

Copier les données des TPs

1. créez un répertoire V3D dans votre homedir

```
cd ~/
mkdir V3D
cd V3D
```
2. Depuis la dropbox ESIR2, copier le dossier contenant toutes les données pour le TP
TP-code.zip

Installation de la librairie VISP et création du projet

La librairie VISP est utilisée. Elle doit être installée sur vos postes de travail. Les sources, ainsi que les APIs de cette librairie sont disponibles à l'adresse suivante :

<http://www.irisa.fr/lagadic/visp/visp.html>

Documents à rendre

Le compte rendu de TP seront envoyés par mail à :

- `eric.marchand@irisa.fr`

sous la forme d'une archive contenant votre rapport ainsi que vos fichiers sources :

- `tar -cvzf tpV3Dtp1_Etudiant1_Etudiant2.tar.gz`

Les rapports doivent répondre aux questions posées dans les sujets et être illustrés avec des images issues des traitements que vous avez réalisés. Faites très attention à :

1. décrire ce que vous avez fait ; expliquer pourquoi et comment vous l'avez fait ;
2. décrire les résultats obtenus ;
3. analyser les résultats obtenus ; faire le lien entre ces résultats et vos cours.

TP1 : géométrie épipolaire

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

Soit un système stéréoscopique donnant deux images I_g et I_d . 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 \mathbf{K}*

Nous supposons que la caméra C_g est positionnée à la position ${}^g\mathbf{T}_o$ définie par

$${}^g\mathbf{T}_o = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 2 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

En utilisant le squelette de code mis à votre disposition, position la caméra C_d à 10 cm à droite de C_g

Question 2 *Donner la matrice ${}^d\mathbf{T}_o$*

Question 3 *Comment appelle t'on un tel système*

Question 4 *Donner les deux images I_g et I_d ainsi obtenues*

On souhaite pouvoir faciliter la mise en correspondance de point de l'image. Pour cela, on s'attachera à représenter le lieu géométrique dans l'image I_g où le correspondant d'un point \mathbf{x}_d de I_d peut se situer.

Question 5 *Caractériser ce lieu. Calculer son équation. Donner les dans le rapport pour les points (100,100) et (50, 75).*

Question 6 *Tracer à l'aide des fonctions de la classe `vpDisplay` le point \mathbf{x}_d dans I_d et le lieu calculé précédemment dans I_g . Vous choisirez une dizaine de points correspondant à des point particulier de l'image et vous vérifierez que vous obtenez bien le résultat attendu. Donner les images obtenus.*

Question 7 *Faite le processus inverse avec un point \mathbf{x}_g de I_g*

Question 8 *Donner la position des épipoles*

Positionner maintenant la caméra C_g à 20 cm devant C_g .

Question 9 *Donner la nouvelle matrice ${}^d\mathbf{T}_o$*

Question 10 *Donner la nouvelle position des épipoles.*

Question 11 *Refaire pour cette nouvelle position les question 5 et 6*

Positionner maintenant la caméra C_g en ${}^d\mathbf{T}_o$ en utilisant le code suivant:

```
vpHomogeneousMatrix dMo(0.1,0,1.9,  
                        vpMath::rad(5),vpMath::rad(5),vpMath::rad(5)) ;
```

Question 12 *Donner la nouvelle matrice ${}^d\mathbf{T}_o$*

Question 13 *Donner la nouvelle position des épipoles*

Question 14 *Refaire pour cette nouvelle position les question 5 et 6*

TP2: Reconstruction 3D

Vous trouverez dans le repertoire data plusieurs images rectifiées `left.pgm` et `right.pgm` ou `tsukuba-l.pgm` et `tsukuba-r.pgm` ou `cedar-left.pgm` et `cedar-right.pgm`. L'objectif de ce TP est de calculer une carte de disparité de l'environnement pour les trois couples d'image ci-dessus.

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:

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

Question 3 *Même question en utilisant la "Sum of Squared Differences" ou SSD. Vous utiliserez différentes tailles des de fenêtrés (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*

TP3: Estimation d'une homographie

Récupérer la trame du TP3. Récupérer le fichier data.zip mis à jour.

L'objectif de ce TP est d'estimer une homographie. Pour cela vous disposez de deux images I1.pgm et I2.pgm. Sur chacune de ces images 5 points sont présents et bien visible. Vous cliquerez sur chacun des points dans l'image 1 puis ferez la même chose dans l'image 2. Vous récupérerez ainsi les coordonnées des points dans les deux images. Le code pour réaliser cette tâche est donnée dans TP3.

Question 1 Combien de points sont nécessaires pour calculer une homographie

Question 2 Connaissant un ensemble N de point mis en correspondance $(u1_i, v1_i)$ and $(u2_i, v2_i), i = 1 \dots N$, construire une fonction qui estime l'homographie 1H_2 .

Question 3 Comment juger de la qualité de votre estimation

Question 4 En prenant les points $(u2_i, v2_i)$, transférer les dans l'image 1 $(u_i^{1'}, v_i^{1'})$, afficher les, calculer l'erreur commise r par rapport aux coordonnées des points $(u1_i, v1_i)$ mesurées par "click". Afficher un cercle centré sur $(u_i^{1'}, v_i^{1'})$ de rayon 10 fois l'erreur commise.

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

TP4: Estimation d'une homographie (bis)

Il faut cependant reconnaître que de gros points blancs sur une image, ce n'est pas courant. Nous allons donc maintenant considérer des points extraits et mis en correspondance automatiquement avec un algorithme.

Récupérer TP4.cpp. Ce programme est très similaire au précédent (TP3), mais l'extraction des points d'intérêt et la mise en correspondance est automatique (et nécessite l'installation d'openCV). Si vous ne souhaitez pas installer openCV vous trouverez dans le fichier `correspondance.txt` les coordonnées des points mis en correspondance sous la forme suivante (`index u1 v1 u2 v2`).

Question 6 Faites afficher les mises en correspondance. Que constatez vous ?

Question 7 Appliquer l'algorithme du TP3. Les erreurs vous semblent elles raisonnables ?

Question 8 Implémenter un RANSAC. Une fois tester les erreurs sont elles maintenant raisonnables ?

Question 9 Faites l'acquisition d'une dizaine d'images et faites une mosaïque. Quelle contraintes doit ont cependant respecter.