

Projeto Demonstrativo 03 - Múltiplas Vistas

Realizado em abril de 2018

Hugo Luís Andrade Silva
Departamento de Ciência da Computação
Universidade de Brasília
 Brasília, Brasil
 hugosilva664@aluno.unb.br

Abstract—Projeto visando familiarização com câmeras estéreo. Inicialmente, são fornecidas imagens já retificadas relativas à câmera esquerda e à câmera direita para que sejam encontrados pontos referentes ao mesmo objeto nas duas câmeras e gerados mapas de profundidade a partir disso. Em seguida, o processo é repetido, mas agora as imagens não são mais retificadas. Por fim, é feita uma aplicação de régua virtual, que permite a medição de comprimentos de objetos a partir de cliques nas imagens da tela. Para imagens capturadas com uma mesma câmera e distâncias próximas, os resultados foram razoáveis, sendo que a qualidade foi caindo com aumento da distância e foram muito piores quando as duas câmeras usadas foram diferentes.

Index Terms—opencv, computervision, imageprocessing

I. INTRODUÇÃO

Em visão computacional, comumente são feitas estimativas da profundidade de objetos a partir de imagens obtidas, sendo que, muitas vezes, essas imagens são obtidas apenas por meio de câmeras. Isso é necessário, por exemplo, para que a navegação de um robô funcione adequadamente. Também é usado para auxiliar reconhecimento de objetos através da identificação de oclusão e em técnicas de geração de imagens 3D em filmes. A ideia por trás de visão estéreo se baseia no fato de que objetos próximos possuem maior deslocamento relativo entre as câmeras, enquanto objetos mais distantes não terão tanta variação de posição. A ilustração na figura 1 é muito utilizada para representar esse tipo de situação com duas câmeras, em que os pontos c e c' correspondem aos centros das câmeras e os planos à frente dos centros são os planos onde a imagem é projetada. Nessa ilustração, o ponto X corresponde ao ponto 3D, x é sua representação na imagem esquerda e x' é sua representação na imagem à direita. Os pontos e e e' são os epipólos, sendo e a projeção do epipólo direito na imagem esquerda e e' a projeção do epipólo esquerdo na imagem da direita.

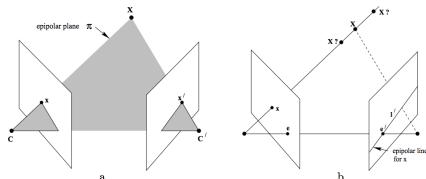


Fig. 1: Exemplo de representação de visão estéreo com geometria epipolar

Cada ponto 3D define um plano epipolar, que é o plano que contém os epipólos e o ponto 3D. As interseções do plano epipolar com os planos da imagem dão-se o nome de linhas epipolares. Vale ressaltar que, dados apenas os epipólos e o ponto x , é possível definir a linha l' , representada à direita na figura 1. Essa linha corresponde à projeção do raio que liga X e x na câmera à direita e, por isso, o ponto x' encontra-se em l' . Isso significa que, caso se deseje encontrar o ponto x' , de posse do ponto x , basta buscar nessa linha.

Conforme explicado em [2], para esse problema, se define a matriz fundamental F , que satisfaz:

$$x'^\top F x = 0 \quad (1)$$

E l' pode ser obtido com:

$$l' = Fx \quad (2)$$

Sendo que, nessas equações, x e x' são expressos em coordenadas homogêneas, F é 3×3 e $l' = [a, b, c]^\top$ com $aP_x + bP_y + c = 0$ para quaisquer P_x e P_y . Ainda em [2], é mostrado que, caso seja definido o centro do mundo em uma das câmeras, a matriz fundamental pode ser expressa como:

$$F = K'^{-1} R K^\top [K R^\top t]_\times \quad (3)$$

Com K e K' sendo as matrizes de intrínsecos, R e t sendo a rotação e translação entre os centros das câmeras e a notação $[a]_\times$, para um vetor $a = [a_1, a_2, a_3]$ correspondente à matriz:

$$[a]_\times = \begin{bmatrix} 0 & -a_3 & a_2 \\ a_3 & 0 & -a_1 \\ -a_2 & a_1 & 0 \end{bmatrix} \quad (4)$$

Isso significa que, com as câmeras calibradas e com a rotação e translação entre elas conhecidas, é possível determinar as linhas epipolares. Após isso, é aplicado um warping nas imagens, de forma que as linhas epipolares em ambas as imagens se tornem horizontais. A esse processo, dá-se o nome de retificação. Por fim, com os pontos $x = (x_L, y_L)$ e $x' = (x_R, y_R)$, é possível obter as coordenadas de $X = (X_c, Y_c, Z_c)$ através das equações:

$$X_c = \frac{b(x_L + x_R)}{2(x_L - x_R)} \quad (5)$$

$$Y_c = \frac{b(y_L + y_R)}{2(x_L - x_R)} \quad (6)$$

$$Z_c = \frac{bf}{(x_L - x_R)} \quad (7)$$

Com b sendo a baseline (distância entre os centros) e f sendo a distância focal.

II. MATERIAIS E METODOLOGIA

A. Requisito 1

Na primeira parte do trabalho, foram fornecidas imagens já retificadas e o objetivo era encontrar as correspondências entre os pontos nas imagens e, a partir disso, gerar mapas de profundidade e disparidade (medida como $X_L - X_R$). O método consiste em, para cada pixel, pegar uma vizinhança $L \times L$ na imagem esquerda e procurar qual a vizinhança na imagem da direita mais próxima dela. Como o procedimento é lento, foram tomadas medidas para torná-lo mais rápido e poder rodar o algoritmo várias vezes:

- A verificação por vizinhança é realizada apenas na mesma linha
- Assume-se que há um máximo de disparidade a ser verificado, por isso, para cada pixel a ser verificado na imagem à direita, a busca não é feita na linha inteira. Em vez disso, é tomado um intervalo com I colunas à direita e à esquerda do pixel verificado, com I sendo 25% do total de colunas da imagem à direita

Além disso, em alguns casos haverá muitas correspondências para um mesmo pixel à esquerda, por isso foram realizadas as seguintes verificações para eliminar esses pontos das detecções:

- São eliminados pixels correspondentes em que o pixel na imagem direita fique mais à direita do que na imagem esquerda. Nesse caso, o pixel à direita detectado é descartado e é procurado o próximo mínimo da função de diferença
- Na parte da linha verificada, não podem haver pixels com distância em colunas maior que L (tamanho da janela) do pixel detectado como mais semelhante e que também sejam semelhantes ao pixel da imagem à esquerda. Na implementação, essa semelhança foi medida como até $1.2 \times valor_minimo$.

Além de tudo isso, o sistema foi feito usando multi-processing com 8 processos sendo usados para executar o algoritmo. Esse número corresponde à quantidade de logical cores do computador utilizado e garante um uso de 100% da CPU. Depois de detectados os pontos, as equações da seção anterior são usadas para convertê-los para 3D e gerar os mapas de profundidade e disparidade.

B. Requisito 2

Esse requisito consiste em realizar a retificação das imagens e, em seguida, realizar o matching de pontos como no requisito anterior. A partir do fornecimento das matrizes K e K' , dos parâmetros de distorção das duas câmeras e da rotação e translação entre seus centros, a ferramenta OpenCV gera mapas que são utilizados para retificar as imagens. Para obter os parâmetros intrínsecos e de distorção, foi utilizado o código do projeto 2. Para obter R e T , foi utilizada a função **stereoCalibrate**, que tem como entrada os parâmetros intrínsecos e as detecções em um padrão de calibração, que, no caso, era um tabuleiro de xadrez. A retificação e geração de mapas foram testadas em 4 settings:

- **Setting A:** a mesma câmera (webcam) com majoritariamente translação entre as fotos
- **Setting B:** a mesma câmera (webcam) com translação e rotação entre as fotos
- **Setting C:** 2 câmeras diferentes (webcam e celular) posicionadas perto uma da outra
- **Setting D:** 2 câmeras diferentes (webcam e celular) posicionadas longe uma da outra

C. Requisito 3

O último requisito pede para que seja feita uma régua virtual. Para isso, foram feitos dois códigos: um para a imagem da câmera à esquerda e outro para a camera à direita. É mostrada a imagem e o usuário clica em dois pontos. Em seguida, os dois pontos detectados pelo matching mais próximos dos pontos clicados são destacados na tela através de uma linha traçada entre eles, permitindo que o usuário veja os pontos cuja distância está de fato sendo medida. Os pontos são convertidos para 3D como no requisito 1 e é calculada a distância euclidiana entre eles.

D. Resultados

E. Requisito 1

Foram fornecidos $f = 25\text{pixels}$ e $baseline = 12\text{cm}$. Com base nesses dados, com uma janela 15×15 , foi gerada inicialmente a figura 2, em que um a cada 1000 pontos detectados nas duas imagens foram destacados. Em seguida, essas detecções foram usadas para gerar mapas de disparidade e profundidade, como na figura 3, em que os valores foram normalizados. Para melhor visualização da profundidade, foi aplicada a função $\log(x)$ antes da normalização. Apesar disso, as imagens da disparidade são mais ricas em detalhes e, por isso, serão usadas nas próximas ilustrações. Por fim, a figura 4 ilustra esses mapas para diferentes tamanhos de janela. A profundidade mais próxima retornada foi de 1cm e a mais distante de 300cm com esses parâmetros hipotéticos.

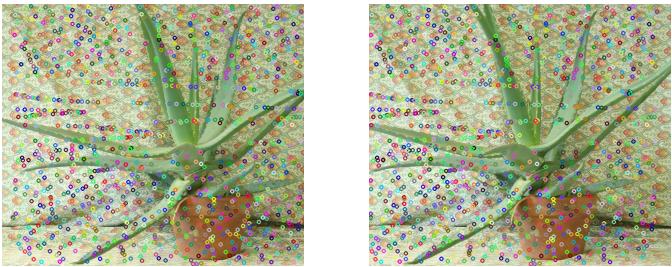


Fig. 2: Alguns pontos detectados em ambas as imagens, pontos correspondentes têm círculos com cores iguais

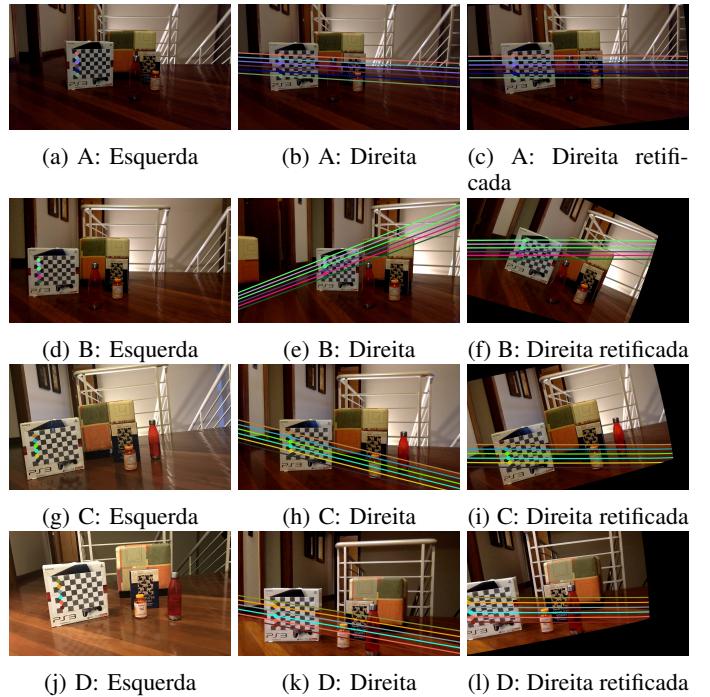
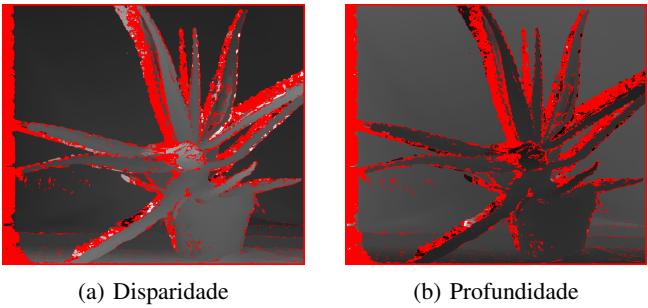


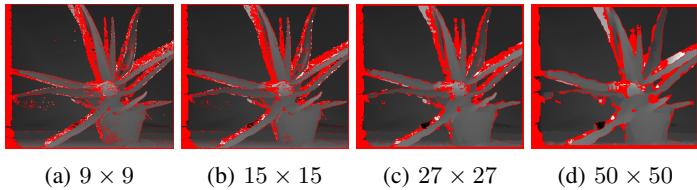
Fig. 5: Alguns pontos detectados, linhas epipolares correspondentes a esses pontos e imagens retificadas



(a) Disparidade

(b) Profundidade

Fig. 3: Mapa de disparidade (mais claros representam objetos mais próximos) e de profundidade (mais escuros são mais próximos)



(a) 9×9

(b) 15×15

(c) 27×27

(d) 50×50

Fig. 4: Disparidade para vários tamanhos de janela

F. Requisito 2

A figura 5 mostra os settings antes e depois da retificação, com algumas das linhas epipolares e pontos correspondentes a essas linhas destacados, a figura 6 mostra as retificações do setting C lado-a-lado e a figura 7 mostra os mapas de disparidade dos quatro settings testados. Para os settings D, a zona de busca foi expandida. A janela usada aqui foi de 15×15 e, em geral, as regiões mais claras corresponderam a 40cm .

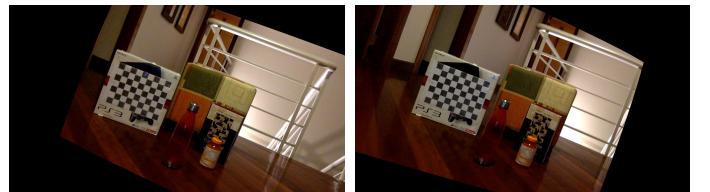


Fig. 6: Setting C, imagens lado a lado, note que os objetos ficaram à direita na imagem esquerda e à esquerda na imagem direita, como deveria ser para planos alinhados

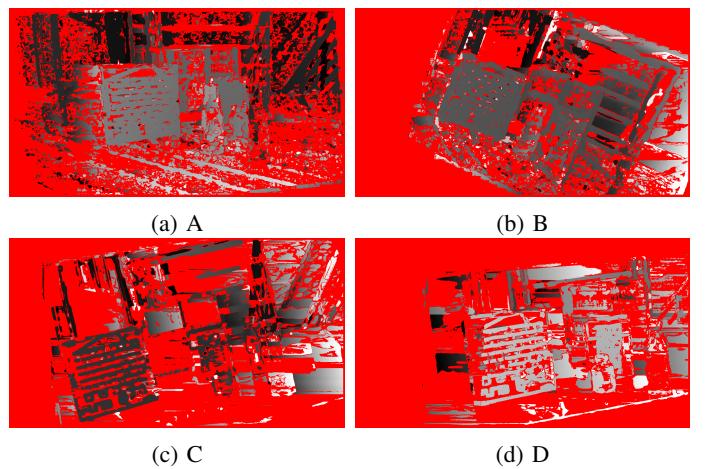


Fig. 7: Disparidade para todos os settings

G. Requisito 3

Dado que o setting A era o de mais fácil solução, a partir dele foram gerados os sets A_2 e A_3 variando a posição de um único objeto, como representado em 8. A figura 9 ilustra a vista de cima da disposição principal utilizada.

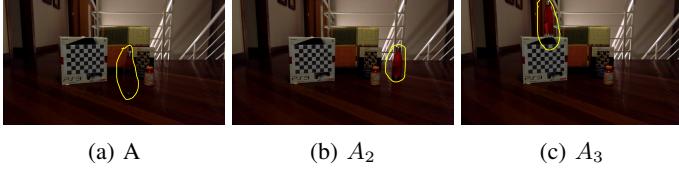


Fig. 8: Settings adicionais



Fig. 9: Vista de cima da cena

A tabela representa as medições feitas a partir de cliques em imagens à esquerda e a tabela os cliques à direita. Para cada objeto, deve-se atentar para que o clique seja feito longe das bordas de forma que o ponto detectado mais próximo esteja de fato no plano do objeto, e o clique deve ser feito perto das regiões mais distintas possíveis, para facilitar o matching. Para cada caso, os cliques foram repetidos algumas vezes e algum resultado próximo ao que mais aparecia foi registrado. Por fim, a tabela III representa as medianas dos valores encontrados.

TABLE I: Tabela com os dados da régua visual esquerda

	GT	A	A_2	A_3	B	C	D
Garrafa	25	26.59	27.69	25.64	432.25	26.88	23.08
Ω_3	13	13.88	12.68	15.80	24.38	16.71	11.90
Livro	18	32	20.08	19.42	21.44	21.21	18.25
Caixa	33	43.78	41.48	41.70	36.31	36.54	33.53
Cubo	42	48.36	53.09	48.96	28.53	49.73	36.55
Corrimão	115	141.38	176.20	N	765.13	166.87	152.64

TABLE II: Tabela com os dados da régua visual direita

	GT	A	A_2	A_3	B	C	D
Garrafa	25	27.32	28.56	26.54	53.45	28.01	22.00
Ω_3	13	14.62	18.13	15.26	17.78	157.20	11.91
Livro	18	27.68	20.22	18.25	22.83	15.15	15.03
Caixa	33	44.27	43.38	42.70	37.97	35.99	33.79
Cubo	42	57.95	51.47	68.39	46.32	149.70	54.96
Corrimão	110	146.51	165.23	166.32	163.82	183.59	123.73

TABLE III: Mediana dos valores encontrados

	GT	Mediana
Garrafa	25	27.1
Ω_3	13	15.53
Livro	18	20.15
Caixa	33	39.725
Cubo	42	50.6
Corrimão	110	165.23

III. RESULTADOS

O principal problema no algoritmo de matching foi o tempo necessário para que termine de rodar, mesmo com todas as alterações tendo em vista deixá-lo mais rápido, foi evidente que não poderia ter sido feito em tempo real. Um método que se baseasse no uso de menos pontos e interpolação de resultados poderia ser mais eficiente, ou uma representação da cena em algum espaço diferente. O uso de janelas maiores, por um lado, tem o efeito de dificultar as detecções, pois mais pixels semelhantes dentro de uma mesma vizinhança são necessários. Por outro lado, a remoção de detecções com muitos mínimos parecidos faz com que janelas pequenas também tenham o matching dificultado. A análise da imagem mostra que há uma perda de detalhes nos contornos com o aumento da janela.

No requisito seguinte, ocorreu uma possível limitação da qualidade da retificação devido ao fato de os pontos detectados serem coplanares e próximos. Uma abordagem baseada em SIFT poderia ter melhorado o trabalho nesse ponto. O problema de tal abordagem seria a obtenção da baseline, mas um padrão de calibração poderia ter sido usado especificamente para isso. Além disso, a diferença de qualidade das câmeras e distâncias focais prejudicou o matching nos settings C e D. Caso fosse usada apenas a câmera do celular, com sua resolução superior, o matching obtido poderia ter sido melhor.

Por fim, no último requisito, nos settings C e D houve uma grande diferença entre as medições da direita e da esquerda, o que significa que os pontos da esquerda podem ter sido correspondidos pelo algoritmo com pontos diferentes dos corretos nesses casos. Uma régua em que fossem mostradas as janelas direita e esquerda ao mesmo tempo e um clique em qualquer uma das imagens destacasse o ponto das duas janelas ajudaria a verificar isso. Algumas das medições funcionaram melhor do que outras para alguns objetos específicos, provavelmente por causa de oclusão. De qualquer forma, a combinação dos resultados na tabela III permite corrigir falhas individuais de settings específicos através do resultado geral e apresentou resultados muito bons, tendo em vista que os objetos estavam longe. O maior erro foi do corrimão, mas isso já era esperado, visto que era o objeto mais distante.

REFERENCES

- [1] OpenCV dev team. OpenCV documentation. <https://docs.opencv.org/>. Accessed: 2018-05-01.
- [2] Richard Hartley and Andrew Zisserman. *Multiple view geometry in computer vision*. Cambridge university press, 2003.
- [3] G Strang. Linear algebra and its applications, 251, 1988.
- [4] Colorado School of Mines William Hoff. Stereo vision lecture. Accessed: 2018-04-27.