

## Rapport de projet Projection d'un objet en réalité augmentée

Hadriel RATIARISON

21109195

hadriel.ratiarison@etu.sorbonne-universite.fr

Walid Abdelaziz KHODJA

21214724

walid\_abdelaziz.khodja@etu.sorbonne-universite.fr

### I. INTRODUCTION

Dans le domaine de la vision par ordinateur, l'intégration de modèles virtuels 3D dans des scènes réelles représente un sujet d'étude intéressant et riche en applications, allant de la réalité augmentée à l'analyse avancée d'images. Ce rapport présente une solution visant à projeter un modèle 3D de la Tour Eiffel sur une série d'images prises avec notre smartphone. Cette approche combine des techniques de traitement d'image, telles que la détection de régions d'intérêt et le calcul d'homographie, avec des méthodes géométriques pour la projection 3D.

Ce travail se place dans un contexte où les scénarios d'alignement entre données virtuelles et réelles posent plusieurs difficultés, notamment la gestion des distorsions introduites par la caméra, la correction des orientations d'images, et la robustesse des algorithmes de projection face aux variations d'éclairage ou de perspective. En explorant ces aspects, ce rapport propose une méthodologie rigoureuse et évalue les performances de chaque étape, permettant la prise en main des méthodes relatives à la réalité augmentée.

### II. MÉTHODES

Pour pouvoir réaliser ce projet, il a fallu d'abord déterminer les paramètres intrinsèques et extrinsèques de la caméra.

#### A. Calibration

La calibration est le procédé qui permet de trouver la matrice intrinsèque  $K$ , qui regroupe les paramètres intrinsèques de la caméra, notamment la distance focale et les coordonnées du centre de l'image. Pour réaliser cette

calibration, nous avons pris 22 photos d'un échiquier sous différents angles qui fera guise de mire. Elles ont ensuite été traitées par le module `cameraCalibrator` de Matlab pour en détecter le motif. La Fig. 1a représente deux exemples d'images utilisées pour la calibration après détection des points. La Fig. 1b représente les positions de la caméra par rapport à l'échiquier pour chaque photo prise.

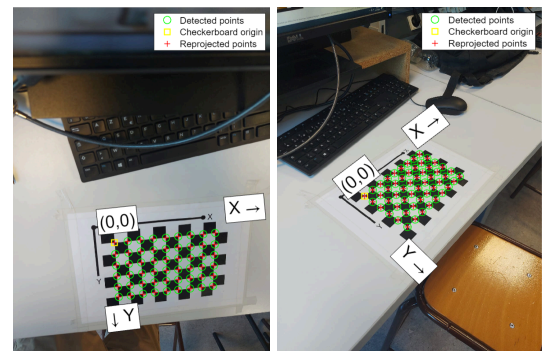


Figure 1a. Deux exemples d'images utilisées pour la calibration

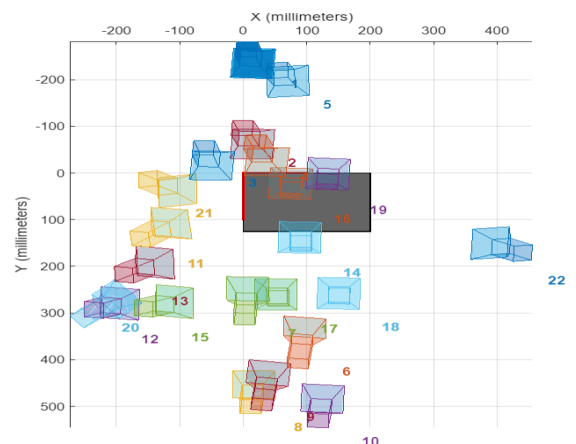
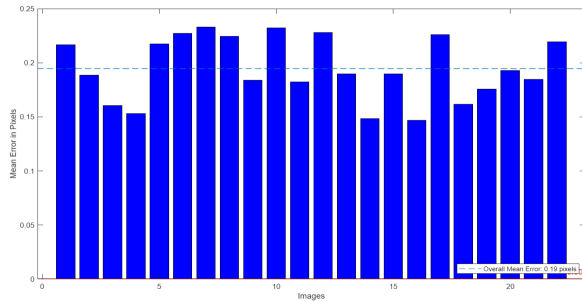


Figure 1b. Position de la caméra pour les 22 photos prises

Il est possible d'évaluer l'erreur de reprojection sur Matlab.

Le Fig. 2 représente l'histogramme d'erreur pour chaque image.



**Figure 2.** Histogramme des erreurs de reprojection pour chaque image.

Dans notre cas, cette erreur vaut 0.19 pixels et elle est donc négligeable.

La calibration nous a permis d'obtenir la matrice intrinsèque K, qui regroupe les paramètres suivants de la caméra :

- $f_x, f_y$  : distances focale de la caméra selon les axes x et y
- $C_x, C_y$  : coordonnées du centre de l'image en pixels

$f_x$	0	$C_x$
0	$f_y$	$C_y$
0	0	1

**Figure 3a.** Composition de la matrice intrinsèque K

Voici la matrice K obtenir grâce à la calibration :

1.1546e+03	0	594.4841
0	1.1537e+03	807.8156
0	0	1

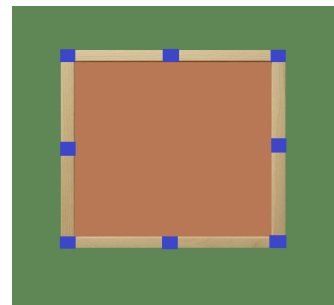
**Figure 3b.** Matrice intrinsèque K de la caméra

## B. Homographie

Le calcul de la transformation homographique constitue une étape fondamentale pour aligner les points d'intérêt réels et projetés.

En utilisant la technique Direct Linear Transform (DLT) et 8 paires de points avec leurs coordonnées dans le repère réel et le repère image, une matrice H a été estimée pour relier les coordonnées des points 2D dans le repère image avec celles des points de référence dans le repère objet.

L'extraction des points se fait grâce à une méthode de traitement d'image qui consiste à appliquer un masque sur l'image qui va mettre en valeur les points intéressants sous forme de regions dans une image binaire (en noir et blanc), puis les centres de gravité de ces régions seront extraits à l'aide de la fonction Matlab `bwlabel()` et d'une boucle.



**Figure 4.** Image de fond utilisée avant impression

Le processus d'extraction des points et de leurs coordonnées correspondantes est manuel seulement pour la première image, dans le but d'associer les points du repère image au repère monde. Étant donné que la fonction d'étiquetage retourne les points dans un ordre différent à chaque itération, un algorithme à été développé afin de faire correspondre chaque point à celui dont il est le plus proche sur l'image précédente.

L'algorithme fonctionne sans erreurs à condition que le décalage des positions entre deux photos consécutives soit relativement petit, sinon il y a un risque que les points de l'image soient associés aux mauvais points du repère monde.

A la fin de cette première étape, la matrice H en 2D est générée. Cependant, afin de réaliser l'application, la matrice H doit primordialement être en 3D. Il faut donc d'abord extraire la matrice de transformation homogène correspondante puis calculer et normaliser la troisième colonne manquante pour la rotation selon l'axe z en

utilisant le produit vectoriel entre la première et la deuxième colonne.

Facteur de normalisation :

$$R = [r1, r2, r3]$$

$$\alpha = \text{nthroot}(\det(R), 4)$$

Normalisation :

$$r1 = r1/\alpha$$

$$r2 = r2/\alpha$$

$$r3 = r3/\alpha$$

$$t = t/\alpha$$

Une fois que la matrice de transformation homogène en 3D est disponible, la matrice H en 3D peut être construite comme suit:

$$H = \alpha * k * \text{trans}_h$$

Avec k la matrice intrinsèque et trans\_h la matrice de transformation homogène.

A cette étape là, n'importe quel point peut être projeté dans le repère image en faisant une multiplication matricielle avec la matrice H en 3D.

### C. Estimation de la pose

La pose qui correspond à la matrice trans\_h est la matrice qui est incluse dans la matrice de projection H. Elle permet de passer d'un repère 3D monde à un repère 2D image. La pose est tout simplement la concaténation des matrices [r1,r2,r3] et de la matrice T. La matrice [r1,r2,r3] exprime la rotation de la caméra dans le repère monde tandis que la matrice T indique sa position. Les vecteurs r1,r2 proviennent directement de la matrice H en 2D. Quant au vecteur r3, il a été calculé comme produit vectoriel entre r1 et r2.

### D. Tracking de primitives

Comme mentionné plus haut, une méthode de traitement d'image a été utilisée afin de détecter les 8 carrés bleus qui sont nos points d'intérêts, comme on peut le voir sur Fig. 4. On va s'intéresser de plus près à la manière dont ces carrés bleus ont été détectés. Pour réaliser ce traitement, l'image a d'abord été convertie du format RGB vers HSV. Cela permet de créer un masque qui sélectionne les pixels de l'image selon sa teinte, sa saturation et sa valeur. Pour la création du

masque, les seuils de ces valeurs ont été définis et adaptés de manière à correspondre à la nuance bleu des carrés. Voici les seuils choisis :

- teinte : 0.55-0.75
- saturation > 0.3
- valeur (luminosité) > 0.2

Il suffit d'appliquer ce masque à l'image afin d'extraire les zones d'intérêt. Des opérations de fermeture et d'ouverture sont ensuite appliquées afin de nettoyer le masque. Ce traitement a été testé sur une première image (Fig. 5a). Le résultat (Fig. 5b) représente une image binarisée des carrés détectés par notre algorithme (en blanc).

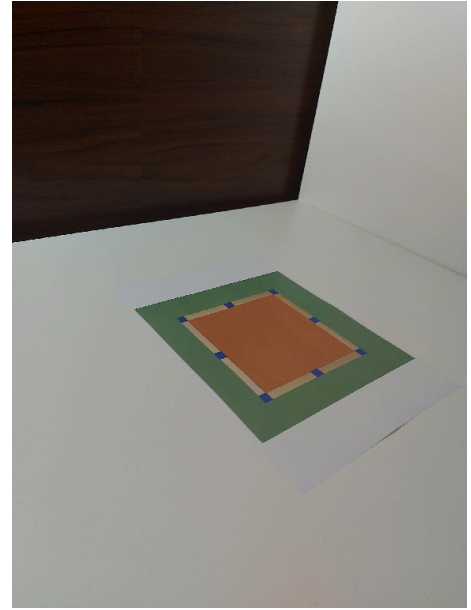


Figure 5a. Image de test pour la détection des carrés bleus

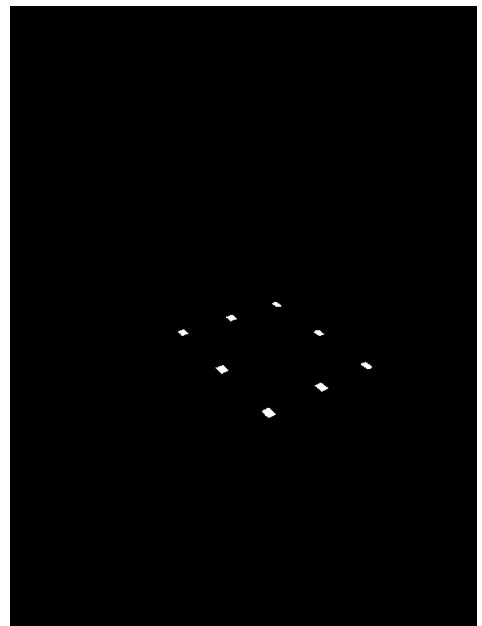


Figure 5b. Résultat de la détection des carrés bleus

On peut constater que les 8 carrés ont bien été détectés. Ensuite, pour pouvoir détecter les coordonnées des centres de gravité de chaque carré, on utilise la fonction `bwlabel()` qui permet d'extraire les coordonnées des pixels des carrés respectifs. Il est ensuite facile de calculer leurs centres de gravité en calculant la moyenne des coordonnées en x et en y des pixels de chaque carré.

On peut maintenant effectuer le tracking de primitives. Celles choisies sont les 4 coins du cadre, cela permet de visualiser clairement le mouvement de la caméra. La Fig. 4 représente l'évolution de la position de ces 4 coins dans le repère de l'image. Chacune des 4 couleurs représente un coin particulier.

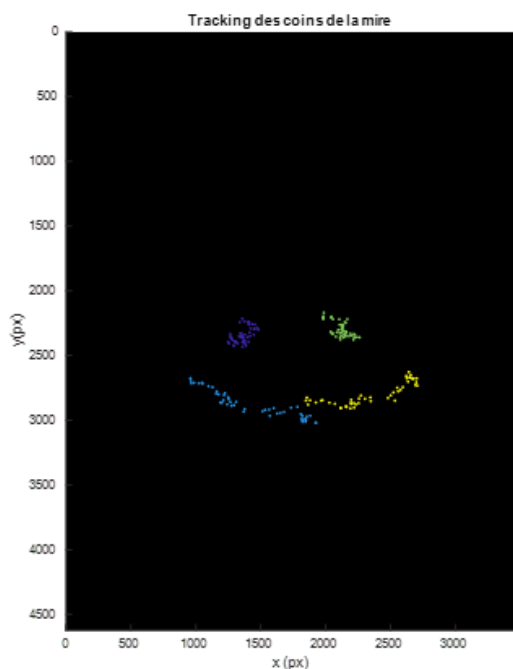


Figure 6. Tracking des 4 coins du cadre

Le déplacement des points nous montre bien une rotation de la caméra autour du cadre, conformément à ce qui a été fait. Cela nous montre que le tracking des primitives a bien été réussi.

### III. EXPÉRIMENTATION ET ÉVALUATIONS

#### A. Evaluation de l'homographie

Afin d'évaluer la précision et la robustesse des différentes méthodes de calcul d'homographie, deux approches ont été comparées, la méthode manuelle basée sur la transformation DLT (Direct Linear Transformation) utilisée dans ce projet et une méthode automatisée utilisant la fonction intégrée `fitgeotrans` de MATLAB.

Les deux techniques ont été appliquées à une série d'images où des points d'intérêt étaient détectés et utilisés pour calculer l'homographie. Ces projections ont ensuite été comparées aux positions réelles des points, permettant de quantifier les erreurs.

L'interprétation des résultats montre que la méthode intégrée de MATLAB offre une performance supérieure en termes de précision, avec une erreur moyenne inférieure par rapport à l'approche manuelle qui a une erreur moyenne de 4.8 pixels comme montré dans la Fig. 7 ci-dessous.

Erreurs moyennes :

Homographie manuelle : 4.8234 pixels

Homographie MATLAB : 3.1168 pixels

Figure 7. Résultats Matlab des erreurs moyennes

La Fig. 8 qui suit montre l'erreur estimée pour chaque point de chaque itération en utilisant les deux méthodes (manuelle et automatique).

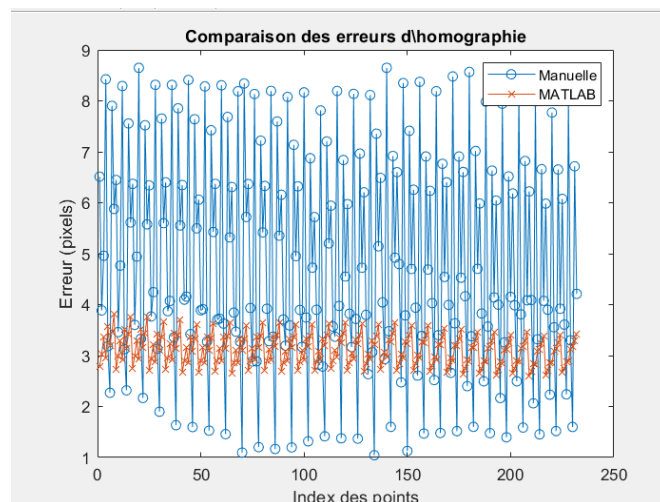


Figure 8. Calcul des erreurs d'homographie

Les résultats de cette figure confirment les erreurs moyennes calculées dans la section précédente.

Cette amélioration peut être attribuée aux algorithmes optimisés et aux mécanismes de régularisation intégrés dans les outils MATLAB, qui réduisent l'influence des erreurs de détection des points ou des bruits.

La méthode manuelle, bien qu'un peu moins performante, a permis de comprendre en détail le processus mathématique sous-jacent et a tout de même démontré une capacité à fournir des résultats acceptables dans des conditions contrôlées.

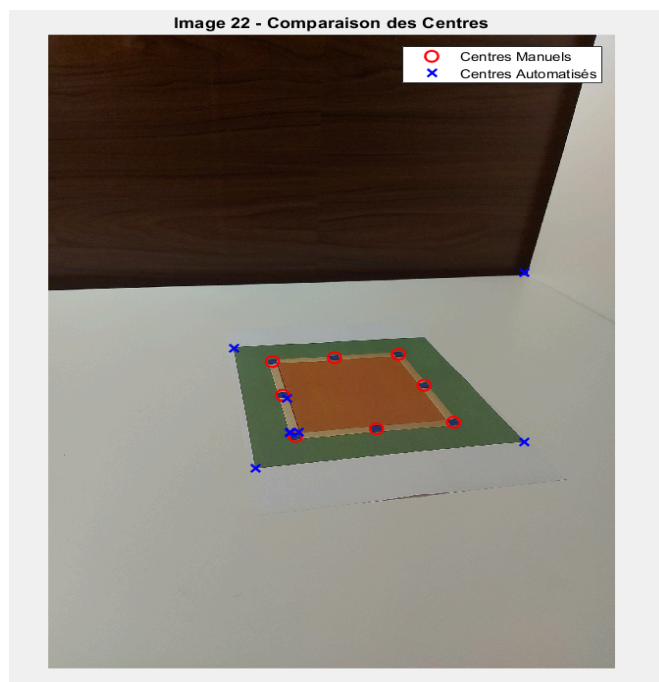
En conclusion, l'analyse comparative met en avant l'efficacité des outils intégrés pour les scénarios nécessitant une grande rapidité et précision. Néanmoins, la méthode manuelle reste un choix valable pour les projets nécessitant une personnalisation ou une adaptation particulière des algorithmes.

### B. Evaluation de la détection des régions bleues

Dans cette section, deux approches de détection des points d'intérêt dans les images seront comparées, une méthode manuelle basée sur la segmentation des couleurs dans l'espace HSV et une méthode automatisée utilisant des algorithmes intégrés de MATLAB, tels que `detectSURFFeatures`. L'objectif est d'évaluer la précision de détection des centres de gravité détectés par chaque méthode.

D'après les résultats obtenus, la méthode automatisée se montre plus faible en précision car dans certaines images elle ne détecte pas toutes les régions tandis que dans d'autres des mauvaises régions sont détectées.

La Fig. 9 ci dessous montre les résultats des deux méthodes sur une des 28 images utilisées.



**Figure 9.** Résultats de détection des centres de gravités

D'après ces résultats, la méthode privilégiée est celle utilisée durant le projet car elle permet de fixer les seuils manuellement ce qui implique une plus grande précision lors de la l'extraction de couleurs spécifiques.

## IV. DIFFICULTÉ RENCONTRÉES

Les difficultés qu'on a pu rencontrer sont les suivantes :

- La mise en pratique des notions théoriques : Le passage des concepts abstraits étudiés en cours à leur implémentation concrète en MATLAB a représenté un défi. Cela a nécessité de comprendre parfaitement les formules et les algorithmes pour les traduire correctement en code.
- L'identification des sources d'erreurs : Lorsqu'un résultat inattendu apparaissait, déterminer la source exacte du problème (mauvaise calibration, erreurs de seuils, ou approximation dans les calculs) a souvent été complexe.
- La validation des résultats : S'assurer que les résultats obtenus étaient non seulement cohérents mais aussi suffisamment précis pour répondre aux exigences du projet a nécessité une phase d'expérimentation et de validation rigoureuse.

## V. CONCLUSION

Ce projet a permis de démontrer une intégration efficace de techniques de vision par ordinateur pour la projection d'un modèle 3D sur des images réelles. En partant de la calibration des caméras jusqu'à la projection et au tracking des points d'intérêt, chaque étape a été exécutée soigneusement. Les résultats obtenus montrent une bonne concordance entre les projections et les points attendus, bien que certaines erreurs puissent survenir en raison des limitations liées à la capture des images.

En perspective, des améliorations pourraient inclure l'utilisation de méthodes d'apprentissage automatique pour une détection et un tracking plus robustes, ainsi que le développement de modèles prenant en compte les effets de l'éclairage et des ombres pour une représentation plus réaliste. Ce travail nous a permis d'acquérir une base solide en vision par ordinateur qu'on pourra exploiter dans des applications futures.