UE MU4RBI09 - S1-22
Rapport de Projet Vision

HAMDI Massyl Yanis, 21118208
Ouali sofiane, 3802574
2022-2023

1. INTRODUCTION :

La vision par ordinateur est une discipline qui consiste à utiliser des algorithmes informatiques pour traiter et analyser des images ou des séquences vidéo. Elle joue un rôle important dans de nombreuses applications telles que la reconnaissance de formes et de visages, la détection de mouvement, la segmentation d'image, la cartographie 3D, et bien d'autres encore.

La vision par ordinateur s'appuie sur différentes techniques de traitement d'image, de traitement du signal et d'apprentissage automatique pour extraire des informations utiles à partir d'images numériques. Elle est largement utilisée dans de nombreux domaines, tels que la robotique, la sécurité, la médecine, les transports, la réalité augmentée et la vision industrielle.

En résumé, la vision par ordinateur est une technique importante qui permet à l'ordinateur de comprendre et d'interpréter des images de manière automatisée, ce qui peut être utilisé dans de nombreux domaines pour améliorer l'efficacité et la précision de nombreuses tâches.

Présentation du projet :

Dans ce projet, à l'aide des outils théoriques vus en cours on a développé une application de réalité augmentée qui permet de superposer un cube en 3D sur une scène vidéo. Le rapport de projet est structuré en trois parties :

1. Méthodes : cette section décrit les différentes étapes de développement de l'application, notamment la calibration de la caméra, l'estimation de l'homographie, l'estimation de la pose et le tracking de primitives.
2. Évaluation de l'erreur : cette partie présente les différence entre les resultat estimé par notre implementation et les résultats fournis par la calibration.
3. Conclusion : enfin, cette section résume les principaux résultats obtenus et propose des pistes d'amélioration pour la suite du projet..

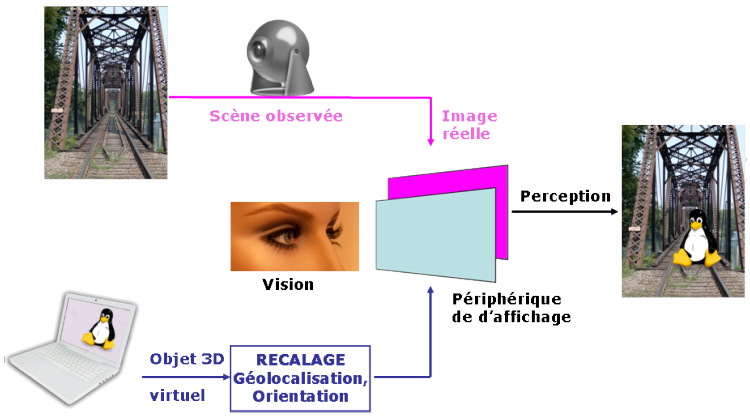

FIGURE 1 – Principe de realite augmentee

2. Méthodes :

I. Calibration :

La calibration de la caméra est le processus consistant à déterminer les paramètres de l'appareil tel que les paramètres intrinsèques, extrinsèques qui constitue la matrice de projection de manière à pouvoir mieux comprendre comment il transforme le monde réel en une image numérique.

En outre la calibration de la caméra est un processus crucial pour garantir la qualité et la précision des images capturées par la caméra et pour permettre leur utilisation dans diverses applications.

Procédure à suivre :
Il faut tout d'abord ouvrire l'application "Camera Calibrator" de MATLAB

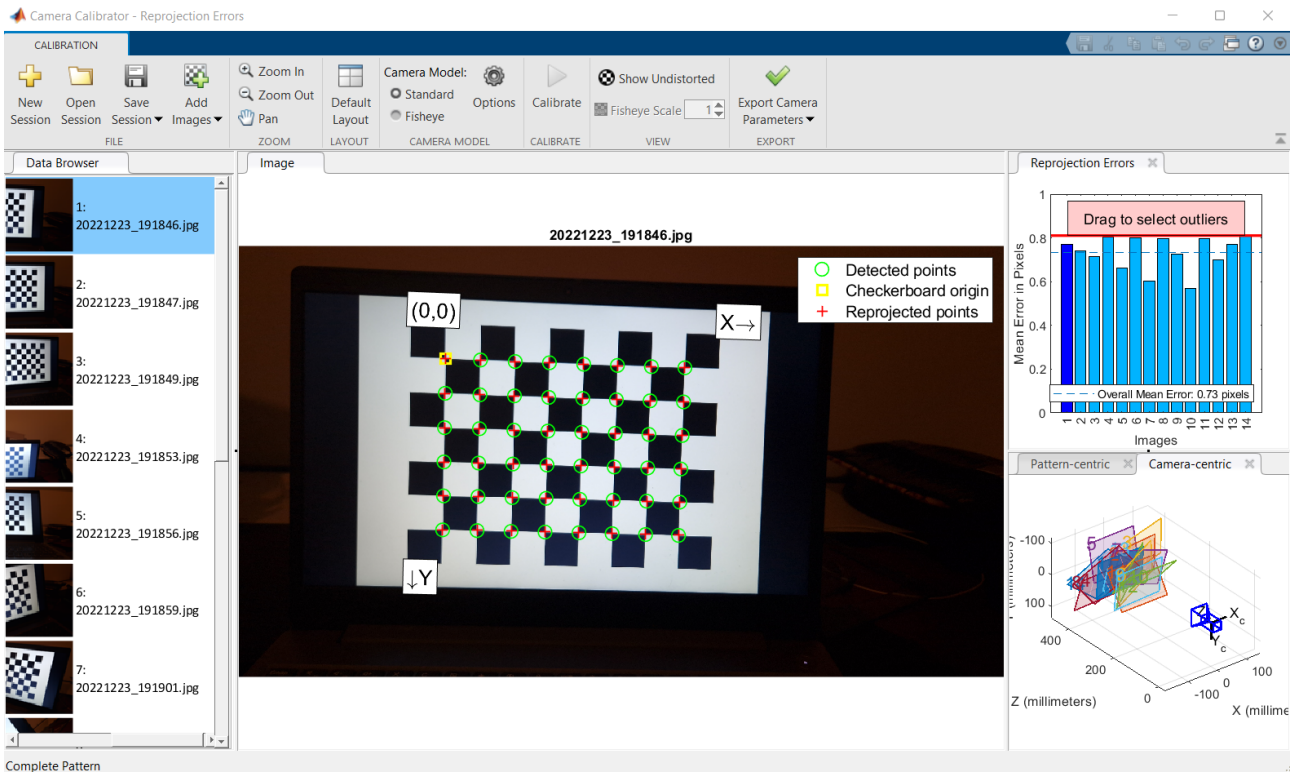

FIGURE 2 – Outil de Calibration

On sélectionne une série d'images d'un pattern (chessboard) avec differante position et angle avec la méme focale, puis on calibrate.

Une fois la calibration est terminée, nous pouvons accéder aux paramètres intrinsèque et extrinsèque de notre camera

Matrice intrinsèque :

$$K = \begin{pmatrix} \alpha_u & 0 & u_0 \\ 0 & \alpha_v & v_0 \\ 0 & 0 & 1 \end{pmatrix}_{3\times3}$$

Avec :
u_0 et v_0 : Le centre optique (le point principal), en pixels.
α_u et α_v : La distance focale.

Matrice extrinsèque :

$R_{3\times3}$: qui est la matrice de rotation permettant de passer du repère lié à l'espace de travail au repère lié à la caméra ;
t_x, t_y et t_z : qui sont les composantes du vecteur de translation permettant de passer du repère lié à l'espace de travail au repère lié à la caméra.

II. Homographie :

Une homographie est une transformation géométrique qui permet de passer d'une image à une autre en conservant les angles et en modifiant les distances et les proportions. Elle peut être utilisée pour décrire la relation entre un plan de scène et une image prise de ce plan de scène.

Une homographie entre deux points situés sur des plans p_a et p_b se décrit telle que :

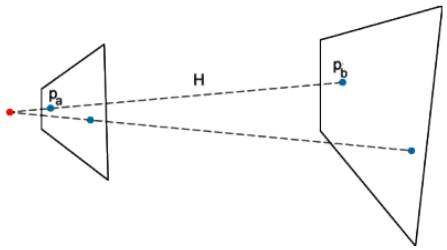

FIGURE 3 – Homographies entre un plan de scène et une image

$$p_b = Hp_a$$

Avec

$$H = \begin{pmatrix} h_1 & h_2 & h_3 \\ h_4 & h_5 & h_6 \\ h_7 & h_8 & h_9 \end{pmatrix}_{3\times3}$$

On prend :

$p_a = \begin{pmatrix} x_a \\ y_a \\ 1 \end{pmatrix}$ point repére image..

$p_b = \begin{pmatrix} x_b \\ y_b \\ 1 \end{pmatrix}$ point repére scene.

$$\begin{pmatrix} x_b \\ y_b \\ 1 \end{pmatrix} = \begin{pmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h \end{pmatrix} \begin{pmatrix} x_a \\ y_a \\ 1 \end{pmatrix}$$

Ayant 8 valeurs inconnues et nos coordonnées possédant chacune 2 dimensions, nous avons besoin d'au moins 4 points de correspondance pour résoudre notre système (afin d'avoir 8 équatons différentes).

Pour calculer une homographie entre un plan de scène et une image, voici les étapes à suivre :

1. On prend une image avec un pattern et on identifie au moins quatre points de correspondance entre le plan 3d et l'image. Ces points doivent être distincts et bien positionnés.
2. On a $p_b \times Hp_a = 0$
   Après quelques manipulations, on obtient :

$$\begin{pmatrix} 0 & 0 & 0 & -x_a & -y_a & -1 & y_b x_a & y_b y_a & y_b \\ x_a & y_a & 1 & 0 & 0 & 0 & -x_b x_a & -x_b y_a & -x_b \\ -y_b x_a & -y_b y_a & -y_b & x_b x_a & x_b y_a & x_b & 0 & 0 & 0 \end{pmatrix} H = 0$$

   seulement 2 equation sont linéairement independant.On combine les equations afin d'avoir autant d'equation que d'inconnus.

$$\begin{pmatrix} 0 & 0 & 0 & -x_a & -y_a & -1 & y_b x_a & y_b y_a & y_b \\ x_a & y_a & 1 & 0 & 0 & 0 & -x_b x_a & -x_b y_a & -x_b \\ & & & & & . & & & \\ & & & & & . & & & \\ & & & & & . & & & \\ 0 & 0 & 0 & -x_a & -y_a & -1 & y_b x_a & y_b y_a & y_b \\ x_a & y_a & 1 & 0 & 0 & 0 & -x_b x_a & -x_b y_a & -x_b \end{pmatrix} H = 0$$

   On obient une equatione $Ah = 0$ avec A la matrice d'equation connue.

3. Pour résoudre le problème on utilise la méthode DLT (Direct Linear Transformation), On doit d'abord calculer la SVD de la matrice A On utilise la SVD ($[U, S, V] = SVD(A)$).
4. Une fois qu'on a calculé la décomposition en valeurs singulières (SVD) de la matrice A et du'on a sélectionné la dernière colonne de la matrice V comme solution, on va mettre ce vecteur sous forme de matrice 3x3 en utilisant la commande reshape.puis on normalise cette matrice pour la rendre plus facile à interpréter et à utiliser dans d'autres calculs.

Autre utilisation de la matrice homographie La rectification d'image, images panoramique.

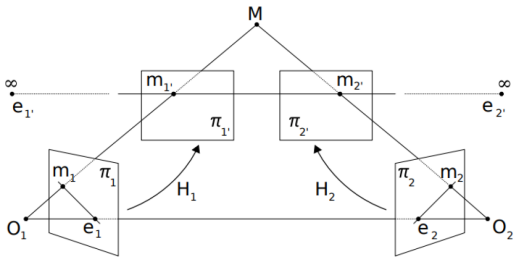

FIGURE 4 –

III. Estimation de la pose :

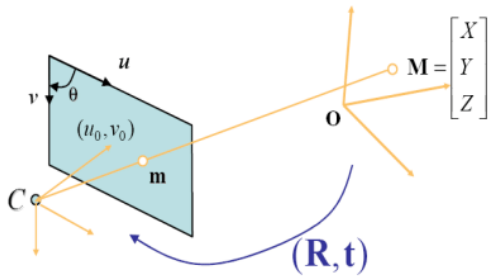

FIGURE 5 –

L'estimation de la pose d'un objet ou d'un système consiste à déterminer sa position et son orientation dans l'espace. Elle peut être utilisée dans de nombreux contextes, dans notre projet on va l'utilise pour faire de la réalité augmentée.

Pour avoir la matrice de projection il est nécessaire de disposer de points 3D dans cette scène. Or, notre mire est en 2D, ce qui signifie que nous n'avons pas d'informations sur l'axe z (profondeur) de ces points. Cela rend impossible le calcul de la matrice de projection, qui permet de transformer les points 3D de la scène en points 2D sur l'image capturée par la caméra. En l'absence de données sur l'axe z, nous ne disposons pas des informations nécessaires pour effectuer cette transformation.

Comme nous ne disposons pas de données sur l'axe z de notre mire 2D, nous ne pouvons pas directement calculer la matrice de projection qui nous permettrait de transformer les points 3D de la scène en points 2D sur l'image capturée par la caméra. Pour contourner ce problème, nous allons utiliser le fait que l'opération qui transforme les points du plan vers l'image est une homographie. nous pouvons utiliser cette homographie pour représenter la transformation des points 3D de la scène en points 2D sur l'image capturée par la caméra, en considérant que tous les points de la scène ont une coordonnée z égale à 0, Cela nous permettra de calculer la matrice de projection sans avoir à connaître les coordonnées z de notre mire 2D

Nous savons que la matrice de projection s'écrit :

$$P = K \begin{pmatrix} r1 & r2 & r3 & t \end{pmatrix}$$

Avec Z = 0 on a :

$$P = K \begin{pmatrix} r1 & r2 & t \end{pmatrix}$$

l'effet r3 est neutralisé par z = 0

Puisque P et H corresponde à la même transformation on peut ecrire :

$$H = \alpha K \begin{pmatrix} r1 & r2 & t \end{pmatrix}$$

α : est un facteur prés
on obtient :

$$\alpha \begin{pmatrix} r1 & r2 & t \end{pmatrix} = K^{-1}H$$

on defénit : Rtransi = $[r1 \;\; r2 \;\; r3]$
Avec : r1 : 1ere colone de $[K^{-1}H]$, r2 : 2eme colone de $[K^{-1}H]$ et r3 : r1 $\times$ r2

$\alpha = \frac{1}{det(Rtransi)^{\frac{1}{4}}}$
ce qui nous donne la matrice de projection :

$$P = K \begin{pmatrix} \alpha r1 & \alpha r2 & \alpha^2 r3 & \alpha t \end{pmatrix}$$

IV. tracking de primitives :

Le tracking de primitives consiste à suivre dans le temps l'évolution d'un ou plusieurs éléments simples (appelées "primitives") dans une séquence d'images. Ces primitives peuvent être des points, des segments, des cercles ou toute autre forme géométrique simple. Le tracking de primitives est souvent utilisé dans le cadre de la détection et du suivi de mouvement, de la reconstruction de scène ou de l'analyse de données. Il existe plusieurs méthodes pour effectuer le tracking de primitives, qui dépendent de la nature des primitives et des données disponibles. Voici quelques exemples de méthodes couramment utilisées :

- Si les primitives sont des points et que leur position est connue dans chaque image, il est possible de les suivre en utilisant des algorithmes de suivi de points tels que le filtre de Kalman ou le filtre de particules. Ces algorithmes permettent de prédire la position future des points en fonction de leur position passée et de leur mouvement observé.
- Si les primitives sont des segments ou des cercles, il est possible de les suivre en utilisant des algorithmes de suivi de contours. Ces algorithmes exploitent les propriétés des contours pour localiser et suivre les primitives dans l'image.
- Si les primitives sont des objets de forme complexe, il est possible de les suivre en utilisant des algorithmes de suivi d'objets. Ces algorithmes utilisent des modèles de l'objet pour le localiser et le suivre dans l'image.

Il est important de noter que le tracking de primitives peut être affecté par divers facteurs tels que le bruit de mesure, le mouvement de la caméra ou le mouvement des primitives elles-mêmes. Il est donc nécessaire de tenir compte de ces facteurs lors de la mise en œuvre de l'algorithme de tracking.

Dans le cadre de ce projet on a utilisé, Vision.PointTracker de Matlab qui permet de suivre des points dans une séquence d'images

Fonctionement : Pour utiliser cet outil, voici les étapes à suivre :

1. Sélectionner les points de référence à suivre dans l'image de départ : cela peut être fait en utilisant la fonction "Ginput", qui permet de cliquer sur l'image pour sélectionner les points de référence.
2. Initialiser le PointTracker : une fois les points de référence sélectionnés, il faut appeler la fonction "initialize" du PointTracker pour démarrer le suivi des points.
3. Suivre les points dans les images suivantes : une fois l'initialisation terminée, le PointTracker suivra automatiquement les points de référence dans les images suivantes. Pour obtenir les positions des points suivis dans chaque image, vous pouvez utiliser la fonction "track" du PointTracker.

3. Evaluation :

Cette partie présente les différence entre les resultat estimé par notre implementation et les résultats fournis par la calibration.

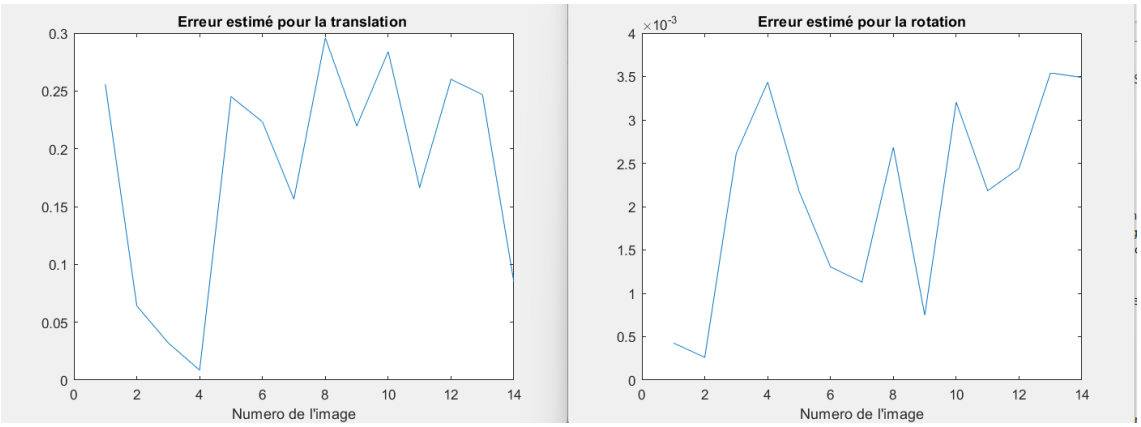

FIGURE 6 – Estimation d'erreur

Les erreurs dans la matrice de rotation et de translation calculée sont très minimes, cela signifie que l'algorithme fonctionne de manière efficace et précise.

INCRUSTATION DE L'OBJETS VIRTUELS :

La procédure consiste à suivre les points dans les images de la vidéo afin de calculer l'homographie, puis à utiliser ces informations pour calculer la matrice de projection et les matrices de rotation et de translation de la caméra. Enfin, l'objet 3D est projeté sur l'image en utilisant la matrice de Projection.

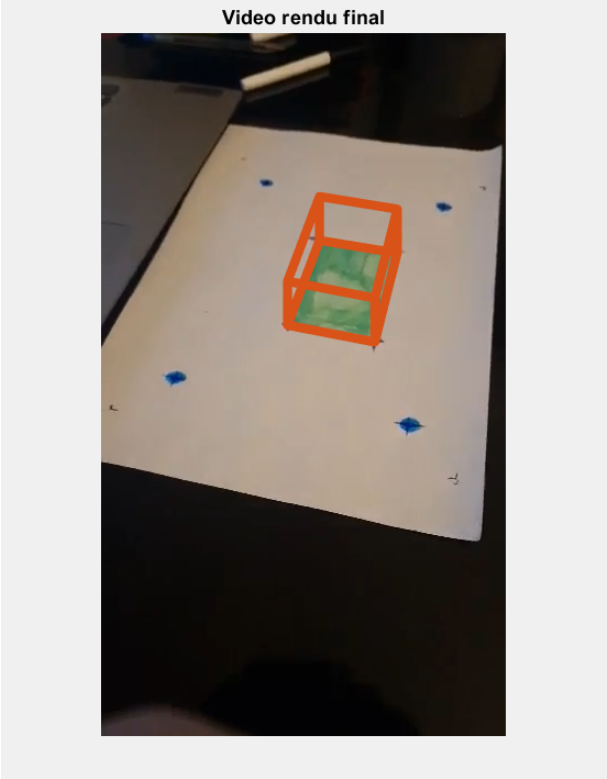

FIGURE 7 – Rendu de l'implémentation

4. Conclusion :

La vision par ordinateur est un terme qui désigne les différentes techniques permettant aux ordinateurs de comprendre le contenu d'une image. Il s'agit d'une sous-catégorie de l'IA (Intelligence Artificielle) et du Machine Learning.

Au cours de ce projet, nous avons utilisé la toolbox Matlab et appris les concepts de vision et de traitement d'image.

En conclusion, ce projet nous a permis d'apprendre de nombreux concepts et techniques liés à la vision par ordinateur et au traitement d'image. Nous avons utilisé la toolbox Matlab et avons acquis de solides connaissances sur les différents aspects de ce domaine, tels que la calibration de la caméra, l'homographie, l'estimation de la pose et les algorithmes de suivi de primitives.

Nous avons également appris à résoudre des problèmes de traitement d'image et à trouver des solutions aux problèmes imprévus.

En fin de compte, nous avons compris l'importance de la vision par ordinateur et les nombreuses possibilités qu'elle offre dans de nombreux domaines.

Dans l'avenir, il est probable que la vision par ordinateur soit de plus en plus présente dans nos vies quotidiennes et que de nouvelles applications et techniques soient développées et que toutes les machines soient capables de voir comme les êtres humains.

Références

1. Marie-Odile Berger, « Positionnement par l'image », https://members.loria.fr/moberger/Enseignement/Mines/pose_homog.pdf
2. J-p Tardif, Sébastien Roy, https://www.iro.umontreal.ca/~roys/ift6145H06/calib4.pdf
3. Multiple View Geometry in Computer Vision Second Edition. Richard Hartley and Andrew Zisserman, Cambridge University Press, March 2004.
4. Computer Vision, Prof. Raul Queiroz Feitosa, Pontifícia Universidade Católica do Rio de Janeiro, http://www.ele.puc-rio.br/~visao/
5. LARESI, « Estimation temps-real de la pose de la camera basee sur un marqueur code 2D », https://www.researchgate.net/publication/280644293_ESTIMATION_TEMPS-REEL_DE_LA_POSE_DE_CAMERA_BASEE_SUR_UN_MARQUEUR_CODE_2D