

La République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
Université Saad Dahleb Blida 1



Institut d'Aéronautique et des Études Spatiales (IAES)
Département Études Spatiales

Mémoire de fin d'études

En vue de l'obtention du diplôme de Master

en : **Aéronautique**

Option : **Télécommunications Spatiales**

THÈME

*Génération d'un Modèle Numérique de Terrain
avec les Images SAR Interférométriques*

Proposé et dirigé par :

Dr. Azmedroub Boussad

Dr. Tahraoui Sofiane

Réalisé par :

M. Ouradi Sidali

M. Sahi Bachir

*Soutenue le ... **Septembre 2020**, devant le jury composé de :*

Présidente

M. Azmedroub Boussad MCB IAES *Encadreur*

M. Tahraoui Sofiane MAB IAES *Encadreur*

Examineur

Promotion : 2019/2020

Table des matières

Table des matières

Table des figures	ii
1 Télédétection	3
1.1 Introduction	3
1.2 Capteurs	3
1.2.1 Formes d'acquisition d'information avec capteurs :	4
1.2.2 Type de Capteurs :	4
1.3 Télédétection	6
1.3.1 Avantages :	6
1.3.2 Inconvénients :	6
1.3.3 Région spectrale utilisée pour l'observation de la terre :	7
1.3.4 Étapes de télédétection :	7
1.4 Radar :	8
1.4.1 Bandes de fréquences des Radars	9
1.4.2 Fonctions principales de radar :	9
1.4.3 Pourquoi utiliser le radar comme outil de télédétection ?	9
1.5 Le Radar à Synthèse d'Ouverture	11
1.6 Satellite ERS de ESA(Europeen Space agency)	12
1.6.1 Fonctionnement du sar	13
1.6.2 Qu'est-ce qu'une image SAR complexe?	14
1.6.3 Amplitude	15
1.6.4 Phase	16
1.6.5 Résolution des images Phase	17
1.6.6 Speckles	18
1.7 Conclusion	21

2	Interférométrie SAR	22
2.1	Généralité	22
2.2	Génération d'interferogramme	23
2.2.1	Contenu de la phase interférométrique	24
2.2.2	Aplatissement de l'interférogramme	26
2.2.3	Altitude d'ambiguïté	27
2.3	Cohérence	28
2.4	Differential InSAR	29
2.5	Conclusion :	31
3	Déroulement de Phase	32
3.1	Introduction	32
3.1.1	Le problème à résoudre	33
3.2	Techniques de déroulement de phase	37
3.2.1	Unweighted Least Mean Squares method (cas ou $p=2$) :	37
3.2.2	Weighted Least Mean Squares method (Cas ou $p=2$) :	38
3.2.3	Minimum Cost Flow method (Cas ou $p=1$)	39
3.2.4	Branch-Cut method (Cas ou $p=0$)	40
3.3	Conclusion	41
4	Réalisation Logiciel	42
4.1	Introduction à la partie théorique	42
4.1.1	Introduction à Snaptool	43
4.2	notre logiciel	44
4.2.1	l'interface GUI	45
4.3	Les fonctions	47
4.3.1	Affichage de Master et Slave	47
4.3.2	Génération de l'interférogramme	50
4.3.3	Estimation de la cohérence	52
4.3.4	Déroulement de phase	55
4.3.5	Affichage 3D	59
4.3.6	Résidus	62
4.3.7	Calculs	64
4.4	Conclusion	67
	Bibliographie	70

Table des figures

1.1	Formes d'acquisition	4
1.2	Types de Scanner	5
1.3	Radar SAR	6
1.4	Étapes de la télédétection	8
1.5	Bandes de fréquences des Radars	9
1.6	Altimètre	10
1.7	Diffusomètre radar	10
1.8	Radar météorologique	11
1.9	Première image SAR acquise en 1957	11
1.10	Visé du SAR	12
1.11	ERS de ESA	12
1.12	Fonctionnement du SAR	13
1.13	Direction du sar	14
1.14	Image SAR complexe	15
1.15	Image d'amplitude	16
1.16	Image de phase	17
1.17	Résolution des images Phase	18
1.18	Speckles sur image SAR	19
1.19	Image Optique	19
1.20	les réflexions multiples	20
1.21	Image après réduction des speckles	20
2.1	acquisition d'image par 2 satellites sous angle différente	22
2.2	Signal reçu	23
2.3	Génération d'interferogramme	24
2.4	paramètres géométriques	25
2.5	Aplatissement d'interferogramme	26

2.6	effet de changement de la taille de la base line	28
2.7	Generation de coherence	29
2.8	Differential InSAR	30
3.1	avant le déroulement	32
3.2	après le déroulement	33
3.3	avant et après le déroulement	33
3.4	1d unwrapping	34
3.5	avant et après le déroulement	35
3.6	logique de l'algorithme	35
3.7	Emplacement de résidus	36
4.1	SNAP	42
4.2	Python	43
4.3	avant le déroulement	44
4.4	Interface de notre logiciel	44
4.5	Bibliothèque utilisée	46
4.6	Une partie du code désigne interface	46
4.7	Code des classes de fenêtre	47
4.8	Code importation des images	48
4.9	Importation des images	48
4.10	Code Affichage Master et Slave	49
4.11	Affichage Master et Slave	49
4.12	le code de wrapping	50
4.13	Interférogramme	51
4.14	représentation de la terre plate	51
4.15	Interférogramme sans la terre plate	52
4.16	SpinBox	52
4.17	code SpinBox	53
4.18	code SpinBox	53
4.19	code de designe progressbar	53
4.20	code de fonction de progressbar	54
4.21	Fonction processEvents	54
4.22	Affichage de la cohérence	54
4.23	Code Fonction 1D	55

4.24	Code Fonction 1D	56
4.25	Unwrapping d'une ligne	56
4.26	Affichage de nombre de cycles	57
4.27	Code calcules unwrapping première colonne	57
4.28	Code pour calculer toutes les lignes	58
4.29	Code pour calculer toutes les colonnes	58
4.30	Affichage unwrapping 2D	59
4.31	Code Affichage unwrapping 3D avec hauteur	60
4.32	Affichage unwrapping 3D avec hauteur	61
4.33	hauteur en utilisant la toolbar	62
4.34	code pour rechercher des résidus	62
4.35	code pour afficher les résidus	63
4.36	code pour afficher les résidus	63
4.37	code pour afficher les résidus	64
4.38	code pour rechercher la hauteur d'un point	65
4.39	Calcul de Hauteur	65
4.40	Code de Calcul de la Hauteur maximale et minimale	66
4.41	Calcul de la Hauteur maximale et ses coordonnées	66
4.42	Calcul de la Hauteur minimale et ses coordonnées	67

Introduction Générale

Avec l'évolution des temps modernes, l'humain a de plus en plus besoin d'avoir un œil sur tout et d'observer tout ce qu'il l'entoure dans le but d'explorer des ressources, de prévenir des risques naturels ou d'exploiter au maximum la géographie comme pour l'optimisation des emplacements des antennes télécommunication et c'est là qu'interviennent les radars.

Le radar (radio détections and ranging) est un système qui utilise les ondes électromagnétiques à des fins multiples, l'une des utilisations les plus répandues est la télédétection où il va nous permettre de percevoir l'existence, la forme, la position et la nature d'un objet grâce au traitement de l'information de l'onde réfléchi, grâce à cela c'est devenu une technique incontournable, notamment pour la génération des modèles numériques de terrain.

Ce dernier, abrégé InSAR (ou obsolète IfSAR) est un technique radar utilisé en géodésie et en télédétection. Cette méthode géodésique utilise deux ou plusieurs images radars à synthèse d'ouverture (SAR) pour générer des cartes de déformation de surface ou d'élévation numérique, en utilisant des différences de phase des ondes revenant au satellite ou à l'aéronef.

Notre travail entre dans le domaine de la télédétection de la surface terrestre à partir de l'espace. Notre objectif principal est la génération d'un modèle numérique de terrain avec l'utilisation des images SAR. Pour la réalisation de cet objectif on doit passer par plusieurs étapes.

Premièrement il faut avoir les données nécessaires. Ces données sont des images SAR interférométriques, c'est-à-dire deux images acquises suivant deux angles différents pour la même scène terrestre. Pour une bonne exploitation de ces images il faut bien comprendre le principe des images SAR .

On présentera ces notions dans le premier chapitre. Ensuite on utilisera ces images pour la réalisation d'un Modèle Numérique de Terrain (MNT).

Le deuxième chapitre donne les bases théoriques sur les étapes de la réalisation d'un MNT (Types d'image InSAR, Génération d'interferogramme, calcul de cohérence...)

Dans le troisième chapitre on Expliquera l'une des étapes cruciale dans l'interferometrie qui est le déroulement de phase et ses techniques

Dans le quatrième chapitre, au lieu de faire juste une manipulation sur un logiciel existant, on a créé le notre pour mieux comprendre le fonctionnement et pour implémenter les techniques voulues .

A la fin en termine avec une conclusion générale qui donne une synthèse sur les résultats obtenus, ainsi que les futures perspectives de notre projet de fin d'études.

Chapitre 1

Téledétection

1.1 Introduction

La télédétection est une technique qui permet, à l'aide d'un capteur, "d'observer" et d'enregistrer le rayonnement électromagnétique, émis ou réfléchi, par une cible quelconque sans contact direct avec celle-ci. Dans notre cas le radar est notre instrument de mesure (on expliquera le but derrière ce choix).

Dans ce chapitre on va aborder des idées généralisées sur la télédétection, ses régions spectrales et ses étapes. aussi sur les systèmes radars , ses fonctions principales et pourquoi il est si crucial dans la télédétection terrestre.

Après ont approfondi plus sur le radar à synthèse d'ouverture SAR, son fonctionnement, les types d'images qu'il peut acquérir et leurs caractéristiques qui vont nous donner un avant gout sur l'InSAR (Interferometry SAR) qu'on va aborder dans le prochain chapitre

1.2 Capteurs

C'est un Système utilisé pour l'acquisition du rayonnement émis ou réfléchi , sa classification est selon la source de rayonnement :

– **capteurs passifs :**

Ils peuvent seulement percevoir l'énergie réfléchie lorsque le Soleil illumine la Terre. Ils n'illuminent pas la cible, ils peuvent seulement recevoir l'énergie émise ou dispersée par la cible.

– **Capteurs actifs :**

Ils fournissent sa propre source de rayonnement électromagnétique et mesure la réponse de la surface. Les capteurs actifs ne dépendent pas du soleil et des conditions atmosphériques, ils peuvent donc opérer 24h/24.

1.2.1 Formes d'acquisition d'information avec capteurs :

Le signal acquis par les capteurs peut provenir de plusieurs sources, alors on a différentes formes d'acquisition ;

- **Par réflexion (a)** : dans ce cas le signal part de l'émetteur et heurt un diffuseur qui va rétro-diffuser le signal vers notre capteur.
- **Par émission (b)** : dans ce cas le signal provient directement de son propre émetteur.
- **Par émission-réflexion (c)** : dans ce cas le signal est émis par notre émetteur propre, puis capté par notre capteur, et c'est le cas qui nous intéresse.

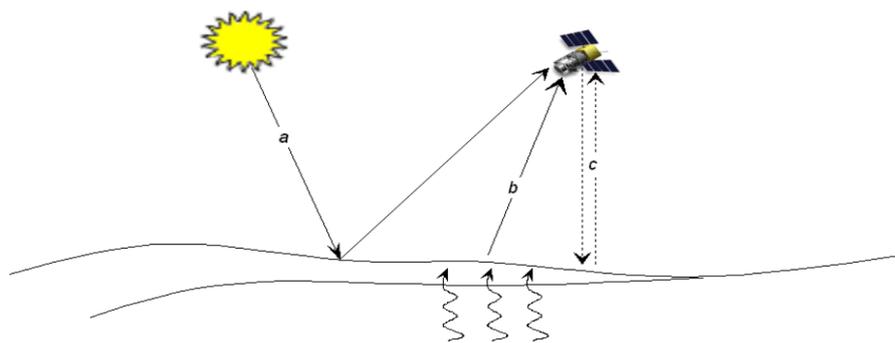


FIGURE 1.1 – *Formes d'acquisition*

1.2.2 Type de Capteurs :

Capteurs électro-optiques : Visible et IR (Technologies de Scannage)

-Un scanner, ou scanneur, aussi appelé numériseur à balayage, est un périphérique informatique qui permet de numériser des documents ou autres, comme les empreintes digitales par exemple.

- **Line scanner** : Cette technique est parfaitement adaptée à l'inspection de matériaux plats continus ou de matériaux en bande afin de détecter des défauts, elle a besoin d'une source de lumière pour procéder

- **Whiskbroom scanner** : Ce dernier veut dire en français scanneur à balai, car il a le même concept comme expliquer dans la figure suivante .il est utilisé pour la télédétection passive depuis l'espace.
- **Pushbroom scanner** :Un scanneur à balai poussoir, également connu sous le nom de scanneur le long de la piste, est un dispositif permettant d'obtenir des images avec des capteurs spectroscopiques..

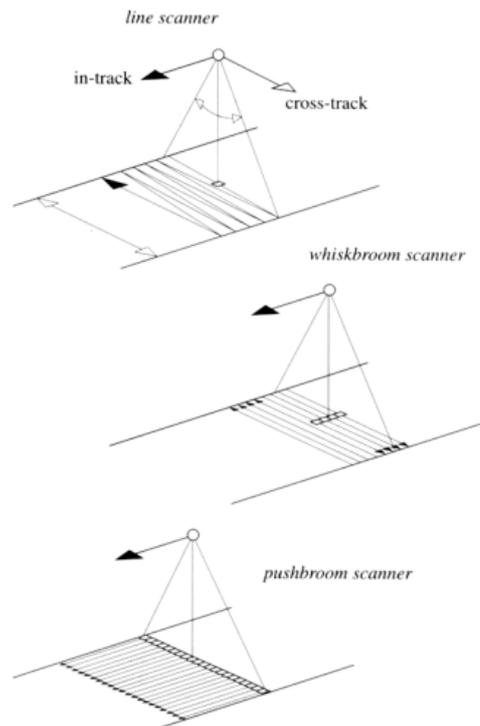


FIGURE 1.2 – Types de Scanner

Capteurs à micro-ondes

C'est une technique qui consiste à extraire les informations d'une cible, en utilisant les micro-ondes. c'est-à-dire sans entre en contact avec celle-ci..

- **Passif** :
 - a-Radiomètre a micro-ondes
- **actif** :
 - a-Sar
 - b-Altimètre Radar
 - c-Diffusiomètre a vent

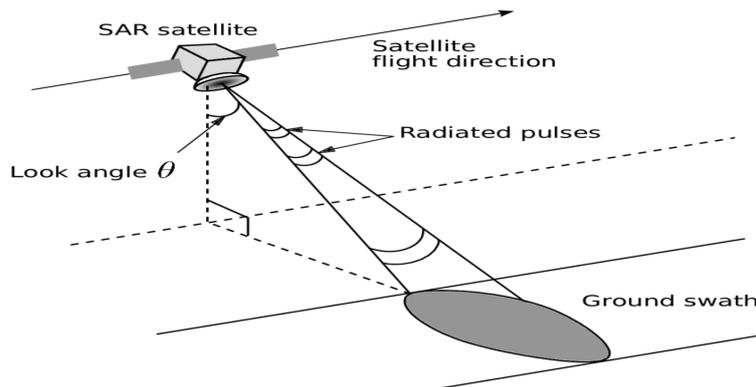


FIGURE 1.3 – Radar SAR

1.3 Télédétection

La télédétection spatiale ('Remote Sensing') est une discipline scientifique qui intègre un large éventail de compétences et de technologies utilisées pour l'observation, l'analyse et l'interprétation des phénomènes terrestres et atmosphériques. Ses principales sources sont les mesures et les images obtenues à l'aide de plates-formes aériennes et spatiales.

L'acquisition d'informations à distance implique l'existence d'un flux d'informations entre l'objet observé et le capteur. Le porteur de cette information est un rayonnement électromagnétique qui peut être émis par l'objet ou venir d'un autre organisme et avoir été réfléchi par celui-ci. Tous les corps (planètes, êtres vivants, objets inanimés) émettent un rayonnement électromagnétique, la quantité et le type de rayonnement émis, cela dépend largement de la température.

1.3.1 Avantages :

- Couverture globale et périodique sur de vastes zones de la Terre.
- Observation multi-échelle.
- Information sur les régions non visibles du spectre
- Traitement numérique des images

1.3.2 Inconvénients :

- Impossibilité d'obtenir des modèles avec grande précision en comparaison avec les observations sur site ou de photographie aérienne.

1.3.3 Région spectrale utilisée pour l'observation de la terre :

Différents rayonnements sont utilisés dans la télédétection, les plus importants sont ;

– **Spectre visible (0,4-0,7 m) :**

Gamme de fréquences de l'œil humain. Rayonnement solaire maximal. Subdivisé en trois groupes : R, G, B.

– **Infrarouge proche (0,7-1,1 m) :**

Appelé IR réfléchi. L'énergie solaire des corps réfléchissants. Le rayonnement dans la région de l'infrarouge réfléchi est utilisé en télédétection de la même façon que le rayonnement visible. .

– **Infrarouge moyen (1,1 -8 mm) :**

Mélange de rayonnement solaire et d'émission. Affecte de manière significative l'atmosphère : employé pour mesurer les concentrations de vapeur d'eau, ozone, aérosols, etc.

– **Infrarouge thermique (8-14 mm) :**

Rayonnement émis par les organismes eux mêmes. La T_p d'un corps peut être déterminée (IR thermique). Les images peuvent être disponibles à tout moment de la journée.

– **Micro-ondes (1mm-1m) :**

Intérêt croissant de la télédétection dans cette bande. Perturbations atmosphériques sont mineures et sont transparentes pour les nuages. Les capteurs actifs sont généralement utilisés.

1.3.4 Étapes de télédétection :

Les étapes énumérées dans cette section représentent en gros tout le secteur de la télédétection.

– **source d'énergie ou d'illumination (A)**

À l'origine de tout processus de télédétection se trouve nécessairement une source d'énergie pour illuminer la cible (soleil, radar, etc.).

– **Rayonnement et atmosphère (B)**

- Durant son parcours entre la source d'énergie et la cible, le rayonnement interagit avec l'atmosphère. Une seconde interaction se produit lors du trajet entre la cible et le capteur.

– **Interaction avec la cible (C) -**

Parvenue à la cible, l'énergie interagit avec la surface de celle-ci. La nature de cette interaction dépend des caractéristiques du rayonnement et des propriétés de la surface.

– **Enregistrement de l'énergie par le capteur (D)–**

L'énergie diffusée ou émise par la cible doit être captée à distance (par un capteur qui n'est pas en contact avec la cible) et être enfin enregistrée.

– **Transmission, réception et traitement (E)**

L'énergie enregistrée par le capteur est transmise, souvent par des moyens électroniques, à une station de réception où l'information est transformée en images (numériques ou photographiques).

– **Interprétation et analyse (F)**

Une interprétation visuelle et/ou numérique de l'image traitée est ensuite nécessaire pour extraire l'information que l'on désire obtenir sur la cible.

– **Application (G)**

La dernière étape du processus consiste à utiliser l'information extraite de la cible, pour nous en faire découvrir de nouveaux aspects (Carte géologique) ou pour aider à résoudre un problème

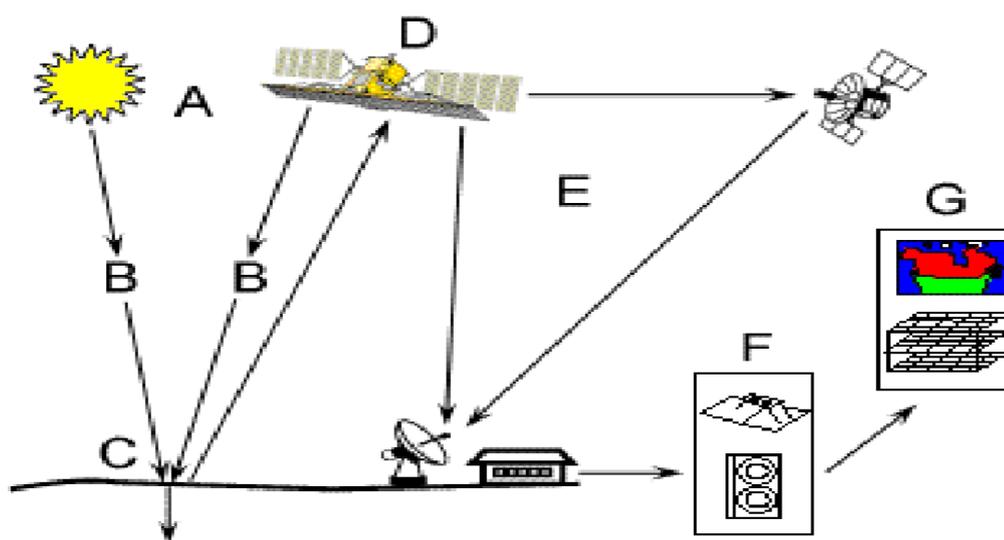


FIGURE 1.4 – Étapes de la télédétection

1.4 Radar :

Le radar (acronyme issu de l'anglais radio detection and ranging) est un système qui utilise les ondes électromagnétiques pour détecter la présence et déterminer la position ainsi que la vitesse d'objets tels que les avions, les bateaux, ou la pluie. Les ondes envoyées par l'émetteur sont réfléchies par la cible, et les signaux de retour (appelés écho radar ou écho-radar) sont captés et analysés par le récepteur, souvent situé au même endroit que l'émetteur.

1.4.1 Bandes de fréquences des Radars

Fréquence	IEEE	Types de radars
3-30 MHz	HF	Radars transhorizon, Radars spécifiques (contre avions furtifs)
30-300 MHz	VHF	
0.3-1 GHz	UHF	Radars longue portée : suivi des satellites
1-2 GHz	L	Radars longue portée : suivi du trafic aérien
2-4 GHz	S	Trafic aérien, côtier et météo
4-8 GHz	C	Surveillance aérienne, conduite de tir
8-12 GHz	X	Radars aéroportés, conduites de tir, radar d'atterrissage
12-18 GHz	Ku	
18-27 GHz	K	
27-40 GHz	Ka	
40-75 GHz	V	Auto-Directeurs de missiles
75-110 GHz	W	

FIGURE 1.5 – Bandes de fréquences des Radars

1.4.2 Fonctions principales de radar :

Plusieurs services sont mis a notre disposition , comme :

- Émettre un signal hyperfréquences (radio) vers une scène
- Capter le signal rétrodiffusé de la scène
- Enregistrer l'intensité (détection) du signal de retour et le délai entre ce dernier et le signal émis (télémétrie)

1.4.3 Pourquoi utiliser le radar comme outil de télédétection ?

Pour répondre à notre question du début d'introduction, on vous offre plusieurs points positifs qui nous poussent à choisir le radar pour la télédétection :

- Pouvoir contrôler la source d'illumination.
- Permet de voir à travers les nuages, la pluie et dans l'obscurité.
- Des Images à haute résolution (3 –10 m).
- Possibilité de représenter et de différencier certaines caractéristiques qui ne peuvent être discernées par les capteurs visibles.
- Certains aspects des surfaces ressortent mieux sur des images radar : –glace, vagues océaniques –humidité du sol, masse végétale –certains objets fabriqués par l'homme –structures géologiques

Types de radar utilisé en télédétection :

- **Altimètre** : Un altimètre est un instrument de mesure permettant de déterminer la distance verticale entre un point et une surface de référence.

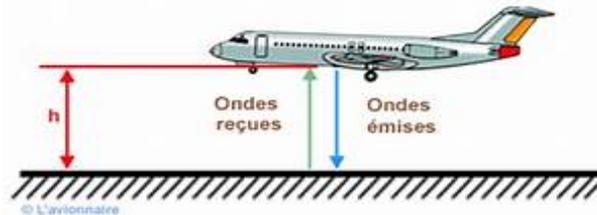


FIGURE 1.6 – Altimètre

- **Diffusomètre radar** : est un radar qui mesure la surface efficace radar de réflexion de la cible en transmettant une impulsion et en mesurant la portion d'énergie retournée.

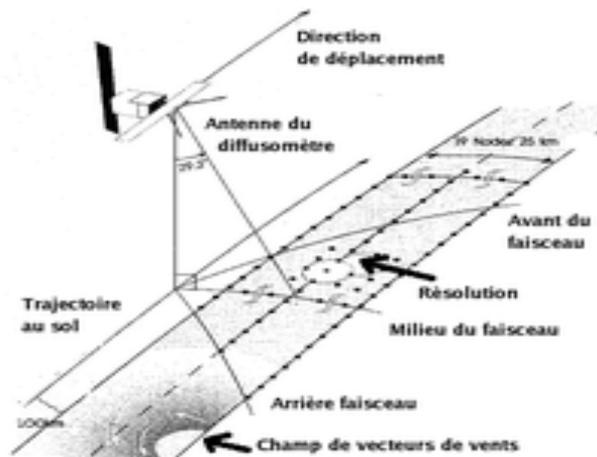


FIGURE 1.7 – Diffusomètre radar

- **Radar météorologique** : Un radar météorologique est un type de radar utilisé en météorologie pour repérer les précipitations, calculer leur déplacement et déterminer leur type (pluie, neige, grêle, etc.)

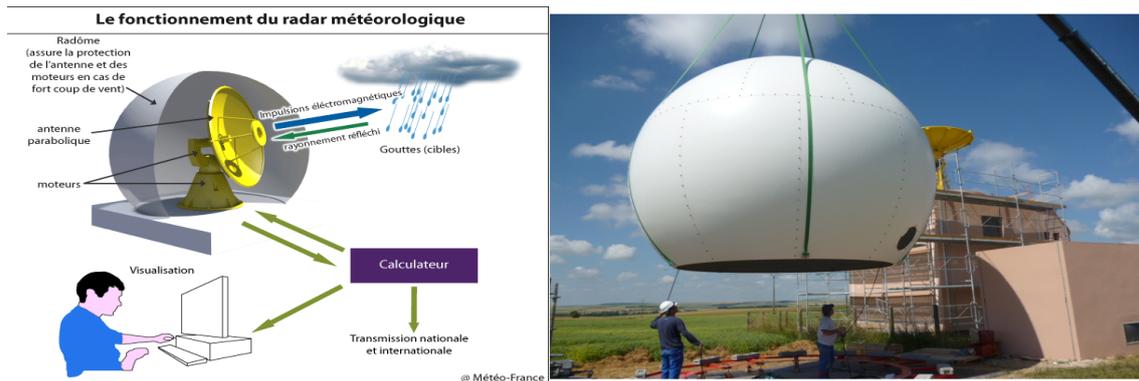


FIGURE 1.8 – Radar météorologique

- **Radar à ouverture synthétique (SAR) et Radar à ouverture réelle (RAR) :** Ce sont deux types de radars utilisés en télédétection pour acquérir diverses informations sur une zone terrestre étudiée

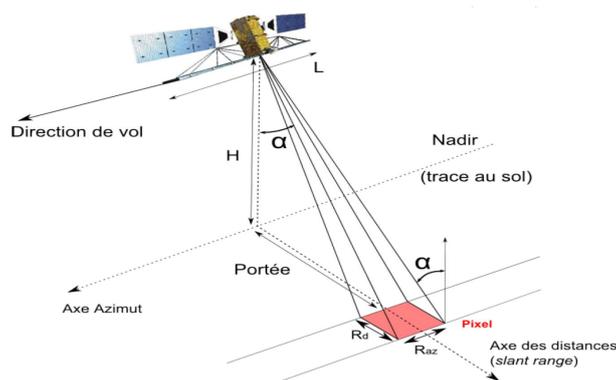


FIGURE 1.9 – Première image SAR acquise en 1957

1.5 Le Radar à Synthèse d'Ouverture

Le radar à synthèse d'ouverture (SAR) est un système d'imagerie qui a des capacités de pénétration dans les nuages, car il utilise des micro-ondes. Il est aussi opérationnel de jour comme de nuit car il s'agit d'un système actif. Enfin, sa «configuration interférométrique», SAR interférométrique ou InSAR, permet des mesures précises du trajet de rayonnement car il est cohérent.

Ce dernier est un radar à visée latérale. On dit qu'il est de type SLAR (Sideway-Looking Airborne Radar) . Cette technique permet d'éliminer l'ambiguïté rencontrée en visée verticale. En effet, cette dernière ne permet pas de discerner entre deux cibles A et B (Figure

suivante) car l'écho revient au même instant au capteur du satellite. La distance radiale étant la même, les points A et B se trouvent alors géographiquement confondus.

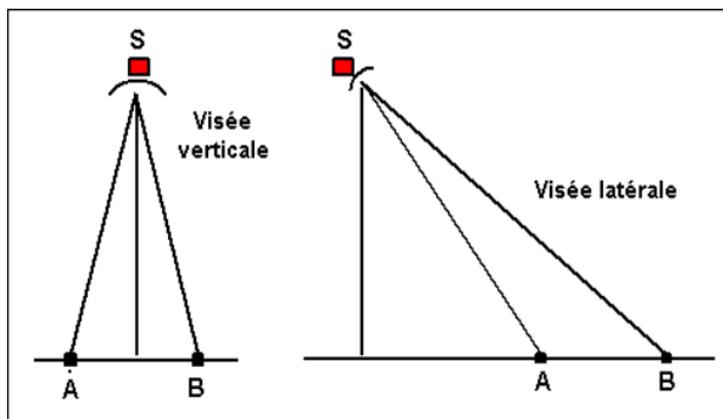


FIGURE 1.10 – Visé du SAR

1.6 Satellite ERS de ESA(Europeen Space agency)

Dans ce qui suit on donnera des exemples sur le satellite ERS alors voici un petit rappel sur ce dernier :

Le satellite européen de télédétection ERS-1, lancé en 1991, transportait une charge utile complète comprenant un radar à synthèse d'image, un altimètre radar et d'autres instruments puissants pour mesurer la température de surface de l'océan et les vents en mer. ERS-2, qui chevauchait ERS-1, a été lancé en 1995 avec un capteur supplémentaire pour la recherche sur l'ozone atmosphérique.



FIGURE 1.11 – ERS de ESA

1.6.1 Fonctionnement du sar

Un satellite porte un radar dont l'antenne est dirigée vers la surface de la Terre dans le plan perpendiculaire à l'orbite (en pratique, ce n'est pas strictement vrai, car il est nécessaire de compenser la rotation de la Terre). L'inclinaison de l'antenne par rapport au nadir est appelée l'angle off-nadir et dans les systèmes contemporains se situe généralement entre 20 degrés et 50 degrés (c'est 21 degrés pour ERS "satellite de ESA"). En raison de la courbure de la surface de la Terre, l'angle d'incidence du rayonnement sur un terrain horizontal plat est plus grand que le off-nadir (généralement 23 degrés pour l'ERS). Cependant, pour des raisons de simplicité, nous supposons ici que la Terre est plate, et donc que l'angle d'incidence est égal à l'angle off-nadir, comme le montre la figure suivante.

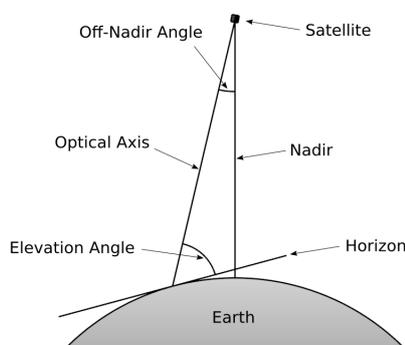


FIGURE 1.12 – Fonctionnement du SAR

Actuellement, les systèmes SAR par satellite opérationnels fonctionnent dans l'une des bandes micro-ondes suivantes :

- **Bande C - 5,3 GHz** (missions ERS et Envisat de l'ESA, Radarsat Canadien et missions de navette Américaines)
- **Bande L - 1,2 GHz** (J-ERS japonais et ALOS)
- **Bande X - 10 GHz** (le X-SAR germano-italien sur les missions navettes)

Dans le cas de l'ERS, la zone éclairée au sol (l'empreinte de l'antenne) est d'environ 5 km dans le sens longitudinal (également appelée **azimuth direction**) et environ 100 km dans la direction transversale (également appelée **ground range direction**).

La direction le long de la ligne de visée (LOS) est généralement appelée **slant-range direction**.

L'empreinte de l'antenne se déplace à la vitesse du satellite le long de son orbite. Pour l'ERS, la vitesse du satellite est d'environ 7430 m / s sur une orbite quasi polaire qui traverse l'équateur à un angle de 9 degrés et à une altitude d'environ 800 km. L'empreinte trace une bande de 100 km de large au sol à la surface de la Terre, avec la capacité d'imager une bande de 445 km de long toutes les minutes (mode carte en bande).

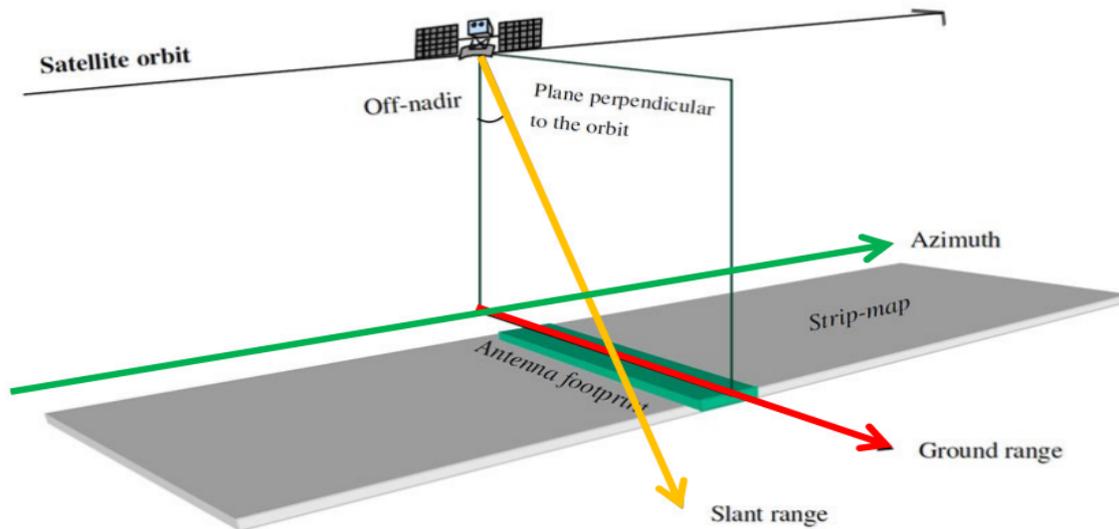


FIGURE 1.13 – Direction du sar

1.6.2 Qu'est-ce qu'une image SAR complexe ?

Une image SAR numérique peut être considérée comme une mosaïque ou une matrice (c'est-à-dire un réseau bidimensionnel formé de colonnes et de lignes) de petits éléments d'image (pixels). Chaque pixel est associé à une petite zone de la surface de la Terre (appelée cellule de résolution). Chaque pixel donne un nombre complexe qui transporte des informations d'amplitude et de phase sur le champ micro-ondes rétrodiffusé par tous les diffuseurs (roches, végétation, bâtiments, etc.) dans la cellule de résolution correspondante projetés au sol.

Différentes lignes de l'image sont associées à différents emplacements d'azimut, tandis que différentes colonnes indiquent différents emplacements de ground range. L'emplacement et la dimension de la cellule de résolution en coordonnées d'azimut et de distance inclinée ne dépendent que des caractéristiques du système SAR.

Dans le cas ERS, la dimension de la cellule de résolution SAR est d'environ 5 mètres en azimut et d'environ 9,5 mètres en slant range. La distance entre les cellules adjacentes est d'environ 4 mètres en azimut et d'environ 8 mètres en ground range. Les cellules de résolution SAR se chevauchent donc légèrement à la fois en azimut et en plage oblique.

L'image u est un signal complexe formé d'une amplitude A et d'une phase θ

$$u = A \exp(j\phi) = A \cos(\phi) + jA \sin(\phi) \quad (1.1)$$

- En imagerie visible, on utilise principalement l'amplitude

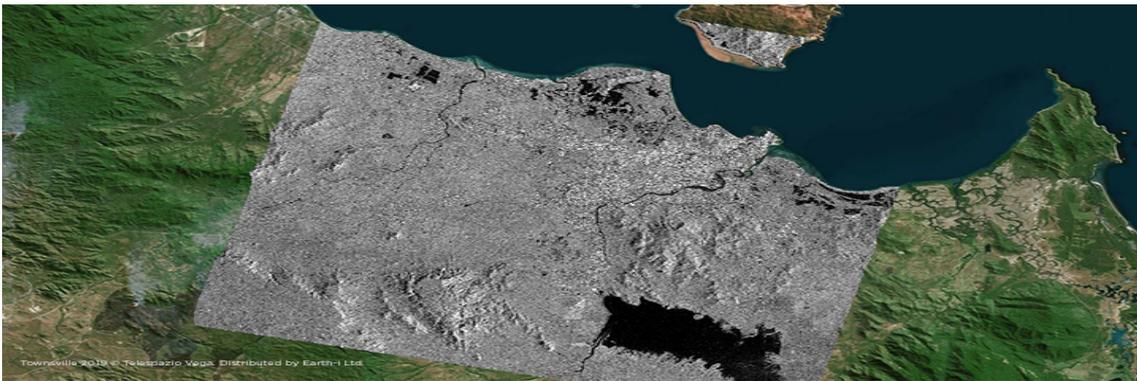


FIGURE 1.14 – Image SAR complexe

- En SAR, on utilise l'amplitude pour une série d'applications, mais la phase peut s'avérer extrêmement utile (cf. interférométrie SAR)

1.6.3 Amplitude

Cette amplitude dépend plus de la rugosité que de la composition chimique des diffuseurs sur le terrain. En règle générale, les roches exposées et les zones urbaines présentent de fortes amplitudes, tandis que les surfaces planes lisses (comme les bassins d'eau calmes) présentent de faibles amplitudes, car le rayonnement est principalement réfléchi loin du radar. Les pixels clairs correspondent à des zones de fort rayonnement rétrodiffusé (par exemple les zones urbaines), tandis que les pixels sombres correspondent à un faible rayonnement rétrodiffusé (par exemple, un bassin d'eau calme).

Amplitude peut être transformée en une unité physique ou radar backscatterer / rétrodiffusion

L'amplitude alors dépend de :

1- propriétés du signal :

- Λ , longueur d'onde
- Polarisation
- θ , angle incidence

2- propriétés de la surface diffusante

- Rugosité
- Forme
- Propriétés diélectriques

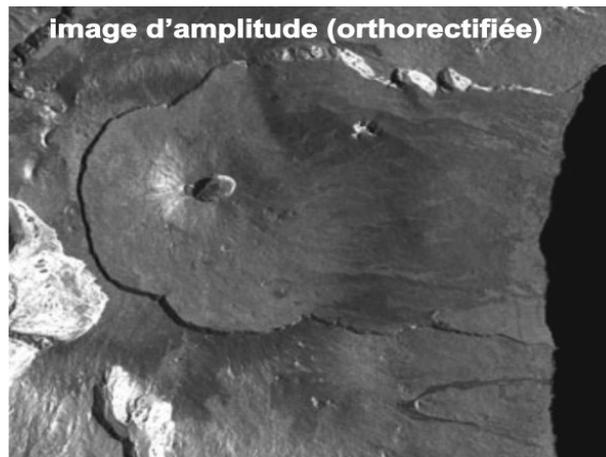


FIGURE 1.15 – Image d'amplitude

1.6.4 Phase

Le rayonnement transmis par le radar doit atteindre les cibles au sol puis revenir au radar afin de former l'image SAR (bidirectionnelle). Des cibles à différentes distances du radar (différentes slant ranges) introduisent des retards différents entre la transmission et la réception du rayonnement.

Cependant, en raison de la nature périodique du signal, des distances de déplacement qui diffèrent par un multiple entier de la longueur d'onde introduisent exactement le même changement de phase. En d'autres termes, la phase du signal SAR est une mesure de seulement la dernière fraction de la distance de déplacement bidirectionnelle qui est plus petite que la longueur d'onde transmise. En pratique, en raison de l'énorme rapport entre la dimension de la cellule de résolution (de l'ordre de quelques mètres) et la longueur d'onde (5,6 cm pour l'ERS), le changement de phase passant d'un pixel à l'autre au sein d'une seule image SAR semble aléatoire et est aléatoire. d'aucune utilité pratique.

Chaque pixel correspond à une valeur de phase comprise entre 0 et 2π , les images de phase ressemblent à du bruit, et ne sont pas exploitables en tant que telles.

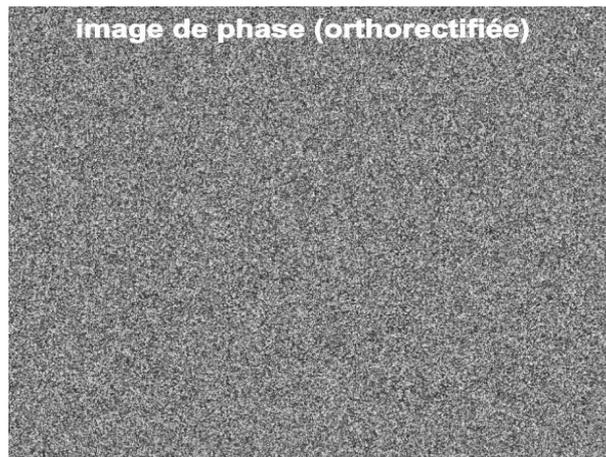


FIGURE 1.16 – Image de phase

1.6.5 Résolution des images Phase

La zone de terrain imagée dans chaque cellule de résolution SAR (appelée cellule de résolution au sol) dépend de la topographie locale. Elle dépend fortement de la pente du terrain dans le plan perpendiculaire à l'orbite (direction de la ground range) et de la pente du terrain dans la direction d'azimut. La dimension de la cellule de résolution au sol en azimut est liée à celle de la cellule de résolution SAR par la déformation en perspective habituelle que nous rencontrons chaque jour en regardant les surfaces sous différents angles (par exemple, une carte postale vue à 90 degrés est une ligne). La dimension de la cellule de résolution au sol à porter est liée à celle de la cellule de résolution SAR par une déformation en perspective inhabituelle. La figure 1.17 montre comment la portée oblique est projetée sur le sol.

À mesure que la pente du terrain augmente par rapport à une surface horizontale plane (c'est-à-dire à mesure que la normale au terrain se déplace vers la ligne de visée (LOS)), la dimension de la cellule de résolution au sol dans la plage augmente. Cet effet est appelé raccourcissement. Lorsque la pente du terrain est proche de l'angle radar hors nadir, la dimension de la cellule devient très grande et tous les détails sont perdus. De plus, lorsque la pente du terrain dépasse l'angle radar hors nadir, les diffuseurs sont imagés dans l'ordre inverse et superposés à la contribution provenant d'autres zones. Cet effet, appelé escale .

En revanche, lorsque la pente du terrain diminue par rapport à la surface de référence horizontale plate, la dimension de la cellule de résolution diminue. La dimension de cellule de résolution minimale (c'est-à-dire égale à la résolution de la plage oblique) est atteinte lorsque le terrain est parallèle au LOS. C'est également la limite inférieure de pente qui peut être imagée par un système SAR, car au-delà de cet angle, le terrain est dans l'ombre.

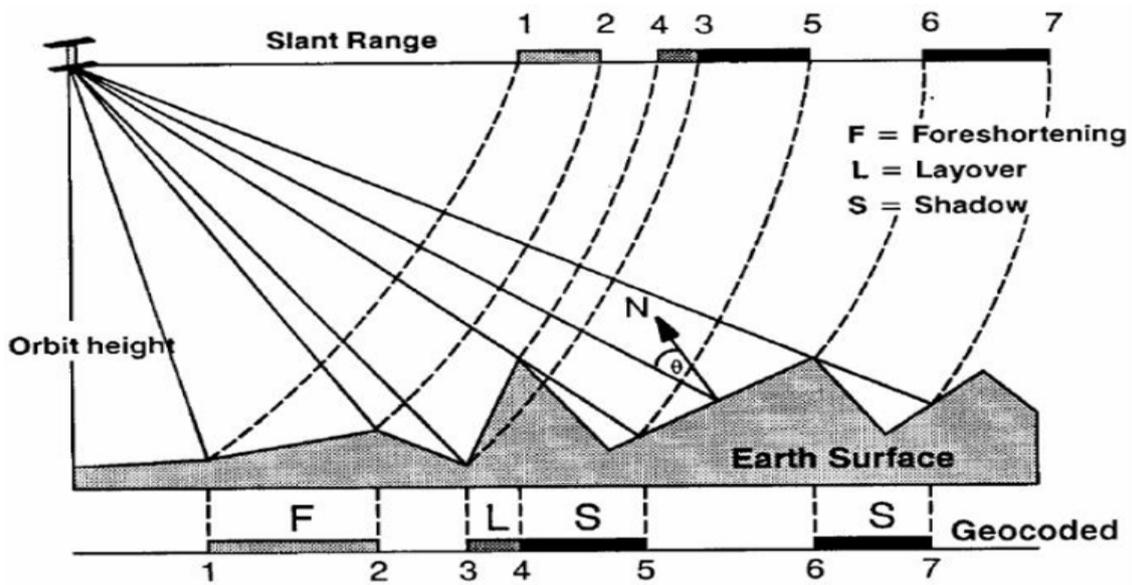
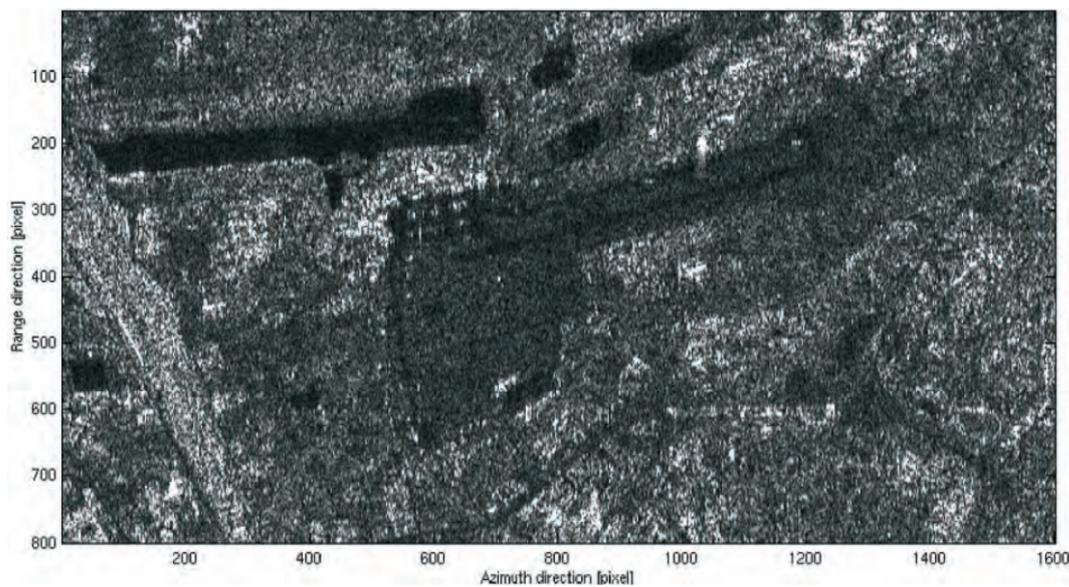


FIGURE 1.17 – Résolution des images Phase

1.6.6 Speckles

La présence de plusieurs diffuseurs dans chaque cellule de résolution SAR génère l'effet dit «speckle» commun à toute système d'imagerie cohérent. Le speckle est présent dans le SAR, mais pas dans les images optiques.

Les Zones de terrain homogènes qui s'étendent sur de nombreuses cellules de résolution SAR (imaginez, par exemple, un grand champ agricole couvert par un type de culture) sont imagées avec différentes amplitudes dans différentes résolutions cellules. L'effet visuel est une sorte d'écran «sel et poivre» superposé à une image d'amplitude uniforme. voici un exemple :

FIGURE 1.18 – *Speckles sur image SAR*

on peut remarquer la différence en utilisant une image optique comme dessus :

FIGURE 1.19 – *Image Optique*

Cet effet de speckle est une conséquence directe de la superposition des signaux réfléchés par de nombreux petits diffuseurs élémentaires (ceux de dimension comparable à la longueur d'onde radar) dans la cellule de résolution. ces signaux, qui ont une phase aléatoire en raison de réflexions multiples entre diffuseurs s'ajoutent au rayonnement directement réfléchi, comme montré sur la figure suivante :

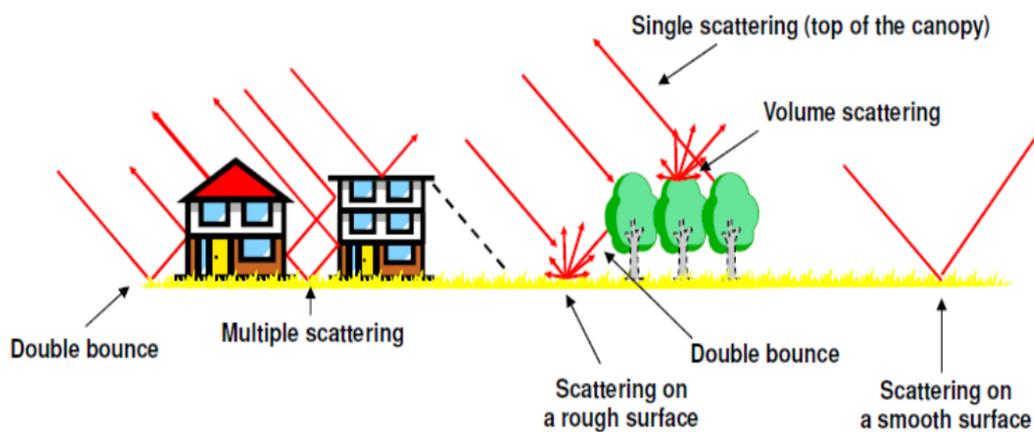


FIGURE 1.20 – les réflexions multiples

Speckle a un impact sur la qualité et l'utilité de l'image SAR détecté. En règle générale, la segmentation de l'image souffre gravement du speckle. Cependant, en prenant plus d'images de la même zone à des moments différents ou angles de vue légèrement différentes, ce dernier peut être considérablement réduit

plusieurs images ont tendance à annuler la variabilité aléatoire d'amplitude et laissez le niveau d'amplitude uniforme inchangé . voici la même image ou on a utilisé plusieurs acquisitions et réduit nos speckles :

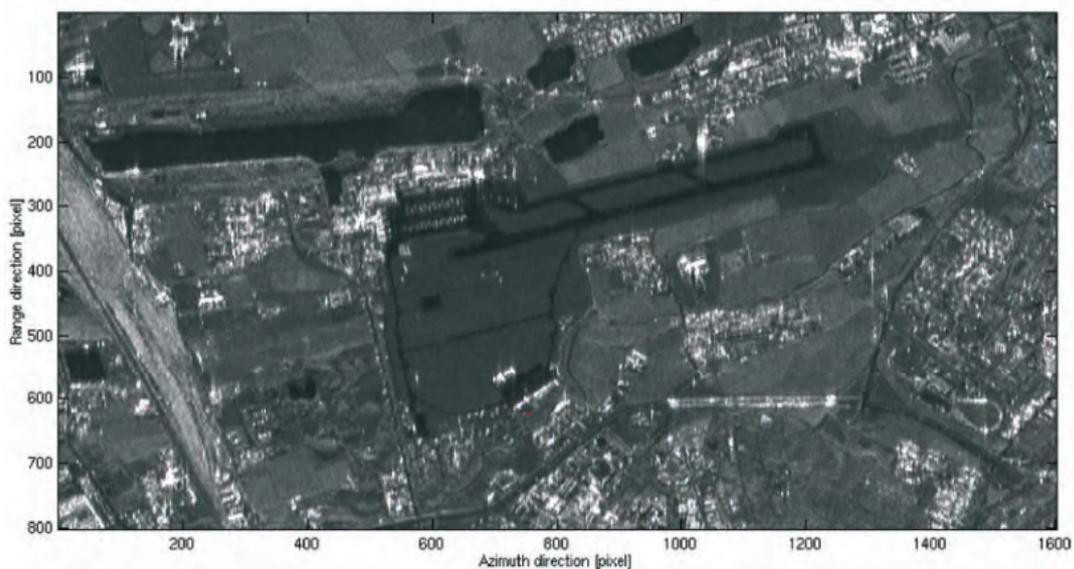


FIGURE 1.21 – Image après réduction des speckles

1.7 Conclusion

Dans ce chapitre on a présenté les bases de la télédétection, ses avantages et ses inconvénients, aussi ses instruments comme les capteurs, les radars et leur mode d'emploi et plus précisément le radar à synthèse d'ouverture, les types d'images qui peut acquérir, la résolution de ses prises, les speckles qui génère et l'avantage de son utilisation pour nous préparer à ce qui suit dans le chapitre suivant, c'est-à-dire Interferometrie SAR.

Chapitre 2

Interférométrie SAR

2.1 Généralité

Un satellite SAR peut observer la même zone sous des angles de vue légèrement différents. Cela peut se faire soit simultanément (avec deux radars montés sur la même plateforme) soit à des moments différents en exploitant des orbites répétées du même satellite. Alors on aura 2 signaux à étudier :

$$S1 = A1 \exp(j\phi1) \quad (2.1)$$

$$S2 = A2 \exp(j\phi2) \quad (2.2)$$

La distance entre les deux satellites (ou orbites) dans le plan perpendiculaire à l'orbite est appelée ligne de base de l'interféromètre et sa projection perpendiculaire à la ground range est la ligne de base perpendiculaire

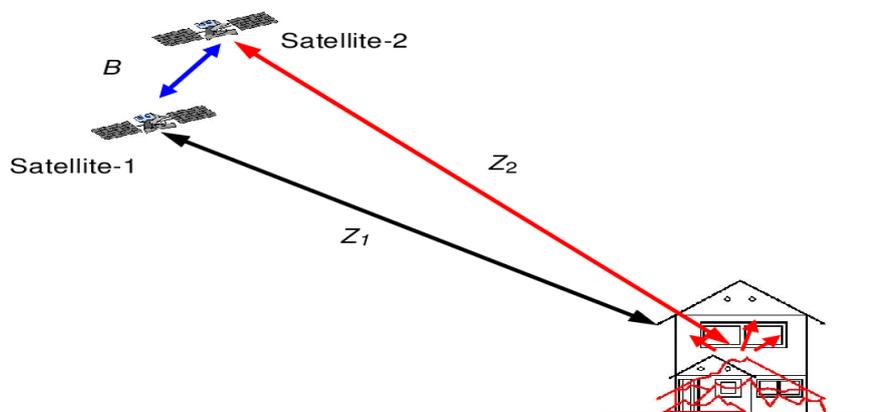


FIGURE 2.1 – acquisition d'image par 2 satellites sous angle différente

Le rayonnement transmis par le radar doit atteindre les diffuseurs sur le sol puis revenir au radar pour former l'image SAR (voyage à double sens). Des Diffuseurs à différentes distances du radar (sur ground range) introduisent différents délais entre la transmission et la réception du rayonnement.

En raison de la nature presque purement sinusoïdale du signal transmis, ce retard t équivaut à un changement de phase ϕ entre le signal émis et reçu . Le changement de phase est donc proportionnel à la distance de déplacement dans les deux sens $2R$ du rayonnement divisé par la longueur d'onde transmise λ . Ce concept est illustré dans la figure suivante

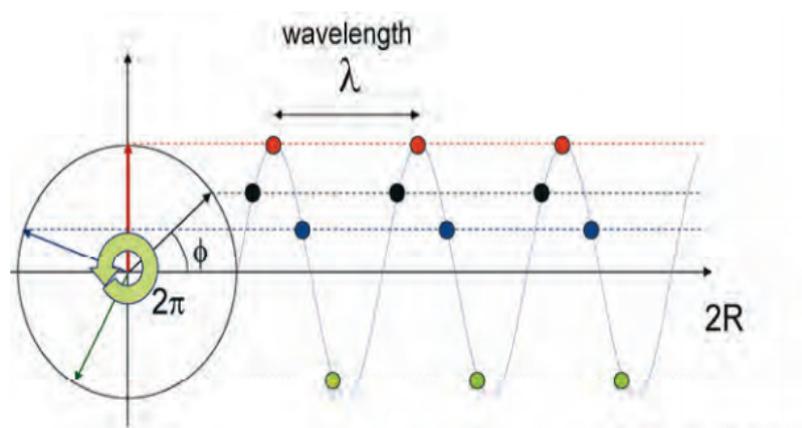


FIGURE 2.2 – Signal reçu

Une fonction sinusoïdale $\sin \phi$ est périodique avec une période de 2π radian

en supposant que la phase du le signal transmis est nul, le signal reçu qui couvre la distance parcourue $2R$ du satellite à la cible allez-retour, montre une phase de :

$$\phi = \frac{2\pi}{\lambda} 2R = \frac{4\pi}{\lambda} R \quad (2.3)$$

2.2 Génération d'interferogramme

L'interférogramme SAR est généré par multiplication croisée, pixel par pixel, de la première image (Master) avec le conjugué complexe de la seconde (Slave)

$$S1.S2^* = A1.A2 \exp(j(\phi1 - \phi2)) \quad (2.4)$$

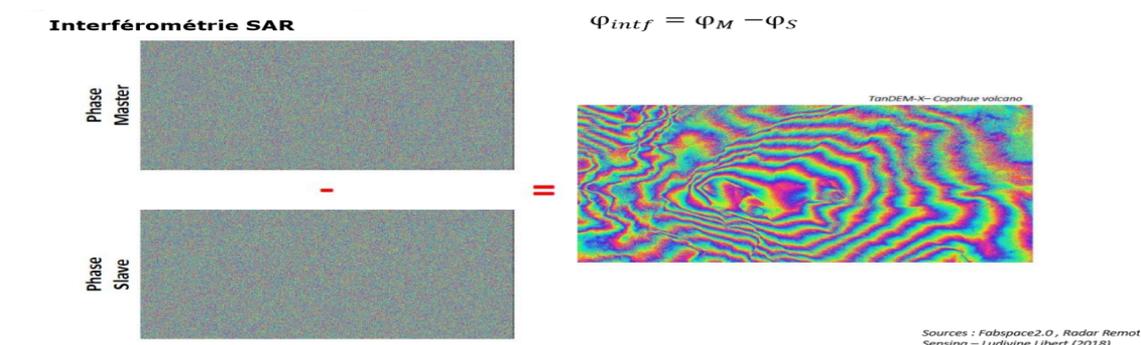


FIGURE 2.3 – Génération d'interferogramme

En utilisant l'équation 2.3 on aura :

$$\phi_1 - \phi_2 = \frac{4\pi}{\lambda}(R_1 - R_2) \quad (2.5)$$

2.2.1 Contenu de la phase interférométrique

La phase interférométrique contient une multitude de termes :

$$\phi_{intf} = \phi_{orb} + \phi_{topo} + \phi_{mvt} + \phi_{atm} + \phi_{bruit} \quad (2.6)$$

$$\phi_{topo} = \frac{h}{\sin \theta} \frac{Bn}{R} \frac{4\pi}{\lambda} \quad (2.7)$$

ϕ_{orb} : cette phase a un rapport avec l'orbite (sa distance par rapport au cellules) ou le satellite est mis en place elle est défini par :

$$\phi_{orb} = \frac{4\pi}{\lambda} \frac{Bn.s}{R_0 \tan \theta} \quad (2.8)$$

ϕ_{mvt} : cette phase a un rapport avec la vitesse et le mouvement de notre satellite

ϕ_{bruit} : Elle est produite par le système électronique d'émission et d'acquisition (temps de réponse des amplificateurs et autres dispositifs). Cette différence de phase est considérée comme nulle, car le dispositif d'émission et d'acquisition est généralement le même.

- L'atmosphère est soit négligée (pas toujours possible) soit corrigée (stacking, split-band, ...)

Supposons que nous ayons un seul diffuseur de points dominant dans chaque cellule de résolution au sol qui ne change pas avec le temps. Ces diffuseurs ponctuels sont observés par deux SAR sous des angles de vue légèrement différents. Dans ce cas, la phase interférométrique de chaque pixel d'image SAR ne dépendrait que de la différence des trajets entre chacun des deux SAR et la cellule de résolution considérée.

Toute contribution de phase possible introduite par les diffuseurs ponctuels n'affecte pas la phase interférométrique car elle est annulée par la différence de cette dernière. Une fois qu'un point de référence au sol a été identifié, la variation de la différence de chemin de déplacement Δr qui se traduit par le passage de la cellule de résolution de référence à une autre peut être donnée par une expression simple (une approximation qui vaut pour les petites lignes de base et les cellules de résolution qui ne sont pas trop loin) qui dépend de quelques paramètres géométriques illustrés à la figure

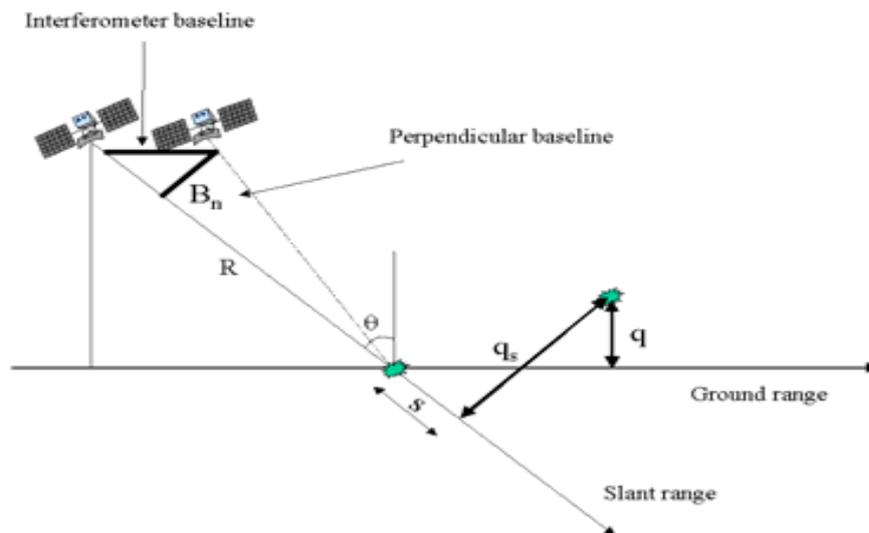


FIGURE 2.4 – paramètres géométriques

Les paramètres sont :

- La ligne de base perpendiculaire B_n
- La distance radar-cible R
- Le déplacement entre les cellules de résolution le long de la perpendiculaire à la plage oblique, q_s

L'expression approximative suivante de Δr est la suivante :

$$\Delta r = -2 \frac{B_n \cdot q_s}{R} \quad (2.9)$$

Δr : la variation de la différence de chemin de déplacement

La variation de phase interférométrique $\Delta\phi$ est alors proportionnelle à Δr divisée par la longueur d'onde transmise λ :

$$\Delta\phi = \frac{2\pi \cdot \Delta r}{\lambda} = -\frac{4\pi}{\lambda} \frac{Bn \cdot qs}{R} \quad (2.10)$$

2.2.2 Aplatissement de l'interférogramme

La phase dépend de l'angle d'incidence, nous observons une variation de la phase même en l'absence de topographie. Il s'agit de la phase orbitale, la contribution peut être modélisée et soustraite pour avoir les bonnes informations.

La variation de cette phase interférométrique peut être divisée en deux contributions :

- Une variation de phase proportionnelle à la différence d'altitude q entre les cibles ponctuelles, rapportée à un plan de référence horizontal
- Une variation de phase proportionnelle au déplacement de la plage oblique s des cibles ponctuelles

$$\Delta\phi = -\frac{4\pi}{\lambda} \frac{Bn \cdot q}{R \sin \theta} - \frac{4\pi}{\lambda} \frac{Bn \cdot s}{R \tan \theta} \quad (2.11)$$

où

θ est l'angle d'incidence du rayonnement par rapport à la référence. Il convient de noter que la ligne de base perpendiculaire est connue à partir de données orbitales précises, et le second terme de phase peut être calculé et soustrait de la phase interférométrique. Cette opération est appelée aplatissement d'interférogramme et, par conséquent, elle génère une carte de phase proportionnelle à l'altitude relative du terrain. Un exemple d'aplatissement de l'interférogramme est illustré à la figure suivante.

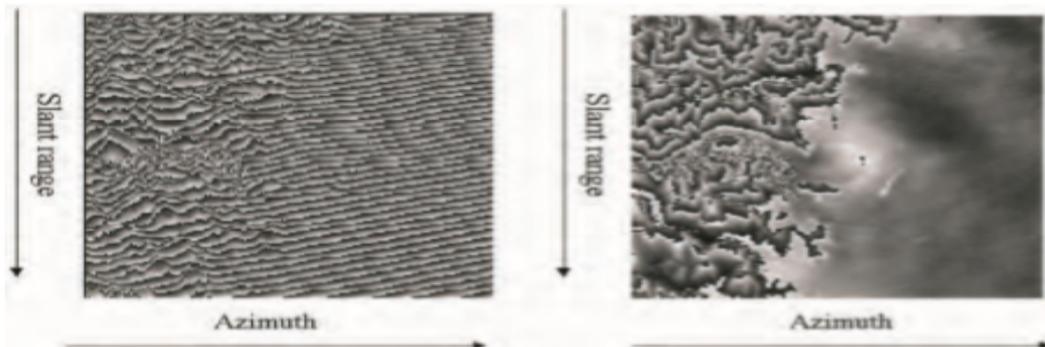


FIGURE 2.5 – Aplatissement d'interferogramme

Un interférogramme d'une partie des Alpes italiennes et du Pianura Padana obtenu à partir des données ERS-1 et ERS-2 (prises à un jour d'intervalle avec une ligne de base

normale d'environ 30 mètres) est illustré à gauche. L'interférogramme aplati est indiqué sur le côté droit. Ici, les discontinuités de phase ressemblent aux lignes de contour. L'altitude entre deux discontinuités adjacentes est appelée altitude d'ambiguïté (symbole ha) et peut être calculée à partir du paramètre interféromètre

2.2.3 Altitude d'ambiguïté

L'altitude d'ambiguïté ha est définie comme la différence d'altitude qui génère un changement de phase interférométrique de 2π (La phase interférométrique est comprise entre $-\pi$ et $+\pi$.) après l'aplatissement de l'interférogramme. L'altitude d'ambiguïté est inversement proportionnelle à la ligne de base perpendiculaire :

$$ha = \frac{\lambda R \sin(\theta)}{2Bn} \quad (2.12)$$

Dans le cas ERS avec $\lambda = 5,6$ cm, $\theta = 23^\circ$ et $R = 850$ km, l'expression suivante est valable (en mètres) :

$$ha \approx \frac{9300}{Bn} \quad (2.13)$$

si une ligne de base perpendiculaire de 100 mètres est utilisée, un changement de 2π de la phase interférométrique correspond à une différence d'altitude d'environ 93 mètres. En principe, plus la ligne de base est élevée (comme illustré sur la prochaine figure), plus la mesure d'altitude est précise, car le bruit de phase équivaut à un bruit d'altitude plus petit. Cependant, il existe une limite supérieure à la ligne de base perpendiculaire, sur laquelle les signaux interférométriques se décorrèlent et aucune frange ne peut être générée. En conclusion, il existe une ligne de base perpendiculaire optimale qui maximise le rapport puissance signal / bruit (où le signal est l'altitude du terrain). Dans le cas de l'ERS, une telle ligne de base optimale est d'environ 300 à 400 mètres.

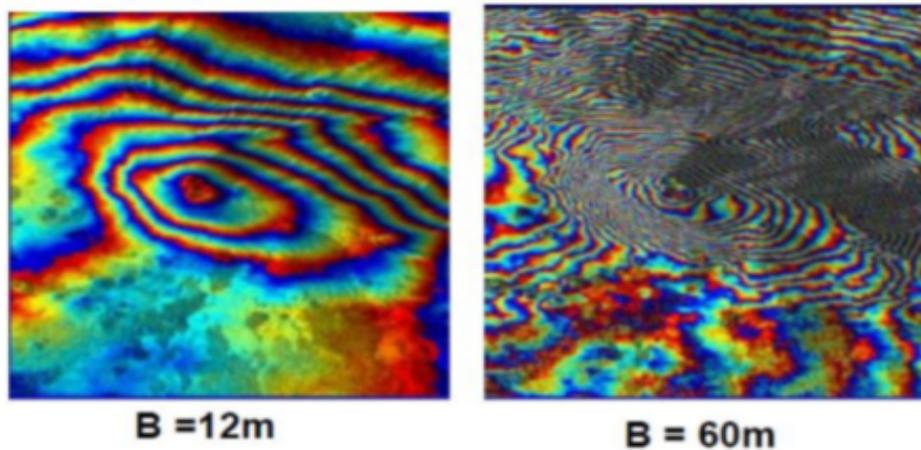


FIGURE 2.6 – effet de changement de la taille de la base ligne

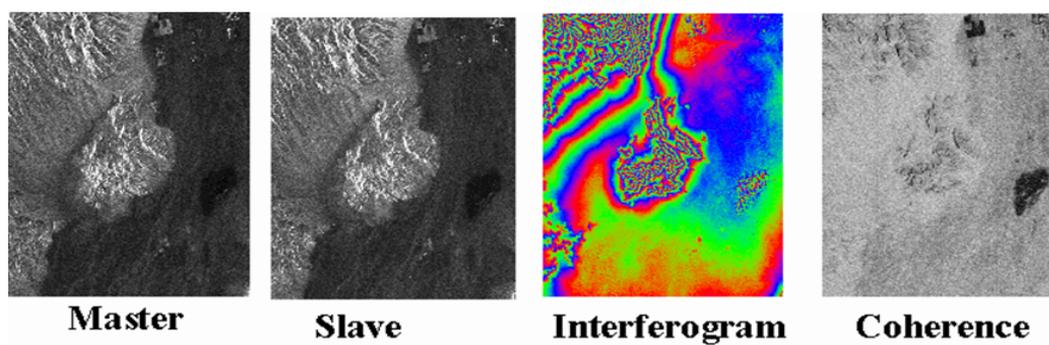
2.3 Cohérence

La cohérence est une mesure de qualité de l'interférogramme. Il s'agit de la corrélation complexe entre l'image maîtresse et l'image esclave. Le bruit de phase peut être estimé à partir de la paire SAR interférométrique au moyen de la cohérence locale γ . La cohérence locale est le coefficient de corrélation croisée de la paire d'images SAR estimé sur une petite fenêtre (quelques pixels de portée et d'azimut), une fois que toutes les composantes de phase déterministes (principalement en raison de l'élévation du terrain) sont compensées. Les composantes de phase déterministes dans une si petite fenêtre sont, en première approximation, linéaires à la fois en azimut et en plage oblique. Ainsi, ils peuvent être estimés à partir de l'interférogramme lui-même au moyen de méthodes bien connues de détection de fréquence de sinusoides complexes dans le bruit (par exemple, transformée de Fourier rapide 2-D (FFT)). La carte de cohérence de la scène est ensuite formée en calculant la valeur absolue de γ sur une fenêtre mobile qui couvre toute l'image SAR.

$$\gamma = \frac{|\sum_w M.S^*|}{\sqrt{\sum_w |M|^2 * \sum_w |S|^2}}, \in [0, 1] \quad (2.14)$$

Où W est une fenêtre autour du pixel (par exemple 5x5) qui doit prendre un nombre impair

- Une corrélation de 1 signifie une cohérence parfaite entre les images, avec de belles franges parfaitement déroulables
- Une corrélation nulle signifie un changement de l'état du sol entre les deux acquisitions, entraînant une perte de cohérence. Le signal n'est pas utilisable pour de l'InSAR

FIGURE 2.7 – *Generation de coherence*

On appelle décorrélation toute perturbation dans la différence de phase entre deux images SAR. Les décorrélations se traduisent par un bruit visible dans l'interférogramme et d'une baisse de la cohérence mesurée par le coefficient de corrélation. On distingue 3 principaux types de décorrélations :

– **Décorrélation spatiale :**

Elle est due à la séparation des deux antennes radars, et l'observation sous des angles d'incidence légèrement différents. Même si les pixels couvrent les mêmes cellules, le motif d'interférence des réflexions élémentaires varie en raison de l'évolution de la géométrie.

– **Décorrélation temporelle :**

Spécifique à l'interférométrie multi-passes, elle est due au changement de l'état de la surface, phénomène difficile à prédire et à modéliser. Les modifications de la distribution des réflecteurs élémentaires et de leurs réponses peuvent provenir de causes très variées entraînant le déplacement des sources (la croissance de la végétation, le vent, l'exploitation agricole du sol) et/ou une modification des caractéristiques radio - électriques des cibles (humidité, température, . . . etc.). Ces paramètres mal maîtrisés sont un obstacle majeur aux applications topographiques de l'interférométrie. En revanche la mesure de corrélation peut fournir une information intéressante pour d'autres applications en classification (suivi des cultures, ...etc.).

– **Décorrélation thermique :**

Elle est due au bruit thermique additif lié au capteur, influant sur la précision de la mesure de la phase.

2.4 Differential InSAR

Interférométrie différentielle est le terme couramment utilisé pour la production d'interférogrammes dont la contribution topographique a été supprimée et remplacé par des

data prise a une date antérieure. Cependant, le terme peut parfois être trompeur, car d'une part l'interférométrie est une technique différentielle dès le début, et d'autre part, le processus de soustraction peut être poussé plus loin ainsi que dans d'autres directions (par exemple, soustraction d'une géophysique attendue, contribution par la modélisation dynamique des tremblements de terre ou des volcans).

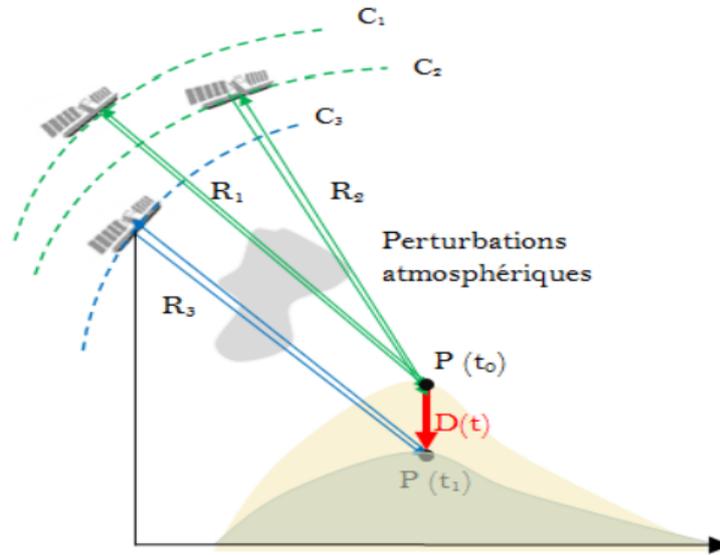


FIGURE 2.8 – Differential InSAR

D'après la géométrie d'acquisition des deux passes illustrée dans la figure précédente, la différence des distances est donnée par l'équation suivante :

$$\Delta R = (R1 - R2) + Dp \quad (2.15)$$

Dp étant la déformation de la cible P dans la direction de visée du satellite. Pour rappel, la phase interférométrique du point P est donnée par l'équation suivante :

$$\phi(p) = \phi1(p) - \phi2(p) = \frac{-4\pi}{\lambda}(R1 - R2) = \frac{-4\pi}{\lambda}\Delta R \quad (2.16)$$

En utilisant l'équation précédente, la différence de phases des deux acquisitions (avant et après) est formulée par :

$$\Delta\phi(p) = \frac{-4\pi}{\lambda}(R3 - R1) + \frac{4\pi}{\lambda}Dp \quad (2.17)$$

Le premier terme est proportionnel aux franges topographiques et orbitales, tandis que le deuxième terme est proportionnel aux franges de déplacement. La variation de cette

phase par rapport à une référence en fonction des paramètres géométriques du modèle d'acquisition interférométrique est donnée par l'expression suivante :

$$\Delta\phi_p = \frac{-4\pi}{\lambda} \left(D_p - \frac{B_{\perp}^0}{R1 \sin \theta} Z_p + B \sin(\theta^0 - a) \right) \quad (2.18)$$

En distribuant les termes, on aboutit à l'équation suivante :

$$\Delta\phi_p = \frac{-4\pi}{\lambda} D_p - \frac{-4\pi}{\lambda} \left(\frac{B_{\perp}^0}{R1 \sin \theta} Z_p \right) + \frac{-4\pi}{\lambda} B \sin(\theta^0 - a) \quad (2.19)$$

Tel que :

$$\frac{-4\pi}{\lambda} B \sin(\theta^0 - a) = \phi_{FO} : \textit{Phaseduealabaseline}. \quad (2.20)$$

$$\frac{-4\pi}{\lambda} \left(\frac{B_{\perp}^0}{R1 \sin \theta} Z_p \right) = \phi_{topo} : \textit{Phaseduealatopographie}. \quad (2.21)$$

Les deux phases ci-dessus sont éliminées. On ne garde donc que la phase de déformation ϕ_p :

$$\phi_p = \frac{-4\pi}{\lambda} D_p \quad (2.22)$$

D'où :

$$D_p = \frac{-\lambda}{4\pi} \phi_p \quad (2.23)$$

Pour un déphasage de 2π , on a donc :

$$D_p = \frac{-\lambda}{4\pi} 2\pi = \frac{-\lambda}{2} \quad (2.24)$$

Où λ est la longueur d'onde de l'impulsion radar.

2.5 Conclusion :

Après avoir appris les bases de la télédétection et des radars, on est passé dans ce chapitre à l'interférométrie et la façon de sa génération. après on a introduit le contenu de sa phase, et l'une des étapes les plus importantes qui est l'aplatissement de ce dernier qui va rendre toutes ses informations relatives à l'altitude du terrain, aussi on a parlé de l'altitude d'ambiguïté qu'on aura besoin dans le chapitre IV, après sa on est passé à la cohérence et son utilité pour savoir la qualité de notre prise et nos images sar, à la fin on a introduit le differential Insar et son modèle géométrique.

Chapitre 3

Déroulement de Phase

3.1 Introduction

Le déroulement de phase est la partie la plus importante de l'InSAR . Le problème posé par la représentation de la phase d'un signal complexe nous mène à chercher des techniques précises car la phase interférométrique n'est connue que modulo- 2π et le maximum de variation de hauteur dans la zone d'intérêt peut donner lieu à des centaines de cycles, une procédure de déballage est nécessaire pour estimer la topographie locale. Si la contribution de phase due à une Terre idéalement plate a été correctement estimée et compensée (c'est-à-dire que l'interférogramme a été aplati), le déroulement de phase permet de passer du motif de frange (similaire à un ensemble de courbes de niveau) à un champ de phase proportionnelle à la topographie locale. Dans la plupart des cas ce dernier est le principal obstacle à surmonter dans la chaîne de transformation pour la reconstruction du MNT (Module Numérique de terrain) In Sar, et ne peut souvent pas être fait de manière automatique.

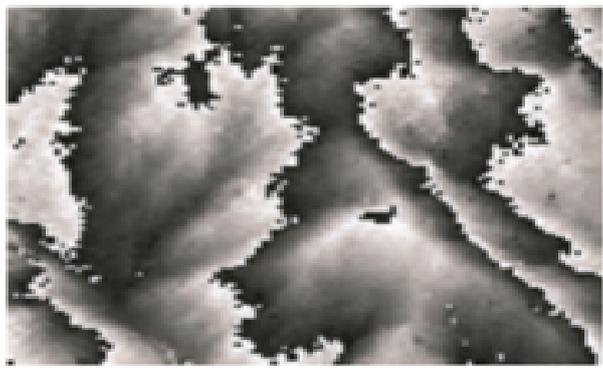


FIGURE 3.1 – avant le déroulement



FIGURE 3.2 – après le déroulement

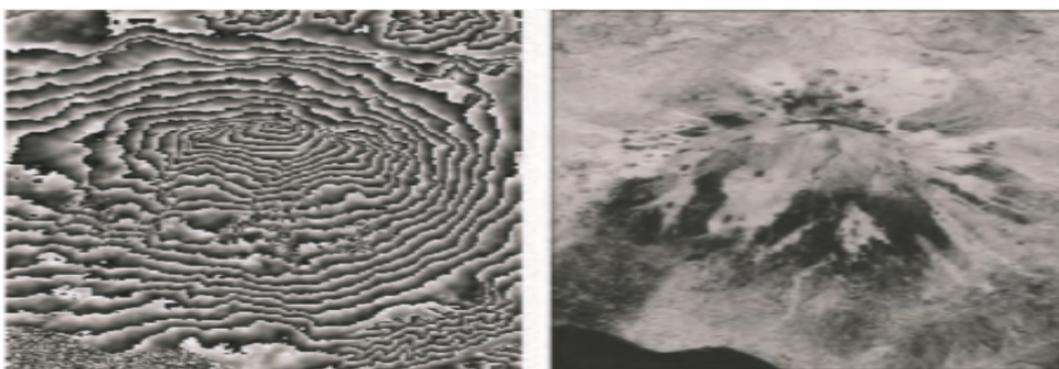


FIGURE 3.3 – avant et après le déroulement

Il existe plusieurs techniques de déroulement de phase bien connues qui seront décrites dans ce qui suit.

Cependant, il convient de noter ici que le déroulement de phase n'a généralement pas de solution unique et que les informations a priori doivent être exploitées pour obtenir la bonne solution. Une fois les phases interférométriques déroulées, une carte d'élévation en coordonnées SAR est obtenue. Il s'agit de la première étape vers l'obtention d'un MNT (Module Numérique de terrain).

3.1.1 Le problème à résoudre

Le déroulement de phase (PU) a pour but de récupérer le nombre entier de cycles n à ajouter à la phase mesurée Φ de sorte que la valeur de phase réelle Ψ puisse être finalement obtenue pour chaque pixel d'image :

$$\Psi = \Phi + 2\Pi.n \quad (3.1)$$

n = nombre de cycles

Φ : Phase mesuré

Ψ : Phase réel

Dans le cas mono-dimensionnel et sous la contrainte de la continuité, il existe une solution unique à une translation près au problème posé (figure suivante) : la phase réelle est l'intégrale de la pseudo dérivée de la phase mesurée le long du chemin reliant 2 sauts de phase [Labrousse96].

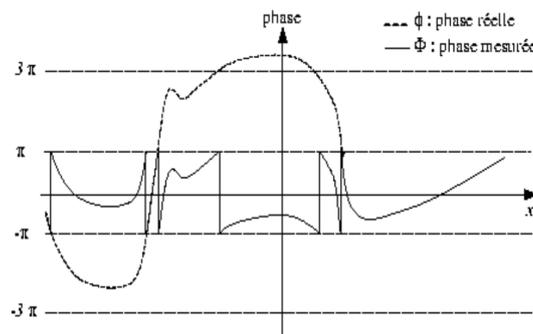


FIGURE 3.4 – 1d unwrapping

Dans le cas InSAR (2D), si aucune information a priori sur Φ n'est disponible, c'est-à-dire qu'aucune contrainte n'est donnée à la solution (par exemple, bande de fréquence maximale et puissance du signal), le déroulement de phase est un problème inverse mal posé et donc un nombre infini de solutions différentes peuvent être trouvées, tous honorant les données. La procédure PU la plus simple serait une simple intégration des différences de phase, à partir d'un point de référence. Cependant, en raison des discontinuités de phase, elle n'est pas toujours précise. Presque tous les algorithmes PU sont basés sur l'hypothèse que le véritable champ de phase unwarped est «lisse» et varie «lentement».

Plus précisément, les valeurs des phases voisines sont supposées être à l'intérieur d'un demi-cycle (Π radians) les unes des autres.

Bien que cette hypothèse soit souvent valable pour la plupart des pixels de l'image, la présence de certaines discontinuités de phase dite **résidus** (c'est-à-dire des variations de phase absolues entre pixels voisins de plus de Π radians) provoque des incohérences, car l'intégration donne des résultats différents selon le chemin

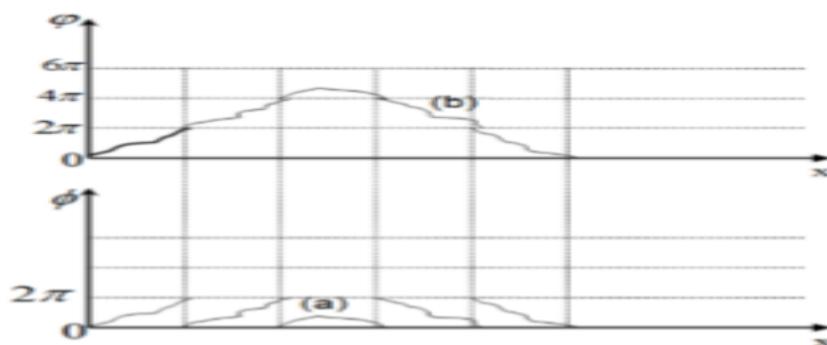


FIGURE 3.5 – avant et après le déroulement

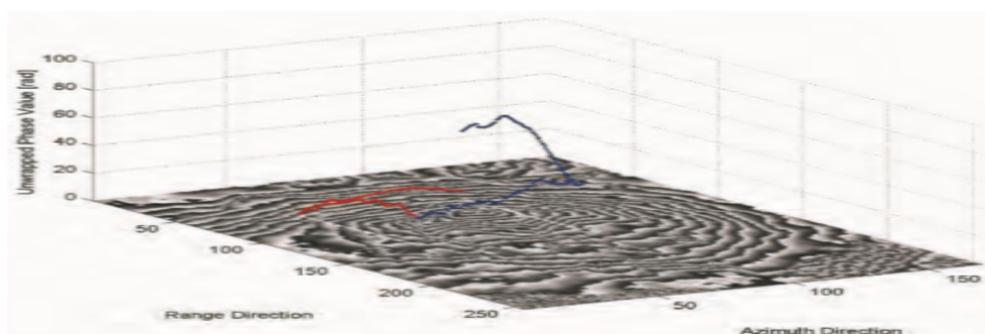


FIGURE 3.6 – logique de l'algorithme

Voilà Un exemple (Figure 3.6) visuel qui montre le principal problème lié aux incohérences de phase : l'intégration des valeurs de phase donne des résultats différents selon le chemin suivi.

Dans cet exemple, les premier et dernier pixels des chemins d'intégration sont communs, mais un chemin traverse une zone de transition, caractérisée par des valeurs de cohérence très faibles, et propage d'importantes erreurs de déroulement de phase.

Cette caractéristique est évidente chaque fois que la somme des différences de phase mesuré (l'intégrale du gradient de phase estimé) autour d'un chemin fermé (de taille de 2×2 pixel en interférométrie) diffère de zéro. Pour être cohérent, un champ de gradient doit être irrotationnel ; c'est-à-dire que la boucle de $\nabla \Phi$ doit être nulle partout comme illustré dans la figure suivante :

$$residu(i, j) = g_x(i, j) - g_y(i + 1, j) - g_x(i, j + 1) + g_y(i, j)$$

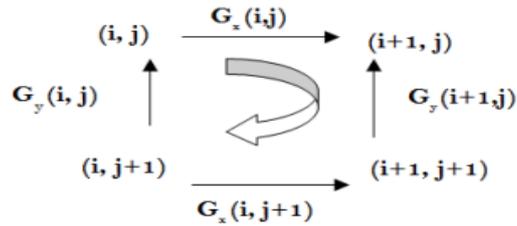


FIGURE 3.7 – Emplacement de résidus

Chaque fois que cette condition est vérifiée sur l'ensemble de l'interférogramme, nous aurons un «problème PU insignifiant».

Malheureusement, ce n'est presque jamais le cas dans le traitement des données InSAR. La composante rotationnelle du champ de gradient peut facilement être estimée en additionnant les différences de phase enveloppées autour des chemins fermés formés par chaque ensemble mutuellement voisin de quatre pixels. Chaque fois que la somme n'est pas nulle, **un résidu** se produirait. Sa valeur est généralement normalisée à un cycle et elle peut être positive (+1) ou négative (-1). La somme des variations de phase enveloppées le long d'un chemin fermé arbitraire est égale à la somme algébrique des résidus enfermés dans le chemin.

Étant donné que les résidus de phase ne font que marquer les extrémités des «lignes de discontinuité», le véritable problème est leur identification complète. Les discontinuités sont essentiellement dues à deux facteurs indépendants :

- le bruit de phase
- La pentes raides du terrain

En interférométrie à répétition, nous traitons généralement de faibles valeurs SNR (≈ 0 dB) en raison de la décorrélation temporelle ou géométrique, et la probabilité sur terrain plat d'une variation de phases (bruyante) supérieures à π n'est pas du tout négligeable (pour la cohérence $\gamma = 0,5$ et un nombre effectif de regards

$$Neff = 3p(|\Delta\phi| > \pi) \approx 0.01 \quad (3.2)$$

Par contre, la géométrie d'acquisition du capteur donne lieu à un échantillonnage inégal du terrain en ground range.

Les pixels voisins dans la direction du ground range peuvent correspondre à deux diffuseurs très éloignés dans le ground range avec des altitudes très différentes. En général, le nombre de discontinuités sera en fonction de la topographie locale (caractérisée, par exemple, par une certaine dimension fractale), l'angle hors nadir, la ligne de base normale et le bruit de décorrélation.

3.2 Techniques de déroulement de phase

Afin de faire face aux discontinuités de phase, différentes stratégies ont été suivies et différents algorithmes ont été développés. À la suite de Ghiglia et Romero et Chen et Zebker , nous les décrivons brièvement, en utilisant le cadre de la «norme Lp minimale».

En faite, presque tous les algorithmes PU cherchent à minimiser la fonction de coût suivante :

$$C = \left\{ \sum_{i,j} w_{i,j}^{(r)} \left| \Delta^{(r)} \psi_{i,j} - \Delta_w^{(r)} \phi_{i,j} \right|^p + \sum_{i,j} w_{i,j}^{(a)} \left| \Delta^{(a)} \psi_{i,j} - \Delta_w^{(a)} \phi_{i,j} \right|^p \right\} \quad (3.3)$$

où $0 \leq p \leq 2$;

Δ :indique une différenciation discrète le long de la Ground range (r) et de l'azimuth (a) respectivement ;

W :sont des poids définis par l'utilisateur ; et les sommations incluent toutes les lignes appropriées i et la colonne j.

Le suffixe w à l'opérateur de différenciation Δ indique que les différences de phase sont enveloppées dans l'intervalle $-\pi$ à $+\pi$ Nous soulignons que cette fonction objective n'a pas été obtenue à partir d'une analyse théorique ou d'une description statistique d'un signal de phase topographique en coordonnées SAR. C'est juste une traduction raisonnable en termes mathématiques de notre hypothèse de base :

$\Delta \Psi = \Delta w \theta$ presque partout. Néanmoins, il a été utilisé pour sa simplicité et en raison du fait que des algorithmes efficaces sont disponibles pour $p = 2$ et $p = 1$.

3.2.1 Unweighted Least Mean Squares method (cas ou $p=2$) :

Analysons d'abord la méthode (ULMS). Dans ce cas, $p = 2$ et aucun poids n'est présent

$$W_{i,j} = 1 \quad \forall_{i,j} \quad (3.4)$$

L'équation du cout 3.6 conduit à un système linéaire d'équations

$$\psi_{i+1,j} - \psi_{i,j} = \Delta_w^{(a)} \phi_{i,j} \quad (3.5)$$

$$\psi_{i,j+1} - \psi_{i,j} = \Delta_w^{(r)} \phi_{i,j} \quad (3.6)$$

Dans ce cas, les conditions aux limites changent et le réseau est résolu par rapport à (au moins) un pixel d'élévation connue.

Les inconvénients de cette approche peuvent être facilement envisagés en examinant la logique qui la sous-tend. Nous ne nous soucions pas du nombre et de la position des discontinuités. Nous écrivons simplement les équations, nous espérons que seules quelques-unes d'entre elles sont erronées et nous profitons de packages de solutions prêts à l'emploi pour obtenir le résultat

Malheureusement, la simplicité et la précision vont rarement de pair. Les solutions LMS non pondérées sont sujettes à de graves erreurs causées par des discontinuités de phase. Chaque mauvaise équation donne lieu à des artefacts de phase autour d'elle, de sorte que les problèmes de propagation d'erreur ne sont en aucun cas surmontés. De plus, la solution ne correspond aux données que dans le cas trivial ($\Psi = \phi + 2\Pi n$), où aucune discontinuité n'est présente.

3.2.2 Weighted Least Mean Squares method (Cas ou p=2) :

La qualité des résultats est quelque peu améliorée par la pondération des équations. En fait, la carte de cohérence associée à l'interférogramme et / ou aux images d'amplitude peut être utilisée avec succès pour identifier les zones où des discontinuités de phase sont susceptibles de se produire, mais en faisant ça, nous perdons la structure régulière de la matrice à inverser.

Des techniques numériques itératives efficaces peuvent être adoptées, mais elles conduisent à des augmentations significatives du temps de calcul. En outre, pour une reconstruction de phase correcte, des poids zéro devraient être appliqués aux discontinuités de phase et aux poids unitaires chaque fois que le gradient de phase est correct, mais cela impliquerait l'identification des «sauts de cycle», ce que les méthodes LMS voudraient éviter.

Les artefacts de phase dans le champ non enveloppé estimé en raison de données bruyantes peuvent être partiellement évités si les gradients de phase locaux sont estimés sur des

fenêtres plus grandes , en utilisant des techniques d'estimation de fréquence 2D plus fiables (par exemple, analyse FFT(**transformée de Fourier rapide**)).

Le problème est fondamentalement un problème de reconstruction de surface variationnelle à partir de mesures indirectes, bien connu dans le domaine de la vision par ordinateur. Cette approche est parfois appelée PU LMS généralisée. Bien sûr, son utilisation nécessite un compromis pas facile entre la précision de l'estimation et la résolution. Des résultats intéressants ont été obtenus en utilisant une implémentation hiérarchique rapide d'un estimateur multi-résolution . Les résultats LMS et WLMS ne respectent pas l'interférogramme d'origine ($\Psi \neq \phi + 2\Pi n$), à l'exception du cas trivial. Bien sûr, la solution peut être forcée d'être congruente avec les données (en arrondissant la différence entre la phase enveloppée et la phase non enveloppée au nombre entier de cycles le plus proche), mais cela n'équivaut pas à minimiser l'équation du coût avec la «contrainte d'entier» . En fait, il s'agit d'un problème NP difficile à résoudre dont la théorie de la complexité suggère qu'il est impossible pour des algorithmes efficaces de résoudre exactement.

3.2.3 Minimum Cost Flow method (Cas ou $p=1$)

Une approche de PU (« phase unwrapping » ou déroulement de phase) basée sur la programmation réseau a reçu beaucoup d'attention, car elle fournit un outil efficace pour une minimisation globale de l'équation. 3.6 sous la norme L1 (pondérée) (écart absolu minimum).

PU est formulé comme un problème d'optimisation contraint. L'algorithme minimise le nombre entier de cycles à ajouter aux variations de phase (c'est-à-dire les données Δw ϕ) pour les rendre cohérentes. Comme déjà mentionné, pour être cohérent, un champ de gradient doit être irrotationnel ; ce n'est que dans ce cas que le champ de phase non enveloppé est indépendant du chemin d'intégration, jusqu'à une constante additive.

La contrainte à satisfaire est alors $\nabla * \nabla \phi = 0$, pas celle d'une solution entière. Cette propriété de bienvenue n'est qu'une conséquence des équations utilisées . Dans l'approche MCF (Minimum Cost Flow), le problème PU est assimilé à un problème général de flux réseau . Cette reformulation du problème PU permet l'utilisation de techniques puissantes développées pour l'optimisation des réseaux .

La théorie des graphes et la programmation réseau étant un sujet mature de recherche opérationnelle, plusieurs routines d'optimisation rapide peuvent être utilisées pour rechercher le flux de coûts minimum. De plus, de nombreux codes sources sont disponibles sur le web [SNAPHU]. Les détails des algorithmes peuvent être trouvés dans [Ahuja93]. En ce

qui concerne l'estimation WLMS, l'utilisateur peut définir des zones de marquage de poids appropriées où les discontinuités de phase sont plus susceptibles de se produire. Même dans ce cas, étant donné que les grandeurs d'écoulement sont limitées à des entiers, le champ de phase non enveloppé final est conforme à l'interférogramme d'origine ($W[\phi] = \Psi$) ; Les algorithmes MCF sont vraiment des programmes PU, car ils ajoutent simplement un nombre entier de cycles à chaque valeur de phase encapsulée.

3.2.4 Branch-Cut method (Cas ou $p=0$)

Lorsque $p = 0$, l'équation 3.6 est égale au nombre d'échantillons pour lesquels les gradients de solution ne sont pas exactement égaux aux gradients mesurés.

Ainsi, les gradients du champ de phase non enveloppé estimé sont exactement égaux aux données en autant d'endroits que possible, c'est l'un des objectifs fréquemment suggérés pour PU et historiquement le premier utilisé dans le traitement des données InSAR. L'idée de base de la méthode Branch-cut est de dérouler l'interférogramme en sélectionnant uniquement les voies d'intégration qui conduisent à des solutions auto-cohérentes. La partie rotationnelle du gradient de phase estimé (c'est-à-dire le champ de résidus) est utilisée comme carte pour piloter l'intégration le long d'un chemin cohérent, limitant la propagation des erreurs.

Comme déjà mentionné, la somme des variations de phase enveloppées le long d'un chemin fermé arbitraire est égale à la somme algébrique des résidus enfermés dans le chemin. Les voies d'intégration qui entourent une «charge» nette doivent être évitées. Ceci est accompli en connectant des résidus de charge opposée avec des coupes de branche que les chemins d'intégration ne peuvent pas traverser.

Habituellement, les coupes sont des lignes droites et sont cultivées de manière arborescente comme dans un algorithme de croissance de région. Les résidus marquent les extrémités des «lignes de discontinuité» (souvent appelées «lignes fantômes» ou «lignes de repliement»). La stratégie adoptée pour identifier le nombre minimum de coupures de branches distingue les différents membres de cette classe d'algorithmes. L'un des inconvénients majeurs de la méthode est que les coupures peuvent se refermer sur elles-mêmes dans les zones de faible cohérence, de sorte que des trous peuvent exister dans le champ de phase non enveloppé et, en général, de mauvais résultats sont obtenus dans les zones de faible SNR.

Une approche différente a été proposée par Ghiglia et Romero. Il s'agit d'un schéma itératif de la solution générale L_p -standard et permet une couverture complète.

Les équations gouvernantes sont équivalentes à celles qui décrivent l'algorithme PU WLMS,

mais les poids dépendent des données. La convergence vers un minimum global n'est pas garantie (cela devrait être prévu étant donné la nature hautement non linéaire de la fonction de coût) et l'algorithme est intensif en calcul, bien que l'application de la méthode soit facile une fois qu'un programme WLMS est disponible. Chen et Zebker ont démontré que le problème de norme L_0 minimum est NP-difficile (en gros, il faudrait trop de temps pour trouver le minimum global de la fonction de coût) et les efforts de recherche devraient être concentrés sur concevoir des algorithmes approximatifs. D'autre part, nous rappelons ici que les critères de la norme L_0 n'ont pas été obtenus au moyen d'une analyse statistique rigoureuse du problème PU pour la reconstruction DEM InSAR ; il s'agit plutôt d'une tentative de minimiser le nombre de discontinuités dans la surface estimée .

3.3 Conclusion

Après qu'on a fini avec les bases de l'interférométrie, on est passé dans ce chapitre à la partie la plus difficile des processus de génération de modèles numérique de terrain, c'est le déroulement de phase .

Alors on a parlé du problème de la plupart des MNT qui est le calcul de nombre de cycles à ajouter à chaque phase, et pour résoudre sa on a proposé plusieurs techniques de déroulement, chacune avec ses points forts et ses points faibles, sachant que pour le moment et avec toutes les recherches dans ce domaine, aucune technique parfaite n'est mise en place .

Chapitre 4

Réalisation Logiciel

4.1 Introduction à la partie théorique

Dans ce chapitre on passe à la partie pratique de la génération du module numérique de terrain . Pour cela on a voulu créer un logiciel qui fait tous les processus qu'on a vus auparavant en une seule clique, pour la réalisation on a utilisé python et on s'est inspiré de SNAP toolbox , qui est un logiciel de traitement de données de l'Esa (european space agence).



FIGURE 4.1 – SNAP



FIGURE 4.2 – Python

4.1.1 Introduction à Snaptool

c'est un logiciel qui se compose d'un ensemble d'outils de traitement, de lecteurs et d'écrivains de produits de données et d'une application d'affichage et d'analyse pour prendre en charge les grandes archives de données des missions SAR de l'ESA, notamment SENTINEL-1, ERS-1 et 2 et ENVISAT , ainsi que des données SAR tierces d'ALOS PAL-SAR, TerraSAR-X, COSMO-SkyMed et RADARSAT-2. Les différents outils de traitement peuvent être exécutés indépendamment de la ligne de commande et également intégrés dans l'interface utilisateur graphique. La boîte à outils comprend des outils d'étalonnage, de filtrage des tâches, de coregistration, d'orthorectification, de mosaïquage, de conversion de données, de polarimétrie et d'interférométrie.

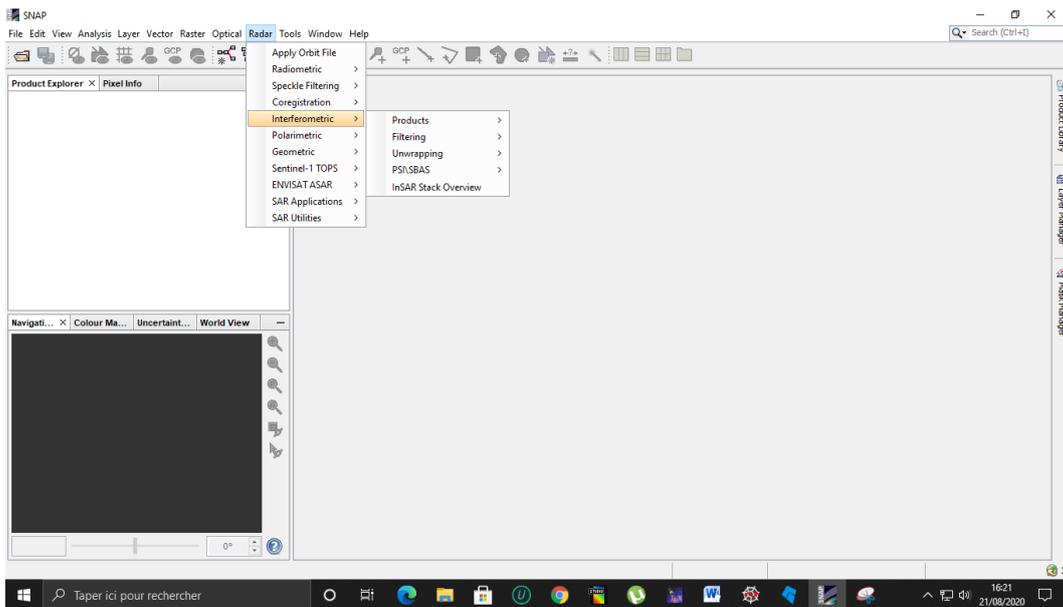


FIGURE 4.3 – avant le déroulement

4.2 notre logiciel

De notre côté, on a travaillé sur l'interférométrie et toutes ses étapes en essayons de la rendre très facile à manipuler , avec cette interface graphique simple qui est représentée dans la figure suivante ;

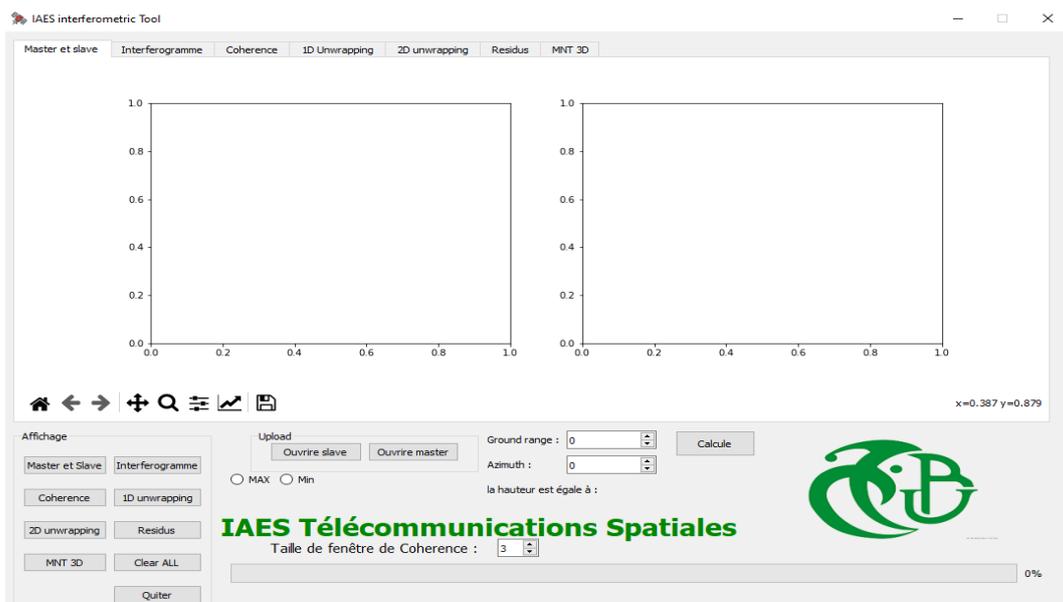


FIGURE 4.4 – Interface de notre logiciel

Comme montre dans la figure précédente notre logiciel se compose de :

- Plusieurs onglets d’affichage sous des noms différents Deux boutons pour sélectionner nos fichiers master et Slave
- Bouton pour afficher Master et Slave.
- Bouton pour calculer et afficher interferogramme.
- Bouton pour calculer et afficher la cohérence.
- SpinBox pour choisir la taille de fenêtre de calcul de cohérence.
- Progressbar pour savoir l’état d’avancement de nos calculs.
- Bouton pour calculer et afficher le déroulement d’une dimension.
- Bouton pour calculer et afficher le déroulement 2D.
- Bouton pour calculer et afficher les résidus.
- Bouton pour calculer et afficher Affichage la hauteur 3D de la map.
- Bouton pour calculer la hauteur selon les coordonnées (ground range , azimuth).
- Bouton pour le point max et min de hauteur de l’MNT.
- Bouton clear All pour supprimer tous les plots .
- Bouton Exit pour quitter .

Pour sa réalisation on est passé par des étapes qui sont :

- l’interface GUI
- Les fonctions interférométriques
- L’affichage des Plots
- L’optimisation du programme

4.2.1 l’interface GUI

Pour réaliser notre interface, on a utilisé la crosse-plateforme Py Qt5 de python pour utiliser ses bibliothèques multiples, comme le montre la figure suivante :

```

import os
import matplotlib.pyplot as plt |
import random
from PyQt5.QtWidgets import QDialog, QApplication, QMainWindow, QMenu, QVBoxLayout, QSizePolicy, Q
from PyQt5.QtGui import QIcon
import pyqtgraph as pg
from PyQt5.QtWidgets import *
from PyQt5.uic import loadUi

from matplotlib.backends.backend_qt5agg import (NavigationToolbar2QT as NavigationToolbar)
import numpy as np # calcule sur les matrices et les vecteurs
import scipy.io as spio # pour lire les données MATLAB
from matplotlib import cm # palette des couleur pour l'affichage import n_show # pour
import n_show
from mplwidget import MplWidget
from pathlib import Path
from matplotlib.backends.backend_tkagg import (
    FigureCanvasTkAgg, NavigationToolbar2Tk)
# Implement the default Matplotlib key bindings.
from matplotlib.backend_bases import key_press_handler
from matplotlib.figure import Figure
from PIL import ImageTk, Image

from matplotlib.backends.backend_qt5agg import FigureCanvasQTAgg as FigureCanvas
from matplotlib.figure import Figure
import matplotlib.pyplot as plt

from PyQt5 import QtCore, QtGui, QtWidgets
import sys

import matplotlib.colors as mcolors
import matplotlib.colorbar as mcolorbar

from PyQt5.QtCore import QFileInfo

```

FIGURE 4.5 – Bibliothèque utilisée

après l'importation des bibliothèques, on est passé au désigne du software qui se fait en précisant les tailles, les noms et la place géométrique des objets sur la window (dans la figure suivante une partie du code de désigne d'interface).

```

Form.setObjectName("Form")
Form.resize(1080, 720)
Form.setMinimumSize(QtCore.QSize(1080, 720))
Form.setMaximumSize(QtCore.QSize(1080, 720))

self.groupBox = QtWidgets.QGroupBox(Form)
self.groupBox.setGeometry(QtCore.QRect(10, 480, 201, 221))
self.groupBox.setObjectName("groupBox")
self.gridLayoutWidget = QtWidgets.QWidget(self.groupBox)
self.gridLayoutWidget.setGeometry(QtCore.QRect(10, 20, 181, 199))
self.gridLayoutWidget.setObjectName("gridLayoutWidget")
self.gridLayout = QtWidgets.QGridLayout(self.gridLayoutWidget)
self.gridLayout.setContentsMargins(0, 0, 0, 0)
self.gridLayout.setObjectName("gridLayout")
self.btnquit = QtWidgets.QPushButton(self.gridLayoutWidget)
self.btnquit.setObjectName("btnquit")
self.btnquit.clicked.connect(QApplication.instance().quit)
self.gridLayout.addWidget(self.btnquit, 6, 1, 1, 1)
self.btn7 = QtWidgets.QPushButton(self.gridLayoutWidget)
self.btn7.setObjectName("btn7")
self.btn7.clicked.connect(self.heighmap)

```

FIGURE 4.6 – Une partie du code désigne interface

Après ça on a mis des fenêtres pour pouvoir retrouver chaque affichage dans son onglet destiné, et non pas afficher l'une au-dessus de l'autre.

Alors on a mis en points un système de classe pour chaque fenêtre, comme montré dans la figure suivante ;

```
class w2(QWidget):
    def __init__(self, parent = None):
        QWidget.__init__(self, parent)
        self.canvas = FigureCanvas(Figure())
        self.canvas2 = FigureCanvas(Figure())
        self.toolbar = NavigationToolbar(self.canvas , self)
        horizontal_layout = QVBoxLayout()
        horizontal_layout.addWidget(self.canvas)
        horizontal_layout.addWidget(self.toolbar)
        self.canvas.axes = self.canvas.figure.add_subplot(121)
        self.canvas2.axes = self.canvas.figure.add_subplot(122)
        self.setLayout(horizontal_layout)
class w3(QWidget):
    def __init__(self, parent = None):
        QWidget.__init__(self, parent)
        self.canvas = FigureCanvas(Figure())
        self.canvas2 = FigureCanvas(Figure())
        self.toolbar = NavigationToolbar(self.canvas , self)
        horizontal_layout = QVBoxLayout()
        horizontal_layout.addWidget(self.canvas)
        horizontal_layout.addWidget(self.toolbar)
        self.canvas.axes = self.canvas.figure.add_subplot(121)
        self.canvas2.axes = self.canvas.figure.add_subplot(122)
        self.setLayout(horizontal_layout)
class w4(QWidget):
```

FIGURE 4.7 – Code des classes de fenêtre

4.3 Les fonctions

Après avoir terminé notre interface, on est passé aux fonctions qu'on veut implémenter à notre programme, et après les relier à leur bouton spécifique.

4.3.1 Affichage de Master et Slave

Pour afficher nos images Master et Slave On va d'abord les importer en utilisant QFileDialog de PYQT5 comme montre sur la Figure suivante :

```

def openslave(self):
    filename = QFileDialog.getOpenFileName()
    filenameee = os.path.basename(filename[0])
    data = spio.loadmat(filenameee)
    self.slave_fine = data['Slave_fine']

def openmaster(self):
    filename = QFileDialog.getOpenFileName()
    filenameee = os.path.basename(filename[0])
    data = spio.loadmat(filenameee)
    self.master_fine = data['Master_fine']

```

FIGURE 4.8 – Code importation des images

Après l'intégration des boutons, et leur connexion aux fonctions respectives d'openslave et open Master, on aura 2 boutons qui permettent d'importer nos images selon leur emplacement dans notre disque dur.

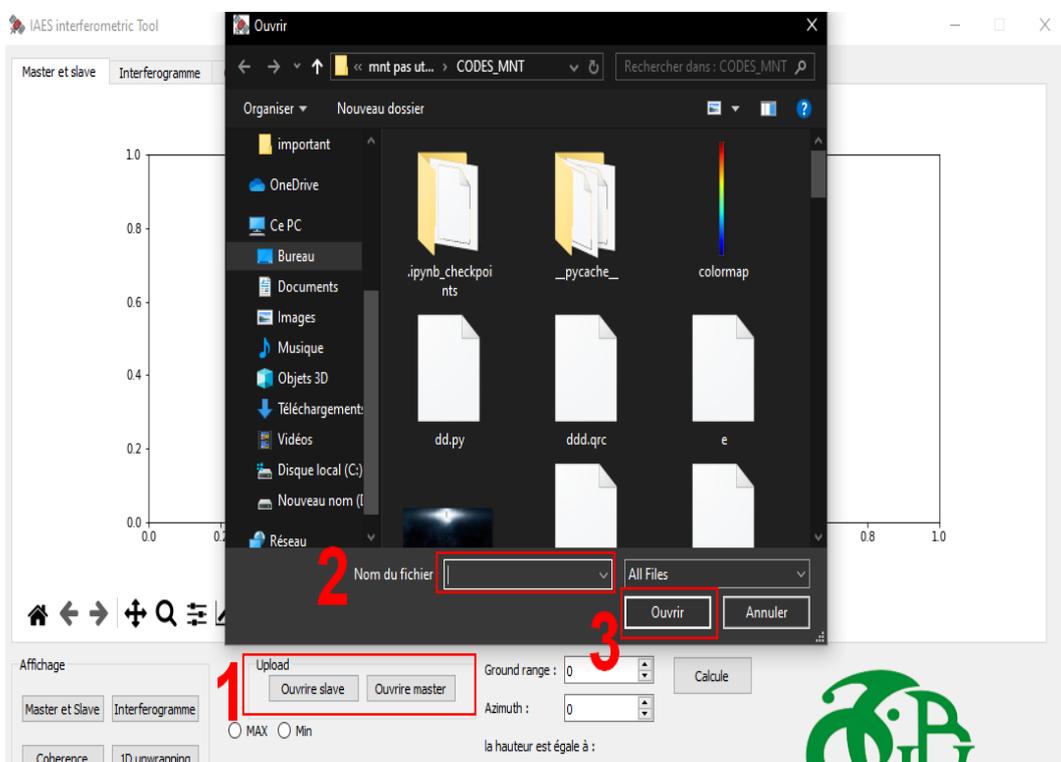


FIGURE 4.9 – Importation des images

Après on les affiche en utilisant matplotlib et on les orientes a leur fenêtre spécifique comme montre dans la figure suivante :

```
def plotmasterslave(self,Form):

    mstr_amp2 = 10 * np.log10( np.absolute( self.slave_fine ) * 2 )
    mstr_amp22 = n_contrast.all(mstr_amp2)
    mstr_amp1 = 10 * np.log10( np.absolute( self.master_fine ) * 2 )
    mstr_amp = n_contrast.all(mstr_amp1)
    self.widget_7 .canvas.axes.clear()
    self.widget_7 .canvas2.axes.clear()
    self.widget_7 .canvas.axes.imshow(mstr_amp, cmap=cm.gray)
    self.widget_7 .canvas.axes.set_title('Master',pad=15)
    self.widget_7 .canvas.axes.set_xlabel('Azimuth(pixel)')
    self.widget_7 .canvas.axes.set_ylabel('Ground Range(pixel)')
    self.widget_7 .canvas2.axes.imshow(mstr_amp22, cmap=cm.gray)
    self.widget_7 .canvas2.axes.set_title('Slave',pad=15)
    self.widget_7 .canvas2.axes.set_xlabel('Azimuth(pixel)')
    self.widget_7 .canvas2.axes.set_ylabel('Ground Range(pixel)')
    self.widget_7 .canvas.draw()
    self.widget_7 .canvas2.draw()
```

FIGURE 4.10 – Code Affichage Master et Slave

Et le résultat sera comme illustré dans la figure suivante :

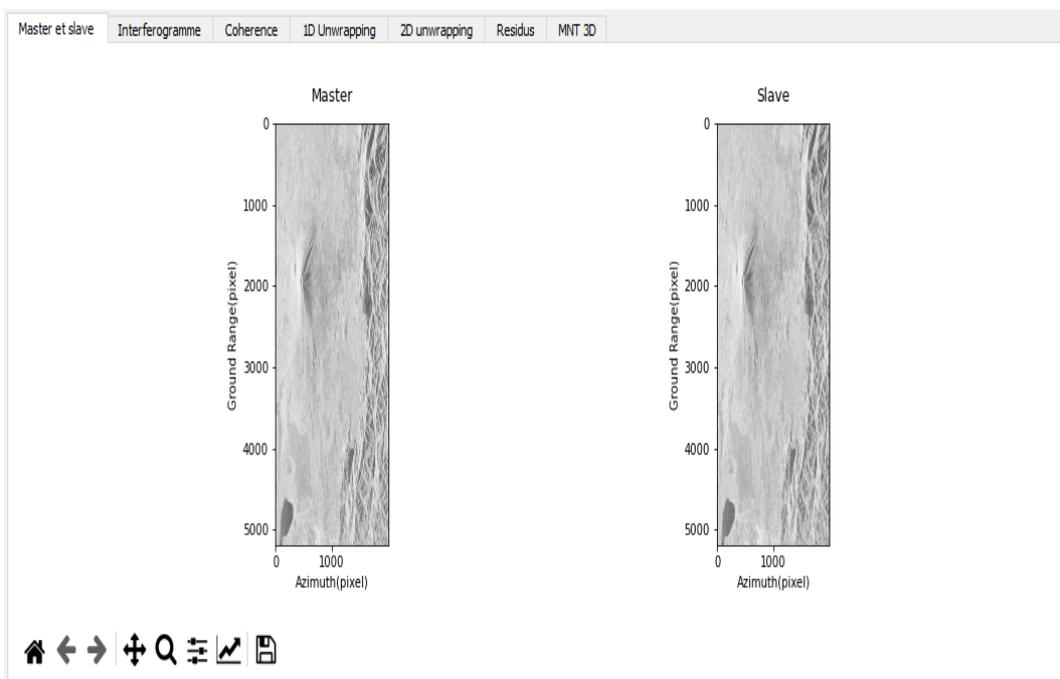


FIGURE 4.11 – Affichage Master et Slave

4.3.2 Génération de l'interférogramme

Comme expliqué dans le chapitre II, la génération de l'interférogramme se fait en utilisant la loi :

$$S1.S2 = A1.A2 \exp(j(\phi1 - \phi2)) \quad (4.1)$$

donc on l'a implémenté dans le programme .après avoir eu l'interférogramme, on soustrait la terre plate en utilisant le wrapping qui est fait pour rendre notre intervalle après la soustraction entre $-\pi$ et π , comme on le voit sur la figure suivante :

```
def plotifgrm(self, Form):
    def wrap(phase):
        wrapped = np.mod( phase + np.pi, 2 * np.pi ) - np.pi
        return(wrapped)
    self.ifgrm = np.angle( self.master_fine * np.conj(self.slave_fine) )
    # supprimé l'effet de la phase de la terre plate
    self.ifgrm_fer = wrap(self.ifgrm - self.modelphase)
    # afficher la phase de la terre plate et l'interférogramme

    self.MplWidget.canvas.axes.clear()
    self.MplWidget.canvas2.axes.clear()
    self.MplWidget.canvas3.axes.clear()
    self.MplWidget.canvas.axes.imshow(self.ifgrm, interpolation='nearest', cmap=cm.jet)

    self.MplWidget.canvas2.axes.imshow(self.modelphase, interpolation='nearest', cmap=cm.jet)
    self.MplWidget.canvas3.axes.imshow(self.ifgrm_fer, interpolation='nearest', cmap=cm.jet)
    self.MplWidget.canvas.draw()
    self.MplWidget.canvas2.draw()
    self.MplWidget.canvas3.draw()
```

FIGURE 4.12 – le code de wrapping

Donc on aura notre première image qui est l'interférogramme avec les valeurs d'origine

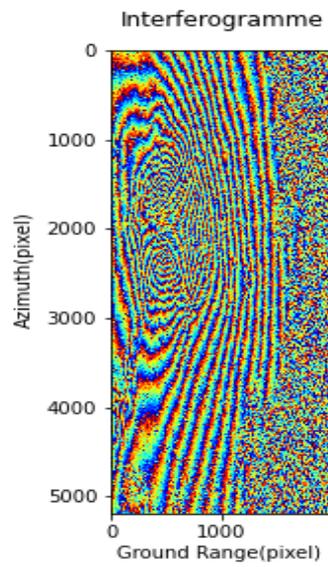


FIGURE 4.13 – Interférogramme

Après la suppression de la terre plate qui est illustrée dans la figure suivante :

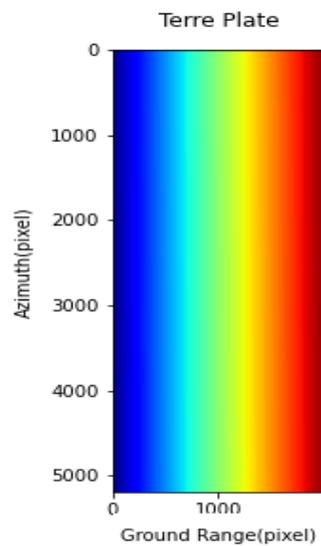


FIGURE 4.14 – représentation de la terre plate

on aura notre interférogramme sans la terre plate avec les informations qui auront un sens

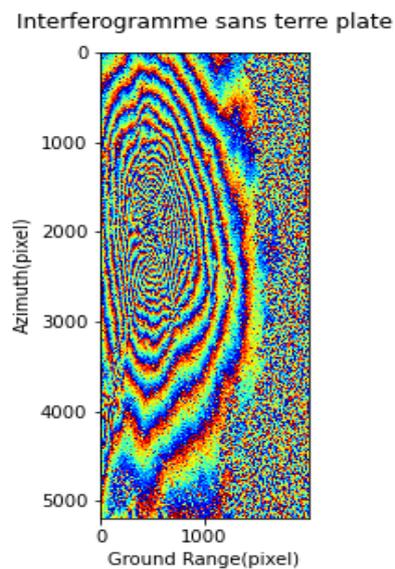


FIGURE 4.15 – Interférogramme sans la terre plate

4.3.3 Estimation de la cohérence

Après avoir terminé et eu notre interférogramme, on passe à la cohérence qui est expliquée dans le chapitre II . Pour cela on a utilisé la loi déjà expliquée auparavant, et on a mis en place une spin Box (figure suivante)



FIGURE 4.16 – SpinBox

qui a des valeurs entre 3 et 15 et prend que des valeurs impaires pour choisir la taille de notre fenêtre qui calcule la cohérence (code dans la figure suivante)

```
self.spinBox = QtWidgets.QSpinBox(Form)
self.spinBox.setGeometry(QtCore.QRect(470, 600, 42, 22))
self.spinBox.setObjectName("spinBox")
self.spinBox.setRange(5,13)
self.spinBox.setSingleStep(2)
```

FIGURE 4.17 – code SpinBox

```
###
#definie la taille de la window
h = self.spinBox.value()
self.label.setText('INFO - Estimating coherence - may take a while...')
w_rw = h*h
w_cl = h
```

FIGURE 4.18 – code SpinBox

Aussi, vu que le calcul de la cohérence Prend du temps, on a mis en place une ProgressBar qui évolue au même temps que l'évolution des calculs des boucles de cohérence. (code dans la figure suivante) :

```
self.progressBar = QtWidgets.QProgressBar(Form)
self.progressBar.setGeometry(QtCore.QRect(230, 640, 831, 23))
self.progressBar.setProperty("value", 24)
self.progressBar.setObjectName("progressBar")
self.progressBar.setValue(0)
```

FIGURE 4.19 – code de design progressbar

et là c'est le code de la fonction qui lui permet de progresser en même temps que les boucles.

```

if (kk/float(rw_stp) > complete_it):
    complete_it=complete_it+complete_step

self.progressBar.setValue(complete_it*100)

```

FIGURE 4.20 – code de fonction de progressbar

Après, on a utilisé une fonction qui nous permet de lancer les boucles de cohérence en arrière plan avec un deuxième cœur de processeur ,pour que l'interface soit toujours utilisable au moment du calcul. (code dans la figure suivante)

```

for mm in range(cl_stt, cl_stp+1):
    QtCore.QCoreApplication.processEvents()

```

FIGURE 4.21 – Fonction processEvents

Et à la fin, après que la progressbar est complète, sa nous affiche la cohérence et l'interférogramme sous l'onglet Cohérence du logiciel. (figure suivante)

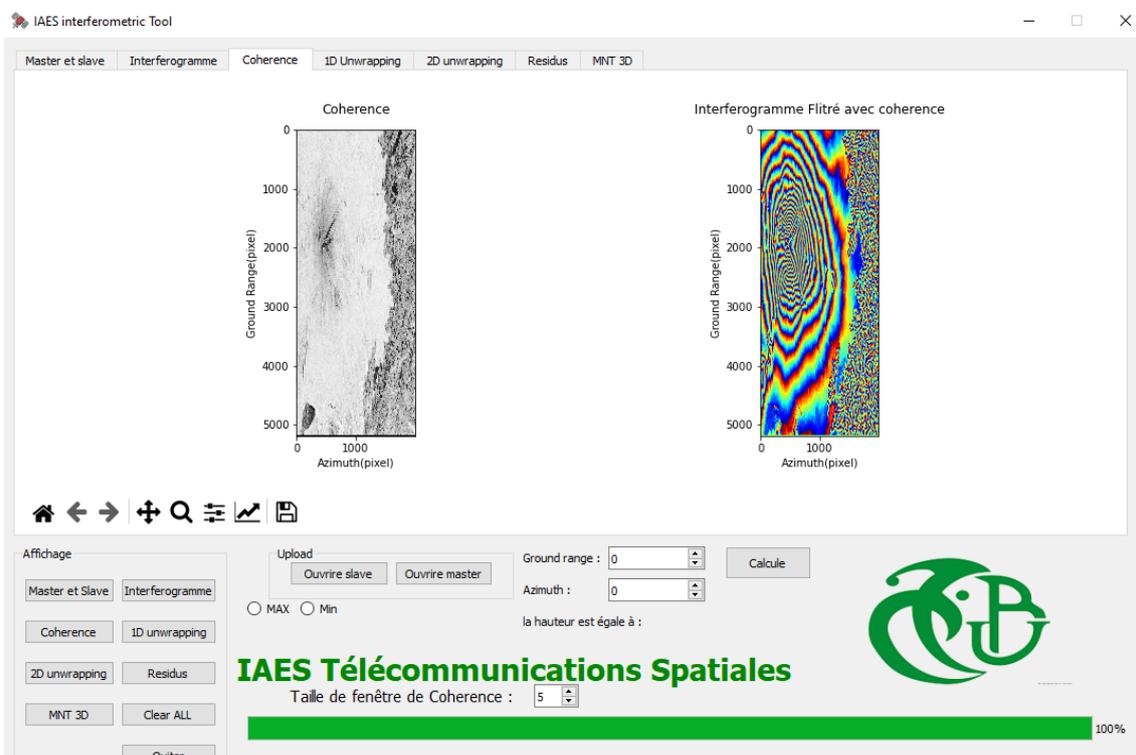


FIGURE 4.22 – Affichage de la cohérence

4.3.4 Déroulement de phase

Pour le déroulement de phase, on a utilisé la méthode ” unweighted Least mean Squares” comme expliqué dans le chapitre III, et pour cela on devait d’abord créer une fonction qui déroule la phase d’une seule dimension (ligne ou colonne).

```
def phase_unwrap_1D(initial_cycles, phase_1D):
    #définir dim comme nombre de dimension de phase_1D
    dim = phase_1D.ndim

    #verifie que ce qu'on comme entré "Phase_1D" a une seul dimension
    if (dim > 1):
        print('input must be one vector - no matrices allowed')
        sys.exit(1)
    else:

        #crée 2 vecteur decalé de 1 et les soustraire pour avoir la difference de phase
        phase_diff = phase_1D[1:] - phase_1D[0:-1]

        # detecter les saut de phase avec leur signe
        ph_up = np.where(phase_diff > np.pi, -1, 0)
        ph_down = np.where(phase_diff < -np.pi, 1, 0)

        # generé l'addition cumulative des saut qu'on a detecter pour avoir les cycles
        cycles_add = np.cumsum(ph_up + ph_down)

        # ajouter le premier element perdu a cause du decalage et le remplacé par 0
        cycles = np.hstack([0,cycles_add]) + initial_cycles

        # dérouler la phase en utilisant sa loi
        uw_phase = phase_1D + cycles * 2 * np.pi

        #return a dictionary with the results
        return({'uw_phase':uw_phase, 'uw_cycles':cycles})
        sys.exit(0)
```

FIGURE 4.23 – Code Fonction 1D

Après on applique notre fonction sur la ligne avant-dernière pour voir l’unwrapping et les cycles dans celle-ci.

```

# Affichage en utilisant les lignes

# sélectionné une ligne
self.rw, self.cl = self.coh_wphase.shape
cl_vec = np.array(range(0,self.cl,1))
rw_selected = self.rw-1 # selected line number (row)
# déroulement 1d de la ligne sélectionné
phase_1D = np.angle(self.coh_wphase[rw_selected,])
uw_phase_1D_dict = phase_unwrap_1D(0, phase_1D)

# Prendre nos donné calculer par notre fonction de 1d unwrap
uw_phase_1D = uw_phase_1D_dict['uw_phase']
uw_cycles_1D = uw_phase_1D_dict['uw_cycles']

```

FIGURE 4.24 – Code Fonction 1D

Alors On aura ce résultat qui représente 1D Unwrapping et les nombres des cycles qui sont dessus :

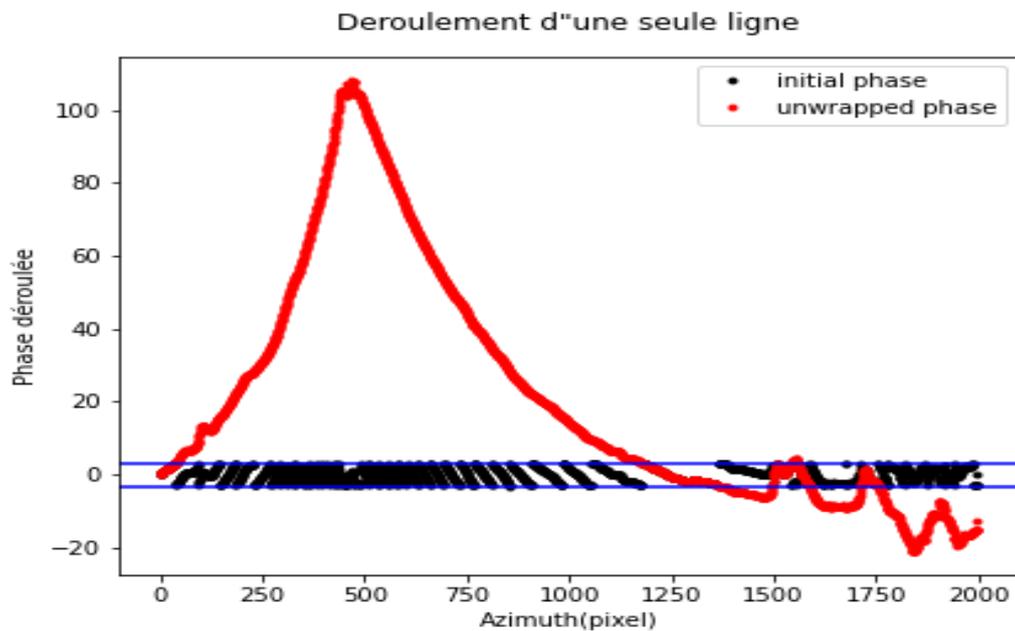


FIGURE 4.25 – Unwrapping d'une ligne

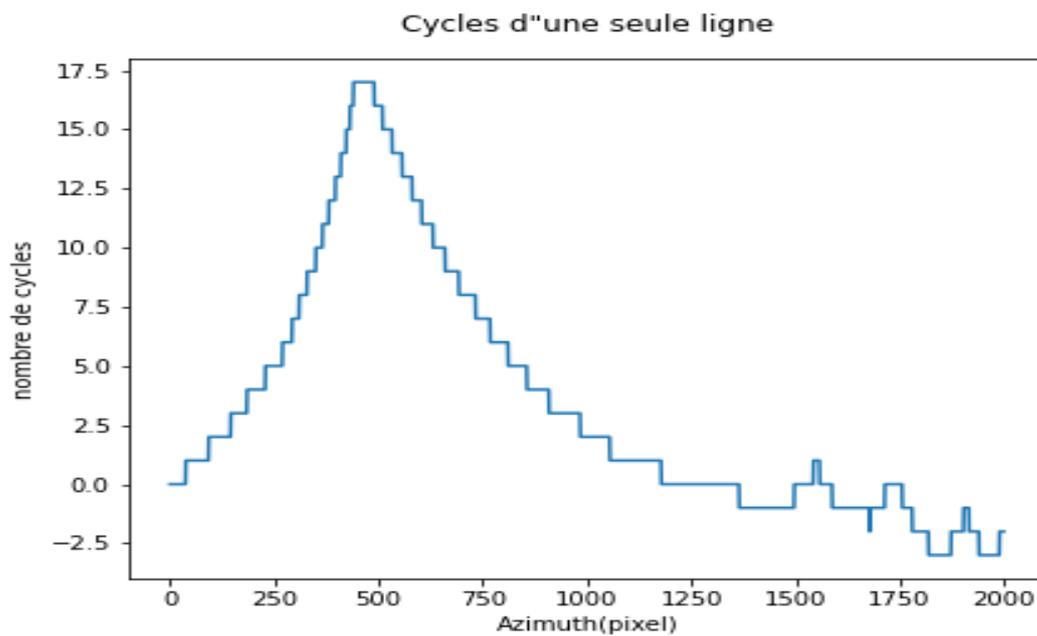


FIGURE 4.26 – Affichage de nombre de cycles

Après, a sélection la première colonne et on applique notre fonction dessus pour avoir les éléments initiaux de nos lignes.

```

self.phase_2D = np.angle(self.part)
rw_all, cl_all = np.shape(self.part);

# 1. déroulé la première colonne pour avoir la valeur initial de chaque ligne
phase_1D = self.phase_2D[:,0]
uw_phase_1D_dict = phase_unwrap_1D(0, phase_1D)
# 2. avoir les information donné par notre fonction de 1D unwrapping
final_phase_1st_cl = uw_phase_1D_dict['uw_phase']
cycles_1st_cl      = uw_phase_1D_dict['uw_cycles']

# 2. créer une matrice vierge
self.final_phase_2D_rw = np.zeros( (rw_all, cl_all) )

```

FIGURE 4.27 – Code calculs unwrapping première colonne

Après avoir eu les éléments initiaux de notre ligne, on passe à l'unwrapping 2D en utilisant les éléments initiaux de chaque colonne.

```

# 2. creer une matrice vierge

self.final_phase_2D_rw = np.zeros( (rw_all, cl_all) )
# 3. appliquer le PU pour chaque ligne en utilisant les donne initial de chaque colone
for k in range(0,rw_all):

    phase_1D = self.phase_2D[k,:]
    tmp_dict = phase_unwrap_1D( cycles_1st_cl[k], phase_1D )
    self.final_phase_2D_rw[k,:] = tmp_dict['uw_phase']

```

FIGURE 4.28 – Code pour calculer toutes les lignes

et tous le travail sera refait de la même façon juste en inversons, on déroule la 1er ligne et après on déroule les colonnes, en utilisant les données initiaux de chacune d'elles qu'on a calculés en déroulons la première ligne.

```

# Affichage en utilisant les colone
#1. déroulé la premiere colone pour avoir la valeur initial de chaque ligne
phase_1D = self.phase_2D[0,:]
uw_phase_1D_dict = phase_unwrap_1D(0, phase_1D)

final_phase_1st_rw = uw_phase_1D_dict['uw_phase']
cycles_1st_rw      = uw_phase_1D_dict['uw_cycles']

# 2. creer une matrice vierge
self.final_phase_2D_cl = np.zeros( (rw_all, cl_all) )
# 3. appliquer le PU pour chaque colone en utilisant les donne initial de chaque ligne
for k in range(0,cl_all):
    phase_1D = self.phase_2D[:,k]
    tmp_dict = phase_unwrap_1D( cycles_1st_rw[k], phase_1D )
    self.final_phase_2D_cl[:,k] = tmp_dict['uw_phase']

```

FIGURE 4.29 – Code pour calculer toutes les colonnes

Pour les résultats on aura les deux images d'unwrapping (par colonne, par ligne) comme illustré dans la figure suivante :

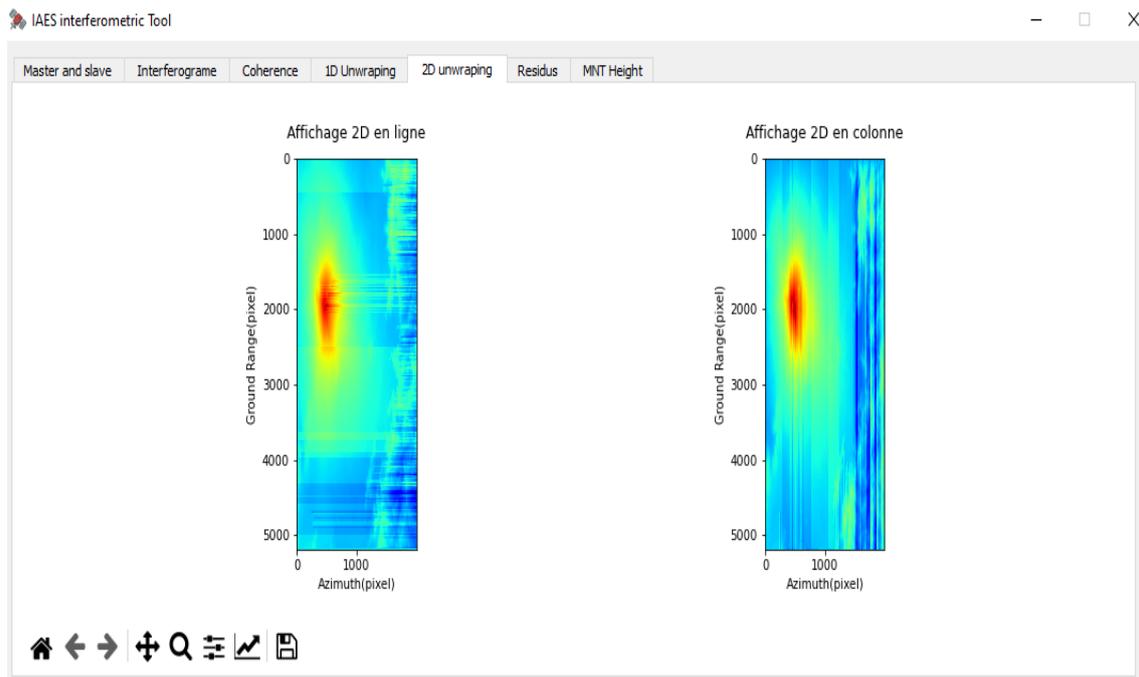


FIGURE 4.30 – Affichage unwrapping 2D

4.3.5 Affichage 3D

Après avoir eu notre phase réelle on a mis en place des fonctions qui nous permettent d'afficher nos résultats et même avoir la hauteur de chaque pixel pour cela on devait avoir **P2H** qui représente la hauteur réelle par rapport à chaque cycle (c'est une donnée spéciale pour chaque radar Sar) dans notre cas elle est égale a :

$$P2h = \frac{167.89}{\frac{2}{\pi}} \quad (4.2)$$

et on la multiplie par notre phase déroulée (des lignes dans notre cas) alors on aura la hauteur réelle qu'on va implémenter dans une figure 3D avec ground range comme axe y est azimuth comme x et hauteur comme axe z et on aura la toolbar déjà implémentée pour faire diverses manipulations voici le code pour la réaliser ;

```
def heighmap(self) :  
    rw_all, cl_all = np.shape(self.part);  
    xlin = range(0,cl_all)  
    ylin = range(0,rw_all)  
    xmesh, ymesh = np.meshgrid(xlin, ylin)  
  
    self.widget_6.canvas.axes.clear()  
    self.widget_6.canvas.axes.plot_surface( xmesh, ymesh, self.final_phase_2D_rw*self.p2h, cm  
    self.widget_6.canvas.axes.set_title('Affichage 3D ',pad=15)  
    self.widget_6.canvas.axes.set_xlabel('Azimuth(pixel)')  
    self.widget_6.canvas.axes.set_ylabel('Ground Range(pixel)')  
    self.widget_6.canvas.axes.set_zlabel('hauteur(m)')  
    self.widget_6.canvas.draw()
```

FIGURE 4.31 – Code Affichage unwrapping 3D avec hauteur

Alors on aura ce résultat :

Affichage 3D

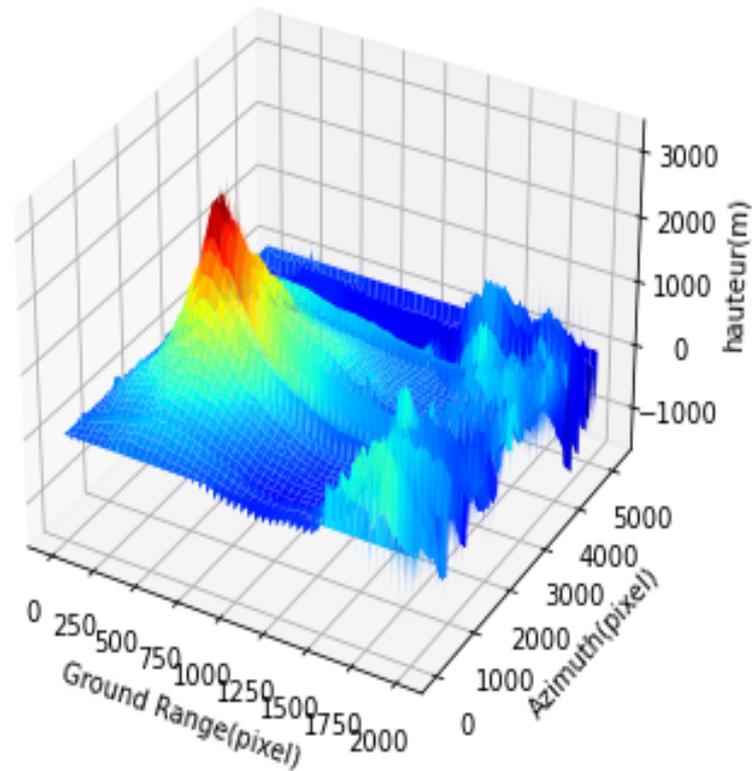


FIGURE 4.32 – Affichage unwrapping 3D avec hauteur

Avec la toolbar on pourra faire tourner notre MNT pour le voir d'un autre point de vue .
Aussi on peut avoir la hauteur de chaque point et chaque pixel sur notre MNT en utilisant le curseur et en se déplaçant sur notre map 3D comme illustrée dessus :

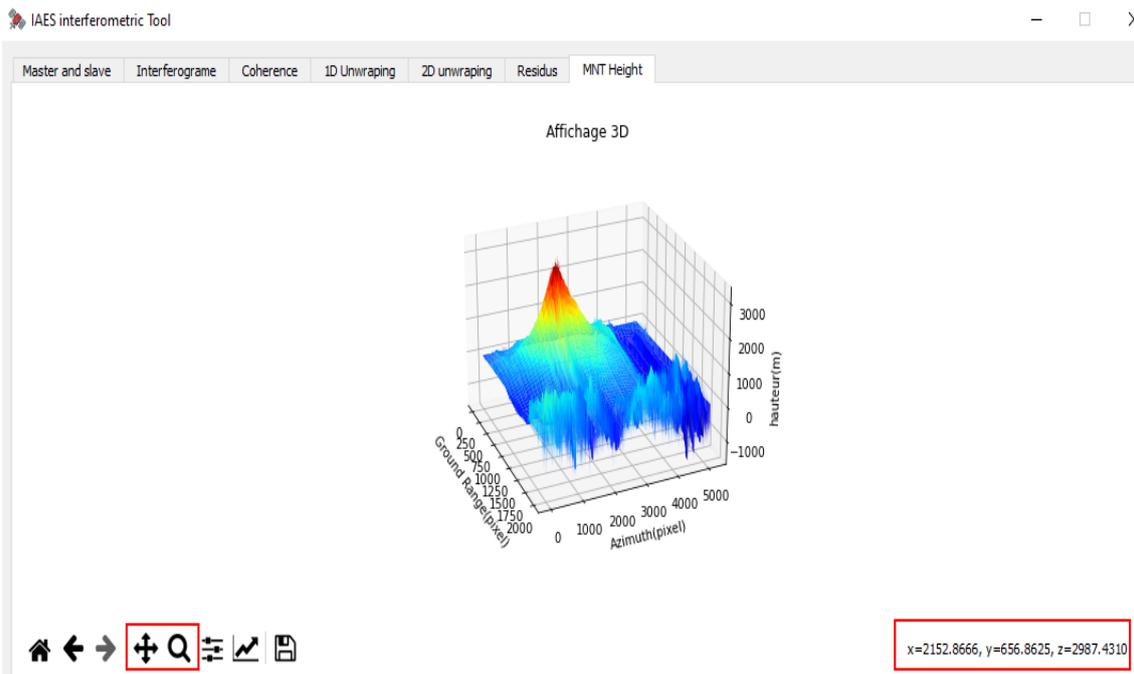


FIGURE 4.33 – hauteur en utilisant la toolbar

4.3.6 Résidus

Pour la détection des résidus on a utilisé la loi comme expliquer dans le chapitre 3, celle du chemin fermé de 2×2 pixels comme montré dans la figure suivante :

```

def residus(ifgrm):
    def wrap(phase):
        wrapped = np.mod( phase + np.pi, 2 * np.pi ) - np.pi
        return(wrapped)
    rw, cl = ifgrm.shape

    res_phase = np.zeros( (rw, cl), dtype=np.double )

    # code pour calculer des lignes entières à la fois

    # code pour calculer des lignes entières à la fois
    for k in range(0, rw_all-1):
        D_ur_ul = wrap( ifgrm[k, 1:] - ifgrm[k, 0:-1] ) #diff: (0,1) - (0,0)
        D_lr_ur = wrap( ifgrm[k+1, 1:] - ifgrm[k, 1:] ) #diff: (1,1) - (0,1)
        D_ll_lr = wrap( ifgrm[k+1, 0:-1] - ifgrm[k+1, 1:] ) #diff: (1,0) - (1,1)
        D_ul_ll = wrap( ifgrm[k, 0:-1] - ifgrm[k+1, 0:-1] ) #diff: (0,0) - (1,0)
        # residual phase: sum up
        res_phase[k, 0:-1] = D_ur_ul + D_lr_ur + D_ll_lr + D_ul_ll

    # cycles de residus
    res_cycles = np.round( res_phase / (2*np.pi) )

    return({'residual_phase':res_phase, 'residual_cycles':res_cycles}) #return dictiona
    sys.exit(0)

residus_dict = residus(self.phase_2D)
residus = residus_dict['residual_cycles']

(self.rx, self.ry) = (residus!=0).nonzero()

```

FIGURE 4.34 – code pour rechercher des résidus

avec `numpy.nonzero()` on aura tous les points où les résidus ne sont pas égaux à zéro et après on plot notre phase déroulée et on plot dessus nos points de résidu rx et ry en utilisant le code suivant :

```
def resid(self):

    self.w5.canvas.axes.clear()
    self.w5.canvas.axes.imshow(self.phase_2D, interpolation='nearest', cmap=cm.jet)
    self.w5.canvas.axes.plot(self.ry, self.rx, 'k.')
    self.w5.canvas.axes.set_title('Residu',pad=15)
    self.w5.canvas.axes.set_xlabel('Azimuth(pixel)')
    self.w5.canvas.axes.set_ylabel('Ground Range(pixel)')
    self.w5.canvas.draw()

    self.w5.canvas2.axes.clear()
    self.w5.canvas2.axes.imshow(self.final_phase_2D_rw, interpolation='nearest', cmap=cm.jet)
    self.w5.canvas2.axes.plot(self.ry, self.rx, 'k.')
    self.w5.canvas2.axes.set_title('Residu',pad=15)
    self.w5.canvas2.axes.set_xlabel('Azimuth(pixel)')
    self.w5.canvas2.axes.set_ylabel('Ground Range(pixel)')

    self.w5.canvas2.draw()
```

FIGURE 4.35 – code pour afficher les résidus

et voici le résultat obtenu :

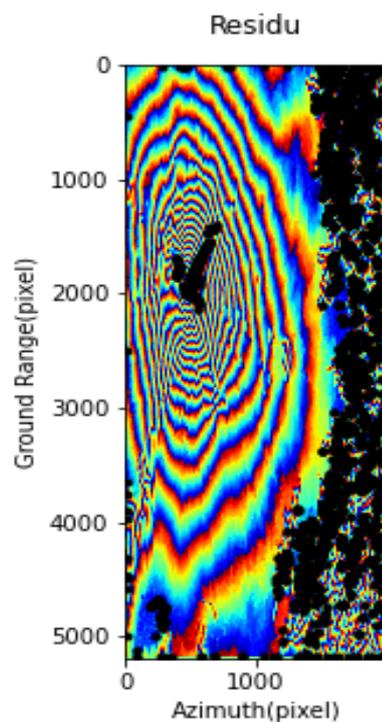


FIGURE 4.36 – code pour afficher les résidus

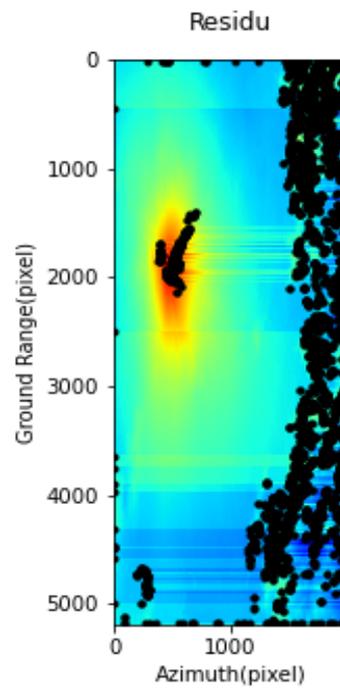


FIGURE 4.37 – code pour afficher les résidus

afri2

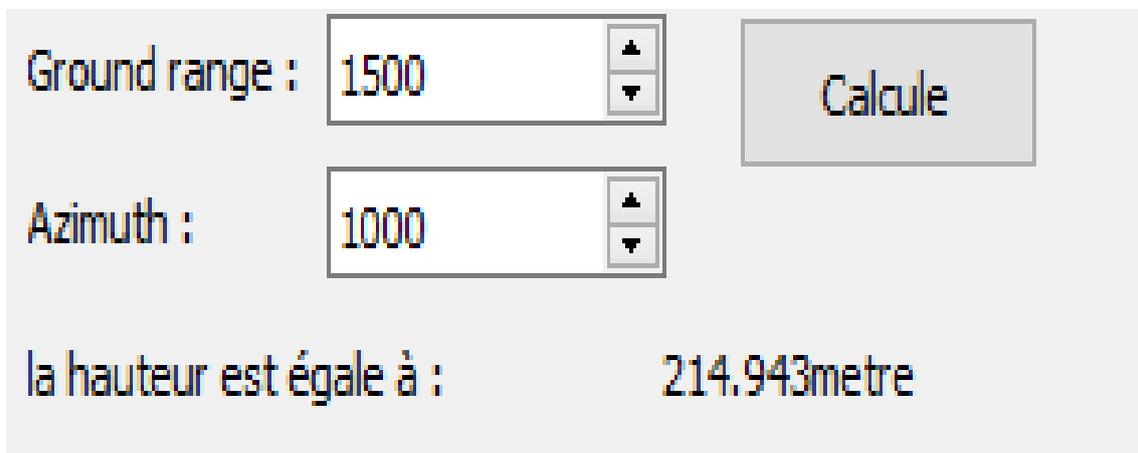
4.3.7 Calcules

on a mis en place aussi une fonction qui nous permet d'avoir la hauteur de chaque pixel (deux chiffres après la virgule) en entrant l'azimut et le ground range voulus et voici le code utilisé;

```
def calc(self) :  
  
    azimu = self.spinBox_2.value()  
    grow = self.spinBox_3.value()  
    hauteur = self.final_phase_2D_rw[azimu][grow]*self.p2h  
  
    resultttt = Decimal(hauteur)  
    roundresult = resultttt.quantize(Decimal('.001'), rounding=ROUND_HALF_UP)  
  
    self.resultatt.setText(str(roundresult)+'metre')
```

FIGURE 4.38 – code pour rechercher la hauteur d'un point

et voici le résultat donné :



Ground range : 1500

Azimuth : 1000

la hauteur est égale à : 214.943metre

FIGURE 4.39 – Calcul de Hauteur

et pour finir on a ajouté une fonction qui nous permet d'avoir :

- La hauteur maximale
- La hauteur minimale
- l'azimuth et le ground range de la hauteur maximale
- l'azimuth et le ground range de la hauteur minimale

et Pour cela on utilise ce code :

```

maxo=np.amax(self.final_phase_2D_rw*self.p2h)
result = np.where(self.final_phase_2D_rw*self.p2h == np.amax(self.final_phase_2D_rw*self.p2h),
                  self.final_phase_2D_rw*self.p2h,
                  None)
l=result[0]
g=result[1]

mino= np.amin(self.final_phase_2D_rw*self.p2h)
resultttt = np.where(self.final_phase_2D_rw*self.p2h == np.amin(self.final_phase_2D_rw*self.p2h),
                     self.final_phase_2D_rw*self.p2h,
                     None)
l2=resultttt[0]
g2=resultttt[1]
if self.radioButton.isChecked() :
    self.azimutandground.setText('ground range :'+str(l))
    self.label_8.setText('azimuth :'+str(g))
    self.resultatt.setText(str(maxo))
if self.radioButton_2.isChecked():
    self.azimutandground.setText('ground range :'+str(l2))
    self.label_8.setText('azimuth :'+str(g2))
    self.resultatt.setText(str(mino))

```

FIGURE 4.40 – Code de Calcul de la Hauteur maximale et minimale

et on a mis en place des radiobutton pour choisir entre la hauteur maximale et minimale, et voilà le résultat final :

Upload

MAX Min

ground range : [1970]
 azimuth : [507]

Ground range : 0
 Azimuth : 0

la hauteur est égale à : 3281.039497644866

FIGURE 4.41 – Calcul de la Hauteur maximale et ses coordonnées

Et pour le minimum dans ce cas on a plusieurs points et non pas un seul, et c'est des points de résidus (on peut les remarquer en bas de l'interférogramme) alors on aura ce résultat :

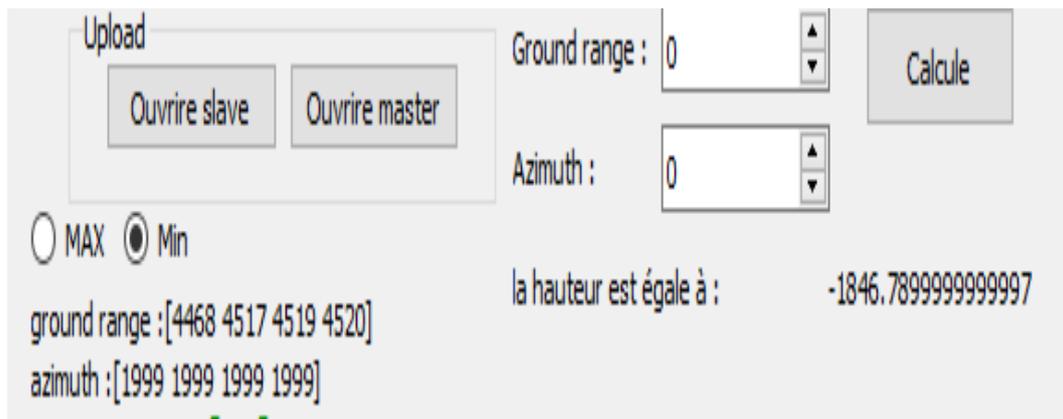


FIGURE 4.42 – Calcul de la Hauteur minimale et ses coordonnées

4.4 Conclusion

Dans ce chapitre, on est passé à la partie pratique de notre thèse, où on a réalisé un logiciel qui permet de mettre en œuvre tous les techniques décrites auparavant, alors on a commencé par la création d'une interface graphique qui comporte tous les boutons et la fenêtre nécessaire avec une programmation orientée objet, après on est passé aux fonctions qui vont nous permettre de générer le MNT, et après on les a relié à leurs boutons et fenêtres d'affichage spécifique .

À la fin on a eu un logiciel qui comporte tous les processus nécessaire pour la génération de notre MNT.

Conclusion Générale

Les images interférométriques SAR acquises au départ, ne sont pas exploitables et ne nous permettent pas d'avoir des informations réelles sur la terre, car elles forment des franges correspondant à un tour de phase de 2π . Aussi cette dernière contient des incohérences et des résidus (point de discontinuité) qui nous empêchent d'avoir un modèle numérique de terrain correcte. Il fallait résoudre ces problématiques en utilisant les algorithmes de déroulement de phase et les bases de traitement d'image. Alors pour résoudre sa On a crée un logiciel avec une interface graphique, en utilisant PyQt5 avec le langage python.

On a commencé avec la génération de l'interférogramme, et la cohérence pour avoir une idée sur la qualité de notre acquisition. Après cette étape il fallait passer au déroulement de phase, où on trouve plusieurs techniques. La technique qu'on a implémenté est (Un-weighted Least mean Squares), celle-ci nous a permis d'avoir deux versions MNT 2 D, une avec (colonne wise) et l'autre avec (line wise), et après on a créé une troisième matrice qui prend les cycles qu'on a obtenu avec notre méthode de déroulement et la rend en hauteur réelle en utilisant le rapport de la hauteur par rapport à chaque cycle (celle-ci étant donné de base avec les données du satellite comme la terre plate). Pour une meilleur visualisation de la topographie de la surface terrestre on a réalisé un affichage en 3D sur python en utilisant (x comme azimuth, y comme ground range et z comme hauteur réelle), mais tous sa avec des résidus surtout aux abords de notre MNT, alors on a implémenté un algorithme qui peut désigner les positions des résidus sur le MNT final.

Ce projet de fin d'étude nous a permis d'aborder et d'acquérir plusieurs compétences dans le domaine de la télédétection, des technologies RADAR et aussi sur le langage de programmation python et sa relation avec le traitement d'image. Pour réaliser tous ce travail, on a dû apprendre les base des RADAR et plus précisément le SAR (Radar a synthèse d'ouverture) en abordant son fonctionnement, sa façon d'acquisition, les types d'images qu'il nous donne, ses avantage et ses inconvenant dans la télédétection et son modèle géométrique.

En perspective et en complément de ce travail, on propose au prochain master de développer

notre logiciel en ajoutant les autres techniques de déroulement de phase et l'utilisation du différentiel InSAR, et même sortir de l'interférométrie et passé à la polarimétrie qui est une technique permettant d'exploiter pleinement les ondes radar en donnant plus de propriété physique sur la zone terrestre étudiée, comme la végétation, les terres agricoles, les minéraux, l'urbanisation et la profondeur des couvertures neigeuses

Bibliographie