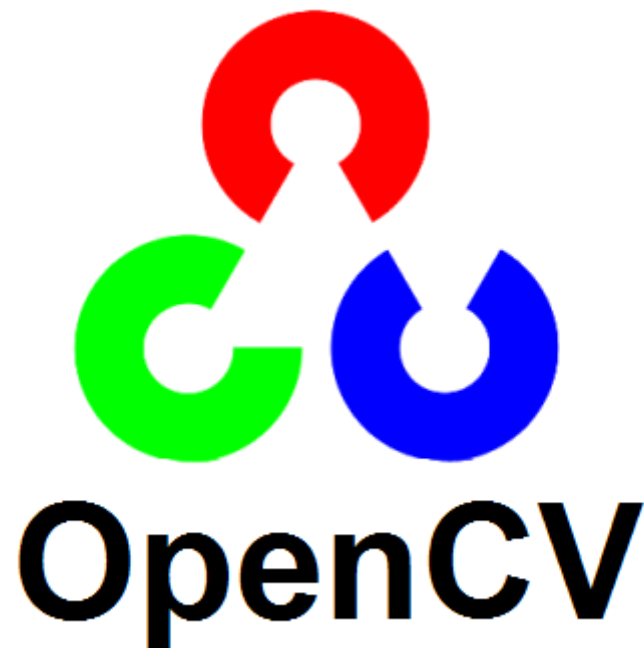


Polytech Clermont-Ferrand

# Déploiement d'application OpenCv : de la compilation à la diffusion

Note d'application



Amin SOUFARI  
GE5 2014-2015

## Table des matières

Introduction .....	2
1 Présentation d'OpenCV .....	2
2 Matériels utilisés .....	2
3 Installation de l'environnement de développement complet .....	3
3.1. Installation de l'IDE Code::blocks .....	3
3.2. Installation de OpenCV .....	3
3.3. Installation de Cmake .....	4
4 Compilation des bibliothèques OpenCv .....	5
5 Création d'une application OpenCV .....	7
6. Portage d'un programme OpenCV venant d'un autre PC.....	9
7. Du programme à l'exécutable.....	11
Conclusion.....	13

# Déploiement d'application OpenCv de la compilation à la diffusion

## Introduction

Afin de répondre à un projet de fin d'études basé sur le traitement d'image, le langage C++ et la librairie OpenCV s'est vu indispensable pour développer le programme. Bien qu'utilisé dans les laboratoires de recherche et dans l'industrie, le développement d'un même programme peut se faire par différents moyens et ressources. Assurer le portage d'un programme pour une autre équipe ou encore pour un autre membre d'une équipe se révèle nécessaire.

Cette note d'application explique comment développer, via Code::blocs, une application OpenCV et comment partager ou réutiliser le programme d'une tierce personne travaillant sur une librairie OpenCV différente.

## 1 Présentation d'OpenCV

OpenCV (**O**pen **S**ource **C**omputer **V**ision) est une librairie open source de traitement et analyse d'images et vidéos avec des interfaces pour les principaux langages de programmation C, C++, Java, C#, Python. Elle a l'avantage d'être optimisée pour les applications temps réelles, de fournir une API bas et haut niveau ainsi qu'une interface pour le langage de programmation parallèle IPP.

Parmi les fonctions les plus récurrentes dans le monde de l'entreprise ou dans les laboratoires on peut retrouver :

- La manipulation d'images (chargement, sauvegarde, copie, conversion...)
- Le traitement d'images (filtrage, détections de discontinuités, morphologie mathématique...)
- L'analyse d'images (composantes connexes, ajustement de primitives...)
- La manipulation et acquisition de vidéos
- La vision (calibration de caméra, stéréovision, recherche d'association...)
- Les manipulations de matrices et algèbre linéaire

## 2 Matériels utilisés

La bibliothèque OpenCv de version supérieure à 2.4.8 (2.4.10)

L'IDE Code::blocs 13.2 avec MinGW

La dernière version de Cmake (3.2)

Dependency Walker v2.2

### 3 Installation de l'environnement de développement complet

#### 3.1. Installation de l'IDE<sup>1</sup> Code::blocks

Le logiciel peut être téléchargé via le site officiel : [Code::blocks](http://codeblocks.org)

 Windows 2000 / XP / Vista / 7:

File	Date	Download from
codeblocks-13.12-setup.exe	27 Dec 2013	Berlios or Sourceforge.net
codeblocks-13.12mingw-setup.exe	27 Dec 2013	Berlios or Sourceforge.net
codeblocks-13.12mingw-setup-TDM-GCC-481.exe	27 Dec 2013	Berlios or Sourceforge.net

FIGURE 1 IDE CODE::BLOCKS

Il est conseillé de télécharger la dernière version de Code Blocks avec l'intégration de MinGW<sup>2</sup> qui prend désormais en charge les applications OpenCv nativement. Ici la version choisie sera la 13.12.

L'installation du programme ne nécessite aucun paramétrage particulier.

#### 3.2. Installation de OpenCV

La dernière version stable est disponible sur le lien officiel : [OpenCV](http://opencv.org)



FIGURE 2 OPENCV POUR WINDOWS

L'installation consiste à décompresser les fichiers OpenCV dans un endroit du disque dur. Il est ici conseillé de décompresser ces fichiers dans un dossier qui sera fixe. Ici, il sera choisi la destination « C:/ »

À l'issue de cette décompression, un dossier « opencv » de près de 3GB contenant 2 dossiers (build et sources) se créé.

<sup>1</sup> IDE =EDI : environnement de Développement Intégré

<sup>2</sup> MinGW : suite logicielle pour la compilation C/C++ contenant le compilateur gcc

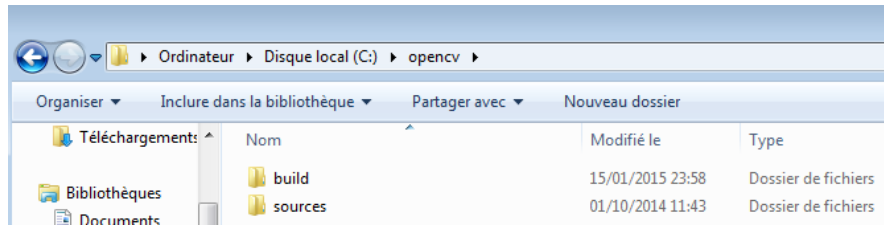


FIGURE 3 DOSSIER OPENCV

### 3.3. Installation de Cmake

Afin d'utiliser les différentes bibliothèques open source à un projet, il est nécessaire de les configurer les par rapport à son IDE. CMake est justement un système de compilation et de construction de projets multiplateforme et Open Source et disponible sur son site officiel : [Cmake](http://www.cmake.org)

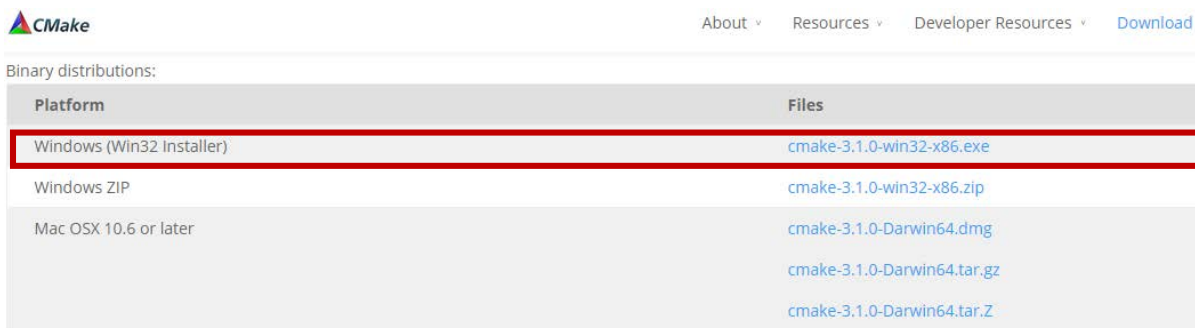


FIGURE 4 CMAKE POUR WINDOWS

Dans la procédure d'installation, il est recommandé de sélectionner l'ajout de Cmake au chemin système de tous les utilisateurs de la manière suivante :

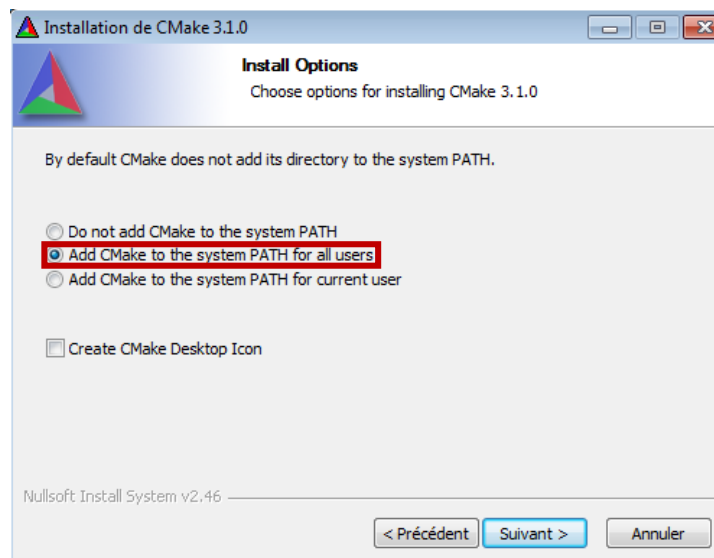


FIGURE 5 PARAMETRE INSTALLATION CMAKE

La suite de l'installation ne nécessite aucun paramétrage particulier.

### 4 Compilation des bibliothèques OpenCv

Cette étape est longue, mais ne doit être réalisée qu'une seule fois sur un PC.

Exécuter Cmake en administrateur, puis indiquer d'une part le chemin du dossier source d'OpenCv :

Dans ce cas : « C:/opencv/sources »

Et d'autre part le chemin où se construiront les différents éléments:

« C:/opencv/build/mingw » (le dossier mingw se créera automatiquement)

Les configurations seront alors définies de la manière suivante :

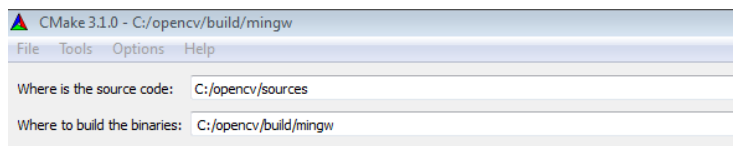


FIGURE 6 CONFIGURATION 1 CMAKE

Une fois cela réalisé, il suffit de cliquer sur « **Configure** » et de valider la fenêtre pop-up de la manière suivante :

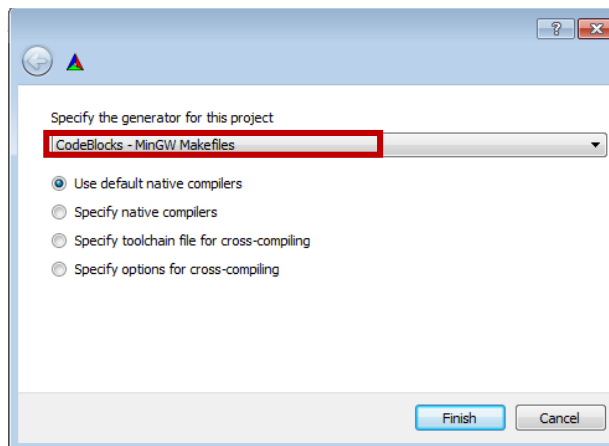


FIGURE 7 CONFIGURATION 2 CMAKE

Une fois la configuration terminée, il suffit de cliquer sur « **Generate** » pour valider le lancement des éléments de configuration par défaut<sup>3</sup>.

<sup>3</sup> Cette configuration permet au développeur de programme de profiter d'un temps d'exécution du programme réduit (par défaut, les bibliothèques sont compilées en version dynamique).

## Déploiement d'application OpenCV : de la compilation à la diffusion

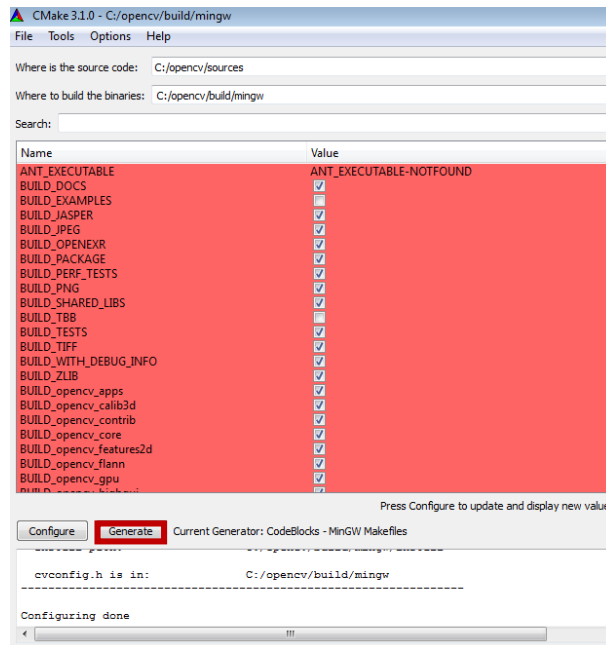


FIGURE 8 CONFIGURATION 3 CMAKE

Pour finir, il suffit de lancer le fichier code::blocks nommé OpenCV situé dans le dossier mingw précédemment créé (C:/opencv/build/mingw)

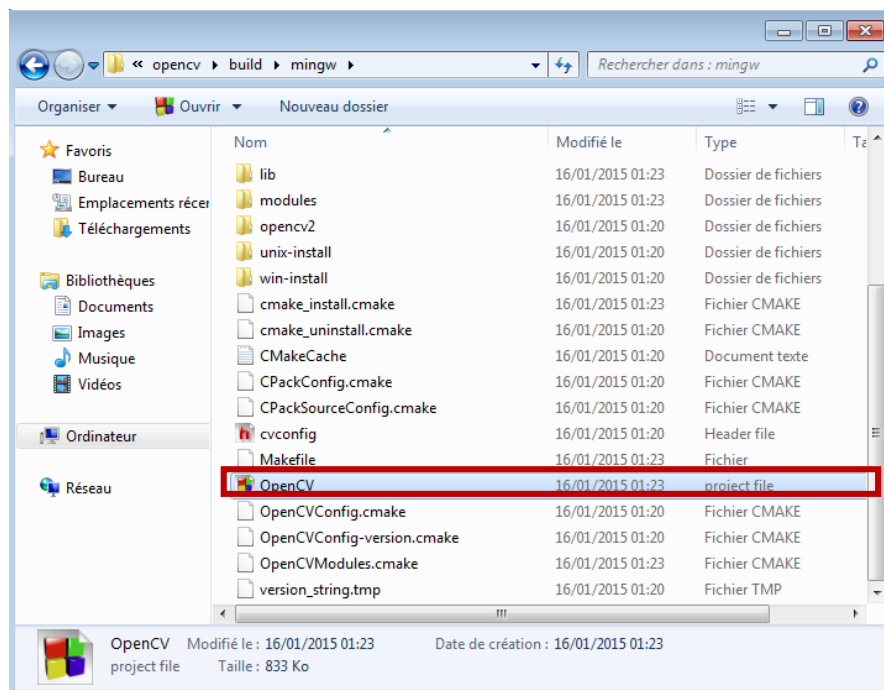


FIGURE 9 COMPILATION DES LIBRAIRIES

Puis compiler le fichier en appuyant sur le clavier les touches « ctrl » et « F9 » ou en cliquant sur le bouton « build ». Nb : Durée de 15- 30 minutes<sup>4</sup>.

<sup>4</sup> Durée qui dépend de la puissance de votre ordinateur (processeur + RAM) et des options de compilation choisies.

## 5 Création d'une application OpenCV

Afin de valider la bonne intégration d'OpenCV sur Code ::blocks il suffit de créer un fichier test.

Pour cela il suffit de sélectionner **File > New > Project > OpenCV project** et de suivre les chemins de configurations suivant :

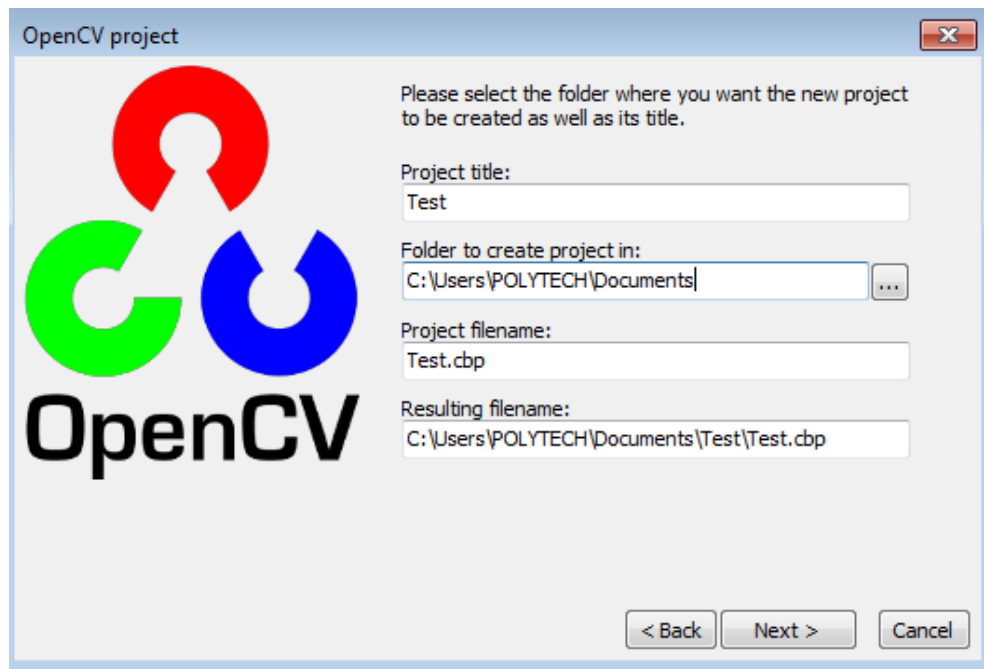


FIGURE 10 CREATION DU PROJET OPENCV 1

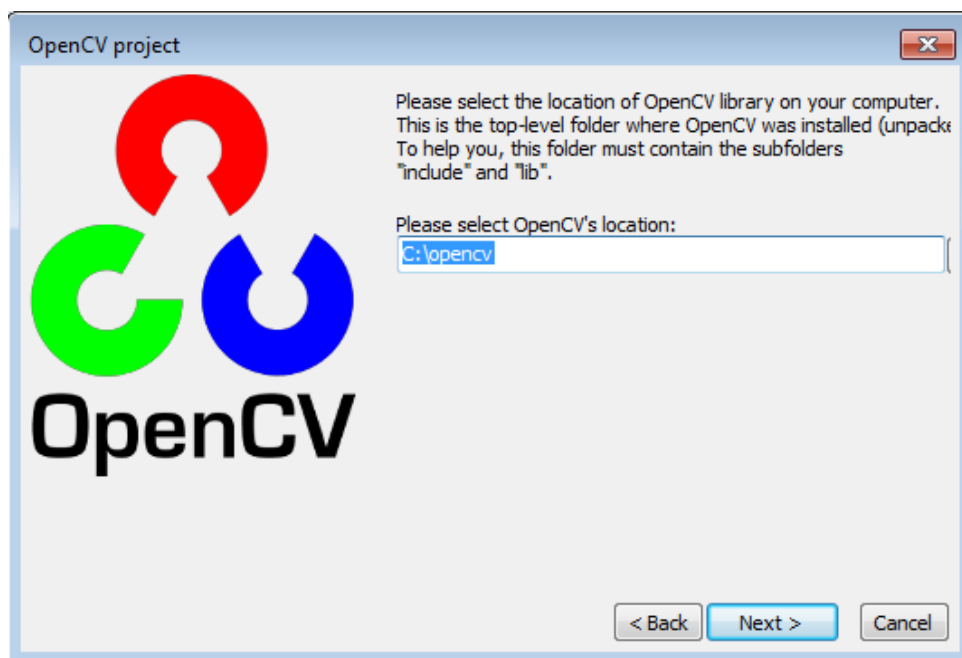


FIGURE 11 CREATION DU PROJET OPENCV 2



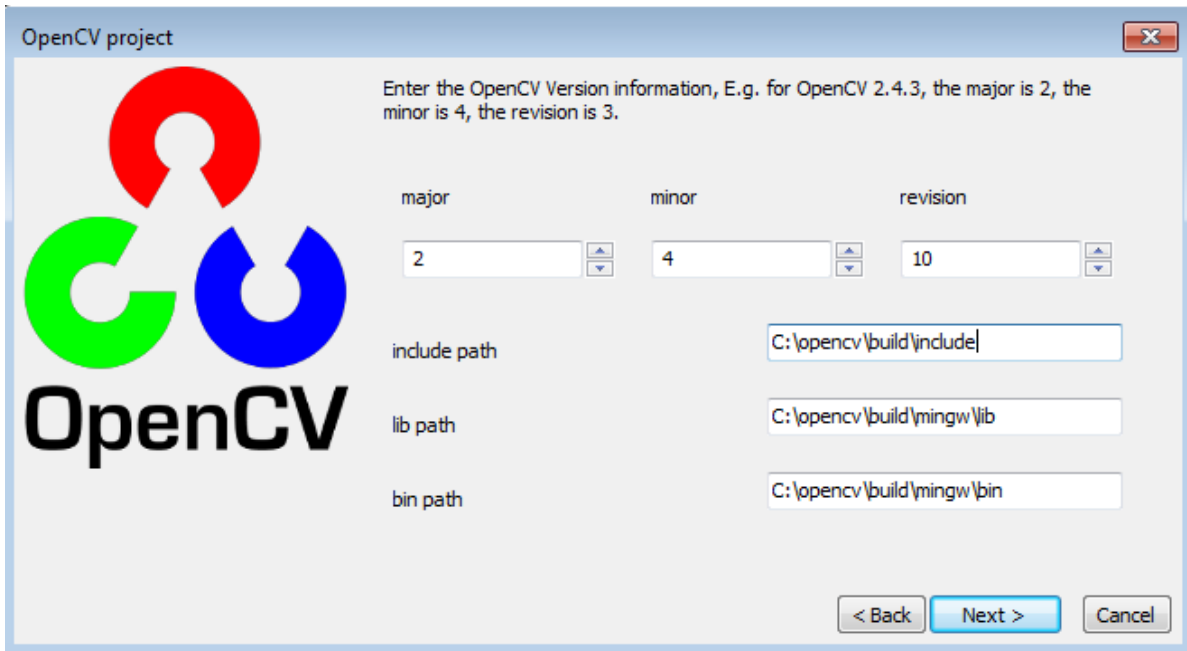


FIGURE 12 CREATION DU PROJET OPENCV 3

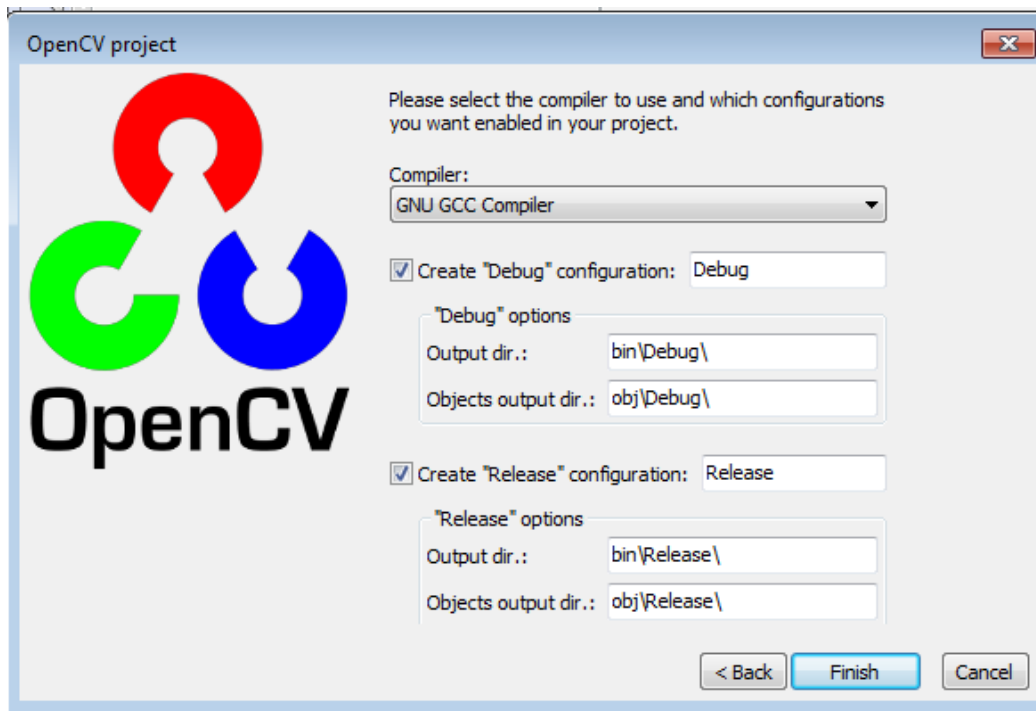


FIGURE 13 CREATION DU PROJET OPENCV 4

Un programme prédéfini est alors créé. Afin de s'assurer de la bonne installation d'OpenCV, il suffit alors de compiler le programme et tester en cliquant sur le bouton « **Build & Run** ». À cette étape, une image pop-up contenant un portrait de Lena <sup>5</sup> devrait apparaître. Dans le cadre contraire, il est conseillé de vérifier les précédents chemins du programme créé.

<sup>5</sup> Lena ou Lenna est une image test standardisé dans le traitement d'image ( <http://en.wikipedia.org/wiki/Lenna> )

## 6. Portage d'un programme OpenCV venant d'un autre PC

Le principe relativement simple, il suffit pour cela de créer un nouveau projet comme précédemment et de reprendre le code source d'un autre programme.

Le procédé sera illustré via un exemple illustratif à partir d'un portage d'un programme Algo1<sup>6</sup> développé sous OpenCV 2.4.9 vers un nouveau programme Test.

- La première étape est de créer un projet opencv tel que le projet Test précédemment créé.
- La seconde étape est de supprimer tous les fichiers issus de ce projet Test en allant sur l'onglet « Project » puis « Remove files ».

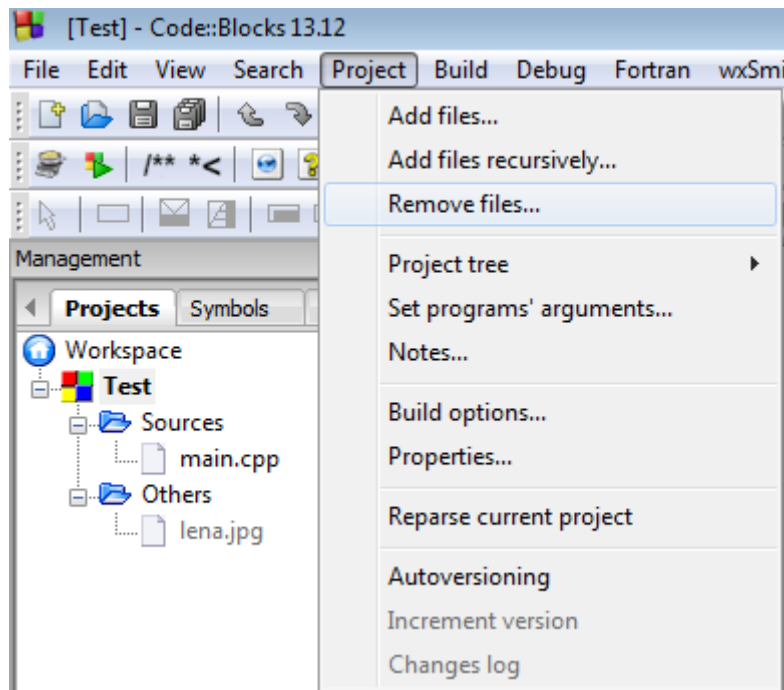


FIGURE 14 SUPPRESSION DES FICHIERS

Une fois ces fichiers effacés du projet, il est recommandé de les supprimer du dossier contenant le projet Test afin de ne pas les récupérer.

- Dernière étape : Ajouter les fichiers sources du nouveau programme

La méthode la plus simple consiste à copier tous les fichiers « \*.c » « \*.cpp » et « \*.h » se trouvant dans le projet du programme Algo1 dans le dossier du projet Test.

<sup>6</sup> Le programme a été développé sur un autre pc doté d'OpenCV 2.4.9 et windows 7 sp1 téléchargeable via [ce lien](#)

## Déploiement d'application OpenCV : de la compilation à la diffusion

À l'issue de cette opération, le dossier contenant le projet Test est similaire à celui présent en figure 15

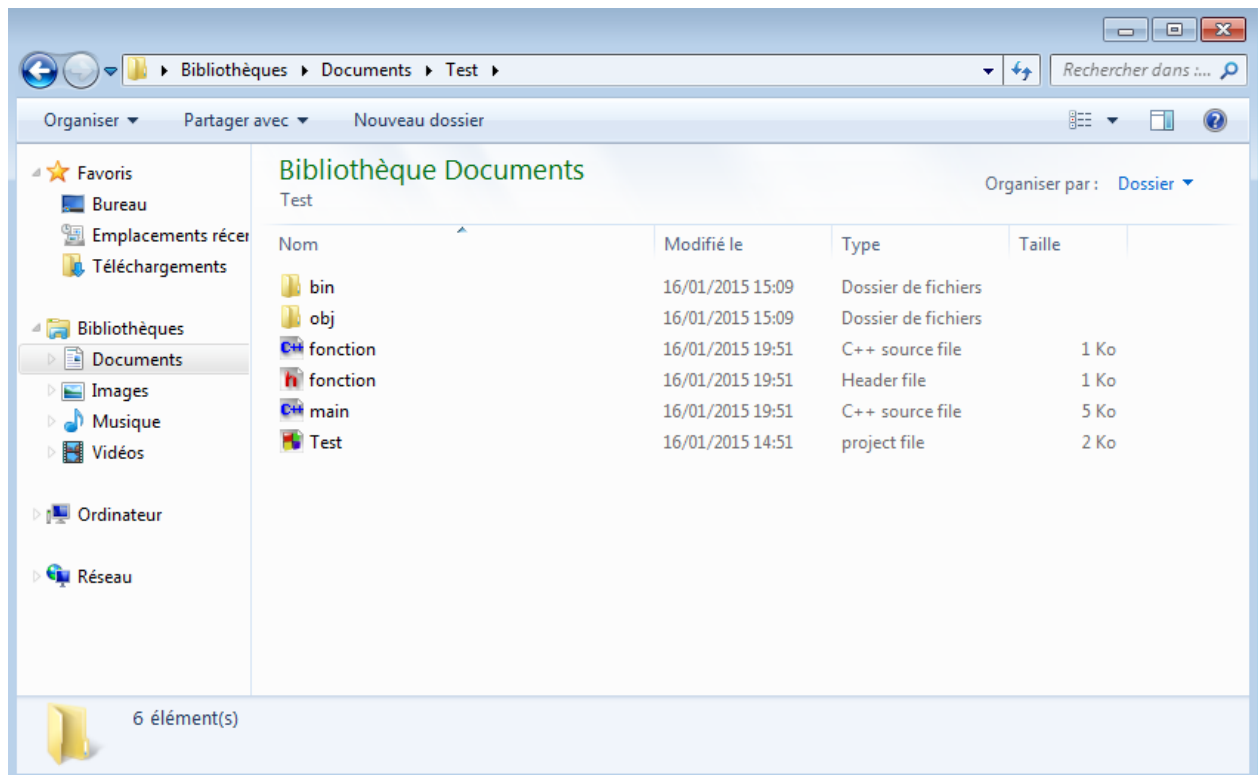


FIGURE 15 CONTENU DU NOUVEAU DOSSIER DU PROJET

Afin de lier ces programmes à l'application OpenCV, il suffit de se diriger sur **Project > Add files** et de sélectionner tous les nouveaux fichiers rajoutés.

Compiler et lancer le programme.

Si tout se passe bien une animation exploitant plusieurs fonctions d'OpenCV devrait se lancer sans erreurs.

### 7. Du programme à l'exécutable

Une fois le programme construit vous pouvez retrouver son exécutable dans le dossier Debug ou Release situé dans le dossier bin du projet du programme OpenCV.

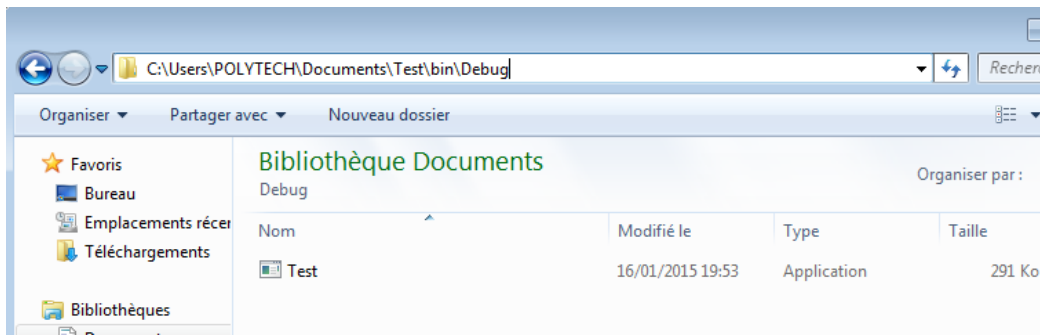


FIGURE 16 FICHIER EXECUTABLE

Suivant la configuration choisie lors de la compilation de la bibliothèque OpenCV via Cmake, le programme est utilisable sous 2 manières différentes.

- Par défaut : Une compilation dynamique du programme

Le programme nécessite des ressources supplémentaires pour pouvoir fonctionner sur un ordinateur tiers.

En effet, en suivant la procédure décrite précédemment, le programme a été construit avec des DDL afin d'obtenir un programme léger et rapide pour un traitement de données très important.

Une bibliothèque dynamique (extension .dll, pour Dynamic Link Library (eng) = Bibliothèque de Lien Dynamique (fr)) se charge une seule fois en mémoire et peut être utilisée par plusieurs exécutables compilés avec OpenCV, réduisant ainsi l'occupation mémoire (RAM) ainsi que l'occupation disque puisque la partie du code commune aux différents exécutables est déportée dans le fichier .dll.

Pour connaître les ddl nécessaires au programme, il suffit au développeur du programme de lancer le programme Dependency Walker téléchargeable sur leur lien [officiel](#).

Le programme indique alors les .DDL manquants de la manière suivante :

## Déploiement d'application OpenCV : de la compilation à la diffusion

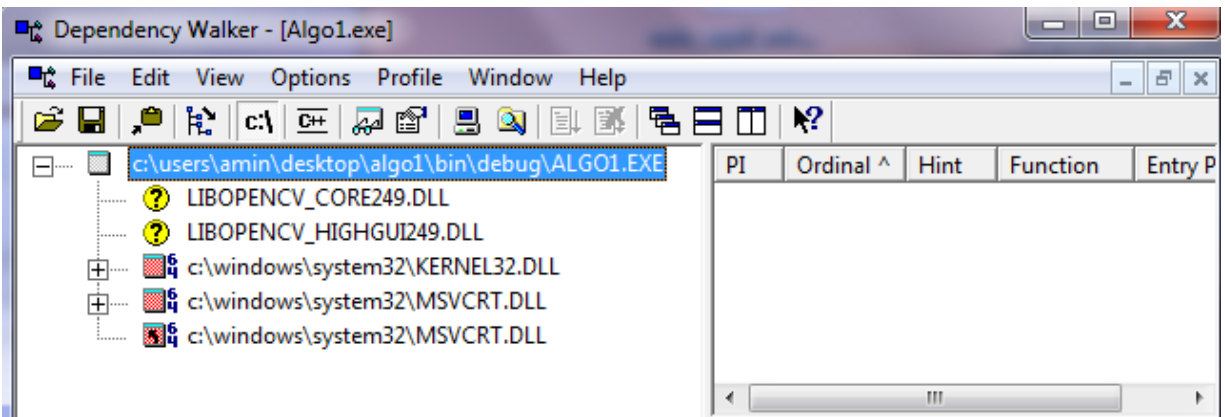


FIGURE 17 DLLS REQUIS PAR LE PROGRAMME ALGO1.EXE

Dependency Walker montre que les DLL : LIBOPENCV\_CORE249.DLL et LIBOPENCV\_HIGHGUI249.DLL font défaut au programme Algo1.exe.

Les applications OpenCV nécessitent des DLL issues de « [C:\opencv\build\mingw\bin](#) »

Il existe alors 2 moyens pour utiliser ce programme :

- Le premier est de placer les différents DDL requis dans le même dossier où se trouve le programme.
- Le second moyen est de placer les DDL dans le dossier « [C:\Windows\system](#) »

### ➤ Une compilation statique du programme

Cette compilation peut être définie au niveau de la compilation faite par le Cmake, en décochant l'option «Build\_share\_Libs ». Cette option désactivée annule le partage des bibliothèques dll au profit des bibliothèques statiques « \*.a ».



FIGURE 18 CONFIGURATION CMAKE POUR DES BIBLIOTHEQUES STATIQUES

Après la compilation, les bibliothèques compilées sont placées dans :

- Dossier « [C:\opencv\build\mingw\lib](#) » dans le cas des bibliothèques statiques (\*.a).
- Dossier « [C:\opencv\build\mingw\bin](#) » dans le cas des bibliothèques dynamiques (\*.dll +\*.a).

Pour finir, il reste à compiler le programme via les bibliothèques statiques et à ajouter l'option « -static » au niveau du linker en paramétrant les options de compilation de la manière suivante :

Sélectionner [Project > Build options > Linker setting](#)

Ajouter dans « [link libraries](#) » les librairies statiques se trouvant dans « [C:\opencv\build\mingw\lib](#) »

Et ajouter dans « Other Linker option » l'option « -static »

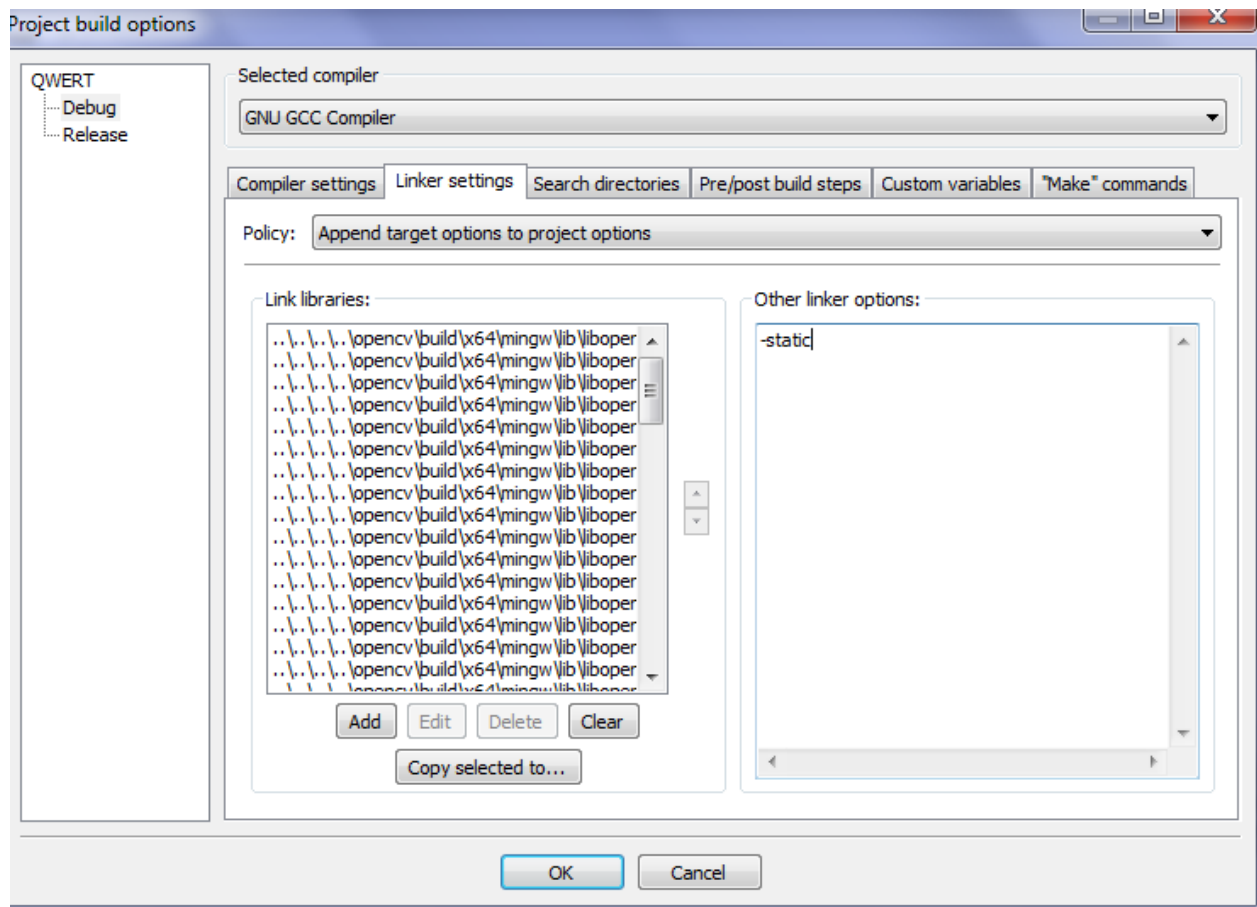


FIGURE 19 CONFIGURATION DANS CODE::BLOCKS

La compilation devrait générer des warnings et un programme exécutable plus gros.

Le programme a alors l'avantage de pouvoir être utilisé sans aucune condition de possession DLL, mais nécessite un temps d'exécution plus long.

## Conclusion

Le protocole d'installation des outils de travaux se révèle nécessaire lors du développement d'un projet. Ceci s'est révélé en effet indispensable ne serait-ce que pour évoluer un programme effectué par une première équipe, ou pour échanger certain programme et fonction entre les membres d'une équipe ou encore pour assurer au client un moyen d'adapter son programme a de nouvelle contrainte technique.