COMPTE RENDU

TP1 : GEOMETRIE PROJECTIVE
ET CALIBRATION DE CAMERA

Par
FRANCOIS Rémy

INTRODUCTION

Le but ce TP est d'apprendre à utiliser la méthode de Zhang afin de calibrer une caméra. Le calibrage permet de recueillir des informations concernant la caméra comme sa distance par rapport à l'image, ou son angle de vue.

ESTIMATION DES PARAMETRES EXTRINSEQUES

Dans son rapport, Zhang décrit le procédé pour calibrer une caméra en 3 étapes :

1. Détermination de la matrice d'homographie
2. Détermination de la matrice extrinsèque
3. Détermination de la matrice intrinsèque

Ces 3 étapes sont réalisées dans le code fourni par le code suivant :

Homographie :

```
H = zeros(3, 3, ni);

m = zeros(3, np, ni);

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
```

Matrice intrinsèque :

```
A = IntrinsicMatrix(b);
```

Matrice extrinsèque :

```
E = zeros(3, 4, ni);

for i = 1:ni

  E(:,:,i) = ExtrinsicMatrix(iA, H(:,:,i));
```

End

Dans son rapport, Zhang indique que les contraintes sont obtenues de la manière suivante :

```
vij=    [hi1*hj1;
        hi1*hj2 + hi2*hj1;
        hi2*hj2;
        hi3*hj1+hi1*hj3;
        hi3*hj2+hi2*hj3;
        hi3*hj3]
```

Ce qui en terme de code, nous donne le code suivant :

```
 a = H(1,i)*H(1,j);
 b = H(1,i)*H(2,j)+H(2,i)*H(1,j);
 c = H(2,i)*H(2,j);
 d = H(3,i)*H(1,j)+H(1,i)*H(3,j);
 e = H(3,i)*H(2,j)+H(2,i)*H(3,j);
 f = H(3,i)*H(3,j);
 v = [a,b,c,d,e,f];
```

Ensuite, il faut calculer la matrice intrinsèque A, ce qui est implémenté avec le code suivant :

```
 _v0 = (b(2)*b(4)-b(1)*b(5))/(b(1)*b(3)-b(2)*b(2));
 _lambda = b(6)-(b(4)*b(4)+_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)*b(2)));
 _gamma = -(b(2)*_alpha*_alpha*_beta/_lambda);
 _u0 = _gamma*_v0/_beta - b(4)*_alpha*_alpha/_lambda;

 A =[_alpha,_gamma,_u0;
    0,_beta, _v0;
    0,0,1];
```

Afin de vérifier le bon fonctionnement du code écrit, nous disposons de données issues de scripts povray.

Notre code retourne les valeurs suivantes :

3498.2767	-3.13105	336.76583
0	3503.8946	220.1142
0	0	1

Tandis que les scripts povray donnent les valeurs suivantes :

3546.099291	0	320
0	3546.09929	240
0	0	1

On remarque que les valeurs sont sensiblement égales, donc notre code fonctionne bien.

ESTIMATION DES PARAMETRES EXTRINSEQUES

Le calcul de la matrice extrinsèque E se fait avec le code suivant :

```
lambda = 1/abs(iA*H(:,1));

lambda = lambda(1);

r1 = lambda * iA * H(:,1);

r2 = lambda * iA * H(:,2);

r3 = CrossProduct(r1,r2);

t = lambda * iA * H(:,3);


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

Comme précédemment, pour vérifier que le code écrit est correct, nous allons comparer les valeurs obtenues avec celles obtenues par povray.

Le code retourne les valeurs suivantes :

1	0.0009052	- 0.0006686	- 48.811577
0.0000377	0.9982951	0.0015769	54.73332
0.0006696	- 0.0015763	0.9982950	9854.3628

Ce qui signifie que nous avons en translation (-48, 55, 9854) et en rotation (1, 1, 1).

Les valeurs fournies par povray sont en translation : (0, 0, 10000) et en rotation : (0, 0, 0).

Les valeurs sont très proches, ce qui signifie que le code écrit est bon.

CONCLUSION

Ce TP nous a permis d'utiliser une méthode pour calibrer une caméra, à savoir la méthode de Zhang. La calibration nous permet de recueillir des informations comme l'angle de la prise de vue, la distance, ou bien la position du centre de l'image. Ces informations peuvent servir dans des situations de réalité augmentée par exemple.