

# Projection perspective

François LEPAN

8 février 2013

## 1 Modèles simples d'objets 3D

### 1.1 Un cube de côté 1 m dont le barycentre est l'origine du repère monde, et dont les côtés sont parallèles aux axes de ce repère

Voici la fonction correspondant à la création de ce cube :

```
function m = cube()
    p1 = [-0.5, -0.5, -0.5, 1];
    p2 = [0.5, -0.5, -0.5, 1];
    p3 = [0.5, 0.5, -0.5, 1];
    p4 = [-0.5, 0.5, -0.5, 1];
    p5 = [-0.5, -0.5, 0.5, 1];
    p6 = [0.5, -0.5, 0.5, 1];
    p7 = [0.5, 0.5, 0.5, 1];
    p8 = [-0.5, 0.5, 0.5, 1];

    m = [p1', p2', p3', p4', p5', p6', p7', p8'];
endfunction

function seg = segmentCube()

    liste = [1:4];
    liste1 = liste+4;

    liste2 = moveNElemToTheEnd(liste, liste(1,1))
    liste3 = moveNElemToTheEnd(liste1, liste1(1,1))

    seg = [liste, liste1, liste; liste2, liste3, liste1]
endfunction
```

### 1.2 Une grille plane composée de 15 carrés (5 selon x et 3 selon y) de côtés 1 m, dont le barycentre est l'origine du repère monde, et située dans le plan $z=0$ .

Voici la fonction correspondant à la création de cette grille :

```

function m = grille()
    x = [1:6] - 3.5;
    y = [-1.5,-1.5,-1.5,-1.5,-1.5,-1.5];
    z = zeros(1,24);
    w = ones(1,24);

    m = [x, x, x, x; y, y+1,y+2, y+3; z; w]
endfunction

function seg = segmentGrille()
    liste = [1:5];
    liste2 = [1:6];
    seg = [liste, liste+6, liste+12, liste+18, liste2, liste2+6, liste2+12;
           liste+1, liste+7, liste+13, liste+19, liste2+6, liste2+12, liste2+18];
endfunction

```

## 2 Matrice extrinsèque

### 2.1 Définition des quatre fonctions

Voici les quatre fonctions qui permettent de calculer des matrices extrinsèques :

#### *RotationX(theta)*

```

function m = RotationX(theta)
    m = [1,      0,      0,      0;
          0,cosd(theta),-sind(theta), 0;
          0,sind(theta),cosd(theta), 0;
          0,      0,      0,      1];
endfunction

```

#### *RotationY(theta)*

```

function m = RotationY(theta)
    m = [cosd(theta),0,sind(theta), 0;
          0,      0,      0,      0;
          -sind(theta),0,cosd(theta),0;
          0,      0,      0,      1];
endfunction

```

#### *RotationZ(theta)*

```

function m = RotationZ(theta)
    m = [cosd(theta),-sind(theta),0 ,0;
          sind(theta),cosd(theta) ,0 ,0;
          0          ,0          ,1 ,0;
          0          ,0          ,0 ,1];
endfunction

```

```
endfunction
```

***Translation(x,y,z)***

```
function m = Translation(x,y,z)
    m = [1,0,0,x;
          0,1,0,y;
          0,0,1,z;
          0,0,0,1];
endfunction
```

## 2.2 Déterminer les matrices extrinsèques positionnant les caméras suivantes :

### 2.2.1 Centre optique (0, 0, -5 m), axe optique orienté selon z, verticale de la caméra selon y

Voici le code correspondant à la création de cette matrice :

```
translation = Translation(0,0,-5);
rotation = RotationX(90);
position = translation * rotation;
```

On peut le vérifier de cette façon, avec ce code :

```
translation = Translation(0,0,-5);
rotation = RotationZ(45);
position = translation * rotation;
```

On obtient la Fig. 1

Si après la rotation de 90 degré sur l'axe X on ajoute `rotation = rotation * RotationY(45);` cela nous donne la même figure que précédant vu que l'axe des Y se trouve dans l'axe de la caméra.

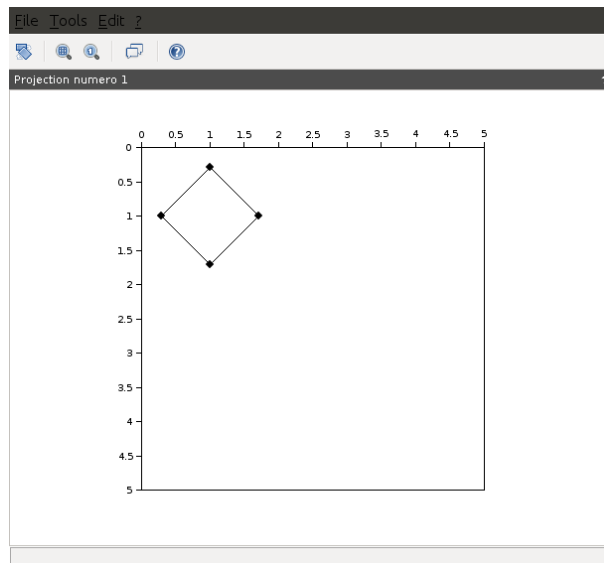


FIGURE 1 – cube translater sur (1,1,-5) puis tourné à 45 degré sur l'axe Z

### 2.2.2 Axe optique selon la diagonale principale du repère, regardant le centre du repère. Centre optique situé à une distance de 5 mètres du centre du repère. Verticale de la caméra dans un plan contenant z

Voici le code correspondant à la création de cette matrice :

```
position1 = Translation(0,0,5)*RotationY(135)*RotationX(45);
```

## 3 Matrice intrinsèque

Voici la fonction permettant de calculer la matrice intrinsèque :

```
function m = Intrinsec(f,Sc, Sl, Oc, Ol, matExtr)
    proj = [f, 0, 0, 0;
            0, f, 0, 0;
            0, 0, 1, 0]

    xiyi = proj * matExtr;

    m1 = [1/Sc, 0, Oc;
          0, 1/Sl, Ol;
          0, 0, 1]

    m = m1*xiyiw;
endfunction
```

Voici le code que j'ai pour tester l'affichage :

```

// le cube
c = cube();
// segment du cube
sc = segmentCube();

position1 = Translation(0,0,5)*RotationY(135)*RotationX(45);
c2=position1*c;
matIntrinsec = Intrinsec(20,8.8/800,6.6/600,400,300,c2);
tiAfficheObjet2D(1,[5,5],matIntrinsec,sc);

```

Je n'arrive pas à afficher les images malgré plusieurs tentatives.