

Calibração de Camera

Abdullah Zaiter
15/0089392
abdullah.zaiter@gmail.com
Ian Moura Alexandre
15/0129661
ianzeba@gmail.com

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

Abstract

Este documento apresenta os fundamentos teóricos, a metodologia e os resultados obtidos para a implementação de um algoritmo para calibração de uma câmera e, com os parâmetros conhecidos da câmera usando OpenCV [?] o desenvolvimento de um sistema de medição utilizando-se a *webcam*.

1 Introdução

^{1, 2} Fazendo-se uso da câmera como um sensor quantitativo, torna-se essencial a sua calibração. A calibração é uma estimação de um modelo não-configurado de câmera, com o intuito de se haver correspondência entre pontos na imagem e pontos no espaço, como dito em [5]. Para isso, acha-se a matriz de projeção M e os parâmetros intrínsecos e extrínsecos. A matriz de projeção apresenta qual é a proporção entre a imagem obtida da imagem da câmera para os objetos no mundo real, obtendo assim o que seria referente a distância focal da câmera. Os modelos utilizados para tal projeção é o modelo *pinhole*, também chamado de modelo de câmera de Tsai [6], em que a sua representação em coordenadas homogêneas é representada pela equação 1, sendo a matriz M uma matriz de dimensões 3×4 , onde f representaria o zoom da imagem.

$$\begin{pmatrix} \tilde{x} \\ \tilde{y} \\ \tilde{z} \end{pmatrix} = \begin{pmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} \quad (1)$$

Para o modelo de uma câmera completa, necessita-se considerar os fatores que podem afetar esta projeção, sendo estes os fatores intrínsecos e extrínsecos. Os fatores intrínsecos são aqueles referentes à aspectos de hardware da câmera, sendo um destes o tamanho dos pixels e seu formato. Dessa forma, é necessário que a matriz M seja dividida pela quantidade de pixels da imagem e o seu referencial levado para o canto esquerdo da imagem. Os fatores

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

¹Abdullah: Desenvolveu o método de calibração de intrínsecos e extrínsecos, e desenvolveu parte da parte da régua visual. No relatório, foi responsável pela Conclusão e Resultados.

²Ian: Desenvolveu a primeira parte do projeto, com a medição da distância entre dois pixels, ajudou no desenvolvimento da régua visual. No relatório, escreveu a introdução e desenvolvimento.

extrínsecos são dependentes das posições linear e angular da câmera. Dessa forma, obtém-se a equação 2, formando-se com as três matrizes a matriz da câmera.

$$\begin{pmatrix} \tilde{u} \\ \tilde{v} \\ \tilde{w} \end{pmatrix} = \begin{pmatrix} \frac{1}{\rho_u} & 0 & u_0 \\ 0 & \frac{1}{\rho_v} & v_0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} \mathbf{R} & t \\ \mathbf{0}_{1 \times 3} & 1 \end{pmatrix}^{-1} \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} \quad (2)$$

Entretanto, o mais comum é que se tenha uma matriz com doze elementos, como obtido na equação 3 e à partir dele é estimado seus valores. Para simplificar a matriz estimada, coloca-se o referencial do mundo no plano do ponto desejado, além de que como o fator de escala na homografia é arbitrário, como mostrado em [1], obtém-se a equação homográfica 4, com 8 parâmetros.

$$\begin{pmatrix} \tilde{u} \\ \tilde{v} \\ \tilde{w} \end{pmatrix} = \lambda \begin{pmatrix} C_{11} & C_{12} & C_{13} & C_{14} \\ C_{21} & C_{22} & C_{23} & C_{24} \\ C_{31} & C_{32} & C_{33} & C_{34} \end{pmatrix} \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} \quad (3)$$

$$\begin{pmatrix} \tilde{u} \\ \tilde{v} \\ \tilde{w} \end{pmatrix} = \begin{pmatrix} C_{11} & C_{12} & C_{14} \\ C_{21} & C_{22} & C_{24} \\ C_{31} & C_{32} & 1 \end{pmatrix} \begin{pmatrix} X \\ Y \\ 1 \end{pmatrix} \quad (4)$$

Tendo a equação 2, a calibração é feita por meio da correspondência de pontos entre diferentes imagens e, sabendo-se um referencial de medida do mundo, pode-se estimar uma homografia e assim mapear pontos na imagem. As aplicações para este tipo de técnica de calibração são diversas, como estimação da posição da cabeça[2], e a realização de medidas em centímetros utilizando-se a transmissão de webcam, como discutido neste documento.

2 Desenvolvimento

Para melhor controle de versão e rastreamento de mudanças, utilizou-se a ferramenta Git em um Repositório do autor no site GitHub. Para o algoritmo realizado, escolheu-se desenvolvê-lo utilizando a linguagem Python, sendo grande parte das funções utilizadas oriundas da biblioteca de visão computacional OpenCV[3]. Para a montagem do código, dividiu-se em quatro requisitos:

- Calcular a distância entre dois píxels da imagem
- Calibração dos Parâmetros Intrínsecos
- Calibração dos Parâmetros Extrínsecos
- Desenvolvimento da Régua Visual

2.1 Distância entre dois pontos na imagem

Na execução desta parte do algoritmo, solicita-se do usuário o índice de sua câmera e em seguida começa-se a transmissão de sua imagem. Para capturar o sinal do mouse do usuário, utiliza-se a função do OpenCV `cv2.SetMouseCallback`, sendo chamado a função `get_mouse_clicks`. Nesta função, espera-se do usuário clicar em dois pontos da imagem, imprimindo-os na tela. Quando selecionados os dois pontos, desenha-se uma linha branca na transmissão do vídeo, imprimindo na tela o comprimento da linha por meio da distância euclidiana entre os dois pontos.

2.2 Cálculo dos Parâmetros Intrínsecos e Extrínsecos

Para o cálculo da matriz de calibração, utiliza-se como padrão de calibração o tabuleiro de xadrez, com padrão de 9x7 quadrados, sendo cada um desses de 3x3 cm. Começa-se esta parte do código solicitando do usuário o intervalo de tempo(Δt) desejado para captura dos padrões de calibração. Inicia-se a transmissão do vídeo, onde a cada Δt é realizada uma captura e depositada em um `deque` de tamanho limitado a cinco e, no momento que o usuário der o comando de captura de dados, os últimos cinco padrões são capturados, passando-se para a etapa de calibração. Obtida a matriz de calibração, salvou-se os resultados em arquivos XML separados para valores de distorção, para as matrizes de intrínsecos e extrínsecos. Tendo estes valores, para se obter valores mais próximos do real, realizou-se o cálculo da média e do desvio padrão de cada parâmetro, passando estes a serem usados para as próximas etapas. Então, finalizando-se a etapa de calibração, tendo feita a parte de calibração, desfez-se a distorção obtida e remapeou-se para se obter a imagem no plano mostrado pelo padrão de calibração.

- calibrate(images):** Esta função recebe como parâmetro as imagens dos padrões de calibração. Nela, verifica-se a presença das intersecções nos padrões de calibração por meio da função do OpenCV `cv.findChessboardCorners` e, caso tenha obtido sucesso ao encontrar os tabuleiros de xadrez, as imagens são salvas com as intersecções identificadas por cores. Feito a verificação para cada uma das cinco imagens, armazena-se os pontos pegos e então realiza-se a operação para captura da matriz de calibração por meio da função de OpenCV `cv.calibrateCamera`. Com essa função, retorna-se também os vetores de rotação e translação. Para realizar a transformação do vetor de rotação em matriz, utiliza-se o `cv.Rodrigues`, como explicado em [4]. Feito isto, concatena-se rotação e translação, formando-se a matriz dos extrínsecos. Por fim, salva-se os coeficientes de distorção, as matrizes de intrínsecos e a matriz de extrínsecos de cada um deles em diferentes arquivos XML, retornando da função os coeficientes de distorção e a matriz da câmera.
- averageMatrixCaluclator(mat):** A função recebe como parâmetro o nome do tipo de matriz a ser tratado (*distortion*, *intrinsics* e *extrinsics*). Nela, direciona-se ao diretório onde encontram-se os arquivos XML previamente salvos e deles se faz a leitura das matrizes em todos os arquivos presentes por meio de um loop, somando-se todos os valores obtidos em uma variável. Finalizado o loop, divide-se o valor da variável pela quantidade de arquivos e retorna-se este valor, referente a média das matrizes.
- stdMatrixCaluclator(mat):** Funciona-se de maneira idêntica à `averageMatrixCaluclator(mat)`, para o cálculo do desvio padrão.
- undistortImage:** Função utilizada para desfazer a distorção presente do padrão de calibração. Nela obtém-se uma nova matriz da câmera por meio da função `NewtonRaphsonUndistort.getOptimal` que recebe a atual matriz de calibração e os coeficientes de distorção, mapeando-se as áreas encontradas do padrão em retângulos, e por meio do método de aproximação de Newton-Raphson, retorna o tamanho da nova projeção e a matriz da projeção.

3 Resultados

3.1 Distância entre dois pontos

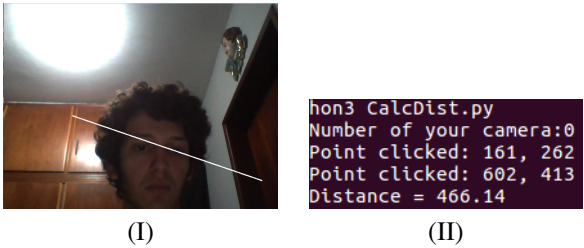


Figure 1: Resultado do cálculo de distâncias: (I) Transmissão do vídeo com a reta entre os dois pontos; (II) Saída do terminal

Observa-se que o primeiro requisito de se calcular a distância em pixels entre dois pontos por meio do comando do mouse do usuário é cumprido, mostrando a distância euclidiana com precisão em milésimos.

3.2 Calibração da Câmera

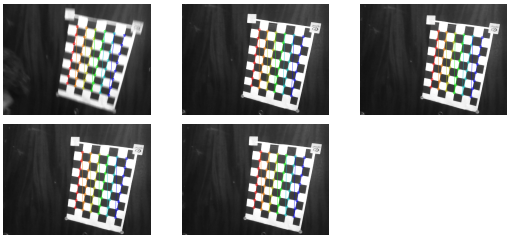


Figure 2: Último conjunto de imagens usado para cálculo dos intrínsecos

Com os dados das outras calibrações, a média e desvio padrão obtidos dos intrínsecos foi, respectivamente:

$$\begin{bmatrix} 5.09 \times 10^2 & 0 & 4.22 \times 10^2 \\ 0 & 4.73 \times 10^2 & 1.89 \times 10^2 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 6,81 & 0 & 9,25 \\ 0 & 10,14 & 3,22 \\ 0 & 0 & 0 \end{bmatrix}$$

E estes foram os dados de média e desvio dos coeficientes de distorção, respectivamente:

$$\begin{bmatrix} 2.229 \times 10^{-1} & -3.811 \times 10^{-1} & -2.663 \times 10^{-2} & 2.95 \times 10^{-2} & 3.02 \times 10^{-1} \\ 7.95 \times 10^{-3} & 4.82 \times 10^{-2} & 9.06 \times 10^{-4} & 2.40 \times 10^{-3} & 6.50 \times 10^{-2} \end{bmatrix}$$

Para a matriz de extrínsecos, o resultado de média e desvio padrão obtido pode ser visualizado logo abaixo:

$$\begin{bmatrix} 9,67 \times 10^{-1} & 1,09 \times 10^{-1} & 2,30 \times 10^{-1} & -2,11 \times 10^2 \\ -1,82 \times 10^{-1} & 9,26 \times 10^{-1} & 3,29 \times 10^{-1} & -2,088 \times 10^1 \\ -1,77 \times 10^{-1} & -3,611 \times 10^{-1} & 9,15 \times 10^{-1} & 3,837 \times 10^2 \end{bmatrix}$$

$$\begin{bmatrix} 4.45 \times 10^{-4} & 4.83 \times 10^{-3} & 1.67 \times 10^{-3} & 6.06 \\ 3.09 \times 10^{-3} & 1.788 \times 10^{-3} & 6.57 \times 10^{-3} & 2.44 \\ 1.819 \times 10^{-3} & 6.018 \times 10^{-3} & 2.65 \times 10^{-3} & 6.78 \end{bmatrix}$$

Para a implementação da função responsável por desfazer a distorção do padrão de calibração, realizou-se os testes com dois métodos diferentes: utilizando-se a função do OpenCV `getOptimalNewCameraMatrix` e a solução sugerida por Yohai Devir[1]. Trabalhando-se com a função do OpenCV, observa-se a necessidade de que as imagens usadas para a calibração tenham uma diferença bem grande posição angular, observando-se para entradas bem próximas e com pouca distorção apresentada um erro de indeterminação matemática, onde o roi se torna zero. A implementação da função feita por Devir visa tratar os erros de indeterminação encontradas na função do OpenCV por meio de uma implementação utilizando-se do método de Newton-Raphson de estimação de raízes, no qual se mostro bem mais preciso e eliminou os problemas antes presentes. Para o cálculo da distancia, baseou-se na equação geral a seguir:

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} f_x & 0 & cx \\ 0 & f_y & cy \\ 0 & 0 & 1 \end{bmatrix} [R|t] \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \tag{5}$$

Onde o vetor X,Y,Z,1, representa a estimativa de posição do objeto, como esse vetor é uma incógnita, calculou-se o vetor, a partir dos parâmetros da câmera, assim, foi necessário calcular a matriz inversa dos parâmetros intrínsecos, gerar a matriz $[R \mid t]$ a partir dos vetores de rotação e de translação, e após isso foi calculada a matriz pseudo inversa desta matriz. Multiplicou-se pela posição do pixel e assim, para determinados dois pixels, foi calculada a distancia pela equação da distancia euclidiana.

3.3 Régua Visual

Os resultados mostrados abaixo são referentes à medição de um lápis com comprimento de 142 mm:

Posição do padrão de calibração	dmin	dmed	dmax
l _{tl} , medida pela trena	48	60	160
l _{tl} , calculada pela calibração extrínseca	47,56	58,6	150,4
l _{raw} ,centre	160,29	124	105
l _{raw} ,perifery	186,30	136,4	100,6
l _{undistorted} ,centre	172,53	133	115,5
l _{undistorted} ,perifery	194,68	146,4	110,3

As imagens 3 mostram duas das medidas realizadas:

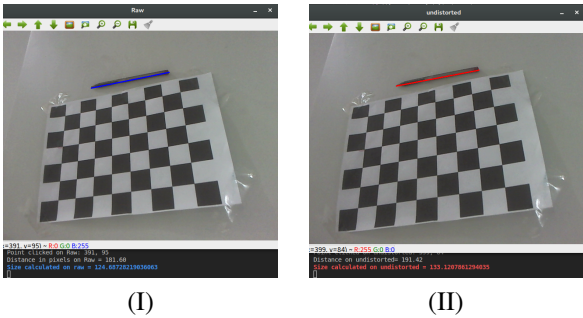


Figure 3: Medidas obtidas a uma distância de 600 mm: (I) Transmissão Raw (Com Distorções); (II) Transmissão Undistorted (Sem Distorções)

Percebe-se pelos dados da tabela que quando a medição é feita distante do padrão do xadrez, a medida realizada acaba sendo bastante comprometido. Deve-se isso muito ao efeito de distorção de bordas, onde se tem distorção das lente[1], algo que deve-se levar em conta quando se mexe com câmeras reais. Outro aspecto notado é de que para regiões muito próximas, o efeito de distorção de bordas apresenta-se em um raio bem menor, comprometendo a medida realizada. Quando a medida realizada é distante, devido ao fato de que os padrões do tabuleiro de xadrez se tornam muito pequenos, causando confusão entre as arestas a serem detectadas, não possibilitando a perfeita captura da medida desejada.

4 Conclusão

Por meio dos resultados obtidos, pode-se dizer que . No cálculo dos intrínsecos, percebe-se que o seu cálculo torna-se bastante comprometido quando toma-se imagens do padrão do tabuleiro de xadrez muito distante, à ponto de que ele confunda a posição entre as diferentes intersecções. Há muitos fatores que interferem no processo de calibração da câmera, alguns são contornáveis como garantir angulação e diferentes posições do objeto de calibração e outros não são como as imperfeições físicas do objeto e da câmera, ruídos, interferências eletromagnéticas e etc. Foi obtido um resultado condizente, a régua implementada não pode ser usada como instrumento de medição exato, porém os resultados foram em ordens de grandeza aceitáveis e próximas das reais. Foram implementadas várias técnicas estudadas em sala de aula e o trabalho permitiu a validação das mesmas, além de grande conhecimento na área de câmeras e calibração.

References

[1] OpenCV Community. Basic concepts of the homography explained with code, February 2018. https://docs.opencv.org/3.4.1/d9/dab/tutorial_homography.html#tutorial_homography_Demol.

[2] Yohai Devir. Opencv calibration fixes, April 2018. <https://github.com/YohaiDevir/OpenCVCalibrationFixes>.

[3] Paulo Dias. Tsai camera calibration. March 2003. http://homepages.inf.ed.ac.uk/rbf/CVonline/LOCAL_COPIES/DIAS1/.

[4] Ari Grant and Sugih Jamin. Rotations. *EECS 487*, 2010.

[5] Berthold K.P. Horn. Tsai’s camera calibration method revisited. 2000.

[6] Craig Hull. What is lens distortion? 2019. <https://expertphotography.com/what-is-lens-distortion/>.

[7] Satya Mallick. Head pose estimation using opencv and dlib. September 2016. <https://www.learnopencv.com/head-pose-estimation-using-opencv-and-dlib/>.

[8] *The OpenCV Reference Manual*. OpenCV Community, 3.4.1 edition, march 2019. URL <https://docs.opencv.org/3.4.1/>.