

Projeto Demonstrativo 2

Pedro Henrique Luz de Araujo

pedrohluzaraujo@gmail.com

Rafael Barbosa de Sousa

rafel.1940.b@gmail.com

Departamento de Ciência da
Computação

Universidade de Brasília

Campus Darcy Ribeiro, Asa Norte
Brasília-DF, CEP 70910-900, Brazil,

Abstract

O presente Projeto Demonstrativo visa a implementar um programa capaz de medir o tamanho de objetos a partir das dimensões correspondentes em uma imagem. Para tanto, calibra-se a câmera usada para captura das imagens, a fim de conhecer os parâmetros da transformação do mundo tridimensional para o plano da imagem, que podem ser intrínsecos ou extrínsecos. Comparamos duas formas de obtenção dos extrínsecos - estimando-os juntamente com os intrínsecos, ou a partir dos intrínsecos e novas imagens - a três distâncias diferentes em cada caso. Obtivemos medidas que se aproximam do esperado, mas concluímos que a técnica pode não servir bem para aplicações que necessitam medidas mais precisas.¹

1 Introdução

O processo de formação de imagens pelas câmeras pode ser entendido como a projeção de uma representação bidimensional de um mundo tridimensional. Tal transformação é um processo que pode ser modelado por uma projeção central na qual um raio de um ponto do espaço tridimensional passa por um centro de projeção e intersecciona um ponto no plano da imagem [8]. Aprender o mapeamento de pontos do mundo para pontos da imagem é central em aplicações como: remoção de distorções; estimação de estrutura tridimensional e medidas de objetos; e estimação de profundidade.

Tal mapeamento pode ser expresso da seguinte forma:

$$\lambda \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \mathbf{P}_{3 \times 4} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}, \quad (1)$$

em que $(u, v, 1)$ e $(X, Y, Z, 1)$ são as coordenadas homogêneas do ponto na imagem e no mundo, respectivamente, e $\mathbf{P}_{3 \times 4}$ é a matriz que realiza a transformação, a qual depende tanto de características da câmera (parâmetros intrínsecos) quanto das coordenadas adotadas (parâmetros extrínsecos).

Segundo [8], existem cinco parâmetros intrínsecos: f_x e f_y são as distâncias focais expressas em pixels (exige-se dois parâmetros para o caso de pixels não quadrados); c_x e c_y

definem as coordenadas do centro da imagem; e θ indica o ângulo entre os dois eixos da imagem. Esse último parâmetro é igual a zero quando tal ângulo é reto, aproximação usada por nós.

Uma vez que os parâmetros intrínsecos adotam coordenadas centradas na câmera, surge a necessidade de deslocar a transformação caso deseje-se outro sistema de coordenadas. Os parâmetros extrínsecos são responsáveis por isso, descrevendo uma matriz de rotação $\mathbf{R}_{3 \times 3}$ e um vetor de translação \mathbf{t} . Portanto, a equação 1 pode ser reescrita como:

$$\lambda \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_1 \\ r_{21} & r_{22} & r_{23} & t_2 \\ r_{31} & r_{32} & r_{33} & t_3 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}. \quad (2)$$

Tal modelo é acurado apenas para uma câmera ideal, em que o ponto no mundo e na imagem e o centro óptico são co-lineares. Entretanto, câmeras reais apresentam distorções, entre elas, a distorção radial. Para remediar essa situação, é preciso corrigir as medidas da imagem para aquelas que seriam obtidas por uma câmera ideal (perfeitamente linear) [4]. A distorção radial pode ser modelada por um polinômio de grau baixo [4], cujos coeficientes são denominados coeficientes de distorção.

A calibração da câmera consiste em calcular todos esses parâmetros. Podemos fazê-lo a partir de correspondências entre pontos no mundo real e suas projeções no mundo da imagem. Com isso, obtemos equações que relacionam tais pontos, em que os parâmetros que queremos calcular são as incógnitas. A partir daí basta resolver as equações empregando métodos de otimização e técnicas de álgebra linear.

2 Metodologia

2.1 Ferramentas

Usamos a biblioteca OpenCV [4], versão 3.3.0, para a captura de vídeo de *webcam* e para obter os parâmetros da câmera. Para realizar operações sobre as matrizes de imagens, utilizamos a biblioteca de computação numérica em Python, NumPy. Usamos ainda, como linguagem, Python 3.5.2, e o gerenciador de bibliotecas Anaconda 3.

2.2 Requisito 1

O requisito 1 trata da obtenção da distância entre dois pixels de uma imagem. A partir de funções da biblioteca OpenCV, é exibida na tela do usuário a captura da *webcam*. O usuário tem a faculdade de clicar em dois pontos quaisquer da captura, momento em que uma linha vermelha é desenhada no *frame* e são exibidos no terminal as coordenadas de ambos os pontos, além da distância entre eles.

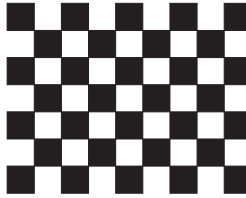
A distância medida é a distância euclidiana em duas dimensões, isto é, a norma do vetor de diferença entre os dois pontos. Sejam $\mathbf{u} = (x_i, y_i)$ e $\mathbf{v} = (x_f, y_f)$ os pontos inicial e final a serem medidos. Então:

$$d = \sqrt{(x_f - x_i)^2 + (y_f - y_i)^2} = \|\mathbf{v} - \mathbf{u}\|, \quad (3)$$

em que d é a distância entre os dois pontos. A norma entre os dois vetores pode ser obtida por meio da biblioteca de álgebra linear do NumPy.

092 2.3 Requisito 2

093 O objetivo do requisito 2 é obter os parâmetros da matriz intrínseca de uma câmera além dos
094 coeficientes de distorção. Para tanto, utilizou-se um padrão de calibração como mostrado
095 na figura 1. Podemos encarar o tabuleiro de xadrez como uma rede de 48 pontos, local-
096 izados nas esquinas dos quadrados internos do tabuleiro. Assumindo o lado do quadrado
097 como a unidade fundamental, possuímos, assim a coordenada dos 48 pontos, considerando a
098 origem como a primeira intersecção entre quatro quadrados (a do canto superior esquerdo).
099 Para estimar os parâmetros intrínsecos precisamos, além das coordenadas dos objetos, das
100 coordenadas correspondentes na imagem.



108 Figure 1: Padrão usado para calibração

110 Para tanto, utilizamos a biblioteca OpenCV para obter automaticamente as coordenadas
111 das esquinas do tabuleiro. Assim, de posse das coordenadas de pontos no mundo real e
112 sua correspondência na imagem, a biblioteca OpenCV pode ser utilizada para estimar os
113 parâmetros intrínsecos. O algoritmo utilizado é baseado em [5] e [6] e consiste em três
114 etapas: inicialização dos valores dos parâmetros intrínsecos e dos coeficientes de distorção;
115 estimativa da pose da câmera a partir desses valores iniciais; e minimização do erro de re-
116 projeção, isto é, a soma das distâncias quadradas entre os pontos da imagem e os pontos
117 projetadas a partir das estimativas dos parâmetros.

118 Para a calibração, capturamos cinco imagens do tabuleiro sob diferentes posições. Tal
119 procedimento foi repetido cinco vezes de modo a obter a média e o desvio padrão da matriz
120 de parâmetros intrínsecos e dos coeficientes de distorção.

121 De posse dessas medidas, foi possível reverter a distorção gerada pela câmera a cada
122 frame capturado pela *webcam*, de modo a possibilitar de medição de distâncias em pixels
123 tanto na captura *raw*, ou distorcida, quanto na captura *undistorted*, ou corrigida. A correção
124 da distorção é simples uma vez que temos os coeficientes de distorção [5]:

$$125 \tilde{x} = x + x[k_1(x^2 + y^2) + k_2(x^2 + y^2)^2] \quad (4a)$$

$$126 \tilde{y} = y + y[k_1(x^2 + y^2) + k_2(x^2 + y^2)^2], \quad (4b)$$

127 em que \tilde{x} e \tilde{y} são as coordenadas após correção e k_1 e k_2 são os coeficientes de distorção.
128

131 2.4 Requisito 3

132 O requisito 3 requer o cálculo dos parâmetros extrínsecos da câmera a partir de três distâncias
133 distintas: d_{min} , a menor distância possível em que é possível observar os 48 pontos do padrão
134 de calibração; d_{max} , a maior distância que permitia a captura dos pontos pelo OpenCV; e
135 d_{med} , uma distância intermediária. Realizamos a medida dos parâmetros extrínsecos de duas
136 maneiras diferentes.
137

Na primeiro cenário, que denominamos *bônus*, os parâmetros extrínsecos foram obtidos juntamente com os intrínsecos, por meio do algoritmo mencionado na subseção anterior.

No segundo cenário, o *regular*, os extrínsecos foram obtidos a partir da matriz de intrínsecos e dos coeficientes de distorção calculados no requisito 2. Primeiramente, medimos o lado do quadrado do padrão de calibração e, adotando o primeiro ponto como origem do mundo e $z = 0$ para todos os pontos do padrão, armazenamos as coordenadas reais de cada um dos 48 pontos em milímetros.

Em seguida, posicionamos a origem padrão de calibração à correspondente distância ao centro da câmera e capturamos uma imagem. Assim, a partir dos parâmetros intrínsecos, dos coeficientes de distorção, das coordenadas reais dos pontos, e das coordenadas correspondentes na imagem obtidas pelo OpenCV, calculamos os parâmetros extrínsecos por meio da função solvePnP, da mesma biblioteca. Esse procedimento foi feito três vezes para cada uma das distâncias, sendo calculado a média e o desvio padrão dos parâmetros extrínsecos de cada distância.

Destacamos que os parâmetros são obtidos na forma de um vetor de rotação \mathbf{r} e um vetor de translação \mathbf{t} . Para obter a matriz de rotação \mathbf{R} basta aplicar o algoritmo Rodrigues em \mathbf{r} . A matriz de extrínsecos, então, é a concatenação de \mathbf{R} e \mathbf{t} .

2.5 Requisito 4

O requisito 4 trata do uso da calibração para obtenção de medidas reais de objetos. Uma vez que as matrizes de calibração da câmera transformam coordenadas do mundo em coordenadas da imagem, podemos gerar uma transformação inversa para, a partir de coordenadas em pixels da imagem, obter medidas reais de objetos. Dada a equação 2, podemos multiplicar a matriz de parâmetros intrínsecos pela de extrínsecos:

$$\lambda \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}. \quad (5)$$

Mas $Z = 0$, já que medimos apenas objetos localizados no plano do padrão de calibração, de modo que podemos retirar a terceira coluna da matriz \mathbf{A} :

$$\lambda \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & a_{14} \\ a_{21} & a_{22} & a_{24} \\ a_{31} & a_{32} & a_{34} \end{bmatrix} \begin{bmatrix} X \\ Y \\ 1 \end{bmatrix}. \quad (6)$$

Assim, obtemos a matriz inversa \mathbf{A}^{-1} :

$$\begin{bmatrix} X \\ Y \\ 1 \end{bmatrix} = \lambda \begin{bmatrix} a_{11} & a_{12} & a_{14} \\ a_{21} & a_{22} & a_{24} \\ a_{31} & a_{32} & a_{34} \end{bmatrix}^{-1} \begin{bmatrix} u \\ v \\ 1 \end{bmatrix}, \quad (7)$$

que é a transformação que usamos para obter as coordenadas reais de objetos.

Por meio dessa transformação, medimos a dimensão real de objetos a partir da correspondente medida na imagem do objeto. Foram medidos dois objetos diferentes: um livro (figura 2) e um porta-fita (figura 3). Os objetos foram medidos nas três distâncias mencionadas no requisito 3. Além disso, realizaram-se medidas nas janelas *raw* e *undistorted*, com o objeto



Figure 2: Livro com largura medida por Pedro.



Figure 3: Porta-fita com comprimento medido por Rafael.

posicionado no centro e na periferia da janela. Em cada uma dessas situações foram realizadas duas medidas, obtendo-se assim a média e o desvio padrão da dimensão do objeto. Por fim, tal procedimento foi realizado com duas câmeras: a de Pedro e a de Rafael. Neste último caso, mediu-se ainda com parâmetros extrínsecos do cenário *regular* e *bonus*.

3 Resultados

3.1 Requisito 1

A figura 4 apresenta o resultado de uma medição de distância entre dois pontos do plano da imagem em pixels. É possível perceber o correto cômputo das coordenadas dos pixels em questão e da distância euclidiana entre eles.

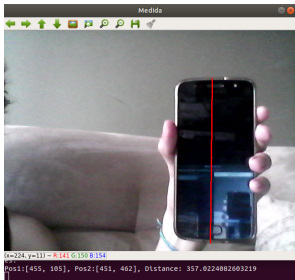


Figure 4: Exemplo de medição do requisito 1.

3.2 Requisito 2

O requisito 2 gerou parâmetros intrínsecos e coeficientes de distorção para as câmeras de ambos autores. As tabelas 1 e 2 apresentam os valores encontrados para os parâmetros intrínsecos, com o desvio padrão como margem de erro.² A figura 5 compara a exibição das janelas *raw* e *undistorted*.

²Os coeficientes de distorção de cada câmera podem ser encontrados na pasta *xmls* do projeto. Optamos por não os apresentar aqui por possuírem baixa interpretabilidade e não acrescentarem às análises que desejamos realizar.

Table 1: Parâmetros da câmera de Pedro.

Parâmetro	Valor
f_x	$7.8 \times 10^2 \pm 0.1 \times 10^2$
f_y	$7.8 \times 10^2 \pm 0.1 \times 10^2$
c_x	$3.7 \times 10^2 \pm 0.3 \times 10^2$
c_y	$2.30 \times 10^2 \pm 0.08 \times 10^2$

Table 2: Parâmetros da câmera de Rafael.

Parâmetro	Valor
f_x	$1.53 \times 10^3 \pm 0.04 \times 10^3$
f_y	$1.54 \times 10^3 \pm 0.04 \times 10^3$
c_x	$9.38 \times 10^2 \pm 0.04 \times 10^2$
c_y	$5.9 \times 10^2 \pm 0.3 \times 10^2$

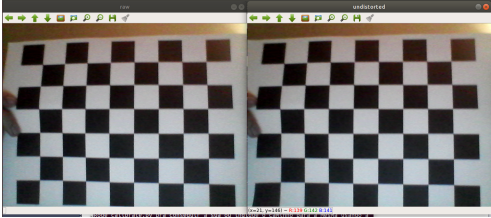


Figure 5: Comparação entre imagem bruta (à esquerda) e com distorção corrigida.

3.3 Requisito 3

Obtivemos vetores de rotação e de translação para as câmeras de Pedro e de Rafael, sendo que neste caso, os vetores foram calculados de duas formas diferentes, como descrito na subseção 2.4.³ A tabela 3 compara a distância real entre a câmera e a origem com a norma do vetor de translação obtido para cada caso.⁴

Table 3: Comparação entre a distância d real e a norma da translação $\|t\|$ calculada.

	d (m)	$\ t\ $ (m)	Erro relativo
d_{min} Pedro	0.402	0.387 ± 0.008	3.73%
d_{med} Pedro	1.030	0.99 ± 0.05	3.88%
d_{max} Pedro	1.740	1.79 ± 0.02	2.79%
d_{min} Rafael	0.290	0.43 ± 0.02	48.27%
d_{med} Rafael	1.205	1.28 ± 0.01	6.22%
d_{max} Rafael	2.700	2.7 ± 0.3	0
d_{min} Bônus	0.290	0.01088 ± 0.00009	96.25%
d_{med} Bônus	1.205	0.0726 ± 0.0003	93.98%
d_{min} Bônus	2.700	0.085 ± 0.001	96.85%

3.4 Requisito 4

As tabelas 4 e 5 demonstram as diversas medições realizadas para o requisito 4 em cada cenário. l_{real} representa a medida real do objeto em questão, l_{raw} a medida na janela raw, $l_{undistorted}$ a medida na janela undistorted, l_{center} medida realizada no centro da tela e $l_{periphery}$ medida realizada nos cantos da tela. Todas as distâncias estão em metros.

³Todos os vetores obtidos estão presentes no diretório “xmls” do projeto.
⁴O erro da norma foi obtido pela propagação dos desvios padrão das medidas do vetor em questão. O erro relativo trata do desvio percentual entre a norma calculada e a distância real.

Table 4: Medidas encontradas a partir de imagens com $l_{real} = 0.196m$ (livro).

Distância	Pedro		
	d_{min}	d_{med}	d_{max}
$\ t_{real}\ $	0.402	1.030	1.740
$\ t\ $	0.387 ± 0.008	0.99 ± 0.05	1.79 ± 0.02
$\ I\ _{raw,center}$	0.161 ± 0.001	0.17 ± 0.01	1.56 ± 0.04
$\ I\ _{raw,perifery}$	0.167 ± 0.002	0.155 ± 0.005	0.170 ± 0.004
$\ I\ _{undistorted,center}$	0.163 ± 0.001	0.172 ± 0.006	0.168 ± 0.04
$\ I\ _{undistorted,perifery}$	0.168 ± 0.003	0.167 ± 0.001	0.167 ± 0.001

Table 5: Medidas encontradas a partir de imagens com $l_{real} = 0.105m$ (porta-fita).

Distância	Rafael			Bonus		
	d_{min}	d_{med}	d_{max}	d_{min}	d_{med}	d_{max}
$\ t_{real}\ $	0.290	1.205	2.700	0.290	1.205	2.700
$\ t\ $	0.43 ± 0.02	1.28 ± 0.01	2.7 ± 0.3	0.01088 ± 0.00009	0.0726 ± 0.0003	0.085 ± 0.001
$\ I\ _{raw,center}$	0.135 ± 0.001	0.106 ± 0.002	0.125 ± 0.001	0.036 ± 0.001	0.062 ± 0.001	0.037 ± 0.001
$\ I\ _{raw,perifery}$	0.156 ± 0.005	0.112 ± 0.001	0.126 ± 0.005	0.037 ± 0.003	0.072 ± 0.001	0.046 ± 0.001
$\ I\ _{undistorted,center}$	0.139 ± 0.006	0.098 ± 0.004	0.121 ± 0.005	0.037 ± 0.002	0.060 ± 0.001	0.038 ± 0.002
$\ I\ _{undistorted,perifery}$	0.137 ± 0.001	0.104 ± 0.001	0.122 ± 0.004	0.031 ± 0.001	0.060 ± 0.001	0.047 ± 0.003

4 Discussão e Conclusões

Primeiramente discutiremos o cenário *regular*, tanto de Pedro e Rafael, para em seguida tecer considerações sobre o cenário *bonus*.

Percebeu-se que a norma do vetor de translação se aproxima mais do valor real na distância d_{max} , apresentado o pior valor em d_{med} . Ainda, nota-se dos dados que as aproximações são boas, com erros inferiores a 6.22%, com a exceção de um *outlier* em d_{min} de Rafael, possivelmente por algum problema no momento de captura das imagens para o cálculo dos parâmetros externos.

As medidas do objeto de Pedro apresentam valores semelhantes em todas as três distâncias e se aproximam da medida real, sendo que os valores que mais fogem da média são aqueles calculados na periferia da imagem distorcida, como esperado. No caso de Rafael, os valores se aproximam bem do esperado nas distâncias d_{med} e d_{max} , apresentando um erro maior em d_{min} . Isso possivelmente está relacionado pelo erro da norma do vetor de translação nessa distância, como mencionamos no parágrafo anterior.

Quanto ao cenário *bonus*, nota-se que os valores da norma da translação e das medidas do objeto são muito inferiores ao esperado. Isso pode ser explicado pelo fato de não serem informadas as medidas do quadrado do padrão de calibração no momento da obtenção dos parâmetros, de modo que os extrínsecos são obtidos em alguma unidade de medida arbitrária. Dito isso, é possível analisar que os tamanhos encontrados são compatíveis nos casos d_{min} e d_{max} . Infelizmente o fato das medidas do cenário *bonus* estarem em unidades arbitrárias dificulta comparações com o cenário *regular*.

O presente trabalho demonstrou que a calibragem de câmeras permite a obtenção de medidas aproximadas do mundo real a partir de uma única imagem. Embora tais medidas se aproximem do valor real, o erro encontrado impossibilitaria o uso da técnica para aplicações que exigem precisão na casa dos centímetros. Observou-se, também, a possibilidade de obtenção dos parâmetros intrínsecos e extrínsecos de uma só vez. Nesse caso, quando não fornecemos ao algoritmo as medidas dos quadrados do plano de calibração, verificamos que ele retorna parâmetros em alguma unidade arbitrária.

O uso de técnicas *stereo*, com mais de uma câmera, possivelmente forneceria medidas mais precisas. Alternativamente, um ambiente mais propício para calibração, com iluminação, suporte, trilhos e outro utensílios também forneceria medidas mais confiáveis.

References

[1] Jean-Yves Bouguet. Camera calibration tool-box for matlab, 2002. URL http://www.vision.caltech.edu/bouguetj/calib_doc/.

[2] G. Bradski. The OpenCV Library. *Dr. Dobb's Journal of Software Tools*, 2000.

[3] David A. Forsyth and Jean Ponce. *Computer Vision: A Modern Approach*. Prentice Hall Professional Technical Reference, 2002. ISBN 0130851981.

[4] Richard Hartley and Andrew Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, New York, NY, USA, 2 edition, 2003. ISBN 0521540518.

[5] Zhengyou Zhang. A flexible new technique for camera calibration. *IEEE Transactions on pattern analysis and machine intelligence*, 22(11):1330–1334, 2000.