



PROJET DE TRAITEMENT D'IMAGES

Compte-Rendu

12/01/2014

Adrien Bertrand

ISEN Toulon - M1B

Table des matières

Présentation	2
Fonctionnalités.....	2
Communes aux deux versions.....	2
Spécifiques à la version console.....	2
Spécifiques à la version GUI	2
Manuel d'utilisation	3
Version console	3
Version GUI.....	4
Remarques générales.....	4
Exemples	5
Quelques commentaires	7
Problèmes connus.....	8
Notes	8
Améliorations possibles	8

Présentation

Le projet consiste à créer un programme comprenant des fonctions de base de traitement d'images, avec quelques outils, des filtres, et des fonctions d'analyse.

En plus de la version en ligne de commande intégralement programmée en C, j'ai réalisé une version GUI multi-plateforme grâce à Qt (avec un « wrapper » C++ pour faire le lien avec le moteur en C).

Fonctionnalités

Communes aux deux versions

- ✓ **Outils :**
 - Négatif
 - Niveau-de-gris
 - Histogramme (standard et cumulé)
 - Seuillage manuel
 - Seuillage automatique
- ✓ **Filtres :**
 - Morphologiques (dilatation, érosion, ouverture, fermeture)
 - Pondérés (Moyenneur, Gaussien - 3x3, 5x5, et personnalisé-)
 - Gradients (Roberts, Prewitt, Sobel)
 - Laplacien (4 et 8)
 - Médian
- ✓ **Analyse :**
 - Egalisation d'histogramme
 - Parallélogramme englobant *
 - Réduction de palette *
 - *Watershed* (ligne de partage des eaux)

Spécifiques à la version console

- ✓ E/S au clavier
- ✓ Drag-and-drop d'un fichier .bmp sur l'exécutable pour le charger directement
- ✓ Encodage spécifique pour autoriser les accents sur la console (si supporté)

Spécifiques à la version GUI

- ✓ Drag-and-drop de l'image source pour l'ouvrir directement
- ✓ Pour les grandes images, les 2 zones (original / modif.) sont « défilables » (*ScrollArea*), et le sont de manière synchronisée, afin de permettre une comparaison visuelle optimale des images.
- ✓ Affichage du résultat des outils (négatif, niveau-de-gris, histogrammes...) dans une popup à part.
- ✓ Menus
- ✓ Raccourcis clavier

* La version GUI désactive ces options pour cause d'implémentation incomplète dans le wrapper C++.

Manuel d'utilisation

Version console

La version en ligne de commande accepte d'être lancée avec en paramètre le chemin d'une image .bmp à charger directement (ce qui rend le drag'n'drop possible sur l'.exe, par ailleurs). Si l'exécutable est lancé tel quel, celui-ci essaiera d'ouvrir un fichier "image.bmp" situé dans son dossier. Si ce fichier n'existe pas, un message d'erreur est affiché et le programme se quitte.

Une fois une image chargée, l'utilisation du programme est simple : il suffit de rentrer le numéro de l'opération voulue parmi les choix proposés à l'écran dans le menu.



```
*****
***** Adrien BERTRAND *****
*** TD Traitement d'images ***
***** v4.5 12/01/2014 *****
*****

Image en cours : image.bmp

* 1) Negatif d'une image
* 2) Image couleur -> NG
* 3) Seuillage manuel d'une image
* 4) Histogramme d'une image
* 5) Seuillage automatique
* 6) Filtres...
  * |____ 61) Morphologiques...
    * |____ 611) Dilatation
    * |____ 612) Erosion
    * |____ 613) Ouverture
    * |____ 614) Fermeture
  * |____ 62) Pondérés...
    * |____ 621) Moyenneur
    * |____ 622) Gaussien 3x3
    * |____ 623) Gaussien 5x5
    * |____ 624) Gaussien NxN...
  * |____ 63) Gradients...
    * |____ 631) Roberts
    * |____ 632) Prewitt
    * |____ 633) Sobel
  * |____ 64) Laplacien (4)
  * |____ 65) Laplacien (8)
  * |____ 66) Median
* 7) Divers...
  * |____ 71) Réduction de la palette...
  * |____ 72) Histogramme cumulé
  * |____ 73) Egalisation d'histogramme
  * |____ 74) Rectangle englobant
* 8) Watershed
-----
* -1) Tout lancer
-----
* 0) Quitter

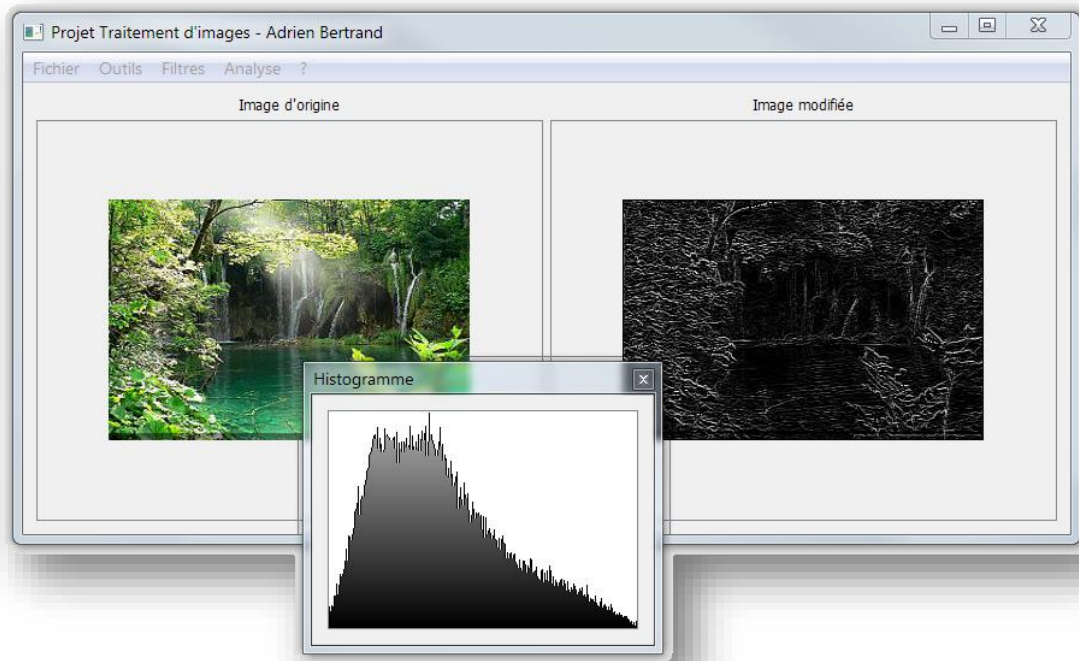
*****
Choix ?
```

Version GUI

L'avantage de la version avec interface graphique est que son utilisation en est d'autant plus simple.

Une fois le programme ouvert, il suffit d'ouvrir une image .bmp (soit par le menu, soit par raccourci-clavier : Ctrl+O, soit par drag'n'drop directement). Celle-ci s'affichera alors sur le panneau de gauche.

Vous pouvez désormais choisir des outils, filtres etc. à partir des menus et voir leur application sur l'image dans le panneau de droite du logiciel.

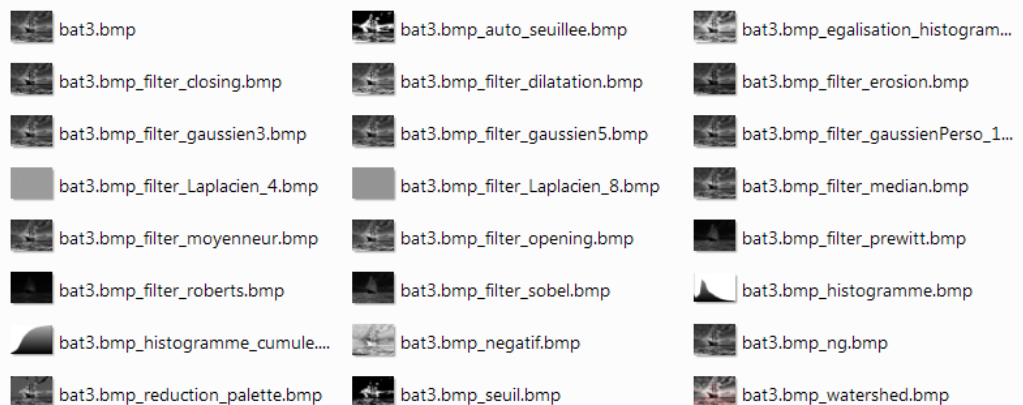


Note : les outils affichent leur produit dans une fenêtre popup pour par souci de clarté.

Attention : cette version ayant été compilée en mode Debug, il faut posséder les bibliothèques (dll) nécessaires afin de pouvoir lancer le programme. Installer l'environnement de développement Qt suffit.

Remarques générales

- Lors de l'application d'un outil, filtre, ou fonction d'analyse, un fichier résultat est créé dans le même dossier que l'image source avec le nom correspondant à ce qui a été effectué :



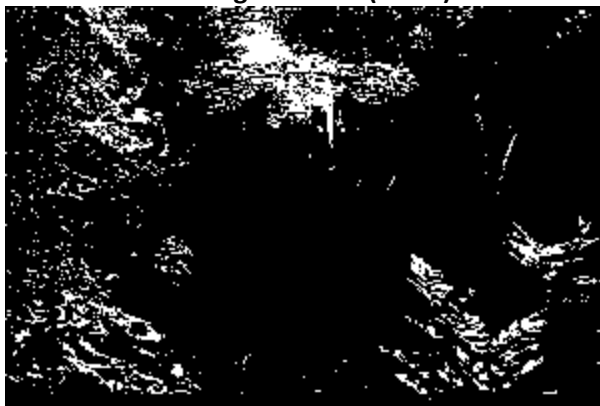
- Au lieu de tester les fonctions une par une, il est possible de lancer un test global qui va les exécuter à la suite automatiquement (choix -1 en console, ou depuis le menu Outils en GUI).

Exemples

Image d'origine :



Seuillage manuel (à 200)



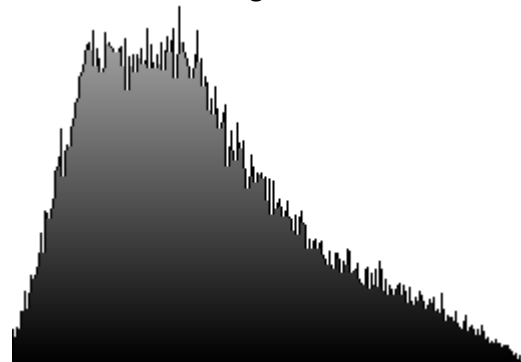
Seuillage automatique



Egalisation d'histogramme



Histogramme





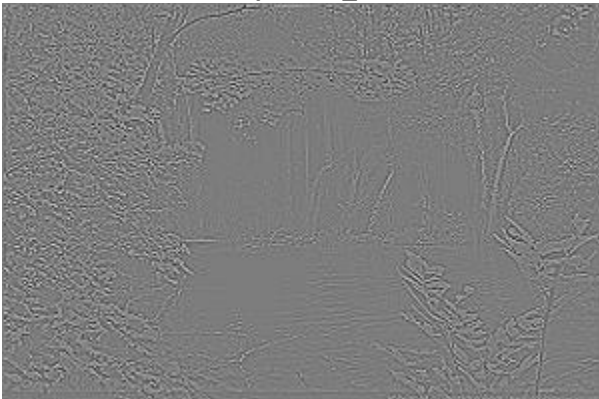
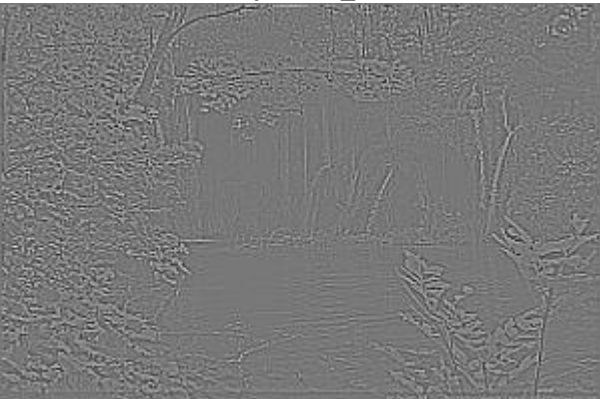




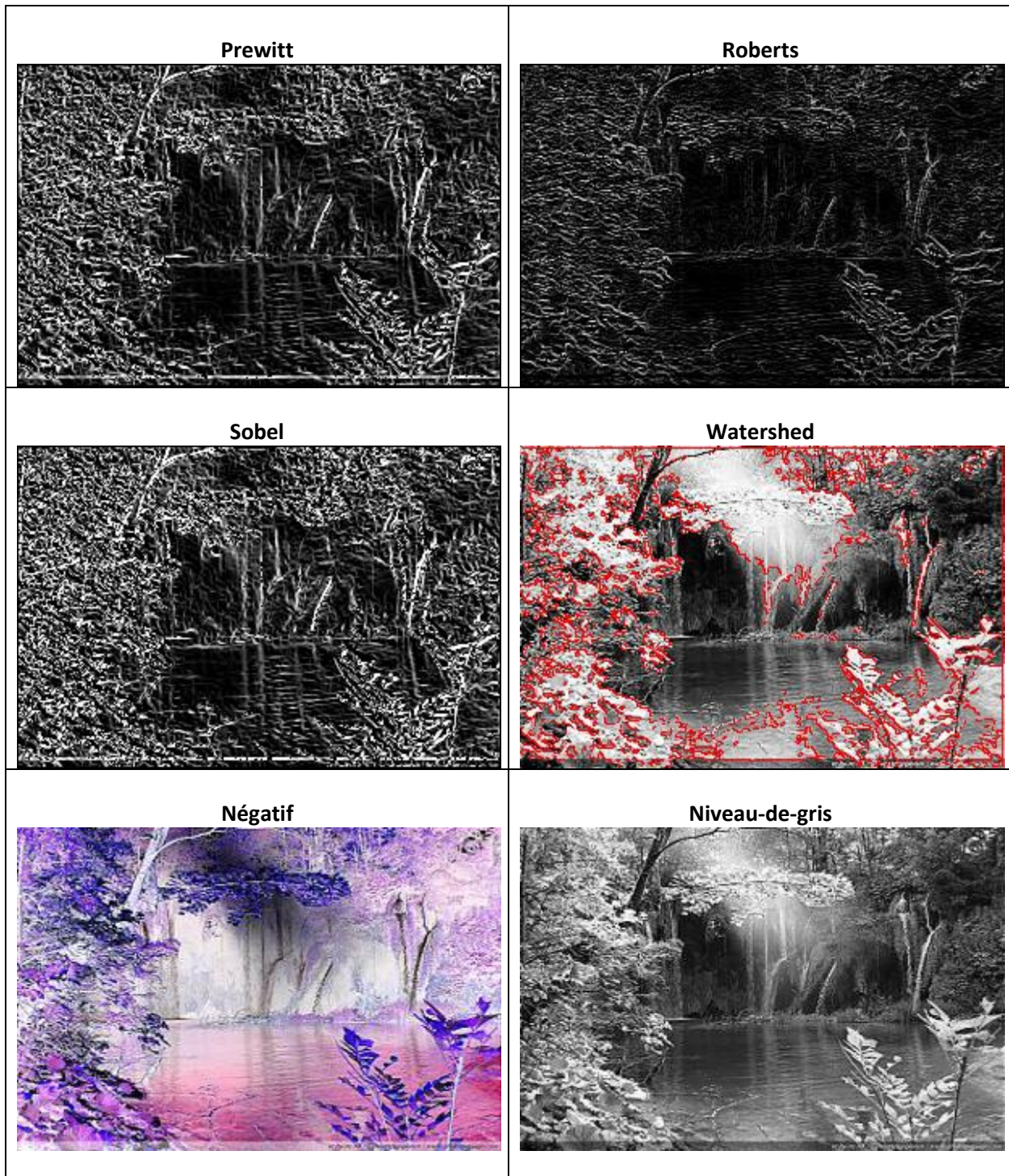
Dilatation



Erosion



<p>Ouverture</p> 	<p>Fermeture</p> 
<p>Gaussien 3x3</p> 	<p>Gaussien 5x5</p> 
<p>Laplacien_4</p> 	<p>Laplacien_8</p> 
<p>Médian</p> 	<p>Moyenneur</p> 



Quelques commentaires

Outils : leur algorithme et implémentation sont triviales, sauf peut-être pour le seuillage automatique où [l'algorithme d'Otsu](#) a été utilisé.

Filtres : La plupart des filtres sont implémentés avec des matrices/masques (dont une partie est trouvable dans le cours), certains étant réalisés grâce à l'opération de convolution sur l'image originale.

Analyse : Le rectangle englobant a été réalisé d'une manière simple et non-dynamique, par rapport à un fond supposé noir – son implémentation est là aussi triviale.

En ce qui concerne le Watershed, 4 algorithmes/implémentations ont été testés (avec plus ou moins de succès), notamment avec du multithreading ; par exemple : l'algorithme de Vincent-Soille.

Problèmes connus

- La totalité de la mémoire allouée n'est pas libérée (environ 5% de pertes selon *Valgrind*)
- La version GUI n'est pas encore toujours totalement stable au bout d'un certain temps
- En mode console, lors du choix de la commande, rentrer un symbole ('*', '.'...) fait rentrer le menu programme dans une boucle infinie
- Le *Watershed* n'est pas totalement fonctionnel – non pas que le programme plante, mais les résultats ne sont pas forcément toujours ceux attendus

Notes

Quelques faits/chiffres ...

- Environ 3000 lignes de code+headers (≈2500 réelles)
- Testé avec succès sur Windows 7, Ubuntu 13.10, et Mac OS X 10.8
- L'utilisation d'OpenMP réduit assez le temps que met le programme à réaliser telle ou telle tâche, mais n'est pas efficiente lorsque toutes les tâches sont lancées à la suite (choix '-1'). La création des threads, notamment, doit prendre du précieux temps...
- Le projet a été réalisé et compilé en mode Release avec l'IDE *Visual Studio 2013* de Microsoft, mais il s'avère que la compilation avec *gcc* génère du code bien plus rapide (de 1,2x à 2x). (tous les deux avec le niveau d'optimisation -O2)
- L'exécutable GUI a été généré par QtCreator 3.0, basé sur Qt 5.2, et en mode Debug
- La documentation HTML est générée par Doxygen 1.8.5, avec GraphViz/dot pour les schémas
- Le projet est open-source en licence GPL3, et est disponible sur <http://github.com/adriweb/>

Améliorations possibles

- Résolution des bugs connus
- Application d'un filtre défini par une matrice définie par l'utilisateur
- Choix de l'application des filtres sur l'image de départ ou successivement sur l'image déjà en traitement.
- Gestionnaire d'Undo/Redo
- Possibilité d'appliquer les filtres sur les images couleurs (en fait, faire les calculs sur les 3 couches et reconstituer la couleur)
- Gestion des autres formats classiques d'images (JPG, png...), par conversion automatisée
- Gestion du zoom (avec synchronisation) dans l'interface graphique
- ...