

Consignes générales.

- Vous disposez d'une heure pour traiter le sujet qui vous a été fourni.
- Vous avez droit à vos notes de cours, le seul logiciel utile pour ce TP est Python.
- Vous créez sur votre ordinateur un dossier nommé de la façon suivante : numéro du sujet suivi du nom de famille (par exemple 8dupont), dans lequel seront sauvegardés tous les documents (programmes Python, fichiers textes) nécessaires à votre évaluation, qui sera effectuée uniquement sur cette base (pas de copie papier). Le surveillant de salle s'occupera de récupérer les fichiers sur une clé USB, laissez simplement les ordinateurs allumés à la fin de l'heure.
- Le surveillant de salle ne peut intervenir qu'en cas de problème technique sur votre ordinateur.
- Bon courage!

Sujet i : traitement d'image sur la plateforme Immersia (Université de Rennes).



FIGURE 1 – Vue de la salle de réalité virtuelle

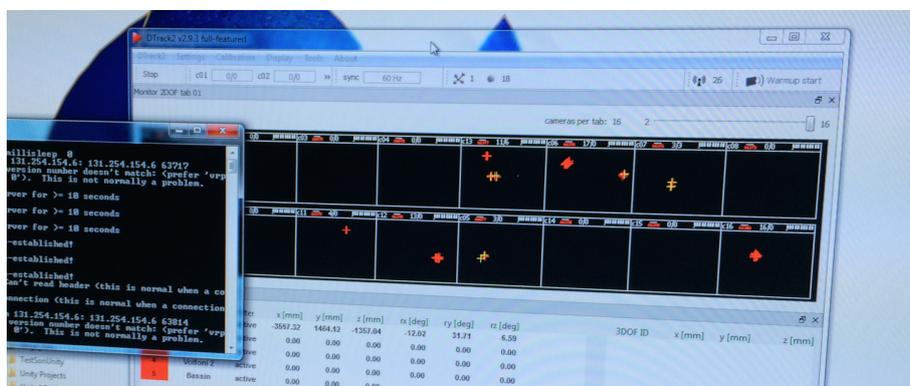


FIGURE 2 – Vue, après traitement, des images issues des caméras infra-rouge

La plateforme Immersia est une salle de réalité virtuelle qui permet de se déplacer dans un environnement virtuel et d'être en interaction avec celui-ci.

L'utilisateur dispose de lunettes de vision 3D équipées de sphères fluorescentes. Des caméras infrarouges permettent de trianguler la position et l'orientation des lunettes par analyse d'image, à chaque sphère est associée une coordonnée sur l'image (voir Figure 2). Une fois la position et l'orientation des lunettes déterminée, l'image conforme à la vision de l'utilisateur est projetée sur les trois murs et le sol.

L'objectif du TP est de traiter la décomposition de l'image en plusieurs petites images afin de détecter la position de chacune des sphères.

On rappelle les morceaux de codes suivants :

```
from PIL import Image : Import de la librairie de traitement d'image
im=Image.open("jambe.jpg") : Ouverture du fichier image
dimensions=im.size : Récupère les dimensions (largeur,hauteur) en pixels sous forme de liste à 2
éléments, c'est un tuple (sorte de liste non modifiable)
data=list(im.getdata()) : Stockage de l'image dans une liste nommée data
im.show() : Affiche l'image im
```

On donne l'image d'un capteur à sphère fluorescentes (jambe.jpg) identiques à celles des lunettes.

1 Créer une nouvelle image, portion de l'image d'origine, étant le coin en haut à droite de l'image divisée en 6 (2 lignes et 3 colonnes)

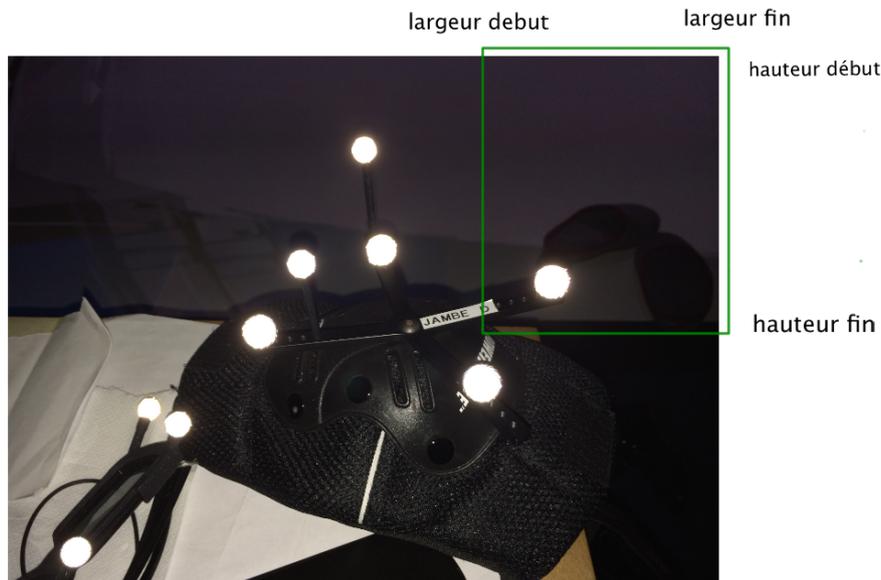


FIGURE 3 – Image à isoler et illustration des variables

pour cela on propose de :

1. créer une variable "largeur" donnant la largeur de l'image d'origine en pixel
2. créer une variable "hauteur" donnant la hauteur de l'image d'origine en pixel
3. créer une variable "débutlargeur" donnant le numéro du pixel (dans la largeur de l'image d'origine) du début de la nouvelle image
4. créer une variable "finlargeur" donnant le numéro du pixel (dans la largeur de l'image d'origine) de la fin de la nouvelle image
5. créer une variable "débuthauteur" donnant le numéro du pixel (dans la hauteur de l'image d'origine) du début de la nouvelle image
6. créer une variable "finhauteur" donnant le numéro du pixel (dans la hauteur de l'image d'origine) de la fin de la nouvelle image
7. créer une nouvelle liste "datanew" pour stocker le code RVB de la nouvelle image
8. mettre en place un algorithme de remplissage de data"new" utilisant deux boucles for et la commande datanew.append
9. créer un nouveau tuple "newsiz" avec les dimensions de la nouvelle image
10. afficher la nouvelle image

2 Créer une fonction permettant de faire ce que l'on veut

Créer la fonction : `decoupe(nlignes,ncolonne,nl,nc)`. Avec :

- `nlignes`, le nombre de lignes de découpe de l'image d'origine ;
- `ncolonne`, le nombre de colonnes de découpe de l'image d'origine ;
- `nl`, le numéro de la ligne de la portion d'image à afficher (nouvelle image) la première ligne étant numérotée 1 ;
- `nc`, le numéro de la colonne de la portion d'image à afficher (nouvelle image) la première colonne étant numérotée 1 ;