

Le but de ce TP est de mettre en place une petite application permettant de contrôler de curseur avec l'orientation de la tête.

1. Préparatifs techniques

Pour cela, nous allons utiliser la bibliothèque OpenCV (<https://opencv.org/>).

Les dépôts dans les machines de TP contiennent la version 2.4 d'OpenCV. Vous pouvez consulter la documentation sur (<https://docs.opencv.org/2.4/>).

Pour l'installer:

```
apt install libopencv-dev
```

Vérifiez la version avec la commande:

```
pkg-config --modversion opencv
```

Votre archive contient un fichier compile.sh qui vous permet de compiler le fichier source main.cpp avec les dépendances OpenCV.

Ce petit programme permet de trouver la position de la tête dans l'image grâce à un classificateur.

Chaque image du flux vidéo est représentée dans une matrice du type « Mat ». Si vous voulez parcourir les pixels de l'image, vous pouvez vous inspirer de : [https://docs.opencv.org/2.4/doc/user_guide/ug_mat.html?highlight=pixel access#accessing-pixel-intensity-values](https://docs.opencv.org/2.4/doc/user_guide/ug_mat.html?highlight=pixel%20access#accessing-pixel-intensity-values)

2. Calcul de l'orientation de la tête.

On se limite ici aux mouvements « haut-bas » et « penche à gauche/ penche à droite » (à différencier du gester « tourner la tête à gauche/ tourner la tête à droite »).

Essayez de trouver la position des yeux à l'aide du tutoriel :

https://docs.opencv.org/2.4/doc/tutorials/objdetect/cascade_classifier/cascade_classifier.html#cascade-classifier

Grâce à la position de la tête et des yeux, représentez l'orientation de la tête avec quelques variables.

3. utilisation du code produit en 2 pour contrôler un curseur souris.

Comment pouvez vous utiliser l'orientation de la tête pour contrôler le curseur ? Implémentez vos idées.

Vous pouvez utiliser Xlib pour contrôler le curseur. Un exemple de contrôle de curseur se trouve dans mouse.cpp.

A défaut d'utiliser Xlib, vous avez également la possibilité si vous le souhaitez, (même si ceci ne sera pas ici documenté et vous auriez à explorer seul la solution) d'utiliser VNC par exemple

(https://fr.wikipedia.org/wiki/Virtual_Network_Computing)

Quelques questions qui peuvent vous inspirer, et vous aider à proposer une solution :

- Est-ce qu'une approche similaire à celle d'une manette serait utile ?
- Quelles sont les limites de cette méthode ?
- Est-ce que vous pouvez imiter un pavé tactile en utilisant l'orientation de la tête?
- Quelles sont les limites de cette méthode ?

4. Bonus : clique gauche et droit

Reprendre OpenCV pour voir dans quelle mesure on peut utiliser le fait de cligner des yeux (cligner de l'oeil gauche, ou droit) pour permettre de définir l'équivalent d'un click souris.

Annexe. Pour travailler à distance :

Si vos dépôts ne contiennent pas la version 2.4 d'OpenCV, vous pouvez l'installer sur l'OS de votre choix en utilisant le répertoire github :

<https://github.com/opencv/opencv/tree/2.4>

Attention! N'utilisez pas la fonctionnalité "clone" du git, téléchargez le .zip (opencv-2.4.zip).

Pour l'installer:

https://docs.opencv.org/2.4/doc/tutorials/introduction/linux_install/linux_install.html#linux-installation

En bref:

Extrayez le fichier et entrez le répertoire.

```
mkdir ../opencvfiles
```

```
mkdir release
```

```
cd release
```

```
cmake -D CMAKE_BUILD_TYPE=RELEASE -D CMAKE_INSTALL_PREFIX=../opencvfiles ..
```

```
make (et attendre)
```

```
make install
```

Pour compiler le programme, utilisez compile.sh.

Ouvrez-le avec un éditeur de texte, et changez les chemins pour qu'ils correspondent à ceux de votre machine.