

TP 2.1 - Installation d'OpenCV

OpenCV est une bibliothèque open source de traitement d'image et de vision par ordinateur. Elle est écrite en C++ mais possède de nombreux wrappers qui permettent son utilisation avec d'autres langages de programmation. Ce n'est pas, loin s'en faut, la seule bibliothèque existante, mais c'est une des plus courantes et des plus adaptées à la vision par ordinateur en temps réel et pour la réalité augmentée.

Vous pouvez installer OpenCV sur votre système d'exploitation ou dans une machine virtuelle. Voici ci-après quelles indications selon votre système. Vous trouverez les tutoriels ici https://docs.opencv.org/4.5.5/df/d65/tutorial_table_of_content_introduction.html. Les TPs peuvent être réalisés soit en python soit en C++.

1 INSTALLATION DE OPENCV PYTHON

Suivez le tutoriel d'installation de <https://pypi.org/project/opencv-python/> en installant **Option 2 - Full package**. Il est conseillé de créer un environnement virtuel dédié.

2 INSTALLATION DE OPENCV C++

2.1 SOUS WINDOWS

Vous avez une machine virtuelle linux à télécharger sur claroline (suivre le lien, attention c'est long et cela fait plus de 6 Go). Vous devez installer VirtualBox <https://www.virtualbox.org> pour travailler sur avec cette machine virtuelle linux. Une fois VirtualBox installé, vous pouvez importer la machine virtuelle fournie. Ignorez l'erreur vous disant qu'il n'y a pas de dossier partagé. Le login est VM et le mot de passe root est CentraleNantes. OpenCV a déjà été installé en suivant la méthode décrite en 2.3.1.

2.2 SUR MacOSX

Pour plus de simplicité, sous OSX je vous propose d'utiliser HomeBrew pour faire l'installation. XCode doit être installé sur votre machine ainsi que les *Apple Command Line Tools*. *Remarque : si vous ne l'avez pas préalablement installé, c'est assez long et fait plusieurs Go.*

Installez Xcode via l'App Store et une fois Xcode installé utilisez la ligne de commande suivante dans un terminal (Application>Utilitaires>Terminal) pour installer les *Apple Command Line Tools*.

```
sudo xcode-select --install
```

Installez homebrew : <https://brew.sh> et mettez le à jour (brew update).

Pour installer opencv utilisez la commande suivante :

```
brew install opencv --with-contrib
```

L'installation est encore assez longue.

Pour faire les TP nous allons préférentiellement utiliser cmake <https://cmake.org> pour compiler et indiquer les dépendances à openCV.

```
brew install cmake
```

Vous pouvez aussi installer directement OpenCV à partir des sources en suivant le tutoriel : https://docs.opencv.org/4.5.5/d0/db2/tutorial_macos_install.html.

2.3 INSTALLATION DE OPENCV SOUS LINUX (OU DANS UNE MACHINE VIRTUELLE LINUX SOUS UN AUTRE SYSTÈME)

Si vous avez linux sur votre machine, vous pouvez directement utiliser ces commandes. Sinon, je vous conseille d'utiliser VirtualBox pour travailler sur une machine virtuelle linux sur votre ordinateur.

2.3.1 INSTALLATION D'UNE MACHINE VIRTUELLE LINUX (OPTIONNEL)

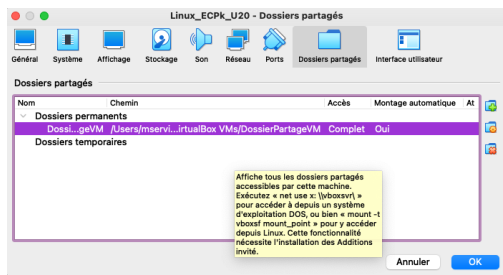
Il vous faut tout d'abord télécharger <https://www.virtualbox.org> et une image iso de Linux, par exemple Ubuntu 20.04.4 LTS <https://ubuntu.com/download/desktop>.

Vous pouvez créer une machine virtuelle Linux (avec à minima 10Go de disque qui lui est alloué et 3 à 4 Go de RAM). Au premier démarrage de la machine virtuelle, indiquez le chemin d'accès à l'image iso de linux et installez le système linux dans votre machine virtuelle.

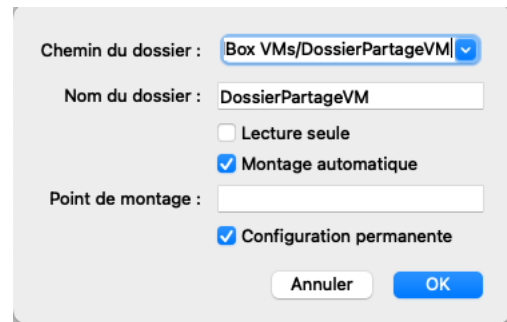
Une fois Linux installé sur la VM, vous pouvez dans le menu Périphérique de VirtualBoxVM "insérer l'image CD des additions invitées" et les installer sur votre VM.

```
sudo apt-get install virtualbox-guest-additions-iso virtualbox-ext-pack
```

Après redémarrage de la VM cela vous permettra d'avoir un dossier partagé entre votre VM et votre système d'exploitation ainsi que l'accès à votre webcam. Ce dossier se configure dans les paramètres de la VM. Vous trouverez un exemple de configuration dans la figure ??.



(a) Dossiers partagés dans les paramètres de la VM



(b) Exemple de configuration d'un dossier partagé

Pour avoir accès au dossier partagé, vous devrez vous ajouter au groupe vboxsf en tapant dans un terminal la commande suivante :

```
usermod -a -G vboxsf exempleusername
```

en changeant évidemment exempleusername par votre nom d'utilisateur.

Pour avoir accès à la webcam :

```
usermod -a -G vboxusers exempleusername
```

Si vous avez une webcam usb, n'oubliez pas de cocher la webcam dans le menu Périphériques>USB de VirtualBox. En cas de problème pour capturer un matériel USB, référez-vous à la documentation suivante : <https://forums.virtualbox.org/viewtopic.php?f=35&t=82639>
Pour éditer les codes, je vous propose d'utiliser Visual Studio Code <https://code.visualstudio.com/>

Installez aussi l'extension C++ <https://code.visualstudio.com/docs/languages/cpp>.

2.3.2 INSTALLATION SOUS LINUX

Toutes les commandes indiquées doivent être entrées dans un terminal. Nous allons commencer par installer les bibliothèques indispensables :

```
sudo apt update && sudo apt install -y g++ wget unzip cmake
```

GTK pour pouvoir afficher des fenêtres :

```
sudo apt-get install libgtk2.0-dev
```

```
sudo apt install libcanberra-gtk-module libcanberra-gtk3-module
```

Puis installer les bibliothèques nécessaires pour manipuler des vidéos :

```
libavcodec-dev libavformat-dev libavutil-dev libswscale-dev
```

Téléchargez les sources de opencv :

```
wget -O opencv.zip https://github.com/opencv/opencv/archive/4.x.zip
```

```
wget -O opencv_contrib.zip https://github.com/opencv/opencv_contrib/archive/4.x.zip
```

Dézippez les sources :

```
unzip opencv.zip
```

```
unzip opencv_contrib.zip
```

Créez un répertoire pour la compilation et allez dedans :

```
mkdir -p build && cd build
```

Lancez la configuration via Cmake :

```
cmake -D CMAKE_BUILD_TYPE=RELEASE \  
-D CMAKE_INSTALL_PREFIX=/usr/local \  
-D OPENCV_GENERATE_PKGCONFIG=ON \  
-D WITH_GTK=ON \  
-D WITH_FFMPEG=ON \  
-D WITH_GDAL=ON \  
-D WITH_GSTREAMER=ON \  
-D OPENCV_EXTRA_MODULES_PATH=../opencv_contrib-4.x/modules \  
-D BUILD_EXAMPLES=ON \  
../opencv-4.x
```

Faites la compilation (c'est long!) et l'installation :

```
cmake --build .  
sudo make install
```

Pour éditer les codes, je vous propose d'utiliser Visual Studio Code

https://code.visualstudio.com/docs/?dv=linux64_deb.

Une fois le .deb téléchargé, installez le :

```
sudo apt install ./<file>.deb
```

et pour le lancer :

```
code &
```

Puis installez l'extension C++ <https://code.visualstudio.com/docs/languages/cpp>.