MESTRADO EM ENGENHARIA E DE COMPUTADORES

Relatório de Visão por Computador

Geometric Camera Calibration

Duarte de Sousa Cruz, 2017264087

Universidade de Coimbra
04 de Maio de 2022

1 Introdução

Neste trabalho vamos calibrar uma câmara. Usando o método geométrico de calibração para calcular os parâmetros da câmara e estimar os coeficientes de distorção das lentes. Para isto vamos usar três métodos diferentes.

- Direct Linear Transform algorithm (DLT)
- Gold Standard algorithm
- Gold Standard algorithm with radial distortion estimation

2 Pontos

Como primeiro passo correspondemos os pontos da imagem (x,y) a pontos (x,y,z). Tivemos que escolher pelo menos 6 pontos e armazenar em dois ficheiros de dados, para não ter que repetir este processo sempre que iniciávamos o programa.

3 Data normalization

A normalização da data é essencial para estes algoritmos e vai ser a base para todas as implementações deste projeto. Para isto temos que aplicar uma matriz transformação para os pontos 2D da imagem (T) e para os pontos 3D (U):

$$T = \begin{bmatrix} s_2 D & 0 & C_x \\ 0 & s_2 D & C_y \\ 0 & 0 & 1 \end{bmatrix} \tag{1}$$

$$U = \begin{bmatrix} s_3 D & 0 & 0 & C_x \\ 0 & s_3 D & 0 & C_y \\ 0 & 0 & s_3 D & C_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2}$$

que consiste numa translação e num escalonamento para que a centroide dos novos pontos seja em $(0,0)^T$ e a sua distância média para a origem seja $\sqrt{2}$ para os pontos da imagem e $\sqrt{3}$ para os pontos 3D.

As equações para s, C_x e C_y são:

$$C_x = -s\overline{x} \tag{3}$$

$$C_y = -s\overline{y} \tag{4}$$

$$s = \frac{\sqrt{2}}{\frac{1}{n}\sum_i \sqrt{(x_i - \overline{x})^2 + (y_i - \overline{y})^2}} \tag{5}$$

sendo $\overline{x}$ e $\overline{y}$ os seus valores médios.

4 Direct Linear Transform

Este algoritmo serve para estimar a matriz da câmara dando os pontos correspondentes. O objetivo é minimizar o erro algébrico do sistema linear do modelo da câmara.

Em primeiro lugar, começamos por normalizar os dados dos pontos xy e XYZ. Depois criamos a matriz A para todos os pontos n, que resulta em uma matriz 2nX12 onde A.M=0.

$$\begin{bmatrix} w_i P_i^T & 0^T & -x_i P_i^T \\ 0^T & w_i P_i^T & -y P_i^T \end{bmatrix} \begin{bmatrix} M^1 \\ M^2 \\ M^3 \end{bmatrix} = 0 \tag{6}$$

A solução para o M é o vetor nulo de A que pode ser obtido pela decomposição SVD. A matriz P vai corresponder à ultima coluna de V, que vem da decomposição.

```
A = [];

for i=1:size(xy,2)
    A = [A; XYZ(:,i)' zeros(1,4) -xy(1,i).*(XYZ(:,i))';
         zeros(1,4) XYZ(:,i)' -xy(2,i).*(XYZ(:,i))'];
end

[U, S, V] = svd(A);

P = V(:,end)/V(end, end);
P = reshape(P, 4, 3)';
```

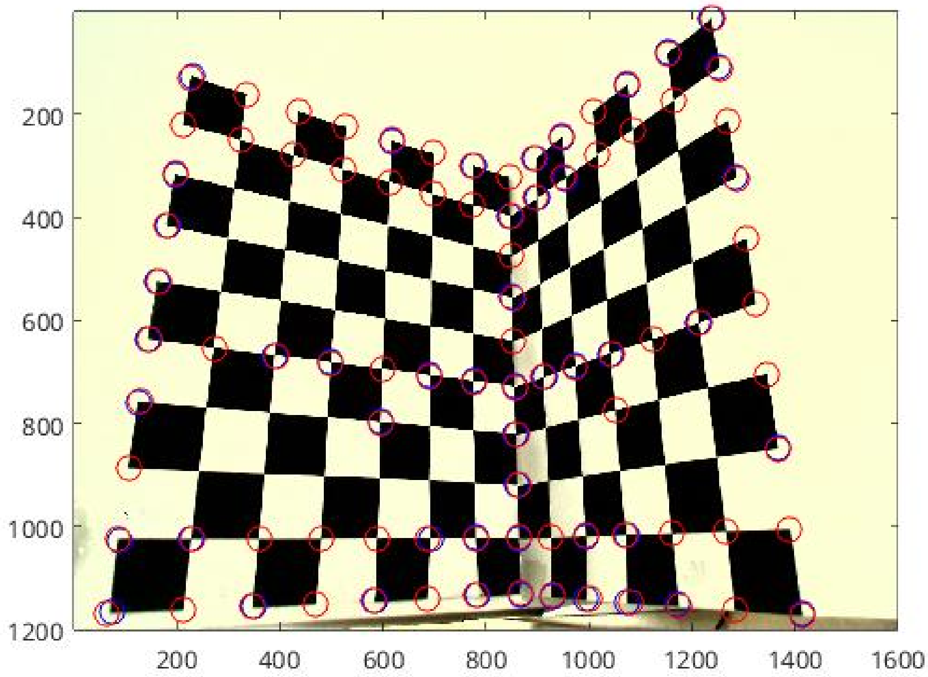

Figure 1: Projeção dos pontos após DLT

5 Camera Matriz Decomposition

Nesta parte da nossa implementação vamos decompor a matriz P em 3 matrizes, K, R e C. Vamos implementar duas decomposições, a decomposição QR e a explicita. Apesar de as composições serem diferentes os resultados devem ser iguais.

5.1 QR-Factorization

A decomposição QR é a decomposição da matriz A no produto $A = QR$, sendo Q a matriz ortogonal e R a matriz upper triangular. O inverso da matriz R vai corresponder à matriz K e o inverso da matriz Q vai corresponder à matriz R.

Para obtermos o centro da câmara C é o ponto onde $PC = 0$, o que significa que conseguimos obter C se fizermos decomposição em valores singulares da matriz P. C vai corresponder à última coluna da matriz V, que vem da decomposição SVD.

```
[U, S, V] = svd(P);
C = V(:,end)/V(end, end);

```

```
[Q, R] = qr(inv(P(1:3,1:3)));

K = inv(R);
R = inv(Q);

D=diag(sign(diag(K)));
K=K*D;
R=D*R;

K=K/K(end,end);
```

5.2 EXPlicit Decomposition

Após estimada a matriz M, o próximo passo vai ser retirar os parâmetros extrínsecos e intrínsecos desta.

$$\mathbf{M} = \begin{bmatrix} \alpha r_1^T - \alpha cot\theta r_2^T + u_0 r_3^T & \alpha t_x - \alpha cot\theta t_y + u_0 t_z \\ \frac{\beta}{sin\theta} r_2^T + v_0 r_3^T & \frac{\beta}{sin\theta} t_y + v_0 t_z \\ r_3^T & t_z \end{bmatrix}$$

$$A = \begin{bmatrix} a_1^T \\ a_2^T \\ a_3^T \end{bmatrix} b = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix} \tag{7}$$

e sabendo que:

$$M \begin{bmatrix} C \\ 0 \end{bmatrix} = 0 \tag{8}$$

Em particular, se M = [A b] então, $C = -A^{-1}b$.

Vamos agora calcular os parâmetros intrínsecos com as fórmulas dadas no formulário.

$$\rho = \frac{1}{|a3|} \tag{9}$$

$$u_0 = \rho^2 (a_1.a_2) \tag{10}$$

$$v_0 = \rho^2 (a_2.a_3) \tag{11}$$

$$cos\theta = \frac{(a_1 * a_2).(a_2 * a_3)}{|a_1 * a_3|.|a_2 * a_3|} \tag{12}$$

$$\alpha = \rho^2 |a_1 * a_3| sin\theta \tag{13}$$

$$\beta = \rho^2 |a_2 * a_3| sin\theta \tag{14}$$

```
rho = epsilon/norm(A(3,:));

cos_t = dot(cross(A(1,:),A(3,:)),cross(A(2,:),A(3,:))) / dot(norm(cross(A(1,:),A(3,:))),\\
norm(cross(A(2,:),A(3,:))));
sin_t = sqrt(1-cos_t^2);

alpha = rho.^2 * norm(cross(A(1,:),A(3,:))) * sin_t;
beta = rho.^2 * norm(cross(A(2,:),A(3,:))) * sin_t;

u0 = rho^2*dot(A(1,:),A(3,:));
v0 = rho^2*dot(A(2,:),A(3,:));
```

Após estes calculos,vamos calcular os parâmetros extrínsecos,

$$r_1 = \frac{(a_1 * a_3)}{|a_2 * a_3|} \tag{15}$$

$$r_3 = \frac{a_3}{|a_3|} \tag{16}$$

$$r_2 = r_3 * r_1 \tag{17}$$

```
r1 = (1/norm(cross(A(2,:),A(3,:))))*cross(A(2,:),A(3,:));
r3 = rho*A(3,:);
r2 = cross(r3,r1);
```

Sabendo todos os parâmetros podemos finalmente calcular as matrizes K, R e t.

$$K = \begin{bmatrix} \alpha & -\alpha cot\theta & u_0 \\ 0 & \frac{\beta}{sin\theta} & v_0 \\ 0 & 0 & 1 \end{bmatrix} \tag{18}$$

$$R = \begin{bmatrix} r_1 \\ r_2 \\ r_3 \end{bmatrix} \tag{19}$$

$$t = \rho K^{-1} b \tag{20}$$

6 Gold Standard Algorithm

Este algoritmo minimiza o erro geométrico $\sum_i d(p_i, \overline{p}_i)^2$, onde p_i são os pontos da imagem e $\overline{p}_i$ são os pontos objeto projetados. $d(p_i, \overline{p}_i)^2$ é a euclidean distance entre p_i e $\overline{p}_i$.

Primeiro vamos normalizar os pontos e correr o algoritmo DLT para ter a matriz P inicial para a otimização. Depois vamos usar o fminsearch para calcular o minimo da soma dos erros de reprojeção entre os pontos clicados no início da implementação e os pontos projetados pela matriz P. Os pontos projetados pela matriz P são dados por:

$$\overline{p}_i = P P_i \tag{21}$$

sendo P_i os pontos 3D escolhidos na primeira fase da calibração.

```
P = [p(1:4);p(5:8);p(9:12)];

for i = 1:size(XYZ,2)
    xy2(i,:)=P*[XYZ(:,i)];
end

for j =1:size(xy2,1)
    xy2(j,1)=xy2(j,1)/xy2(j,3);
    xy2(j,2)=xy2(j,2)/xy2(j,3);
end

for x =1:size(XYZ,2)
    d=sqrt((xy(1,x)-xy2(x,1)).^2+((xy(2,x)-xy2(x,2)).^2));
end

soma = sum(d);
f = soma;
```

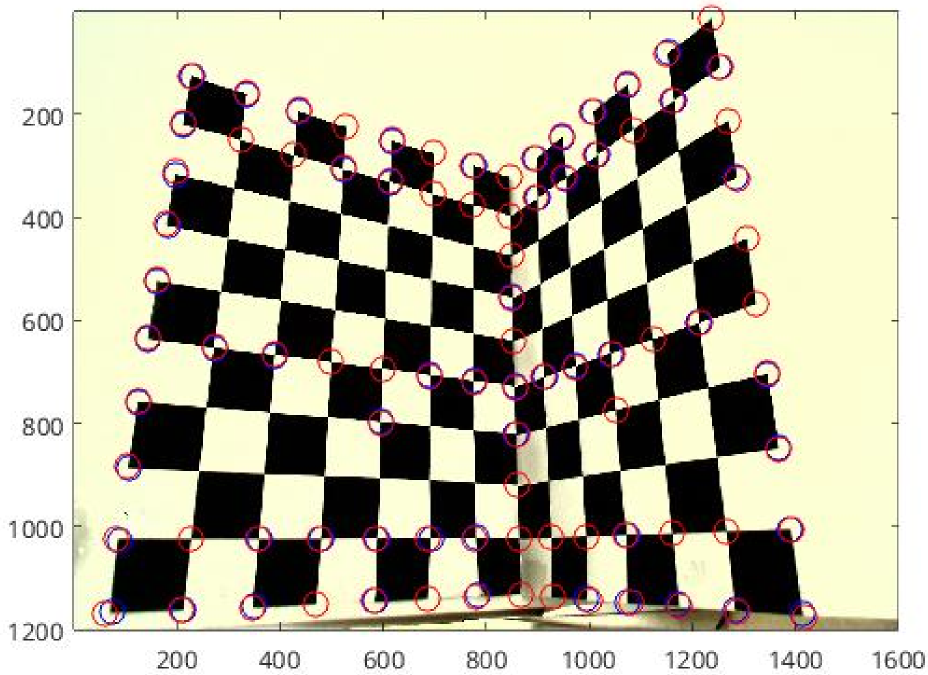

Figure 2: Projeção dos pontos após Gold Standard Algorithm

7 Gold Standard algorithm with radial distortion estimation

Durante a otimização do P os coeficientes de distorção radial das lentes também podem ser estimados.

O modelo de distorção radial é dado por:

$$\begin{bmatrix} x_d \\ y_d \end{bmatrix} = L(\check{r}) \begin{bmatrix} \check{x} \\ \check{y} \end{bmatrix} \tag{22}$$

onde $(\check{x}, \check{y})$ é a projeção linear ideal do ponto 3D P

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = [R| - RC]P \tag{23}$$

$$\check{x} = \frac{x}{z} \check{y} = \frac{y}{z} \tag{24}$$

Para calcular o $L(\check{r})$, temos que primeiro obter o $\check{r}$ que é a distância radial do ponto $(\check{x}, \check{y})$ para o ponto principal da imagem (p_x, p_y).

$$\check{r} = \sqrt{\check{x}^2 + \check{y}^2} \tag{25}$$

Após isto só falta aplicar a expansão de Taylor,

$$L(\check{r}) = 1 + k_1 r^2 + k_2 r^4 \tag{26}$$

sendo $k_1 e k_2$ os coeficientes de distorção radial, que vão ser estimados na função fminsearch.

Como a imagem usada tem pouco distorção é normal os coeficientes de distorção radial serem pequenos. Os valores que obtemos na nossa implementação são $k_1 = 3.8 * 10^-4 e k_2 = 4.0 * 10^-4$

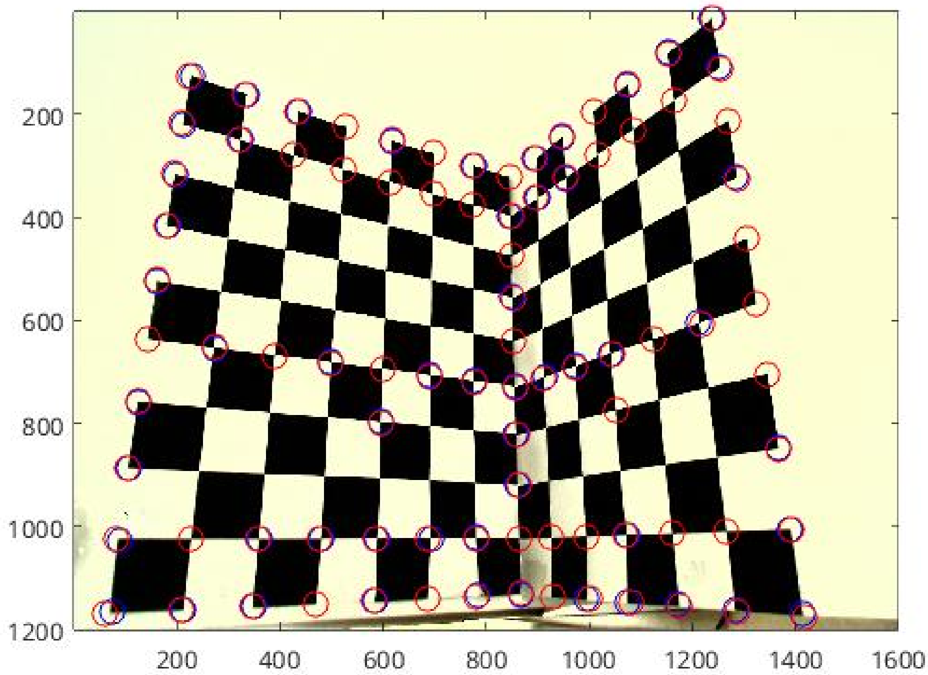

Figure 3: Projeção dos pontos após Gold Standard com estimação da distorção radial

Na imagem abaixo, vemos uma distorção radial admissível.

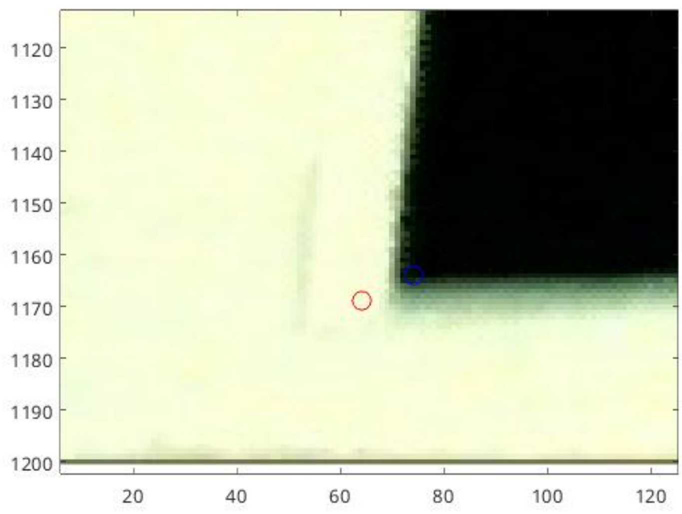

Figure 4: Projeção dos pontos após Gold Standard com estimação da distorção radial

8 Resultados e Observações

Os resultados correram dentro do esperado e conseguimos implementar todas as funções sem contar com a função extra. Conseguimos obter erros dentro dos limites admissíveis.

Observamos que os erros de reprojeção nos algoritmos Gold Standard sem e com estimação de distorção radial foram respectivamente 2.63 2.20. Concluindo que os erros foram bem respeitáveis.