Dimitri Charneux

Rendu tp:

Éléments de géométrie projective et calibration de caméra.

Introduction

Durant ce TP, nous allons voir comment calibrer une caméra grâce à la méthode de Zhengyou Zhang, à partir de plusieurs photographies prises avec un angle de vue différent, des réglages identiques, et avec une mire placée en Z=0.

1. Méthode de Zhang

La méthode de Zhang est un méthode qui consiste à calculer les paramètres intrinsèques et extrinsèques d'une caméra, à partir de plusieurs images d'un plan provenant de la même scène, mais prises avec un angle de vue différent. Il faut également connaître, pour chaque images, les coordonnées du plan dans la scène.

Nous avons tout d'abord calculé la matrice des homographies, qui permet de calculer des transformations linéaires entre deux plans projectifs, en estimant l'homographie entre la mire et la scène dans toutes les images. La méthode calculant cette matrice nous a été donnée, elle utilise les coordonnées homogènes des points scène et image pour calculer la matrice d'homographie en suivant les formules de Zhang.

Nous avons ensuite utilisé cette matrice pour calculer les lignes de contraintes grâce aux méthodes suivantes que nous avons dû écrire à partir des formules de la publication de Zhang :

```
function v=ZhangConstraints(H)
  v = [ZhangConstraintTerm(H, 1, 2); ...
       ZhangConstraintTerm(H, 1, 1) - ZhangConstraintTerm(H, 2, 2)];
endfunction
```

```
function v=ZhangConstraintTerm(H, i, j)
   v = rand(1, 6);
   v = [H(1,i) * H(1,j);
        H(1,i) * H(2,j) + H(2,i) * H(1,j);
        H(2,i) * H(2,j);
        H(3,i)*H(1,j) + H(1,i)*H(3,j) ;
        H(3,i)*H(2,j) + H(2,i) * H(3,j) ;
        H(3,i) * H(3,j)]';
endfunction
```

Ces lignes de contraintes ont été transformées en un vecteur *b* qui nous a permis de calculer les paramètres intrinsèques et extrinsèques de la caméra comme nous allons le voir plus précisément dans les parties suivantes.

Le code de la fonction principale de la méthode de Zhang est fourni en annexe.

Maintenant que nous avons vu le fonctionnement de la méthode de Zhang, nous allons voir plus précisément ce que sont les paramètres intrinsèques d'une caméra et comment les calculer.

2. Paramètres intrinsèques

Les paramètres intrinsèques d'une caméra représentent des paramètres qui lui sont internes. Ces paramètres comprennent les facteurs d'agrandissements de l'image prise par la caméra, les coordonnées de la projection de son centre optique sur le plan image ainsi qu 'un paramètre indiquant l'orthogonalité de ses cellules. Ces paramètres sont propres à chaque caméra et ne varient pas en fonction de sa position dans l'espace de travail.

Pour la suite de ce rapport, nous allons utiliser les notations suivantes :

- α et β représentent les facteurs d'agrandissement de l'images.
- γ représente l'orthogonalité des cellules de la caméra.
- u0 et v0 représentent les coordonnées du centre optique de la caméra.

Pour calculer la matrice intrinsèque de la caméra, nous avons tout d'abord déterminer la valeur des différents paramètres cités précédemment grâce aux formules suivantes :

$$\begin{aligned}
v_0 &= (B_{12}B_{13} - B_{11}B_{23})/(B_{11}B_{22} - B_{12}^2) \\
\lambda &= B_{33} - [B_{13}^2 + v_0(B_{12}B_{13} - B_{11}B_{23})]/B_{11} \\
\alpha &= \sqrt{\lambda/B_{11}} \\
\beta &= \sqrt{\lambda B_{11}/(B_{11}B_{22} - B_{12}^2)} \\
\gamma &= -B_{12}\alpha^2\beta/\lambda \\
u_0 &= \gamma v_0/\beta - B_{13}\alpha^2/\lambda \ .
\end{aligned}$$

Nous avons ensuite mis ces paramètres dans une matrice pour quelle ait la forme suivante :

$$\mathbf{A} = \begin{bmatrix} \alpha & \gamma & u_0 \\ 0 & \beta & v_0 \\ 0 & 0 & 1 \end{bmatrix}$$

Le code Scilab permettant de calculer ces paramètres et de remplir la matrice des paramètres intrinsèques est fourni en annexe. Ce code nous a permis d'obtenir les résultats suivants :

Matrice intrinsèque fournie :

3546.1	0	320
0	3546,1	240
0	0	1

Matrice intrinsèque calculée :

3514,5	-3,15	336,8
0	3520,1	220,1
0	0	1

Comme nous pouvons le voir ci-dessus, les résultats calculés grâce à la méthode de Zhang et les résultats fournis lors du TP sont très proches. Nous pouvons en déduire que la méthode de Zhang est fiable et que nous l'avons implémentée correctement.

Maintenant que nous connaissons les paramètres internes à la caméra, nous allons voir comment calculer ses paramètres extrinsèques.

3. Paramètres extrinsèques

Les paramètres extrinsèques d'une caméra définissent la position de la caméra dans l'espace ainsi que sa rotation. Ces paramètres sont composés de deux éléments : une matrice de rotation de taille 3x3 permettant de passer du repère lié à l'espace de travail à celui lié à la caméra, et un vecteur de translation qui permet de passer entre ces deux repère.

La matrice de rotation a été calculée grâce aux formules suivantes avec $\lambda = 1/\|\mathbf{A}^{-1}\mathbf{h}_1\|$:

$$\mathbf{r}_1 = \lambda \mathbf{A}^{-1}\mathbf{h}_1$$
$$\mathbf{r}_2 = \lambda \mathbf{A}^{-1}\mathbf{h}_2$$
$$\mathbf{r}_3 = \mathbf{r}_1 \times \mathbf{r}_2$$

Nous avons ensuite calculé le vecteur de translation $\mathbf{t} = \lambda \mathbf{A}^{-1}\mathbf{h}_3$ puis nous avons construit une matrice de taille 3x4 contenant la matrice de rotation accolée au vecteur de translation.

$$\begin{pmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \end{pmatrix}$$

Voici les résultats obtenues grâce à cette formule :

Matrice calculée pour l’image 1 :

0.9999998	0.0009052	0.0009052	- 48.811566
0,0000377	0,9982948	0,0000376	54,733308
0,0006727	-0,0015836	-0,0000011	9899,9694

Résultats souhaités pour l’image 1 :

Translation = (0, 0, 10000), Rotation = (0, 0, 0)

Matrice calculée pour l’image 2 :

0,7108278	0,0007745	0,0005505	-46,0182100
-0,0039613	0,9987511	-0,0039563	43,730543
-0,7033550	-0,0006393	0,0004497	7924,4

Résultats souhaités pour l’image 2 :

Translation = (0, 0, 8000), Rotation = (0, 0.785398, 0)

Matrice calculée pour l’image 3 :

0,9848432	0,1745697	0,1719238	-43,812298
-0,1734468	0,9833136	-0,1705526	49,196866
0,0002845	0,0005549	0,0000002	8911,7289

Résultats souhaités pour l’image 3 :

Translation = (0, 0, 9000), Rotation = (0, 0, -0.174532)

Matrice calculée pour l’image 4 :

1	0,0045336	0,0045336	-143,86961
0,0000023	0,7020868	0,0000016	42,078156
-0,0000611	-0,7176924	0,0000439	8913,4894

Résultats souhaités pour l’image 4 :

Translation = (-100, 0, 9000), Rotation = (-0.785398, 0, 0)

On constate que pour chaque image, les résultats obtenues pour les vecteurs de translation sont vraiment très proches des valeurs originales, comme par exemple (-48, 54, 9899) au lieu de (0, 0, 10000) pour l'image 1.

Concernant les matrices de rotation, je n'ai pas compris comment elles fonctionnent donc je ne commenterai pas les résultats.

Conclusion

Durant ce TP, nous avons étudié la publication de Zhang concernant la calibration de caméra pour apprendre à calculer les paramètres extrinsèques et intrinsèques d'une caméra à partir de plusieurs photographies d'une mire.

Annexe

Code de la fonction principale de la méthode de Zhang :

```
// Lire les coordonnees des points de la mire dans la scene
M = read('points.txt', -1, 2)';
np = size(M, 2);
M = [M; zeros(1, np); ones(1, np)];
sansZ = [1, 2, 4];
// Initialiser la matrice des contraintes
V = [];
// Matrices des homographies
H = zeros(3, 3, ni);
// Coordonnees de tous les points image
m = zeros(3, np, ni);
// Boucler pour toutes les images
for i = 1:ni
    // Lire les points de l'image
    m(1:2,:,i) = read('points-'+string(i)+'.txt', -1, 2)';
    m(3,:,i) = ones(1, np);
    // Estimer l'homographie entre la mire et l' image
    H(:,:,i) = ZhangHomography(M(sansZ,:), m(:,:,i));
    // Ajouter deux lignes de contraintes dans V
    V = [V; ZhangConstraints(H(:,:,i))];
end
// Calculer le vecteur b
b = SmallestRightSingular(V);
// Estimation de la matrice intrinseque
A = IntrinsicMatrix(b);
iA = inv(A);
// Estimations des matrices extrinseques
E = zeros(3, 4, ni);
for i = 1:ni
    E(:,:,i) = ExtrinsicMatrix(iA, H(:,:,i));
end
```

Code de la fonction permettant de calculer la matrice d'homographie :

```
function H=ZhangHomography(M, m)
    // Normalisation des points de la scene
    n = size(M, 2);
    tM = mean(M, 'c');
    sM = stdev(M, 'c');
    nM = [1/sM(1), 0, -tM(1)/sM(1); 0, 1/sM(2), -tM(2)/sM(2); 0, 0, 1];
    M = nM * M;
    // Normalisation des points de l'image
    tm = mean(m, 'c');
    sm = stdev(m, 'c');
    nm = [1/sm(1), 0, -tm(1)/sm(1); 0, 1/sm(2), -tm(2)/sm(2); 0, 0, 1];
    m = nm * m;
    // Matrice L
    u = m(1,:);
    v = m(2,:);
    L = [M', zeros(n,3), -u'.*M(1,:)', -u'.*M(2,:)', -u'; ...
        zeros(n,3), M', -v'.*M(1,:)', -v'.*M(2,:)', -v'];
    // Calcul de l'homographie
    H = inv(nm) * matrix(SmallestRightSingular(L), 3, 3)' * nM;
    H = H / H(3, 3);
endfunction
```

Fonction permettant de calculer la matrice intrinsèque :

```
function A=IntrinsicMatrix(b)
    A = rand(3, 3);
    v0 = (b(2) * b(4) - b(1) * b(5))/(b(1) * b(3) - b(2)^2);
    lambda = b(6) - (b(4)^2 + v0*(b(2) * b(4) - b(1) * b(5))/b(1));
    alpha = sqrt(lambda/b(1));
    beta = sqrt(lambda * b(1)/(b(1)*b(3)-b(2)^2));
    gamma = -b(2)*(alpha^2)*beta/lambda;
    u0 = gamma * v0 / beta - b(4) * alpha^2/lambda;
    A = [alpha,gamma,u0;0,beta,v0;0,0,1]
endfunction
```

Fonction permettant de calculer la matrice extrinsèque :

```
function E=ExtrinsicMatrix(iA, H)
    E = rand(3, 4);
    lambda = 1/norm(iA * H(:,1));
    r1 = lambda*iA*H(:,1);
    r2 = lambda*iA*H(:,2);
    r3 = r1 .* r2;
    t = lambda*iA*H(:,3);

    E = [r1,r2,r3,t];
endfunction
```