

# Projeto Demonstrativo 2 - Calibração de Câmeras

Frederico Guth (18/0081641)  
Tópicos em Sistemas de Computação,  
Turma TC - Visão Computacional (PPGI)  
Universidade de Brasília  
Brasília, Brasil  
fredguth@fredguth.com

**Resumo**—Uma câmera faz um mapeamento geométrico do mundo para uma imagem e é possível, fazer o caminho inverso, mapeando a posição de objetos no mundo a partir da sua imagem. Para isso, entretanto, é preciso conhecer os parâmetros intrínsecos e extrínsecos da câmera, o que chamamos de calibração. Neste projeto mostramos como calibrar a câmera e obter informações do mundo 3D a partir de imagens.

**Index Terms**—câmeras, calibração, parâmetros intrínsecos, parâmetros extrínsecos, projeção reversa

## I. INTRODUÇÃO

Uma câmera faz um mapeamento geométrico do mundo 3D para o plano de uma imagem 2D. Conhecendo seus parâmetros intrínsecos, como distância focal e distorções da lente, e extrínsecos, sua rotação e translação no sistema de coordenadas do mundo real, é possível estimar a posição 3D de um objeto a partir de sua imagem [1].

Isto possibilita diversas aplicações: por exemplo, a mensuração da altura de pessoas registradas em vídeos de câmeras de segurança ou a estimativas de posições de atletas em campo, entre outras.

### A. Objetivos

Os objetivos deste projeto são a aplicação prática da teoria de calibração de câmeras e o desenvolvimento de uma "régua visual", capaz de medir um objeto através da sua imagem.

Mais especificamente deseja-se:

- 1) medir um segmento de reta em imagens através de cliques de mouse;
- 2) realizar a calibração de uma câmera digital, armazenando os parâmetros intrínsecos e os coeficientes de distorções em arquivos XML;
- 3) realizar a calibração de uma câmera digital a partir de diferentes distâncias da câmera, calculando os parâmetros extrínsecos da mesma e avaliando a diferença dos resultados;
- 4) medir um objeto através de sua imagem e comparar com suas dimensões reais;
- 5) analisar os resultados obtidos.

## II. REVISÃO TEÓRICA

### A. Câmera Estenopeica com Coordenadas Homogêneas

Se os pontos do mundo  $X$  e da imagem  $x$  são representados por coordenadas homogêneas, podemos expressar matematicamente a projeção da câmera como uma matriz [1]:

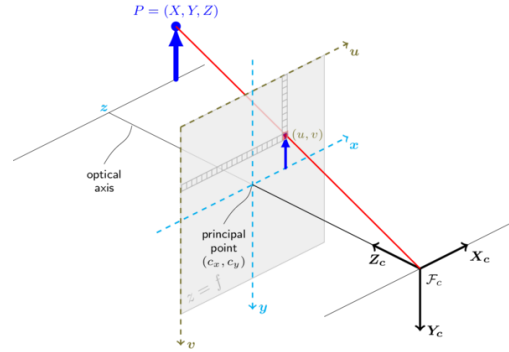


Figura 1. Modelo de Câmera Estenopeica [2]

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

onde  $\lambda$  é um fator de escala e  $P$  é a matriz 3x4 de projeção, também chamada matriz de calibração.

$P$  pode ser decomposta em duas entidades geométricas: os parâmetros intrínsecos  $K$  e extrínsecos  $R$  e  $t$  de calibração [1]

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

$$t = -R\tilde{C} \quad (3)$$

onde  $\tilde{C}$  é a origem do sistema de coordenadas da câmera [3].

Os parâmetros intrínsecos de calibração descrevem a transformação entre a imagem ideal e a imagem em pixels

$$K = \begin{pmatrix} f & s & c_x \\ 0 & \alpha f & c_y \\ 0 & 0 & 1 \end{pmatrix} \quad (4)$$

e os extrínsecos são a rotação  $R$  e translação  $t$  que transformam pontos no espaço do objeto para pontos no espaço da imagem [1].

Como há 6 graus de liberdade nos parâmetros extrínsecos e 5 nos intrínsecos, é necessário pelo menos 6 correspondências  $x_i \leftrightarrow X_i$  do mesmo ponto no espaço da imagem e no espaço do objeto para obter  $P$  [1]. Mas dado que há um erro inerente nas medidas experimentais, para melhorar a qualidade da estimativa é preciso usar  $n > 6$  correspondências (como será visto em III, usaremos 48 pontos) e, assim, não há uma única matriz  $P$  que satisfaz esse sistema de equações.

Precisamos, portanto, adicionar restrições para encontrar uma solução única.

Um método comum é adicionar a restrição  $p_{34} = 0$  [3], mas uma melhor abordagem [1] é:

$$P = \min_{P'} \sum_i d(x_i, P'Xi)^2 \quad (5)$$

onde  $d(x_i, P'Xi)$  é a distância euclidiana entre o ponto observado e o estimado.

### B. Distorções

O modelo até aqui descrito descreve uma câmera ideal, mas as lentes das câmeras reais podem gerar distorções, que também são parâmetros intrínsecos que precisam ser considerados.

A distorção radial é a mais importante e causa uma curvatura no mapeamento.

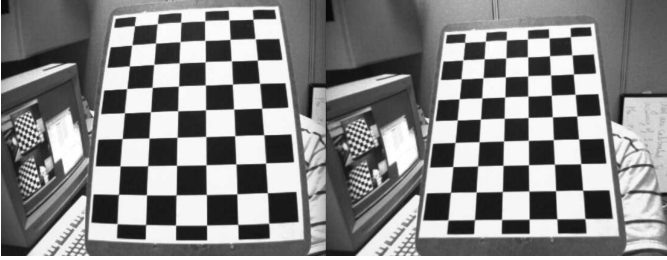


Figura 2. Distorção Radial [2]

A correção dessa distorção pode ser modelada da seguinte maneira [2]:

$$x_{retificado} = x(1 + k_1r^2 + k_2r^4 + k_3r^6)$$

$$y_{retificado} = y(1 + k_1r^2 + k_2r^4 + k_3r^6)$$

Outra distorção comum é a tangencial, que ocorre quando o plano da lente não está alinhado. Para corrigir:

$$x_{retificado} = x + [2p_1xy + p_2(r^2 + 2x^2)]$$

$$y_{retificado} = y + [p_1(r^2 + 2y^2) + 2p_2xy]$$

Esses cinco parâmetros são conhecidos como coeficientes de distorção ( $k_1$   $k_2$   $p_1$   $p_2$   $k_3$ ) [2].

## III. METODOLOGIA

O modelo da câmera e seus parâmetros foram descritos na seção de Revisão Teórica. Nesta seção, descreve-se como estimá-los experimentalmente.

### A. Materiais

Foram utilizados:

- uma tábua de compensado
- papel contact
- fita adesiva
- um padrão de calibração xadrez impresso em papel A4
- uma trena
- uma régua
- computador MacBook Pro (Retina, 13-inch, Early 2015), Processador Intel Core i5 2,7 GHz, 8GB de RAM
- Python 3.6.3 :: Anaconda custom (64-bit)
- OpenCV 3.4.0
- sete programas em python especialmente desenvolvidos para o projeto. Todos estão disponíveis no repositório: [git@github.com:fredguth/unb-cv-3183.git](https://github.com:fredguth/unb-cv-3183.git)

### B. Preparação

- 1) Imprime-se o padrão de calibração em folha A4 e o cola à tábua de compensado usando o Papel Contact.
- 2) Com o programa *requisito1.py*, abre-se uma imagem jpg e com cliques do mouse desenha-se um segmento de reta sobre a imagem entre o primeiro e o segundo clique, registrando-se a distância  $\|p_2 - p_1\|_2$  na própria imagem aberta.

### C. Obtenção dos parâmetros intrínsecos

- 1) Executa-se o programa *requisito2.capture.py* que abre um stream de vídeo e grava a imagem sempre que detecta os cantos dos quadrados no padrão de calibração de tabuleiro de xadrez. O programa pede o número do experimento e grava as imagens capturadas no diretório do mesmo. Foram feitos 8 experimentos que geraram entre 25 e 70 imagens cada.
- 2) Executa-se o programa *requisito2.calibrate.py* para cada experimento. O programa detecta os cantos dos quadrados do padrão de calibração xadrez e refina essa detecção para obter os parâmetros intrínsecos  $K$  e os coeficientes de distorção. É importante mover o quadrado no campo de captura da câmera em diversas orientações e posições, em diferentes distâncias. Os parâmetros intrínsecos e coeficientes de distorção são automaticamente armazenados em arquivos xml nos diretórios dos respectivos experimentos.
- 3) Dado que já temos os coeficientes de distorção, com o programa *requisito2.measure.py* retificamos as imagens da câmera e permitimos medir distâncias na imagem retificada em pixels.
- 4) Após todos os experimentos executados, executa-se o programa *requisito2.analyse.py* que gera a média e o desvio padrão dos parâmetros intrínsecos, salvando-os no diretório *exp-0*.

### D. Obtenção dos parâmetros extrínsecos

- 1) Com a régua, mede-se o tamanho de um quadrado no padrão de calibração.

- 2) Executa-se o programa *requisito3.py*, que computa a correspondência entre pontos da imagem e do espaço do objeto, atribuindo como origem do sistema de coordenadas do mundo, o ponto de intersecção do canto superior esquerdo do tabuleiro e através da função *solvePnP* da OpenCV [4], obtem os parâmetros extrínsecos  $R$  e  $t$ .
- 3) Quando o programa pede o número do experimento, escolhe-se 0, uma vez que queremos usar os parâmetros intrínsecos médios, medidos na etapa anterior.
- 4) Deixa-se a câmera em um ponto marcado pela fita adesiva, tentando colocá-la ortogonal ao plano da sua base.
- 5) Com a câmera posicionada, posiciona-se o tabuleiro na distância mais próxima possível da câmera em que o padrão de calibração tem seus cantos detectados (os cantos ficam marcados com pontos coloridos na *stream* da câmera), marcando com a fita adesiva este ponto como  $d_{min}$ .
- 6) Repete-se o passo anterior, tentando encontrar a distância mais afastada da câmera. Marca-se este ponto com fita adesiva como  $d_{max}$ ; e também um ponto intermediário entre  $d_{min}$  e  $d_{max}$ , que chamamos  $d_{med}$ .
- 7) Medimos as distâncias  $d_{min}$ ,  $d_{med}$  e  $d_{max}$  com a trena.
- 8) Com todos os pontos marcados e a câmera posicionada, acionamos o modo de captura apertando a tecla *espaço*.
- 9) Leva-se o tabuleiro de xadrez para  $d_{min}$  e após algumas capturas (identificação do padrão xadrez na imagem *stream* da câmera), apertamos a tecla *espaço* novamente para sair do modo captura.
- 10) Repete-se os últimos dois passos para as distâncias  $d_{med}$  e  $d_{med}$ . As imagens captadas com as respectivas distâncias medidas são gravadas no diretório do experimento selecionado. É importante verificar se há pelo menos 3 imagens gravadas para cada distância.

#### E. Obtenção da altura de um objeto através da sua imagem

Com os parâmetros intrínsecos e extrínsecos, é possível medir um objeto no mundo real a partir da sua imagem. Neste projeto, entretanto, não se obteve êxito em desenvolver essa funcionalidade no programa *requisito4.py*

### IV. RESULTADOS

Nesta seção apresentamos, para cada etapa da III, os resultados obtidos. Todos os dados podem ser acessados no repositório do projeto (III-A).

#### A. Medição em pixels de segmentos de imagens

A figura IV-A mostra a medição em pixels de segmentos entre cliques na imagem. Além da saída no *console*, o programa mostra a distância em pixels na própria tela.

#### B. Calibração dos parâmetros intrínsecos

A figura IV-B mostra diversas das imagens captadas para calibração, onde tentou-se explorar ao máximo a diferença de posição, distância e até luminosidade do ambiente. O resultado

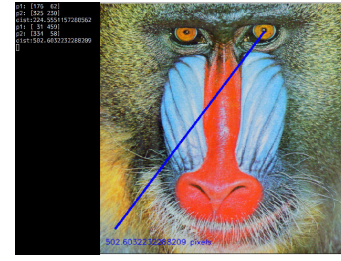


Figura 3. Medição em pixels de segmentos em imagens

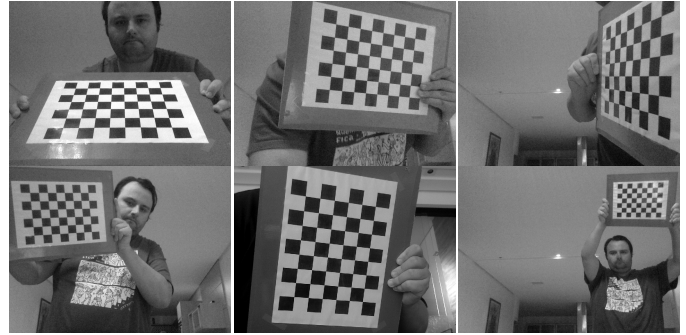


Figura 4. Algumas das dezenas de *snapshots* geradas em diferentes posições

obtido para os parâmetros intrínsecos foram:

$$K = \begin{pmatrix} 673.08 & 0 & 312.32 \\ 0 & 668.10 & 236.50 \\ 0 & 0 & 1 \end{pmatrix} \pm \begin{pmatrix} 56.27 & 0 & 13.24 \\ 0 & 55.75 & 22.71 \\ 0 & 0 & 1 \end{pmatrix}$$

$$(k_1 = 0.150 \pm 0.039 \quad k_2 = -0.744 \pm 0.448 \quad p_1 = 0.003 \pm 0.003 \quad p_2 = -0.003 \pm 0.007 \quad k_3 = 1.820 \pm 1.932)$$

#### C. Calibração dos parâmetros extrínsecos

Na figura IV-C mostramos como foram capturadas as imagens para obtenção dos parâmetros extrínsecos. O resultado está apresentado na tabela IV-C. O erro ficou entre 2 e 6%.

Tabela I  
COMPARAÇÃO DISTÂNCIAS ESTIMADAS VERSUS REAL

em cm	<b>Real</b>	$d_1$	$d_1$	$d_3$	$d$	$\sigma$
$d_{min}$	32.3	31.35	31.93	31.97	31.75	0.35
$d_{med}$	72.3	69.51	68.87	68.57	68.98	0.48
$d_{max}$	148.1	138.78	141.87	144.31	141.65	2.77

Um pouco acima da margem de 2% do desvio padrão.



Figura 5. Montagem III-D.5

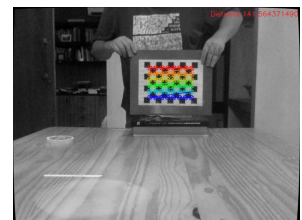


Figura 6. Medição III-D.9

#### D. Medição de Objeto 3D a partir de sua imagem

### V. DISCUSSÃO E CONCLUSÕES

Neste trabalho, utilizamos o modelo de câmera estenopeica para fazer mapeamentos de posições no mundo real para a imagem e vise-versa. O resultado obtido para as distâncias IV-C na calibração dos parâmetros extrínsecos tiveram um erro menor do que 6%. Entretanto, cabe ressaltar que para isso foi necessário capturar centenas de imagens na fase de calibração dos intrínsecos. Tentativas feitas que usavam poucas dezenas não lograram êxito.

### REFERÊNCIAS

- [1] T. de Campos, “3d visual tracking of articulated objects and hands,” Ph.D. dissertation, University of Oxford, 2014.
- [2] OpenCV, “Opencv: Camera calibration and 3d reconstruction,” Dezembro 2017, [Online; Generated on Fri Dec 22 2017 22:15:38]. [Online]. Available: [https://docs.opencv.org/3.4.0/d9/d0c/group\\_\\_calib3d.html](https://docs.opencv.org/3.4.0/d9/d0c/group__calib3d.html)
- [3] R. I. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*, 2nd ed. Cambridge University Press, ISBN: 0521540518, 2004.
- [4] G. Bradski, “The OpenCV Library,” *Dr. Dobbs’ Journal of Software Tools*, 2000.