Travaux pratiques
ESIR 2 V3D Vision 3D

TP1 à TP4
2023 – 2024

Documents à rendre

- Un code source à trou est fourni en Python. Ils sont disponibles à l'adresse suivante :
  - http://people.irisa.fr/Eric.Marchand/
- Les documents (compte-rendu des 3 TP et code source complété) sont à rendre marchand@irisa.fr et thibault.noel@inria.fr au plus tard deux semaines après la dernière séance encadrée sous la forme d'une archive zip
- Le compte-rendu doit répondre aux questions posées dans les sujets et être illustré avec des images issues des traitements que vous avez réalisés. Attachez une importance particulière à décrire ce que vous avez fait, expliquer pourquoi et comment vous l'avez fait, décrire les résultats obtenus, les analyser, faire le lien entre ces résultats et le cours.
- Le code source complété doit contenir des commentaires qui doivent aider à comprendre l'algorithme développé.
- Les développements se feront en Python et seule la librairie Numpy devra être utilisée.

TP1 : géométrie épipolaire

L'objectif de ce TP est de se focaliser sur les éléments de base de la géométrie épipolaire.

A l'aide des fonctions fournies dans le fichier utils.py, dans le fichier utils_to_complete.py compléter à partir du cours les fonctions utilitaires suivantes :

- inverseHomogeneousMatrix()
- multiplyHomogeneousMatrix()
- skew()

Soit un système stéréoscopique donnant deux images I1 et I2. La calibration des caméras est connue.
On supposera dans un premier temps que la matrice de calibration est donnée par :

$$\mathbf{K} = \begin{pmatrix} 800 & 0 & 200 \\ 0 & 800 & 150 \\ 0 & 0 & 1 \end{pmatrix}$$

Question 1 : A quoi correspondent les valeurs dans la matrice **K** ?

Nous supposerons que la caméra c2 est positionnée à la position c2Tw définie par (voir Figure 1) :

$${}^{c2}\mathbf{T}_w = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 2 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

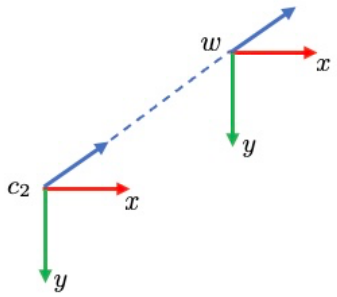

Figure 1 : Disposition du repère de la caméra 2 (c2) et par rapport à celui du monde (w).

Cas 1 : La caméra c1 est positionnée 10 cm à gauche de c2.

Question 2 : Donner la matrice c1Tw. En utilisant le squelette de code mis à votre disposition, compléter la matrice c1Tw.

Question 3 : Comment appelle-t-on un tel système et préciser ?

On souhaite pouvoir faciliter la mise en correspondance de points dans les deux images. Pour cela, on s'attachera à représenter le lieu géométrique d'un point x1 dans l'image I1 où son correspondant x2 dans I2 peut se situer.

Question 4 : Caractériser ce lieu. Calculer son équation. Dans le rapport, donner les coordonnées des points x1 pour les points x2 (100,100) et (50, 75).

Question 5 : Afficher les points x2 dans I2 et les lieus calculés précédemment dans I1. Vérifier que vous obtenez bien le résultat attendu et donner l'image obtenue.

Cas 2 : Positionner maintenant la caméra c1 20 cm devant c2.

Question 6 : Donner la nouvelle matrice c1Tw.
Question 7 : Quelle est la position de l'épipôle ?
Question 8 : Refaire pour cette nouvelle position les questions 4 et 5.

Cas 3 : Positionner maintenant la caméra c1 en c1Tw tel que la translation vaut (0.1, 0, 1.9) et la rotation en degrés (5, 5, 5) en utilisant la représentation minimale angle-axe aussi appelée theta-u.

Question 9 : Donner la nouvelle matrice c1Tw.
Question 10 : Donner la nouvelle position des épipôles.
Question 11 : Refaire pour cette nouvelle position les questions 4 et 5.

TP2 : Estimation d'une homographie par DLT

L'objectif de ce TP est d'estimer une homographie à l'aide de N points mis en correspondance. Pour cela vous disposer de deux images I1 et I2. Sur chacune de ces deux images 5 points sont présents et bien visibles. Les coordonnées des 5 points mis en correspondance dans les deux images sont données par les couples (u1i, v1i) dans l'image 1 et (u2i, v2i) dans l'image 2.

Question 1 : Combien de points sont nécessaires pour calculer une homographie ?

Question 2 : Connaissant un ensemble de N point mis en correspondance (u1i,v1i) et (u2i,v2i), i = 1...N, compléter la fonction DLT() qui estime l'homographie c1Hc2.

Question 3 : Comment juger de la qualité de votre estimation ?

Question 4 : En prenant les points (u1i, v1i), les transférer dans l'image 2 (u2'i, v2'i) et les afficher.

Pour chaque point, calculer la norme de l'erreur (ou résiduel) entre le point transféré (u2'i, v2'i) et le point (u2i, v2i) donné.

Quelle est l'unité de la norme de l'erreur ?

Afficher un cercle centré sur (u2'i, v2'i) de rayon 100 fois l'erreur commise.

Calculer la moyenne du résidu correspondant au transfert de tous les N points.

Question 5 : Transférer tous les pixels de l'image 2 dans l'image 1 en utilisant la même méthode. Vous devriez avoir votre première mosaïque.

Commenter le résultat de la mosaïque.

TP3 : Estimation d'une homographie par RANSAC

L'objectif de ce TP est d'estimer une homographie à l'aide de N points mis en correspondance automatiquement à l'aide d'un algorithme de type SIFT. Parmi les paires de points mis en correspondance, l'algorithme utilisé peut rendre de faux appariements. Il s'agit ici de développer une méthode d'estimation de l'homographie à l'aide d'un algorithme de type RANSAC robuste aux mauvais appariements.

Dans ce TP vous disposer de deux images I1 et I2. Sur chacune de ces deux images, les coordonnées de N points appariés dans les deux images sont données par les couples (u1i, v1i) dans l'image 1 et (u2i, v2i) dans l'image 2 avec i = 1...N.

Question 1 : Faites afficher les mises en correspondance. Que constatez-vous ?

Question 2 : Appliquer l'algorithme DLT d'estimation de l'homographie du TP2 et tracer un cercle correspondant à la norme de l'erreur.

Les erreurs vous semblent elles raisonnables ?

Question 3 : Implémenter un RANSAC pour estimer l'homographie. Les erreurs sont-elles maintenant raisonnables ?

Quels sont les appariements considérés comme aberrants ?

TP4 : Reconstruction 3D

Vous trouverez dans le répertoire data plusieurs images rectifiées tp4-1-left.png et tp4-1-right.png ou tp4-2-left.pgm et tp4-2-right.pgm ou tp4-3--left.jpg et tp4-3-right.jpg.

L'objectif de ce TP est de calculer une carte de disparité de l'environnement uniquement pour les couples d'images rectifiées.

Question 1 : Verifier qualitativement que les images sont bien rectifiées. Expliquer comment vous faites.

Question 2 : Reconstruire l'environnement en utilisant le critère ``Winner takes all''.

Pour mémoire le critère à minimiser est donné par :

$$E_{WTA}(\Delta\mathbf{u}) =\mid I_2(\mathbf{x}_i + \Delta\mathbf{u}) - I_1(\mathbf{x}_i) \mid$$

Question 3 : Même question en utilisant la ``Sum of Squared Differences'' ou SSD. Vous utiliserez différentes tailles des de fenêtres (1, 3, 7, 20) et différents noyaux (dont un Gaussien).

Pour mémoire le critère à minimiser est donné par:

$$E_{AC}(\Delta\mathbf{u}) = \sum_i w(\mathbf{x}_i)[I_0(\mathbf{x}_i + \Delta\mathbf{u}) - I_0(\mathbf{x}_i)]^2$$

Question 4 : Comparer les différentes approches.