Modèles de caméras

Introduction

Les notions d'imagerie abordées en cour sont illustrées dans ce td à l'aide de matlab, outil de calcul numérique. La table 1 montre un exemple de script matlab permettant de lire, manipuler et sauvegarder des images.

Le but de ce TD est d'étudier le modèle de caméra de type *pinhole*. La scène observée est un cube de coté unitaire. Un repère Monde R_w est positionné au centre du cube, dont les axes sont parallèles au repère présenté figures (1,2,2,4).

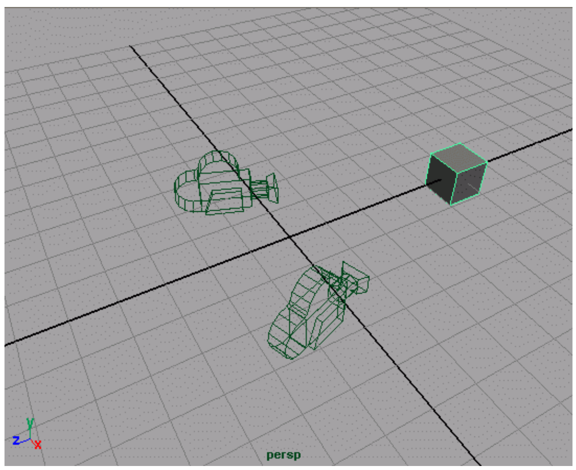

Figure 1 – Scène 3D

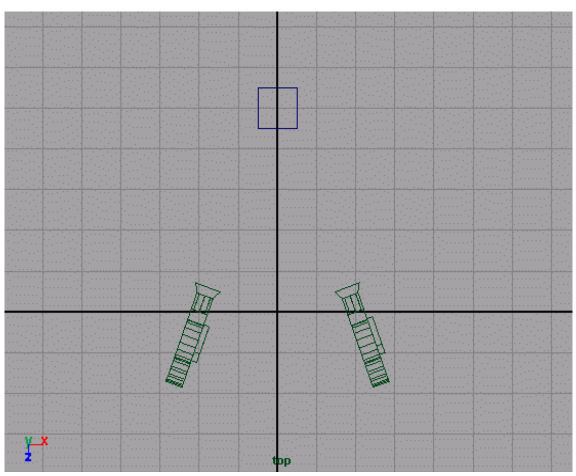

Figure 2 – Scène 3D : vue de dessus

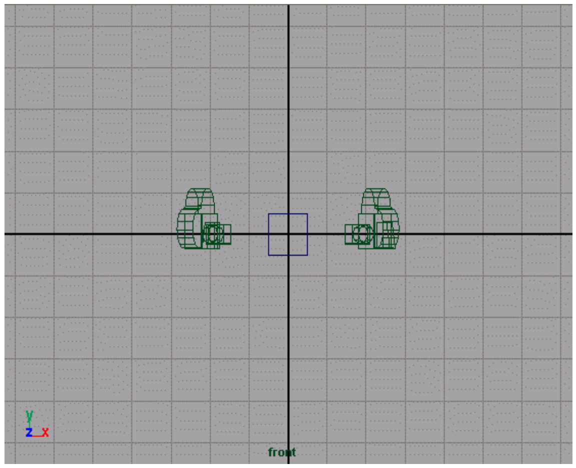

Figure 3 – Scène 3D : vue de face

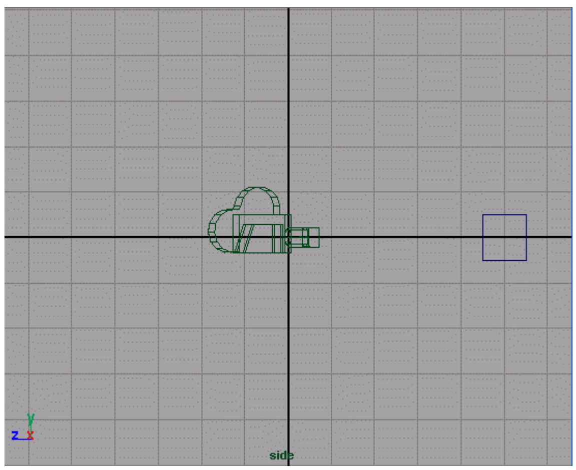

Figure 4 – Scène 3D : vue de coté

La première partie du TD consiste à revoir la notion de changement de repère 3D modélisé à l'aide de transformations linéaires. La deuxième partie du TD à calculer le modèle d'une caméra, connaissant les paramètres intrinsèques et extrinsèques.

1 Calcul de la caméra à partir des paramètres extrinsèques et intrinsèques

Les paramètres intrinsecs de la camréra sont les suivants :
- f=1750
- u0 = 800
- v0 = 600

La figure 5 représente la vue de la caméra de gauche, inclinée de 20^0.

Travail demandé : (Aidez vous de `geom.m` table 3)

1. Calculer la matrice des paramètres extrinsecs de la caméra (Passage entre le repère

monde et le repère caméra).

2. Calculer la matrice de projection (paramètres intrinsecs de la caméra).
3. On reprojette certains points sur l'image pour valider le calcul des paramètres de la caméra

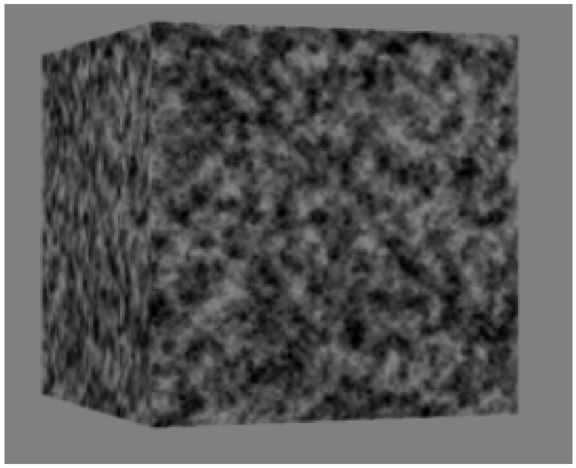

Figure 5 – Vue caméra de gauche (Zoom)

2 Stérovision

Le principe de la stéréovision consiste, à estimer les coordonnées 3D d'un point à partir de sa projection dans deux caméras calibrées. La figure 6 illustre ce principe. Lorsque l'on connait la projection du point recherché dans l'une des deux caméras, l'espace de recherche du point correspondant dans la deuxième image se limite à une droite. Cette contrainte s'appelle contrainte épipolaire. Lorsque les deux caméra sont parfaitement alignées (axe x confondus et axes y parallèles), la zone de recherche de la projection d'un point 3D dans une image se limite à la même ligne que celle de la projection du point dans la deuxième image. Cela accélère de manière importante les temps de calcul. De manière pratique, les caméras ne sont jamais parfaitement alignées. Il faut alors avoir recours à une phase de rectification des images qui consiste à appliquer une homographie pour obtenir des images calibrées.

Une des difficultés en stéréovision est la phase de mise en correspondance des points des deux images. Il existe plusieurs techniques, plus ou moins précises et plus ou moins rapides.

1. Montrez que la contrainte epipolaire impose, dans l'exemple courant, qu'un point 3D se projetant sur la ligne i de l'image de gauche, se projette également sur la ligne i de l'image de droite
2. Ecrire le problème d'estimation des coordonnées d'un point 3D exprimé dans le repère monde par un système matriciel de dimension 3 faisant intervenir les projections u et v du point dans la caméra droite et la projection u du point dans la caméra gauche.
3. Editer la fonction Matlab `stereo.m`. Cette fonction lit les images droite et gauche, calcule des points d'intérêt (HARRIS), et des correspondances entre les points par

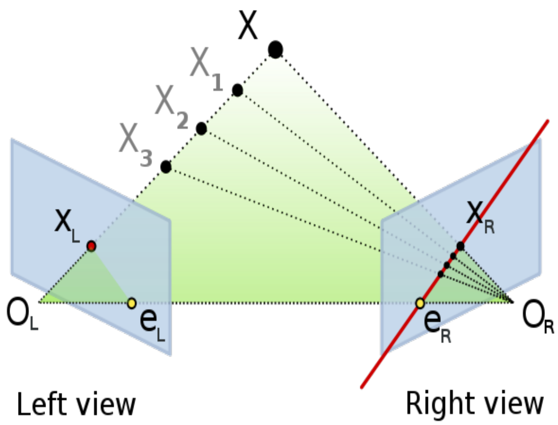

Figure 6 – Illustration de la contrainte épipolaire

corrélation centrée croisée normalisée (ZNCC). Vous devez modifier cette fonction pour estimer les coordonnées 3D de chaque couple de point mis en correspondance, et afficher le nuage de points 3D ainsi généré.

Pour info, d'autres mesures de distances entre *patchs* peuvent être utilisés :

– La somme des écarts des carrés des distances (ssd) :

$$d = (\mathbf{x} - \mathbf{y})^T(\mathbf{x} - \mathbf{y})$$

– La somme des valeurs absolues des distances (sad) :

$$d = \sum_{i=1}^{n} |x_i - y_i|$$

– La corrélation croisée normalisée(ncc) :

$$d = \frac{\sum_{i=1}^{n} x_i y_i}{||\mathbf{x}||\ ||\mathbf{y}||}$$

– La corrélation centrée croisée normalisée :

$$d = \frac{\sum_{i=1}^{n} (x_i - \overline{\mathbf{x}})(y_i - \overline{\mathbf{y}})}{||\mathbf{x} - \overline{\mathbf{x}}||\ ||\mathbf{y} - \overline{\mathbf{y}}||}$$

où $\overline{\mathbf{x}}$ est la moyenne de $\mathbf{x}$.

Il est aussi possible d'utiliser une fonction de calcul de corrélation présente dans Matlab. Il faudra afficher le point correspondant sur l'image de droite est calculer ses coordonnées 3D, à partir du système défini dans la question précédente.

Annexe : matrices de transformation

Dans le cas d'une transformation homogène, le type de représentation est matriciel. Le passage d'un repère initial R_i à un repère final R_f s'exprime par l'intermédiaire d'une matrice M, appelée **matrice de changement de repère, matrice de passage ou matrice de transformation homogène**. Cette matrice de dimension (4×4), notée ${}^iM_f$ s'exprime sous la forme :

$$ {}^iT_f = {}^iM_f = ({}^is_j \; {}^in_j \; {}^ia_j \; {}^iP_j) = \begin{pmatrix} s_x & n_x & a_x & P_x \\ s_y & n_y & a_y & P_y \\ s_z & n_z & a_z & P_z \\ 0 & 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} {}^iR_f & {}^iP_f \\ 0 & 1 \end{pmatrix} \tag{1} $$

A l'aide de la matrice ${}^iM_f$, il est possible d'exprimer les coordonnées d'un point quelconque P de l'espace dans le repère R_i à partir de ces coordonnées homogènes exprimées dans le repère R_f par la relation :

$$ \boxed{\begin{pmatrix} x' \\ y' \\ z' \\ 1 \end{pmatrix}_{R_i} =^i M_f. \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix}_{R_f} = \begin{pmatrix} {}^iR_f & {}^iP_f \\ 0 & 1 \end{pmatrix} . \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix}_{R_f}} \tag{2} $$

Lorsque deux repères sont uniquement liés par une translation, il est possible de passer de l'un à l'autre en utilisant une matrice de transformation homogène de translation pure. Nous utiliserons les notations suivantes :

- Trans(a, b, c) pour indiquer une translation (a selon l'axe x, b selon l'axe y et c selon l'axe z)
- Trans(x, a) pour indiquer une translation a selon l'axe x
- Trans(y, b) pour indiquer une translation b selon l'axe y
- Trans(z, c) pour indiquer une translation c selon l'axe z

$$ M = \begin{pmatrix} & & & a \\ & I_3 & & b \\ & & & c \\ 0 & 0 & 0 & 1 \end{pmatrix} = \tag{3} $$

Lorsque deux repères sont uniquement liés par une rotation, il est possible de passer de l'un à l'autre en utilisant une matrice de transformation homogène de rotation pure. Nous utiliserons les notations suivantes :

- $Rot(x, \theta_x)$ pour indiquer une rotation (θ_x autour de l'axe x)
- $Rot(y, \theta_y)$ pour indiquer une rotation (θ_y autour de l'axe y)
- $Rot(z, \theta_z)$ pour indiquer une rotation (θ_z autour de l'axe z)

Dans la matrice de transformation homogène, la rotation est décrite par la matrice R présentée dans l'équation (1) page 5. Lorsque la rotation est nulle autour des trois axes, R devient la matrice identité (c'est le cas pour les rotations pures) :

$$ R = I_3 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \tag{4} $$

Exemple 1 : Une rotation θ_x autour de l'axe x.

$$ R = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos\theta_x & -\sin\theta_x \\ 0 & \sin\theta_x & \cos\theta_x \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \mathrm{c}\theta_x & -\mathrm{s}\theta_x \\ 0 & \mathrm{s}\theta_x & \mathrm{c}\theta_x \end{pmatrix} \text{ (autre notation)} \tag{5} $$

Notons $(\underline{i_{R_f}}, \underline{j_{R_f}}, \underline{k_{R_f}})$ la base associée au repère R_f et $(\underline{i_{R_i}}, \underline{j_{R_i}}, \underline{k_{R_i}})$ la base associée au repère R_i. La matrice de rotation R est obtenue en décrivant $(\underline{i_{R_f}}, \underline{j_{R_f}}, \underline{k_{R_f}})$ en fonction de $(\underline{i_{R_i}}, \underline{j_{R_i}}, \underline{k_{R_i}})$:

$$\begin{array}{lcllllllll} \underline{i_{R_f}} & = & 1.\underline{i_{R_i}} & + & 0.\underline{j_{R_i}} & + & 0.\underline{k_{R_i}} & = & {}^i\underline{s}_f \\ \underline{j_{R_f}} & = & 0.\underline{i_{R_i}} & + & \cos\theta_x.\underline{j_{R_i}} & + & \sin\theta_x.\underline{k_{R_i}} & = & {}^i\underline{n}_f \\ \underline{k_{R_f}} & = & 0.\underline{i_{R_i}} & - & \sin\theta_x.\underline{j_{R_i}} & + & \cos\theta_x.\underline{k_{R_i}} & = & {}^i\underline{a}_f \end{array} \tag{6}$$

Exemple 2 : Une rotation θ_y autour de l'axe y.

$$R = \begin{pmatrix} \cos\theta_y & 0 & \sin\theta_y \\ 0 & 1 & 0 \\ -\sin\theta_y & 0 & \cos\theta_y \end{pmatrix} = \begin{pmatrix} \mathrm{c}\theta_y & 0 & \mathrm{s}\theta_y \\ 0 & 1 & 0 \\ -\mathrm{s}\theta_y & 0 & \mathrm{c}\theta_y \end{pmatrix} \text{ (autre notation)} \tag{7}$$

Exemple 3 : Une rotation θ_z autour de l'axe z.

$$R = \begin{pmatrix} \cos\theta_z & -\sin\theta_z & 0 \\ \sin\theta_z & \cos\theta_z & 0 \\ 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} \mathrm{c}\theta_z & -\mathrm{s}\theta_z & 0 \\ \mathrm{s}\theta_z & \mathrm{c}\theta_z & 0 \\ 0 & 0 & 1 \end{pmatrix} \text{ (autre notation)} \tag{8}$$

```
%%%%%%%%%%%%%%%%%%%%%%%%%
%␣script␣d'exemple␣matlab
%%%%%%%%%%%%%%%%%%%%%%%%%
%
%␣lecture␣d'une␣image
[I,map]=imread('meadownb.jpg');
%%%%%%%%%%%%%%%%%%%%%%%%%
%␣Affichage␣d'une␣image
image(I);
%␣pause␣il␣faut␣appuyer␣sur␣une
%␣touche␣pour␣reprendre
pause␣;
colormap(gray(256));
%%%%%%%%%%%%%%%%%%%%%%%%%
%␣Convertion␣de␣l'image␣en
%␣format␣double␣afin␣de␣la␣modifier
Id=double(I);
%%%%%%%%%%%%%%%%%%%%%%%%%
%␣Extraction␣d'une␣partie␣de␣l'image
%␣des␣lignes␣100␣à␣300␣et
%␣des␣colonnes␣200␣à␣400
I3=Id(100:300,200:400);
%%%%%%%%%%%%%%%%%%%%%%%%%
%␣Sauvegarde␣d'une␣image
imwrite(uint8(I3),gray(256),'toto.jpg','JPEG');
```

Table 1 – Exemple de script matlab.

```
%%%%%%%%%%%%
%␣Exemple␣de␣fonction␣matlab␣fichier␣fo.m
%␣Les␣parametres␣d'entree␣sont␣a,b,c
%␣les␣parametres␣de␣sortie␣sont␣t1␣et␣t2
%%%%%%%%%%%
%%%%%%%%%%%
%␣Prototype
function␣[t1,t2]=fo(a,b,c)
%
t1=a+b;
t2=b+c;
```

Table 2 – Exemple de fonction matlab.

```
% $$$ T Chateau Lasmea
% geom.m
% Chargement de l'image
%%%%%%%%%%%%%%%%%%%%%%%
%Image 1 (caméra gauche)
clear;
[I1,map1]=imread('cam1.tif');
I1=I1(:,:,1);
image(I1);
colormap(gray(256));
%% Définition des paramêtres extrinsecs
%% On définit le reprère monde au centre de l'objet
%%%%%%%%%%%%%%%%%%%%%%
% A remplir par l'étudiant

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% Paramètres intrinsecs de la caméra
f=1750;
u0=800;
v0=600;
% Matrice de projection
%%%%%%%%%%%%%%%%%%%%%%%
% Aremplir par l'étudiant
P=

% Matrice totale :
%%%%%%%%%%%%%%%%%%%
% Aremplir par l'étudiant
Tr=

%%%%%%%%%%%%%%%%%%%%%
% Projection des points
p1=[0.5 ; -0.5 ; 0.5 ; 1]
pi1=Tr*p1
pi1=pi1./(pi1(3))
hold on;
plot(pi1(1),pi1(2),'+r');
p1=[0.5 ; 0.5 ; 0.5 ; 1]
pi1=Tr*p1
toto=RT*p1
pi1=pi1./(pi1(3))
hold on;
plot(pi1(1),pi1(2),'+b');
```

Table 3 – Fonction principale à compléter .