

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—This document is a model and instructions for L^AT_EX. This and the IEEEtran.cls file define the components of your paper [title, text, heads, etc.]. *CRITICAL: Do Not Use Symbols, Special Characters, Footnotes, or Math in Paper Title or Abstract.

Index Terms—component, formatting, style, styling, insert

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 distorção 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 [?].

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 que sejam desenvolvidos programas usando a biblioteca OpenCV capazes de:

- 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 [?]:

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

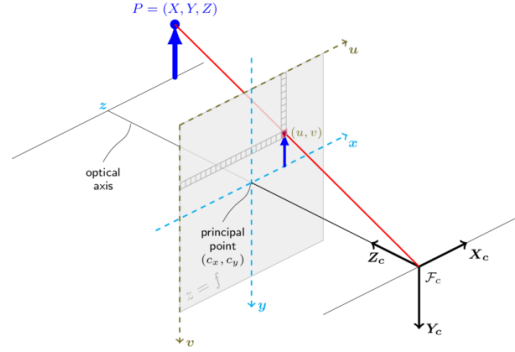


Figura 1. Modelo de Câmera Estenopeica [?]

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

P pode ser decomposto em duas entidades geométricas: os parâmetros intrínsecos e extrínsecos de calibração [?]

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

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

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

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 e vice-versa [?].

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 [?].

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 resolve esse sistema de equações. Precisamos, portanto, adicionar restrições para encontrar uma solução única.

Um método comum é adicionar a restrição $p_{34} = 0$ [?], mas uma melhor abordagem [?] 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 causa uma curvatura no mapeamento. (inserir imagem distorção)

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

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

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

Outra distorção comum é a tangencial, que ocorre quando o plano da lente não está alinhado perfeitamente em paralelo ao plano da imagem. 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).

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 /textit./exp-0/.

D. Obtenção dos parâmetros extrínsecos

- 1) 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 [?], obtem os parâmetros extrínsecos R e t.
- 2) Quando o programa pede o número do experimento, escolhe-se 0, uma vez que queremos usar os parâmetros intrínsecos médios, medidos na etapa anterior.
- 3) Deixa-se a câmera em um ponto marcado pela fita adesiva, tentando colocá-la ortogonal ao plano da sua base.
- 4) Com a câmera posicionada, posiciona-se o tabuleiro na distância mais próximo possível da câmera em que o padrão de calibração tem seus cantos detectados (os cantos ficam marcados com pontos coloridos no *stream* da câmera), marcando com a fita adesiva este ponto como d_{min} .

- 5) Repete-se o passo anterior, tentando encontrar a distância mais afastada da câmera em que os cantos do padrão de calibração ainda são detectáveis. Marca-se este ponto com fita adesiva como d_{max} ; e um ponto intermediário entre d_{min} e d_{max} , que chamamos d_{med} .
- 6) Medimos a distância da câmera à origem do padrão de calibração usando a trena. Para cada distância, obtemos três vezes os valores de R e t.
- 7) Com todos os pontos marcados e a câmera posicionada, acionamos o modo de captura apertando a tecla *espaço*
- 8) 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.
- 9) 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 (no caso, */exp-0/*).

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

A. Medição em pixels de segmentos de imagens

B. Obtenção dos parâmetros intrínsecos

C. Obtenção dos parâmetros extrínsecos

V. DISCUSSÃO E CONCLUSÕES

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
- [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’s Journal of Software Tools*, 2000.