Eléments de géométrie projective et calibration de caméra

Yann DUBOIS

September 23, 2017

1 Introduction

L'un des problème de la vision artificielle est de calibrer une caméra situé dans une scène. Pour calibrer une caméra, il faut pouvoir retrouver la matrice intrinsèque et la matrice extrinsèques. Certaines personnes ont cherché une solution à ce problème, c'est le cas de Tsai et de Zhang. Dans ce compte rendu, nous allons nous concentrer sur la méthode de Zhang. Nous commencerons par analyser son travail, puis nous implémenterons sa méthode en commençant par trouver la matrice intrinsèque puis la matrice extrinsèques (Nous reviendrons plus tard sur ces termes).

2 Etude de la publication de Zhang

Zhang propose une méthode de calibration de caméra utilisant une mire plane mais plusieurs position de la caméra (qui n'est pas une translation simple). On obtient alors plusieurs point de la scène qu'on notera Mi. Nous allons utiliser les coordonnées homogènes pour définir les points car nous pouvons définir des points situé à l'infinie dans la géométrie projective avec cette méthodes. On obtient donc pour un point M :

$$M = \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

On modélise la formation d'un point sur le plan image en utilisant la matrice intrinsèque (noté A), la matrice extrinsèque composée d'une matrice de rotation R et d'un vecteur de translation (t). Nous reviendrons dans les parties associées aux différentes matrices pour expliquer ce qu'elle sont vraiment et analyser leurs valeurs. On obtient alors cette équation :

$$s \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = A \begin{bmatrix} R & t \end{bmatrix} M$$

L'idée de Zhang est de considérer que le plan du modèle est situé sur Z = 0 dans les coordonnées du monde. On obtient cette équation :

$$s \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = A \begin{bmatrix} r_1 & r_2 & r_3 & t \end{bmatrix} \begin{bmatrix} X \\ Y \\ 0 \\ 1 \end{bmatrix}$$

$$= A \begin{bmatrix} r_1 & r_2 & t \end{bmatrix} \begin{bmatrix} X \\ Y \\ 1 \end{bmatrix}$$

A noter que r_1, r_2 et r_3 sont les vecteurs constituant la matrice de rotation de la matrice extrinsèque de la caméra. D'après l'équation trouvé par Zhang, on peut considérer une partie de l'équation comme une homographie, c'est à dire une matrice 3x3 avec 8 degrès de liberté. On obtient alors :

$$H = A \begin{bmatrix} r_1 & r_2 & t \end{bmatrix}$$

En partant de cette information, nous pouvons travailler avec et tenter de retrouver la matrice intrinsèque associé à notre caméra.

3 Estimations des paramètres intrinsèques

D'après nos analyses précédentes, nous pouvons déduire cette équation :

$$\begin{bmatrix} h_1 & h_2 & h_3 \end{bmatrix} = \lambda A \begin{bmatrix} r_1 & r_2 & t \end{bmatrix}$$

En partant du prince que r_1 et r_2 sont orthonormaux (c'est à dire perpendiculaire l'un à l'autre et de même norme), on obtiens ces contraintes :

$$h_1^T A^{-T} A^{-1} h_2 = 0$$
$$h_1^T A^{-T} A^{-1} h_1 = h_2^T A^{-T} A^{-1} h_2$$

La première contrainte indique que h_1 et h_2 sont perpendiculaires entre eux et la deuxième indique que h_1 et h_2 sont de même longueur (même norme).

On note :

$$B = A^{-T} A^{-1} \equiv \begin{bmatrix} B11 & B12 & B13 \\ B12 & B22 & B23 \\ B13 & B23 & B33 \end{bmatrix}$$

Pour simplifier les calculs, on va noter b, un vector 6D représentant la matrice symétrique B.

$$b = \begin{bmatrix} B11 & B12 & B22 & B13 & B23 & B33 \end{bmatrix}^T$$

On obtient alors cette égalité :

$$h_i^T B h_j = v_{ij}^T b$$

$$v_{ij} = \begin{bmatrix} h_{i1}h_{j1} & h_{i1}h_{j2} + h_{i2}h_{j1} & h_{i2}h_{j2} & h_{i3}h_{j1} + h_{i1}h_{j3} & h_{i3}h_{j2} + h_{i2}h_{j3} & h_{i3}h_{j3} \end{bmatrix}^T$$

Les 2 contraintes peuvent donc s'écrire de cette façon :

$$\begin{bmatrix} v_{12}^T \\ (v_{11} - v_{22})^T \end{bmatrix} b = 0$$

Etant capable d'estimer l'homographie H, on peut calculer les v_{ij} pour ensuite trouver la valeur de b en cherchant le vecteur singulier droit correspondant à la plus petite valeur singulière de la matrice V (l'ensemble des v_{ij}).

Lorsque b est trouvé, on peut chercher A tel que :

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

u_0 et v_0 désignent la position du centre de l'image qui devrait être égaux respectivement à la taille en longueur et en hauteur divisés par 2. γ designe l'angle d'inclinaison des axes définissant le plan image, ici, il devrait théoriquement être égale à 0 car les axes sont orthogonaux. α et β correspondent aux facteurs d'echelles des deux axes de l'images.

Les différents paramètres de A peuvent être trouvés avec les formules suivantes données par Zhang :

$$v_0 = (B_{12}B13 - B_{11}B_{23})/(B_{11}B_{22} - B_{12}^2)$$
$$\lambda = B_{33} - [B_{13}^2 + v_0(B_{12}B13 - B_{11}B_{23})]/B_{11}$$
$$\alpha = \sqrt{\lambda / B_{11}}$$
$$\beta = \sqrt{\lambda B_{11}/(B_{11}B_{22} - B_{12}^2)}$$

```
-->A
 A  =

    3498.2767  - 3.1310506    336.76583
    0.           3503.8946    220.1142
    0.           0.           1.
```

Figure 1: Résultat de la matrice A

```
Width=640
Height=480
Antialias=On
Antialias_Threshold=0.3
Declare=vPositionCameraX=0
Declare=vPositionCameraY=-10000
Declare=vPositionCameraZ=0
Declare=dAngleX=0
Declare=dAngleY=0
Declare=dAngleZ=0
Input_File_Name=scene.pov
Output_File_Name=image-1.png
```

Figure 2: Paramètres de l'image donnés à povray

$$\gamma = -B_{12}\alpha^2\beta/\lambda$$
$$u_0 = \gamma v_0/\beta - B_{13}\alpha^2/\lambda$$

Sur l'exemple qui est donné en TP, nous obtenons une matrice A (voir Figure 1) que nous pouvons contrôler avec les résultats de povray qui a générer les images avec lesquelles nous travaillons (voir Figure 2).

Après analyses des résultats, on obtient que les valeurs de u_0 et v_0 sont cohérente. En effet, $640/2 = 320$ et $480/2 = 240$, u_0 et v_0 valent respectivement 336 et 220. On a ici un taux d'erreur entre 5% et 8%. Le coefficient γ est négatif et est plus petit d'un facteur 1000 par rapport à α et β ce qui le rende négligeable comme espéré. Bien sûr la valeur n'est pas égale à 0 comme attendu, mais ceci est causé par l'estimation qui est faite. Pour terminer avec la matrice intrinsèque, les coefficients α et β devaient être égaux, ce qui est presque le cas à $\pm$ 10 environs

Une fois que la matrice intrasèque est trouvé, nous pouvons chercher la pose de la caméra ou la matrice extrinsèque.

4 Estimations des paramètres extrinsèques

Trouver la matrice extrinsèque reviens à chercher la matrice de rotation et le vecteur de position de la caméra dans l'espace 3D. On note cette matrice :

$$E = \begin{bmatrix} R & t \end{bmatrix}$$

Avec R la matrice de rotation et t la matrice de position. Zhang nous donne les formule à appliquer pour trouver les vecteurs associés à la matrice de rotation et le vecteur de position.

$$r_1 = \lambda A^{-1} h_1$$

```
-->E
 E  =

(:,:,1)

    0.9999998    0.0009052  - 0.0006686  - 48.811558
    0.0000377    0.9982948    0.0015769    54.73331
    0.0006696  - 0.0015763    0.9982946    9854.3606
(:,:,2)

    0.7124496    0.0007762    0.7024372  - 46.123202
  - 0.0039703    1.0010299  - 0.0000903    43.830322
  - 0.7017120  - 0.0006379    0.7131865    7905.8899
(:,:,3)

    0.9848432    0.1745697  - 0.0003743  - 43.812292
  - 0.1734468    0.9833136  - 0.0004946    49.196567
    0.0002832    0.0005524    0.9986883    8870.6728
(:,:,4)

    1.           0.0045336    0.0000411  - 143.8696
    0.0000023    0.7020868    0.7143857    42.078157
  - 0.0000609  - 0.714386     0.7020868    8872.4252
```

Figure 3: Estimation de la pose de la caméra

$$r_2 = \lambda A^{-1} h_2$$
$$r_3 = r_1 \times r_2$$
$$t = \lambda A^{-1} h3$$
$$\lambda = 1/||A^{-1} h_1|| = 1/||A^{-1} h_2||$$

Lorsqu'on essaie d'estimer cette matrice avec notre script, on obtient les matrices de la Figure 3

Les coefficient correspondants aux positions de la caméra sont plutôt correct, même si les coefficients de Z et Y ont l'air d'avoir été interchangé. C'est sans doute un effet du passage de povray vers scilab.

Par contre, l'estimation, créant du bruit, nous donne une matrice de rotation qui ne ressemble pas à ce qu'on attend d'une telle matrice. Pour rappel, la matrice de rotation attendu est de cette forme (paramétrisation de Rodriguez) :

$$R = \cos\theta I_{3x3} + (1 - \cos\theta) n^T + \sin\theta \begin{bmatrix} 0 & -n_3 & n_2 \\ n_3 & 0 & -n_1 \\ -n_2 & n_1 & 0 \end{bmatrix}$$

Znhang propose une solution pour réduire le bruit de la matrice de rotation et donc de l'estimer plus précisément même si ça reste approximatif. Il propose de calculer la norme de Frobenius de la différence la meilleure matrice de rotation R avec une matrice 3x3 notée Q donnée à approximer. Ce qui revient à résoudre ce problème :

$$\min_R ||R - Q||_F^2$$

Nous voulant pas me tromper en essayant d'expliquer sa méthode, je ne vais pas m'attarder dessus. Mais pour en parler un peu, la technique consiste à calculer des traces de matrices.

En ce qui concerne le vecteur position, le problème est important lorsqu'on obtient des valeurs basse (par exemple lorsqu'on doit trouver 100 mais qu'on trouve 143). Ici on a un taux d'erreur de 43% ce qui est beaucoup. Mais lorsqu'on regarde avec des plus haute valeur, l'erreur n'est pas aussi importante. Il est donc difficile de trouver une technique qui ne risque pas de trop modifier les hautes valeurs.

5 Conclusion

Dans ce TP nous avons vu une technique permettant de calibrer une caméra artificiellement en retrouvant sa matrice intrinsèque et sa matrice extrinsèque. Nous avons vu que les valeurs obtenues ne sont pas exacts à 100% et que l'erreur faite sur la matrice intrinsèque implique une erreur plus importante sur la matrice extrinsèque.

References

[Zha98] Zhengyou Zhang. A flexible new technique for camera calibration. *Technical Report*, 1998.