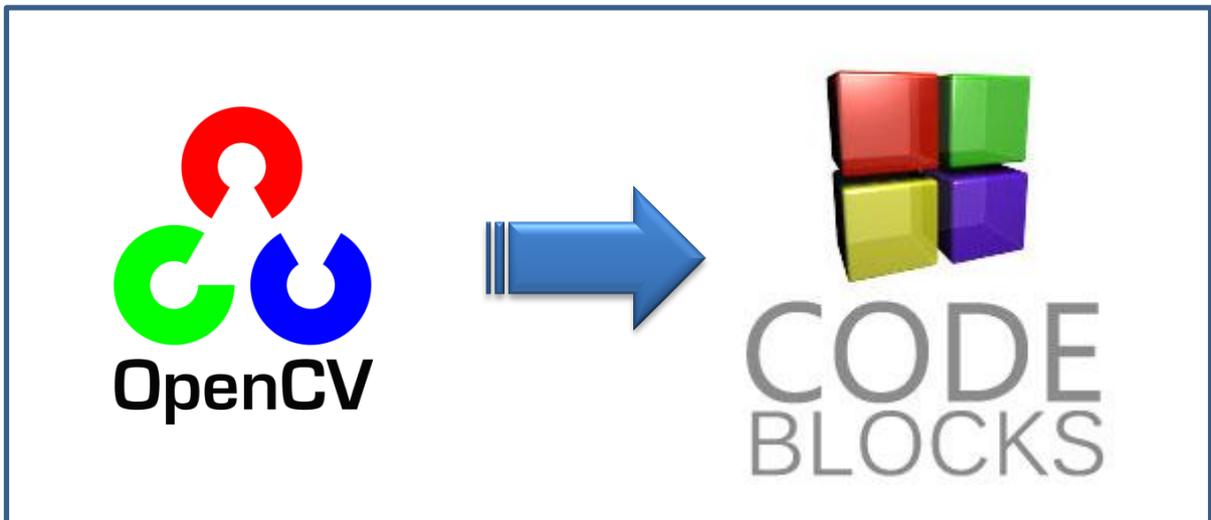


MISE EN ŒUVRE D'OPENCV SOUS CODE BLOCKS



Rachid Tidliouine

GE5

2013/2014

Liste des figures

Figure 1: la version récente de CodeBlocks.....	4
Figure 2: OpenCV selon le système d'exploitation	5
Figure 3: Menu CodeBlocks, définition des options de compilation et débogage.....	5
Figure 4: définir les répertoires à prendre en compte par le compilateur	6
Figure 5: répertoires pour compilation	6
Figure 6: répertoires à prendre en compte par le linker de CodeBlocks	6
Figure 7: Linker setting.....	7
Figure 8: Link Bibliothèques	7
Figure 9: propriétés du système	8
Figure 10: propriétés système.....	8
Figure 11: définir le path	9
Figure 12: erreur, il manque une bibliothèque	9
Figure 13: un premier programme simple.....	10
Figure 14: résultat après exécution	10
Figure 15: application des fonctions OpenCV sur une image	11
Figure 16: à gauche l'image d'entrée et à droite le résultat	11
Figure 17: application des fonction OpenCV sur une image (2)	12
Figure 18: à gauche l'image d'entrée et à droite le résultat	12
Figure 19: code pour afficher le flux de la caméra du PC	13

Sommaire

Introduction.....	4
1 Qu'est-ce que OpenCV ?	4
2 Installation de CodeBlocks	4
3 Intégration d' OpenCV dans Code Blocks	5
4 Exemples pour la prise en main	10
4.1 Premier programme*	10
4.2 Manipulation des images	11
4.2.1 Exemple 1*	11
4.2.2 Exemple 2*	12
4.3 Manipulation des vidéos	13
Conclusion	13

Introduction

Dans le cadre de notre formation d'ingénieur Génie Electrique à Polytech Clermont Ferrand, j'ai travaillé sur le projet « Estimation de l'activité de saumons par vision ». Notre principal objectif était de calculer la vitesse angulaire moyenne des poissons à partir des vidéos de 4 bassins circulaires filmés de haut. Pour ce faire, nous avons utilisé les bibliothèques OpenCV que nous avons intégré dans l'outil Code Blocks.

Cette note d'application explique comment mettre en œuvre OpenCV sous code Blocks, suivi par des exemples pédagogiques pour la prise en main de l'outil.

1 Qu'est-ce que OpenCV ?

OpenCV (**Open** Source **C**omputer **V**ision) est une bibliothèque proposant un ensemble de plus de 2500 algorithmes de vision par ordinateur, accessibles au travers d'API pour les langages C, C++, et Python. Elle est distribuée sous une licence BSD (libre) pour les plate-formes Windows, GNU/Linux, Android et MacOS.

Initialement écrite en C il y a 10 ans par des chercheurs de la société Intel, OpenCV est aujourd'hui développée, maintenue, documentée et utilisée par une communauté de plus de 40 000 membres actifs. C'est la bibliothèque de référence pour la vision par ordinateur, aussi bien dans le monde de la recherche que celui de l'industrie.¹

2 Installation de CodeBlocks

La version utilisé dans cette note d'application est la version 10.05 elle est disponible sur ce lien:

<http://sourceforge.net/projects/codeblocks/files/Binaries/10.05/Windows/codeblocks-10.05mingw-setup.exe/download>

Il existe également une version plus récente sur le site² de CodeBlocks.

File	Date	Size	Download from
codeblocks-12.11-setup.exe			BerliOS or Sourceforge.net
codeblocks-12.11-setup_user.exe	28 Nov 2012	28.2 MB	BerliOS or Sourceforge.net
codeblocks-12.11mingw-setup.exe			BerliOS or Sourceforge.net
codeblocks-12.11mingw-setup_user.exe	28 Nov 2012	96.8 MB	BerliOS or Sourceforge.net

NOTE: The codeblocks-12.11mingw-setup.exe file *includes* the GCC compiler and GDB debugger from TDM-GCC (version 4.7.1, 32 bit).

NOTE: The codeblocks-12.11(mingw)-setup_user.exe will NOT request ADMIN rights and can be installed into write accessible folders only. Trying to install to a folder like "Program Files" will result in an access error therefore. Use this special installer if you do not have admin access on your Windows machine. IF UNSURE, USE "codeblocks-12.11mingw-setup.exe"!

Figure 1: la version récente de CodeBlocks

¹ <http://fr.openclassrooms.com/informatique/cours>

² <http://www.codeblocks.org/downloads/binaries>

Si CodeBlocks est déjà installé sur votre ordinateur sans mingw, vous pouvez télécharger ce dernier sur ce lien : <http://sourceforge.net/projects/mingw/files/> et le définir comme compilateur dans les options de Codeblocks.

3 Intégration d' OpenCV dans Code Blocks

➤ Téléchargement des bibliothèques OpenCV

L'archive qui contient les bibliothèques OpenCV est disponible sur le lien suivant:

<http://opencv.org/downloads.html>

il existe une version pour chaque système d'exploitation. Ainsi, il faut en choisir celle qui est compatible avec Windows (figure2).



Figure 2: OpenCV selon le système d'exploitation

➤ Aller dans le menu de Code Blocks : *Setting -> Compiler and debugger*

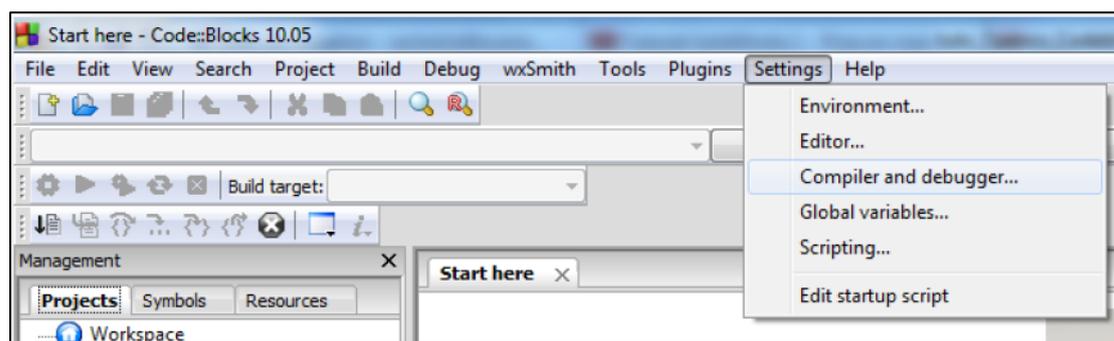


Figure 3: Menu CodeBlocks, définition des options de compilation et débogage

➤ Définir le répertoire des bibliothèques OpenCv dans CodeBlocks

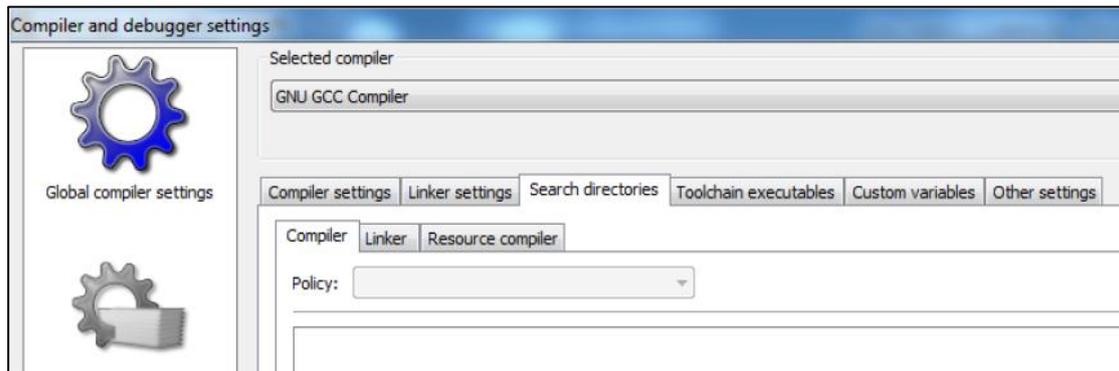


Figure 4: définir les répertoires à prendre en compte par le compilateur

➤ Dans l'onglet Compiler, définir le répertoire dans lequel OpenCV est sauvegardé (ajouter les 3 dossiers comme dans la figure 5)

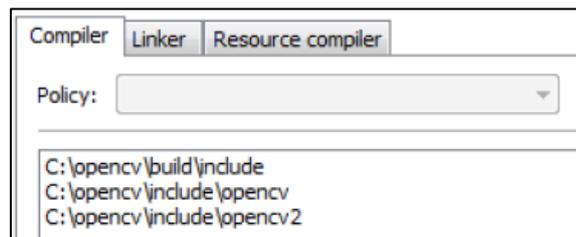


Figure 5: répertoires pour compilation

➤ Définir les répertoires du Linker comme dans la figure 6

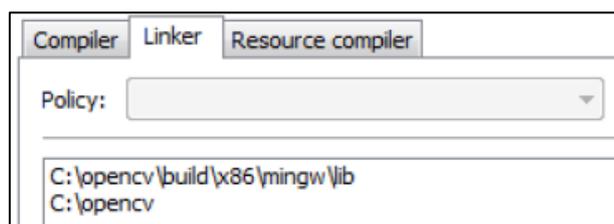


Figure 6: répertoires à prendre en compte par le linker de CodeBlocks

- Aller dans l'onglet Linker setting

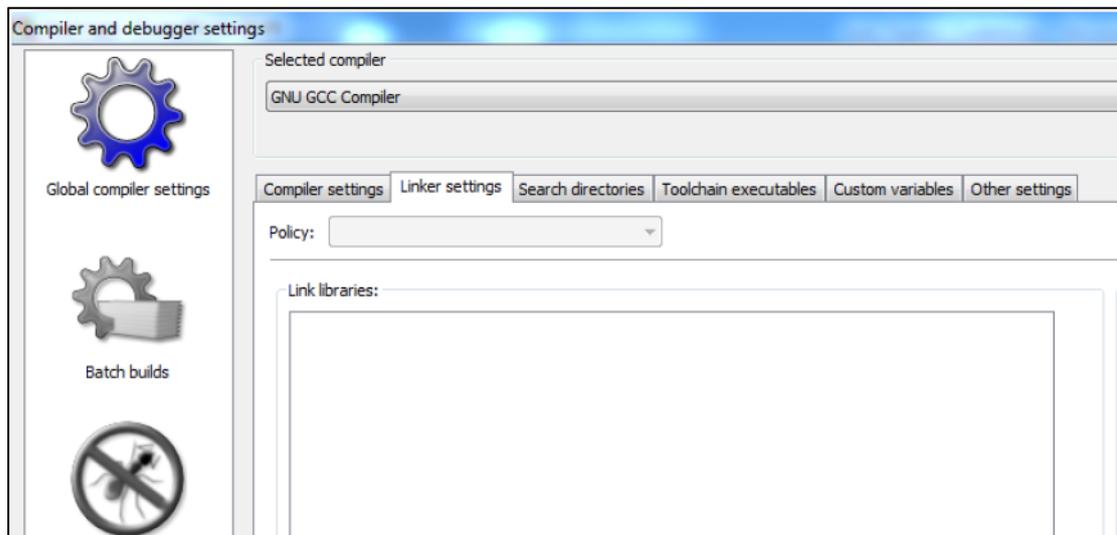


Figure 7: Linker setting

- Ajouter les librairies de Linker qui se trouvent dans le répertoire :
(votre_disque):\...\opencv\build\x86\mingw\lib

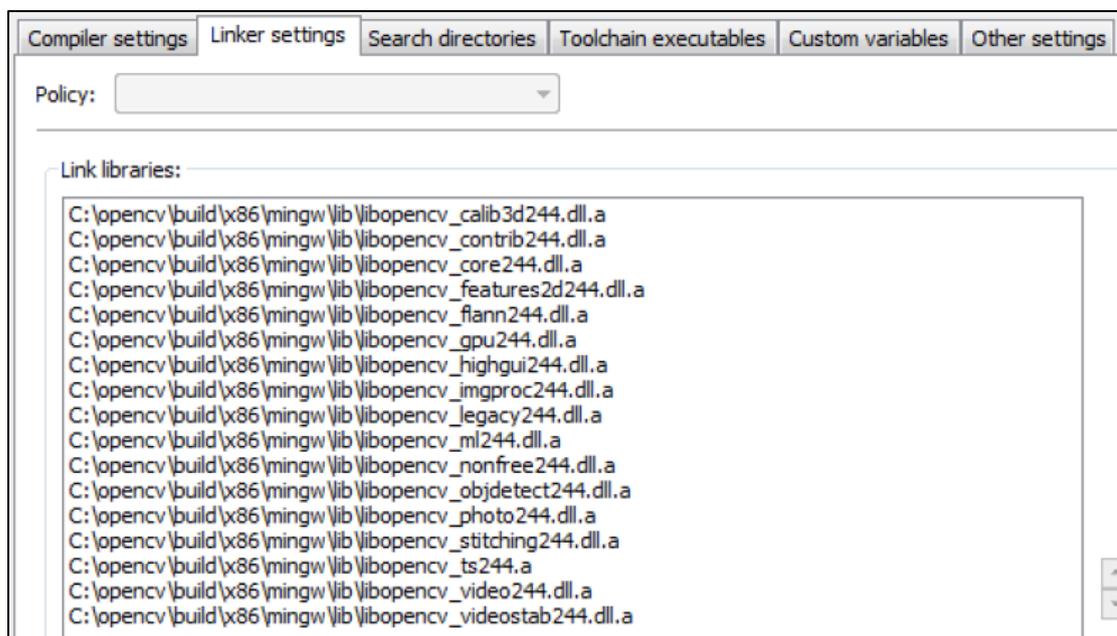


Figure 8: Link Libraries

- Afin de définir les variables d'environnement, aller sur :
Menu démarrer -> clic droit sur ordinateur-> propriétés

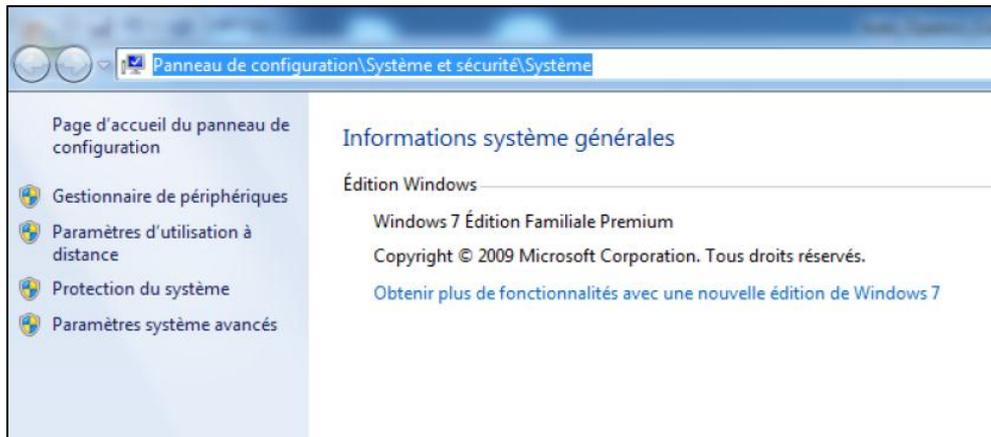


Figure 9: propriétés du système

- Choisir Paramètres système avancés (voir figure 9)
- Choisir variables d'environnement ... (figure 10)

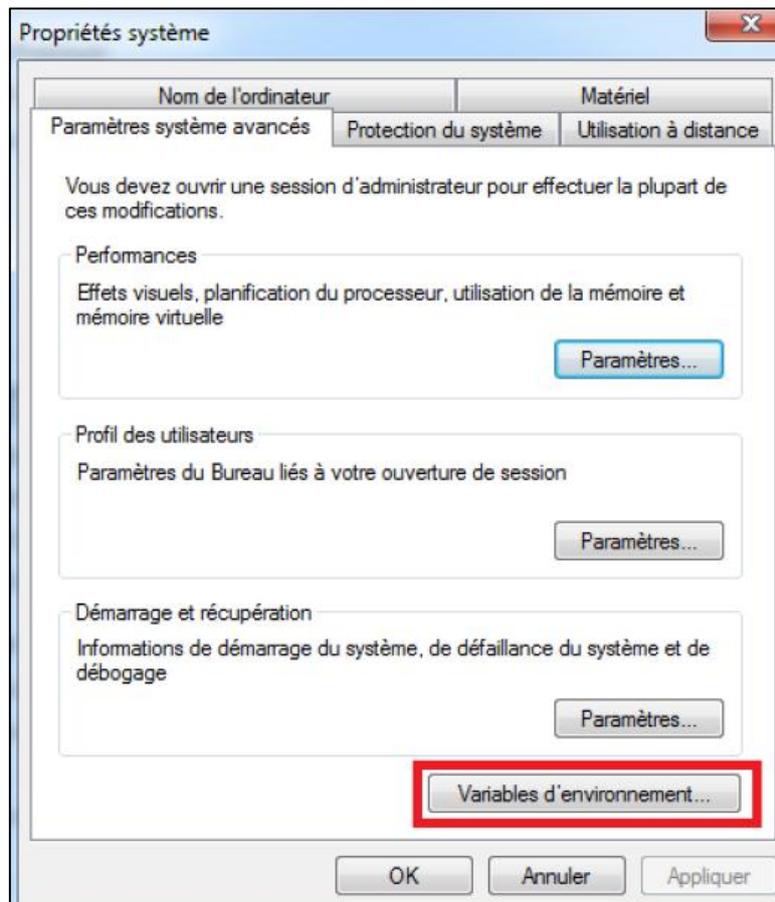


Figure 10: propriétés système

- **Ajouter dans la variable Path la valeur suivante :**
;C:\opencv\build\x86\mingw\bin

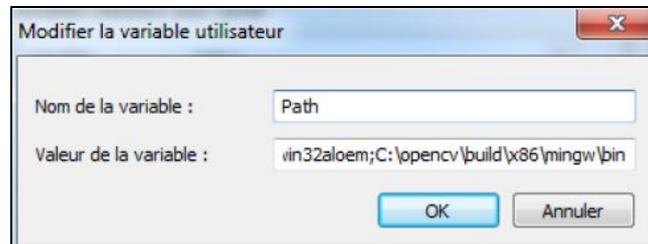


Figure 11: définir le path

- **Tester avec un des codes fournis dans le chapitre 4**
- **Si une erreur se produit lors de l'exécution, comme dans la figure 12, il faudra la télécharger sur internet et de l'ajouter dans ce chemin :**
opencv\build\x86\mingw\bin

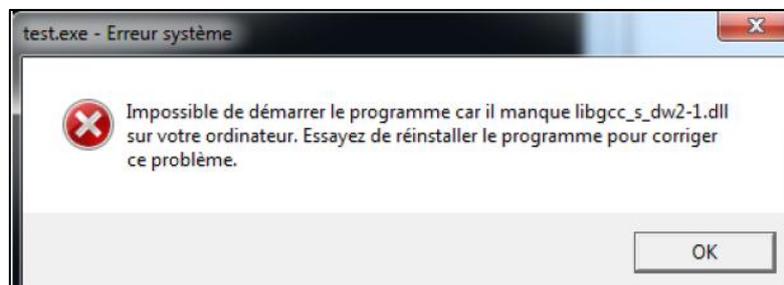


Figure 12: erreur, il manque une librairie

4 Exemples pour la prise en main

4.1 Premier programme*

Voici un premier programme qui prend une image et l'affiche :

```
1  #include <stdio.h>
2  #include <opencv2/opencv.hpp>
3  using namespace cv;
4  int main( int argc, char** argv )
5
6  {
7      Mat image;
8      image = imread(argv[1], 1); // image couleur
9      if( argc != 2 || !image.data )
10     {
11         printf( "No image data \n" );
12         return -1;
13     }
14     //-- affichage d'une fenetre
15     namedWindow( "Display Image", CV_WINDOW_AUTOSIZE );
16     //-- affichage de l'image
17     imshow( "Display Image", image );
18     //-- attente d'une touche clavier pour sortir
19     waitKey(0);
20     return 0;
21 }
22
```

Figure 13: un premier programme simple

➤ Résultat

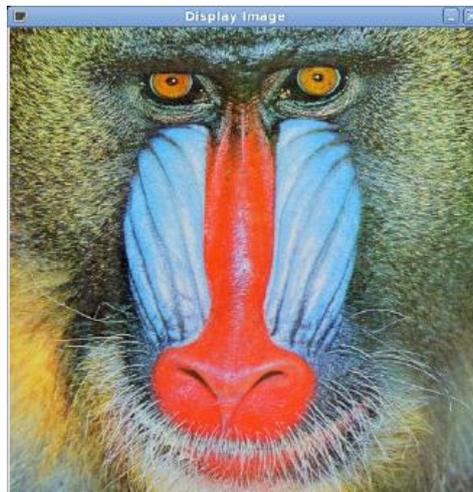


Figure 14: résultat après exécution

4.2 Manipulation des images

4.2.1 Exemple 1*

```
1  #include <stdio.h>
2  #include <opencv2/opencv.hpp>
3  using namespace cv;
4  using namespace std;
5
6  int main( int argc, char** argv )
7  {
8      Mat image;
9      //image = imread(argv[1], 1); // image couleur
10     image = imread(argv[1], 0); // image niveau de gris
11     if( argc != 2 || !image.data )
12     {
13         printf( "No image data \n" );
14         return -1;
15     }
16     cout << "taille image : " << image.rows << " x " << image.cols << endl;
17     for ( int l=0;l<image.rows;l++)
18     {
19         for ( int c=0;c<image.cols;c++)
20         {
21             image.at<char>(l,c) = 255- image.at<char>(l,c);
22         }
23     }
24     // on aurait pu (dû...) plutôt faire la commande suivante :
25     //image = 255 - image;
26     image(Rect(100,100,400,200)) *= 2; // modification de la zone : on multiplie tous les ndg par 2
27     namedWindow( "Display Image", CV_WINDOW_AUTOSIZE );
28     imshow( "Display Image", image );
29     waitKey(0);
30     return 0;
31 }
32
```

Figure 15: application des fonctions OpenCV sur une image

➤ Résultat :

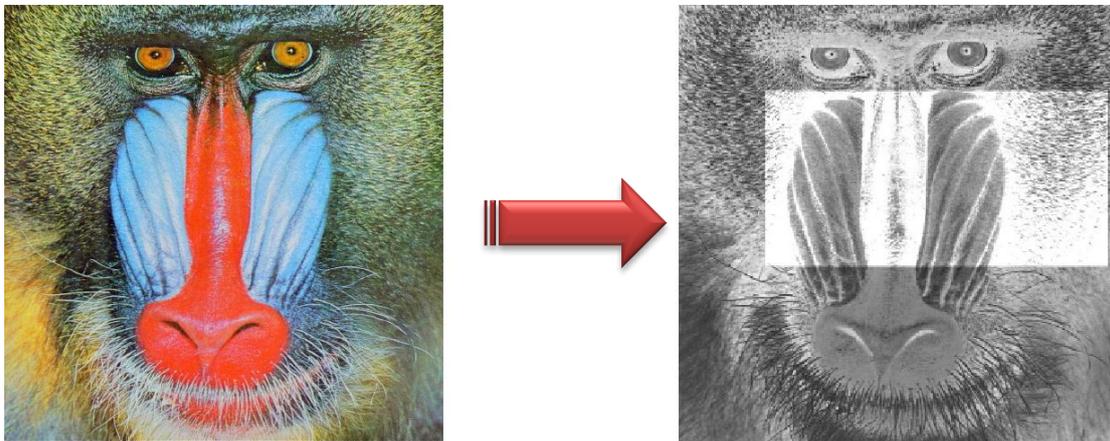


Figure 16: à gauche l'image d'entrée et à droite le résultat

4.2.2 Exemple 2*

```
1  #include <stdio.h>
2  #include <opencv2/opencv.hpp>
3
4  using namespace cv;
5  using namespace std;
6
7  int main( int argc, char** argv )
8  {
9      Mat image;
10     image = imread(argv[1], 0); // image niveau de gris
11     if( argc != 2 || !image.data )
12     {
13         printf( "No image data \n" );
14         return -1;
15     }
16     circle (image, Point(100,100),50, 255, -1); // cercle plein BLANC (couleur 255)
17     namedWindow( "Display Image", CV_WINDOW_AUTOSIZE );
18     imshow( "Display Image", image );
19     waitKey(0);
20     return 0;
21 }
22
```

Figure 17: application des fonction OpenCV sur une image (2)

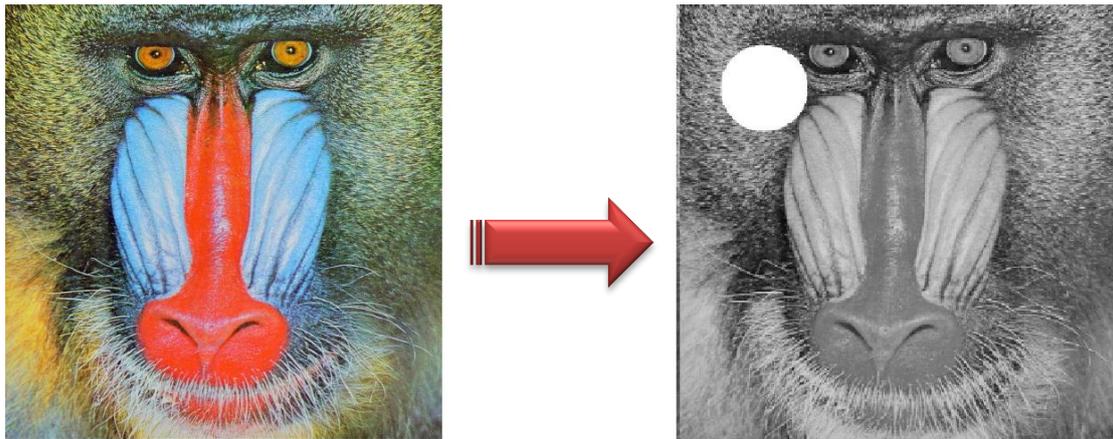


Figure 18: à gauche l'image d'entrée et à droite le résultat

* : exemples fournis par M. Chapuis

4.3 Manipulation des vidéos

Voici un programme qui lit et affiche le flux de la caméra du PC.

```
1  #include <stdio.h>
2  #include <opencv2/opencv.hpp>
3  using namespace cv;
4
5  int main()
6  {
7      namedWindow("ref",1);
8      Mat frame;
9      // cap(0) => caméra du PC
10     // cap("nomVideo.extension") => lire une video sur le disque
11     VideoCapture cap(0);
12     while(1)
13     {
14         // lire une image du flux video
15         cap>>frame;
16         // affichage
17         imshow("ref",frame);
18         waitKey(60);
19     }
20     return 0;
21 }
22
```

Figure 19: code pour afficher le flux de la caméra du PC

Conclusion

Pour faire du traitement d'image, il existe plusieurs outils tels que Labview et OpenCV. Cependant, ce dernier est le plus intéressant (surtout pour les débutant), car il est libre d'utilisation et il permet aux développeurs de voir le code source des algorithmes ce qui n'est pas le cas pour Labview.