$

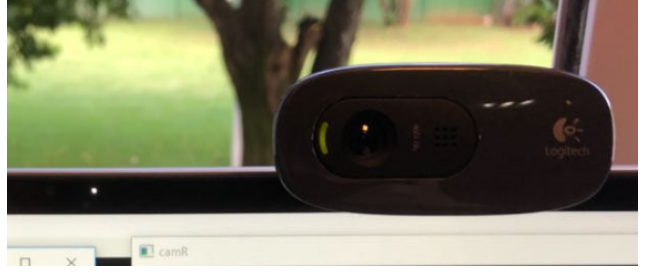


Figura 1. Posicionamento das câmeras para gerar a visão estéreo

$$f = \frac{f_{lx} + f_{ly} + f_{rx} + f_{ry}}{4} \quad (5)$$

Após obter a calibração dos parâmetros intrínsecos das câmeras, foi utilizado o método *StereoCalibrate* do *OpenCV* para fazer a calibração dos extrínsecos, pois esse método nos retorna também as matrizes de rotação e translação entre uma câmera e outra. Para obter o *baseline* dessa geometria, basta retirarmos a norma da matriz de translação entre as câmeras.

Agora que temos todos os parâmetros intrínsecos e extrínsecos das duas câmeras, a *baseline* e a distância focal, só nos falta retificar as imagens para podermos aplicar o requisito 1 nelas e obter os mapas de profundidade. Para retificar as imagens, foi utilizado o método *stereoRectify* do *OpenCV* que nos retorna o novo mapeamento dos pixels para podermos retificar as imagens.

Após calcularmos a *baseline*, a distância focal e a retificação das imagens estéreo, só precisamos aplicar o requisito 1 nas nossas imagens estéreo e temos o mapa de profundidade delas.

2.3. Requisito 3

Com todo o trabalho realizado nos requisitos 1 e 2, o requisito 3 fica extremamente simples. Para ele basta rodarmos o requisito 2 nas nossas imagens e gerar o mapeamento das coordenadas do mundo utilizando as equações 1, 2 e 3, quando temos a informação das coordenadas do mundo para dois pontos da imagem (escolhidos por cliques do mouse), basta utilizarmos um cálculo de distância Euclidiana conforme a equação 6 para obter o tamanho do objeto no mundo.

$$dist = \sqrt{(X_1 - X_2)^2 + (Y_1 - Y_2)^2 + (Z_1 - Z_2)^2} \quad (6)$$

3. Materiais e métodos

Para a realização desse experimento foi utilizado um ambiente instalado com Python 3.5.2, OpenCV 3.2.0 e NumPy 1.14.2. Foi criado um único arquivo com o código fonte em python, esse arquivo recebe parâmetros de entrada que determinam qual requisito será executado e sob quais condições.



Figura 2. Imagens de disparidade e profundidade para o exemplo do bebê. Profundidade a esquerda e disparidade a direita

Os materiais externos usados para esse projeto demonstrativo foram, uma régua milimetrada com precisão de 0.05mm, o padrão de calibração (tabuleiro de xadrez com quadrados 29x29 mm) colado em um livro para dar suporte, um celular de 14.3cm (tamanho medido do conjunto celular + capinha) para testar o funcionamento do programa e duas webcams: uma Logitech c270 e a webcam *built-in* do laptop (as duas câmeras foram posicionadas conforme a figura 1 para simular uma visão estéreo).

4. Resultados

4.1. Requisito 1

Utilizando as imagens da planta e do bebê fornecidas pelo professor, foi possível adquirir imagens de disparidade e de profundidade. Ambas as imagens podem ser verificadas nas figuras 2 e 3

Ambas as imagens foram geradas utilizando como parâmetro $W = 13 \times 13$ para o tamanho da janela de busca do SAD e $\text{maxDisparidade} = 128$ pixels, esses dois parâmetros são para o cálculo da disparidade e mudam bastante o resultado.

Nas imagens de profundidade, pontos mais escuros são mais próximos que os mais claros. Para ambas as imagens, a profundidade máxima calculada foi de 893.2cm e a profundidade mínima foi de 1.6cm, isso provavelmente é por causa de algum erro na hora do cálculo dos limites da matriz normalizada.

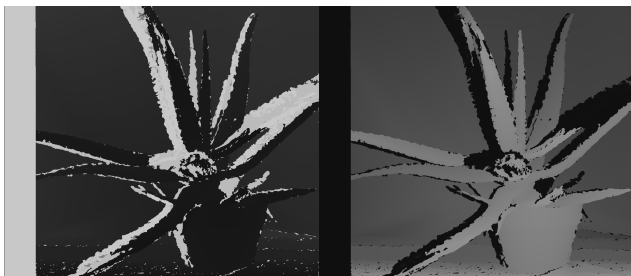


Figura 3. Imagens de disparidade e profundidade para o exemplo do aloe. Profundidade a esquerda e disparidade a direita

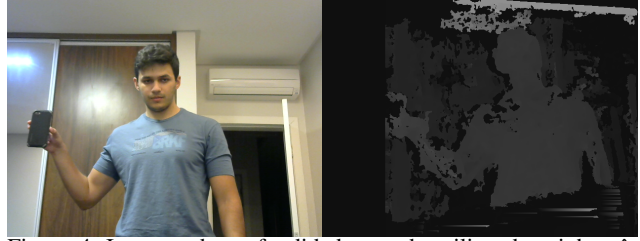


Figura 4. Imagem de profundidade gerada utilizando minha câmera estéreo. Vista esquerda da cena a esquerda e profundidade a direita

4.2. Requisito 2

Para as minhas próprias imagens e calibrando as câmeras, foram obtidos resultados razoáveis. Pela calibração das câmeras, obtemos que para a geometria montada da câmera estéreo o $\text{baseline} = 75.05$ e a distância focal = 64.76.

A melhor imagem de profundidade obtida pode ser vista na figura 4 acompanhada da vista esquerda dessa mesma imagem. Para essa imagem e para todas as outras usadas por mim foram obtidos em média que a profundidade máxima é de 1.04cm e a profundidade máxima é de 243.5cm.

Considerando que a parede atrás de mim estava a uma distância de 260cm aproximadamente e que a parede, por ser lisa, é um ponto onde há uma grande dificuldade no cálculo do mapa de disparidade, o erro de menos de 10% é bastante razoável.

4.3. Requisito 3

Para poder analisar bem os resultados foram utilizadas 5 imagens de teste e como o mapa de profundidade por mais que tenha ficado razoável não ficou extremamente preciso foi criada a tabela 1 para resumir os resultados gerais. O objeto a ser medido em todos os testes foi um celular que mede 14.3cm.

5. Discussão e Conclusões

No requisito 1, podemos ver que as imagens de disparidade ficaram boas, conforme esperado, no entanto as imagens de profundidade não ficaram tão boas para análise. Acredito que as imagens de profundidade ficaram bastante escuras e não muito boas por causa de pontos *outliers* que podem ter aparecido no mapa de disparidade e que não fo-

	Melhor Medida	Média	Distância (Z)
Img 1	49,38cm	53,12cm	30,3cm
Img 2	10,17cm	9,51cm	72,5cm
Img 3	12,78cm	11,43cm	82cm
Img 4	9,96cm	9,37cm	91cm
Img 5	13,60cm	13,30cm	127cm

Tabela 1. Tabela de tamanhos estimados nas coordenadas do mundo.

ram tratados. Ao normalizar dados que apresentam *outliers* (pontos fora da curva) a tendência é que os pontos centrais fiquem muito próximos e quando mostramos isso em uma imagem fica difícil separar as cores. Esses *outliers* indesejados também podem ter sido responsáveis pelos valores estranhos de profundidade mínima e máxima.

Como nesse projeto foram utilizadas duas câmeras diferentes, isso aumentou a facilidade porque poupa o usuário de ficar movendo a mesma câmera para tirar fotos de duas vistas da mesma cena e minimiza os erros que essa movimentação pode causar. Contudo, como as duas câmeras utilizadas eram de modelos diferentes, isso gerou bastante dificuldade em alguns passos, um deles aconteceu porque uma câmera tem uma abertura muito menor do que a outra, logo há muitos pontos que não podem ser mapeados de uma imagem na outra, e essa diferença de abertura faz com que a qualidade da imagem de uma das câmeras seja bem pior que da outra (uma vez que as duas tem a mesma resolução). O principal ponto onde ter duas câmeras diferentes mais atrapalhou foi no cálculo da distância focal, pois os valores da distância focal das câmeras variavam em 25% e a abordagem de utilizar a média da distância focal entre as duas câmeras pode não ter sido a melhor abordagem, no entanto não houve tempo para testar outras abordagens.

Podemos ver na tabela 1 que quando o objeto estava muito próximo da câmera, os resultados pioravam muito e conforme os objetos ficavam mais distantes o resultado ia melhorando, isso provavelmente ocorre porque as imagens utilizadas para calibração das câmeras foram feitas a uma distância média das câmeras, logo o sistema não estava bem calibrado para imagens muito próximas. Outro ponto que poderia trazer um ganho seria utilizar mais de um padrão de calibração para calibrar a câmera estéreo e posicionar eles de forma que pontos não ficassem coplanares para que a coordenada Z do mundo real influenciasse mais nos parâmetros de calibração e não fosse algo novo introduzido apenas no requisito 3.

Nesse projeto demonstrativo foi utilizada apenas a abordagem que utiliza calibração de câmeras para trabalhar com câmeras estéreo, a outra abordagem existente para esse problema é utilizar geometria epipolar e calcular a matriz fundamental F. Com a abordagem de calibração de câmeras ficou bem claro que esse é um problema que não funciona muito bem para precisão e/ou ambientes não controlados, no entanto na teoria de múltiplas vistas o melhor jeito de melhorar os resultados é adicionando redundâncias aos problemas, logo se tivéssemos inserido mais informações sobre a cena ou mais câmeras, talvez o resultado fosse melhor.

Referências

- [1] *Hartley and Zisserman's Tutorial on Multiple View Geometry, Part 1, CVPR 1999. Disponibilizado no moodle*
- [2] Slides do professor, aula 06 slide 14 .
- [3] OpenCV. *Camera Calibration*. Tutorial link. Acessado em 13 de abril de 2018.
- [4] Wikipedia. *Sum of absolute differences*. Link. Acessado em 01 de maio de 2018.