



Formation OpenCV

Philippe FOUBERT

Premiers pas avec OpenCV



Par où commencer ?

→ La documentation !

- Directement sur Internet <http://docs.opencv.org/master/>
- Documentation générée lors de la compilation de la librairie

→ En particulier :

- **Reference Manual**
- **Cheat Sheet**
- **Tutorials** que vous avez (ou pas) choisi de compiler en même temps que la librairie

La démarche

- 1. Nous allons commencer par quelques exemples simples**
- 2. Nous apprendrons à connaître les fondamentaux de la librairie OpenCV**
- 3. Nous aborderons des exemples plus élaborés**

Charger et afficher une image

```
1 #include <opencv2/core.hpp>
2 #include <opencv2/highgui.hpp>
3 #include <iostream>
4 using namespace cv;
5
6 int main(int /*argc*/, char** /*argv*/)
7 {
8     int res(0);
9     const char filename[] = "smarties.jpg";
10
11     Mat img = imread(filename, IMREAD_COLOR);
12     if(img.empty())
13     {
14         std::cout << "Cannot load image!" << std::endl;
15         res = -1;
16     }
17     else
18     {
19         namedWindow("image", WINDOW_AUTOSIZE);
20         imshow("image", img);
21         waitKey(0);
22     }
23     return res;
24 }
```

Headers OpenCV : nous utilisons les modules core (classe Mat) et highgui (fonction imread)

En utilisant le **namespace** « cv », nous pouvons, par exemple, écrire imread au lieu de cv::imread

imread : charge l'image « smarties.jpg » en couleur (IMREAD_COLOR) en mémoire (dans l'objet img, instance de la classe Mat)

On crée une fenêtre vide nommée « image » dont la taille s'ajustera au contenu (WINDOW_AUTOSIZE)

On remplit cette fenêtre avec l'image qui a été chargée en mémoire

On attend qu'une touche soit pressée par l'utilisateur pour quitter

A vous de jouer !

1. **Ecrire une application qui charge l'image en niveaux de gris et l'affiche**
2. **Ecrire une application qui :**
 - a) charge l'image en couleur
 - b) convertit cette image en niveaux de gris
 - c) affiche chacune de ces 2 images
 - d) enregistre sur le disque dur l'image en niveau de gris au format PNG (*Portable Network Graphics*)