Rapport de Conception

Université Grenoble Alpes  
UFR IM²AG  
M1 Informatique  
2015-2016

PROJET IMAGE 2016

FAKHOUR Basma, DERUE Frédéric, POUVARET Line

**Table des matières**

[Introduction 2](#_Toc441685736)

[I- Fonctions implémentées 3](#_Toc441685737)

[II- Organisation de l’équipe 4](#_Toc441685738)

[III- Architecture de l’application 4](#_Toc441685739)

[IV- Options fonctionnelles et techniques 5](#_Toc441685743)

[IV.1. Mode transactionnel 5](#_Toc441685747)

[IV.2. Composition de deux images 6](#_Toc441685748)

[IV.3. Méthode de redéfinition de la taille d’une image 7](#_Toc441685749)

[IV.4. Méthodes de seam carving appliquées 8](#_Toc441685750)

[VI Principales limites du logiciel 8](#_Toc441685751)

[I- Interfaces graphiques de l’application : 10](#_Toc441685752)

[II- Annexes – scenarii de test 12](#_Toc441685753)

# Introduction

Le projet consiste à créer une application d’édition, de manipulation et de traitement d’image. Le but de cette application est de pouvoir ouvrir, enregistrer des fichiers d’image dans les formats les plus courants (extensions : [gif](https://fr.wikipedia.org/wiki/Graphics_Interchange_Format), [jpg](https://fr.wikipedia.org/wiki/Joint_Photographic_Experts_Group), [png](https://fr.wikipedia.org/wiki/Portable_Network_Graphics), etc.). L’application permet également à l’utilisateur de modifier l’image en traitant le contenu de celle-ci.

Ce projet a été réalisé du 4 janvier 2016 au 28 janvier 2016 et en trinôme.

## Fonctions implémentées

* **Fonctions de base** :

1. Lecture et écriture d’une image dans les formats standards (gif, jpg, pnm, png notamment). A noter que l’écriture ne fonctionne pas dans le cas de gif.

2. Réalisation d’une interface permettant l’affichage d’une image et de menus pour les fonctionnalités (lecture, écriture, etc.).

3. Découpage : définir la zone utile de l’image, pouvoir déplacer cette zone et éliminer le reste.

4. Affichage de la couleur d’un pixel avec la possibilité de choisir l’espace RGB ou YUV.

5. Affichage des histogrammes de couleurs (RGB ou YUV).

6. Transformation d’une image couleur en niveaux de gris.

7. Inversion des couleurs d’une image.

8. Flou : rendre l’image floue grâce à un filtre gaussien ou moyenneur dont la taille peut être définie par l’utilisateur.

9. Fusion : mélanger deux images en modifiant la transparence de chacune d’entre elle.

* **Fonctions intermédiaires** :

1. Redimensionnement d’une image : possibilité d’agrandissement/rétrécissement de l’image sur la hauteur et/ou largeur.

2. Modifications des histogrammes : fonction d’étalement et d’égalisation d’histogramme implémentées.

3. Filtrage : fonction de détection de contour (calcul des gradients avec les filtres Prewitt, Sobel ou Scharr) et possibilité de faire une convolution personnalisée avec un filtre à définir.

4. Amélioration : accentuation des contours dans une image (appelée Réhaussement dans notre programme), effectuée grâce à un filtre passe haut.

* **Fonctions avancées** :

1. Redimensionnement intelligent : compression/étirement (verticalement) de l’image tout en préservant le contenu de cette dernière (SeamCarving).

## Organisation de l’équipe

Enfin, la responsabilité de l’analyse, codage, réalisation et debug était répartie entre les trois membres du groupe.

## Architecture de l’application

En vue de réaliser le projet, nous avons eu le choix entre plusieurs technologies de développement et parmi cette multitude de choix nous avons opté pour les technologies suivantes :

* Langage C++ :

Nous avons utilisé le langage C++ pour développer notre application et ce pour la performance de transformation de données (calcul matriciel) et notamment pour le rendu du seam carving.

* Bibliothèque Qt :

Nous avons utilisé la bibliothèque logicielle multiplateforme Qt afin de réaliser l’interface graphique du programme. Elle nous a paru plus intuitive à manipuler et bien sûr plus pratique du fait de sa portabilité. Sa documentation très complète nous a permis de nous familiariser rapidement avec les différents aspects de conception de notre interface graphique.

* QT Creator :

QT Creator est un [environnement de développement intégré](https://fr.wikipedia.org/wiki/Environnement_de_d%C3%A9veloppement_int%C3%A9gr%C3%A9) [multiplateforme](https://fr.wikipedia.org/wiki/Multiplate-forme) faisant partie du [framework](https://fr.wikipedia.org/wiki/Framework) [Qt](https://fr.wikipedia.org/wiki/Qt). Orienté pour la programmation en C++, il permet la composition des interfaces graphiques tout en prenant en compte les évènements et actions créées.

* QWT :

QWT est l'abréviation de Qt Widgets for Technical Applications (Widgets Qt pour les applications techniques). C'est une collection de widgets et de classes de soutien pour créer des applications techniques.

Nous l’utilisons dans notre application afin d’afficher les histogrammes YUV et RGB des images.



## Options fonctionnelles et techniques

L’objet de cette section est de détailler les principales options qui ont été prises au cours du développement de l’application, en précisant les implications fonctionnelles et techniques ainsi que les éventuelles limites dont elles ont fait l’objet.



### Mode transactionnel

Pour certaines actions (copier-couper, fusion de 2 images) supposant des ajustements de l’utilisateur, nous utilisons une logique de transaction où l’utilisateur a la possibilité de valider  ou supprimer  l’action en cours. Pour d’autres actions (les traitements d’image par convolution par exemple), l’action est directement validée.

Il est possible de revenir en arrière de toutes les actions validées grâce à un bouton annuler . Cette possibilité de revenir à une image antérieure se fait grâce à un objet QList. Nous n’avons pas mis en place un système de sauvegarde fichier des images ainsi collectées dans l’optique de libérer des ressources mémoire.

### Composition de deux images

Pour satisfaire deux des fonctionnalités demandées : le coller et la fusion de deux images, nous avons été confrontés à la problématique de la composition de deux images, c’est-à-dire laisser à l’utilisateur la possibilité de les placer l'une par rapport à l'autre dans l'optique de produire à un certain moment une troisième image de synthèse.

Cette possibilité de composition de deux images s’est vite avérée déterminante dans le choix de notre interface graphique, notamment en terme de contraintes posées au widget Qt d’affichage des images.

Nous avons opté pour l’utilisation de QPainter et sa fonction drawPixmap pour afficher une QImage à une position donnée. Une contrainte technique QPainter interdit de composer des images secondaires en rajout au-delà des limites de l’image principale qui sert de support au widget Qt. Pour la composition de deux images, nous nous sommes donc donné trois objets QImage :

* Une image principale s’apparentant à un fond de couleur homogène (couleur blanche choisie par défaut) qui délimite la taille maximale de la scène à composer,
* Une première image avec un pointeur de position (via un QPoint),
* Une seconde image avec également un pointeur de position (via un QPoint).

Dans le cadre d’un copier – couper / coller, la taille maximale de la scène est l’image de départ.

Pour une fusion de deux images, la taille de l’image principale est déterminée comme la somme des hauteurs et des largeurs des deux images manipulées pour permettre toutes les compositions possibles avec pour cas extrême la contiguïté des deux images.

Pour faciliter l’ergonomie et la manipulation, l’utilisateur peut sélectionner l’une ou l’autre des images (à l’exception de l’image principale non déplaçable) par des clicks souris. Pour déplacer l’image sélectionnée, une fonctionnalité de 'drag and drop' a été implémentée modifiant notamment les pointeurs de position relatifs des deux images.

Les parties superposées des deux images affichent la seconde image en rendu final et il n’est pas possible de modifier cet ordre. Toutefois un indice de transparence a été rajouté à chacune des deux images pour permettre un affichage des couleurs avec des coefficients alpha et ‘fusionner’ les couleurs superposées des deux images en jouant sur l’opacité.

### Méthode de redéfinition de la taille d’une image

Un algorithme de redimensionnement d’une image de départ a été implémenté selon une approche itérative s’efforçant de corriger les dysfonctionnements constatés sur des cas particuliers lors de l’itération précédente. Les principales fonctionnalités mises en œuvre sont les suivantes :

* Lorsque la largeur et la hauteur de l’image sont toutes deux à redimensionner, l’opération est scindée en deux et une image intermédiaire est produite avec la nouvelle largeur souhaitée et la hauteur d’origine. Cette façon de faire est moins performante en temps de traitement et en ressource machine mais elle évite des interpolations bilinéaires et limite les calculs d’interpolation à des interpolations linéaires à une dimension.
* Une interpolation linéaire a été effectuée lorsqu’une ou plusieurs lignes devaient être insérées entre 2 lignes de l’image de départ. Pour connaître le nombre de lignes à insérer, une division de la taille cible de l’image par sa taille originelle a été faite dans le sens de la largeur et de la hauteur. Le quotient, **Q**, indique le nombre de lignes d’interpolation à effectuer. Le reste de cette division, **R**, indique un nombre de lignes à insérer au cours d’un parcours de l’image d’origine. La suite des fonctionnalités décrites précise une manière optimisée les **R** lignes restantes.
* Recherche du plus petit pas en puissance de deux (de 2 à 32) qui divise la taille initiale de l’image avec un quotient inférieur à **R**. Lors du parcours de l’image source, les lignes dont la position est divisible par ce pas de 2 seront injectées dans l’image cible. Sans cette échantillonnage, l’image perdait sa structure (son équilibre) et la partie la plus à gauche était surreprésentée.
* A l’issue de l’échantillonnage avec la puissance de 2, on considère les **R’** lignes qu’il reste à injecter. Si **R’** est petit on détermine un pas de répartition des lignes que l’on ajoute. Dans le cas contraire, on reporte toutes les lignes sources et on détermine un pas de répartition des lignes que l’on n’insère pas. L’idée est que le pas calculé soit le plus grand possible pour avoir la meilleure distribution sur l’image et éviter que l’image produite ne ‘penche’ trop d’un côté.

### Méthode de seam carving appliquée

Nous avons cherché à mettre en place une logique de traitement du seam carving qui convienne aussi bien en compression qu’en extension. Deux stratégies nous ont semblé possibles :

1. Une méthode 1 qui consiste à itérer sur la recherche de la ligne d’énergie la plus faible, à la rajouter par interpolation ou la supprimer (en cas de suppression) puis à réitérer à partir de l’image obtenue jusqu’à atteindre la taille souhaitée.
2. Une méthode 2 qui consiste à itérer la recherche de la ligne d’énergie la plus faible dans un contexte où les points des précédentes lignes trouvées ont été interdits par un système de BitMap sur l’ensemble des pixels de l’image.

La méthode 1 présentait le risque en extension d’injecter l’ensemble des nouvelles lignes au niveau de la même zone. Nous avons implémenté la seconde méthode. Nous ne l’avons développé que dans le sens de la hauteur.

A chaque itération on parcourt l’ensemble des points les plus à gauche de l’image puis on progresse vers la droite, en sélectionnant le point suivant qui présente l’énergie la plus faible (somme des valeurs absolues des gradients dx et dy). Trois points sont alors possibles : (x+1,y), (x+1,y-1) et (x+1,y+1). En priorité est affecté la position suivante à la même hauteur. On tient compte de la bitmap des positions interdites (les bords du haut et du bas + les positions des lignes précédentes trouvées) dans le choix de la position. Si les points correspondent à des points interdits, on arrête la détermination de la ligne.

Nous déterminons ainsi à l’avance (lors du click sur l’action seam carving) l’ensemble des lignes affectables. L’utilisateur peut alors choisir un nombre de points et un mode d’application : compression ou extension. L’image cible est alors directement présentée. Une option permet également de visualiser les lignes calculées sur l’image d’origine.

Notre implémentation nous donne en résultat qu’un nombre restreint de lignes utilisables : de l’ordre de 10% de la taille de l’image.

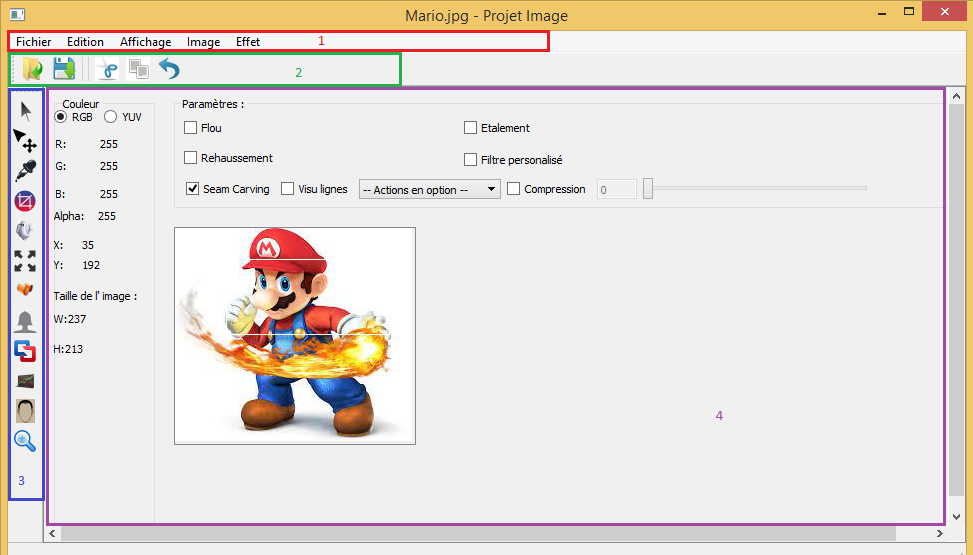
## V Principales limites du logiciel

* L’application ne permet que le traitement de deux images à la fois (au maximum).
* Le recadrage nécessite la saisie les tailles souhaitées, ne se fait pas par redimensionnement dynamique de taille de fenêtre à l’aide de la souris.
* Le seam carving n’est implémenté que dans le sens vertical.
* Chaque composante d’un histogramme s’affiche sur une fenêtre à part (l’idée aurait été de les regrouper dans une même fenêtre)
* L’application ne garde pas l’historique précis des tâches effectuées sur l’image (possibilité de revenir à un endroit en particulier) mais offre la possibilité de revenir en arrière pas à pas.
* L’application ne fournit pas la fonctionnalité de GRAB CUT. En effet, la librairie OpenCV étant compliquée à installer sur chaque poste de travail, nous avons préféré nous en passer.
* La fusion ne gère pas la transparence des images à la validation et par conséquent à l’enregistrement de l’image.
* Stockage de l’historique des modifications sur une image en mémoire : ce choix entraîne des soucis d’allocation dans le cas de très grosses images ainsi que lorsque l’on garde un historique trop grand. La solution appropriée aurait été de ne garder en mémoire que 2 à 3 modifications et stocker les plus anciennes dans un fichier. Ainsi les modifications stockées en mémoire auraient servi de buffer entre le fichier et l’application.
* Taille du noyau gaussien limité dans le cas d’une convolution à cause des limites physiques des types utilisés (double dans notre cas). Taille maximale observée pour un noyau gaussien : 15.

## Interfaces graphiques de l’application :

1. Fenêtre principale

La figure ci-dessous représente la fenêtre principale de notre application, l’utilisateur a accès à toute les fonctionnalités du logiciel depuis cette dernière

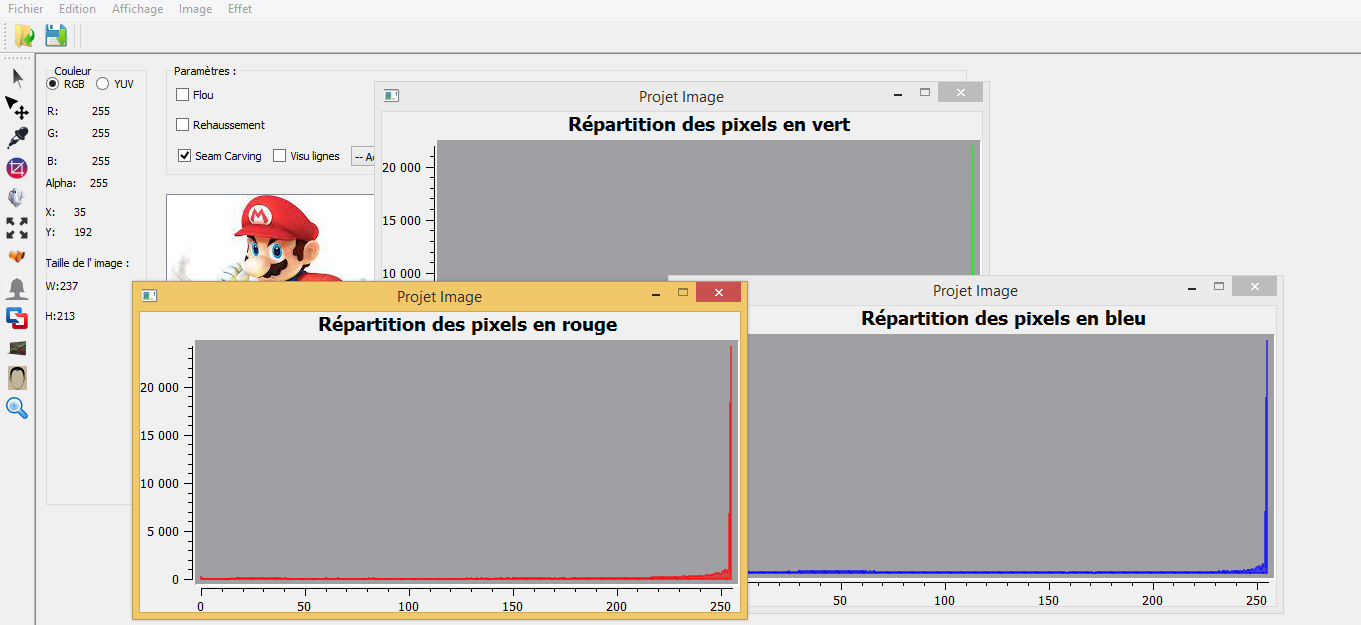


La fenêtre principale est constituée de plusieurs parties

* La barre de menus (1)
* Le raccourci de la barre de tâches (2)
* La barre d’outil (3)
* La fenêtre d’affichage des images (4)

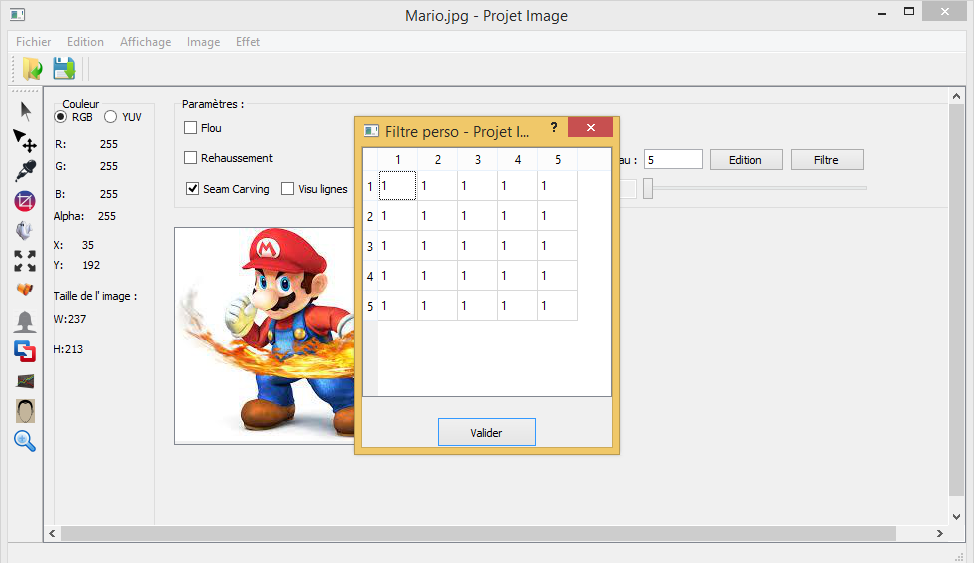
1. Fenêtre des histogrammes

Les histogrammes des images sont affichés sur trois fenêtre répartition des pixels en rouge, vert et bleu ou bien respectivement composante Y, U et V cela dépend du choix qu’on coche, comme le montre la fenêtre ci-dessous :



1. Fenêtre du masque personnalisé

Le bouton Edition qui se trouve dans les paramètres permet à l’utilisateur la modification des valeurs du masque pour une convolution de l’image, l’utilisateur définit la taille du masque ou bien le masque prend la valeur par défaut soit : 3



## Annexes – scenarii de test

Ci-dessous on se permet d’étaler les séquences de test que nous avons effectué pour s’assurer que les fonctionnalités de l’application sont opérationnelles.

1. **Scénario 1** : Test de l’ouverture d’une image

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Description du test** | **Etape** | **Type d’image supporté** | **Le résultat attendu** | **Bouton** | **Boutons visibles** |
| Dans ce scénario on va tester est ce que l’ouverture des images est opérationnelle | 1-Pour ce faire on clique sur le bouton d’ouverture sur la barre de menu | JPG, GIF, PNG | - L’image est ouverte dans l’écran principale  - Les boutons de traitement d’image sont visibles |  |  |

1. **Scénario 2** : Test de la sélection d’image

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Description du test** | **Etape** | **Le résultat attendu** | **Bouton** | **Boutons visibles** |
| On teste dans ce scénario si le cadrage de l’image fonctionne bien | 1- On doit sélectionner l’image à l’aide de la souris de sélection | - Un triangle doit apparaitre lors de la sélection de l’image  - Les boutons copier et couper doivent apparaitre sur la barre du menu |  |  |

1. **Scénario 3** : Test du déplacement de l’image

|  |  |  |  |
| --- | --- | --- | --- |
| **Description du test** | **Etapes** | **Le résultat attendu** | **Bouton** |
| Le bouton de déplacement doit permettre à l’utilisateur de déplacer l’intégralité de l’image | 1-On clique sur le bouton de déplacement et on déplace l’image sélectionnée | - Le bouton déplacer permet de déplacer toute l’image |  |

1. **Scénario 4** : Test de la pipette de couleur

|  |  |  |  |
| --- | --- | --- | --- |
| **Description du test** | **Etapes** | **Le résultat attendu** | **Bouton** |
| La pipette de couleur permet de montrer les valeurs RGB et YUV de chaque pixel sélectionné sur l’image | 1-Quand on veut connaitre les couleurs de l’image on clique dessus avec la pipette.  2-L’utilisateur a le choix entre deux radios boutons soit RGB ou YUV | -Choix RGB :  Les labels R, G, B, alpha, X et Y prennent des valeurs appropriées.  -Choix YUV :  Les labels Y, U, V, alpha, X et Y prennent des valeurs appropriées |  |

1. **Scénario 5** : Test du bouton de découpage (CROP)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Description du test** | **Etapes** | **Le résultat attendu** | **Bouton** | **Boutons visibles** |
| Le CROP permet de sélectionner l’image et de soit valider ou supprimer la partie utile que l’on veut ainsi qu’effacer le reste de l’image | 1-Cliquer sur le bouton CROP et sélectionner l’image qu’on veut découper.  2-Cliquer sur valider ou supprimer ce qu’on a sélectionné | -La partie sélectionnée et validée prend la place de l’image originale  -La partie sélectionnée et ensuite supprimée ne fait aucun changement sur l’image  -Les boutons valider et supprimer sont visibles après avoir cliqué sur le bouton crop et invisibles après qu’on effectuer nos changements |  |  |

1. **Scénario 6** : Test du bouton du flou

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Description du test** | **Etapes** | **Le résultat attendu** | **Bouton** | **Boutons visibles** |
| Le bouton Flou rend l’image flou par moyennage des valeurs des pixels | 1-Cliquer sur le bouton Flou. | -L’image est floutée à chaque fois qu’on clique sur le bouton  -Le bouton annuler est visible |  |  |

1. **Scénario 7** : Test du bouton inverser couleur

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Description du test** | **Etapes** | **Le résultat attendu** | **Bouton** | **Boutons visibles** |
| Le bouton inverse les couleurs d’une image | 1-Cliquer sur le bouton inverser couleur | -Les couleurs de l’image sont inversées  -Le bouton annuler est visible |  |  |

1. **Scénario 8** : Test du bouton image en niveaux de gris

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Description du test** | **Etapes** | **Le résultat attendu** | **Bouton** | **Boutons visibles** |
| Le bouton transforme les couleurs d’une image en niveaux de gris | 1-Cliquer sur le bouton image en niveaux de gris | -L’image est transformée en gris  -Le bouton annuler est visible |  |  |

1. **Scénario 9** : Test du bouton histogramme

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Description du test** | **Etapes** | **Le résultat attendu** | **Bouton** | **Boutons visibles** |
| Le bouton affiche l’histogramme de l’image en se basant soit sur RGB ou YUV | 1-Cliquer sur le bouton histogramme  2- L’utilisateur a le choix entre deux radios boutons soit RGB ou YUV | - Choix RGB  Les trois fenêtres R, G et B sont affichées.  - Choix YUV  Les trois fenêtres Y, U et V sont affichées. |  | NA |

1. **Scénario 10** : Test du bouton détection de contour

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Description du test** | **Etapes** | **Le résultat attendu** | **Bouton** | **Boutons visibles** |
| Le bouton affiche Les contours d’une image | 1-Cliquer sur le bouton Contour | - les contours sont affichés à chaque fois qu’on clique sur le bouton  - Le bouton undo est visible |  |  |

1. **Scénario 11** : Test du bouton Seam carving

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Description du test** | **Etapes** | **Le résultat attendu** | **Bouton** | **Boutons visibles** |
| Le bouton permet de redimensionner l’image tout en préservant le contenu de cette dernière | 1-Cliquer sur le bouton Seam carving  2-Choisir une option dans le combo box  3-Glisser le slider pour voir le redimensionnement | - Le bouton active un checkbox seam carving dans le groupBox paramètres  - Le checkbox Visu ligne permet de visualiser les chemins sur l’image  -Le combobox action en option donne trois valeurs :   * Def zones à protéger * Def zones à supprimer * Recalcule ligne d’énergie   - Un check box compression qui sert a  -le slider pour redimensionner l’image |  | NA |

1. **Scénario 12** : Test du bouton fusion d’image

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Description du test** | **Etape** | **Le résultat attendu** | **Bouton** | **Boutons visibles** |
| Le bouton permet de Fusionner deux images, de pouvoir manipuler l’opacité de chacune de ses images et enfin d’enregistrer ses modifications pour en garder une dernière image | 1-Cliquer sur le bouton Fusion  2-Changer l’opacité des deux images  3-Valider pour enregistrer la fusion  4-Supprimer pour annuler les modifications effectuées sur l’image | - L’utilisateur choisit une image qu’il veut fusionner  - Deux slider sont visibles pour pouvoir modifier l’opacité des deux images  -Les deux boutons valider et supprimer sont visibles |  |  |

1. **Scénario 13** : Test du bouton Annuler

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Description du test** | **Etapes** | **Le résultat attendu** | **Bouton** | **Boutons visibles** |
| Le bouton permet d’annuler les modifications effectuées sur l’image d’origine | 1-Cliquer sur le bouton Annuler | - Retour à l’étape précédente |  | NA |

1. **Scénario 14** : Test du bouton Enregistrer sous

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Description du test** | **Etapes** | **Type d’image supporté** | **Le résultat attendu** | **Bouton** | **Boutons visibles** |
| Dans ce scénario on va tester est ce que l’enregistrement des images est opérationnelle | 1-Pour ce faire on clique sur le bouton enregistrer sous qui se trouve dans le menu fichier | JPG, JPEG, PNG | -L’image est enregistrée |  | NA |

1. **Scénario 15** : Test du menu Flou

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Description du test** | **Etapes** | **Le résultat attendu** | **Menu Flou** | **Boutons visibles** | **Paramètre Etalement** |
| Le menu Flou rend l’image flou par moyennage des valeurs des pixels et cette fois selon deux modes de flou soit gaussien ou moyenneur | 1-Le menu flou donne le choix entre deux flous gaussien ou moyenneur  2- On modifie la taille du noyau dans les paramètres du Flou | -La matrice chargé par défaut du flou gaussien et moyenneur est 3  -L’image est floutée |  |  |  |

1. **Scénario 16** : Test du menu Détection de contour

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Description du test** | **Etapes** | **Le résultat attendu** | **Menu Détection** | **Boutons visibles** | **Paramètre Etalement** |
| Le menu détection de contour accentue Les contours d’une image | 1-Le menu Détection de contour donne trois choix Prewitt, Sobel et Scharr | - Les contours sont affichés selon le type de filtre qu’on a choisi |  |  | NA |

1. **Scénario 16** : Test du menu Etalement

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Description du test** | **Etapes** | **Le résultat attendu** | **Menu Flou** | **Boutons visibles** | **Paramètre Etalement** |
| Le menu étalement est une fonction qui fait un étalement de l’histogramme | 1-Le menu étalement réalise l’étalement primitif de l’histogramme  2-Le paramètre étalement permet de définir alpha et beta du filtre qu’on veut utiliser | -L’histogramme est étalé par défaut avec les valeurs alpha :1.0 et beta :1.5 |  |  |  |

1. **Scénario 17** : Test du menu Egalisation

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Description du test** | **Etapes** | **Le résultat attendu** | **Menu Flou** | **Boutons visibles** | **Paramètre Egalisation** |
| Le menu égalisation est une fonction qui fait une égalisation de l’histogramme | 1-Le menu égalisation réalise l’étalement primitif de l’histogramme  2-Le paramètre étalement permet de définir alpha et beta du filtre qu’on veut utiliser | - L’histogramme est égalisé |  |  | NA |

1. **Scénario 18** : Test du menu Rehaussement

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Description du test** | **Etapes** | **Le résultat attendu** | **Menu Flou** | **Boutons visibles** | **Paramètre Egalisation** |
| Le menu rehaussement permet de faire un rehaussement de contraste de l’image | 1-Le menu rehaussement réalise le rehaussement primitif de l’histogramme  2-Le paramètre rehaussement permet de définir le coefficient de rehaussement | -Le contraste de l’image est rehaussé |  |  |  |

1. **Scénario 19** : Test du paramètre filtre personnaliser

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Description du test** | **Etapes** | **Le résultat attendu** | **Noyau** | **Boutons visibles** | **Paramètre Egalisation** |
| Le paramètre filtre personnaliser va permettre à l’utilisateur de saisir un masque | 1-Cocher le check box filtre personnaliser  2-Saisir une  taille du noyau  3-Cliquer sur édition pour paramétrer le noyau  4-une fenêtre de filtre s’ouvrira pour choisir les valeurs qu’on veut  5-Enfin cliquer sur filtre pour visualiser le résultat | -Le filtre permet la convolution de l’image |  |  |  |

1. **Scénario 18** : Test de la modification de la taille de l’image

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Description du test** | **Etape** | **Le résultat attendu** | **Menu** | **Boutons visibles** |
| Le test consiste à changer la taille de l’image | 1-Cliquer sur le menu recadrer  2-LineEdit largeur et hauteur apparaisse  3-Saisir la valeur du width  4-Le bouton modifier apparait  2-Saisir la valeur du height  3-Cliquer sur le bouton modifier | -L’image est soit agrandis ou rétrécit |  |  |