

**Ecole Nationale des Sciences Appliquées**

0

**El Mekki Rachid**

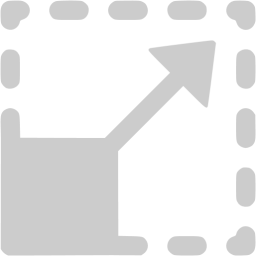
**Boufardi Nabil**

**& El Bouaichi Ilham**

**Berrada Zineb**

**El Mansouri Yasmine**

**Equipe**

**

**Redimensionnement**

**Thème  Traitement d’images**

**Niveau 1ère année Cycle Ingénieur**

**Filière Génie Informatique**

**Encadrant Mme M. Zrikem et Mr N. El Marzouqi**

**Date de Remise  29/06/2017**

**:**

**:**

**:**

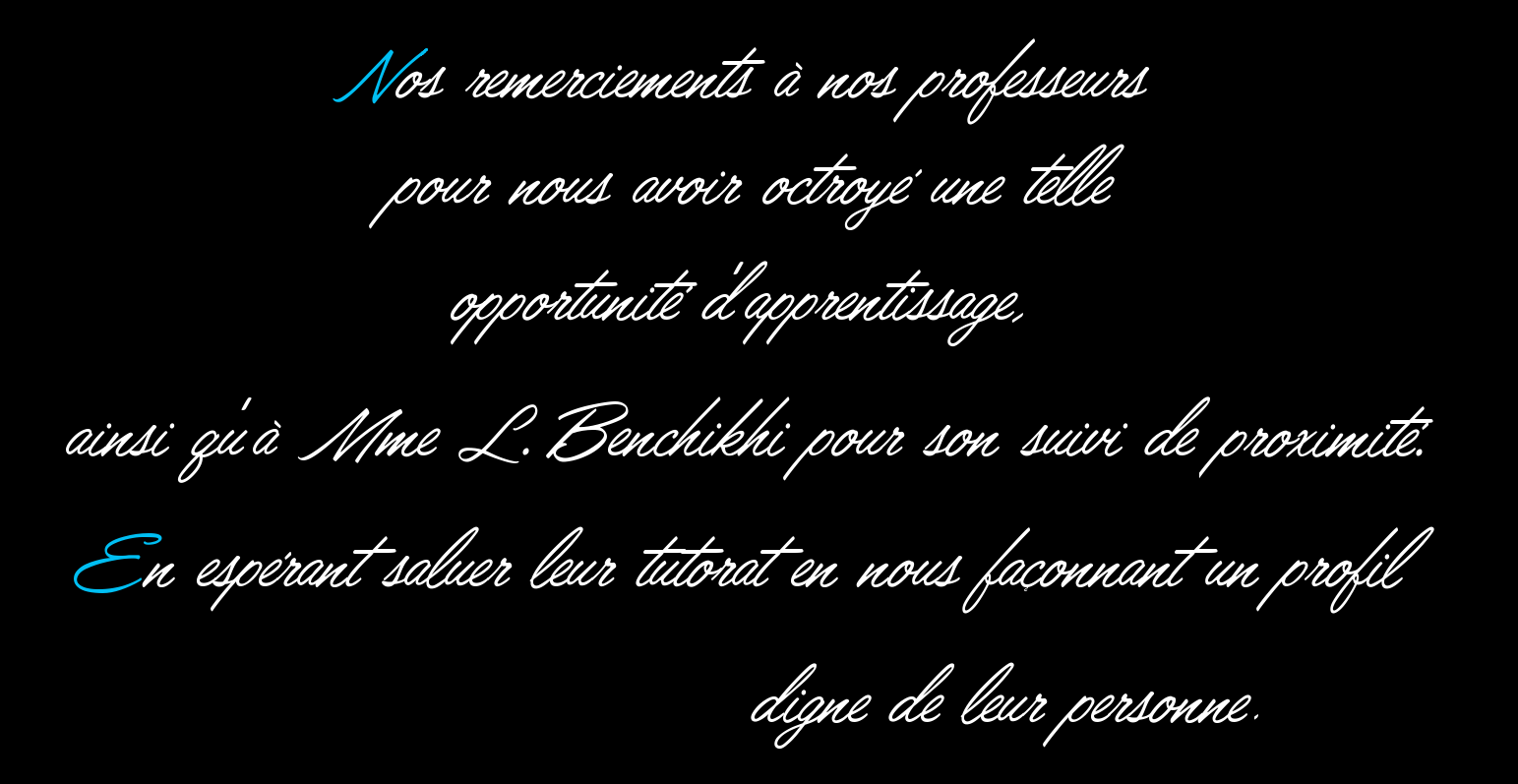
**:**

**:**

**:**

Rapport Final du Projet

**De Marrakech / Génie Informatique**



**Sommaire**

**1 Bref Positionnement**

**2 Cahier de Charge**

**3 Calendrier d’Avancement**

**4 Interface**

**5 Etude Mathématique**

**6 Implémentation**

**7 Appréciations**



**Bref Positionnement**

**Le redimensionnement ou le ré-échantillonnage est une technique qui permet de créer une nouvelle version d'une image avec une taille différente.**

**L'augmentation de la taille est appelée sur-échantillonnage, et la réduction de la taille de l'image est appelée sous-échantillonnage.**

**Il s'avère que ces opérations ne sont pas sans perte. Par exemple, si on sous-échantillonne une image puis sur-échantillonne, on obtient un résultat légèrement différent de l'image initiale.**

**De plus, le redimensionnement n'est pas un processus trivial. Il implique nécessairement un compromis entre efficacité, lissage et netteté.**

**Il nous semble ainsi qu’une explication des critères : lissage et netteté s’avère importante pour une bonne compréhension du projet, surtout qu’on peut déjà anticiper avoir à prioriser l’un au dépend de l’autre selon l’image traitée.**

**Principe**





**Netteté**

**La netteté, à savoir Flou, netteté ou contraste, d'une photographie recouvre en fait deux notions : la résolution et le piqué. La résolution se réfère au niveau de détails que l'on peut distinguer ; c'est une qualité de l'appareil liée approximativement en numérique au nombre de pixels. Le piqué, qui se traduit par des contours bien définis, est lié à la qualité de l'objectif, à la précision de la visée et de la mise au point ; lui seul peut être amélioré**

**Lissage**

**Le Lissage est une opération importante en traitement d'images, utilisée pour atténuer un bruit qui corrompt l'information, généralement avant un autre traitement.**

**par un traitement ultérieur.**

**L’interface**

**Afin de supporter notre travail, nous avons réalisé une interface graphique à l’aide de Qt.**

**Brièvement, Qt est une API orientée objet et développée en C++ par Qt Development Frameworks, elle offre des composants d'interface graphique (widgets), d'accès aux données, de connexions réseaux, de gestion des fils d'exécution, d'analyse XML, etc.**

**Qt tient sa renommée de différentes caractéristiques à savoir :**

* **Elle permet la portabilité des applications qui n'utilisent que ses composants par simple recompilation du code source.**
* **C’est une bibliothèque multiplateforme, elle supporte différents environnement dont les Unix qui utilisent le système graphique X Window System ou Wayland, Windows, Mac OS X et également Tizen.**
* **Qt supporte des bindings avec plus d'une dizaine de langages autres que le C++, comme Ada, C#, Java, Ruby, Visual Basic, etc.**
* **Qt est notamment connu pour être le framework sur lequel repose l'environnement graphique KDE, l'un des environnements de bureau par défaut de plusieurs distributions GNU/Linux.**

**Qt**

**C’est quoi ?**

**L’interface est divisée en trois grandes parties à savoir :**

* **La barre de menu : composée d’un sous menu développable laissant apparaître des options :**

**Nouveau  : pour choisir une image.**

**Enregistrer : pour enregistrer l’image traitée.**

**Quitter  : pour quitter l’application.**

**Chaque option est munie d’une icône qui nous servira comme référence dans la deuxième partie ainsi qu’un raccourci clavier.**

**La boîte à outils est divisée en quatre grandes parties :**

* **Taille : Où l’utilisateur peut entrer la taille de la nouvelle image, la première zone est dédiée au nombre de lignes et la deuxième aux colonnes.**
* **Critères : Renferme les options que désire ou tolère l’utilisateur, ce dernier n’a qu’a cocher ce qu’il veut le choix de l’algorithme se fera suivant ce qu’il a choisit.**
* **Démarrer : Pour valider les entrées et lancer le programme.**
* **Redimensionnement rapide : Au cas où l’utilisateur veut rapidement redimensionner l’image sans choisir de critère, un algorithme prédéfini sera exécuté.**
* **La fenêtre principale : Où seront affiché tout les traitements effectué, on y trouve une sous fenêtre « Boîte à outils ».**

**La barre est menu d’un séparateur entre sous menu, au cas où il y’en aurai plusieurs.**

**La barre peut être placée n’importe où dans la fenêtre.**

* **La barre d’outils rapide : composée d’un accès rapide aux options de la barre de menu, chaque option est référencée par son icône, comme on peut le voir grâce à l’exemple, figure suivante :**

**Lorsque l’utilisateur choisi une image une sous fenêtre apparaît contenant l’image choisie :**

**Lorsque l’utilisateur choisi la taille, les critères et appuie sur « Démarrer » ou lorsqu’il appuie simplement sur « Redimensionnement rapide », une autre fenêtre apparaît contenant l’image traitée, figure suivante**

**Comme c’est une fenêtre elle peut être agrandie, réduite ou même fermée.**

**L’agrandissement de la fenêtre n’affecte, bien évidement, pas la taille de l’image.**

**L’utilisateur peut choisir une autre image mais c’est la dernière qui sera gardée en mémoire et ainsi traité lorsqu’il appuie sur « Démarrer »**

**Comme pour la précédente elle peut être agrandie, réduite ou même fermée.**

**Encore une fois l’agrandissement n’affecte pas la taille de l’image.**

**L’utilisateur peut enchainer les traitements sur la même image pour comparer les résultats. Afin de connaître un peut les algorithmes utilisées, ou juste savoir si la méthode a changé d’un traitement à un autre, le titre de la fenêtre porte un nom significatif a chaque traitement. (voir figure suivante)**

**Spline Cubique**

**Comment ?**

**Dans le redimensionnement comme beaucoup d’autres concepts il est impératif de préserver un degré de lissage acceptable lors de la représentation des objets, ici des images. Les interpolations polynomial sont très délicates et peuvent conduire à des résultats bien loin de la réalité, notamment à cause du phénomène de Runge\*.**

**Il est donc préférable pour mesurer la régularité d’une image, ainsi que sont lissage d’utiliser les dérivées d’une fonction, en effet, plus la fonction est différentiable, plus elle est régulière et plus la courbe qui lui est associée est lisse. Si on utilise un polynôme de degré 1, par exemple, pour éviter le phénomène de Runge, on utilisera des droites pour relier chaque point, ce qui donne la figure suivante :**

En effet on peut voir que ce n’est pas très lisse, cette méthode est appelée méthode des « Spline linéaire », elle est continue mais n’est pas dérivable aux extrémités.

**Ainsi, pour obtenir un lissage de qualité et ne pas tomber dans le piège du polynôme de grand degré, une méthode appelée « Spline Cubique » est née. Cette dernière représente un compromis très intéressant entre la régularité et le degré des polynômes utilisés, puisque qu’elle utilise des polynômes de degré 3, et donne des résultats comme le montre la figure suivante :**

**On voit bien que la courbe est plus lisse sur les extrémités, ceci est dû au fait que la fonction est continue et dérivable en tout point.**

Une image est découpée en plusieurs colonnes, chacune contient plusieurs lignes, ainsi si nous nous concentrons juste sur les lignes on aura un intervalle de [0,l] où l le nombre de lignes, comme il est plus judicieux de réaliser une interpolation par morceaux qu’une interpolation global, on découpe le précédent intervalle en l+1 élément, et ainsi l intervalles de type [xi,xi+1] où xi+1= xi + 1 .

Avec x0=0 et xl=l

De même avec les colonnes on obtiendra alors c intervalles de type [yi,yi+1] où yi+1= yi + 1 Avec y0=0 et yl=l

Comme on a une image 2D on considère deux polynômes de degré 3, l’un pour les ligne et l’autre pour les colonnes, en les multipliant on obtient f(x,y) : f(x,y)=px(x)\*py(y)=∑∑ai,j \* x^i \* y^j avec i,j=0 …. 3.

Avec ce polynôme on prend en considération 16 pixels ce qui nous donne 16 inconnu ai,j à déterminer. On considère donc un carré de 4x4 pixels voisins, ce qui nous donne les équations suivantes :

f(0 ,0) = I(0,0)= a00 f(1 ,0) = I(0,1)= a00+ a01+ a02+ a03

f(1 ,0) = I(0,1)= a00+ a10+ a20+ a30 f(1 ,0) = I(0,1)= ∑∑ai,j avec i,j=0 .… 3

Les conditions sur les dérivés:

Dérivée croisées :

(1) (0,0)=(0,0) (2) (0,1)=(0,1)

(3) (1,0)=(1,0) (4) (1,1)=(1,1)

Une image est découpée en plusieurs colonnes, chacune contient plusieurs lignes, ainsi si nous nous concentrons juste sur les lignes on aura un intervalle de [0,l] où l le nombre de lignes, comme il est plus judicieux de réaliser une interpolation par morceaux qu’une interpolation global, on découpe le précédent intervalle en l+1 élément, et ainsi l intervalles de type [xi,xi+1] où xi+1= xi + 1 .

Avec x0=0 et xl=l

De même avec les colonnes on obtiendra alors c intervalles de type [yi,yi+1] où yi+1= yi + 1 Avec y0=0 et yl=l

Comme on a une image 2D on considère deux polynômes de degré 3, l’un pour les ligne et l’autre pour les colonnes, en les multipliant on obtient f(x,y) : f(x,y)=px(x)\*py(y)=∑∑ai,j \* x^i \* y^j avec i,j=0 …. 3.

Avec ce polynôme on prend en considération 16 pixels ce qui nous donne 16 inconnu ai,j à déterminer. On considère donc un carré de 4x4 pixels voisins, ce qui nous donne les équations suivantes :

f(0 ,0) = I(0,0)= a00 f(1 ,0) = I(0,1)= a00+ a01+ a02+ a03

f(1 ,0) = I(0,1)= a00+ a10+ a20+ a30 f(1 ,0) = I(0,1)= ∑∑ai,j avec i,j=0 .… 3

Les conditions sur les dérivés:

Dérivée croisées :

(1) (0,0)=(0,0) (2) (0,1)=(0,1)

(3) (1,0)=(1,0) (4) (1,1)=(1,1)

