|  |  |
| --- | --- |
| MODULE 1Installation de la bibliothèque de vision par ordinateur OpenCV  | Objectifs de ce module :* Installation de la bibliothèque de programmation en vision OpenCV.
* Configuration de Visual Studio 2015.
* Configuration de CodeBlocks avec Mint 18. (à venir)
 |

***Table des matières***

Sujets Page

MODULE 1 1

Installation de la bibliothèque de vision par ordinateur OpenCV 1

Introduction 3

Présentation sommaire d’OpenCV 3

Installation de OpenCV sous Windows 7 3

*Projet avec Visual Studio 2012 sous Windows 7* 6

Test de la configuration sous Visual Studio 2012 8

*Installation de OpenCV sous Linux (Kubuntu 12.04 )* 10

*Projet avec CodeBlocks sous Kubuntu 12.04* 11

Introduction

La vision par ordinateur est un domaine de l’informatique qui est se situe à la croisée des chemins entre les mathématiques, le traitement de signal et l’intelligence artificielle. En fait, c’est pouvoir extraire l’information d’une scène, d’une image ou d’un flux vidéo, de manière à prendre des décisions en fonction des caractéristiques d’un ou de plusieurs objets qui composent l’image. Donc, être capable de discerner et reconnaître les formes, les couleurs, les textures, les contours, etc.

Nous allons donc utiliser une librairie de vision par ordinateur qui se nomme « 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](http://creativecommons.org/licenses/BSD/) (libre) pour les plate-formes Windows, GNU/Linux, Android et MacOS.

Initialement écrite en C il y a plus de 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.

Afin de mieux vous présenter son étendue et ce qu'elle permet de faire, jetons un œil aux principaux modules accessibles au travers de son API C.

## Présentation sommaire d’OpenCV

* 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 ...
* est optimisée pour les applications temps réel
* fournit une API bas et haut niveau
* est utilisé aussi bien dans les laboratoires de recherche que dans l'industrie

Fonctions :

* manipulation d'images (chargement, sauvegarde, copie, conversion...)
* manipulation et acquisition de vidéos
* manipulations de matrices et algèbre linéaire
* structure de données utilitaires variées (listes, files, ensembles, graphes...)
* traitement d'images (filtrage, détections de discontinuités, morphologie mathématique...)
* analyse d'images (composantes connexes, ajustement de primitives, transformée de distance...)
* vision (calibration de caméra, stéréovision, recherche d'association...)
* reconnaissance de forme (modèles de Markov, ACP...)
* interface graphique (affichage d'images, de vidéos, gestion des évènements...)

Modules :

* *cv* – fonctions principals d’OpenCV.
* *cvaux* – Fonctions auxiliaires (souvent expérimentale).
* *cxcore* – Structures de donnés et module d’algèbre linéaire.
* *highgui* – fonctions pour l’affichage graphique à l’écran.

Installation de OpenCV sous Windows 7

Pré-requis :

 Version Windows : Nous utiliserons dans ce document la version Windows 7 64 bits

 Version OpenCV : 3.2

 Version Visual Studio : 

Tout d’abord, allez chercher la librairie à partir du site Web officiel situé au <http://www.opencv.org>.

* Cliquez sur « Latest Downloads » (Tel que montré par la flèche rouge) 
* Cliquez sur la version 3.2 « OpenCV for Windows ».



* Double-cliquez sur le fichier (opencv-3.2.0-vc14.exe) que vous venez de télécharger pour débuter l’installation.
* Choisissez un emplacement pour l’extraction des fichiers. Nous avons choisi le répertoire « C:\openCV32Source ».

En vous déplaçant dans le répertoire que vous venez d’utiliser pour la décompression, allez dans « opencv » et vous retrouverez une hiérarchie de répertoire ressemblant à ceci :



Le dossier « build/include » contient deux répertoires. Le répertoire « opencv » contient les fichiers d’entête « .h » pour la compatibilité avec les fonctions d’opencv en langage C alors que le répertoire « opencv2 » contient les fichiers d’entête c++ (les fichiers se terminant par .hpp).

Depuis la version 3.0, la bibliothèque est divisée en plusieurs modules que vous devez compiler pour la plate-forme de votre choix. Il est conseillé d’utiliser un utilitaire comme « cmake » pour configurer adéquatement les fichiers de configuration et préparer les fichiers de compilation pour Visual Studio.

# Préparation des fichiers avec cmake

## Téléchargement de cmake

Allez sur le site de cmake : [www.cmake.org](http://www.cmake.org) et téléchargez la version pour Windows 64 bits (ou la version 32 bits « Windows-win32-x86-Installer » si votre système d’exploitation Windows est une version 32 bits) comme montré ci-dessous :

Version 64 bits



Version 32 bits

## Configuration des fichiers avec cmake

Démarrez l’installation de cmake et cliquez sur “next” pour la plupart des options et vous arriverez à la fenêtre ci-dessous.

Cliquez « Next » et terminez l’installation.

Démarrez “cmake” et entrez ensuite les éléments suivants:

**2**

**1**

**4**

**3**

Étape 1: Choisir l’emplacement qui contient les fichiers sources téléchargés précédemment. Dans le cas ci-dessus, il s’agissait de “C:/OpenCV32Source/opencv/sources”

Étape 2: Choisir l’emplacement qui recevra les fichiers produits par la compilation.

Étape 3: Cliquez sur “Configure” et choisissez la version de votre environnement visual Studio. Dans notre cas, il s’agit de Visual Studio 2015, version 32 bits.



 Cliquez sur “Finish” et après quelques secondes, vous recevez une fenêtre ressemblant à ceci:



Acceptez les défauts et cliquez sur “Generate”.

Lorsque vous recevez le message “Generating Done”, vous pouvez cliquer sur “Open Project” pour ouvrir les projets dans Visual Studio. Vous pouvez fermer cmake.

## Compilation avec Visual Studio des fichiers créés par cmake

Lorsque visual studio est ouvert avec les fichiers du projet créés par cmake, vous pouvez alors lancer la compilation avec “Regénérer la solution”.

Ça y est ! Vous pouvez maintenant tester votre premier programme OpenCV.

# Création des variables d’environnement

Nous vous conseillons fortement de créer des variables d’environnement pour spécifier vos dossiers qui contiennent les fichiers précédemment créés.

Les variables d’environnement ont l’avantage d’être plus flexible pour l’usage car au lieu d’entrer directement le nom du répertoire et son chemin, ce qui peut entraîner des problèmes sur autre ordinateur qui ne possèderait pas ces chemins, les variables pointent au bon dossier et vous n’avez qu’à utiliser ces variables dans les options de Visual Studio. De plus, si vous créez les mêmes variables d’environnement sur l’autre ordinateur sur lequel vous travaillez, Visual Studio retrouvera sans problème les dossiers et vous pourrez continuer à travailler sur le projet sans problèmes.

1. Création de la variable pour le dossier des fichiers « Include » :
* Cliquez sur le menu démarrer de Windows
* Cliquez bouton droit de souris sur « Ordinateur »
* Cliquez « Propriétés »
* Dans la fenêtre, cliquez « Paramètres Systèmes Avancés »
* Dans la fenêtre ci-dessous, cliquez « Variables d’environnement » tel que montré par la flèche rouge.



Vous recevez la boîte de dialogue suivante :



* Cliquez « Nouvelle… » dans la portion du bas.

Nous créons alors une variable système et non une variable uniquement utilisable dans le compte de l’utilisateur.

 Dans la boîte de dialogue ci-dessous :

1. Entrez le nom de la variable, nous avons choisi : « OPENCV\_DIRINCLUDE » pour le dossier « include ».
2. Entrez la valeur de la variable, c’est-à-dire, le nom du dossier complet qui contient les fichiers « include » (.h) du dossier initialement compressé de OpenCV. Dans notre cas, il s’agit de « C:\OpenCV32Source\opencv\build\include »

Recommencer les étapes 1 à 2 ci-dessus pour la variable du dossier des bibliothèques :

 Nom de la variable : OPENCV\_DIRLIB

 Valeur : C:\OpenCV32Build\lib\debug

Ajoutez maintenant une variable pour le dossier des fichiers dll.

 Nom de la variable : OPENCV\_DIRDLL

 Valeur : C:\OpenCV32Build\bin\debug

# *Projet avec Visual Studio 2015 sous Windows 7*

Voici les étapes nécessaires à la réalisation d’un projet utilisant la bibliothèque OpenCV et l’environnement de développement Visual Studio version 2015. Le code ne réalisera pas beaucoup de chose au niveau visuel mais nous permettra de tester si notre configuration de Visual Studio est correcte pour un projet OpenCV.

Étape 1 : Démarrez Visual Studio, créez un projet « Application console Win32 » et choisissez un projet vide sans « En-tête précompilé ».

Étape 2 : Dans les propriétés du projet, ajouter les répertoires où se trouve les fichiers d’en-tête « include » et les fichiers de bibliothèque « .lib » pour la compilation et l’édition des liens. Voici un exemple :

**1**

**2**



**3**

1. Dans les répertoires Include, ajoutez la variable $(OPENCV\_DIRINCLUDE)
2. Dans les répertoires bibliothèques, ajoutez  la variable $(OPENCV\_DIRLIB)
3. Cliquez ensuite sur « Propriétés de configuration » et sur « Débogage »

Dans l’élément « Environnement », ajoutez :

 PATH=%PATH;$(OPENCV\_DIRDLL)

1. Dans l’élément « Éditeur de liens », « Entrée » et « Dépendances supplémentaires », entrez les éléments additionnels suivants :

**opencv\_core320d.lib;opencv\_highgui320d.lib;**

**opencv\_imgproc320d.lib;opencv\_imgcodecs320d.lib**

1. Dans « Éditeur de liens », « Général », allez dans « Répertoires de bibliothèques supplémentaires » et ajoutez :

**$(OPENCV\_DIRLIB)**

Test de la configuration sous Visual Studio 2015

#include <iostream>

//Ce fichier d'en-tête contient les éléments et structures de base

#include <opencv2\core\core.hpp>

// Ce fichier d'en-tête contient les éléments nécessaires pour l'affichage à l'écran

#include <opencv2\highgui\highgui.hpp>

using namespace cv; // L'espace de nom pour accéder aux classes

int main()

{

// La classe Mat est la classe principale qui représente une image

 Mat ImgSource;

 return 0;

}

Ajoutez un fichier .cpp vide à votre projet et ajoutez le code suivant :

Compilez et testez le programme.

# *Installation de OpenCV sous Linux (Kubuntu 12.04 )*

Les premières étapes sont très similaires à celles que l’on réalise sous un système de type Windows.

* Tout d’abord, allez chercher la librairie à partir du site Web officiel situé au <http://www.opencv.org>.
* Cliquez sur « Latest Downloads » (Tel que montré par la flèche rouge)



* Cliquez sur la version 3.2 « opencv for Linux/Mac ».

Vous obtenez un fichier se terminant par tar.bz2. C’est donc un fichier compressé contenant tous les fichiers nécessaires. Sous Linux, vous devrez cependant procédé à la compilation de opencv pour votre plate-forme et votre distribution.

* Décompressez le fichier.

tar –jxvf opencv\_version.tar.bz2

Vous obtenez un repertoire “OpenCV-Version” qui contient tous les fichiers nécessaires à la compilation.

Compilation

* Assurez-vous d’avoir téléchargé les paquets suivants avant de procéder :
	+ - cmake et build-essential -> sudo apt-get install build-essential cmake
* Créez un repertoire temporaire pour la compilation des fichiers:
	+ - mkdir OpenCV-Temp
* Déplacez-vous dans ce répertoire et tapez :
	+ - Cmake -D CMAKE\_BUILD\_TYPE=RELEASE -D CMAKE\_INSTALL\_PREFIX=/usr/local .. (Attention les .. sont importants)
* make
* sudo make install

Les fichiers d’entête et les fichiers de bibliothèque sont compilés pour votre plate-forme et se trouvent dans le répertoire que vous avez mentionné pour l’option « CMAKE\_INSTALL\_PREFIX ». Dans notre exemple, le répertoire se trouve dans « /usr/local ».

# *Projet avec CodeBlocks sous Kubuntu 12.04*

On suppose que vous avez déjà compilé OpenCV pour votre plate-forme comme montré dans la section précédente.

* Ouvrir « CodeBlocks » (Si codeblocks n’est pas installé -> sudo apt-get install codeblocks)

Ouvrez un projet de type « Console Application ».

On doit indiquer à codeblocks où se situent les répertoires contenant les fichiers d’en-têtes et de bibliothèque pour pouvoir effectuer la compilation de notre projet OpenCV.

* Indiquez le répertoire des fichiers d’en-tête
* Cliquez Project -> Build Options
* Sous l’onglet « Search Directories », cliquez l’onglet « Compiler » et ajoutez le répertoire « /usr/local/include »
* Sous l’onglet « Linker », ajoutez le répertoire « /usr/local/lib ».
* Sous l’onglet « Linker Settings », ajoutez les fichiers suivants : libopencv\_core, libopencv\_highgui.
* Cliquez le bouton « Valider » pour continuer.

Puisque le code réalisé sous la plate-forme Windows est portable, vous pouvez utiliser le même code dans l’exemple sous Linux.

* Procédez ensuite à la compilation et l’édition des liens sous codeblocks. Tapez la touche « F9 » pour lancer la compilation et exécuter le programme.
* Si vous obtenez « Test réussi ! », tout est alors fonctionnel.