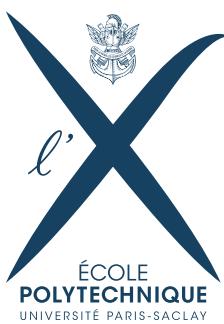




## INF552 Analyse d'images et vision par ordinateur : algorithmes et applications

17 janvier 2016

Ruoqi HE & Chia-Man HUNG



# 1 INTRODUCTION

Dans ce rapport, nous présenterons la réalisation d'une skybox à partir des photos prises par nous-même. C'est un procédé graphique permettant de donner, dans un espace tridimensionnel, l'illusion que cet espace est plus étendu qu'il ne l'est réellement. Notre méthode repose principalement sur l'estimation des matrices de rotation.

Pour représenter une vue panoramique de 360 degrés qui couvre toutes les directions, la méthode la plus simple est celle d'une skybox. Une skybox est constituée de six images : une par face du cube dont le milieu se trouve la caméra. L'objectif est donc de bien projeter les photos sur le cube afin de réaliser une skybox. La difficulté de la mise en œuvre réside d'abord dans l'obtention des photos émulant le panorama. Notre problème consiste ensuite à déterminer la rotation et le paramètre d'échelle associés à chaque photo.

# 2 ÉVOLUTION

## 2.1 PRISE DE PHOTOS

Nous voulons réaliser une skybox du Grand Hall et d'un casert. L'appareil que nous avons utilisé est Canon EOS 70D avec l'objectif Canon EF 28-135mm f/3.5-5.6 USM IS pour le Grand Hall, et Sigma 10-20mm f/4-5.6 EX DC HSM (objectif grand angle) pour le casert. Pour chaque endroit nous avons pris plusieurs tours de photos en variant l'angle d'élévation. Chaque tour de photos s'est composé d'une vingtaine de photos individuelles écartées d'un angle quasi-uniforme. Les photos sont prises dans une position fixe utilisant un trépied, avec la configuration manuelle identique, y compris la distance focale, l'ouverture, la vitesse d'obturation, ISO, et la balance des blancs (non auto). Les photos sont ensuite traitées par le logiciel Adobe Lightroom pour supprimer la distorsion en utilisant le fichier correspondant à l'objectif. Le fichier correspondant à Sigma 10-20mm f/4-5.6 EX DC HSM n'est pas assez satisfaisant. Nous avons eu recours au logiciel Adobe Lens Profile Creator pour la calibration manuelle. Les photos du Grand Hall sont aussi redimensionnées en 1280x960 pixels et celles du casert en 1920x1440 pixels.

## 2.2 PREMIER ESSAI

Notre première idée est de travailler uniquement sur l'homographie entre les images pour les projeter sur les six surfaces de la skybox. Nous avons repris une partie de travail du TD6

du cours qui permet de trouver l'homographie entre deux photos ayant une partie en commun et prises depuis la même position. Lorsqu'on l'applique à chaque paire de photos voisines dans un tour complète de photos, on observe que le produit de toutes les homographies n'est pas proche de l'identité – il existe en fait une très grande erreur accumulée. Il est a priori difficile de corriger cette erreur – simplement multiplier toutes les homographies par une puissance de la matrice d'erreur n'a pas de sens physique. Il est aussi difficile de projeter une photo sur le cube 3D connaissant seulement son homographie avec la photo de référence. Nous allons désormais chercher à connaitre la rotation de la caméra pour chaque photo utilisant les homographies.

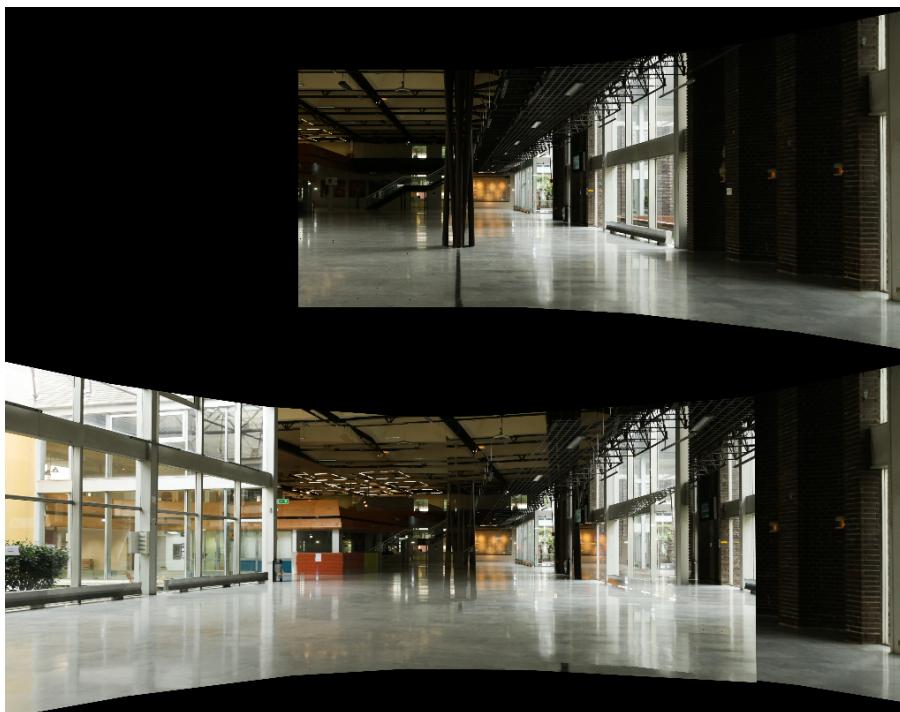


FIGURE 1 – Premier essai – la dernière photo ne colle pas bien à la première

## 2.3 ESTIMATION DE ROTATION

---

Notre deuxième idée consiste à trouver les rotations de la caméra pour chaque photo à partir des homographies entre les photos voisines, pour ensuite calculer la projection sur la skybox. Il existe une relation directe entre une homographie et une rotation, que nous expliquerons en détail dans 2.3.1. Toutefois, la rotation calculée à partir d'une homographie et de cette relation n'est pas toujours une matrice  $R$  vérifiant  $RR^T = I_3$  à cause des erreurs numériques dans l'estimation des homographies. Nous avons donc essayé de décomposer une matrice en une rotation, une translation et une dilatation. (Voir Référence [1].) Nous voulions trouver la rotation la plus proche. Dans l'annexe de Référence [1], étant donné une matrice quelconque  $M$ , la partie orthogonale de la décomposition polaire est la rotation la plus proche au sens de

Frobenius. Cette partie orthogonale peut être obtenue à partir de la décomposition en valeurs singulières, que nous expliquerons dans 2.3.2.

### 2.3.1 • RELATION HOMOGRAPHIE-ROTATION

Étant données deux photos ayant une partie en commun, que l'on les numérote 1 et 2 respectivement, on a les relations suivantes.

- (1)  $H_{1/2}m_1 = m_2$
- (2)  $R_1 M_{|1} = R_2 M_{|2} = M$
- (3)  $R_{2/1}M_{|2} = M_{|1}$

$$(4) K_i[R_i^{-1}|0]M = m_i, i \in \{1, 2\} \text{ où } K_i = \begin{pmatrix} \alpha_i & 0 \\ & \alpha_i & 0 \\ & & 1 \end{pmatrix}$$

puisque ici on considère le centre de photo comme l'origine des coordonnées pixels ( $m_i$ ). Les relations ci-dessus sont en coordonnées homogènes.

Notations :

$m_i, i \in \{1, 2\}$  est un point de la photo  $i$  dans les coordonnées pixels de la photo  $i$ .

$H_{1/2}$  est l'homographie d'un point dans les coordonnées pixels de la photo 1 au même point dans les coordonnées pixels de la photo 2.

$M_{|1}, M_{|2}$  et  $M$  sont le même point dans le référentiel de la caméra 1 et 2 et dans le référentiel universel respectivement.

$R_i, i \in \{1, 2\}$  est la rotation de la caméra  $i$  dans le référentiel universel.

$R_{2/1}$  est la rotation de la caméra 2 dans le référentiel de la caméra 1.

$K_i, i \in \{1, 2\}$ , donné dans le cours, caractérise la caméra  $i$ .

$\alpha$  peut être caractérisé par la distance en pixels entre le centre optique et la rétine virtuelle. Lorsqu'on augmente  $\alpha$ , on diminue le champ de vue.

Solution des équations :

$$H_{1/2} = \begin{pmatrix} 1 & & \\ & 1 & \\ & & \alpha_2^{-1} \end{pmatrix} R_{2/1}^{-1} \begin{pmatrix} 1 & & \\ & 1 & \\ & & \alpha_1 \end{pmatrix}$$

Dans l'implémentation, on prend le référentiel de la caméra associée à la première photo du premier tour de photos horizontal (central) comme le référentiel universel.

Estimation de  $\alpha$  :

On suppose que  $\alpha$  est le même pour toutes les photos. On peut exprimer la rotation  $R$  en fonction de  $\alpha$  par la relation homographie-rotation en connaissant l'homographie. Pour que  $R$  soit une rotation,  $R$  doit vérifier  $RR^T = I_3$ , qui peut être traduit par une équation de  $\alpha$ . Cette équation nous sert à calculer  $\alpha$ . Après l'avoir mise en œuvre, on observe que les  $\alpha$  calculés

variant beaucoup (environ de 10%) d'une homographie à l'autre. On a finalement testé de différents  $\alpha$  pour trouver un  $\alpha$  tel que l'image transformée par l'homographie correspondante de la dernière photo du premier tour horizontal colle au mieux la première photo du même tour.

### 2.3.2 • DÉCOMPOSITION EN VALEURS SINGULIÈRES

Théorème :

Soit  $M$  une matrice  $n \times n$  à coefficients réels. Alors il existe une factorisation de la forme :

$$M = U\Sigma V^T$$

avec  $U$  une matrice unitaire  $n \times n$ ,  $\Sigma$  une matrice  $n \times n$  dont les coefficients diagonaux sont des réels positifs ou nuls et tous les autres sont nuls, et  $V^T$  est la matrice transposée de  $V$ , matrice unitaire  $n \times n$ . On appelle cette factorisation la décomposition en valeurs singulières de  $M$ .

La rotation  $R$  la plus proche de la matrice au sens de Frobenius est donnée par la relation suivante.

$$R = UV^T$$

Malheureusement, l'homographie obtenue à partir de cette rotation ne permet pas de bien coller deux photos voisines. Nous nous sommes rendu compte que la rotation la plus proche au sens de la norme Frobenius de matrice n'est pas forcément la meilleure solution pour joindre deux photos.

### 2.3.3 • MÉTHODE INTUITIVE

Nous voulons une méthode plus compréhensible et contrôlable pour estimer la rotation. Nous avons donc utilisé l'idée suivante : Si on connaît l'homographie  $H$  entre deux photos, alors on peut projeter le centre de la deuxième photo sur la première. Si on connaît en plus le coefficient  $\alpha$  de la caméra, on peut trouver la direction  $V_c$  correspondant à ce centre dans le référentiel 3D de la première photo  $Ref_1$ . Cette direction est la direction vers l'avant ( $+z$ ) de la caméra pour la deuxième photo. On fait le même pour un point 500 pixels à droite du centre dans la deuxième photo ; cela correspond à une autre direction  $V_d$  dans  $Ref_1$ . La direction vers la droite ( $+x$ ) de la caméra pour la deuxième photo est donc parallèle au plan engendré par  $V_c$  et  $V_d$ . On peut la calculer en utilisant la condition qu'elle est orthogonale avec la direction vers l'avant. Ainsi les trois axes de la camera sont connus, la matrice de rotation  $R_{2/1}$  est donc construite en mettant les coordonnées des trois axe dans chaque colonne. Voir Figure 2.

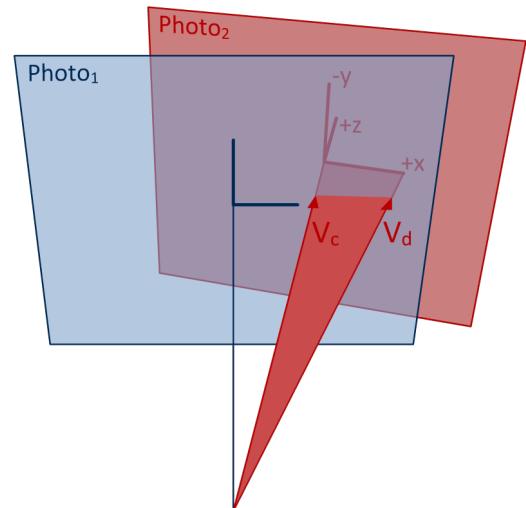


FIGURE 2 – Déduction de rotation

Avec cette méthode, nous avons progressivement calculé la rotation pour toutes les photos. Ensuite, nous les avons retransformées en homographie pour corriger la projection des photos. Nous observons qu'en utilisant l'homographie corrigée pour la projection, les photos se collent bien. Néanmoins, il existe une erreur plus visible entre deux photos voisines par rapport à l'homographie originale.

## 2.4 CORRECTION DE ROTATION

---

Nous appliquons la méthode ci-dessus pour calculer les rotations pour le premier tour horizontal de photos. Quand on finit le tour et on revient à la première photo, la rotation est encore différente de l'identité à cause des erreurs accumulées. Les erreurs ne proviennent pas que de l'estimation incorrecte de  $\alpha$  car il existe aussi un décalage vertical. Mais cette fois-ci la correction de la rotation est facile à effectuer. Si  $R_{fix}$  est la rotation de correction à effectuer sur la dernière photo (qui est elle-même la première photo), les corrections à effectuer pour les photos intermédiaires sont donc une portion de cette rotation – c.à.d. une interpolation entre l'identité et  $R_{fix}$ . Cela peut être calculé à l'aide de la fonction Slerp du Quaternion :

$$\text{Interpolation}(R_1, R_2, t) = Q^{-1}(\text{Slerp}(Q(R_1), Q(R_2), t))$$

où  $R_1$  et  $R_2$  sont les matrices de rotation que l'on veut interpoler.

Slerp est l'interpolation linéaire sphérique dans l'espace des quaternions.

$Q$  est la fonction qui transforme une matrice de rotation en un quaternion et  $Q^{-1}$  la transformation inverse. Voir Référence [2] pour les détails.

Pour chaque photo, on utilise l'angle horizontal parcouru pour déterminer le coefficient d'interpolation utilisé. Après cette correction, nous obtenons déjà un bon panorama pour le premier tour horizontal. (Voir Figure 4.) Néanmoins, vu que chaque photo peut avoir une distance de mise au point différente qui change légèrement le champ de vue, et que le centre optique peut avoir des déplacements avant-arrière, le coefficient  $\alpha$  peut varier selon les photos. Nous avons donc décidé de fixer /alphaniquement pour la première photo du premier tour, et en déduire sa valeur pour toutes les autres photos. Si on connaît  $\alpha$  pour une photo et son homographie avec une autre photo, on peut d'abord projeter le centre de

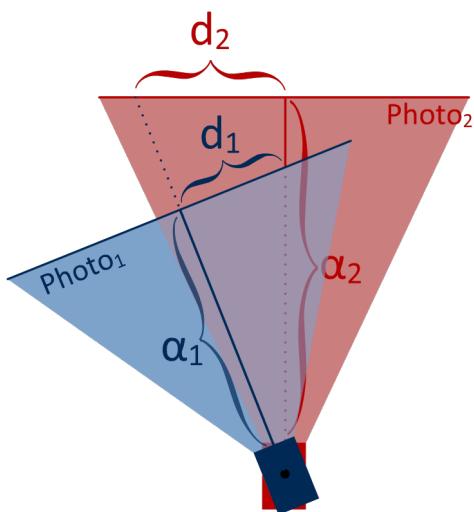


FIGURE 3 – Déduction de  $\alpha$

chaque photo sur l'autre photo. La distance entre les deux centres peut éventuellement être différente dans les deux photos. Le rapport de  $\alpha$  peut alors être déduit par le rapport de ces deux distances. Voir Figure 3.

La valeur de  $\alpha$  a aussi une erreur accumulée – la première photo n'a plus le même alpha après le tour. On applique donc aussi une correction de alpha pour chaque photo dans le tour.



FIGURE 4 – Premier tour horizontal de photos

## 3 SOLUTION FINALE – RÉALISATION

Pour compléter notre skybox, on construit d'abord le tour de photos horizontal. Puis on l'étend couche par couche en haut et en bas. Pour chaque nouvelle photo ajoutée on calcule son homographie avec la photo la plus proche dans la couche précédente, et en déduire sa rotation. Après que les rotations de toutes les photos sont calculées, on veut alors les projeter sur les six faces carrées de la skybox. Les rotations des faces de la skybox sont connues ; leur alpha est égal à la moitié de sa largeur car leur champ de vue est de 90 degrés. En utilisant la relation dans 2.3.1, on peut alors calculer l'homographie d'une photo avec les faces de la skybox, ainsi réaliser la projection.

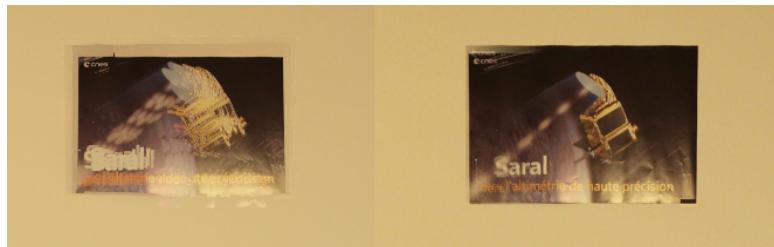


FIGURE 5 – Choix de couleur – gauche : valeur maximale / droite : Z-Buffer + interpolation

Puisque plusieurs photos peuvent se superposer au même endroit, il faut décider comment choisir la couleur utilisée pour la skybox. Nous avons d'abord choisi d'utiliser la valeur maximum de couleur pour chaque pixel. La skybox générée par cette méthode a une apparence assez satisfaisante, et elle nous permet de bien observer l'erreur dans la positionnement de chaque photo. Mais pour rendre l'erreur moins visible et pour créer un résultat plus beau, on a choisi à la fin d'utiliser une sorte de Z-Buffer. L'idée c'est de poser des rétines virtuelles des photos au tour de la caméra, et on voit l'image « le plus proche », comme illustré dans la figure 6. Pour réaliser cela, on crée une carte de profondeur en plus (Z-Buffer) pour chaque face de la skybox. Pendant la projection, le pixel ayant la plus petite profondeur est retenu. On améliore encore le résultat en implémentant une zone d'interpolation. Quand le nouveau pixel a une profondeur proche de celui du pixel existant, on interpole ses couleurs selon la différence de profondeur. Voir Figure 5 pour la comparaison et Figure 6 pour le principe de Z-Buffer et d'interpolation.

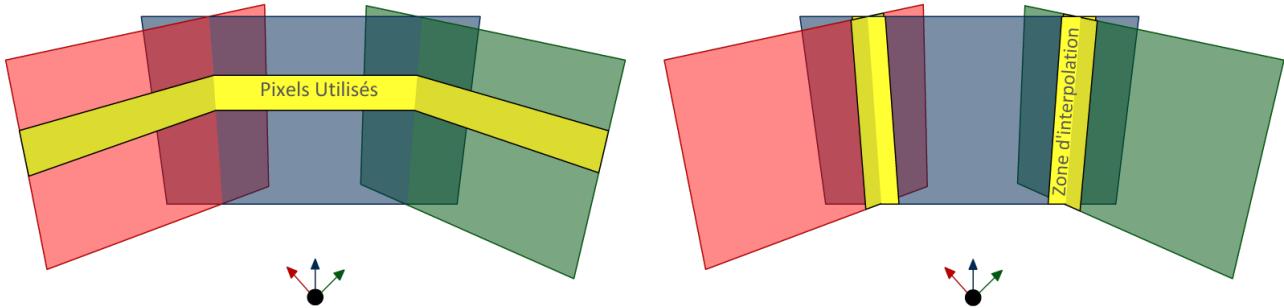


FIGURE 6 – Collage de photos par Z-Buffer et interpolation

Quand on construit des différentes couches de photo, nous nous sommes rendu compte qu'il est difficile de trouver l'homographie pour beaucoup de photos. On a donc utilisé deux séries de photos dans les calculs. Une série contient les photos normales pour mettre sur la skybox, tandis que l'autre série sont les même photos mais fortement contrastées, utilisée pour trouver les homographies. Les photos fortement contrastées contiennent beaucoup plus de points caractéristiques, ce qui favorise l'estimation d'homographie.

Néanmoins, il se peut que l'homographie ne puisse pas être trouvée entre une paire de photos malgré tous nos efforts. Dans ce cas nous essayons d'abord de changer la paire de photo pour l'estimation de l'homographie, et si cela ne marche pas non plus, on ne la considère plus dans notre skybox. Normalement si localement il manque une seule photo, le résultat sera encore acceptable parce que les zones manquantes peuvent être remplies par les photos voisines.

## 4 RÉSULTAT

---

Grand Hall :

Voir Figure 7.

Au total, nous avons utilisé 204 photos.

Casert :

Voir Figure 8.

Au total, nous avons utilisé 59 photos. Le nombre de photos est largement réduit puisque cette fois-ci, nous avons utilisé un objectif grand angle. Le problème de distorsion est plus grave, même après la correction.

Une fois les six faces du cube obtenues, on visualise la skybox générée à l'aide d'Unity. Voir les démos.



FIGURE 7 – Les six faces de la skybox du Grand Hall

## 5 PROBLÈMES RESTANTS

---

Il reste encore quelque problèmes liés à notre méthode ou liés au matériel.

- Comparaison de deux photos sans point caractéristique (Akaze dans notre cas)  
Pour le plafond, le mur, le sol, le ciel, etc, il existe souvent très peu de points caractéristiques, voire sans aucun point caractéristique, même si on augmente le contraste des photos. On ne peut donc pas appliquer la méthode Ransac pour trouver l'homographie entre deux photos voisines. Ainsi, notre méthode, basée sur l'homographie, ne fonctionne pas.
- Erreurs accumulées  
Il y a des erreurs partout et les erreurs sont malheureusement accumulées. Citons par exemple la rotation de la dernière photo du premier tour horizontal. Même si on la fixe par une matrice d'erreur (Voir Section 2.4), les autres rotations sont faussées dans un certain sens.
- Paramètre à ajuster

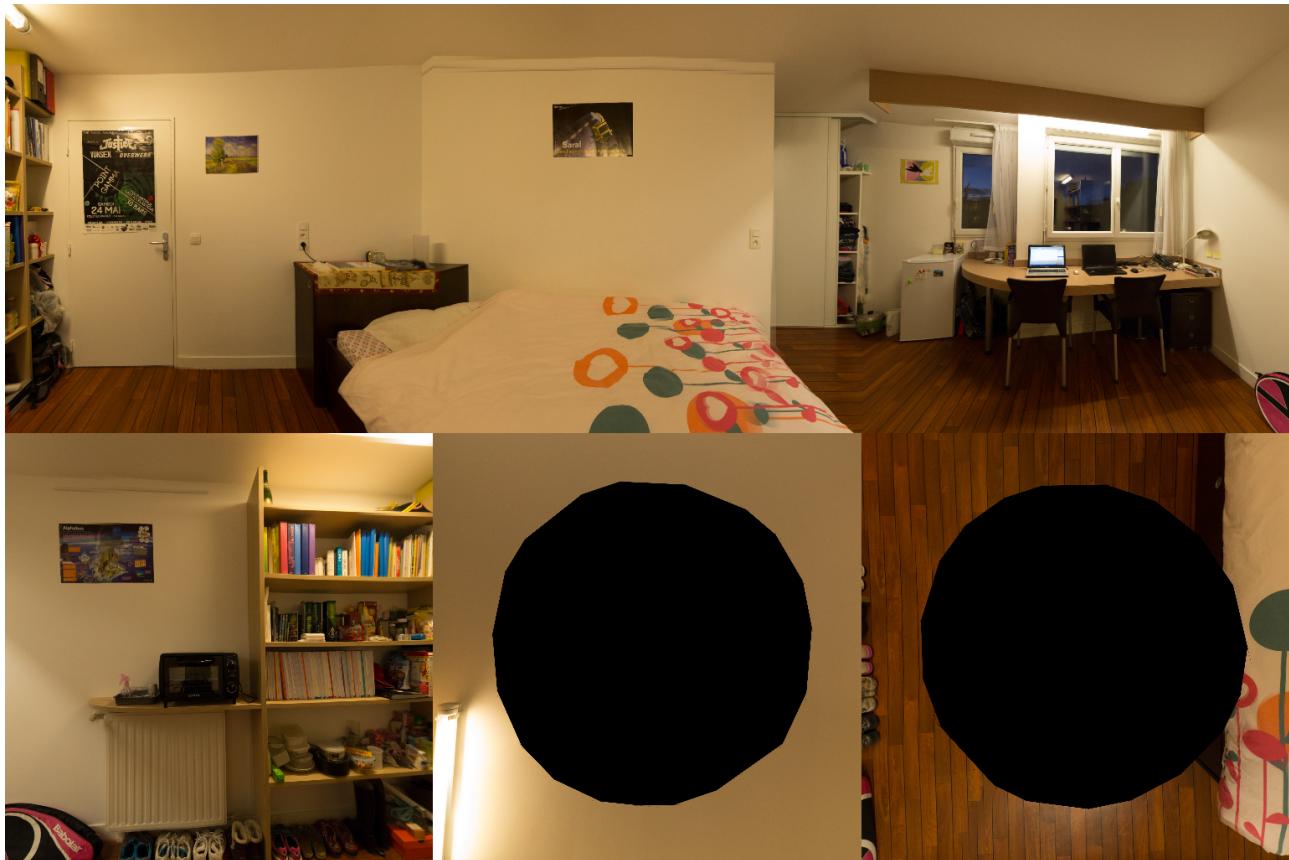


FIGURE 8 – Les six faces de la skybox d'un casert

La valeur  $\alpha$  de la première photo est à ajuster manuellement.

— Centre optique de la caméra non fixé

Voir Figure 9. Lorsque nous tournons la caméra sur le trépied pour prendre des tours de photos, le centre optique de la caméra n'est pas un point fixe, alors que nous n'avons pas pu prendre en compte ce décalage inconnu dans notre calcul. Plus précisément, le centre optique se situe au milieu de l'objectif, qui tourne lui-même au tour du centre de rotation du trépied.

— Distorsion

Même après la correction de distorsion du logiciel Adobe Lightroom et la calibration manuelle à l'aide du logiciel Adobe Lens Profile Creator, la distorsion est moins grave mais elle existe toujours.



FIGURE 9 – Centre optique de la caméra non fixé

## 6 CONCLUSION

Nous avons réussi à déterminer la rotation de nos photos, ce qui est au cœur de notre problème, grâce à la relation homographie-rotation et l'homographie calculée entre deux photos voisines par la méthode de Ransac. Cette rotation est ensuite corrigée pour annuler le décalage du premier tour horizontal en interpolant une matrice d'erreur par une fonction de Slerp. Une fois la rotation fixée, nous avons projeté les photos sur les six faces du cube. Le collage de photos est améliorée par un Z-Buffer et une interpolation. Notre résultat ainsi obtenu est plutôt satisfaisant. Les démos d'Unity sont très réalistes. Néanmoins, la face du haut et celle du bas ne sont pas complétées à cause de la limite du matériel et quelques petits défauts peuvent être observés. Nous avons pensé à traiter ce problème par la programmation par contraintes. En bien définissant la fonction objective (la somme des erreurs entre les photos voisines) à optimiser et les contraintes du problème, nous pourrons espérer un résultat encore plus satisfaisant. A ce stade, les équations ne sont pas linéaires et il est encore difficile de les résoudre.

## 7 RÉFÉRENCES

- [1] Shoemake, K., & Duff, T. (1992, May). Matrix Animation and Polar Decomposition. In *Proceedings of the conference on Graphics interface* (Vol. 92, No. 1992, pp. 258-264).
- [2] van Waveren, J. M. P. (2005). From Quaternion to Matrix and Back.