

---

# MOSAÏQUAGE D'IMAGES PAR BLENDING MULTI-RÉSOLUTION

---

## Rapport Master 2

Institut Géographique National



Maître de stage :  
Didier BOLDO

Rapporteur universitaire :  
Pascal GUITTON



Institut Géographique National 2, Avenue Pasteur 94 160 Saint Mandé
---

Marc NAMUR

— du 02 avril au 28 septembre 2007 —

## **Résumé**

L'Institut Géographique National réalise, renouvelle périodiquement et diffuse la couverture photographique aérienne du territoire national. En premier lieu, les épreuves photographiques des vues aériennes sont réalisées par des avions spécialement équipés pour la prise de vues à axe vertical. Ensuite, après une phase d'aérotriangulation, un ensemble d'ortho-images est obtenu. Enfin, les images sont assemblées pour former une mosaïque complète.

Pour composer cette mosaïque, il est particulièrement important que les limites entre les images soient les moins visibles possible. Cependant les ortho-images possèdent des différences radiométriques au niveau de leurs zones de recouvrement. C'est pourquoi un modèle pseudo physique de correction radiométrique, prenant en compte les effets directionnels et permettant un calcul automatique de ligne de raccord, a été mis en application. Toutefois des artefacts visuels sont toujours présents.

Le but du projet est de combiner la ligne de raccord précédemment calculée avec un blending multi-résolution ; ce traitement opère à plusieurs niveaux de fréquences et permet d'homogénéiser les apparences des deux images le long de la ligne de raccord tout en préservant les détails contenus dans les images originales.

# Table des matières

<b>Remerciements</b>	<b>3</b>
<b>1 Introduction</b>	<b>4</b>
1.1 Présentation . . . . .	4
1.2 Contenu du rapport . . . . .	4
1.3 L'institut Géographique National . . . . .	5
1.3.1 Présentation de l'entreprise . . . . .	5
1.3.2 Processus de production . . . . .	5
1.3.3 Obtention des ortho-images . . . . .	5
1.3.4 Différences radiométriques . . . . .	5
1.3.5 Le laboratoire MATIS . . . . .	6
<b>2 Mosaïquage d'images</b>	<b>7</b>
2.1 Nadir au sol le plus proche . . . . .	7
2.2 Ligne de raccord . . . . .	7
2.2.1 Mosaïque . . . . .	7
2.3 Blending . . . . .	8
2.3.1 Enblend . . . . .	8
2.3.2 Smartblend . . . . .	10
2.3.3 Watersheds / Graph cuts . . . . .	12
2.3.4 Gradient domain blending . . . . .	12
<b>3 Introduction au projet</b>	<b>13</b>
3.1 Objectifs et méthodologie proposée . . . . .	13
3.2 Enblend - Schéma de fonctionnement . . . . .	14
3.2.1 Description . . . . .	15
<b>4 Besoins non fonctionnels et fonctionnels</b>	<b>16</b>
4.1 Besoins non fonctionnels . . . . .	16
4.1.1 Données . . . . .	16
4.1.2 Temps d'exécution . . . . .	16
4.2 Besoins fonctionnels . . . . .	16
4.2.1 Géoréférencement . . . . .	16
4.2.2 Canaux alpha . . . . .	17
4.2.3 CGraphe . . . . .	17
4.2.4 Adaptation . . . . .	17
4.2.5 Sauvegarde en plusieurs fichiers . . . . .	17

<b>5</b>	<b>Architecture</b>	<b>18</b>
5.1	Diagramme UML - cas d'utilisation . . . . .	18
5.2	Diagramme de classes . . . . .	19
5.2.1	Classes implémentées . . . . .	21
5.2.2	Classes importées . . . . .	22
<b>6</b>	<b>Implémentation</b>	<b>23</b>
6.1	Blending sans canaux alpha . . . . .	23
6.2	Géoréférencement . . . . .	24
6.2.1	Utilisation . . . . .	24
6.3	Fichiers . . . . .	24
6.4	Construction du masque . . . . .	24
6.4.1	Description . . . . .	24
6.4.2	Modifications . . . . .	24
6.5	Sauvegarde en plusieurs fichiers . . . . .	25
<b>7</b>	<b>Tests</b>	<b>26</b>
7.1	Tests de non-régression . . . . .	26
7.1.1	Fonctionnement avec 2 fichiers . . . . .	26
7.1.2	Fonctionnement avec n fichiers . . . . .	27
7.2	Test d'intégration . . . . .	27
7.3	Test comparatif . . . . .	28
7.4	Tests systèmes . . . . .	30
7.4.1	Ligne de mosaïquage . . . . .	30
7.4.2	Cas réel . . . . .	31
<b>8</b>	<b>Bilan</b>	<b>32</b>
<b>A</b>	<b>Manuel utilisateur</b>	<b>33</b>
A.1	Commandes . . . . .	33
A.1.1	Commande supplémentaire . . . . .	33
A.2	Options . . . . .	33
A.2.1	Options supplémentaires . . . . .	34
	<b>Index</b>	<b>35</b>
	<b>Glossaire</b>	<b>36</b>
	<b>Bibliographie</b>	<b>38</b>

# Remerciements

J'aimerais remercier tout d'abord Gilles Martinoty sans qui la mise au point du blending sur un ensemble d'ortho-images du territoire national aurait été beaucoup plus difficile. Plusieurs fois il m'a aidé à faire le point sur les problématiques rencontrées, et il m'a guidé tout au long du développement du logiciel.

J'aimerais remercier aussi Didier Boldo, Nicolas Paparoditis, Jean-Pierre Papelard et Erwann Houzay qui m'ont apporté une aide précieuse sur des questions de programmation notamment, et m'ont donné des conseils avisés quant aux implémentations à mettre en oeuvre.

Je remercie également François Boyero qui a mis en place l'environnement de développement et réglé les problèmes techniques rapidement.

Enfin je remercie l'ensemble de l'équipe du MATIS, pour le calme et la bonne humeur qui règnent dans le laboratoire, et les stagiaires avec qui j'ai aussi beaucoup appris.

# Chapitre 1

## Introduction

### Avant-propos

Pour faciliter la lecture de ce document, la première occurrence des termes techniques du rapport est mise en valeur. Une définition en est donnée dans le glossaire page 37. De plus, pour une utilisation directe du logiciel, un manuel utilisateur a été ajouté en annexe A page 33.

### 1.1 Présentation

Ce rapport présente un stage effectué à l’Institut Géographique National. Son objectif est d’obtenir un outil fonctionnel permettant de réaliser un *mosaïquage* amélioré entre plusieurs *ortho-images*, ortho-images qui sont issues de photos aériennes du territoire national. Tout d’abord l’entreprise dans laquelle le stage a été effectué est présentée, et ensuite le laboratoire du *MATIS*<sup>1</sup>, dans lequel j’ai effectué le stage. Enfin différents modèles pour réaliser le mosaïquage sont décrits, y compris celui mis en oeuvre au cours du stage, i.e. le *blending* multi-résolution.

### 1.2 Contenu du rapport

Après ces descriptions, une analyse de modèles existants pour réaliser le mosaïquage d’images est donnée, notamment celui mis en application dans le logiciel *enblend*. Ensuite, après une courte introduction au projet et une liste des fonctionnalités requises par le programme, l’architecture du programme est présentée sous forme de diagrammes UML et les fonctionnalités additionnelles implémentées sont détaillées. Enfin, l’ensemble des tests effectués au cours et après la réalisation du programme sont décrits.

---

<sup>1</sup>Méthodes d’Analyse et de Traitement d’Images pour la Stéréorestitution

## 1.3 L'institut Géographique National

### 1.3.1 Présentation de l'entreprise

Un bref historique est donné ici pour présenter l'entreprise dans laquelle le stage a été réalisé. L'Institut Géographique National a été créé par décret le 27 juin 1940 et a succédé au Service Géographique de l'Armée, fondé en 1887. En 1947, l'IGN reçoit la mission de réaliser la couverture photographique de la France, mais aussi de l'Afrique du Nord, de l'Afrique occidentale, du Madagascar, des états associés d'Indochine et des départements et territoires d'Outre-Mer. Établissement public à caractère administratif depuis le 1<sup>er</sup> janvier 1967, il est placé sous la tutelle du ministère de l'écologie, du développement et de l'aménagement durables.

### 1.3.2 Processus de production

C'est dans le cadre de la réalisation et de la diffusion de la couverture photographique du territoire national que se situe ce projet. Pour mener à bien cette mission, un certain nombre de procédés sont mis en oeuvre, notamment l'acquisition des images, l'*aérotriangulation*, l'égalisation radiométrique et l'*ortho-rectification*.

### 1.3.3 Obtention des ortho-images

La géométrie d'un cliché aérien diffère de celui d'une carte du fait du relief du terrain, de la géométrie du capteur (projection conique, distorsion), de la trajectoire du rayon lumineux dans l'atmosphère et de l'orientation du cliché. Le calcul d'une ortho-image nécessite donc des images numériques orientées, un *MNT* décrivant le relief du terrain, et la définition d'un graphe de mosaïquage permettant de fournir en tout point de l'ortho-image le cliché à partir duquel l'information radiométrique est obtenue. L'objectif est de fabriquer une carte dont le contenu est l'information radiométrique de la photo.

### 1.3.4 Différences radiométriques

Toutefois, une fois l'ortho-rectification effectuée, des différences radiométriques persistent dans les zones où les clichés se superposent.<sup>2</sup> Ces différences sont issues principalement des approximations du MNT, plus particulièrement dans les zones de fort relief, et aussi des phénomènes de *hot spots*, liés à l'éclairement du soleil. De plus, lorsque l'orientation du cliché est rasante par rapport au relief, des *coulées de pixels* apparaissent.

---

<sup>2</sup>Les recouvrements des clichés sont de 60%(±5%) pour le recouvrement longitudinal et de 20%(±5%) pour le recouvrement latéral [2]



FIG. 1.1 – coulée de pixels

À cela s'ajoute les différences de dates de prises de vue entre les clichés, les ombres qui diffèrent suivant l'heure de la prise de vue, et d'autres phénomènes non négligeables tels que les conditions météorologiques, la hauteur de vol de l'avion, les caméras utilisées et leurs réglages.

Même si des traitements préalables permettent de corriger une partie de ces effets à partir de modèles physiques, il reste des variations importantes entre deux clichés couvrant une même zone, notamment lorsqu'ils n'appartiennent pas à une même bande de prise de vue.

### 1.3.5 Le laboratoire MATIS

Le MATIS, un des 4 laboratoires du service de recherche, mène des recherches contribuant à l'amélioration de la production et de la mise à jour des bases de données image ou vecteur, actuelles et futures, à partir des données images brutes. Le laboratoire est structuré en 4 actions de recherche :

- *ARCHI* : photogrammétrie terrestre et architecturale.
- *ARTIC* : analyse de la radiométrie, de la texture des images pour la classification et la cohérence bases de données images.
- *BRIGE* : reconstruction du bâti pour la cartographie à grande échelle.
- *LASER* : lidar aéroporté.

En particulier, un des objectifs de l'action de recherche *ARTIC* est de réunir les images couvrant l'ensemble du territoire pour former une mosaïque uniforme.



## Chapitre 2

# Mosaïquage d'images

Les ortho-images géoréférencées obtenues à la fin de la phase d'ortho-rectification possèdent des différences au niveau de leur zone de recouvrement. Pour les réunir en une mosaïque finale, il faut donc déterminer la contribution de chacune des images dans cette zone. Les modèles suivants font intervenir principalement deux opérations : l'une étant de déterminer une ligne de raccord délimitant les zones de contribution des images sources, l'autre consistant à lisser la zone de transition entre les images.

### 2.1 Nadir au sol le plus proche

Un premier modèle consiste à prendre uniquement pour chaque pixel celui du *nadir au sol* le plus proche, de manière à minimiser les différences radiométriques liées à l'éloignement du centre de prise de vue. Même si ce modèle a l'avantage d'être plus proche de la réalité observée, la mosaïque finale ainsi obtenue contient des frontières marquées au niveau des zones de transition.

### 2.2 Ligne de raccord

C'est pourquoi un modèle plus élaboré a été mis en application. Il consiste à trouver une ligne de raccord séparant les ortho-images initiales telle que les différences radiométriques au niveau de cette ligne soient minimales. Pour les différences radiométriques inter-images et dans le cas où chaque pixel de l'image finale est un pixel d'une des images source, la solution optimale est obtenue.

#### 2.2.1 Mosaïque

L'algorithme a été développé à l'IGN par Hervé Le Men<sup>1</sup> et Didier Boldo<sup>2</sup>. Dans un premier temps un graphe de la mosaïque est construit à partir des contours des images sources. Ensuite, dans chaque zone de recouvrement, une ligne de raccord de coût minimal est déterminée. Le graphe est ainsi transformé en un graphe de lignes de raccord.

---

<sup>1</sup>Direction technique. IGN, 2 avenue Pasteur 94165 Saint-Mandé. herve.le-men@ign.fr

<sup>2</sup>Laboratoire MATIS. IGN, 2 avenue Pasteur 94165 Saint-Mandé. didier.boldo@ign.fr

## Algorithme

La fonction de coût est basée sur :

- les coefficients de corrélation linéaire entre imageries homologues.
- les différences radiométriques inter-images.

Le terme basé sur le coefficient de corrélation va avoir tendance à amener la ligne de raccord vers des détails homologues et le terme basé sur la différence radiométrique sera plus pertinent pour traverser les zones homogènes.



FIG. 2.1 – mosaïquage d’ortho-images

## 2.3 Blending

Toutefois, sur certaines ortho-images, des différences radiométriques sont toujours présentes. Pour améliorer le résultat perceptif obtenu dans la mosaïque finale, d’autres modèles font intervenir plusieurs pixels de chaque image source dans le pixel de l’image finale.

### 2.3.1 Enblend

Le logiciel open-source *enblend* est développé par Andrew Mihal<sup>3</sup>. Il permet de composer des images qui se recouvrent en partie, et qui présentent des différences radiométriques au niveau de leur zone de recouvrement. Le programme va trouver des lignes de raccord entre les images et effectuer une opération de blending pour homogénéiser la zone de transition, et ainsi la rendre invisible ou tout au moins difficile à percevoir. À noter que cet algorithme n’est pas limité à l’utilisation d’images aériennes, et qu’il peut tout aussi servir au mosaïquage de photos quelconques.

<sup>3</sup>Université de Berkeley, Californie. [acmihal@users.sourceforge.net](mailto:acmihal@users.sourceforge.net)

## Algorithme

L'algorithme utilisé dans *enblend* est basé sur le blending multi-résolution, modèle décrit dans le papier de Peter J. Burt et Edward H. Adelson, « A multiresolution spline with application to image mosaics » [7], et également sur le site d' *enblend* [5]. Il permet de mieux homogénéiser les apparences des deux images le long de la ligne de raccord, sans introduire de flou dans les hautes fréquences.

Le blending effectué par le logiciel s'opère initialement sur deux images afin d'obtenir une image résultante homogène. À chaque étape l'image de sortie est utilisée avec l'image d'entrée suivante, et le processus est itéré. Une fois que toutes les images ont été traitées, nous obtenons le panorama ou la mosaïque finale.

Pour chaque paire d'images, les étapes du blending sont les suivantes :

- *enblend* construit un masque binaire décrivant la contribution de chacune des deux images. La ligne de séparation entre les deux images est construite à l'aide du diagramme de Voronoï :

  1. Une image en niveaux de gris est construite, image contenant
    - 0 dans les zones où seule la première image est présente.
    - 255 dans les zones où seule la deuxième image est présente.
    - 1 partout ailleurs.
  2. À partir de l'image précédente, *enblend* crée un squelette de la ligne de raccord en utilisant le diagramme de Voronoï . i.e. il remplace les pixels de valeur 1 par la valeur différente de 1 la plus proche dans l'image.
  3. Enfin, *enblend* optimise le placement de la ligne de raccord en utilisant un contour actif.

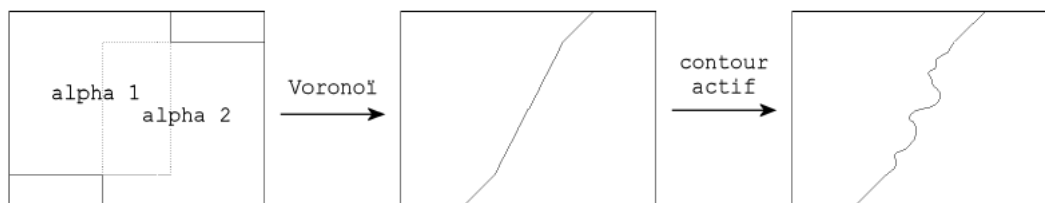


FIG. 2.2 – enblend - construction du masque binaire

- Une pyramide gaussienne du masque est alors construite (son nombre de niveaux peut être spécifié par l'utilisateur).
- Ensuite, une pyramide laplacienne est construite pour chacune des deux images source.
- Enfin, l'application de chaque niveau de la pyramide gaussienne du masque à chaque niveau des pyramides laplaciennes des images permet d'effectuer un traitement sur plusieurs niveaux de fréquences.

NB : les premiers niveaux des pyramides laplaciennes des images, niveaux correspondant aux hautes fréquences, sont associés aux premiers niveaux de la pyramide gaussienne du masque. Il en résulte que la zone de traitement utilisée pour effectuer le blending est plus grande à mesure que l'on descend dans les fréquences des images source. i.e. le premier niveau des pyramides est uniquement une juxtaposition de la première image et de la seconde.

### 2.3.2 Smartblend

Le logiciel *Smartblend* est développé par Michael Norel<sup>4</sup>. Comme *enblend*, le logiciel positionne la ligne de mosaïquage et effectue un blending des images constituant le panorama ou la mosaïque finale. *Smartblend* corrige aussi des erreurs liées à la *parallaxe* ou à des différences d'exposition lors de la prise de la photo. Additionnellement, *Smartblend* contient des traitements spécifiques pour les objets mobiles. Il est basé, tout comme *enblend*, sur le modèle de blending multi-résolution[7]. Lorsque l'option `-PyramidVerbose` est activée, le logiciel copie les pyramides utilisées pour le blending dans des fichiers images :

---

<sup>4</sup>Odessa, Ukraine. [minorlogic@yahoo.com](mailto:minorlogic@yahoo.com)

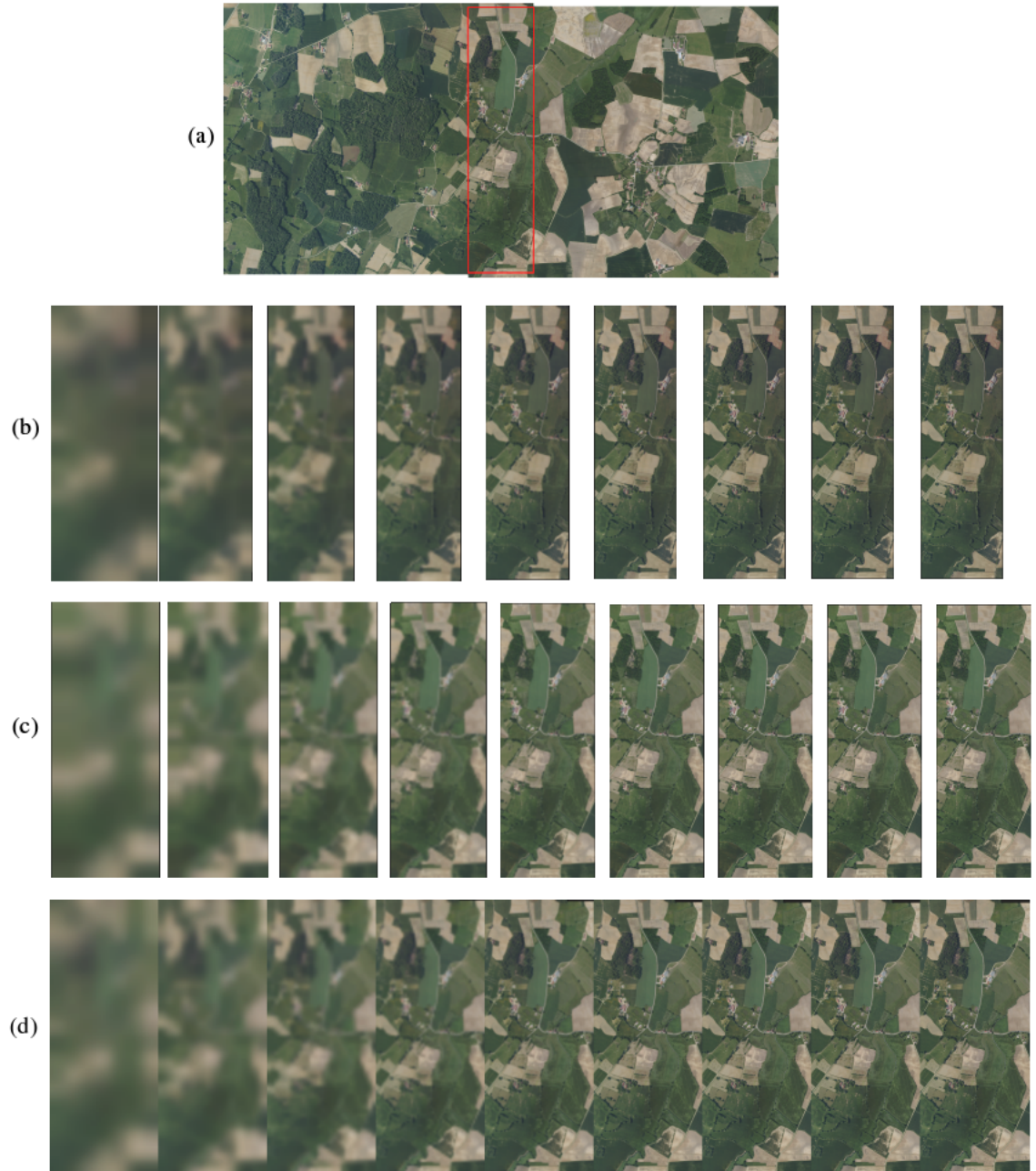


FIG. 2.3 – *Smartblend*

- (a) mosaïquage de deux ortho-images prises à des dates différentes.
- (b) pyramide gaussienne de la zone de recouvrement dans l'image de gauche.
- (c) pyramide gaussienne de la zone de recouvrement dans l'image de droite.
- (d) pyramide finale.

Pour le lissage de la zone de transition, la méthode mise en application par *Smart-blend* est similaire à celle utilisée par *enblend*. Comme dans *enblend*, la zone de recouvrement est élargie pour pouvoir appliquer tous les niveaux de la pyramide gaussienne du masque. Mais les résultats obtenus sont différents car la ligne de raccord utilisée par chacun des deux logiciels est différente, et également différente de celle calculée par *Mosaïque*.

### 2.3.3 Watersheds / Graph cuts

De manière à trouver une ligne de raccord optimale en n'effectuant pas la recherche sur l'ensemble des pixels de la zone de recouvrement, il est possible d'utiliser la méthode de segmentation de lignes de partage des eaux — *watersheds* — pour diviser la zone de recouvrement en régions traitées indépendamment. Cette méthode, combinée avec celle des *graph cuts* pour associer chaque région délimitée par la ligne de partage des eaux à une des images source, est décrite dans le papier de Nuno Gracias et Art Gleason, « Fast Image Blending using Watersheds and Graph Cuts » [6].

### 2.3.4 Gradient domain blending

Dans le papier « Seamless Image Stitching in the Gradient Domain » [4], Anat Levin, Assaf Zomet, Shmuel Peleg et Yair Weiss introduisent un nouveau modèle pour le blending, opérant sur les gradients des images. Ce modèle est également utilisé dans l'article présenté au cours de la conférence *ACM SIGGRAPH 2004*, « Interactive Digital Photomontage » [3].

Pour évaluer la qualité du blending effectué, des fonctions de coût basées sur « la similarité entre l'image composée et chacune des images d'entrée et la visibilité de la ligne de raccord entre les images composées » sont utilisées. Les résultats sont aussi comparés avec ceux obtenus avec le blending multi-résolution. Ils sont meilleurs dans le domaine des gradients, lorsque la ligne de mosaïquage ne coïncide pas exactement entre les deux images.

## Chapitre 3

# Introduction au projet

### 3.1 Objectifs et méthodologie proposée

L'objectif est d'obtenir un outil fonctionnel permettant de réaliser un mosaïquage amélioré, étant données deux images. Les différentes étapes ont été les suivantes :

1. Prise en main du logiciel open source *enblend* ; analyse de la qualité du code et de la faisabilité de la réutilisation du code.
2. Prise en main du code de mosaïquage développé à l'IGN ; analyse de l'intégration possible du blending multirésolution.
3. En fonction des analyses réalisées, développements informatiques nécessaires pour obtenir un prototype fonctionnel.
4. Améliorations complémentaires de l'algorithme : passage dans l'espace Lab, coefficient de corrélation local, etc.

Une des premières étapes dans le développement de l'extension d'*enblend* a été d'analyser le fonctionnement du logiciel. Le schéma de fonctionnement suivant en présente la fonction principale, *enblendMain* (cf. fichier *enblend.h*), et a permis de mieux comprendre les différentes étapes du blending mis en oeuvre dans le logiciel, pour ensuite lui ajouter des fonctionnalités.

### 3.2 Enblend - Schéma de fonctionnement

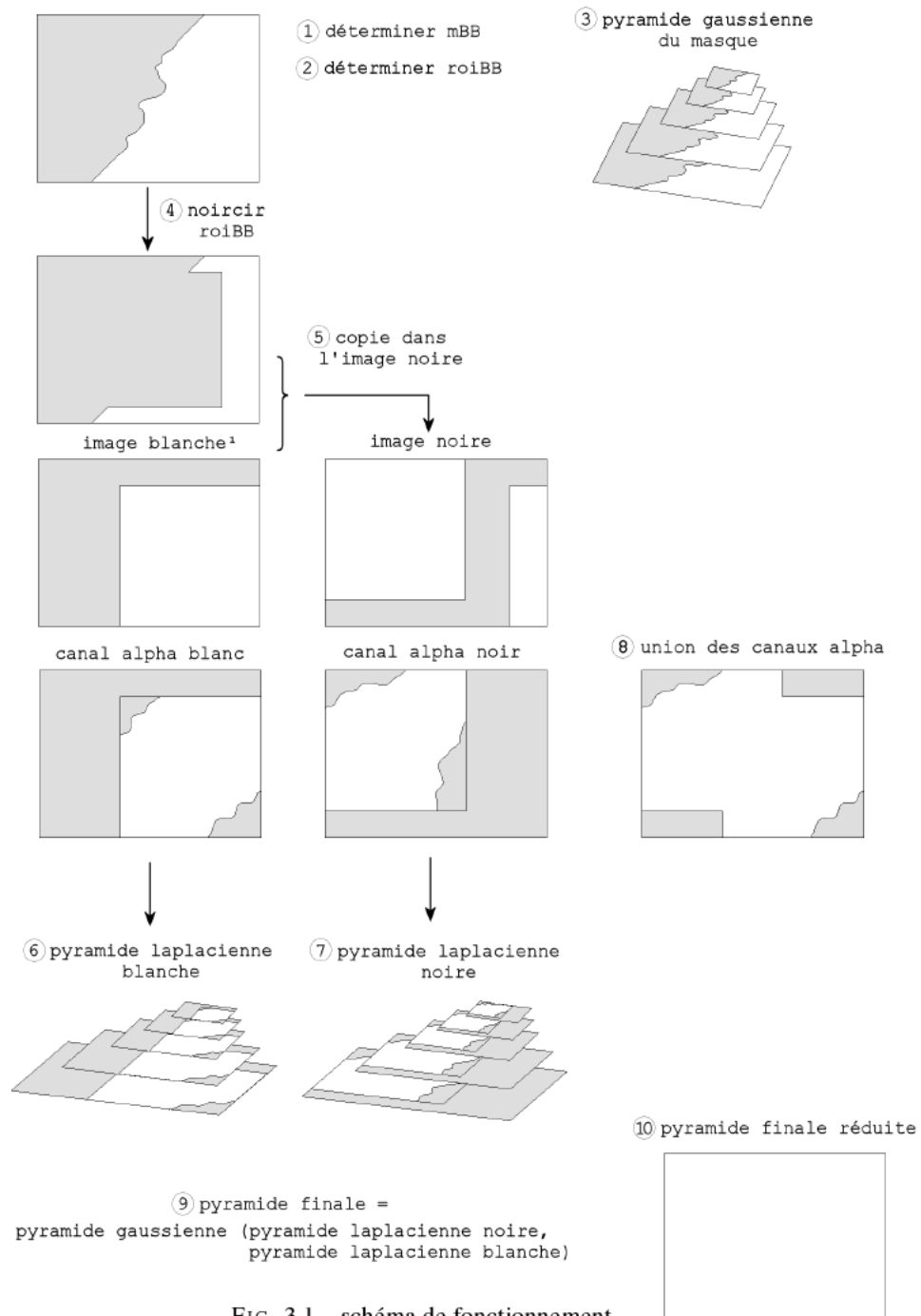


FIG. 3.1 – schéma de fonctionnement

<sup>1</sup>Dans le programme, l'image et le canal alpha correspondant à la zone blanche du masque sont stockés dans une paire nommée *whitePair*.



### 3.2.1 Description

Le schéma précédent décrit le fonctionnement du logiciel *enblend*, à partir du moment où le masque binaire a été obtenu, et jusqu'à l'obtention de l'image finale.

1. Tout d'abord, les boîtes englobantes *mBB* — mask transition line's bounding box — et *roiBB* — region of interest's bounding box — sont déterminées. *mBB* correspond à la plus petite boîte englobante contenant la ligne de mosaïquage.
2. *roiBB* est une boîte englobante initialement égale à *mBB*. Une bordure dont la taille est fonction du nombre de niveaux de la pyramide gaussienne lui est ajoutée. En effet, à chaque niveau de la pyramide, la taille de la zone de traitement autour de la ligne de mosaïquage est multipliée par 2.
3. Ensuite la pyramide gaussienne du masque est créée.
4. La partie correspondant à *roiBB* est alors noircie dans le masque.
5. La zone de l'image blanche correspondant à la partie active du masque est copiée dans l'image noire. Dans le programme ces deux images sont stockées dans des paires, à savoir *whitePair.first* — image blanche — et *blackPair.first* — image noire — car elles correspondent respectivement aux zones blanche et noire du masque.
6. L'image blanche et son canal alpha — *whitePair.first* et *whitePair.second* — sont utilisés pour créer la pyramide laplacienne blanche.
7. De même, l'image noire et son canal alpha — *blackPair.first* et *blackPair.second* — sont utilisés pour créer la pyramide laplacienne noire.
8. L'union des canaux alpha est placée dans un nouveau masque.
9. L'application de la pyramide laplacienne du masque aux pyramides laplaciennes blanche et noire génère la pyramide finale.
10. La réduction de la pyramide finale, qui s'opère par expansion des différents niveaux de la pyramide, puis addition de ces niveaux, génère l'image finale, résultante du blending multi-résolution.

## Chapitre 4

# Besoins non fonctionnels et fonctionnels

Une liste des besoins non fonctionnels et fonctionnels a ensuite été rédigée pour planifier l'ensemble des implémentations à réaliser.

### 4.1 Besoins non fonctionnels

#### 4.1.1 Données

Le programme devra pouvoir prendre en entrée des ortho-images au format *tif* compressées ou non, chacune représentant plusieurs milliers de pixels en hauteur comme en largeur. Les ortho-images sont de dimensions variables, et leur taille sera amenée à grandir à mesure que les photographies aériennes numériques gagneront en précision.

#### 4.1.2 Temps d'exécution

Le temps de traitement nécessaire au blinding de deux ortho-images devra pouvoir s'effectuer en l'espace de quelques minutes. Le calcul pourra être distribué à l'aide de la plate-forme *Condor* pour le traitement sur un nombre d'images plus important.

#### Calcul distribué

Même si dans un premier temps le processus de blinding s'effectuera sur deux images, le nombre d'images données en entrée pourra s'étendre à celui de la mosaïque complète du territoire national français. En ce sens, le programme pourra effectuer le traitement d'une mosaïque contenant  $n$  images avec le fichier du graphe des lignes de raccord correspondant.

### 4.2 Besoins fonctionnels

#### 4.2.1 Géoréférencement

Chaque *ortho-photographie* est positionnée sur le territoire national à l'aide de données de géoréférencement contenues dans un fichier de même nom que l'image

et portant l'extension *.grf*. Le programme devra pouvoir localiser les ortho-images et les disposer à partir de ces informations. La lecture des fichiers de géoréférencement et l'utilisation des coordonnées cartographiques qu'ils contiennent seront alors utilisées lors du blending.

#### 4.2.2 Canaux alpha

Des masques pour les images doivent être générés pour permettre à *enblend* de déterminer quelles zones il doit utiliser pour effectuer le blending. Ces masques pourront ensuite permettre d'éviter les zones que l'on souhaite ne pas traiter, par exemple des zones où des coulées de pixels sont apparentes. Dans un premier temps, un masque entièrement blanc, signifiant que l'intégralité de l'image est à utiliser pourra être créé si jamais aucun masque n'est présent dans l'image.

#### 4.2.3 CGraphe

Le programme devra pouvoir charger une ligne de mosaïquage au format *.gpo* et l'utiliser lors du blending. Pour ce faire, le portage sous linux et l'utilisation de la bibliothèque *CGraphe* de Sylvain Airault<sup>1</sup> permettra de récupérer les informations relatives au graphe de lignes de raccords.

#### 4.2.4 Adaptation

Les données précédentes, à savoir le géoréférencement, les canaux alpha et les données des fichiers créés par *Mosaïque* devront être converties si nécessaire pour leur utilisation ultérieure par *enblend*.

- les coordonnées cartographiques devront être converties en coordonnées images, si possible en utilisant les méthodes de la bibliothèque *CGraphe* existante.
- les canaux alpha devront être créés à partir des images originales dans le logiciel, ou plus en amont dans la chaîne de production.
- la ligne de mosaïquage devra être convertie de telle manière qu' *enblend* puisse l'utiliser, et générer un masque qui lui corresponde. Il pourra ainsi effectuer le blending multi-résolution en tenant compte des informations contenues dans le graphe. La ligne de mosaïquage initialement utilisée par *enblend* sera alors désactivée.

#### 4.2.5 Sauvegarde en plusieurs fichiers

Le programme est conçu pour sauvegarder le résultat du blending dans une image contenant la mosaïque ou le panorama complet. En option, il devra permettre de sauvegarder le résultat du blending effectué sur *n* images en *n* images de sortie. De cette manière, il sera possible de constituer un ensemble d'images ayant subi l'opération de blending, ensemble qui viendra s'ajouter à la base de données initiale.

---

<sup>1</sup>Laboratoire MATIS. IGN, 2 avenue Pasteur 94165 Saint-Mandé. sylvain.airault@ign.fr

## Chapitre 5

# Architecture

### 5.1 Diagramme UML - cas d'utilisation

Le diagramme UML suivant représente le fonctionnement du logiciel *enblend* une fois modifié et les différents cas d'utilisation. Il donne une vue d'ensemble des fonctionnalités ajoutées, la principale étant le chargement de lignes de raccords à partir d'un fichier *.gpo* et son utilisation dans le logiciel *enblend*.

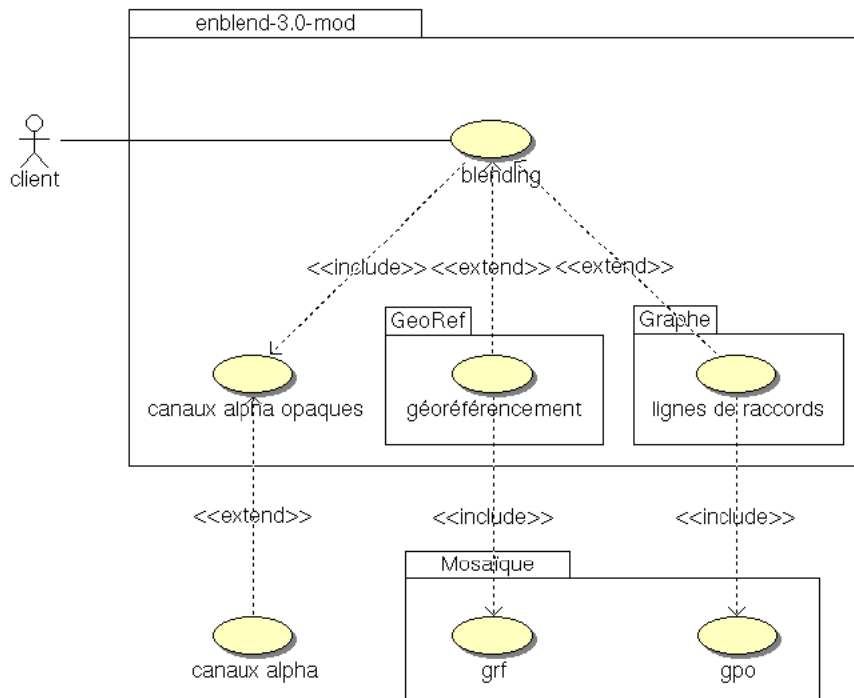


FIG. 5.1 – cas d'utilisation

À noter sur ce diagramme que les canaux alpha issus des images source sont liés aux canaux alpha créés dans le logiciel par une relation d'héritage et non d'inclusion car ces canaux ne sont pas nécessairement présents dans les images source. De plus, les fichiers de géoréférencements sont créés à partir d'un programme extérieur, fichiers qui doivent être lus par le logiciel au moyen de la classe *CGeoRef*. Il en est de même pour les fichiers de graphes de lignes de raccord créés dans le logiciel *Mosaïque*, et qui sont lus par la classe *CGrappe*.

## 5.2 Diagramme de classes

Dans ce diagramme de classes UML, seules les méthodes et variables membres utilisées par les nouvelles classes implémentées sont référencées. Les bibliothèques *Boost* et *Vigra* initialement utilisées dans le logiciel *enblend* ont été réutilisées.

- La première est utilisée pour la gestion des fichiers, notamment lorsque la sortie pour  $n$  images d'entrée est répercutée sur  $n$  images de sortie. (cf. section *Plusieurs images en sortie* page 17)
- La seconde est utilisée pour gérer les données relatives aux images, à savoir leur position avec la classe *Point2D*, et leur emprise avec la classe *Rect2D*.  
— Un objet *Rect2D* est l'association d'un objet *Point2D* et d'un objet *Size2D*, correspondant respectivement à la position et aux dimensions d'un rectangle. —

La conversion vers les données utilisées par *enblend* est plus aisée car le logiciel utilise également ces structures de données. Notamment la ligne de mosaïquage est stockée dans une variable de type *SeamSegment* :

```
typedef list <pair <bool, Point2D> > SeamSegment;
```

Initialement le logiciel *enblend* utilise des données de type *Segment* pour stocker la ligne de mosaïquage :

```
typedef slist <pair <bool, Point2D> > Segment;
```

La valeur du booléen est positionnée à *true* lorsque le segment est un contour actif. Dans la classe *Seamline*, une liste est utilisée et non une *slist* — *single linked list* ou liste simplement chaînée — car des éléments y sont ajoutés en fin.



### 5.2.1 Classes implémentées

#### classe *Seamline*

La classe *Seamline* a été implémentée pour modifier la ligne de mosaïquage d'*enblend*. La ligne de mosaïquage calculée par le logiciel *Mosaïque* et chargée à partir d'un graphe au format *.gpo* y est insérée. Le constructeur est défini comme suit :

```
Seamline( Rect2D  const &uBB, Rect2D  const &iBB );
```

Il contient deux paramètres initialement utilisés dans *enblend*, *uBB* — *union bounding box* — et *iBB* — *intersection bounding box* —, qui correspondent respectivement à l'union et à l'intersection de l'emprise des deux images. Deux méthodes publiques sont implémentées dans cette classe :

```
void replaceRawSegments( Contour * const rawSegments ,  
                          CGraphe &graph ,  
                          pair<size_t , size_t> const &zones ,  
                          Point2D const &imagesBBUpperLeft );  
  
void saveSeamContour( list<Point2D> * const seamContour ,  
                      Contour const &rawSegments );
```

- La première méthode est utilisée pour remplacer la ligne de mosaïquage d'*enblend* contenue dans *rawSegments*, par l'arc du graphe *graph* commun aux zones *zones*. Le paramètre *imagesBBUpperLeft* correspond au coin haut gauche des deux images en cours de traitement dans le repère de l'ensemble des images.
- La deuxième méthode est utilisée pour sauvegarder la ligne de mosaïquage dans une liste de points pour un affichage ultérieur.

#### classe *Files*

La classe *Files* a été implémentée pour créer un répertoire et ensuite y placer les images de sortie. En effet, les images source ne doivent en aucun cas être modifiées, et c'est pourquoi le logiciel opère sur une copie des fichiers d'entrée. Le constructeur utilise un pointeur sur l'ensemble des noms de fichiers d'entrée :

```
Files( list<char*> * const inputFileNameList );
```

Quatre méthodes publiques sont implémentées dans cette classe :

```
void retrieveCoordinates( vector<Rect2D> * const imagesBB );  
  
void copyInto( path const &blendingPath ,  
              vector<string> * const outputFileNames );  
  
void retrieveNames( string * const fileName1 ,  
                  string * const fileName2 ,  
                  string const &gpoFileName1 ,  
                  string const &gpoFileName2 );  
  
bool isInside( string const &gpoFileName );
```

- La première méthode est utilisée pour retrouver les coordonnées des images à partir des fichiers de géoréférencement qui leur sont associés.
- La seconde est utilisée pour copier l'ensemble des fichiers d'entrée dans un répertoire *blendingPath*. La liste des chemins d'accès à ces nouveaux fichiers est sauvegardée dans un vecteur de chaînes de caractères.
- Les chemins d'accès aux fichiers ont été enregistrés avec *Mosaïque* dans le fichier de graphe. La méthode *retrieveNames* utilise la bibliothèque *Boost* pour retrouver les noms des fichiers à partir des chemins d'accès.
- La dernière méthode est utilisée pour déterminer si le nom de fichier *gpoFileName* indiqué par le fichier de graphe *.gpo* existe dans la liste des fichiers donnée en entrée. En effet, il se peut que des fichiers mentionnés par le fichier *.gpo* soient manquants.

### 5.2.2 Classes importées

#### classe CGeoRef

La classe *CGeoRef* est importée pour lire les données de géoréférencement contenues dans des fichiers portant le même nom que les fichiers image, mais avec l'extension *.grf*. La méthode suivante est utilisée pour lire un fichier de géoréférencement :

```
int LireGeoRef( CString );
```

Les données membres *m\_Emprise.m\_Xmin* et *m\_Emprise.m\_Ymax*, ainsi que *m\_ResolX* et *m\_ResolY* sont utilisées pour accéder aux coordonnées référencées.

#### classe CGraphe

La classe *CGraphe* est importée pour lire les informations contenues dans un graphe au format *.gpo*. Les méthodes suivantes sont utilisées :

```
void InitGraphe(int Ns, int Nl, double X_NO, double Y_NO,
               double ResolX, double ResolY,
               bool Direct = true, int PasCalcul = 1);
int LireSurDisque( CAttribut NomFic);
int ArcCommun(int Zone1, int Zone2);
int NumZone(int ArcId);
CPointTerrain GrapheTerrain(const CPt& ptGr);
CPt TerrainGraphe(const CPointTerrain& ptTer);
```

Les deux premières méthodes sont utilisées pour initialiser un graphe ou le charger à partir d'un fichier *.gpo*. La méthode *ArcCommun* permet de retrouver le numéro de l'arc commun à deux zones. La méthode *NumZone* renvoie le numéro de la zone à droite de l'arc si *ArcId* est strictement positif et celui à gauche de la zone si *ArcId* est strictement négatif. Les deux dernières méthodes sont utilisées pour passer du repère image au repère cartographique, et vice-versa.

Les données membres suivantes sont utilisées : *m\_ListeZone* et *m\_ListeArc* pour accéder respectivement aux zones et aux arcs, et *m\_ListeAttribut* pour accéder aux noms de fichiers.



## Chapitre 6

# Implémentation

### 6.1 Blending sans canaux alpha

La première implémentation consiste à modifier le programme de telle sorte qu'il puisse traiter des ortho-images quand bien même elles n'auraient pas de canaux alpha. Un canal alpha opaque est créé lorsqu'aucun n'est présent. Le message d'avertissement indiquant qu'aucun canal alpha n'est présent est d'abord désactivé dans la partie principale du programme, i.e. dans `enblend.cc`. La création du masque s'opère au plus bas niveau possible dans le programme, c-à-d au moment où le logiciel importe les images (fonction `import` dans `assemble.h`). Elle s'effectue si aucun canal alpha n'est présent :

```
if (info.numExtraBands() == 0) {
    importImage(info, image);
    cout << "Alpha channel is missing."
          << "Creating white alpha channel... ";
    initImage(alpha.first,
              alpha.first + Diff2D(info.width(), info.height()),
              alpha.second,
              NumericTraits<AlphaPixelType>::max());
    cout << "Alpha channel created." << endl;
} else {
    // code original qui importe l'image
    // et son canal alpha. (vigra::imageImportAlpha)
}
```

À noter que les modifications effectuées et l'ensemble des classes implémentées suivent les mêmes règles typographiques qu'*enblend* et en particulier sont écrites en anglais. Le commentaire présent ici dans la boucle *else* est juste une indication du rapport, mais n'est pas présent dans le code.

## 6.2 Géoréférencement

Les ortho-images sont positionnées dans un système national de référence de coordonnées géographiques. Au niveau informatique, un fichier de géoréférencement est associé à chaque fichier image et contient les informations nécessaires pour la localiser. De manière à pouvoir lire et utiliser ces informations, le portage sous linux de la bibliothèque *CGeoref* a été effectué. La directive `#ifndef linux` a été utilisée pour rendre conditionnelles les instructions spécifiques aux autres systèmes d'exploitations et les remplacer par des équivalents linux.

### 6.2.1 Utilisation

L'option `-p` a été ajoutée au programme et permet d'indiquer que les positions des images doivent être déterminées à partir des fichiers de géoréférencement associés. Les coordonnées cartographiques (axe des ordonnées orienté vers le Nord) sont alors transformées en coordonnées image (axe des ordonnées orienté vers le bas) avec la méthode *TerrainGraphe* de la bibliothèque *CGeoRef*.

## 6.3 Fichiers

Pour ne pas modifier les fichiers d'entrée, une copie est effectuée avec la bibliothèque *filesystem* de *Boost*. Des exceptions sont levées en cas d'erreur lors de la copie. La bibliothèque *Boost* était déjà utilisée dans *enblend* et a donc été réutilisée. Son avantage est de pouvoir utiliser des systèmes de fichiers issus de diverses plateformes. En particulier, le fichier de graphe *.gpo* contient les noms des fichiers images intervenant dans la mosaïque. Or ces noms ont été enregistrés avec leur chemin sur une autre plateforme. La bibliothèque *Boost* permet de retrouver les noms de fichiers avec des méthodes indépendantes de la plateforme utilisée.

## 6.4 Construction du masque

### 6.4.1 Description

La principale implémentation consiste à charger la ligne de mosaïquage telle qu'elle est générée par le programme *Mosaïque* puis à l'utiliser dans le logiciel *enblend*. La ligne de mosaïquage est d'abord sauvegardée dans un fichier de graphe de mosaïquage, portant l'extension *.gpo*. Ce fichier est ensuite lu et utilisé dans la version modifiée d'*enblend*. Une fois la ligne de mosaïquage importée, elle est utilisée pour générer le masque binaire qu'*enblend* utilise pour effectuer le blending.

### 6.4.2 Modifications

Pour charger la ligne de mosaïquage, l'option `--load-mask` d'*enblend* est utilisée. Cette option est initialement prévue pour charger directement le masque à partir d'une image. C'est pourquoi le chargement de l'image est désactivé dans le cas où le masque est un fichier *.gpo*. (cf. fichier *mask.h*)

### Ligne de mosaïquage

Ensuite, la ligne de mosaïquage générée par *Mosaïque* doit être utilisée pour modifier celle générée par *enblend* :

- les segments de la ligne de mosaïquage d' *enblend* situés en dehors de la zone de recouvrement sont conservés, et ceux de la zone de recouvrement — correspondant au contour actif d' *enblend* — sont remplacés. Cette modification s'opère après que le logiciel a créé la ligne de mosaïquage en utilisant le diagramme de Voronoï. En effet, il est inutile de laisser *enblend* déplacer le contour actif puisqu'il va être remplacé par la ligne de *Mosaïque*.
- La valeur des booléens associés à la ligne — indiquant si le contour est actif ou non — est positionnée à *false*, car cette ligne ne doit pas être déplacée.

Pour effectuer cette modification, les coordonnées de *Mosaïque* initialement exprimées dans le repère du graphe sont passées dans le repère utilisé par *enblend*, dont l'origine est située au coin haut gauche de l'union de l'emprise des images en cours de traitement.

Toutefois, lorsque le mosaïquage s'effectue sur plus de deux fichiers, une partie des lignes de raccord entre paires d'images peut être manquante dans le graphe de mosaïquage final. En effet, supposons qu'un graphe contienne déjà deux ortho-images au moins, et qu'une nouvelle ortho-image soit insérée. La ligne précédemment construite est alors modifiée pour tenir compte de la dernière ortho-image insérée. Or l'information qui avait servi à combiner les premières ortho-images n'est pas gardée dans le fichier de graphe.

C'est pourquoi, de manière à compléter la ligne de mosaïquage entre le dernier point de la ligne d' *enblend* et le premier point de la ligne de *Mosaïque* de part et d'autre de la zone de recouvrement, un contour actif est ajouté. i.e. deux segments sont ajoutés pour relier la ligne d' *enblend* à celle de *Mosaïque*, et les booléens qui leur sont associés sont positionnés à *true*.

## 6.5 Sauvegarde en plusieurs fichiers

Le blending effectué par *enblend* est initialement conçu pour sauvegarder le résultat sous forme d'un panorama ou d'une mosaïque finale. Cependant dans le cas du blending des ortho-images provenant du territoire national, le nombre d'images et leurs dimensions rendent le stockage du résultat final en une seule image inadapté. C'est pourquoi l'option *-d* a été ajoutée au programme et permet de découper l'image finale, i.e. de sauvegarder en *n* images les *n* images d'entrée. Pour ce faire, les images d'entrée sont d'abord copiées dans un répertoire. Le blending est ensuite effectué, et enfin, au moment de la sauvegarde, les emprises des images sous forme de *Rect2D* sont utilisées pour sauvegarder le résultat en plusieurs fichiers.

# Chapitre 7

## Tests

### 7.1 Tests de non-régression

Tout au long du développement du programme, les fonctionnalités initiales d' *enblend* doivent être préservées. C'est pourquoi les résultats donnés par le logiciel *enblend* version 3.0 et ceux donnés par la version modifiée du logiciel sont comparés. Un ensemble de scripts a été écrit et lancé plusieurs fois pour vérifier que les deux versions — modifiée et non modifiée — généraient les mêmes résultats. Lorsqu'un canal alpha est ajouté à l'image, la lettre A est ajoutée à la fin du nom de fichier, et lorsque les tags renseignant la position de l'image sont spécifiés, la lettre P y est ajoutée.

#### 7.1.1 Fonctionnement avec 2 fichiers

Le premier test est effectué sur deux ortho-images contenant un canal alpha, i.e. *1317AP.tif* et *1332AP.tif*. Les positions des images ont été déterminées à partir du géo-référencement et, pour ce test, ajoutées aux fichiers avec la commande *tiffset* :

```
tiffset -s <numero du tag> <valeur> <image tif>
```

Comme le décrivent les spécifications du format *TIFF* [1], les tags indicés 286 et 287 correspondent respectivement aux positions selon l'axe des abscisses et des ordonnées. Les résolutions horizontales et verticales sont spécifiées avec les tags 282 et 283 et l'unité de résolution — 2 par défaut pour le pouce, 3 pour le centimètre — avec le tag 296.

Une fois les images positionnées, les deux blendings sont effectuées et les images résultantes comparées avec la commande *diff* :

*non-regression-2.sh*

```
~/enblend-3.0/src/enblend 1317AP.tif 1332AP.tif \  
-o enb3-1317AP-1332AP.tif  
  
~/enblend-3.0-mod/src/enblend 1317AP.tif 1332AP.tif \  
-o enb3-mod-1317AP-1332AP.tif  
  
diff -s enb3-1317AP-1332AP.tif enb3-mod-1317AP-1332AP.tif
```

Pour chaque blending, l’affichage dans le terminal est le suivant :

```
Loading next image: src/1317AP2.tif
Loading next image: src/1332AP2.tif
Creating blend mask: 1/4 2/4 3/4 4/4
Optimizing 1 distinct seam.
Strategy 1, s0: 1/4 2/4 3/4 4/4
Strategy 2: s0
Using 8 blending levels
Generating Gaussian pyramid: g0 g1 g2 g3 g4 g5 g6 g7
Generating Gaussian pyramid: g0 g1 g2 g3 g4 g5 g6 g7
Generating Laplacian pyramid: 10 11 12 13 14 15 16 17
Generating Gaussian pyramid: g0 g1 g2 g3 g4 g5 g6 g7
Generating Laplacian pyramid: 10 11 12 13 14 15 16 17
Blending layers: 10 11 12 13 14 15 16 17
Collapsing Laplacian pyramid: 17 16 15 14 13 12 11 10
Writing final output...
```

Puis l’affichage dans le terminal du résultat de la commande *diff* :

```
Les fichiers enb3-1317AP-1332AP.tif
et enb3-mod-1317AP-1332AP.tif sont identiques.
```

### 7.1.2 Fonctionnement avec n fichiers

Le deuxième test est similaire au précédent. Il est réalisé avec 3 images d’entrée, afin de vérifier que les itérations sur les fichiers suivants fonctionnent comme dans le logiciel original. (cf. section *algorithme* page 9). De la même manière que précédemment les images de sortie sont comparées avec la commande *diff* :

## 7.2 Test d’intégration

La première fonctionnalité implémentée consistait à intégrer des canaux alpha opaques lorsqu’aucun n’était présent dans les images source. De manière à tester cette fonctionnalité, deux blendings sont effectués :

- d’une part le blending avec la version modifiée du logiciel sur des images source sans canaux alpha.
- d’autre part le blending avec le logiciel original sur les mêmes images auxquelles des canaux alpha opaques ont été ajoutés.

Les images obtenues sont ensuite comparées avec la commande *diff*.

*integration.sh*

```
~/enblend-3.0-mod/src/enblend 1151P.tif 3688P.tif \
-o enb3-mod-1151P-3688P.tif

~/enblend-3.0/src/enblend 1151AP.tif 3688AP.tif \
-o enb3-1151AP-3688AP.tif

diff -s enb3-mod-1151P-3688P.tif enb3-1151AP-3688AP.tif
```

Les deux fichiers sont identiques et les messages générés avec la version modifiée du logiciel sont ceux du logiciel original auxquels ont été ajoutés les messages de génération des canaux alpha.

### 7.3 Test comparatif

Pour visualiser les effets du blending sur des cas typiques de différences radiométriques présentes sur les ortho-images, un répertoire *comparatif* a été créé, dans lequel a été placé pour chaque test, un répertoire :

- *src* contenant les images source.
- *mosaique* contenant le résultat du mosaïquage effectué par *Mosaïque*.
- *enblend* le résultat du blending effectué par *enblend*.
- *enblend+mosaique* contenant le résultat obtenu par la version modifiée d' *enblend* à laquelle a été intégrée la ligne de mosaïquage de *Mosaïque*.
- *smartblend* contenant pour quelques cas, le résultat du blending effectué par *Smartblend*.



(a)



(b)

- (a) mosaïquage de 2 ortho-images prises à des dates différentes (échelle 2/5)  
 (b) mosaïquage des 2 mêmes ortho-images avec blending (échelle 2/5).



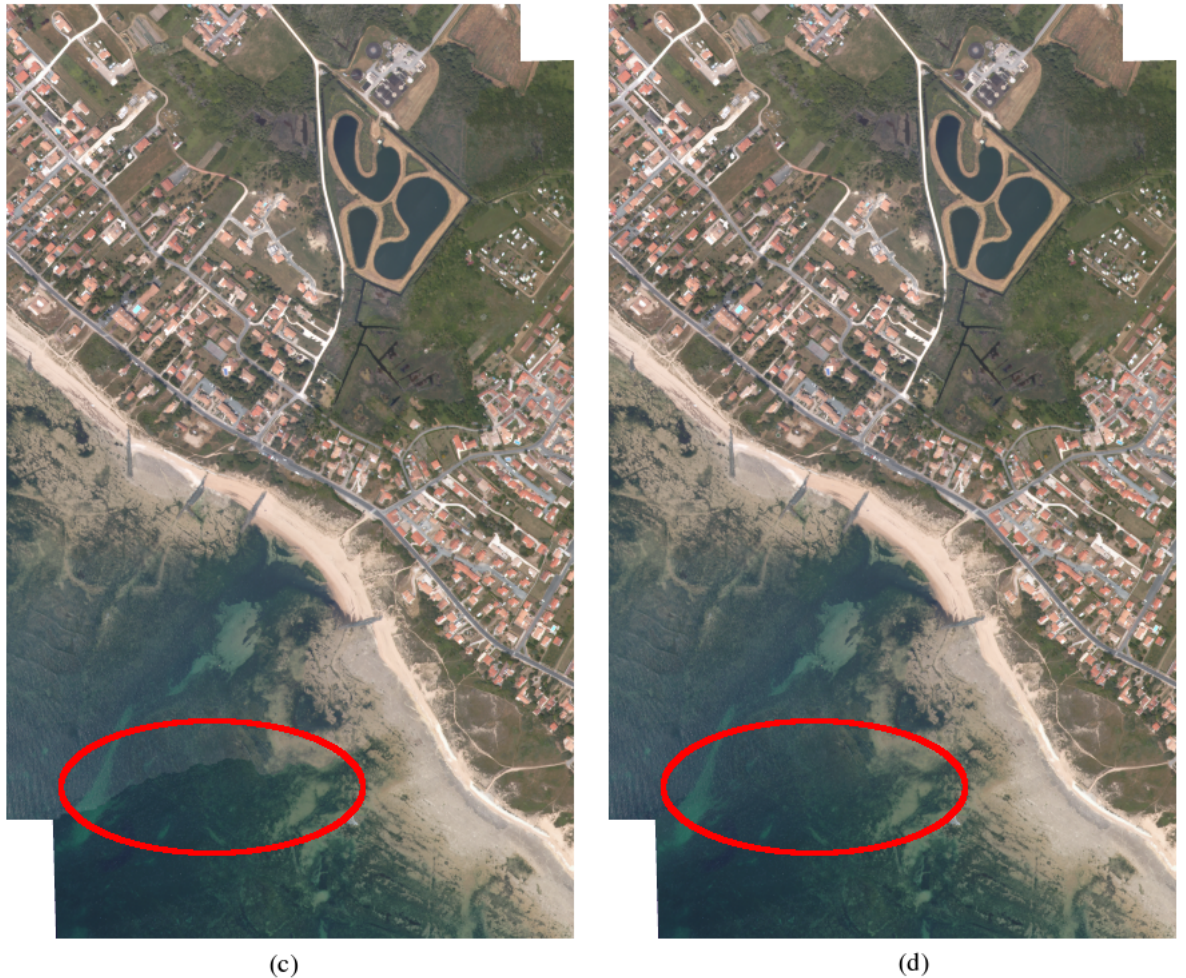


FIG. 7.1 – Test comparatif

(c) mosaïquage de 2 ortho-images en zone côtière (échelle 1/10)

(d) mosaïquage des 2 mêmes ortho-images avec blending (échelle 1/10)

## 7.4 Tests systèmes

### 7.4.1 Ligne de mosaïquage

Pour vérifier la cohérence de la ligne de mosaïquage obtenue après modification, l'option `-r` a été ajoutée au programme pour la colorer en rouge. La ligne est tout d'abord sauvegardée dans une liste de points à l'aide de la méthode `saveSeamContour` de la classe `Seamline`, puis *rasterisée* avec la bibliothèque `libXMI` [8] — initialement utilisée dans *enblend* — et copiée dans l'image finale. Les deux dernières étapes sont effectuées au moment de la sauvegarde de l'image, dans `assemble.h`.



### 7.4.2 Cas réel

De manière à vérifier la fiabilité globale du programme, une opération de blending a été effectuée sur la commune de la Roche-sur-Yon, soit un ensemble de 641 ortho-images. Les positions des ortho-images sont déterminées à partir des fichiers de géoréférencement qui leur sont associés et le graphe de mosaïquage est chargé avec l'option `--load-mask`.

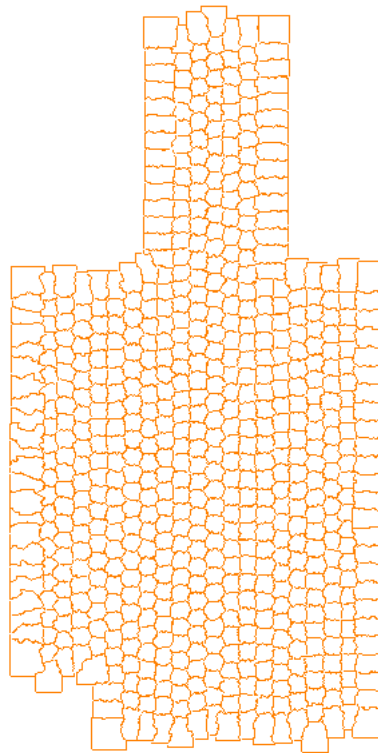


FIG. 7.2 – graphe de mosaïquage de la Roche-sur-Yon

#### Temps de traitement

Le graphe du mosaïquage contient 1920 arcs, dont 1807 qui délimitent des frontières entre paires d'images, et où le blending doit avoir lieu. L'ensemble du traitement s'est déroulé en un peu moins de 6h30 sur une machine bi-processeur ®Intel Pentium 4, soit en moyenne un peu moins de 13s par blending. Le même test sans l'option `--fine-mask` s'est déroulé approximativement dans le même intervalle de temps, et il est donc préférable d'activer cette option, i.e. de ne pas calculer le masque à une résolution dégradée.

## Chapitre 8

# Bilan

L'intégration de la ligne de mosaïquage de *Mosaïque* a été réalisée, non sans encombres, car nombre de détails techniques étaient à prendre en compte, mais finalement avec succès. Toutefois, même si le blending est adapté à des corrections radiométriques telles que celles présentées figure 7.1, il n'est pas destiné à corriger des défauts géométriques tels que des cisaillements de réseaux ou des artefacts tels que des coulées de pixels.

Le programme automatise une partie des corrections à effectuer sur les ortho-images, et les modifications qu'il réalise peuvent encore être améliorées :

- les segments ajoutés pour compléter les morceaux de ligne de mosaïquage entre paires d'images qui n'ont pas été gardés dans le graphe final peuvent être calculés en utilisant l'algorithme de *Mosaïque*. Pour le moment un segment a été ajouté. — segment qui est considéré par *enblend* comme un contour actif lorsque l'option `--fine-mask` est activée —.
- pour améliorer la qualité visuelle au niveau de la ligne de mosaïquage, la méthode de blending multi-résolution peut être implémentée dans le domaine des gradients. Le modèle présenté dans « Seamless image stitching in the gradient domain »[4] est aussi à considérer, à condition qu'il ne cause pas de modifications géométriques sur l'ortho-image finale.

Finalement, même si le blending n'est pas directement lié aux phénomènes physiques observés, il contribue à la mise en place d'une base de données d'ortho-images complémentaire à la base de données initiale, qui peut être utilisée dans les applications destinées à les visualiser.

## Annexe A

# Manuel utilisateur

L'utilisation du logiciel ne nécessite pas d'interface graphique, l'ensemble des fonctionnalités est accessible par ligne de commande. Cette liste a été complétée avec les nouvelles commandes implémentées. Les commandes principales sont décrites ici :

### A.1 Commandes

La commande générique pour utiliser le programme est la suivante :

```
enblend [options] <entrée> -o <sortie>
```

L'ensemble des images sur lesquelles doit opérer le blending multi-résolution doit être spécifié en entrée et le nom de l'image de sortie donné en paramètre après l'option -o. Une liste d'options permet de paramétrer le fonctionnement du blending, liste accessible avec la commande d'aide linux usuelle :

```
enblend -h
```

#### A.1.1 Commande supplémentaire

Une commande a été ajoutée et permet de sauvegarder le résultat du traitement des *n* images d'entrées en *n* images de sortie. Dans ce cas, un répertoire *blend* est créé et les images de sortie y sont placées avec leurs noms originaux.

```
enblend [options] <entrée> -d
```

### A.2 Options

La liste des principales options est donnée ci-dessous :

-v	donner des détails sur les opérations effectuées.
-l<nombre>	nombre de niveaux des pyramides (de 1 à 29).
--fine-mask	ne pas utiliser d'approximation pour le masque.
--save-mask=<fichier>	enregistrer le masque dans un fichier.

### A.2.1 Options supplémentaires

Deux options ont été ajoutées : la première pour positionner les ortho-images à partir des fichiers de géoréférencements qui leur sont associés, la seconde pour visualiser la ligne de raccord utilisée lors du blending.

- p positionne les images à partir du géoréférencement.
- r colore en rouge la ligne de raccord.

# Index

<b>B</b>	
blackPair .....	13
blending multi-résolution .....	8, 12, 13
Boost .....	17, 22
<b>C</b>	
canaux alpha .....	15, 21
cas d'utilisation .....	16
CGeoref .....	20, 22
CGraphe .....	15, 20
Condor .....	14
copyInto .....	19
<b>D</b>	
diagramme de Voronoï .....	9
diff .....	24
<b>E</b>	
enblend .....	8
<b>F</b>	
Files .....	19
filesystem .....	22
<b>G</b>	
géoréférencement .....	14
gradient domain blending .....	10
graphe de mosaïquage .....	28
<b>I</b>	
iBB .....	19
Institut Géographique National .....	5
Intellectual Blending .....	10
isInside .....	19
<b>L</b>	
ligne de mosaïquage .....	17
<b>M</b>	
MATIS .....	4, 6
mBB .....	13
mosaïquage .....	7
Mosaïque .....	7
<b>N</b>	
nadir .....	7
<b>P</b>	
Point2D .....	17
pyramide gaussienne .....	9, 12
pyramide laplacienne .....	9, 12
<b>R</b>	
rawSegments .....	19
Rect2D .....	17
replaceRawSegments .....	19
retrieveCoordinates .....	19
retrieveNames .....	19
roiBB .....	13
<b>S</b>	
saveSeamContour .....	19
Seamline .....	19
Size2D .....	17
slist .....	17
Smartblend .....	9
<b>T</b>	
tiffset .....	24
<b>U</b>	
uBB .....	19
<b>V</b>	
Vigra .....	17
<b>W</b>	
whitePair .....	13

# Glossaire

## **aérotriangulation**

méthode photogrammétrique qui consiste à déterminer les orientations et positions des images aériennes à partir d'un petit nombre de points de coordonnées connues, repérés sur des clichés de photographies stéréoscopiques.

## **blending**

opération qui consiste à combiner ou mixer des éléments de telle manière que les parties constituantes soient indistinguables les unes des autres.

## **coulée de pixels**

artefact présent sur une ortho-image lorsque la direction de prise de vue dans l'image est rasante par rapport au MNT.

## **hot spot**

région dans l'alignement de l'appareil de prise de vue et du soleil. Aucune ombre n'y est visible, les rayons lumineux y sont réfléchis en direction du cliché de manière plus importante qu'ailleurs créant une zone très lumineuse et très peu contrastée, donc pauvre en information.

## **Modèle Numérique de Terrain (MNT)**

ensemble discret de valeurs numériques qui modélise le relief d'une zone géographique et permet de le représenter. Le quadrillage utilisé est généralement régulier.

## **mosaïquage**

opération qui consiste à produire une seule image à partir de l'ensemble des clichés aériens ortho-rectifiés nécessaires pour couvrir le département.

## **nadir au sol**

point au sol situé directement à la verticale d'un capteur.

## **ortho-image**

image artificielle de la surface de la Terre, qui correspond à une projection orthogonale, autrement dit telle que tous les points de mêmes coordonnées planimétriques sont projetés sur un même point dans l'ortho-image.

## **ortho-photographie**

une ortho-image est issue d'une image numérique et une ortho-photographie d'un cliché argentique scanné.

**ortho-rectification**

application à une image de traitements destinés à corriger les déformations dues au relief du terrain, à l'inclinaison de l'axe de prise de vue et à la distorsion de l'objectif.

**parallaxe**

incidence du changement de position de l'observateur sur l'observation d'un objet.

**rasteriser**

convertir une image vectorielle en image bitmap.

# Bibliographie

- [1] TIFF. Revision 6.0, Adobe Systems Inc., 1992.
- [2] BD ORTHO - Descriptif de contenu. Version 2, 2007.
- [3] A. Agarwala. Interactive digital photomontage. *ACM Transactions on Graphics*, 23(3) :294–302, 2004.
- [4] A. Levin and A. Zomet and S. Peleg and Y. Weiss. Seamless image stitching in the gradient domain, 2003.
- [5] Andrew Mihal. enblend. <http://enblend.sourceforge.net>.
- [6] Nuno Gracias, Art Gleason, Shahriar Negahdaripour, Mohammad Mahoor. Fast Image Blending using Watersheds and Graph Cuts. 2006.
- [7] Peter J. Burt and Edward H. Adelson. A Multiresolution Spline with Application to Image Mosaics. *ACM Transactions on Graphics*, 2(4) :217–236, 1983.
- [8] Robert S. Mayer. *The GNU libxmi 2-D Rasterization Library*. Free Software Foundation, [www.gnu.org/software/libxmi](http://www.gnu.org/software/libxmi), 2.0 edition, 1998-1999.