



**POLYTECH**<sup>®</sup>  
CLERMONT-FERRAND

Polytech Clermont Ferrand

# Développement d'une application OpenCV avec interface Qt

Note d'application



JULLIEN François  
GE5 2013/2014



# I. Table des matières

---

I.	Table des matières .....	3
II.	Introduction.....	4
III.	bibliothèque graphique OpenCV.....	5
IV.	Qt.....	6
1.	Présentation .....	6
2.	La philosophie .....	6
3.	Qmake .....	6
V.	Tutorial .....	7
1.	Installation de la librairie OpenCV.....	7
2.	Installation de Qt.....	7
3.	L'interfacage.....	7

## II. Introduction

---

Durant les projets Génie Electrique au sein de Polytech Clermont-Ferrand, nous avons répondu à une problématique du conservatoire national du saumon sauvage qui désirait mesurer le taux de croissance de smolt (bébé saumon) de manière automatique. Ils ont donc proposé le sujet « Mesure par vision du taux de croissance d'un smolt », pour ce projet le client nous a fourni des images d'écaïlle de smolt, sur lesquelles nous avons eu à faire de la reconnaissance graphique sur des images d'écaïlles de smolt.

Pour avoir plus d'informations sur le traitement réalisé pour l'analyse des images d'écaïlle je vous invite à consulter l'application note de Mr Hakim EL Gharrabi : [Principe de détermination des phases de croissances à partir d'une image d'écaïlle de Smolt.](#)

### III. bibliothèque graphique OpenCV

---

Le projet a fait appel à du traitement d'image dans le but de déterminer les phases de croissances d'un saumon, pour déterminer les phases de croissances nous avons analysé la colorimétrie de l'écaille pour en déduire des phases de croissances plus ou moins rapides qui corresponde aux phases d'hiver ou d'été.

Néanmoins avant d'arriver à trouver le traitement à appliquer à l'image nous avons réalisé de nombreux essais d'algorithmes : Canny, edge detector, FFT... c'est pourquoi nous avons choisi d'utiliser OpenCV qui est une bibliothèque graphique offrant l'avantage de faciliter l'utilisation d'algorithmes traitement d'image qui est de plus open source.

Je vous invite à consulter l'application note de Mr Khaled HARRACHE pour la mise en œuvre et l'utilisation de OpenCV : [mise en œuvre et utilisation de la librairie OpenCV](#)

## IV. Qt

---

Lors de l'avancée du projet, la mise en place d'une interface graphique est parue comme une nécessité pour permettre au client d'utiliser de manière simple l'algorithme de traitement d'image développé. Nous avons choisi le Framework Qt

### 1. [Présentation](#)

Qt est un Framework spécialement conçu pour la réalisation d'interface graphique. Il est développé en C++ avec une orientation objet. Depuis son origine, Qt est cross-plateforme en supportant de nombreuses plateformes comme Unix, Windows, Mac OS, Android... Aujourd'hui, Qt propose de nombreuses classes permettant de gérer facilement les fichiers, la 3D...

Qt est aujourd'hui sous licence LGPL 2.1 permettant une utilisation gratuite et la diffusion du code développé.

Le fait que Qt supporte aussi le binding, utilisation de plusieurs langages de programmation, permet une meilleure intégration.

### 2. [La philosophie](#)

L'API Qt offre une convention de nommage très régulière en préfixant toutes ses classes d'un Q ainsi qu'une typographie CamelCase semblable au Java, cela permet de trouver facilement les fonctions correspondantes dans les différentes classes.

Il y a aussi une notion de signaux et slots, le principe est de connecter différents objets entre eux, cela rendant beaucoup plus aisé l'interfaçage des différentes classes et des signaux de l'API.

### 3. [Qmake](#)

Qt étant conçu pour être portable, il a été nécessaire de créer un programme permettant l'automatisation des réglages correspondant à la plateforme. C'est pourquoi a été développé Qmake qui prend comme argument un fichier .pro décrivant tous les paramètres du projet et génère l'application spécifique à la plateforme désirée. C'est grâce au fichier .pro que nous allons interfacier OpenCV avec Qt.

```
TARGET = monAppli
SOURCES = main.cpp mainwindow.cpp
HEADERS = mainwindow.h
FORMS = mainwindow.ui
QT += sql
```

Exemple de fichier .pro

# V. Tutorial

---

Pour interfacier OpenCV avec Qt nous allons décomposer l'installation en trois parties :

## 1. [Installation de la librairie OpenCV](#)

- Premièrement, il faut récupérer la dernière version de OpenCV disponible sous <http://opencv.org/downloads.html>
- Ensuite, décompresser l'archive et créer un nouveau dossier à l'emplacement de l'archive décompressée. On peut alors se placer dans ce nouveau dossier pour exécuter cmake qui va créer le make file associé à la machine permettant de recompiler OpenCV sur son ordinateur.

```
$ mkdir release
$ cd release
$ cmake -D CMAKE_BUILD_TYPE=RELEASE -D CMAKE_INSTALL_PREFIX=/usr/local ..
```

- Il ne reste plus qu'à compiler et installer

```
$ make
$ sudo make install
```

## 2. [Installation de Qt](#)

Pour l'installation de Qt il y a plusieurs possibilités suivant si l'on choisit l'environnement de développement associé ou non, personnellement j'ai choisi d'utiliser Qt creator comme interface de développement qui est facile d'installation et contient déjà tous les outils nécessaires associés (Documentation, mingw,..). En suivant ce lien <http://qt-project.org/downloads> vous trouverez toutes les possibilités pour procéder à l'installation, quel que soit votre choix.

## 3. [L'interfaçage](#)

C'est grâce au .pro et à Qmake qu'il est facile d'inclure OpenCV, on ajoute au .pro les commandes suivantes qui vont servir à spécifier l'emplacement des dll de OpenCV.

```
OPENCV_DIR = C:/opencv/build
INCLUDEPATH += $$ {OPENCV_DIR}/include
LIBS += "../bin/libopencv_*.dll"
```

Il suffit alors de faire des includes dans le prototype des classes qui utilisent les librairies OpenCV comme ci-dessous :

```
#include <iostream>
#include <opencv2/opencv.hpp>
#include <opencv2/highgui/highgui.hpp>
```