SCIENCES SORBONNE UNIVERSITÉ

Master Robotique,Automatique : ISI

COMPTE RENDU DE PROJET

VISION PAR ORDINATEUR

Sujet : Réalité augmentée

Realisé par :

Lisa TOUZARI

Yacine GHEMMOUR

Chihab Eddine LEKHMAMRA

1 Calibration d'une camera

1.1 Utilisation de cameraCalibrator pour le calcul de la matrice de projection P

L'étalonnage d'une caméra ou sa calibration consiste à determiner la matrice de projection P.
L'outil Matlab nous fournit une application cameraCalibrator, qui à partir d'un ensemble d'images de mires prises avec la caméra qu'on veut calibrer, nous donne l'ensemble de : matrice intrinsèque (K) et extrinsèque (R,T).

1.1.1 Test de cameraCalibrator sur nos données

Dans cette partie on va essayer de determiner la matrice de projection de notre camera.
On entre nos données, c'est à dire les images prises avec notre caméra, dans camaraCalibrator et on execute les instructions suivantes :

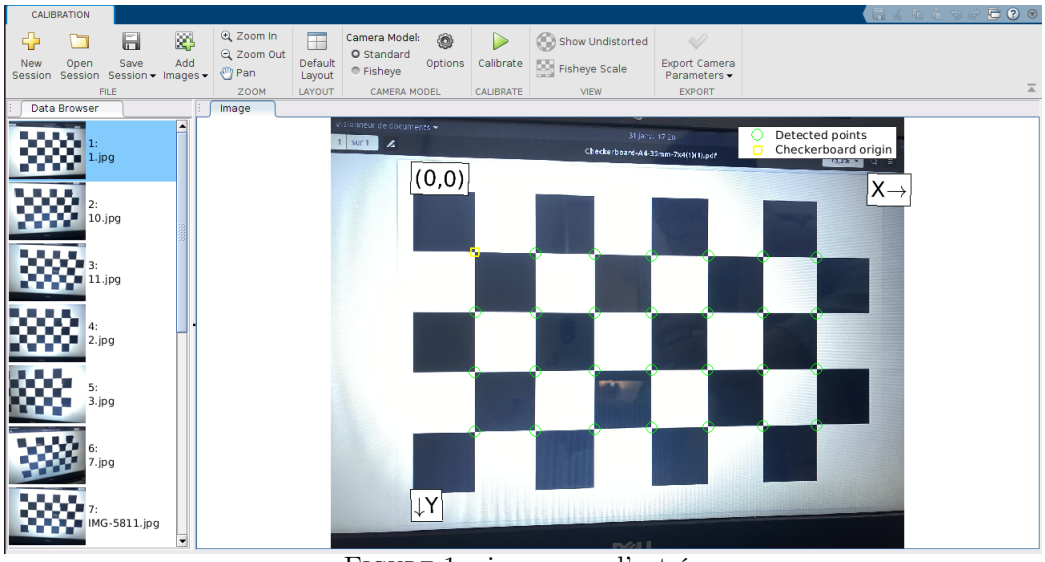

FIGURE 1 – images en d'entrée

Après chargement des données, on calibre notre caméra.

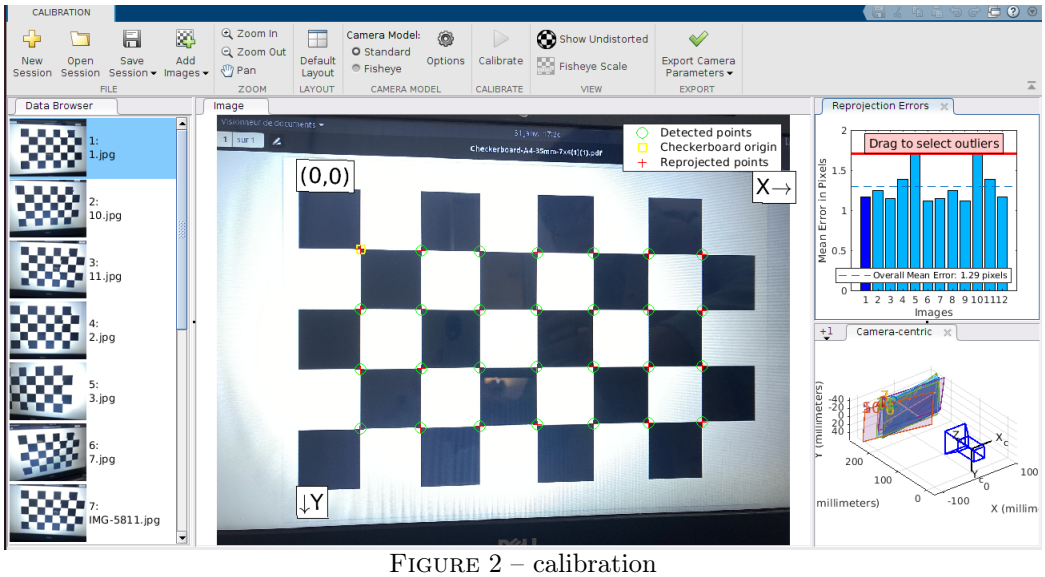

FIGURE 2 – calibration

Au niveau de la fenêtre command window, on execute les instructions suivantes pour recupérer la matrice intrinsèque K, et aussi la matrice de rotation et translation de la première image.Ainsi que le calcul de la matrice de projection P à patir de K,R,T.
K=cameraParams.IntrinsicMatrix'
R=cameraParams.RotationMatrices(:, :,1)
T=cameraParams.TranslationVectors(1, :)

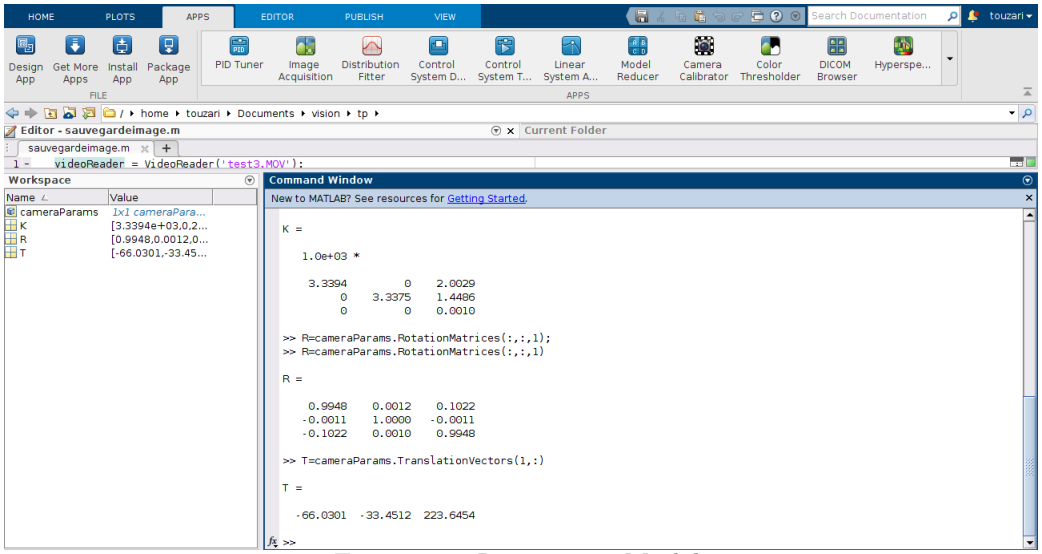

FIGURE 3 – Instructions Matlab

1.2 Calcul de la matrice de projection en utilisant la méthode de la DLT

Pour le calcul de la mtrice P, on suppose dans ce cas que l'ensemble des points monde M et leurs points images correspondants sont connus. Avec la méthode de la DLT, on pourra réduire le nombre de couples de points (M_i,m_i) necéssaires pour le calcul de P. Dans le cas d'un calcul direct, on aurai eu besoin d'un

minimum de 12 couples, pour le calcul des 12 inconnus de notre matrice, mais avec la DLT on utilise que 6 couple.
Le principe de la méthode et de réecrire notre système sous forme de l'équation A.X=0, X étant un vecteur colonne de 12 lignes (composantes de P).
On construit alors la matrice A à partir des couples de points et on applique l'instruction Matlab SVD. Cette instruction permet d'écrire la matrice A sous forme de A=U.D.V'. Et V étant égal à X, on pourra facilement reconstruire alors notre matrice P(3*4).

1.3 Matrice d'homographie

Dans le cas particulier où les points monde sont des points 2D et non pas des points 3D, on peut faire une simplification de calcul en éliminant la composante z . On peut calculer notre matrice d'homograohie de la même façon que la matrice P. Cette matrice vérifie l'équation m_i=H*M_i.

1.4 Comparaison entre les résultats donnés par cameraCalibrator et de l'homographie

Les points M_{monde} et m_{image} étants supposés connus, on peut vérifier alors si notre matrice d'homographie est correct en exécutant l'instruction m_{image}= H*M_{monde}, sans oublier d'éliminer la composante z de M_{monde} et de diviser chaque colonne de la matrice m_{images} obtenue par son troisième terme.
Pour verifier le resultat de la matrice de projection obtenue avec cameraCalibrator, on execute l'instruction m_{image}= H*M_{monde}, on remarque alors que les points images obtenus avec les deux méthodes sont presque les mêmes.

2 Problème de réalité augmentée

Le problème traité dans le cadre de ce projet est un problème de réalité augmentée.On veut injecter un objet virtuel dans une séquence video.On peut traiter ce problème en 5 étapes.

2.1 étape 1 : Déterminer la matrice intrinsèque K de notre caméra

Comme expliqué précédemment, on se servira de l'outil Matlab, et l'application cameraCalibrator pour determiner la matrice K de la caméra utilisée pour filmer la scène à modifier.
Il suffira alors de prendre un esnsemble d'images d'une mire avec notre camera et traiter les images obtenues avec cameraCalibrator pour trouver la matrice intrinsèque.

2.2 étape 2 : La mise en correspondance entre les points d'interêt entre les différentes images

Dans la deuxième étape de ce projet, on va filmer une scène dans laquelle on définit 4 points d'interêt(4 points monde) dont on connait les coordonnées 3D. Avec l'outil Matlab toujours, on sectionne notre video en un ensemble d'images.
On repère sur la première image nos 4 points d'interêt et on relève leurs coordonnées en pixel(c'est à dire on aura les points images correspondants aux points monde dans la première image).
Le problème alors est de repérer la projection de nos points d'interêt sur l'ensemble des images. Pour cela, on met en place un tracker qui va suivre les points d'interêt dans chaque image et ainsi recupérer les coordonnées pixel correspondantes.

2.3 étape 3 : Calcul de la matrice Homographie pour chaque image

Dans notre cas, on a définit nos 4 points monde comme étant des point 2D, c'est à dire que la composante z est nulle, c'est pour ça on cherche à determiner la matrice d'homographie.
Après avoir recupéré l'ensemble des points images correspondants aux points monde qu'on a predéfinit, on pourra alors facilement calculer la matrice d'homographie pour chaque image en appliquant la méthode de la DLT.

2.4 étape 4 : Reconstitution de la matrice de projection de notre caméra

A cette étape, on voit plus clairement l'interêt des étapes précédentes, qui est de déterminer les matrices de projection correspondantes à chaque image afin de pouvoir projeter les points 3D virtuels sur nos images.
On peut reconstruire chaque matrice P à partir d'une matrice H avec la méthode suivante :
On sait que :
H=$\alpha * K * (r1, r2, T)$ =>k^{-1}H=$(\alpha * r1, \alpha * r2, \alpha * T)$
$det(\alpha * r1, \alpha * r2, \alpha^2 *$r1 $\times r2)=\alpha^4$

De ces deux équations on peut alors déterminer la valeur de notre alpha. On aura deux cas possibles,on choisis la valeur de notre alpha selon le resultat de la simulation.
Pour reconstruire alors la matrice P,il suffira de diviser k^{-1}H par alpha, avec P= K(r1,r2,$r1 \times r2$,T)

2.5 étape 5 : Projection des points 3D virtuels sur nos images

On predéfinit une matrice de points 3D virtuels qu'on veut projeter sur nos images(c'est à dire notre vidéo), on se doit alors de trouver les points images correspondants dans chaque image. Comme on a précédement déterminé l'ensemble des matrices de projection de chaque image alors tout simplement pour trouver les points images virtuels on multiplie par P : $m_{virtuels}$=P*$M_{virtuels}$.
Après projection des points sur les images, on va reconstituer la nouvelle vidéo à partir des nouvelles images.