



École Polytechnique de l'Université de Tours  
64, Avenue Jean Portalis  
37200 TOURS, FRANCE  
Tél. +33 (0)2 47 36 14 14  
[www.polytech.univ-tours.fr](http://www.polytech.univ-tours.fr)

## Département Informatique

### Cahier de spécification système & plan de développement

<b>Projet :</b>	<b>Développement d'un outil de traitement d'images par filtrage bilatéral</b>	
<b>Emetteur :</b>	Natacha MARLIO-MARETTE	Coordonnées : natacha.marlio-marette@etu.univ-tours.fr
<b>Date d'émission :</b>	8 janvier 2015	

### Validation

Nom	Date	Valide (O/N)	Commentaires
-----	------	--------------	--------------

**M. Hidane** : 08/01/2015 ; O ; Ok pour 2014-2015

### Historique des modifications

Version	Date	Description de la modification
---------	------	--------------------------------

**00** : 06/11/2014 ; Rédaction du cahier de spécification

**01** : 18/12/2014 ; Rédaction du plan de développement



# Table des matières

---

<b>Cahier de spécification système</b>	<b>6</b>
1.1 Introduction . . . . .	6
1.2 Contexte de la réalisation . . . . .	6
1.2.1 Contexte . . . . .	6
1.2.2 Objectifs . . . . .	6
1.2.3 Bases méthodologiques (selon les projets) . . . . .	6
1.3 Description générale . . . . .	6
1.3.1 Environnement du projet . . . . .	6
1.3.2 Caractéristiques des utilisateurs . . . . .	7
1.3.3 Fonctionnalités et structure générale du système . . . . .	7
1.3.4 Contraintes de développement . . . . .	7
1.4 Description des interfaces externes du logiciel . . . . .	8
1.4.1 Interfaces matériel/logiciel . . . . .	8
1.4.2 Interfaces homme/machine . . . . .	8
1.4.3 Interfaces logiciel/logiciel . . . . .	9
1.5 Architecture générale du système . . . . .	9
1.6 Description des fonctionnalités . . . . .	10
1.6.1 Définition de la fonction <code>attenuation()</code> . . . . .	10
1.6.2 Définition de la fonction <code>raising()</code> . . . . .	11
1.7 Conditions de fonctionnement . . . . .	11
1.7.1 Performances . . . . .	11
1.7.2 Capacités . . . . .	11
1.7.3 Contrôlabilité . . . . .	12
 <b>Plan de développement</b>	 <b>14</b>
2.1 Découpage du projet en tâches . . . . .	14
2.1.1 Implémentation du filtre bilatéral . . . . .	14
2.1.2 Utilisation du filtre bilatéral en débruitage . . . . .	14
2.1.3 Utilisation du filtre bilatéral en décomposition . . . . .	15
2.1.4 Pyramide multi-échelle à partir du filtre bilatéral . . . . .	15
2.1.5 Manipulation des détails . . . . .	15
2.1.6 Interface graphique . . . . .	16
2.1.7 Accélération . . . . .	16
2.1.8 Modification de l'algorithme de décomposition . . . . .	17
2.1.9 Travail sur les modèles 3D . . . . .	17
2.1.10 Gestion de projet . . . . .	18
2.2 Planning . . . . .	18
 <b>A Références</b>	 <b>20</b>
 <b>Glossaire</b>	 <b>21</b>

<b>Index</b>	<b>23</b>
<b>Bibliographie</b>	<b>23</b>

# Cahier de spécification système



## 1.1 Introduction

Ce cahier de spécification système décrit le contexte dans le quel est réalisé ce projet de fin d'étude suivit d'une description générale ainsi qu'une description de l'interface homme-machine et des fonctionnalités. A la suite, le plan de développement comprendra les différentes tâches du projet ainsi qu'un planning.

Ce sujet a été proposé par un chercheur, Moncef HIDANE, en analyse et reconnaissance des formes. Le maître d'ouvrage est le laboratoire RFAI(Reconnaissance des Formes et Analyse d'Images) de l'université François Rabelais de Tours. La maitrise d'œuvre sera réalisé par moi-même, Natacha MARLIO-MARETTE, étudiante en cinquième année au sein du département informatique de l'école Polytech Tours.

## 1.2 Contexte de la réalisation

### 1.2.1 Contexte

Ce projet répond à la demande d'un chercheur de l'INSA de Blois en analyse et reconnaissance des formes en collaboration avec le laboratoire RFAI de l'université de François Rabelais de Tours.

Ce projet de fin d'étude porte sur la mise en place d'un filtre bilatéral . Ce filtre est utilisé afin de lisser les petites variations d'intensité et préservé les contours des détails, le résultat obtenu sera appelé la couche de base. L'image finale obtenue après application du filtre est la résultante de la différence entre l'image originale et la couche de base.

### 1.2.2 Objectifs

On peut décomposer le projet en trois grandes parties. Tout d'abord, il faudra mettre en place une implémentation du filtre bilatéral en utilisant la méthode naïve. Avec cette implémentation, on pourra tester le filtre sur des images bruitées puis effectuer une décomposition pyramidale multi-échelle à partir du filtre et manipuler les détails selon deux niveaux : l'atténuation ou le réhaussement des détails.

Dans un second temps, une interface graphique sera mise en place permettant de manipuler les détails d'une image selon les deux niveaux vu précédemment. Par la suite, on cherchera à accélérer le traitement réaliser par le filtre bilatéral.

Pour finir, la décomposition multi-échelle sera ré-implémenté selon une nouvelle méthode et l'image devra être manipulé à partir d'un graphe dont la structure devra être construite. Si le temps le permet, un travail sur les modèles 3D pourra être réaliser.

### 1.2.3 Bases méthodologiques (selon les projets)

La manipulation des images devra se faire à l'aide de la librairie CImg [1] et l'interface sera développée avec le framework Qt [2]. Le projet devra être codé en C++.

## 1.3 Description générale

### 1.3.1 Environnement du projet

Ce projet ne s'inscrit dans aucun existant. Le prototype logiciel ne nécessite aucune machine particulière n'ayant pas de contraintes logicielles ni matérielles pour l'utilisateur.

### 1.3.2 Caractéristiques des utilisateurs

Le système ne rencontrera qu'un unique type d'utilisateur celui du prototype logiciel. L'utilisateur du logiciel n'aura pas besoin d'avoir des connaissances informatiques afin d'utiliser celui-ci. L'interface sera la même pour tous les utilisateurs du logiciel, ils pourront donc tous effectuer les mêmes manipulations avec celui-ci.

### 1.3.3 Fonctionnalités et structure générale du système

L'utilisateur pourra effectuer diverses actions avec le logiciel. Ces actions seront :

- Charger une image
- Choisir le niveaux de détails du filtre
- Introduire des informations (paramètres) manuellement
- Sauvegarder l'image généré

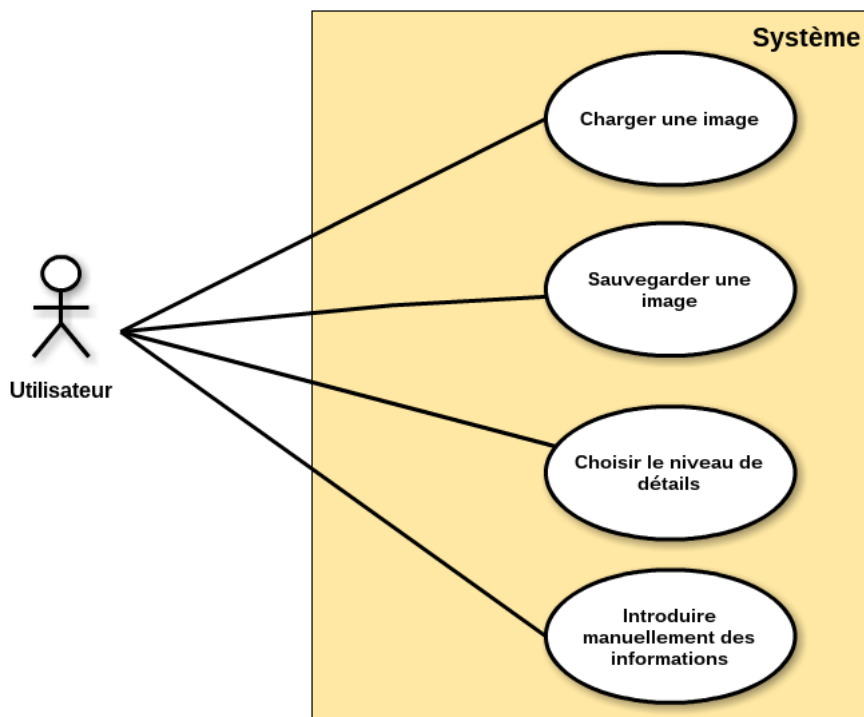


FIGURE 1.1 – Diagramme des cas d'utilisation du logiciel

### 1.3.4 Contraintes de développement

#### Contraintes de développement

Le développement logiciel n'est contraint par aucune particularité lié au matériel. L'utilisation d'une librairie, CImg, est requise afin de pouvoir manipuler les images et réaliser le filtre. L'interface graphique devra être développée à l'aide du framework Qt. Le choix du langage de programmation c'est naturellement porté vers le C++, la librairie CImg étant dans ce langage et le framework Qt utilisable seulement dans ce langage également.

## 1.4 Description des interfaces externes du logiciel

### 1.4.1 Interfaces matériel/logiciel

Il n'y a pas d'interface matériel à considérer.

### 1.4.2 Interfaces homme/machine

L'interface homme/machine comporte deux écrans (voir image 1.2 et 1.3). Le premier écran permet de charger ou de sauvegarder une image via un panneau de navigation dans les dossiers de l'utilisateur. Cet écran est accessible par le bouton "navigation" dans les sections "Charger" et "Sauvegarder".

Une fois que l'utilisateur a choisi l'image qu'il veut manipuler, il peut cliquer sur le bouton "Charger", ce qui affichera l'image dans le cadre de gauche.

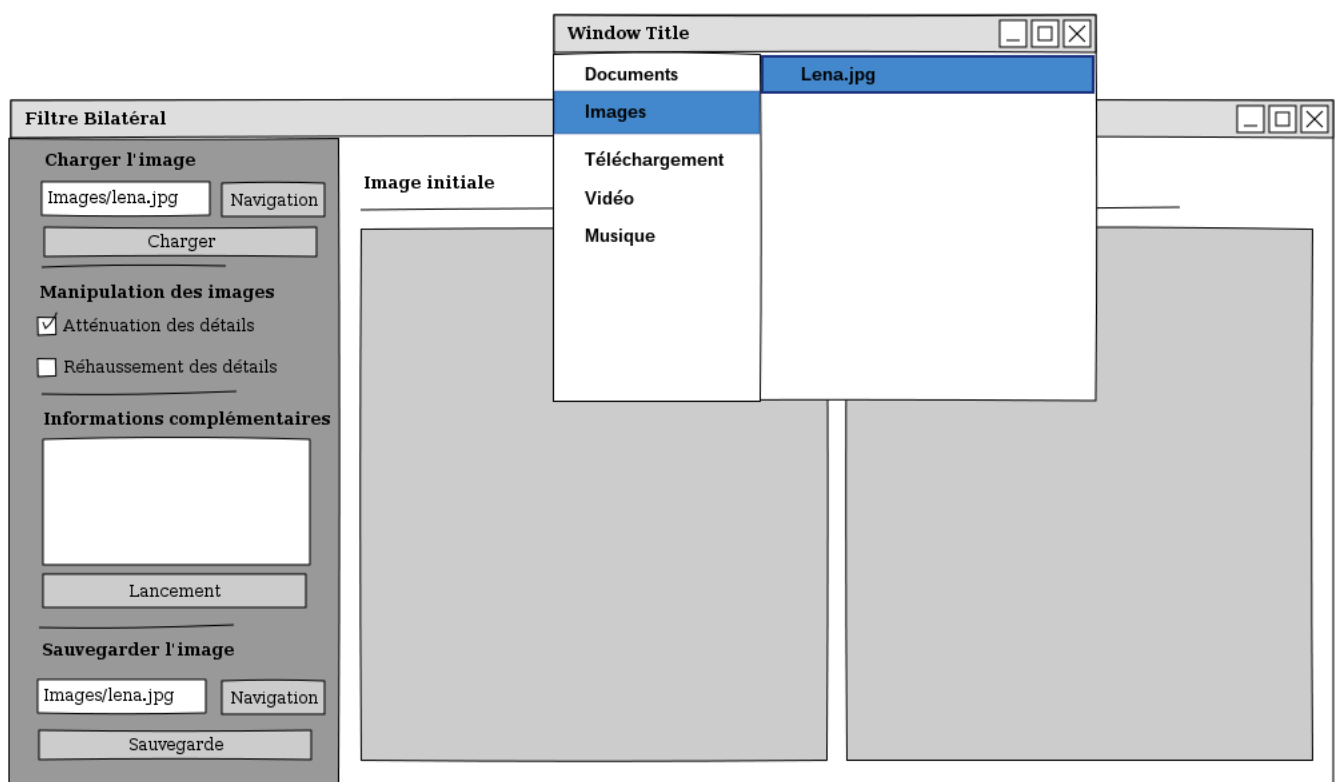


FIGURE 1.2 – Interface pour choisir l'image initiale

L'utilisateur peut choisir la façon dont il souhaite manipuler l'image. Deux choix lui sont offerts :

- Atténuation des détails
- Rehaussement des détails

Il a la possibilité de rentrer manuellement des informations complémentaires dans la zone de texte prévu à cet effet. Afin de lancer l'exécution du filtre sur l'image, il doit appuyer sur le bouton "lancement".

Une fois que le filtre a fini de s'exécuter sur l'image de base, celle-ci s'affiche dans le cadre de gauche et l'image résultante de la manipulation dans celui de droite. L'utilisateur pourra sauvegarder l'image résultante où il le souhaite via le bouton "navigation" puis avec le bouton "Sauvegarder".



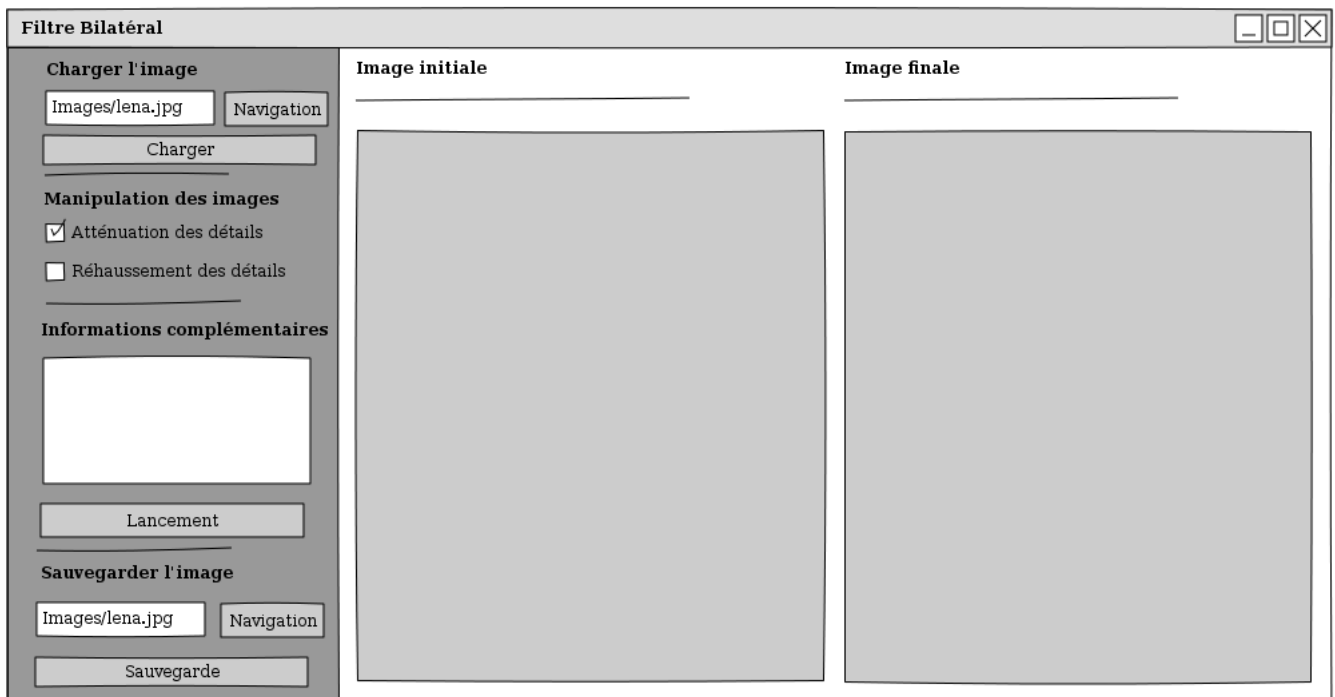


FIGURE 1.3 – Interface après le lancement du filtre

### 1.4.3 Interfaces logiciel/logiciel

Il n'y a pas d'interfaces logiciel/logiciel à considérer.

## 1.5 Architecture générale du système

Afin de modéliser notre problème qui est l'organisation de l'architecture de l'application avec une interface utilisateur, un design pattern Modèle-Vue-Contrôleur sera mis en place. Ce patron de conception va permettre de séparer les données (le modèle), l'interface utilisateur (la vue) et la logique de contrôle (le contrôleur). Le système sera donc composé de trois principaux éléments (voir le diagramme de classe 1.4).

Le composant View contiendra tous les attributs avec lequel l'utilisateur pourra interagir (boutons, zone de texte). Il permettra seulement d'afficher l'interface utilisateur et n'effectuera aucun traitement. Les actions des utilisateurs seront transmises au composant Contrôleur qui réalisera l'action correspondante. Celui-ci aura pour rôle de gérer les messages transmis par la vue ainsi que la synchronisation entre la vue et le modèle (ImgModel). Le modèle est contenu dans le composant ImgModel. Il contiendra l'image et les méthodes nécessaires à sa manipulation.

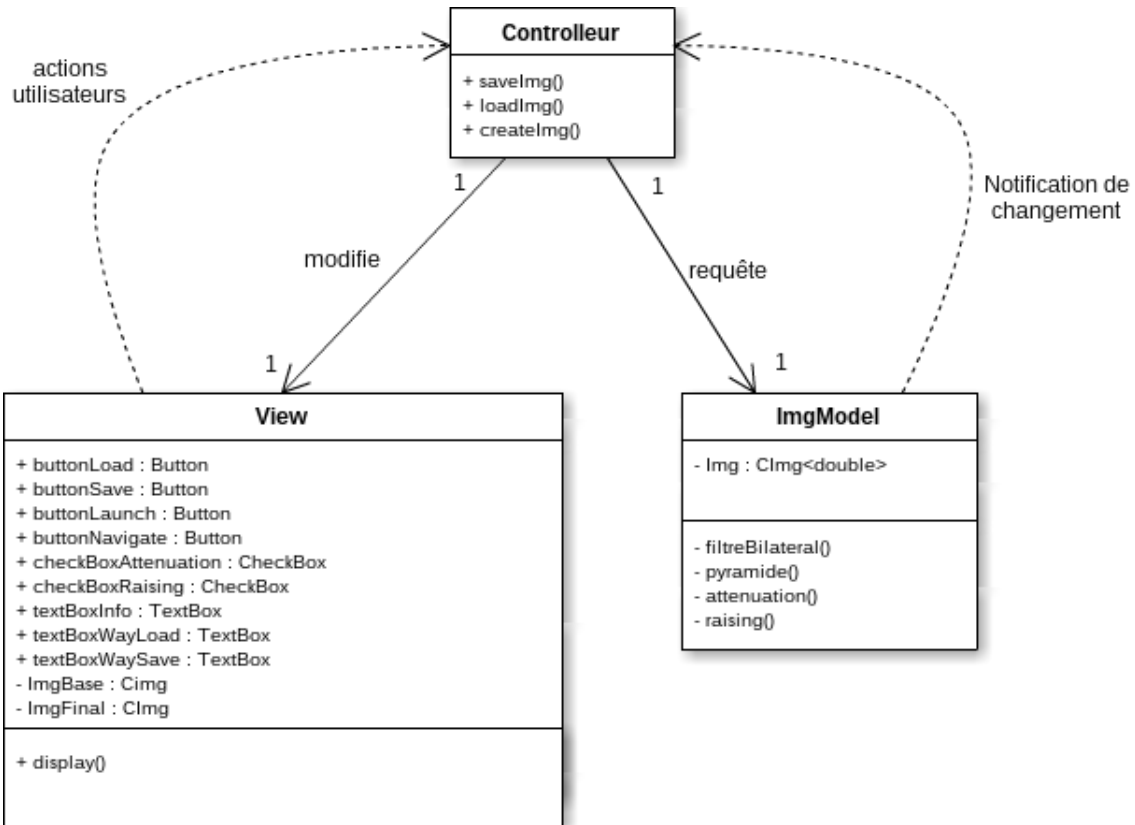


FIGURE 1.4 – Diagramme de classe du logiciel

## 1.6 Description des fonctionnalités

### 1.6.1 Définition de la fonction `attenuation()`

#### Identification de la fonction `attenuation()`

Présenter la fonction :

- nom de la fonction : `attenuation`;
- rôle, présentation générale : cette fonction a pour rôle d'atténuer les détails de l'image ;
- priorité associée à la réalisation de la fonction : primordiale.

#### Description de la fonction `attenuation()`

La fonction utilisera l'attribut `img` de type `Cimg<double>` et le modifiera afin d'atténuer les détails de l'image. Afin de pouvoir atténuer les détails de l'image, il faudra au préalable appliquer dessus un filtre bilatéral puis ensuite décomposer le rendu de façon pyramidale qui permettra d'obtenir un ensemble de vecteurs comme sur l'image 1.5.

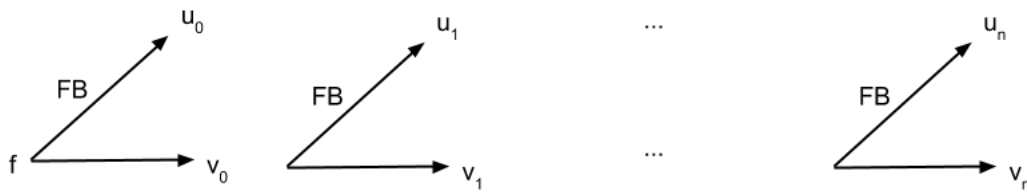


FIGURE 1.5 – Vecteurs de la décomposition après un filtre bilatéral de l'image

L'atténuation se fait en utilisant un coefficient multiplicateur  $\alpha$ .

$$f = \sum_{i=0}^n u_i + v_n \quad (1.1)$$

$$T(f) = \sum_{i=0}^n \alpha_i u_i + v_n \quad (1.2)$$

Il n'y a pas de gestion des erreurs de prévisions pour cette fonction.

### 1.6.2 Définition de la fonction raising()

#### Identification de la fonction raising()

Présenter la fonction :

- nom de la fonction : raising;
- rôle, présentation générale : cette fonction a pour rôle de rehausser les détails de l'image;
- priorité associée à la réalisation de la fonction : primordiale.

#### Description de la fonction raising()

Cette fonction est similaire à celle d'atténuation (1.6.1). Mais au contraire de celle-ci, elle a pour but rehausser les détails de l'image. Il faudra tout d'abord appliquer un filtre bilatéral à l'image puis décomposer ce rendu de façon pyramidale. Le résultat de la décomposition sera comme précédemment multiplié par un coefficient.

## 1.7 Conditions de fonctionnement

### 1.7.1 Performances

Au moment de la réalisation de ce cahier, on ne peut pas encore chiffrer les performances attendues car c'est l'un des objectifs de ce projet. En effet, il faudra chercher par la suite à optimiser le code. Les modifications apportées à l'image devront se faire en un temps de l'ordre de la seconde, c'est à dire inférieur à une minute. Cette performance sera demandée sur des images de taille 256x256.

### 1.7.2 Capacités

L'application pourra limiter la taille des images si celle-ci sont trop grosses car elle nécessiteront un temps de calcul trop important. Mais devra au minimum pouvoir fonctionner avec des images de taille 256x256.



### 1.7.3 Contrôlabilité

L'exécution d'un traitement pourra être suivi au travers d'un fichier de log indiquant son avancement ainsi que la durée de celle-ci.

# Plan de développement

## 2.1 Découpage du projet en tâches

Toutes les tâches hormis la dernière (Gestion de projet) sont dépendantes les unes des autres et seront de ce fait réalisées dans l'ordre dans lequel elles sont présenté ci-dessous.

### 2.1.1 Implémentation du filtre bilatéral

#### Description de la tâche

Cette tâche consiste en l'implémentation d'une méthode naïve du filtre bilatéral . La méthode naïve devra appliquer un filtre bilatéral sur une image de petite taille, c'est à dire  $256 \times 256$ . Cette méthode s'appuie sur un algorithme d'une implémentation direct du filtre bilatéral [3] (voir l'algorithme 1).

#### Cycle de vie

Cette tâche se situe dans la phase de développement du cycle de vie.

#### Livrables

À la fin de cette tâche, un livrable sera produit. Il prendra en paramètre le nom de l'image ainsi que les valeurs de  $\sigma_s$  et  $\sigma_r$ . À l'écran sera affichée dans deux fenêtres différentes l'image de base et la nouvelle image créée. Cette dernière sera enregistrée dans le dossier où se trouve l'exécutable et portera le même nom que l'image de base plus les valeurs de  $\sigma_s$  et  $\sigma_r$ . De plus, un fichier texte sera crée (s'il n'existe pas encore) et contiendra le nom des images de base et résultante ainsi que les valeurs des paramètres et le temps d'exécution.

#### Estimation de charge

Cette tâche est estimé à 4 jours/homme.

### 2.1.2 Utilisation du filtre bilatéral en débruitage

#### Description de la tâche

Durant cette tâche, il cherche à tester le filtre bilatéral précédemment crée (voir 2.1.1) en débruitant auparavant l'image. Lors de chaque test, il faudra :

- Enregistrer l'image résultante
- Prendre le temps de calcul
- Échantillonner
- Calculer le PSNR.

#### Cycle de vie

Cette tâche s'inscrit à la suite de celle vu précédemment (2.1.1) et correspondra aux tests fonctionnels de celle-ci.

#### Livrables

À la fin de cette tâche sera livré un fichier contenant toutes les informations demandés ainsi que toutes les images obtenues.

#### Estimation de charge

Cette tâche est estimé à 2 jours/homme.

### 2.1.3 Utilisation du filtre bilatéral en décomposition

#### Description de la tâche

L'objectif de cette tâche est de trouver la valeur optimal des paramètres  $\sigma_s$  et  $\sigma_r$  afin de réaliser une décomposition plus ou moins forte. Nous allons tester toutes les valeurs de ces paramètres compris dans un certain ensemble à déterminer et nous garderons ceux qui fourniront les résultats les plus pertinents.

#### Cycle de vie

Il s'agit d'une phase de test qui permettra de trouver les paramètres optimaux pour la suite du projet et du développement de celui-ci.

#### Livrables

Les valeurs des paramètres trouvés seront livrés à l'issue de cette tâche.

#### Estimation de charge

Cette tâche se fera en parallèle d'autres car elle prendra du temps dû à la réalisation de nombreux tests. Elle est estimé à 4 jours/hommes.

### 2.1.4 Pyramide multi-échelle à partir du filtre bilatéral

#### Description de la tâche

Tout d'abord il faudra étudier l'article "Edge-Preserving Decompositions for Multi-Scale Tone and Detail Manipulation" [4] et implémenter à la suite de cette lecture une méthode de décomposition multi-échelle du filtre bilatéral. Une fois l'implémentation réalisée, il faudra faire beaucoup de tests de celle-ci.

#### Cycle de vie

Cette tâche s'inscrit dans la partie de développement et de tests fonctionnels.

L'implémentation de la méthode de décomposition multi-échelle du filtre bilatéral devra donner un exécutable.

#### Estimation de charge

Cette tâche est estimé à 10 jours/homme.

### 2.1.5 Manipulation des détails

#### Description de la tâche

Il faudra mettre en place deux niveaux de manipulation des détails, le premier permettra d'atténuer les détails et le second de les réhausser. Cela sera possible grâce à l'utilisation de coefficients multiplicateurs qui modifieront le niveaux de détails des différentes images résultant de la décomposition mise précédemment en place.

#### Cycle de vie

Il s'agit d'une phase de développement.

## Livrables

La manipulation des détails sera possible grâce à un exécutable dont les paramètres permettront d'indiquer le niveau de manipulation des détails souhaités ainsi que l'image de base.

## Estimation de charge

La charge de travail de cette tâche est estimé à 7 jours/homme.

### 2.1.6 Interface graphique

#### Description de la tâche

L'interface graphique permettant à l'utilisateur de modifier une image en choisissant le niveaux de détails à manipuler sera réalisée durant cette tâche. Elle reprendra la charte graphique définie précédemment (voir 1.4.2 page 8). C'est à dire que l'utilisateur devra pouvoir

- charger une image
- choisir un des deux niveaux de détails et afficher le résultat
- avoir la possibilité d'introduire des informations complémentaires manuellement
- sauvegarder l'image résultante.

#### Cycle de vie

Il s'agit d'une tâche de développement qui engendra une phase de test afin de vérifier qu'il n'y a pas de faille.

## Livrables

Le livrable délivré à la fin de cette tâche sera le prototype fonctionnel avec l'interface graphique développé tout au long du projet.

## Estimation de charge

Cette tâche est estimé à 10 jours/homme en comptant la formation à l'environnement de développement (Qt) nécessaire à la réalisation de l'interface graphique.

### 2.1.7 Accélération

#### Description de la tâche

Cette tâche débutera par la lecture d'un article sur l'accélération de l'implémentation du filtre bilatéral [5]. Ensuite, il faudra étudier l'implémentation du filtre bilatéral pour MatLab [6]. Une nouvelle implémentation devra être réalisée suite à l'étude de l'article et de l'implémentation pour MatLab. Des tests seront effectués afin de comparer les nouveaux résultats avec ceux obtenus précédemment avec Clmg . Cette nouvelle partie devra être intégré à l'application.

#### Cycle de vie

Cette tâche s'inscrit dans différente partie du cycle de vie, en effet au début il faudra faire de l'analyse et étudier la mise en place d'une nouvelle implémentation. Puis, ensuite la tâche passera dans une phase de développement et ensuite de test.



### **Livrables**

À l'issue de cette tâche seront délivré un exécutable comportant la nouvelle implémentation du filtre bilatéral ainsi qu'une comparaison des résultats entre cette méthode et celle de CImg.

### **Estimation de charge**

Cette tâche est estimé à 7 jours/homme.

## **2.1.8 Modification de l'algorithme de décomposition**

### **Description de la tâche**

Cette tâche a pour objectif de changer l'algorithme utilisé afin de décomposer une image car il n'utilisera plus le filtre bilatéral. Le nouvel algorithme sera basé sur la méthode moindres carrés pondérés. Ensuite, il faudra construire une structure de graphe et manipuler l'image à partir de ce graph.

### **Cycle de vie**

Cette tâche aura tout d'abord une phase d'analyse avant de passer à une phase de développement puis de test.

### **Livrables**

Le livrable sera un exécutable permettant la manipulation d'image avec cette nouvelle méthode.

### **Estimation de charge**

La charge de travail est estimé à 5 jours/homme.

## **2.1.9 Travail sur les modèles 3D**

### **Description de la tâche**

Cette tâche est pour le moment facultative, elle sera réalisée s'il reste du temps. Elle aura pour but d'appliquer un filtre bilatéral sur un modèle 3D.

### **Cycle de vie**

Ce sera une tâche d'analyse, de développement puis de test.

### **Livrables**

Une fois la tâche fini, il faudra livré un exécutable permettant l'application d'un filtre bilatéral sur des modèles 3D.

### **Estimation de charge**

La charge de travail de cette tâche est estimé à 5 jours/homme.

## 2.1.10 Gestion de projet

### Description de la tâche

Cette tâche sera effective durant tout le projet et permettra de le gérer. Elle contient la rédaction de tous les cahiers et rapport (cahier de spécification système, plan développement, rapport de fin de projet) mais aussi la préparation pour les soutenances. Cette tâche prend aussi en compte la gestion du code grâce à un système de versionning (Git) mis en place sur le projet, la gestion du travail et du temps sans oublier l'analyse et la compréhension du projet.

### Cycle de vie

Cette tâche interviendra dans toutes les phases du cycle de vie du projet.

### Livrables

Les différents rapports et présentations seront délivrés à l'issue de cette tâche.

### Estimation de charge

Cette tâche sera effective durant toute la durée du projet.

## 2.2 Planning

Le diagramme de Gantt (image ci-dessous) reprend le planning suivant l'estimation des tâches faites précédemment. À la fin de chaque tâche sera délivré un livrable sous la forme d'un rapport ou d'un composant logiciel. Les périodes où seulement la tâche "Gestion de projet" est effectuée sera réservé à la rédaction des différents livrables et la préparation des soutenances.

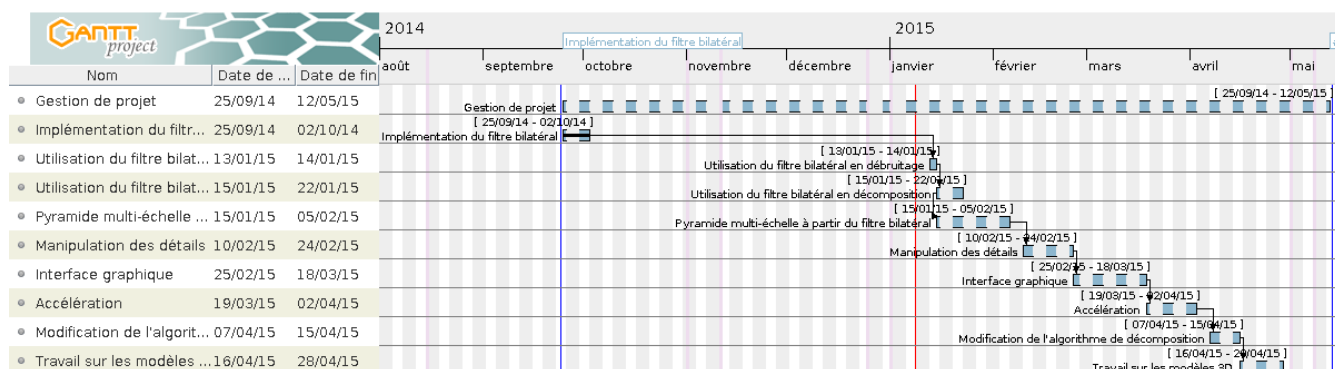


FIGURE 2.6 – Diagramme de Gantt du planning du projet

# Annexes

# Références

---

---

**Algorithme 1** Implémentation naïve du filtre bilatéral

---

```
1: pour chaque pixel p de S faire /* Parcours de chaque pixel de l'image S */
2:    $BF[I]_p = 0$  /* Initialisation */
3:    $W_p = 0$ 
4:   pour chaque pixel q de S faire /* */
5:      $w = G_{\sigma_s}(\|p - q\|)G_{\sigma_r}(\|I_p - I_q\|)$ 
6:      $BF[I]_p += wI_q$ 
7:      $W_p += w$ 
8:   fin pour
9:    $BF[I]_p = I_p/W_p$  /* Normalisation */
10: fin pour
```

---

# Glossary

---

**PSNR** signifie Peak Signal to Noise Ration, permet d'évaluer la qualité d'une image. 14

# Index

---

Atténuation des détails, 6, 8, 10, 15

Clmg, 6, 7, 16

Décomposition multi-échelle, 6, 10, 11, 15

Filtre bilatéral, 6, 10, 11, 14

Planning, 18

Réhaussement des détails, 6, 8, 11, 15

# Bibliographie

---

- [1] Cimg. [cimg.sourceforge.net/](http://cimg.sourceforge.net/).
- [2] Qt. [qt-project.org/](http://qt-project.org/).
- [3] Jack Tumblin Sylvain Paris, Pierre Kornprobst and Frédo Durand. Bilateral filtering : Theory and applications. *Computer, Graphics and Vision*, 4(1), 2008.
- [4] Zeev Farbman Raanan Fattal Dani Lischinski Richard Szeliski. Edge-preserving decompositions for multi-scale tone and detail manipulation. *ACM Transactions on Graphics*, 27(3), 2008.
- [5] Sylvain Paris and Frédo Durand. A fast approximation of the bilateral filter using a signal processing approach. *Computer Vision - ECCV 2006*, 2006.
- [6] Matlab. [fr.mathworks.com/products/matlab/](http://fr.mathworks.com/products/matlab/).