

Projeto Demonstrativo 02 - Calibração de Câmeras

Aloisio Dourado Neto - Matrícula: 18/0040511

Disciplina: Visão Computacional - Professor: Teófilo de Campos

Brasília, 18 de março de 2018

aloisio.dourado.bh@gmail.com

Resumo—Para estimar a projeção de pontos do mundo real em imagens captadas por uma câmera é fundamental conhecermos os parâmetros internos da câmera utilizada. O processo de estimação de tais parâmetros é conhecido como calibração. O objetivo desse projeto é avaliar aspectos relacionados com a calibração de câmeras digitais. Como exemplo de aplicação de câmeras calibradas será desenvolvida uma aplicação (régua visual) que permite estimar dimensões de objetos a partir da imagem captada por uma câmera calibrada.

Index Terms—Calibração de câmeras, computer vision, OpenCV

I. INTRODUÇÃO

A estimação da posição de objetos do mundo 3D a partir de imagens captadas por câmeras é uma das aplicações mais comuns em visão computacional [1] [2]. Entretanto, para que seja possível projetar pontos do mundo 3D na imagem 2D gerada, é necessário estimar os parâmetros internos da câmera bem como a posição da câmera em relação ao sistema de coordenadas utilizado para representar o espaço 3D. Adicionalmente, considerando que as lentes utilizadas nas câmeras (especialmente naquelas de baixo custo) apresentam distorção, é importante a correção desse efeito indesejado para a realização de estimativas mais confiáveis [3].

O objetivo geral do presente trabalho é avaliar os aspectos relacionados com a calibração de câmeras digitais. Para o atingimento desse objetivo geral foram definidos os seguintes requisitos:

- **Requisito 1 - medição de um segmento de reta em pixels:** Desenvolver uma aplicação em OpenCV que capture, por meio de cliques de *mouse*, dois pontos da imagem e trace um segmento de reta na tela ligando os dois pontos.
- **Requisito 2 - calibração dos parâmetros intrínsecos da camera:** Realizar a calibração de uma câmera digital com o auxílio de um padrão de calibração impresso, gerando arquivos "XML" com os parâmetros intrínsecos e distorções provenientes do processo de calibração.
- **Requisito 3 - calibração dos parâmetros extrínsecos:** Aproveitando os parâmetros intrínsecos estimados no requisito 2, estimar os parâmetros extrínsecos da câmera, também com o apoio de um padrão de calibração impresso.
- **Requisito 4 - implementação de uma régua visual:** Terminado o processo de calibração acima e usando os recursos de medição de segmentos de reta em *pixels* desenvolvidos no requisito 1, desenvolver uma aplicação em

OpenCV que converta as coordenadas das extremidades do segmento de reta para coordenadas do espaço 3D e estime a distância entre estes dois pontos no sistema de coordenadas do mundo.

- **Requisito 5 - análise:** Discutir os principais elementos causadores das variações das medições obtidas em relação à medida real do objeto medido, avaliando características do hardware da câmera e fatores podem ser os causadores destas possíveis diferenças e propondo fatores podem ser melhorados/aprimorados para melhorar precisão das estimativas.

O referencial teórico necessário para realização do presente trabalho consta no livro "Multiple View Geometry" [4], especialmente nos capítulos 6, 7 e 8 referentes, respectivamente a modelos de câmeras, estimação da matriz de projeção e calibração. Adicionalmente, foram utilizadas algumas simplificações para estimativas de pontos coplanares apresentadas na Tese de Doutorado do Prof. Teófilo de Campos [5]. A documentação das funções do OpenCV necessárias para o trabalho, encontra-se no livro "Learning OpenCV 3" [6]. Adicionalmente existe um tutorial sobre calibração de câmeras disponível no próprio site do Software [3].

II. METODOLOGIA

Nessa seção apresentaremos a teoria utilizada no presente trabalho, assim como o ambiente de desenvolvimento e os procedimentos adotados para a implementação dos requisitos.

A. Fundamentação Teórica

Uma câmera pode ser definida como um dispositivo que possibilita mapear pontos do mundo 3D em uma imagem 2D [4]. O mapeamento entre coordenadas do mundo e a imagem 2D projetada pode ser obtido pela equação 1, onde X é um vetor de coordenadas homogêneas $(X, Y, Z, 1)^T$ que representa as coordenadas do mundo e x é um vetor de coordenadas homogêneas da imagem $(x, y, 1)^T$ e P é uma matriz 3×4 que é conhecida como a matriz de projeção da câmera [4]. A matriz de projeção P pode ser decomposta conforme a equação 2, onde K é a matriz 3×3 que representa os parâmetros intrínsecos da câmera, R é a matriz 3×3 de rotação e t é o vetor de translação [4]. R e t compõem os requisitos extrínsecos da camera em relação ao mundo.

$$x = PX \quad (1)$$

$$P = K[R|t] \quad (2)$$

Em uma câmara CCD, uma matriz de parâmetros intrínsecos simplificada pode, sem perda significativa de precisão, ser representada com o formato descrito na equação 2 [4], onde α_x e α_y representam a distância focal da câmara em *pixels* nas direções x e y respectivamente. Já x_0 e y_0 representam a coordenada do ponto central da câmara, também em *pixels*.

$$K = \begin{bmatrix} \alpha_x & 0 & x_0 \\ 0 & \alpha_y & y_0 \\ 0 & 0 & 1 \end{bmatrix} \quad (3)$$

O presente projeto prevê a realização da calibração em dois momentos. Inicialmente deve-se estimar os parâmetros intrínsecos da câmara (matriz K) e, num segundo momento, quando da realização de medições no mundo real, apenas estimar os parâmetros extrínsecos (R e t). A obtenção dos parâmetros intrínsecos pode ser realizada diretamente por meio de funções de calibração presentes no OpenCV [6] [3] e, a seguir, apresentaremos a teoria relacionada à estimação dos parâmetros extrínsecos, dado que se conhece previamente a matriz K .

Com vistas a estimar os parâmetros de calibração da câmara representados na matriz de projeção P , considerando que se possua um conjunto de pontos do mundo real (X_i) para os quais se conheça os correspondentes pontos na imagem da câmara (x_i), a partir da equação 1, é possível montar um sistema de equações dado por $x_i = PX_i$.

No caso do presente trabalho, considerando que todos os pontos do objeto de calibração impresso são coplanares, é possível assumir que todos os pontos do objeto de calibração estejam na coordenada $Z = 0$, de modo que é possível utilizar a simplificação proposta em [5]. Assim sendo, a equação 1 pode ser simplificada para a equação $(x_i \ y_i \ 1)^T = P_{3 \times 3}(X_i \ Y_i \ 1)^T$, que, por sua vez, pode ser rescrita na forma $AP = 0$, onde A é dada pela equação 4 e $p = (p_{11}, p_{12}, p_{13}, p_{21}, p_{22}, p_{23}, p_{31}, p_{32}, p_{33})$ é um vetor de ordem 9 correspondente aos elementos da matriz de projeção.

$$A = \begin{bmatrix} X_i & Y_i & 1 & 0 & 0 & 0 & -X_i x_i & -Y_i x_i & -x_i \\ 0 & 0 & 0 & X_i & Y_i & 1 & -X_i y_i & -Y_i y_i & -y_i \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \end{bmatrix} \quad (4)$$

Convém notar a diferença de sinal na última coluna da matriz A apresentada aqui e a apresentada em [5]. Isso se deve à diferença em relação ao posicionamento do sistema de coordenadas da câmara utilizada nos dois trabalhos. Aqui estamos usando o sistema de coordenadas proposto em [4] que utiliza o eixo Z na direção oposta à da utilizada em [5].

O sistema de equações $AP = 0$ pode ser resolvido por meio de decomposição em valores singulares (SVD), como na expressão, onde um possível conjunto de valores para os elementos de P corresponde à última coluna da matriz de valores singulares à direita V . Os valores dos elementos de P podem então ser usados como valores iniciais em um processo

de otimização linear que vise minimizar a soma dos resíduos dados nas expressões 5 e 6.

$$\left(x_i - \frac{p_{11}X_i + p_{12}Y_i + p_{13}}{p_{31}X_i + p_{32}Y_i + p_{33}} \right)^2 \quad (5)$$

$$\left(y_i - \frac{p_{21}X_i + p_{22}Y_i + p_{23}}{p_{31}X_i + p_{32}Y_i + p_{33}} \right)^2 \quad (6)$$

O procedimento possibilita obter uma matriz de projeção P otimizada para os pontos do objeto de calibração planar utilizado nesse projeto. Considerando que no requisito 2 está prevista a obtenção acurada da matriz de parâmetros extrínsecos K , precisamos agora decompor P em $K[R|t]$ conforme a equação 2, dado que K é conhecido previamente. Para tanto, podemos continuar a aplicar a abordagem proposta em [5]. Seja a matriz H dada pela por $H = (h_1 \ h_2 \ h_3) = K^{-1}P$. O vetor de translação t , corresponde à coluna h_3 , enquanto uma estimativa inicial para a matriz de rotação R_{ini} pode ser obtida pelas equações 7, 8 e 9.

$$(r_1 r_2 t) = \frac{2}{\|h_1\| + \|h_2\|} (h_1 h_2 h_3) \quad (7)$$

$$r_3 = r_1 \times r_2 \quad (8)$$

$$R_{ini} = (r_1 \ r_2 \ r_3) \quad (9)$$

Uma vez que as colunas de R_{ini} podem não ser mutuamente ortogonais e com norma unitária, em razão de ruído na imagem ou problemas de discretização, faz-se necessário obter uma matriz corrigida R usando a decomposição de valores unitários como nas equações 10 e 11.

$$U S V^T \leftarrow SVD(R_{ini}) \quad (10)$$

$$R = U V^T \quad (11)$$

Considerando que desejamos estimar a posição 3D de pontos a partir de suas projeções na imagem, temos que realizar a operação reversa da realizada na 2. isso apenas é possível uma vez que os pontos para os quais desejamos estimar coordenadas são coplanares.

Os procedimentos descritos não levam em consideração os parâmetros de distorção da câmara. Entretanto, as rotinas de calibração do OpenCV permitem estimar tais parâmetros. [6] [3].

B. Materiais utilizados

O projeto foi desenvolvido em um *desktop* com processador Intel I7 com seis núcleos e 12 threads, 32 GB de memória RAM, dotado de 1 GPU Nvidia 1050Ti e 1 GPU Nvidia 1080Ti. O sistema operacional utilizado foi o Ubuntu 16.04 LTS, com Nvidia Cuda 8. O ambiente de desenvolvimento para Python foi montado utilizando o gerenciador de bibliotecas Anaconda 3, OpenCV 3.4 e Python 3.6.4.

Para realização dos experimentos foi preparada a bancada apresentada na imagem 1. Como padrão de calibração foi utilizado um padrão impresso do tipo tabuleiro de xadrez. Para assegurar que os pontos do padrão fiquem coplanares, o padrão

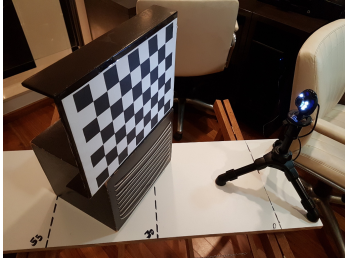


Figura 1. Bancada para os experimentos.

foi fixado em uma superfície rígida. A câmera utilizada é uma *webcam* simples da marca Multilaser, com ajuste de foco. Para os experimentos o foco foi mantido fixo, com regulagem otimizada para uma distância de 55cm da lente. A camera foi posicionada em um mini-tripé sobre uma bancada com marcações de distância horizontal a partir da posição da lente.

C. Implementação dos Requisitos

1) *Requisito 1*: Esse requisito visa explorar as funções de controle do mouse e desenho de retas do OpenCV. Ao ser executado o programa correspondente, é ativada a captura da imagem da *webcam* padrão do sistema e a imagem capturada é mostrada na janela. Usando a função `setMouseCallback()` do OpenCV, o programa permite que se clique em pontos da tela. A cada clique, é apresentando um círculo vermelho ao redor do ponto selecionado desenhado por meio da função `circle()`. Após o segundo clique, um segmento de reta é desenhado entre os pontos com a função `line()`. A distância euclidiana em pixels entre os dois pontos, calculada conforme a equação 12, é apresentada no terminal. O programa permite desenhar diversos segmentos de reta.

$$d_{euclid} = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2} \quad (12)$$

2) *Requisito 2*: Esse requisito visa realizar a calibração dos parâmetros intrínsecos da câmera. O programa possui dois estágios: a) calibração e b) desenho de segmentos de reta. No estágio inicial são realizadas 5 rodadas de calibração, cada uma com 10 *snapshots* do padrão de calibração capturados pela câmera. O padrão deve ser posicionado de forma que todos os 48 pontos estejam visíveis nas imagens.

Em cada rodada de calibração, um vetor de coordenadas é inicializado, com as coordenadas do mundo 3D correspondentes ao padrão de calibração. São tomados 10 snapshots e a identificação dos pontos no tabuleiro é realizada com a função `findChessboardCorners()` sobre a imagem em preto e branco. Em seguida, as coordenadas dos pontos de calibração são refinadas a nível de sub *píxel* usando a função `cornerSubPix()`. Por fim, os pontos são desenhados sobre a imagem usando a função `drawChessboardCorners()`. A função `calibrateCamera()` é utilizada para obter a matriz de intrínsecos K , os parâmetros de distorção e os dados de rotação e translação a partir dos pontos dados obtidos nos 10 snapshots.

Os elementos da matriz K e os parâmetros de distorção obtidos nas rodadas de calibração são acumulados em dataframes da biblioteca Pandas do Python. Ao término das 5 rodadas de calibração, são calculadas as médias e os desvios padrão de cada um dos parâmetros usando recursos do Pandas. Os arquivos XML dos parâmetros são gerados a partir das médias dos parâmetros. Essa implementação representa uma melhoria em relação à proposta original do requisito, que previa armazenamento de diversos arquivos XML intermediários. Após a geração dos arquivos XML de parâmetros, o programa permite a medição em pixels na imagem original e na imagem com a distorção corrigida, de modo similar ao que foi feito no requisito 1.

3) *Requisito 3*: Esse requisito visa realizar a calibração dos parâmetros extrínsecos da câmera, considerando que os parâmetros intrínsecos foram obtidos no requisito 2. Inicialmente, o programa captura um conjunto pontos do padrão de calibração para os quais conhecemos as coordenadas do mundo 3D. Então, utilizando o procedimento descrito na fundamentação teórica, obtém o vetor de translação e a matriz de rotação e as apresenta no terminal. A norma do vetor de translação também é apresentada. Espera-se que a norma do vetor de translação corresponda, aproximadamente, à distância do objeto à camera. O procedimento de calibração de extrínsecos foi realizado com o padrão posicionado em 3 diferentes distâncias em relação à câmera. Em cada posição foram realizadas 3 medidas, para que seja possível avaliar a precisão da calibração.

4) *Requisito 4*: Esse requisito visa implementar uma régua virtual. A partir do requisito 3, foi adicionada a capacidade de marcar pontos e desenhar segmentos de reta nas janelas RAW e Undistorted, depois da calibração dos extrínsecos. Para cada ponto selecionado foi utilizada a transformação reversa apresentada na fundamentação teórica para obter as coordenadas do mundo com $Z=0$. O programa apresenta no terminal a distância euclidiana dos pontos em pixels e em centímetros.

Para avaliar a implementação desse requisito, foi utilizado um objeto planar que contém um segmento reta de 5 centímetros, posicionado no mesmo plano do padrão de calibração, conforme a figura 2. Medidas foram realizadas em 3 distâncias da câmera. Em cada distância foram realizadas medidas nas imagens RAW e Undistorted, no centro e na periferia da imagem.

III. RESULTADOS

Nessa seção apresentaremos os resultado obtidos nos experimentos.

A. Requisito 1

A figura 3 apresenta a execução do requisito 1, com um segmento de reta desenhado sobre a imagem. O comprimento do segmento em pixels é apresentado no terminal.

B. Requisito 2

A matriz de calibração intrínseca obtida a partir das médias das rodadas de calibração e os parâmetros de distorção com os

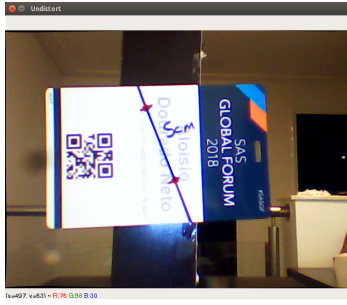


Figura 2. Objeto planar utilizado no requisito 4

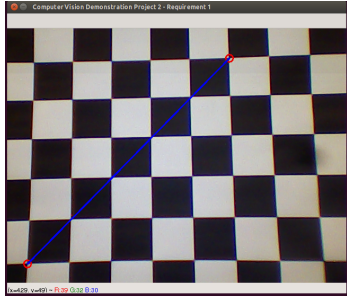


Figura 3. Execução do requisito 1

respectivos desvios padrão dos parâmetros são apresentados a seguir:

$$K = \begin{bmatrix} 849.7 & 0 & 329.2 \\ 0 & 864.8 & 216.8 \\ 0 & 0 & 1 \end{bmatrix}, \text{ std} = \begin{bmatrix} 52.2 & 0 & 13.0 \\ 0. & 49.6 & 7.6 \\ 0 & 0 & 0 \end{bmatrix} \quad (13)$$

$$\text{dist} = \begin{bmatrix} -2.25661237e - 01 \\ 3.72194987e + 00 \\ 6.18564334e - 03 \\ -7.00773416e - 03 \\ -1.17241379e + 01 \end{bmatrix}, \text{ std} = \begin{bmatrix} 1.01752861e - 01 \\ 1.61197433e + 00 \\ 2.25987665e - 04 \\ 2.43222455e - 02 \\ 6.66162928e + 00 \end{bmatrix} \quad (14)$$

C. Requisito 3

A relação entre a norma do vetor de translação e a distância medida no requisito 3 é apresentada na tabela I. Pela tabela, observamos que o erro aumenta e a precisão cai à medida em que a distância aumenta.

D. Requisito 4

Os resultados obtidos no requisito 4 são apresentados na tabela II. Observa-se um aumento do desvio padrão das medidas à medida em que a distância aumenta. Isso pode ser

Dist. (cm)	norm(t)	erro	std	std%
30	29,41	-1,96%	0,00052	0,002%
55	50,15	-8,82%	0,00411	0,007%
110	97,56	-11,31%	0,04970	0,045%

Tabela I

RESULTADOS DO REQUISITO 3

	D_{min}		D_{med}		D_{max}	
Posição	med	std	med	std	med	std
dist(cm)	30		55		110	
norm(t)	29,41	0,00052	50,15	0,00411	97,56	0,04970
$l_{raw:cen}$	4,17	0,03786	4,21	0,05508	4,08	0,09539
$l_{raw:per}$	4,18	0,02309	4,29	0,01732	4,25	0,07024
$l_{und:cen}$	4,21	0,06000	4,26	0,05508	4,28	0,04619
$l_{und:per}$	4,17	0,02887	4,17	0,05132	4,18	0,09815

Tabela II

RESULTADOS DO REQUISITO 4

explicado pela dificuldade de seleção dos pontos com o mouse quando a imagem fica menor (mais distante). Entretanto, não se observa variação significativa entre as medidas realizadas na imagem RAW e na imagem com distorção corrigida, o que pode significar que a câmera utilizada tem pouca distorção. Observa-se também que o erro médio das medidas é da ordem de 16%, maior que o erro médio observado na norma do vetor de translação.

IV. DISCUSSÃO E CONCLUSÕES

Em razão de não termos observado diferenças significativas entre as medições realizadas nas imagens RAW e com distorção corrigida, podemos concluir que a câmera utilizada apresenta baixo nível de distorção.

Entretanto o nível de erro médio de 16% indica baixa acurácia do método, o que limita sua aplicação como instrumento de medida.

Diversos fatores podem contribuir para esta baixa acurácia: ruído na imagem e baixa resolução da câmera, problemas de discretização que podem se multiplicar nas diversas operações matemáticas realizadas e uma eventual baixa performance do método utilizado para minimizar os resíduos durante o cálculo dos elementos da matriz P.

Para obter melhores estimativas sem alterar o hardware utilizado, o ponto que pode ser melhorado nesse processo é a minimização dos resíduos na determinação da matriz de projeção. O presente trabalho utilizou o método *least_squares()* sem qualquer tuning. Outras abordagens podem que incluam possibilidade de tuning de hiper-parâmetros podem ser experimentadas.

REFERÊNCIAS

- [1] E. Murphy-Chutorian and M. M. Trivedi, "Head pose estimation in computer vision: A survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 31, no. 4, pp. 607–626, April 2009.
- [2] G. Shakhnarovich, P. Viola, and T. Darrell, "Fast pose estimation with parameter-sensitive hashing," in *Proceedings Ninth IEEE International Conference on Computer Vision*, Oct 2003, pp. 750–757 vol.2.
- [3] OpenCV, "Camera calibration with opencv," 2016. [Online]. Available: https://docs.opencv.org/3.1.0/d4/d94/tutorial_camera_calibration.html
- [4] R. Hartley and A. Zisserman, *Multiple view geometry in computer vision*. Cambridge University Press, OCLC: 804793563. [Online]. Available: <http://public.eblib.com/choice/publicfullrecord.aspx?p=256634>
- [5] T. de Campos, "3d visual tracking of articulated objects and hands," Ph.D. dissertation, University of Oxford, 2006.
- [6] A. Kaehler and G. Bradski, *Learning OpenCV 3: Computer Vision in C++ with the OpenCV Library*. O'Reilly Media, 2016. [Online]. Available: <https://books.google.com.br/books?id=Lpm3DQAAQBAJ>