

Relatório do Projeto Demonstrativo 2: Calibração de Câmeras

Aluno: Jessé Barreto de Barros
Matrícula: 17/0067033
Sistemas Mecatrônicos - PPMEC-UnB
Disciplina: Visão Computacional
Turma: A
Data: 22/04/2017

1. Objetivos

O principal objetivo deste projeto demonstrativo é a avaliação dos aspectos envolvidos na calibração de câmeras e a realização de uma "regua visual" utilizando as imagens provenientes de uma câmera digital calibrada.

2. Introdução

2.1. Câmeras e Modelo de uma Câmera *Pin-Hole*

Uma câmera funciona mapeando objetos presentes em um sistema de coordenadas tridimensional para um sistema de coordenadas bidimensionais (uma imagem). Esse processo é representado nas Figuras 1 e 2.

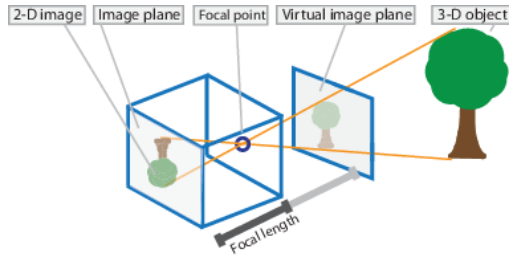


Figura 1. Funcionamento de uma câmera *pinhole* [1].

A câmera utilizada é modelada através do modelo de projeção de uma câmera *pin-hole*.

Nesse modelo a câmera possui apenas uma pequena entrada para que a maioria dos raios de luz sejam bloqueados e com isso reduz o efeito de borrramento devido à exposição de muita luz no sensor *CMOS* da câmera digital utilizada.

Além dessa pequena entrada a câmera também possui uma lente que é responsável por focalizar raios paralelos em um ponto focal no interior da câmera e com isso aumentando o foco de objetos que estão presentes na imagem.

O modelo matemático de uma câmera é apenas um modelo para representar a projeção de objetos tridimensionais no sistema de coordenadas global O_w em um imagem que

possui um sistema de coordenadas bidimensional O_i dado as características óticas da câmera.

Nesse modelo um ponto em O_w com distância relativa à câmera representado por $p_c = [x \ y \ z]^T$ é projetado no plano da imagem na posição em pixels $[u \ v]^T$ pertencente à O_i . Essa projeção é representada pelas equações 1, 2 e 3.

$$\begin{bmatrix} \lambda_x & u_d \\ \lambda_y & v_d \end{bmatrix} = \begin{bmatrix} f_x & 0 \\ 0 & f_y \end{bmatrix} + \begin{bmatrix} c_x \\ c_y \end{bmatrix} \begin{bmatrix} \frac{x}{z} \\ \frac{y}{z} \end{bmatrix} \quad (1)$$

$$\begin{aligned} u &= u_{d_{radial}}(1 + k_1 r^2 + k_2 r^4) \\ v &= v_{d_{radial}}(1 + k_1 r^2 + k_2 r^4) \end{aligned} \quad (2)$$

$$\begin{aligned} u &= u_{dt} + [2p_1 u_{dt} v_{dt} + p_2(r^2 + 2u_{dt}^2)] \\ v &= v_{dt} + [p_1(r^2 + 2v_{dt}^2) + 2p_2 u_{dt} v_{dt}] \end{aligned} \quad (3)$$

Nas equações 1, 2 e 3, os parâmetros intrínsecos da câmera são: (a) distância focal da câmera ($f_x \ f_y$), (b) posição do pixel central na imagem ($c_x \ c_y$), (c) as correções das distorções radiais e tangenciais, respectivamente, os parâmetros de distorções (c) a distorção radial ($k_1 \ k_2 \ p_1 \ p_2$) e (d) valores de escala que relacionam pixels para a unidade espacial utilizada ($\lambda_x \ \lambda_y$).

Os parâmetros extrínsecos são a orientação da câmera em relação ao sistema de coordenadas global O_w .

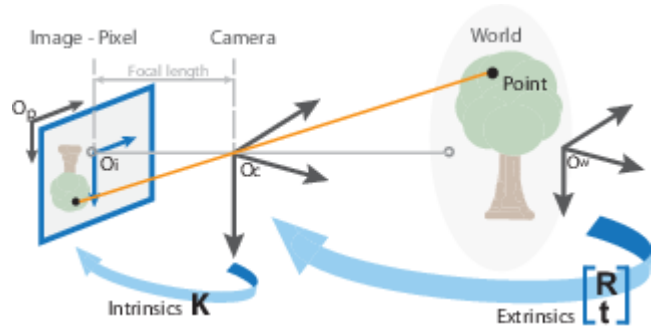


Figura 2. Projeção de ponto presente em O_w para plano da imagem O_i [1].

2.2. Calibração de uma Câmera

A calibração de uma câmera consiste na identificação dos seus parâmetros intrínsecos através de um algoritmo que através de um conjunto de imagens de um objeto que possua padrões conhecidos, como, por exemplo, um tabuleiro de xadrez, em diferentes posições e orientações consiga estimar esses parâmetros.

Se necessário, através de técnicas para a movimentação de corpos rígidos é possível obter os parâmetros extrínsecos da câmera, que é basicamente a sua posição e orientação no sistema de coordenadas global.

Um exemplo de algoritmo de calibração de imagem utilizado é o presente no *OpenCV* que é baseado nos algoritmos [3] e [2].

3. Metodologia

O projeto foi dividido em quatro requisitos:

- **Requisito 1:** Desenvolver uma aplicação em *OpenCV* que a partir da utilização de uma entrada (clique de mouse), seja capaz de capturar a posição inicial $(x_i \ y_i)$ (primeiro clique de mouse) e a posição final $(x_f \ y_f)$ (segundo clique do mouse) e permita desenhar uma linha na tela e ainda, calcule o comprimento desta linha em *pixels*. O comprimento deverá ser dado pela distância Euclidiana Bidimensional, ou seja, $\sqrt{(x_f - x_i)^2 + (y_f - y_i)^2}$.
- **Requisito 2:** Realização da calibração de uma câmera digital gerando arquivos XML com os parâmetros intrínsecos e distorções provenientes do processo de calibração. Repetindo este requisito 5 vezes e obtendo a média desses resultados da calibração.
- **Requisito 3:** Após a calibração e utilizando a imagem de vídeo das saídas "raw" e "undistorted" definidas pela aplicação e utilizando os quadrados presentes em um tabuleiro de xadrez, pois os mesmos possuem dimensões conhecidas:
 1. Dividir a imagem do vídeo em um *grid* de 9 células para orientar a posição dos quadrados do tabuleiro de xadrez em diferentes regiões da imagem e posicionar o tabuleiro de xadrez com distâncias de 15 cm, 80 cm, 150 cm e 300 cm.
 2. Medir em *pixels* as dimensões desses quadrados nessas diferentes posições e obtendo o valor médio dessa medição para a imagem retificada e para a imagem original.
 3. Fazer uma tabela desses resultados para as células com o tabuleiro nas diferentes posições. Para as imagens calibradas e para as imagens originais.

- **Requisito 4:** Apresentar os resultados e compará-los para cada uma das regiões nas imagens originais e calibradas e indicar possíveis melhorias.

3.1. Requisito 1

Para o requisito 1 foi implementada uma aplicação em C++ utilizando o *OpenCV* que abre uma imagem padrão, *lena512.jpg*, que deve estar armazenada junto com os binários da aplicação, ou o caminho de uma segunda imagem como parâmetro da execução do binário via terminal.

3.2. Requisito 2

Para o requisito 2 foi implementada uma aplicação em C++ utilizando o *OpenCV* que através da saída de vídeo de uma câmera realiza a calibração da mesma 5 vezes e salva os parâmetros de distorção e intrínsecos em arquivos XML. Caso o usuário informe via terminal para a execução do binário o argumento `--just-undistort-it` a aplicação carrega os dados provenientes de outras calibrações, calcula a média e o desvio padrão desses parâmetros e os salva em um arquivo XML separado que possa ser utilizado nos próximos requisitos.

Além disto, mostra as imagens originais e calibradas da saída de vídeo e desenha um grid para melhorar visualizar os efeitos visuais da calibração sobre a imagem.

3.3. Requisito 3

Para o requisito 3 foi implementada uma aplicação em C++ utilizando o *OpenCV* que carrega em memória as médias dos parâmetros obtidos com as calibrações e mostra em tela a imagem original e a imagem calibrada.

A aplicação recebe 3 desenhos de linhas para cada célula de cada imagem e salva essas linhas com o intuito de obter o comprimento dessas linhas para cada região da imagem. Esse processo é repetido para cada uma das quatro distâncias (15 cm, 80 cm, 150 cm e 300 cm).

3.4. Requisito 4

Todos os dados obtidos nos requisitos anteriores foram compilados para a comparação e análise dos resultados.

4. Resultados

4.1. Requisito 1

Para o requisito 1 a imagem padrão (*lena.jpg*) foi utilizada para coletar diferentes pontos iniciais e finais para desenhar diferentes linhas na imagem e verificar o comprimento de cada uma dessas linhas. Na Figura 3 é possível ver a execução do programa e o desenho dessas linhas vermelhas e a impressão dos seus comprimentos no terminal.



Figura 3. Imagem colorida (lena.jpg) mostrando os resultados do requisito 1. É possível visualizar as linhas em vermelho que foram adicionadas e o comprimento das mesmas no terminal.

4.2. Requisito 2

Para o requisito 2 utilizou-se de algoritmos para identificar um tabuleiro de xadrez na imagem (*findChessboardCorners()*) e com isso estimar os parâmetros da câmera com diversas imagens desse tabuleiro de xadrez. Durante o processo de calibração as imagens do vídeo são visualizadas em duas janelas uma que contém as imagens diretas do *stream* e a outra mostra as capturas da imagem tentando buscar um tabuleiro de xadrez a cada 30 frames.

A cada 25 frames quando a janela de captura é atualizada, se um tabuleiro de xadrez consegue ser identificado corretamente na imagem a janela de capturas mostra esse tabuleiro de xadrez e junto com isso mostra a localização das pontas dos quadrados do tabuleiro, conforme Figura 4, caso contrário a janela de captura mostra a imagem da captura má sucedida em escala de cinza, como na Figura 5.

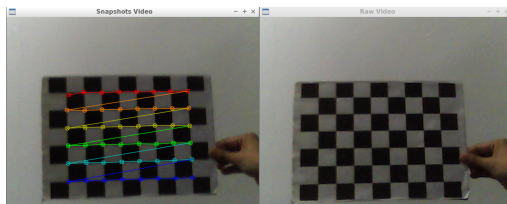


Figura 4. Caso de sucesso na identificação do tabuleiro de xadrez para a calibração da câmera.

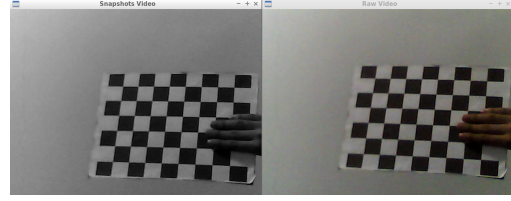


Figura 5. Caso de falha na identificação do tabuleiro de xadrez na imagem para a calibração da câmera.

Após 25 capturas bem sucedidas na identificação do tabuleiro de xadrez e dos pontos das bordas dos seus quadrados a aplicação começa a realizar os cálculos para a calibração da imagem. Se não houver nenhum problema com essa etapa serão apresentados um par de janela com a imagem "raw" e com a imagem "undistorted", Figura 7



Figura 6. Apresentação do resultado da calibração.

Essa etapa é repetida 5 vezes com intuito de obter os parâmetros da câmera com maior confiabilidade. Os resultados dessas calibrações geraram os parâmetros presentes nas tabelas 1 e 2.

k_1	$-0.314443201 \pm 0.0765057206$
k_2	$9.11147118 \pm 10.6450605$
p_1	$-0.0113758519 \pm 0.00301586115$
p_2	$0.00449113920 \pm 0.00220187241$

Tabela 1. Tabela com os dados de distorção da câmera.

f_x	$1737.13745 \pm 481.915649$
f_y	$1728.73950 \pm 473.786957$
c_x	$368.850861 \pm 7.76539803$
c_y	$209.627579 \pm 28.1807537$

Tabela 2. Tabela com os dados intrínsecos da câmera.

4.3. Requisito 3

Para o requisito 3 a aplicação carrega a média dos parâmetros obtidos no requisito 2 e apresenta nas janelas "raw" e "undistorted", as imagens originais e corrigidas com a calibração, respectivamente.

No entanto a imagem original antes de ser retificada passa por um método para desenhar um *grid* de 9 quadrados nela, conforme a Figura 7.

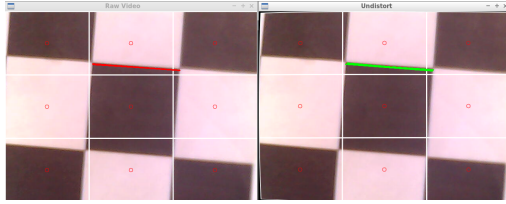


Figura 7. Desenho dos *grids* sobre a imagem original e sobre a imagem retificada.

Para cada uma das células desse *grid* foi medido o tamanho em pixels de um dos quadrados do tabuleiro de xadrez que possui 3 cm de lado cada.

O processo foi repetido novamente para cada uma das distâncias foi efetuada com a média dos comprimentos dos quadrados do tabuleiro para cada distância e células. As Tabelas 3 e 4 apresenta esses dados para as imagens originais e para as imagens retificadas com os parâmetros da câmera.

5. Discussão e Conclusões

O projeto demonstrativo concluiu os objetivos propostos e com esse projeto foi possível aprender mais sobre o processo de calibração de câmeras digitais para a obtenção dos parâmetros da câmera.

Com o requisito 2 foi possível verificar os efeitos das diferentes distorções (radial ou tangencial) sobre uma imagem e também verificar empiricamente a sensibilidade do processo de calibração sobre fatores externos como a iluminação ou a movimentação do tabuleiro não cobrir toda a imagem. Esses fatores geram parâmetros de calibração errados que distorcem erroneamente a imagem.

Com o requisito 3 foi possível verificar que os efeitos de distorção afetam mais células nas bordas da imagem e esse efeito pode ser observado inclusive comparando as Tabelas 3 e ???. No entanto, nesse experimento percebe-se que devido a uma pequena inclinação do tabuleiro de xadrez durante as diferentes capturas de dados percebe-se que as células superiores, ou seja, células com índices menores, são menores do que as células inferiores e esse efeito é amplificado com o fator de escala.

No requisito 3 é evidente a dependência do fator de escala entre o tamanho de um objeto com um objeto de tamanho conhecido devido a distância entre a lente da câmera e o objeto. Percebe-se que conforme essa distância aumenta o tamanho de um mesmo objeto em pixel é reduzido.

Referências

- [1] Mathworks - documentation: What is camera calibration? <https://www.mathworks.com/help/vision/ug/camera-calibration.html>. acessado em 20/11/2016.
- [2] J.-Y. Bouguet. Matlab calibration tool, 2015.
- [3] Z. Zhang. A flexible new technique for camera calibration. *IEEE Transactions on pattern analysis and machine intelligence*, 22(11):1330–1334, 2000.

Índice	15 cm	80 cm	150 cm	300 cm
Célula ₁	229.7 ± 0.8	39.72 ± 0.8	16.05 ± 0.6	11.07 ± 0.07
Célula ₂	229.3 ± 0.4	46.43 ± 1.0	24.02 ± 1.3	12.34 ± 0.4
Célula ₃	222.4 ± 1.0	39.00 ± 0.6	22.67 ± 0.4	12.66 ± 0.4
Célula ₄	245.3 ± 0.9	47.50 ± 0.8	27.70 ± 1.1	12.69 ± 0.4
Célula ₅	235.6 ± 1.1	51.36 ± 1.5	25.70 ± 1.1	12.34 ± 0.8
Célula ₆	235.8 ± 1.7	49.03 ± 1.9	20.01 ± 1.3	12.38 ± 1.0
Célula ₇	241.4 ± 3.5	49.01 ± 0.6	24.68 ± 0.4	13.35 ± 0.4
Célula ₈	238.0 ± 0.7	51.01 ± 1.3	24.68 ± 1.1	11.75 ± 0.4
Célula ₉	233.7 ± 1.1	48.01 ± 0.6	22.02 ± 1.3	13.34 ± 1.5

Tabela 3. Tabela com os dados da média dos quadrados do tabuleiro para cada célula para diferentes distâncias para imagem original.

Índice	15 cm	80 cm	150 cm	300 cm
Célula ₁	225.9 ± 1.5	41.81 ± 1.7	20.12 ± 1.3	11.70 ± 1.1
Célula ₂	226.3 ± 1.1	46.68 ± 0.4	23.70 ± 0.4	12.35 ± 0.9
Célula ₃	219.8 ± 0.9	40.33 ± 1.1	21.68 ± 4.2	11.36 ± 1.5
Célula ₄	244.0 ± 0.6	49.45 ± 0.8	25.36 ± 0.9	13.73 ± 0.8
Célula ₅	235.3 ± 2.2	49.02 ± 1.3	24.40 ± 0.4	13.34 ± 0.4
Célula ₆	235.0 ± 3.3	48.01 ± 1.3	23.01 ± 1.3	14.05 ± 0.6
Célula ₇	241.0 ± 1.3	47.71 ± 1.5	24.36 ± 1.8	12.70 ± 0.4
Célula ₈	241.3 ± 0.8	52.69 ± 1.0	25.03 ± 0.6	13.69 ± 0.4
Célula ₉	235.7 ± 1.7	53.02 ± 0.6	23.07 ± 1.9	13.34 ± 1.5

Tabela 4. Tabela com os dados da média dos quadrados do tabuleiro para cada célula para diferentes distâncias para a imagem retificada.