

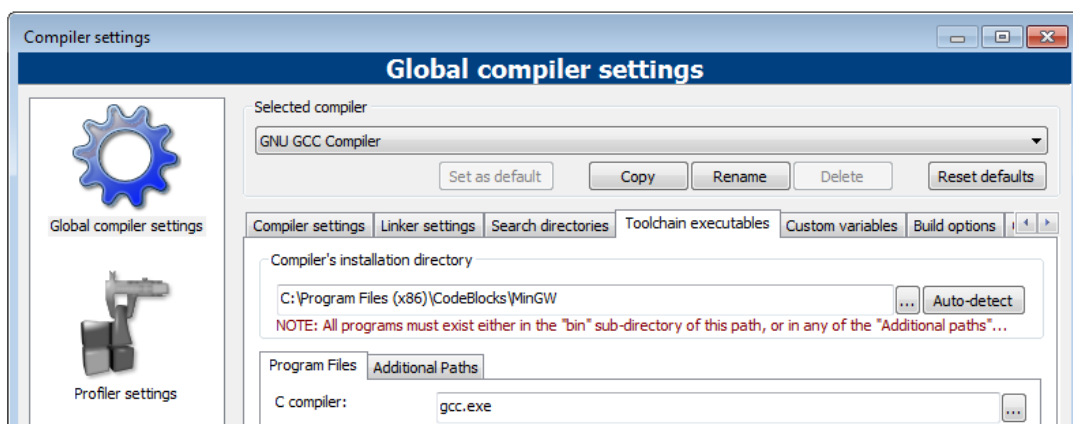
TP traitement d'images 1

1 Configuration du compilateur et du debugger (Windows)

Le compilateur MinGW existe en 2 versions :

- MinGW (anciennement mingw32), qui est livré avec CodeBlocks et génère du code 32 bits
- MinGW-w64, qui génère du code 64 bits

Les bibliothèques que vous allez utiliser ont été compilées avec MinGW-w64. Les programmes que vous allez écrire doivent l'être aussi. Dans CodeBlocks, allez dans le menu **Settings** > item **Compiler...** > onglet **Toolchain executables** :



Dans la zone de texte **Compiler's installation directory**, modifiez le chemin afin que MinGW-w64 soit utilisé à la place de la version par défaut : C:\TDM-GCC-64.

Dans la zone de texte **C compiler** sélectionnez gcc.exe.

Dans la zone de texte **C++ compiler** sélectionnez g++.exe.

Dans la zone de texte **Linker for dynamic libs** sélectionnez g++.exe.

2 Présentation et mise en place

Le but de ce TP est de vous faire implémenter deux outils de base du traitement d'images (la binarisation et la convolution), afin de les utiliser pour résoudre une tâche classique d'analyse (extraction des contours significatifs d'un objet dans une image).

Pour chaque question, il vous est demandé de copier/coller vos résultats dans un rapport. A la fin de la séance, vous enverrez votre rapport au format pdf ainsi que le fichier TP.cpp à l'adresse : julien.olivier2@insa-cvl.fr

Afin de démarrer, il vous est fourni un programme en C++ sous openCV implémentant le négatif d'une image. Sous CodeBlocks, ouvrez le projet TI_TP1.cbp. Le dossier opencv-3.4.1-x64 et le projet TI_TP1 doivent partager le même dossier parent dans l'arborescence des fichiers (si ce n'est pas le cas, vous devrez modifier les chemins pour le compilateur et le linker dans les *Build options* du projet). Compilez et exécutez.

Si tout se passe bien, le programme affiche une image initiale et, après un appui sur le clavier, une image traitée (le traitement crée le négatif de l'image initiale).

Dans un soucis de simplicité, l'image d'origine est convertie en réels dès le départ. **Vous n'utiliserez que des images de ce type au cours de vos traitements.** Attention : avant d'afficher une image composée de réels

avec `imshow()` il est nécessaire de diviser chaque valeur par 255 afin de la normaliser entre 0 et 1 (voir dans le code).

3 Implémentation des outils de bases

3.1 Binarisation

Inspirez-vous de la fonction `negatif()` afin d'écrire une fonction `binarisation()`. Celle-ci prendra en paramètre le seuil de binarisation.

Déterminez le meilleur seuil de binarisation pour l'image *circuit.bmp*. **Inscrivez ce seuil dans votre rapport et ajoutez une image avant et après binarisation.**

3.2 Convolution

implémenter une fonction `convolution()` permettant d'obtenir la réponse de la convolution d'une image avec un filtre passé en paramètre.

Quelques points-clés :

- La taille des filtres ne sera pas fixée.
- Les pixels de la couronne ne seront pas traités.
- Simplifiez-vous la vie : les filtres seront également des `cv::Mat`.

Appliquez un filtre moyenneur à l'image *brain.jpg*. **Vous indiquerez dans votre rapport le filtre utilisé et ajouterez une image avant et après traitement.**

Pour les petits malins : il vous est demandé d'écrire une fonction de convolution, pas d'utiliser celle présente dans `openCV` :)

3.3 Bonus

Gérez les pixels sur la couronne de l'image d'origine à l'aide d'un *constant padding* (même remarque pour les petits malins).

4 Analyse

4.1 Gradients

A l'aide des fonctions implémentées précédemment, réalisez une extraction des contours de l'image *lego.jpg* en utilisant les filtres de sobel.

L'image résultant de chaque étape devra apparaître dans votre rapport :

- Gradients en X.
- Gradients en Y.
- Norme du gradient.
- Binarisation afin de réduire les contours parasites.

4.2 Encore des gradients

Appliquez la chaîne de traitement de la question précédente à l'image *brain.jpg* et **ajoutez l'image résultat à votre rapport**. Quelles remarquez-vous ?

Proposez un traitement afin de résoudre le problème. **Vous indiquerez votre choix dans votre rapport et ajouterez une image de chaque étape.**