

Projeto Demonstrativo 2 - Calibração de Câmeras

Lukas Lorenz de Andrade
Departamento de Ciências da Computação
UnB

Matrícula: 16/0135109
lukaslorenzdeandrade@gmail.com

Resumo—Ao se tratar de visão computacional, o *framework* OpenCV se destaca pela sua versatilidade e suporte para várias arquiteturas. Neste trabalho, por meio daquela, será desenvolvido um programa capaz de medir a distância entre dois pontos no mundo real (3D) a partir da calibração da câmera e das distâncias entre os respectivos pixels (em tempo real). Os resultados do processo de calibração são salvos em arquivos de extensão xml, cujos parâmetros extrínsecos da câmera são comparados por meio de dois métodos distintos propostos pela OpenCV, além de uma análise qualitativa dos dados. O trabalho representa o potencial desse *framework* e sua capacidade de resolver problemas neste quesito, propiciando excelentes soluções de fácil implementação.

Index Terms—Visão Computacional, Calibração de Câmeras, Régua Virtual, OpenCV 3.4.4, Python 3.5.

I. INTRODUÇÃO

A necessidade por automatização de processos propiciou a solução dos problemas encontrados por meio de novas tecnologias que substituíssem a intervenção humanas. Assim, não só para análises estatísticas, mas também para tomadas de decisão ou alerta, passou-se a usar câmeras como sensores para diversas funcionalidades, a saber: predição de movimento, georeferenciamento, análise de situações, robôs (drones) autônomos, entre outros.

Ademais, devido a tal necessidade, o surgimento de plataformas/*frameworks* que apresentavam base para os problemas enfrentados se tornou crucial para o desenvolvimento de novas tecnologias. Para tanto, as bibliotecas fornecidas pela OpenCV fornecem um aparato para diversas linguagens de programação quando se trata de processamento de imagens e vídeos, visão computacional (uso dessas imagens para a atuação em processos^[1]), aprendizado de máquina e fotografia computacional.

Além disso, devido a crescente demanda por soluções inteligentes pelo uso de redes neurais e métodos de inteligência artificial tanto para a classificação, predição, realidade aumentada ou para reconstrução 3D, o domínio dessa ferramenta vem sendo cada vez mais requerido. Tal fator se dá pela maioria dos *frameworks* usados nessas áreas, como Caffe, Keras, entre outras, usarem a OpenCV para o processamento e os métodos de visão computacional na parte do tratamento dos dados.

Dessa forma, este trabalho consiste na implementação de uma régua virtual que será explicitado nas sessões II, III e IV. Na sessão II, será apresentada a metodologia usada para

adquirir o resultado final de cada etapa, mostrado na sessão III, junto com a análise dos dados obtidos. Por fim, na sessão IV será apresentada as considerações finais dados os parâmetros assumidos, assim como os resultados obtidos.

^[1] Por processos entende-se qualquer sistema físico ou computacional (*Software*) que realize uma tarefa.

A. Objetivos

Os objetivos do trabalho se resumem a medição de objetos utilizando-se apenas dos parâmetros estipulados da câmera, em contraposição da medição por um objeto de referência. Entretanto, os objetivos específicos do programa se resumem a:

(1) Desenvolva 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. Utilize uma função de chamada ao sistema pelo mouse, por exemplo: OpenCV. Calcule o comprimento desta linha em pixels e imprima-o no console. O comprimento deverá ser dado pela distância Euclidiana bidimensional.

$$d(P1, P2) = \sqrt{(x_i - x_f)^2 + (y_i - y_f)^2} \quad (1)$$

(2) Realize a calibração de uma câmera digital, gerando os arquivos "XML" com os parâmetros intrínsecos e distorções provenientes do processo de calibração. Utilize para o desenvolvimento da aplicação de forma mais simples, o programa de calibração disponibilizado no tópico da atividade, incluindo o padrão de calibração. A partir da saída do vídeo na janela "raw" e "undistorted" desta aplicação, acople-a ao código da aplicação desenvolvida no Requisito 1 de forma que seja possível medir distâncias em pixels nessas duas janelas.

(3) Meça, com uma boa régua, os retângulos do padrão de calibração impresso e use esse padrão para determinar o sistema de coordenadas do mundo (world coordinate frame). Armazene as coordenadas reais dos 48 pontos do padrão. Assuma que o sistema de coordenadas do mundo tenha a origem (0,0,0) no primeiro ponto da página (i.e., primeira interseção entre quatro retângulos) e que o papel esteja alinhado com as coordenadas X e Y (ou seja, Z=0 para todos os pontos do padrão de calibração).

Implemente um programa que use as medidas acima, bem como os dados de calibração obtidos para o requisito 2,

para calcular os parâmetros extrínsecos da câmera. Repita esse processo de calibração de extrínsecos com o padrão de calibração situado a pelo menos 3 distâncias diferentes do centro da camera:

- dmin: o mais próximo possível, desde que todos os 48 pontos sejam visíveis;
- dmax: o mais distante possível, desde que esses pontos possam ser capturados pelo algoritmo;
- dmed: e numa distância intermediária.

Ao realizar tais calibrações, meça com uma trena a distância entre o centro da câmera e a origem do padrão de calibração e use isso para avaliar os resultados de calibração obtidos, ou seja, verifique o quão similar é essa distância e a norma do vetor de translação do mundo para a câmera — t —. Anote seus resultados e discuta o efeito da distância no calculo dos parâmetros de translação. Para cada posição acima (dmin, dmed e dmax), repita o processo pelo menos 3 vezes para poder calcular a média e o desvio padrão da distância entre a coordenada do mundo e a da câmera. Esse desvio padrão informa quão confiáveis são as estimativas.

(4) Terminado o processo de calibração acima e utilizando a imagem de vídeo das saídas "raw" e "undistorted" definidas pela aplicação e algum objeto com dimensão (altura ou largura) conhecida faça, para cada posição (dmin, dmed e dmax) as seguintes operações:

- Meça o objeto conhecido usando o procedimento do requisito 1 em ambas imagens (raw e undistorted);
- Com cliques de mouse, você obterá medidas em pixels em ambas imagens. Usando os parâmetros de calibração obtidos, mapeie essas medidas para metros;
- Anote os resultados obtidos (medidas, em metro) do objeto para cada distância (dmin, dmed e dmax), e também usando medidas feitas nas imagens raw e undistorted. Anote também o valor real do tamanho do objeto (conforme medido, por exemplo, com uma régua);
- Repita o processo acima várias vezes para cada distância da câmera, adquirindo medições do objeto situado próximo ao centro da imagem e também próximo a uma das bordas da imagem;
- Faça tabelas consolidadas (já utilizando os valores médios e desvio padrão) com estes resultados, como indicado abaixo.

Tabela I
EFEITOS DA DISTÂNCIA NA MEDIÇÃO

Posição do Padrão de calibração	dmin	dmed	dmax
$ t $ medida pela trena			
$ t $ medida pela calibração extrínseca			
$l_{raw,centre}$			
$l_{raw,perifery}$			
$l_{undistort,centre}$			
$l_{undistort,perifery}$			

Onde $|t|$ é a distância da câmera até o objeto e l é dimensão medida do objeto.

(5) Discuta os principais elementos causadores das variações das medições obtidas com a esperada (medida real do objeto

avaliado). Avalie a característica do hardware da câmera e quais fatores podem ser os causadores destas possíveis diferenças. Proponha/discuta quais fatores podem ser melhorados/aprimorados para que esta medição seja feita com a maior precisão possível.

II. METODOLOGIA

O projeto foi desenvolvido usando o *framework* OpenCV 3.4.1 na linguagem Python 3.5. Assim, a metodologia consiste na elaboração de um programa e as medições do padrão de calibração conforme o descrito nos itens (3) e (4) da sessão anterior.

A. Programa Desenvolvido

Para cumprir o primeiro requisito, desenvolveu-se uma função que abre a câmera, por meio de funções da OpenCV, e com um duplo clique de mouse, pega-se as respectivas coordenadas dos dois pontos escolhidos. Após isso, calcula-se a distância euclidiana (1) entre os dois pontos. Fez-se tal função o mais genérica possível a fim de se reaproveitar.

Para o cumprimento dos requisitos (2) e (3), realizou-se a calibração da câmera conforme o tutorial [1], calculando-se os parâmetros extrínsecos e intrínsecos pela função `cv2.calibrateCamera` e os extrínsecos pela função `cv2.solvePnP`, comparando-os para as calibrações requeridas (5 vezes por distância $|t|$ e a três diferentes distâncias $|t|$). Salvando os resultados em arquivos XML.

Por fim, para o cumprimento do requisito (4), utilizou-se as fórmulas do material de apoio, representadas em 2, para estipular os valores das coordenadas reais do objeto e medir as distâncias (para o caso *undistorted*), e as equações apresentadas em [3] (para o caso *raw*).

$$\begin{bmatrix} x_p \\ y_p \end{bmatrix} = (1+k_1r^2+k_2r^4+k_3r^6) \begin{bmatrix} x_d \\ y_d \end{bmatrix} + \begin{bmatrix} 2p_1x_dy_d + p_2(r^2 + 2x_d^2) \\ p_1(r^2 + 2y_d^2) + 2p_2x_dy_d \end{bmatrix} \quad (2)$$

B. Como Executar o Programa

A fim de se facilitar o uso do programa, alterou-se o permissãoamento do arquivo para que se tornasse um compilável, de forma que sua execução se dá por `./main.py` no diretório dos arquivos enviados.

III. ANÁLISE DO PROJETO

Nesta sessão serão apresentados os resultados obtidos conforme cada critério apresentado na sessão I.A.

A. Cálculo da Distância Entre Pixels

Os resultados apresentados no terminal da figura III-A foram adquiridos clicando-se nas pupilas dos dois olhos, de forma a ser calculada a distância entre pixels e mostrada a distância real, cujo resultado é igual a zero por ainda não ser parte deste requisito.

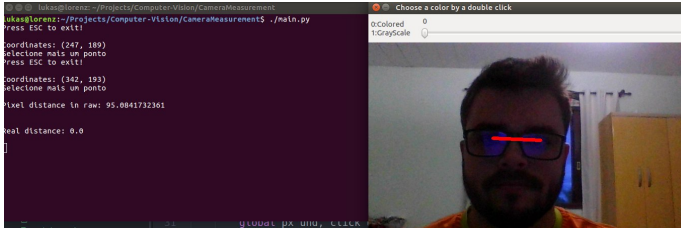


Figura 1. Resultado da distância entre pixels de um olho a outro.

B. Calibração

A fim de se otimizar o processo, a função *calibration* calcula os parâmetros intrínsecos e extrínsecos da câmera, sendo estes calculada em dois métodos e apresentada no terminal para avaliação, conforme mostrada na figura III-B.

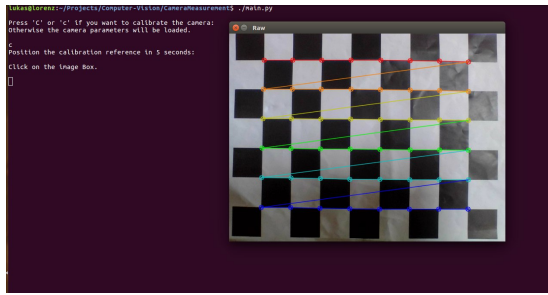


Figura 2. Resultado da detecção do padrão de calibração na menor distância (266mm).

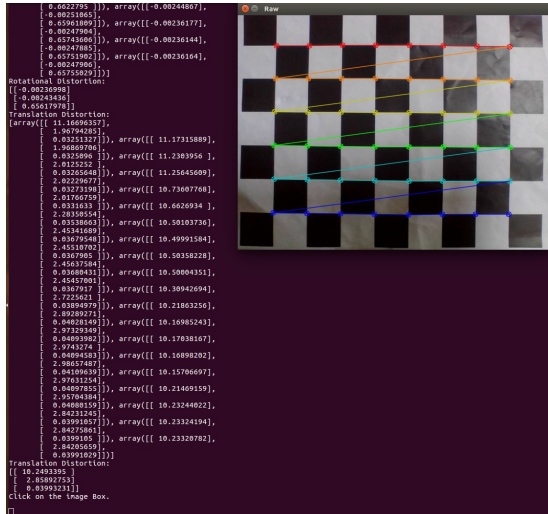


Figura 3. Resultado do processo de comparação entre os dois métodos de calibração de extrínsecos, dado as condições expressas na figura III-B. Percebe-se que os formatos gerados pelas funções diferem mas são coerentes para os cálculos de distância. No entanto, preferiu-se os resultados gerados pela *cv2.solvePnP*.

Apesar das médias dos parâmetros não serem expressas no terminal, são calculadas e armazenadas no arquivo *Camera-Parameters** para cada distância $|t|$.

C. Régua Visual

Neste requisito, escolheu-se uma caixa retangular com as dimensões 215x88mm para ser o objeto a ser mensurado. Para tanto, os pixels obtidos por meio de um duplo clique na imagem *undistort* são transformados em coordenadas reais por meio da equação 2 e sua distância calculada pela métrica euclidiana (equação 1).

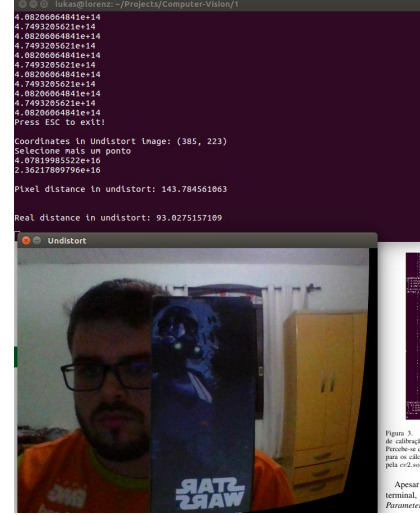


Figura 4. Resultado da largura do objeto a 260mm.

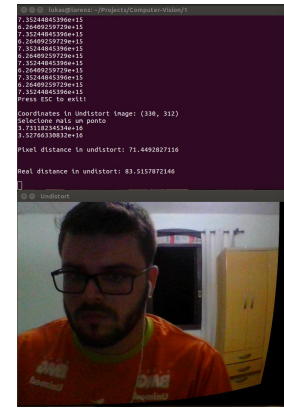


Figura 5. Resultado da largura do objeto a 530mm.

Tabela II
EFEITOS DA DISTÂNCIA NA MEDIÇÃO EM MM

Posição do Padrão de calibração	dmin	dmed	dmax
$ t $ medida pela trena	260	530	970
$ t $ medida pela calibração extrínseca			
$l_{raw,centre}$			
$l_{undistort,centre}$	93.02	83.50	74.89
$l_{undistort,periphery}$	115.40	201.69	250.90

IV. CONCLUSÃO

Com base nas informações da sessão anterior, tem-se que o processo de calibração se mostrou relativamente satisfatório,

de forma que para melhorar os resultados obtidos se daria com maior precisão caso a calibração fosse realizada de forma mais precisa, ou seja, que se conseguisse garantir a consistência do eixo óptico da câmera para cada $|t|$ realizada.

Dentre os parâmetros de Hardware que influenciam na medição são: a distância focal, a resolução ou densidade de pixels/mm e os efeitos de distorção em borda (distorção radial da câmera, de forma que a medida que o objeto se distancia do centro, mais distorcido ele se encontra). Contudo, para minimizar tais efeitos, buscou-se tirar médias dos parâmetros extraídos da calibração (armazenados nos arquivos ***Camera-Parameters*.xml***, onde o índices 0, 1 e 2 são respectivos as distâncias $|t_{min}|$, $|t_{med}|$ e $|t_{max}|$), de forma que o erro tende a manter-se fixo.

REFERÊNCIAS

- [1] https://docs.opencv.org/3.4.1/d6/d00/tutorial_py_root.html, Acessado em 25 de Abril de 2018.
- [2] <https://github.com/lorenz-lukas/Computer-Vision>, Acessado em 25 de Abril de 2018.
- [3] <https://stackoverflow.com/questions/42035721/how-to-measure-object-size-in-real-world-in-terms-of-measurement-like-inches-cm>, Acessado em 25 de Abril de 2018