30/09/2013

Guénon Marie et Favreau Jean-Dominique

Master SSTIM

Compression des images numériques

Compte rendu TD1 : transformée en ondelettes, filtre de Haar

Table des matières

[Préambule : 2](#_Toc368316217)

[Librairies : 2](#_Toc368316218)

[Filtre de Haar : 2](#_Toc368316219)

[Code : 2](#_Toc368316220)

[Annexes : 3](#_Toc368316221)

**But :** réaliser une transformée en ondelettes (analyse / synthèse) au moyen du filtre de Haar.

# Préambule :

Après une première approche avec Scilab, nous avons préféré utiliser le langage de programmation C++ pour implémenter ce projet. En effet, ce langage nous permet d’avoir une approche simplifiée du traitement d’image ainsi que des calculs optimisés : les calculs matriciels et les itérateurs sont déjà implémentés et optimisé pour le genre de calculs que nous allons devoir effectuer. Nous obtiendrons donc ainsi de meilleurs résultats, autant au niveau qualitatif, qu’au niveau de la rapidité d’obtention.

# Librairies :

De manière native, C++ ne sait pas charger d’image. Nous avons donc du installer la librairie :

OpenCv2

Cette librairie permet de charger, afficher et traiter les images.



# Filtre de Haar, Analyse :

On applique les quatre filtres différents à chaque point de notre image de départ :

* Moyenne des pixels avoisinants, on applique le filtre : et on obtient



* Gradient X, on applique le filtre : et on obtient



* Gradient Y, on applique le filtre : et on obtient



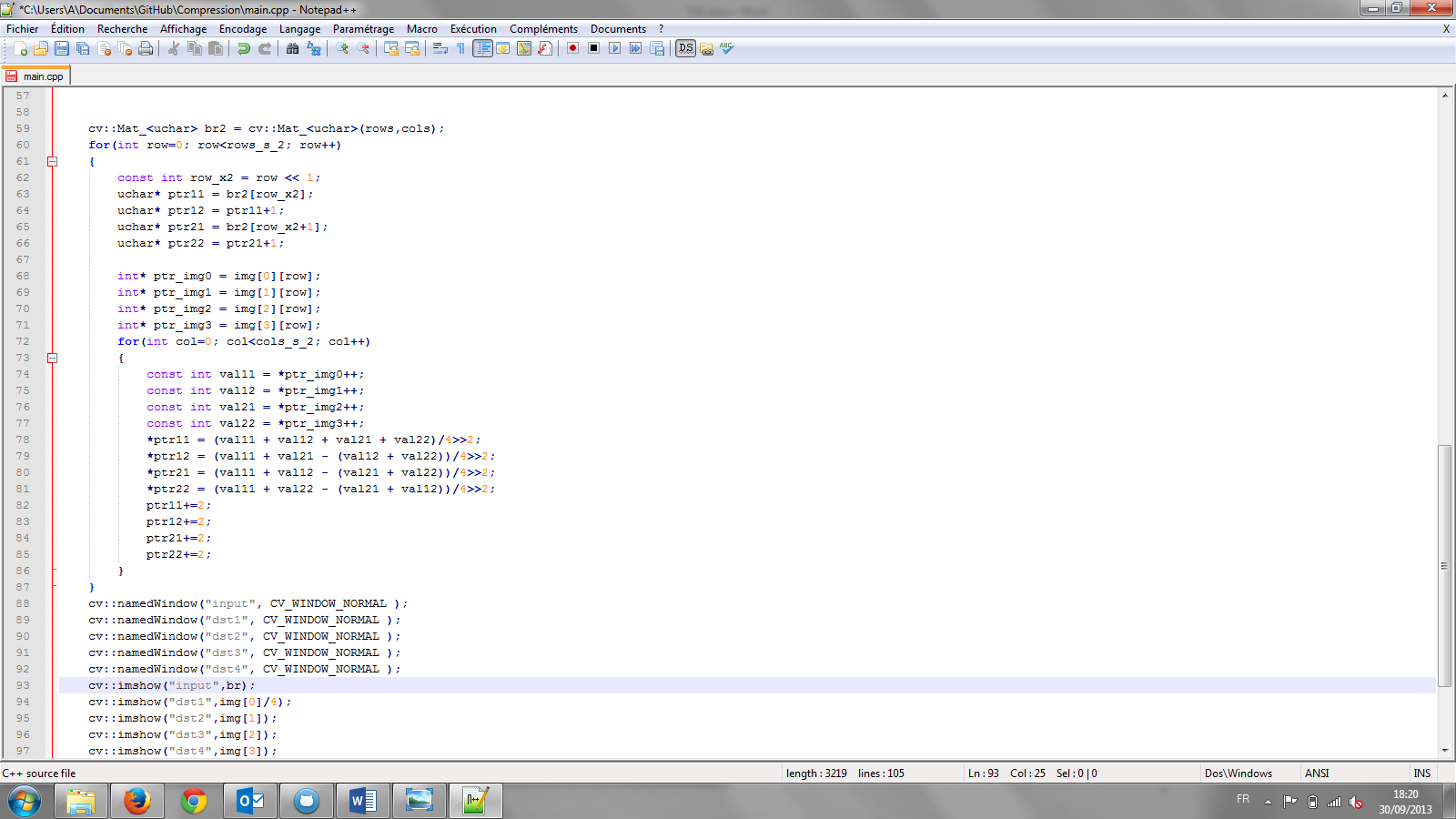
* « Gradient diagonal », on applique le filtre : et on obtient



# Reconstitution, Synthèse :

Comme vu précédemment, nous avons une image de départ qui a été analysée en quatre images grâce aux filtres décrits ci-dessus :

Pour reconstituer l’image de départ, Il suffit d’inverser le système présenté ici et nous obtenons les formules suivantes :



# Annexes :