

Calibração de câmeras

Bruno Fernandes Carvalho - 15/0007159

Dep. Ciência da Computação - Universidade de Brasília (UnB)

Princípios de Visão Computacional - Turma A

Data de realização: 20/04/2017

brunofcarvalho1996@gmail.com

Abstract

There are many applications that need real world's information from digital images, as knowing the real size of an object estimated from the distance of it in pixels. In this project, it is proposed a calibration method to obtain relevant data of the camera, providing enough information to create a visual ruler. Assuming that an object was positioned from a known camera distance, it was possible to estimate its dimension in different positions of the image. Also, it was seen the distance and camera lens distortion influence over the estimated value.

Abstract

Existem diversas aplicações que precisam de alguma informação do mundo real a partir de imagens digitais, como saber o tamanho real de um objeto estimado pela distância dos pixels na imagem. Nesse trabalho, é proposto um método de calibração para obter dados relevantes da câmera e que permitem criar uma régua visual. A partir de um objeto posicionado à distâncias conhecidas da câmera, foi possível estimar sua dimensão em diferentes posições da imagem, sendo verificado também a influência da distância e da distorção das lentes da câmera sobre o valor estimado.

1. Objetivos

Essa atividade tem como objetivo entender o processo de calibração de câmeras e como isso permite realizar diversas aplicações, como a régua visual.

2. Introdução

Para entender melhor a calibração, é necessário entender como funciona uma câmera pinhole: seu princípio básico de funcionamento é que os raios de luz devem convergir para um único ponto, evitando que vários raios de luzes iguais ajudem a formar a imagem. Isso evita borramento

e superposição de raios que foram refletidos pela mesma região do objeto. A figura 1 deixa mais claro essa ideia.

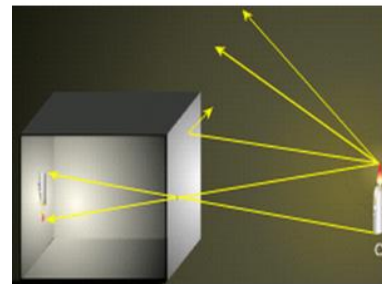


Figura 1. Câmera obscura com um buraco pequeno de abertura

Além disso, para melhorar a convergência dos raios, são adicionadas lentes convergentes. Em contrapartida, isso induz distorção radial e tangencial da imagem. Também existe um parâmetro importante que permite relacionar a imagem virtual com a real: distância focal. É por esse ponto focal que os raios de luz convergem devido ao posicionamento das lentes. A figura 2 seguir mostra um esquemático simples de uma câmera pinhole.

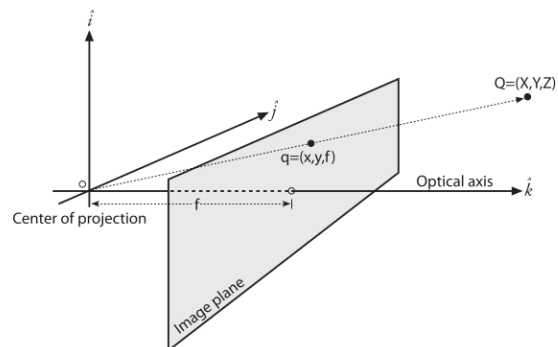


Figura 2. Representação de uma câmera pinhole

O processo de calibração consiste em obter duas matrizes: a matriz da câmera contendo os parâmetros intrínsecos desta e a matriz de distorção, que estima os valores de distorção da lente da câmera. Tendo essas matrizes, é possível corrigir distorções criadas a partir do processo de obtenção da imagem e ter informações relevantes do mundo real a partir dos pixels. Uma dessas relações é saber quantos pixels representam quantos cm no mundo real.

A matriz intrínseca é dada por:

$$\begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix}$$

sendo que f_x e f_y representam as distâncias focais da câmera em pixel e c_x e c_y são o centro ótico da imagem em pixel. As variáveis f_x e f_y formam uma relação com a distância focal real (F), já que $f_x = F * s_x$ e $f_y = F * s_y$, sendo s_x e s_y a quantidade de pixels por cm. A partir de relações geométricas da figura 2 e dos parâmetros intrínsecos obtidos, é possível relacionar um ponto no mundo com o um ponto na imagem, como mostrado a seguir:

$$x_{im} = f_x \left(\frac{X}{Z} \right) + c_x \quad (1)$$

$$y_{im} = f_y \left(\frac{Y}{Z} \right) + c_y \quad (2)$$

Vale deixar claro que (X, Y, Z) são as coordenadas de um ponto no mundo e (x_{im}, y_{im}) são as coordenadas de um pixel na imagem. Assim, a partir de dois pontos num plano, como um objeto plano ou uma imagem, é possível calcular a distância euclidiana bidimensional entre esses dois pontos, como mostrado abaixo.

$$D = \sqrt{(x_f - x_i)^2 + (y_f - y_i)^2} \quad (3)$$

Já a matriz de distorção é dada por: $[k_1, k_2, p_1, p_2, k_3]$, sendo que k_n representa os parâmetros de distorção radial e p_n os parâmetros de distorção tangencial. A primeira é causada pelo aumento no grau de convergência dos raios de luz nas regiões radiais da lente e a segunda ocorre devido a erros no paralelismo das lentes da câmera. Ambas podem ser corrigidas usando as seguintes fórmulas:

$$\begin{aligned} x_{corrected} &= x(k_1 r^2 + k_2 r^4 + k_3 r^6) \\ y_{corrected} &= y(k_1 r^2 + k_2 r^4 + k_3 r^6) \\ x_{corrected} &= x + [2p_1 xy + p_2(r^2 + 2x^2)] \\ y_{corrected} &= y + [2p_2 xy + p_1(r^2 + 2y^2)] \end{aligned} \quad (4)$$

Para estimar os parâmetros dessas duas matrizes, utiliza-se algum objeto com algum padrão conhecido (tabuleiro de xadrez), permitindo que métodos do OpenCV achem os cantos internos de cada quadrado. Se várias fotos forem tiradas com ângulos e posições diferentes, a posição dos cantos de cada quadrado no tabuleiro dão informação suficiente para que seja calculado por meio do método dos mínimos quadrados os parâmetros internos da câmera.

3. Materiais e Metodologia

Foi desenvolvida uma aplicação que calibra uma câmera USB conectada ao computador e, após isso, permite que o usuário crie uma régua visual e compare a medida estimada pelo programa com a medida real de um objeto. Além disso, o usuário pode escolher entre calibrar a câmera ou apenas carregar os arquivos XML previamente obtidos.

Na parte de calibração, foi usado como base o código disponibilizado no Moodle. Para complementá-lo, foi adicionado um loop que calcula os valores de calibração (parâmetros intrínsecos e de distorção) cinco vezes, salvando as matrizes obtidas em cada uma das vezes em arquivos XML diferentes. No final desse processo, é mostrado no terminal as matrizes de desvio padrão tanto dos parâmetros intrínsecos quanto dos de distorção e salvo em arquivos XML a média das matrizes de cada uma das vezes, permitindo obter valores mais precisos e menos sujeitos a ruídos e variações aleatórias de luminosidade do pixel.

Depois da calibração, foi feito um algoritmo que, a partir de dois cliques do usuário numa das imagens mostradas ("raw" e "undistorted"), desenha nela uma reta entre os dois pontos clicados. Também é mostrado no terminal o ponto inicial e final, assim como a distância euclidiana (3) entre eles em pixels e em cm. Para estimar essa distância em cm, é necessário usar as equações (1) e (2) para transformar a coordenada da imagem para o mundo. Pode-se observar que essa estimativa é feita com os parâmetros intrínsecos obtidos anteriormente.

Além disso, para testar essa régua visual, dividiu-se a imagem em um grid de 9 quadrados, para que o objeto pudesse ser posicionado no centro de cada um e o usuário criasse uma reta na imagem com a dimensão do objeto. Essa divisão também foi feita para observar a influência da distorção nos quadrados mais afastados do centro sobre a distância real estimada.

Para testar a aplicação, foi utilizada a própria câmera do computador, um tabuleiro de xadrez impresso a laser (igual ao disponibilizado no Moodle) e um pen-drive como objeto para testar a régua visual desenvolvida.

A calibração foi feita utilizando 25 diferentes imagens (rotação em torno dos três eixos) do tabuleiro de xadrez em cada uma das cinco repetições, sendo que esse processo foi feito em um local com boa iluminação para evitar erros no cálculo da posição dos pixels de canto. Tendo a matriz final de distorção e dos parâmetros intrínsecos, foi testada a régua visual da seguinte forma: um pen-drive com tamanho conhecido foi colocado em cada um dos quadrados do grid à distâncias conhecidas da câmera (30cm, 80cm, 150cm e 300cm). Em cada distância e em cada quadrado, foram realizadas três medidas usando a régua visual e calculada a média. Conforme descrito acima, foi trocada a distância inicial de 15cm para 30cm, com finalidade de melhorar a visualização do objeto em cada quadrado.

É importante dizer que o usuário deve informar a distância do objeto à câmera durante a execução do programa. Se após essa definição o usuário quiser trocar o valor, basta apertar 'p' (pause) com uma das janelas de imagem selecionadas. Vale deixar claro também que, para o desenvolvimento dos algoritmos, foi utilizado a linguagem de programação C++, assim como técnicas de orientação a objetos. Na hora da execução do código no terminal, é necessário informar se deseja executar o programa para calibrar (argumento "CALIB") ou apenas para carregar arquivos XML com os nomes "IntrinsicsFinal.xml" e "DistortionFinal.xml" (argumento "LOAD").

4. Resultados

Seguindo a metodologia explicada na seção anterior, foi feita a calibração da câmera do computador cinco vezes, obtendo os seguintes parâmetros que foram utilizados no teste da régua visual:

$$f_x = 758, f_y = 781, c_x = 332, c_y = 324 \\ k_1 = 0,37, k_2 = -0,17, p_1 = 0,069, p_2 = 0,071$$

Observando esses valores encontrados, é possível dizer que eles foram razoáveis e caracterizam bem os aspectos intrínsecos da câmera, além de removerem a distorção das lentes da mesma. Outro fator que confirma que a calibração foi bem feita são os dados obtidos durante o teste da régua visual, que aproximam as dimensões de um objeto pela imagem.

Conforme dito antes, foi usado um pen-drive de 5 cm de altura. Posicionando-o em diferentes distâncias da câmera, foram tiradas medidas da régua visual nas duas janelas disponibilizadas na aplicação: "raw" e "undistorted". A primeira contém a imagem captura sem alterações e a segunda contém a mesma imagem, só que corrigida usando

os parâmetros da matriz de distorção. Realizando várias medidas em cada uma das células, obteve-se a média delas nas tabelas 1 e 2. Vale deixar claro que todas as medidas estão em cm.

Tabela 1. Distância real medida na janela raw

Distância	30cm	80cm	150cm	300cm
Célula 1	3,8	4,2	4,48	4,59
Célula 2	4,13	4,2	4,41	4,55
Célula 3	3,6	4,4		4,79
Célula 4	3,55	4,0	4,65	5,1
Célula 5	3,7	4,6	4,68	4,88
Célula 6	3,8	4,3	4,78	4,92
Célula 7	3,97	3,8	4,89	4,56
Célula 8	4,21	3,6	4,65	4,7
Célula 9	4,28	3,6	4,5	4,9

Tabela 2. Distância real medida na janela undistorted

Distância	30cm	80cm	150cm	300cm
Célula 1	3,92	4,46	4,58	4,91
Célula 2	3,88	4,34	4,61	4,57
Célula 3	4,12	4,45	4,82	4,87
Célula 4	3,57	4,1	4,7	5,11
Célula 5	3,7	4,55	4,65	4,95
Célula 6	3,8	4,54	4,73	4,92
Célula 7	3,99	4	5,02	4,65
Célula 8	4,03	4,2	4,56	4,98
Célula 9	4,3	4,1	4,98	5,2

A partir das duas tabelas, é sensato dizer que as medidas se aproximam razoavelmente da dimensão real esperada, apresentando em média um erro de 10%. Além disso, esse erro é aumentado devido ao pequeno tamanho do objeto de teste, já que qualquer erro do usuário na hora de medir influencia bastante o número de pixels da reta criada. Outro fator que atua fortemente no erro é a qualidade ruim da câmera de teste, que contribui para medidas imprecisas na régua visual.

Pode ser visto também que nas células radiais da janela "raw" as distâncias medidas foram mais inconsistentes comparadas com as da janela "undistorted". Essa diferença ocorre pois há uma correção na posição dos pixels na segunda janela, que influencia principalmente as medidas feitas longe do centro (onde ocorre maior distorção das lentes).

Outra característica marcante é que aumentando a distância do objeto à câmera, as medidas foram se tornando mais precisas e se ajustando melhor à dimensão real esperada. Uma possível explicação para esse efeito é que menos pixels representam uma distância maior quando o objeto está longe da câmera. Assim, quando o usuário vai manusear a régua, o erro de poucos pixels já resulta em grandes diferenças no tamanho real estimado, melhorando, nesse

caso, a precisão do experimento. Essa comparação pode ser vista nas duas imagens mostradas a seguir, que mostram a diferença das medidas do mesmo objeto à distâncias diferentes.

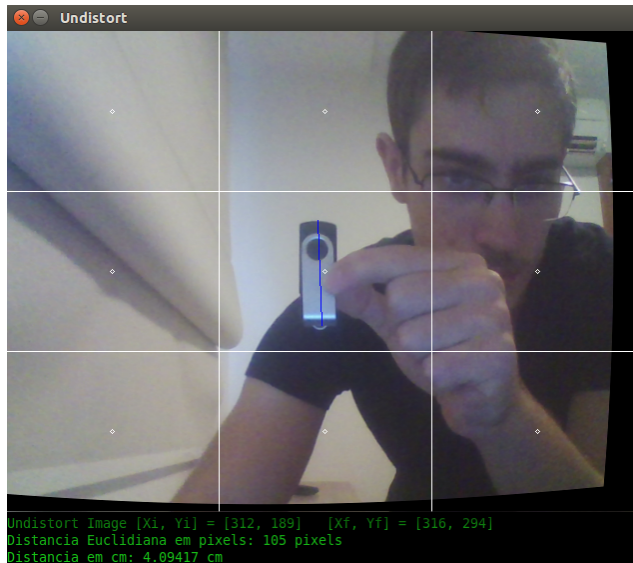


Figura 3. Objeto à 30 cm de distância



Figura 4. Objeto à 300 cm de distância

5. Conclusão

Ficou claro como deve ser procedido para calibrar uma câmera e obter os seus parâmetros intrínsecos, que possibilitam transformar medidas de uma imagem para o mundo real. Além disso, entendeu-se como ocorre distorção na imagem devido às lentes (principalmente em regiões longe do centro), assim como as equações que permitem essa correção.

Após os testes feitos na aplicação da régua visual, observou-se que houve um erro de 10% em média, o que foi considerado uma boa estimativa das medidas reais estimadas, já que foi levado em conta a má qualidade da câmera, os erros de medida do usuário e fatores aleatórios na captura da imagem, como luminosidade e vibrações na câmera. Outro fator que confirma a satisfatoriedade do experimento é que o objeto de teste era pequeno, então pequenas variações na medida geram erros percentuais altos.

Referências

- [1] G. Bradski and A. Kaehler, *Learning OpenCV*, 1st ed. O'Reilly.
- [2] Forsyth and Ponce, *Computer Vision - A Modern Approach*, 1st ed. Pearson.
- [3] OpenCV API Reference. (2017, 31 Março). [Online]. Available: <http://docs.opencv.org/2.4/modules/refman.html>

[1] [2] [3]