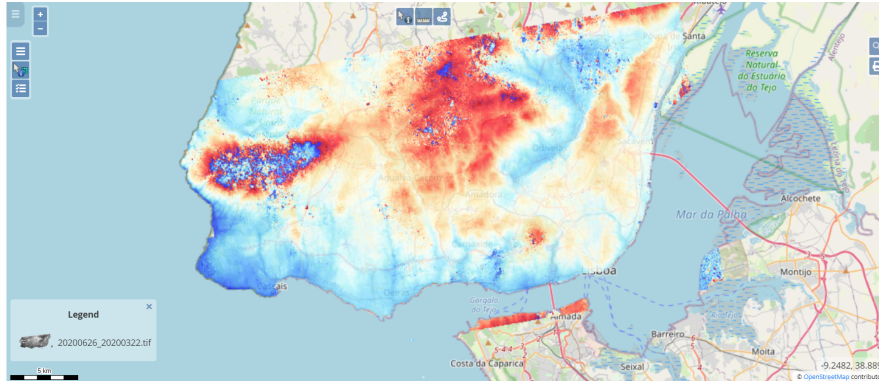




TÉCNICO
LISBOA



Automatização do Processo Interferométrico para Visualização e Gestão de Dados Espaciais

Steven Edward Coelho dos Santos

Dissertação para obtenção do Grau de Mestre em

Engenharia Aeroespacial

Orientador(es): Prof. Agostinho Rui Alves da Fonseca
Prof. Ana Paula Martins Falcão Flôr

Júri

Presidente: Prof. Fernando José Parracho Lau
Orientador: Prof. Ana Paula Martins Falcão Flôr
Vogal: Doutora Ana Maria Fonseca

Janeiro de 2021

Declaro que o presente documento é um trabalho original da minha autoria e que cumpre todos os requisitos do Código de Conduta e Boas Práticas da Universidade de Lisboa.

Agradecimentos

Trilhar este caminho cheio de obstáculos, desafios, surpresas e vivências, apenas foi possível com o apoio, energia e a força de várias pessoas, a quem dedico especialmente este projeto.

Primeiramente gostaria de agradecer a minha família, em especial, aos meus pais que tanto se esforçaram e sacrificaram para permitir-me estar onde estou e alcançar os meus objetivos. Agradeço pelos valores que desde cedo transmitiram-me, permitindo ser a pessoa que hoje sou.

Aos meus orientadores, Professora Ana Paula Martins Falcão Flôr e Professor Agostinho Rui Alves da Fonseca, que ao longo deste período acompanharam, orientaram e disciplinaram este trabalho, um muito obrigado. Obrigado pela confiança depositada em mim e obrigado por todo o incentivo e exemplo de disciplina e profissionalismo demonstrado.

Às instituições, nomeadamente à Escola Técnica João Varela e o Instituto Superior Técnico, que me deram as ferramentas necessárias para chegar tão longe nesta caminhada e a todos os professores que, de algum modo tiveram impacto em mim e permitiram-me através dos ensinamentos, percorrer este caminho.

Aos meus amigos que presenciaram e acompanharam-me ao longo desta grande jornada e que sempre me fizeram sentir em casa mesmo estando longe, principalmente nos momentos mais difíceis e aos que por circunstâncias da vida não conseguiram estar presentes fisicamente, mas contribuíram e acompanharam este trajeto de igual modo, um enorme obrigado. Um agradecimento especial ao Alexandre Barros, Alicia Monteiro, Bruno Gil, Eline Barbosa, Evelise da Cruz, Hernany Rocha, Janiza Silva, Josiane Spencer, Juary Spencer, Larissa Duarte, Leonel Lisboa, Marco Almeida, Mariline Barbosa, Pedro Spencer e Tiara de Pina.

Aos meus amigos do Mestrado Integrado em Engenharia Aeroespacial e para a vida toda que me acompanharam ao longo destes implacáveis e fantásticos anos do Técnico, um grande obrigado. Foram vários os momentos que servimos de suporte uns dos outros, que partilhamos nossas dores e nossas alegrias, momentos que nos fizeram tornar uma família. A saber, Délio Bengaló, João Soares, José Fernandes, Yannick Duarte e Yuri Fontes.

Por último, mas não menos importante, um gigantesco obrigado às minhas duas famílias C1 e C2. Partilhar um teto com tanta gente nem sempre é uma tarefa fácil, passamos sempre por situações ou momentos menos felizes, mas em contrapartida, servimos de suporte, apoiamos uns aos outros, partilhamos momentos e ensinamentos que marcam as nossas vidas e permitem-nos ser melhores cidadãos e pessoas. Só tenho a agradecer o apoio e suporte ao longo desses anos e dizer que foi um prazer ter dividido um teto convosco. Um agradecimento especial ao Afonso Luz, Alice Lourenço, Ariana Marta, Carlos Pires, Catarina Machuqueiro, Cátia Franco, Cristiano Mesquita, Diogo Pires, Gabriel Carvalho, Gonçalo Cardoso, Gonçalo Pinho, Gonçalo Pereira, João Faria, João Lobo, João Pedro Gomes, Jorge Cerejo, José Coelho, José Costa, Leonardo Pedroso, Lucas Munhão, Mário Silva, Matheus Orsi, Miguel Rebocho, Pedro Acácio, Pedro Pereira, Renata Quintino, Sara Ramos, Tiago Costa, Tiago Nunes e Vera Dias.

Resumo

A diversidade de sensores que captam imagens da superfície terrestre tem vindo a aumentar, gerando grandes volumes de informação que importa considerar em estudos de vulnerabilidade e de mitigação de riscos naturais. A integração num visualizador único das áreas captadas é da máxima importância numa lógica de otimização de meios e recursos.

Neste trabalho dá-se preferência a imagens captadas por radares de abertura sintética (*Synthetic Aperture Radar* - SAR), por serem sistemas ativos e pela independência em relação às condições meteorológicas. De entre os vários sensores que atualmente operam nos comprimentos de onda do radar, a plataforma *Sentinel-1*, pertencente ao programa, *Copernicus* serviu de base ao desenvolvimento desta dissertação.

A diferença de fase entre duas imagens obtidas com geometrias de aquisição diferentes, para um mesmo elemento retrodispersor, permite a determinação da altitude do objeto, numa técnica designada de interferometria e a sua utilização conjuntamente com imagens SAR tem por nome, Interferometria Radar de Abertura Sintética (*Interferometric Synthetic Aperture Radar* - InSAR). Trata-se de uma técnica capaz de medir alterações na escala de milímetros com alta resolução espacial e temporal. Esta diferença de fase contém várias contribuições, dos quais se destacam as condições atmosféricas, um fator perturbador, principalmente, quando se tem por objetivo comparar interferogramas.

O principal objetivo deste trabalho passa pela automatização do processo interferométrico desde o *download* das imagens SAR à disponibilização dos interferogramas, corrigidos dos efeitos da atmosfera, numa plataforma de visualização, tendo para isso sido criado um conjunto de scripts, capazes de desempenhar tais tarefas.

Palavras-chave: Interferometria SAR, Radar de Abertura Sintética, Sentinel-1, SNAP, GACOS, Infraestruturas de Dados Espaciais.

Abstract

In last years, the number of sensors that capture images of land surface has increasing, as a result of a strong bet of many countries and space agencies. Due to that, large amount of data has been recorded and their integration in a single viewer platform is of utmost importance to support informed decisions.

Images captured by Synthetic Aperture Radars (SAR) sensors have been selected in this study, as they are acquired by active sensors and more independent of meteorological conditions. Among of the various sensors that operate in radars' wavelength, the Sentinel-1 mission, part of the Copernicus program, was used as the base to the development of this thesis.

The phase difference between two images obtained with different acquisition geometries, for the same scatter target, allows the determination of object's altitude, based on interferometric technique and its application to SAR images it is known as Interferometric Synthetic Aperture Radar (InSAR) technique, capable of measure displacements in the scale of millimeters with high special and temporal resolutions. This phase difference contains several contributions such as the topography and the atmosphere effect, which acts as perturbation factor, especially, when the goal is to compare interferometric products.

The main objective of this study was the automatization of the interferometric process, starting with the download of SAR images until their visualization in a WEBGIS platform. To reach this, a set of scripts, capable of perform the necessary tasks, was created.

Keywords: SAR Interferometry, Synthetic Aperture Radar, Sentinel-1, SNAP, GACOS, Spatial data infrastructure.

Conteúdo

Agradecimentos	v
Resumo	vii
Abstract	ix
Lista de Tabelas	xv
Lista de Figuras	xvii
Lista de Acrónimos	xix
Lista de Símbolos	xxiii
1 Introdução	1
1.1 Motivação	2
1.2 Objectivos	2
1.3 Estrutura da Dissertação	2
2 Revisão de conceitos básicos de SAR, InSAR e plataformas de dados espaciais	5
2.1 Imagens SAR	6
2.1.1 Geometria	6
2.1.2 Comprimentos de onda e polarização	8
2.1.3 Efeitos da geometria do terreno	9
2.2 Missões SAR	10
2.2.1 Sentinel 1	10
2.3 Interferometria SAR	12
2.3.1 Relação entre a diferença de fase e a altitude	13
2.3.2 Etapas do processo interferométrico	13
2.4 Plataformas de dados espaciais	17
2.4.1 Infraestruturas de Dados Espaciais	17
2.4.2 Exemplos de plataformas de dados espaciais	20
3 Metodologia Desenvolvida	23
3.1 <i>Download</i> dos produtos SAR	24
3.2 Processamento das imagens SAR	24
3.2.1 Processamento interferométrico	26
3.2.2 Correção atmosférica	30

3.3	Comparação entre os Interferogramas corrigidos	31
3.4	Visualização dos dados espaciais	31
4	Caso de estudo	33
4.1	Enquadramento Geográfico	33
4.2	Processamento das Imagens SAR	34
4.2.1	Requisitos de sistema	34
4.2.2	Imagens SAR	35
4.2.3	Processamento Interferométrico e correção atmosférica	36
4.3	Comparação entre os Interferogramas corrigidos	38
4.4	Visualização dos dados espaciais	39
4.4.1	Plataforma iMSEP	39
5	Conclusão	41
5.1	Perspectivas de desenvolvimentos futuros	42
	Bibliografia	43
A	Imagens SAR	49
A.1	Modos de aquisição de imagens SAR e respectivos níveis	49
A.1.1	Modos de aquisição	49
A.1.2	Níveis	50
A.2	Séries Temporais imagens SAR	52
B	Códigos e Scripts	53
B.1	<i>Download</i> imagens SAR	53
B.1.1	<i>dhusget script</i>	53
B.1.2	<i>download.py</i>	53
B.2	SNAP <i>Graph Builder</i>	56
B.3	Processamento interferométrico	58
B.3.1	<i>slaves_prep.py</i>	58
B.3.2	<i>splitting.py</i>	59
B.3.3	<i>ifg_formation.py</i>	63
B.3.4	<i>unw_tc.py</i>	66
B.3.5	<i>ifg_correction_prep.py</i>	71
B.3.6	<i>ztd_correction_prep.py</i>	72
B.3.7	<i>correction_prep.py</i>	73
B.3.8	<i>stamps_export.py</i>	77
B.4	Correcção atmosférica	79
B.4.1	<i>main_auto_atm_correction.m</i>	79
B.4.2	<i>auto_atm_correction.m</i>	80

B.4.3	check_diference.m	82
-------	-----------------------------	----

Lista de Tabelas

2.1	Comprimento de onda e frequência das microondas.	8
2.2	Modos de captura de imagens SAR e as respectivas resoluções e coberturas	11
2.3	Modos de captura de imagens SAR e os respectivos níveis	11
4.1	Características da área de estudo.	33
4.2	Características das imagens SAR da área de estudo.	35
A.1	Série Temporal de imagens SAR.	52

Lista de Figuras

2.1	Geometria de aquisição de informação dos satélites SAR.	6
2.2	Convenções de nomenclatura angular no sistema de imagem de radar.	7
2.3	Distorções radiométricas em imagens de radar.	7
2.4	Satélites ou missões SAR existentes e planeadas, e as respectivas bandas espectrais.	8
2.5	Vista lateral e frontal do campo eléctrico de uma onda electromagnética.	9
2.6	Distorções geométricas em imagens de radar.	10
2.7	Satélite da missão Sentinel 1	11
2.8	Geometria de um sistema de satélite interferométrico SAR.	12
2.9	Ilustração do registo entre duas imagens SAR.	14
2.10	Interferograma da ilha do Fogo, Cabo Verde, produzida através das imagem adquiridas pelo satélite Sentinel 1, em 3 de Novembro de 2014 e 27 de Novembro de 2014.	14
2.11	Imagem de Coerência da zona Lisboa produzida através das imagem adquiridas pelo satélite Sentinel 1, em 21 de Maio de 2020 e 26 de Junho de 2020.	15
2.12	Ilustração do processo de desenrolamento interferométrico.	16
2.13	Estrutura hierárquica de uma IDE.	17
2.14	Arquiteturas orientadas por serviços.	19
2.15	Esquema padrão da arquitetura das Infraestruturas de dados espaciais.	19
2.16	Plataforma InSAR da Noruega.	20
2.17	Geoportal Italiano.	21
3.1	Fluxograma da metodologia utilizada	23
3.2	Processamento das imagens SAR.	24
3.3	Estrutura e organização dos directórios.	25
3.4	Subdirectório bin com os Python <i>scripts</i>	26
3.5	Fluxograma do processamento interferométrico.	26
3.6	Fluxograma do processamento interferométrico condensado.	27
3.7	Gráficos xml referente ao <i>Splitting</i> das imagens SAR e actualização de orbitas.	28
3.8	Gráfico xml referente ao Co-registo e formação dos interferogramas.	28
3.9	Gráficos xml referente ao Desenrolamento de fase e correcção topográfica.	29
3.10	Gráficos xml referente ao Co-registo, formação dos interferogramas e exportação para o STAMPS.	29

3.11 Gráficos xml do processo interferométrico.	29
3.12 Fluxograma da correcção atmosférica.	30
3.13 Subdirectório <i>matlab</i> com os <i>scripts MATLAB</i>	31
4.1 Área de estudo.	34
4.2 Ficheiro de configuração para download de imagens SAR - <i>download.conf</i>	35
4.3 Ficheiro de configuração do processamento de imagens - <i>project.conf</i>	36
4.4 Interferogramas antes e após a correcção atmosférica referentes aos últimos produtos SAR processados.	37
4.5 Índice de Similaridade estrutural entre os interferogramas corrigidos sequencialmente.	38
4.6 Erro quadrático médio entre os interferogramas corrigidos sequencialmente.	38
4.7 Visualização parcial dos interferogramas corrigidos na plataforma iMSEP.	39
A.1 Modos de aquisição das imagens dos satélites Sentinel 1.	49
A.2 Níveis dos produtos Sentinel 1.	50
B.1 Alteração no operador <i>Read</i> do ficheiro <i>coreg_ifg_mlflt_subset.xml</i>	57

Lista de Acrónimos

AMTI *Airborne Moving target indicator*

ASI *Agenzia Spaziale Italiana*

CPU *Central Processing Unit*

CSA *Canadian Space Agency*

CW *Continuous wave*

DLR *Deutsches Zentrum für Luft- und Raumfahrt*

DS *Distributed Scatterer*

EOS *Earth Observing System*

ERS *European Remote-Sensing*

ESA *European Space Agency*

EW *Extra Wide Swath*

FM-CW *Frequency-Modulated Continuous-Wave*

GACOS *Generic Atmospheric Correction Online Service for InSAR*

GB *Gigabyte*

GB *Gigabyte*

GN *Geoportale Nazionale*

GPF *Graph Processing Framework*

GRD *Ground range Detected geo-referenced products*

IDE *Infraestrutura de Dados Espaciais*

IMSEP *internet-based information Management System for Environmental Protection and Disaster Risk Management*

INTA *Instituto Nacional de Técnica Aeroespacial*

ISRO *Indian Space Research Organization*

IW *Interferometric Wide Swath*

InSAR *Interferometric Synthetic Aperture Radar*

JAXA *Japan Aerospace Exploration Agency*

KARI *Korea Areospace Research Institute*

MTI *Moving target indicator*

N. R. Canada *National Resources Canada*

NSC *Norwegian Space Centre*

NVE *Norwegian Water Resources and Energy Directorate*

OCN *Ocean*

OGC *Open Geospatial Consortium*

PB *Petabyte*

PS *Persistent Scatterer*

RADAR *RAdio Detection And Ranging*

RAM *Random Access Memory*

SAFE *Sentinel Standard Archive Format for Europe*

SAR *Synthetic Aperture Radar*

SLAR *Side Looking Airborne Radar*

SLC *Slant-range single-look complex products*

SM *Strip Map*

SNAPHU *Statistical-Cost, Network-Flow Algorithm for Phase Unwrapping*

SNAP *Sentinel Application Platform*

SOA *Service-Oriented Architectures*

STAMPS *Stanford Method for Persistent Scatterers*

VCPU *Virtual Centralized Processing Unit*

WFS *Web Feature Service*

WMS *Web Map Service*

WV *Wave*

WebGIS *Web Geospatial Intelligence System*

XML *Extensible Markup Language*

ZTD *Zenithal Tropospheric Delay*

Lista de Símbolos

Símbolos gregos

- α Angle of attack.
- β Angle of side-slip.
- κ Thermal conductivity coefficient.
- μ Molecular viscosity coefficient.
- ρ Density.

Símbolos romanos

- C_D Coefficient of drag.
- C_L Coefficient of lift.
- C_M Coefficient of moment.
- p Pressure.
- \mathbf{u} Velocity vector.
- u, v, w Velocity Cartesian components.

Subcritos

- ∞ Free-stream condition.
- i, j, k Computational indexes.
- n Normal component.
- x, y, z Cartesian components.
- ref Reference condition.

Sobrescritos

- * Adjoint.
- T Transpose.

Capítulo 1

Introdução

Os satélites Radar de Abertura Sintética (*Synthetic Aperture Radar - SAR*) fornecem imagens no espectro das microondas, com muitas aplicações das quais informações estruturais sobre a estabilidade da orografia do terreno e ao mapeamento dos limites de derramamento de óleo nas águas oceânicas por embarcações. São sistemas ativos o que lhes confere a capacidade de funcionar de forma independente da luz solar e em condições meteorológicas mais adversas, como os ambientes atmosféricos saturados por poeiras ou cinzas.

As imagens SAR são complexas, uma vez que cada pixel regista em simultâneo a amplitude e a fase do sinal retrorefletido: a amplitude fornece informações relativas às características da superfície do solo e a fase regista informações relativas à distância entre o sensor e objeto a mapear.

O cálculo da diferença de fase a partir de duas imagens obtidas com geometrias de aquisição diferentes (diferentes posições de órbita e/ou em momentos diferentes), para um mesmo elemento retrodispersor, permite a determinação da altitude do objeto, numa técnica designada por interferometria e a sua utilização conjunta com imagens SAR tem por nome, Interferometria Radar de Abertura Sintética (*Interferometric synthetic aperture radar - InSAR*). Essas técnicas são capazes de medir alterações na escala de milímetros com alta resolução espacial e temporal, revelando-se, por isso, muito promissoras.

As técnicas de Interferometria Radar de Abertura Sintética são abordagens poderosas usadas em geodésia e detecção remota, para estimar, detectar, medir e monitorizar alterações na superfície da Terra, a partir do espaço, muitas vezes promovidas por atividade tectónicas (Grandin et al. [1]), erupções vulcânicas (González et al. [2]) ou extracção de águas subterrâneas (Chaussard et al. [3]).

O programa, Copernicus, programa de observação da Terra liderado pela Comissão Europeia em parceria com a Agência Espacial Europeia (*European Space Agency - ESA*), levou ao lançamento da missão Sentinel-1 (A/B), a qual permite um mapeamento global sistemático da superfície da Terra com base na técnica de observação do terreno por varredura progressiva e conduz, conseqüentemente a um conjunto de dados sem precedentes de grandes séries temporais de imagens. As imagens disponibilizadas pelo programa serviram de base ao desenvolvimento da presente dissertação.

De realçar que à semelhança do Copernicus existem outros programas de diferentes agências com facilidades similares, dos quais o Centro Aeroespacial Alemão (*Deutsches Zentrum für Luft- und Raum-*

fahrt - DLR) que disponibiliza as imagens TerraSAR-X desde 2007 (DLR [4]), a Agência Espacial Italiana (*Agenzia Spaziale Italiana* - ASI), que disponibiliza as imagens COSMO-SkyMED desde 2007 (ASI [5]), a Agência Japonesa de Exploração Aeroespacial (*Japan Aerospace Exploration Agency* - JAXA) que disponibiliza as imagens ALOS-Palsar desde 2006 (JAXA [6]) e a Agência Espacial Canadiana (*Canadian Space Agency* - CSA), que disponibiliza as imagens RADARSAT desde 1995 (CSA [7]).

1.1 Motivação

Do anteriormente exposto, pode concluir-se que o tema da monitorização com recurso a imagens de satélite tem ganho uma importância crescente, reforçando o papel das imagens de satélites como fontes de informação geográfica. A diversidade de sensores capazes de captar imagens da superfície terrestre com diferentes resoluções espaciais, espectrais e temporais tem permitido a geração de grandes volumes de informação nem sempre devidamente exploradas quer por falta de meios computacionais quer por dificuldades relacionadas com os formatos e acessos às plataformas de armazenamento.

Associando o nível de maturidade que as técnicas interferométricas atingiram à necessidade de monitorização e à quantidade de imagens disponíveis, surge a motivação da presente dissertação. A integração num visualizador único de todas as áreas mapeadas e dos resultados interferométricos processados é da máxima importância numa lógica de otimização de meios e recursos.

No desenvolvimento deste tema será dada preferência a imagens captadas por sensores radar de abertura sintética, pelos motivos apresentados anteriormente, isto é, por serem sistemas ativos e pela independência em relação às condições meteorológicas.

1.2 Objectivos

Com este trabalho pretendeu-se contribuir para promoção do uso de imagens SAR nos serviços de monitorização da superfície terrestre, recorrendo às técnicas interferométricas. Face ao exposto, o objetivo principal deste estudo é a automatização do processo interferométrico desde o *download* das imagens ou produtos SAR à disponibilização dos interferogramas corrigidos numa plataforma de visualização, isto é, automatizar o *download* dos produtos SAR, formação e correção dos interferogramas, comparação e, finalmente, a disponibilização dos interferogramas corrigidos para visualização.

1.3 Estrutura da Dissertação

Esta dissertação encontra-se estruturada em cinco capítulos, que, por sua vez, estão organizados em três secções: enquadramento, processamento e resultados.

O Capítulo 1 enquadra o tema em estudo, apresenta a motivação e identifica os principais objetivos a atingir.

No Capítulo 2 é feita uma breve revisão dos conceitos básicos dos Radares de Abertura Sintética, Interferometria SAR e Plataformas de dados espaciais, necessários à compreensão e desenvolvimento

do trabalho. São descritos de forma sucinta os detalhes geométricos do processo de aquisição das imagens, identificados os comprimentos de onda e os tipos de polarizações mais frequentes neste tipo de operação. São apresentados algumas das agências espaciais que dispõem de satélites ou missões SAR com ênfase ao Sentinel-1. Segue-se a descrição dos fundamentos da interferometria, dos quais o cálculo da fase interferométrica, a relação existente entre a diferença de fase e a altitude e as etapas que compõem o processo interferométrico. Finalmente, é apresentada uma breve revisão dos conceitos básicos referentes às infraestruturas de dados espaciais e exemplificado uma plataforma InSAR com algumas das suas referências mais técnicas e um Geoportal.

No Capítulo 3 é apresentada a metodologia utilizada nesta dissertação, que se encontra dividida em duas fases: automatização do acesso às imagens SAR; e automatização do processamento interferométrico e disponibilização dos resultados numa plataforma de visualização.

De seguida no capítulo 4 é apresentado um exemplo de aplicação da metodologia numa determinada área, que serviu de base aos testes efectuados.

Finalmente no capítulo 5 é feito um resumo do presente trabalho, expondo as conclusões, algumas considerações finais e reflexões sobre as perspectivas futuras e alguns desafios que se colocam.

Capítulo 2

Revisão de conceitos básicos de SAR, InSAR e plataformas de dados espaciais

A tecnologia RADAR (*RA*dio *D*etection *A*nd *R*anging) tem o seu princípio na diferença entre a emissão de ondas electromagnéticas e a receção do eco retornado de objectos reflectores ou de alvos. A intensidade desse sinal de eco fornecerá as informações sobre o alvo, enquanto o alcance, ou distância ao alvo, é determinado a partir do tempo que a energia irradiada demora a viajar até o alvo e ser recebida em forma de eco. Existem vários tipos de radares, onde se destacam os *Continuous wave* (CW) e *Frequency-Modulated Continuous-Wave* (FM-CW) , *Moving target indicator* (MTI) , *Airborne Moving target indicator* (AMTI) , pulso doppler, rastreamento (*Tracking*), orientação de mísseis (*Missile Guidance*) e os radar de abertura sintética (*Synthetic Aperture Radar* - SAR) Skolnik [8].

A utilização das microondas na detecção remota iniciou-se na década de 50 com fins, essencialmente, militares, que a partir da década de 70, estendeu-se a fins civis ESA [9].

Os sistemas de detecção remota podem classificar-se em activos e passivos. Os activos incorporam a própria fonte de energia enquanto que os sistemas passivos detetam a interação entre a radiação solar reflectida a partir da Terra, nos comprimentos de onda visíveis e de infravermelhos do espectro electromagnético. Os últimos não emitem radiação própria, mas apenas recebem luz natural e radiação de calor reflectida pela superfície da Terra. Deste modo, os sistemas activos possuem uma grande vantagem de poder obter registos em qualquer altura, não dependendo da altura do dia nem da estação do ano.

De entre os vários tipos de radares mencionados acima, seguiremos o estudo com os radares de abertura sintética - SAR. Estes radares são sistemas activos que podem ser transportados por diferentes plataformas, sendo os satélites, de facto, o meio de transporte mais comum e mais utilizado pela comunidade científica, devido ao potencial de cobertura global e disponibilidade de agências espaciais. Além dos satélites, os sensores SAR podem ser transportados em outras plataformas, como o *space shuttle* (Farr and Kobrick [10]), veículos aéreos não tripulados (Rosen et al. [11]) ou estruturas em carril (Monserrat et al. [12], Di Traglia et al. [13]). Para o presente estudo utilizou-se imagens registadas por satélites, e deste modo, apenas estas serão consideradas no desenvolvimento do trabalho.

2.1 Imagens SAR

As imagens que resultam dos sistemas SAR são imagens complexas, isto é, o eco registado é constituído por um valor de amplitude e de fase do sinal, resultando numa matriz complexa em que cada pixel apresenta um valor com uma componente real e imaginária. A amplitude é responsável por fornecer informações relativa as características da superfície do objecto e a fase, por sua vez, regista informações relativas à distância entre o sensor e o solo/objeto Bamler [14].

De seguida, são descritas de forma resumida a geometria inerente ao processo de aquisição das imagens e dos diferentes parâmetros que a compõem, a gama de frequência dos principais sistemas, o tipo de polarização dos sistemas SAR e os efeitos do relevo na geometria das imagens.

2.1.1 Geometria

Estes sistemas iluminam a superfície da Terra com geometria lateral (*Side Looking Airborne Radar – SLAR*), como ilustrado na figura 2.1, assumindo que em linha recta e a uma altitude H ao longo da direcção x , transmitem pulsos de microondas segundo a direcção y , designada por distância medida sobre o solo (*Ground Range*), iluminando assim a área. A direcção do movimento da plataforma é conhecida como direcção azimutal (*Azimuth direction*) e é obtida a partir de processamento de sinal. A velocidade de varrimento segundo y é inferior à velocidade de varrimento segundo R , direcção designada por distância inclinada (*Slant-Range*).

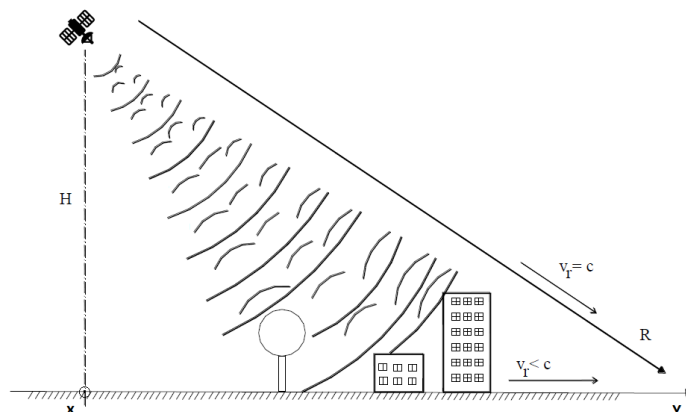


Figura 2.1: Geometria de aquisição de informação dos satélites SAR Falcão [15].

O ângulo em que um sinal atinge a superfície da Terra varia de acordo com a altitude do satélite e a sua configuração, mas também com a topografia local. Conforme demonstrado na figura 2.2, o ângulo de incidência (*incidence angle*) é definido como o ângulo entre o sinal de radar de entrada e a normal de uma superfície. Consequentemente, quanto maior o ângulo de incidência maiores são as distorções geométricas, visto que, um maior ângulo de incidência significa uma abertura lateral mais pronunciada. Consequentemente dois objetos/alvos $A1$ e $A2$ com o mesmo tamanho no solo (figura 2.3), tem dimensões aparentes na faixa inclinada ($A1'$ e $A2'$) diferentes, isto é, os objetos/alvos próximos do radar parecerem comprimidos em relação aos elementos distantes.

Recorrendo à trigonometria, a distância no solo pode ser calculada a partir do conhecimento da distância inclinada e da altitude da plataforma.

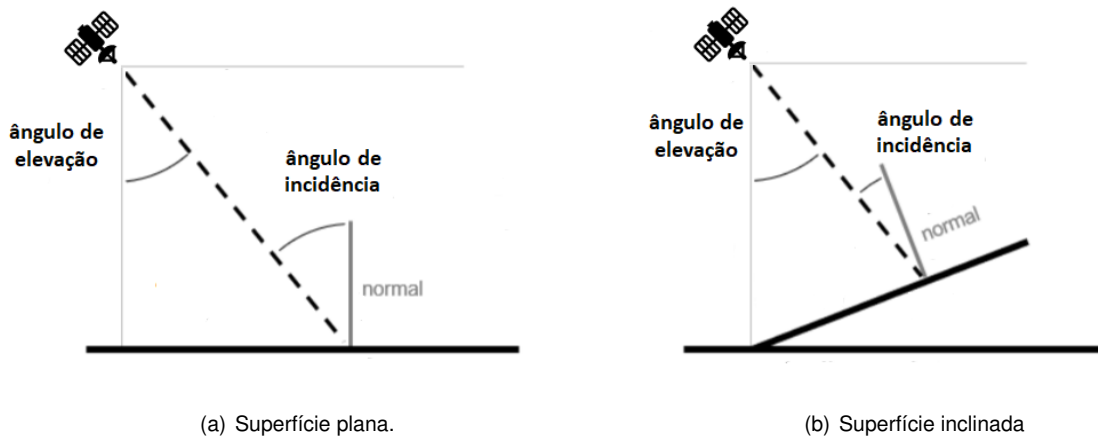


Figura 2.2: Convenções de nomenclatura angular no sistema de imagem de radar Lusch [16].

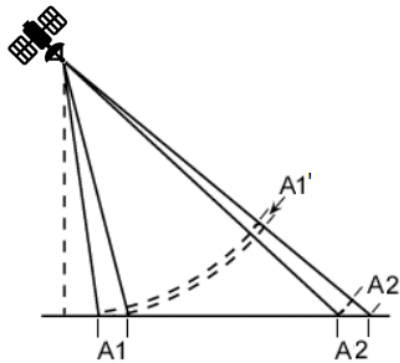


Figura 2.3: Distorções radiométricas em imagens de radar.

Para além das variações da retro-reflexão na direção do alcance (segundo R) ao longo de um plano, demonstra-se que a topografia local também influencia diretamente o retroespalhamento de uma superfície (subsecção 2.1.3).

2.1.2 Comprimentos de onda e polarização

A gama de variação dos comprimentos de onda mais comuns nos sistemas SAR compreende-se entre os 0.75 cm e 120 cm, distribuídos em sete bandas espectrais (tabela 2.1), sendo L, C e X as bandas mais utilizadas. Moreira et al. [17], ESA [18].

Banda Espectral	Ka	Ku	X	C	S	L	P
Frequência [GHz]	25-40	12-17.6	7.5-12	3.75-7.5	2-3.75	1-2	0.25-0.5
Comprimento de Onda [cm]	0.75-1.2	1.7-2.5	2.5-4	4-8	8-15	15-30	60-120

Tabela 2.1: Comprimento de onda e frequência das microondas.

De seguida, encontra-se listado e ilustrado na figura 2.4, um conjunto de agências que dispõem de satélites ou missões SAR e as respectivas bandas espectrais:

- *European Space Agency* (ESA - consorcio países Europeus): ERS-1, ERS-2, Envisat, Sentinel-1
- *Japan Aerospace Exploration Agency* (JAXA - Japão): JERS-1, ALOS-1, ALOS-2
- *Canadian Space Agency* (CSA - Canada): Radarsat-1, Radarsat-2, Radarsat constellation
- *Deutsches Zentrum für Luft- und Raumfahrt e.V.* (DLR - Alemanha): TerraSAR-X, TanDEM-X
- *Indian Space Research Organization* (ISRO - Índia): RISAT-1, NISAR (w/ NASA)
- *Comision Nacional de Actividades Espaciales* (Argentina): SAOCOM
- *Italian Space Agency* (ASI - Itália): COSMO-SkyMed
- *Instituto Nacional de Técnica Aeroespacial* (INTA - Espanha): PAZ
- *Korea Aerospace Research Institute* (KARI - Coreia): KOMPSAT-5
- *National Aeronautics and Space Administration* (NASA - Estados Unidos): NISAR (w/ ISRO)

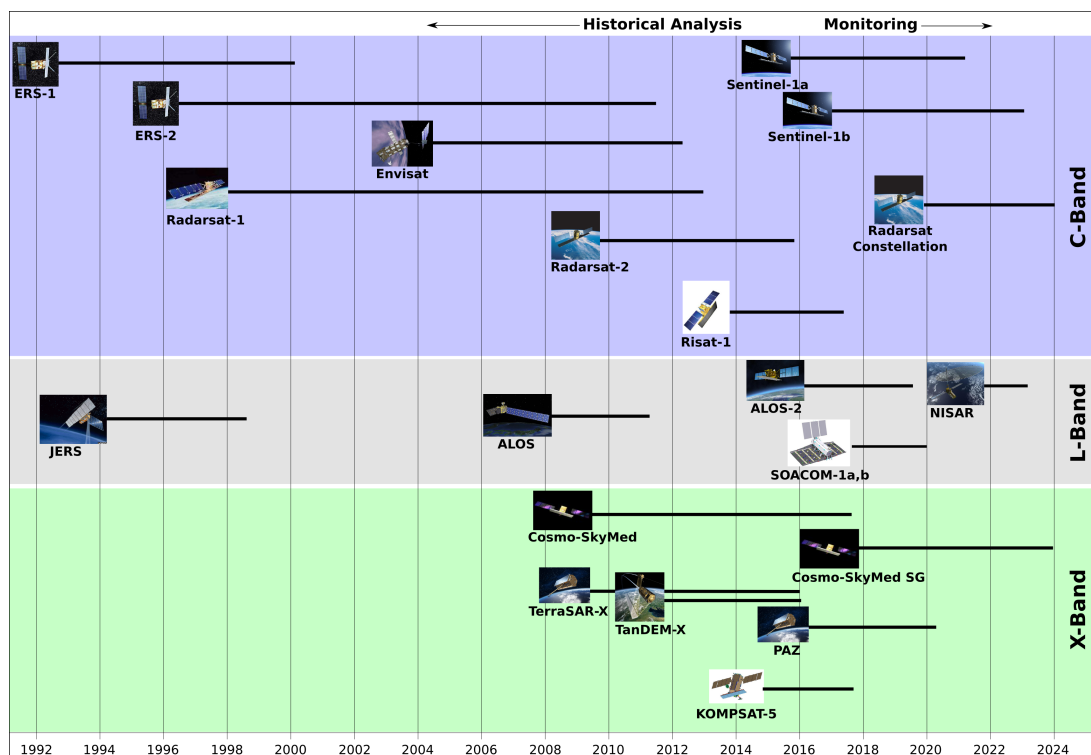


Figura 2.4: Satélites ou missões SAR existentes e planeadas, e as respectivas bandas espectrais. UNAVCO [19].

Uma outra característica importante é o tipo de polarização. Esta refere-se à orientação, fase relativa e repetibilidade dos campos eléctricos e magnéticos das ondas electromagnéticas. A polarização pode ser linear, circular ou elíptica. A polarização das ondas dispersadas depende da polarização das ondas incidentes e das características geométricas e dieléctricas dos elementos dispersores.

A polarização mais comum no mapeamento com radares é a linear (figura 2.5), onde operam com microondas polarizadas horizontalmente (H) ou verticalmente (V) Lusch [16], Giuli [20]. Consequentemente, denotando a polarização de transmissão e recepção por um par de símbolos, os sistemas de radares podem ter quatro diferentes modos: HH, VV, HV e VH. As duas primeiras combinações são denominadas como polarização paralela ou semelhante, visto que, as polarizações de transmissão e recepção são as mesmas. As duas últimas combinações são chamadas de polarização cruzada, dado que as polarizações de transmissão e recepção são ortogonais entre si Canada [21].

Há sistemas que operam na mesma polarização HH ou VV (e.g. ERS 1, ERS 2) e existem os chamados sistemas polarimétricos que adquirem de cada ponto do terreno as componentes HH, VV, HV e VH (e.g. Envisat).

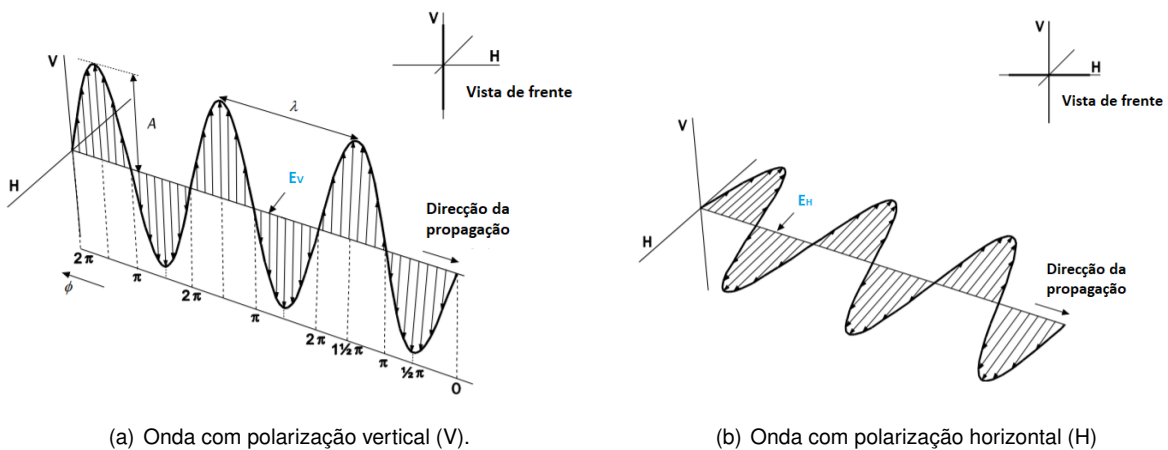


Figura 2.5: Vista lateral e frontal do campo eléctrico E de uma onda electromagnética com comprimento de onda λ , amplitude A, fase ϕ (0 à 2π) Sanden and Johannes [22].

2.1.3 Efeitos da geometria do terreno

Para além das distorções radiométricas de uma imagem de radar causadas pelo ângulo de incidência conforme descrito na subsecção 2.1.1, a topografia da superfície tem fortes efeitos na localização dos alvos na imagem inicial Lusch [16], van Zyl et al. [23]

A Figura 2.6 apresenta os três principais tipos de distorções geométricas causadas por variações na topografia: cavalgamento (*layover*), encurtamento (*foreshortening*) e sombra (*shadow*).

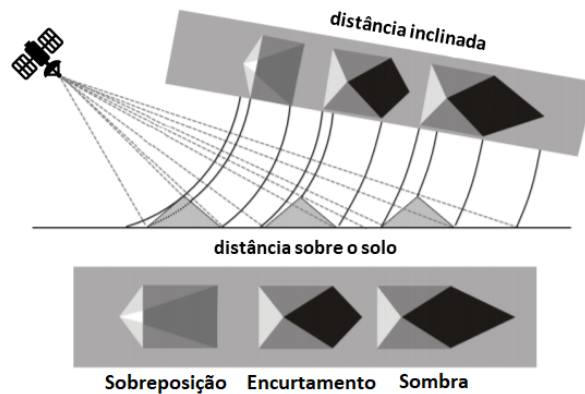


Figura 2.6: Distorções geométricas em imagens de radar Braun [24]

Quando o feixe do radar atinge a base de um alvo/objeto alto e inclinado (e.g. montanhas), antes de atingir o topo, ocorre o denominado **foreshortening**. A intensidade do fenómeno varia com o ângulo da encosta ou a variação da encosta da montanha em relação ao ângulo de incidência do feixe do radar. O **foreshortening** máximo ocorre quando o feixe do radar é perpendicular ao declive, de modo que a base e o topo são capturados simultaneamente pelo feixe Canada [25], ESA [26].

O fenómeno de **layover** ocorre quando o feixe de radar atinge o topo de um alvo com uma certa altura, antes de atingir a base, ou seja, o sinal de retorno do topo do alvo será recebido antes do sinal da base. Como resultado, a parte superior do alvo é deslocada em direcção ao radar em relação a sua posição real no solo. A semelhança do **foreshortening**, **layover** é mais acentuado para pequenos ângulos de incidência Canada [25], ESA [26].

Tanto o efeito de **foreshortening** como o efeito de **layover** resultam em **shadow** de radar. Esse fenómeno por sua vez, ocorre quando o feixe do radar não consegue iluminar uma superfície do solo ou do alvo. Ocorrem atrás dos alvos verticais ou encostas com vertentes muito íngremes. Como o feixe do radar não ilumina essas superfícies, resultarão em regiões escuras nas imagens. Quanto maior o ângulo de incidência, maior são os efeitos de **shadow** Canada [25], ESA [26].

2.2 Missões SAR

Como ilustrado na figura 2.4, existe actualmente uma diversidade de missões que permite o mapeamento global da superfície da terra. Para a presente dissertação, como já foi mencionado, utilizaram-se as imagens do Sentinel 1 como fonte de dados/informação, visto Portugal se tratar de um país membro da ESA, o que proporciona a disponibilização das imagens quase em tempo real e sem custos.

2.2.1 Sentinel 1

A constelação Sentinel é composta por vários satélites da Agência Espacial Europeia, projectados para fornecer uma grande quantidade de dados e imagens para o programa Copernicus da Europa.

O Sentinel-1 (figura 2.7) é composto por uma constelação de dois satélites, Sentinel-1A (lançado em 3 de Abril de 2014) e Sentinel-1B (lançado em 25 de Abril de 2016), desfazados 180 graus e incorpora

um instrumento SAR que opera na banda C (tabela 2.1) com quatro modos de aquisição de imagem, com diferentes resoluções e cobertura (tabela 2.2) ESA [27]. Trata-se de uma missão que oferece capacidade de polarização dupla, tempos de revisitação e de disponibilização do produto muito curtos. Os dois satélites juntos fornecem, pelo menos, duas medições do continente europeu a cada seis dias e de quase todo o território fora da Europa, com intervalos regulares de 6, 12 ou 24 dias (ESA [28]). Os produtos são distribuídos ou disponibilizados no formato SAFE (*Sentinel Standard Archive Format for Europe*) ESA [29].

Os quatro modos de captura ou aquisição de imagens, *Strip Map* (SM), *Interferometric Wide Swath* (IW), *Extra Wide Swath* (EW) e *Wave* (WV), produzem ou disponibilizam produtos SAR de três níveis: nível 0, nível 1 e nível 2 (tabela 2.2) ESA [28]. Como mencionado acima em relação à polarização, os produtos estão disponíveis em polarização única (VV ou HH) para o modo WV e única e dupla (VV + VH ou HH + HV) para os modos SM, IW e EW. O Anexo A.1, apresenta mais informações acerca destes modos e os respectivos níveis.

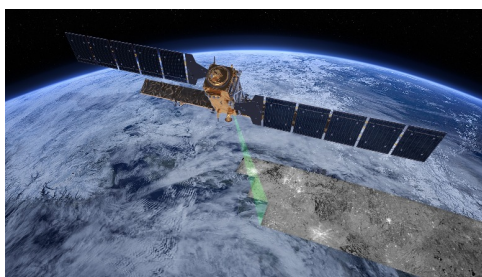


Figura 2.7: Satélite da missão Sentinel 1 EOS [30].

Modos	Resolução espacial [m]	Cobertura/Largura da faixa [Km]
SM	5x5	80
IW	5x20	250
EW	20x40	400
WV	20x20	100

Tabela 2.2: Modos de captura de imagens SAR e as respectivas resoluções e coberturas.

Modos	Nível 0	Nível 1	Nível 2
SM	Dados não tratados	SLC, GRD	OCN
IW	Dados não tratados	SLC, GRD	OCN
EW	Dados não tratados	SLC, GRD	OCN
WV	-	-	OCN

Tabela 2.3: Modos de captura de imagens SAR e os respectivos níveis.

Ainda é de realçar que já se encontra planeado o lançamento de outros dois satélites (Sentinel-1C e Sentinel-1 D) para substituir os dois primeiros no final de suas vidas operacionais (geralmente de 7 anos com recursos para 12 anos) ESA [31].

2.3 Interferometria SAR

A interferometria radar explora as imagens adquiridas com os radares de abertura sintética, observando uma mesma área com ângulos ligeiramente diferentes. Essa aquisição pode ser feita em simultâneo, com dois radares na mesma plataforma, denominada de Interferometria de passagem única, ou em momentos diferentes, intitulado de Interferometria com repetição de passagem, explorando órbitas repetidas do mesmo satélite Ferretti et al. [32], Lillesand et al. [33].

A distância entre dois satélites no plano perpendicular à órbita é chamada de base interferométrica e a sua projeção perpendicular à distância inclinada é designada por base perpendicular (figura 2.8).

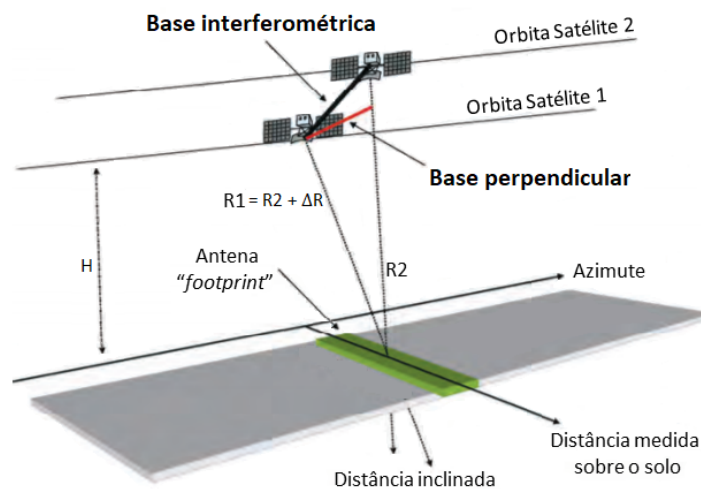


Figura 2.8: Geometria de um sistema de satélite interferométrico SAR Ferretti et al. [32].

A interferometria explora a diferença de fase entre duas observações complexas de radar SAR tiradas de posições de sensores ligeiramente diferentes e extrai informações sobre a superfície da Terra.

A fase da imagem SAR é determinada, principalmente, pela distância entre a antena de satélite e os alvos terrestres. O interferograma SAR é gerado pela multiplicação cruzada, pixel por pixel, da primeira imagem SAR com o conjugado complexo da segunda Ferretti et al. [32]. A amplitude do interferograma é a amplitude da primeira imagem multiplicada pela segunda, enquanto a fase (interferométrica) é a diferença de fase entre as imagens.

Essa diferença de fase resulta numa nova "imagem", chamada interferograma (figura 2.10), um padrão de franjas que contem todas as informações relativas à geometria do terreno, alvo ou objeto. A fase do interferograma está altamente correlacionada com a topografia do terreno. Deste modo, a diferença de fase interferométrica ($\Delta\phi$) é proporcional ao deslocamento (ΔR) dividido pelo comprimento de onda transmitido (λ), de acordo com:

$$\Delta\phi = \phi_1 - \phi_2 = \frac{4\pi}{\lambda}(R_1 - R_2) = \frac{4\pi}{\lambda}\Delta R. \quad (2.1)$$

Sendo R_1 e R_2 as distâncias dos satélite 1 e 2, respectivamente, ao objecto/alvo sobre a superfície terrestre.

Na diferença de fase existem vários factores contribuintes, onde se destaca a fase de Terra plana (curvatura da terra), a fase topográfica (superfície topográfica da terra), condições atmosféricas (humidade, temperatura e mudança de pressão entre as duas aquisições), e outros ruídos (mudança dos espalhadores, diferente ângulo de visão e espalhamento de volume) e, por último, a eventual deformação superficial que ocorre entre as duas aquisições Ferretti et al. [32], Rocca [34].

2.3.1 Relação entre a diferença de fase e a altitude

Depois de abordar a fase interferométrica na secção anterior, determina-se, agora, a relação existente entre a diferença de fase e a correspondente altitude.

Considerando o sistema ilustrado na figura 2.8, utilizada tanto na técnica InSAR de passagem única, como na técnica InSAR com repetição de passagem, a relação entre a altura de um ponto sobre a superfície de referência e a diferença de fase absoluta obtida a partir de um par de imagens SAR é dado por:

$$\delta H = \frac{\lambda R_1 \sin(\theta_0)}{4\pi B \cos(\theta_0 - \alpha)} \delta \phi. \quad (2.2)$$

Sendo θ_0 o ângulo de elevação do satélite 1 em relação à superfície de referência, α o ângulo de inclinação entre os sensores, R_1 a distância do satélite 1 ao objecto/alvo sobre a superfície terrestre e B a base interferométrica. Ferretti et al. [32], Hanssen [35].

2.3.2 Etapas do processo interferométrico

As diferentes geometrias de aquisição conjuntamente com o facto das direcções de navegação dos sensores não serem exatamente paralelas determinam que as grelhas de aquisição não sejam coincidentes, existindo entre elas variação de escala, rotação e translação. Deste modo, é necessário um procedimento sequencial que se inicia com o registo das imagens, construção do interferograma, determinação da coerência, filtragem e reamostragem.

Registo

A primeira operação consiste no registo das imagens, onde uma é escolhida para ser a base (*master*) e as outras imagens são as escravas (*slaves*). Os pixels nas imagens *slaves* serão movidos para se alinharem com a imagem *master*. O registo garante que cada alvo terrestre contribua para o mesmo pixel (alcance, azimute) em ambas as imagens (*master* e *slave*).

O registo é efectuado no geral em duas fases. A primeira fase é designada por registo grosseiro e consiste no registo de uma imagem sobre a outra com resolução de um elemento imagem. A segunda fase, por sua vez designada por registo fino, consiste no registo de uma imagem sobre a outra com resolução do sub elemento de imagem, envolvendo reamostragem.

A figura 2.9 ilustra o processo de registo entre duas imagens, I_1 e I_2 . O registo grosseiro com resolução de um elemento de imagem consiste na determinação de elementos de imagem correspon-

dentos. As disparidades de posição são determinadas com base nos valores obtidos após o cálculo da correlação cruzada. Concretizado o registo grosseiro, segue-se a reamostragem das imagens a um ritmo dez vezes superior, (figura 2.9 b). Para tal, calcula-se a correlação cruzada, agora para as imagens reamostradas, e determinam-se as disparidades de posição, obtendo-se o registo fino.

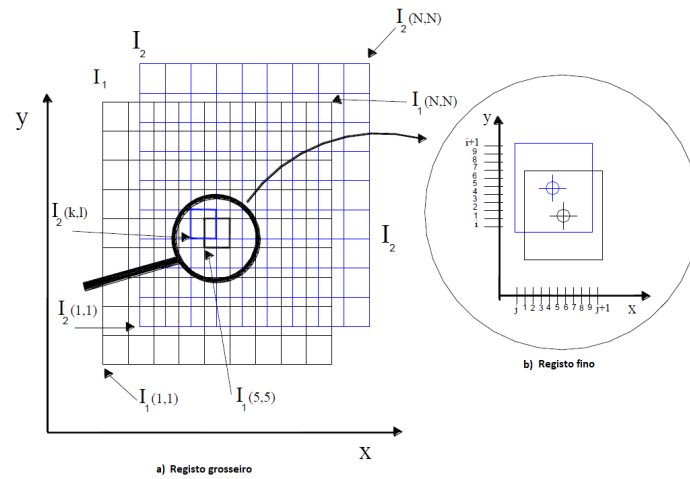


Figura 2.9: Ilustração do registo entre duas imagens SAR, I_1 e I_2 Falcão [15].

Construção do Interferograma

Concluído o processo de registo das imagens gera-se o interferograma a partir da multiplicação cruzada de elemento a elemento da imagem master com o conjugado complexo do *slave* Ferretti et al. [32].

A Figura 2.10 ilustra um interferograma obtido através de dois produtos SLC adquiridas pelo sensores Sentinel-1 em 03 de Novembro de 2014 e 27 de Agosto de 2014:

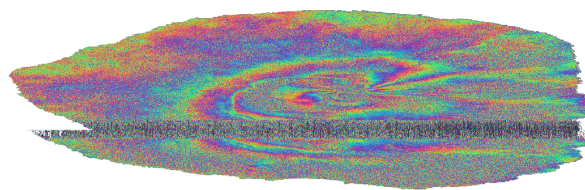


Figura 2.10: Interferograma da ilha do Fogo, Cabo Verde, produzida através das imagem adquiridas pelo satélite Sentinel 1, em 3 de Novembro de 2014 e 27 de Novembro de 2014.

Determinação da Coerência

A coerência é uma medida que permite medir o nível de correlação entre duas imagens, variando, entre zero (sem correlação) e 1 (correlação máxima). Quando aplicada a um par interferométrico é um indicador de qualidade. Deste modo, a análise da coerência permite a selecção de pares de imagens para exploração em aplicações interferométricas. A coerência pode, em muitas circunstâncias, servir como medidor de previsão, visto que, quanto maior a coerência entre duas ondas electromagnéticas,

mais fácil é prever as propriedades de uma dessas ondas tendo conhecimento da outra. Trata-se de uma informação útil quando analisada dentro de um determinado intervalo de tempo ou espaço Woodhouse [36].

A Figura 2.11 apresenta a imagem de coerência para uma área de Lisboa, com valores compreendidos no intervalo [0.0, 0.84]. As áreas de alta coerência (edifícios) estão associados aos tons mais claros ou brilhantes enquanto que, as áreas com pouca coerência estão associadas às cores escuras (vegetação).

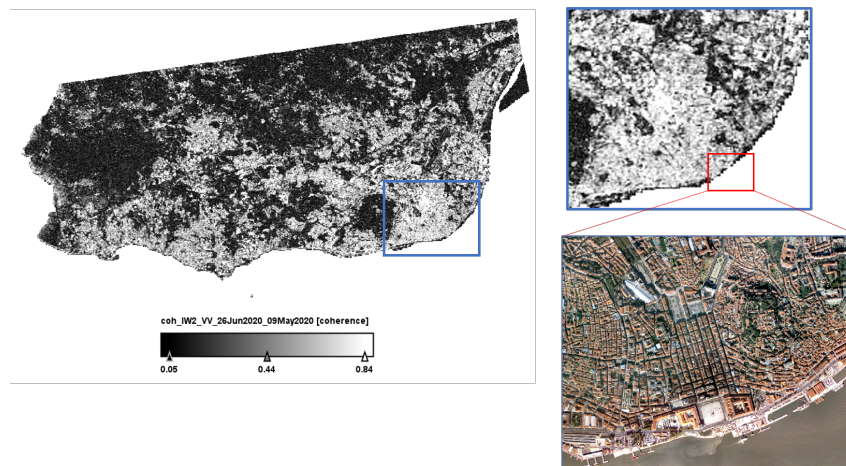


Figura 2.11: Imagem de Coerência da zona Lisboa produzida através das imagem adquiridas pelo satélite Sentinel 1, em 21 de Maio de 2020 e 26 de Junho de 2020.

Filtragem

Após a formação do interferograma, a diferença de fase pode ser corrompida por ruído de de-correlação temporal e geométrica, dispersão de volume e outros erros de processamento. As informações de fase em áreas de-correlacionadas não são restauradas. Entretanto, a qualidade das franjas existentes no interferograma pode ser aumentada aplicando filtros de fase especializados, como por exemplo, o filtro *Goldstein* que usa uma Transformação Rápida de Fourier (*Fast Fourier Transform* - FFT) para aumentar a relação sinal-ruído da imagem Goldstein and Werner [37]. A abordagem é baseada na aplicação de filtragem sobre as áreas que mais necessitam, preservando as áreas menos afectadas pelo ruído, otimizando, assim, o aproveitamento e precisão dos resultados. Para efectuar um desenrolamento adequado é necessário aplicar esta etapa.

Multilooking

O processamento de *multilooking* representa uma média espacial das componentes complexas do sinal e é aplicada sobre os interferogramas complexos a partir de uma janela média determinada no sentido de reduzir a variação estatística do ruído, ou seja, é usado como um algoritmo eficiente para reduzir o ruído de fase em uma imagem interferométrica Huang and Van Genderen [38]. O *multilooking* pode melhorar a fiabilidade do desenrolamento de fase (Goldstein et al. [39]), ao mesmo tempo que

reduz a resolução espacial. O número de *looks* deve ser adequado ao objectivo de cada aplicação, assim como às características do sensor utilizado e ao comportamento da fase em diferentes cenários geográficos. Para a correcção de *baseline* geométrica, por exemplo, utiliza-se valores mais altos. Para a análise de interferogramas, preservando ao máximo a resolução dos dados, deve-se trabalhar com janelas menores. É importante que o número de *looks* tenha razão igual a 1 para a obtenção de pixels quadrados.

Phase unwrapping

Para poder relacionar a fase interferométrica com a altura topográfica, a fase deve primeiro ser desenrolada. A ambiguidade da fase é definida como a diferença que gera uma mudança contínua de fase interferométrica de 2π no interferograma enrolado.

De acordo com Zebker et al. [40], o desempenho do algoritmo de desenrolamento de fase depende de dois factores: do SNR (*signal to noise ratio*) do interferograma e do espaçamento entre as franjas interferométricas. As duas abordagens principais para o desenrolamento de fase são métodos locais e globais. Os métodos locais incluem algoritmos baseados nos resíduos (*residue-based algorithms*), como proposto por Goldstein et al. [39], e em crescimento de regiões. O objectivo é identificar e isolar os pontos onde há inconsistência de fase, para que se evite a propagação de erro durante o desenrolamento. Os métodos globais buscam minimizar o erro global através, por exemplo, do método de mínimos quadrados proposto por Zebker and Lu [41] e de técnicas multi-resolução, dentre outros.

Depois de excluir qualquer número inteiro de ciclos de fase, a variação de fase entre dois pontos no interferograma enrolado fornece uma medição da variação da cota (figura 2.12). Consequentemente, os resultados desenrolados devem ser interpretados como uma diferença de cotas ou deslocamentos relativos entre os pixels de duas imagens.

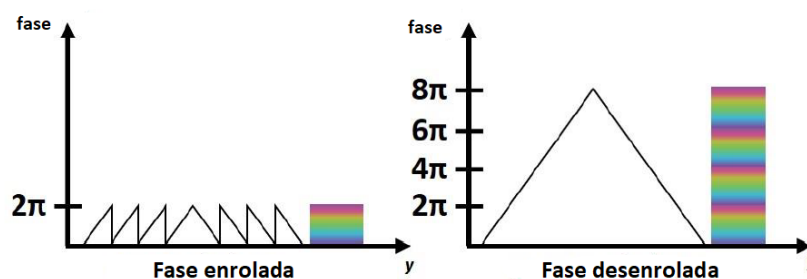


Figura 2.12: Ilustração do processo de desenrolamento interferométrico Braun [42].

2.4 Plataformas de dados espaciais

Os constantes avanços das infraestrutura de dados espaciais (IDE) e das entidades responsáveis pelas infraestruturas, juntamente com o crescente volume de dados produzidos diariamente, permite dizer que é necessário dar importância não só apenas as questões técnicas e de infraestrutura, mas também para as questões de acessibilidade dos dados envolvidos por estas estruturas.

Com os avanços tecnológicos da *internet* e os seus associados têm sido criadas ótimas condições para o desenvolvimento de ambientes propícios para a publicação, acesso, exploração e distribuição de informações e dados geográficos.

2.4.1 Infraestruturas de Dados Espaciais

O conceito de Infraestruturas de Dados Espaciais (IDE), que teve origem no início da década de 1990 (Council et al. [43]), abrange uma estrutura tecnológica, política, com padrões e recursos humanos necessários para adquirir, processar, armazenar, disseminar e utilizar de forma eficaz as informações geográficas. Estas infraestruturas, através da partilha de recursos, evitam esforços redundantes e reduzem o custo de produção de novas bases de dados ou de bases de dados já existentes, isto é, facilitam o acesso às informações espaciais já geradas, de maneira a evitar custos de produção de informações duplicadas, visto que a aquisição de dados espaciais requer tempo e recursos financeiros Rajabifard and Williamson [44], Phillips et al. [45], Budhathoki et al. [46].

As IDE podem existir a diversos níveis de intervenção no território (figura 2.13), desde o nível corporativo e local, que são IDE que abrangem menores áreas mas com maiores níveis de detalhes, até chegar aos níveis regionais e globais, que abrangem grandes áreas territoriais Rajabifard et al. [47].

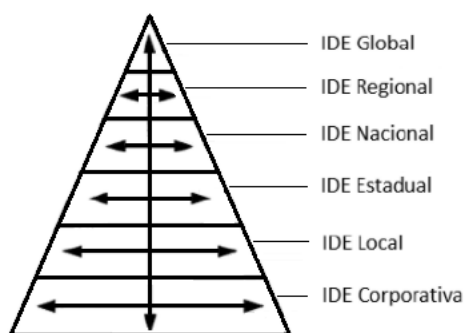


Figura 2.13: Estrutura hierárquica de uma IDE Rajabifard et al. [47].

A primeira geração dessas infraestruturas foram projetos nacionais com um forte envolvimento dos governos, com o objectivo de incentivar o desenvolvimento económico a partir da ampla oferta de bens e serviços para acesso público e destinados a múltiplas finalidades Masser [48].

Antes do desenvolvimento e implementação destas infraestruturas, somente os profissionais da área possuíam acesso aos dados espaciais. Estes realizavam estudos e, posteriormente, transmitiam os resultados aos cidadãos e aos órgãos públicos. Com o desenvolvimento de IDE, os cidadãos e órgãos públicos, para além de acesso aos resultados, passam a ter acesso às informações espaciais,

o que lhes dá a capacidade de poderem desenvolver análises e obterem os seus próprios resultados.

Uma arquitetura IDE totalmente funcional deve fornecer aos utilizadores as possibilidades de descobrirem o tipo de dados que procuram, de visualizar os dados *online* e aceder a esses dados caso sejam satisfatórios para os fins pretendidos. Deste modo, os Geoportais podem ser vistos como componentes dessas infraestruturas Maguire and Longley [49], Tait [50]. Enquanto que uma IDE concretiza uma oferta de dados geográficos, a partir de provedores que publicam serviços Web específicos, os Geoportais procuram meios para que os utilizadores (público em geral) possam dispor de um acesso interativo aos dados, incluindo visualizadores e ferramentas de descoberta com base em meta-dados.

Como qualquer infraestrutura, uma IDE implica a existência de algum tipo de coordenação para formulação e implementação de políticas, além da produção de meta-dados com certos padrões Maguire and Longley [49]. A política formal e o ambiente legislativo dentro do qual a iniciativa da IDE se encontra é uma consideração importante, e o alinhamento com as prioridades da política é fundamental para o sucesso da IDE. Essas políticas podem ser classificadas em políticas estratégicas e políticas operacionais. Enquanto que as políticas operacionais abordam tópicos relacionados com o ciclo de vida dos dados geográficos e ajudam a facilitar o acesso e o uso desses dados (por exemplo, diretrizes e manuais relacionados com a recolha, gestão, disseminação e uso dos dados), as políticas estratégicas abordam problemas de alto nível e estabelecem orientações para as organizações (e.g. reforço do cumprimento de determinados padrões e procedimentos).

Para o desenvolvimento destas políticas são tomadas várias questões em consideração onde se destacam: produção e partilha de dados; privacidade e dados confidenciais/sensíveis; proteção da propriedade intelectual; licenciamento de dados; arquivamento e preservação de dados; adopção de informações geográficas e computação em nuvem; e serviços baseados em localização.

Componentes e Arquitetura de uma IDE

As IDE são compostas por um conjunto de componentes técnicas e não técnicas que facilitam a partilha de informações geográficas: dados, meta-dados, serviços de rede e acordos de partilhas Cipolloni [51]. Esses conjuntos de componentes que representam a IDE garantem algumas das funcionalidades principais para os utilizadores, como:

- Partilha de dados e geocodificação;
- Pesquisa e visualização de dados e metadados;
- *Download* de dados (de acordo com a política de dados);
- Publicar e partilhar documentos, mapas e camadas.

Conceitualmente, a arquitetura IDE consiste em provedores de dados, serviços e consumidores que utilizam aplicativos para aceder às informações espaciais (figura 2.14) Fortes and de Araújo [52].

A implementação destas infraestruturas tem sido realizada recorrendo a arquiteturas orientadas por serviços (*Service-Oriented Architectures* - SOA), permitindo, assim, ser independente das tecnologias Bernard and Craglia [53]. Os serviços *Web* (*web services*) são vistos como a melhor e mais popular forma de implementar estas infraestruturas, sendo sistemas de software projetados para proporcionar

uma integração de sistemas e comunicação entre aplicações diferentes. Esta interoperabilidade é obtida por meio de um conjunto de padrões abertos baseados em XML ¹. Esses padrões fornecem uma abordagem comum para definir, publicar e usar serviços da web W3C [55].



Figura 2.14: Arquiteturas orientadas por serviços Vinoski [56]

As IDE seguem vários padrões ou normas definidos pela organização internacional de padrões de consenso, *Open Geospatial Consortium* (OGC), tais como o *Web Feature Service* (WFS) e o *Web Map Service* (WMS) Klopfer [57]. Sendo o primeiro, serviços especificados para acesso e manipulação de dados geográficos na Web e o segundo, protocolos padrões que especificam como os servidores de mapas devem descrever e disponibilizar as suas informações geográficas.

Na figura 2.15 pode-se observar um esquema padrão da arquitetura das Infraestruturas de dados espaciais.

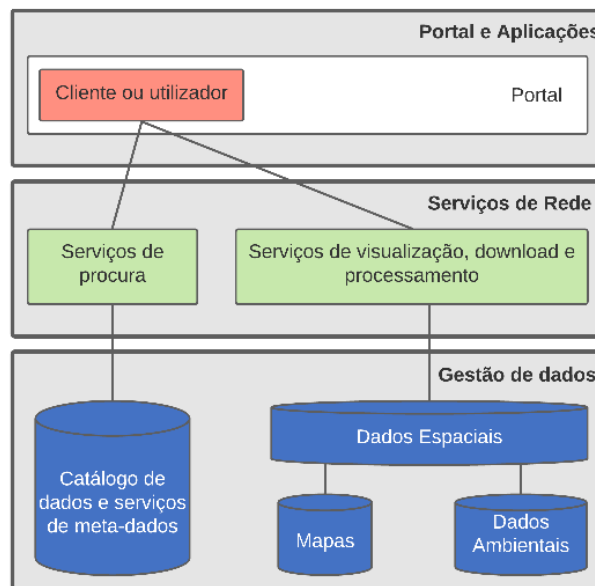


Figura 2.15: Esquema padrão da arquitetura das Infraestruturas de dados espaciais Cipolloni [51]

¹XML (*Extensible Markup Language*) é um formato de texto simples e muito flexível. Projetado originalmente para atender aos desafios da publicação electrónica em grande escala, desempenha um papel importante na troca de uma ampla variedade de dados principalmente na Web W3C [54]

2.4.2 Exemplos de plataformas de dados espaciais

Nesta sub secção são apresentados alguns exemplares de plataformas de entre várias existentes, com algumas das suas especificações mais técnicas, que permitem à população em geral, mas principalmente nos locais de criação das mesmas, visualizar, analisar, gerir e partilhar informações geográficas.

InSAR Norway

InSAR Norway (www.insar.ngu.no) é um serviço de mapeamento gratuito e aberto para toda Noruega, de dados InSAR. Trata-se de um serviço que surgiu de uma parceria conjunta entre *Geological Survey of Norway* (NGU), *Norwegian Water Resources and Energy Directorate* (NVE) e *Norwegian Space Centre* (NSC).

O projecto iniciou-se em 2016, com o intuito de estabelecer ou implementar um centro nacional de deformação terrestre. Um dos principais objectivos do centro é tornar os mapas de deformação facilmente disponíveis para todos (figura 2.16) NGU [58].

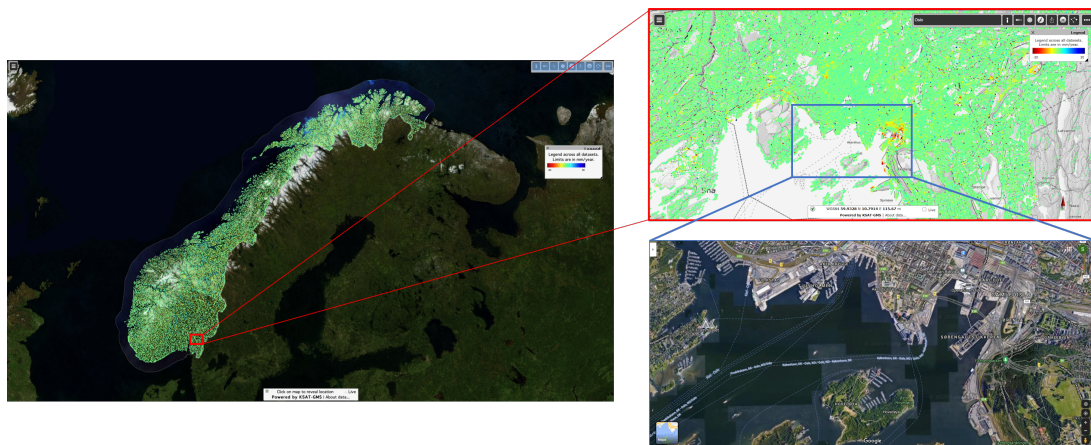


Figura 2.16: Plataforma InSAR da Noruega

Trata-se de um serviço baseado em um fluxos de *Persistent Scatterer - PS* para dados Sentinel-1, atualizado a cada 12 meses. *Persistent Scatterer* é uma técnica que se foca na identificação de pixéis nas imagens SAR, caracterizados por valores baixos de ruído. A obtenção desses valores está relacionada com dois tipos de difusores no terreno: aqueles em que a resposta ao radar é dominada por um objeto com reflexão forte, permanecendo constante ao longo do tempo, Difusor Persistente (*Persistent Scatterer*); e aqueles onde a resposta ao radar, embora também constante ao longo do tempo, apresenta pequenas diferenças na reflexão entre os objetos, Difusores Distribuídos *Distributed Scatterer - DS* Hooper [59].

A produção é executada em um cluster dedicado que consiste em 19 nós de processamento de alto desempenho com as seguintes características:

- 14 Unidades central de processamento (*Central Processing Unit - CPU*) físicas;
- 512 GB de Memória de Acesso Aleatório (*Random Access Memory- RAM*);

- Acesso rápido a grande sistema de armazenamento (2 PB) .

Continuamente são aplicadas três etapas de pré-processamento: recuperação de dados; co-registo em relação a imagem *master*; e geração de interferogramas com o mesmo *master* com resolução total CLMS [60]. Desconsiderando o armazenamento, os recursos necessários para o pré processamento são relativamente pequenos e são geridos por um único nó de processamento como sendo um trabalho de rotina noturno que pode ter ou não a duração de horas, para todo o território Norueguês CLMS [60].

O ciclo de produção é executado, atualmente, uma vez por ano após o fim da fase sem neve com uma duração total de 80 min por unidade de processamento por vCPU CLMS [60].

Geoportale Nazionale

Geoportale Nazionale - GN , é um geoportal italiano (www.pcn.minambiente.it), gerido e mantido pelo Ministério do Ambiente e Protecção da Terra e do Mar. Inclui a cartografia disponível nas áreas italianas, a partir do qual se pode obter informação sobre os principais temas ambientais e territoriais. Trata-se de um sistema de gestão e partilha de dados territoriais e ambientais, baseado em tecnologias webGIS.

A plataforma permite ou disponibiliza um visualizador de mapa (figura 2.17), um catálogo de meta-dados que permite a pesquisa de dados e um conjunto de serviços de acordo com os padrões do *Open Geospatial Consortium* (OGC), que permitem a visualização, download de dados e processamento dos mesmos (utilizando os serviços WMS, WFS e WCS) NG [61], Rossi [62].



Figura 2.17: Geoportal Italiano

A arquitetura da infraestrutura permite que os fornecedores mantenham o controle e a gestão dos seus dados e serviços, disponibilizando-os aos utilizadores a partir dos meta-dados associados.

A partilha de informações cartográficas obtidas a partir de técnicas avançadas de monitorização remota compartilhadas entre órgãos da administração pública ocorre por meio da Infraestrutura Nacional de Dados (IDN) Rossi [62].

Capítulo 3

Metodologia Desenvolvida

Neste capítulo é apresentada de forma sucinta a metodologia desenvolvida no trabalho e apresentadas as fases do processo, desde o acesso às imagens, a formação e correção dos interferogramas e, por fim, a disponibilização dos resultados numa plataforma de visualização de dados espaciais (figura 3.1). Genericamente a metodologia (figura 3.1) divide-se em duas fases:

1. Automatização do acesso às imagens a partir do acesso ao servidor e exportação no formato SLC;
2. Automatização do processamento interferométrico e disponibilização dos resultados na plataforma de visualização.

O processamento das imagens por sua vez encontra-se dividido em duas partes, correspondentes à geração dos interferogramas e à remoção dos efeitos da atmosfera (por simplificação denominado na presente dissertação por correção atmosférica).

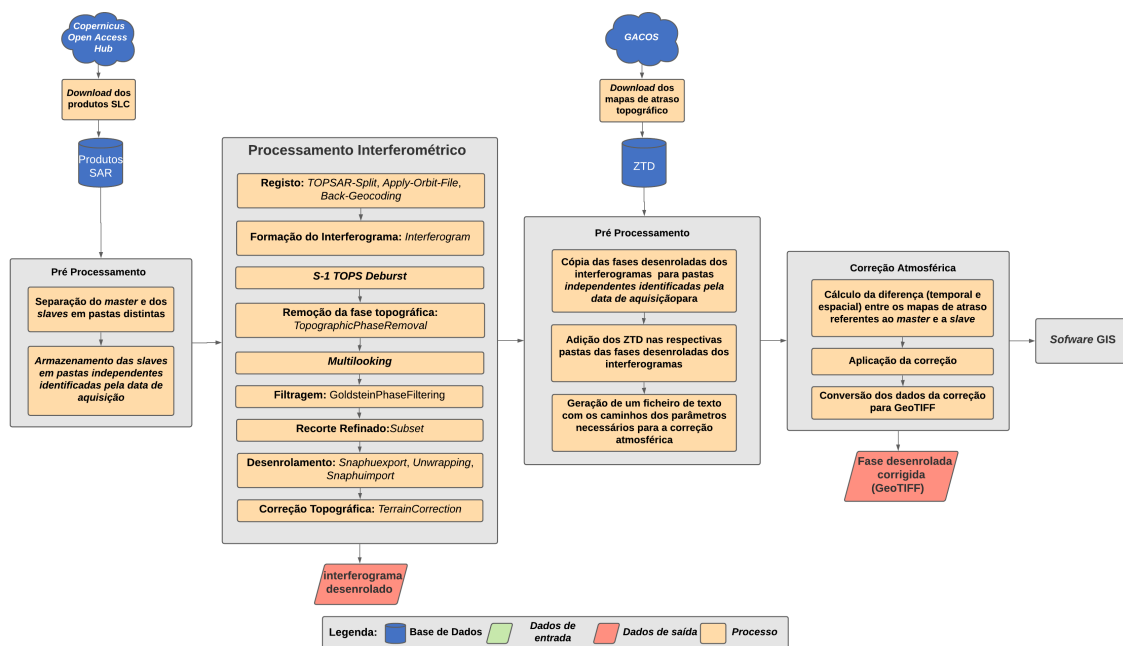


Figura 3.1: Fluxograma da metodologia utilizada.

3.1 Download dos produtos SAR

Como referido anteriormente, serão utilizadas as imagens SAR da missão Sentinel-1, disponibilizadas pela plataforma da ESA referente ao Programa Copernicus. Para os países associados da ESA a utilização das imagens SAR é livre e gratuita na sua generalidade, bastando para tal aceder ao site da ESA (através da criação de uma conta), Programa *Copernicus* (*Copernicus Open Access Hub*), efetuar a pesquisa por área geográfica e fazer o respetivo *download* das imagens.

Dado que um dos principais objetivos desta dissertação passa pela automatização dos processos, optou-se pelo desenvolvimento de uma ferramenta que agiliza a obtenção das imagens sem ter que recorrer manualmente à plataforma, isto é, a partir de *APIs* e *Batch Scripting*. A solução construída utiliza o *dhusget script* (ESA [63]) que permite a configuração parametrizada do *download* de produtos. Dado que o *script* apenas pode ser utilizado num ambiente *LINUX*, desenvolveu-se um *script* em Python, *download.py*, que, através de um ficheiro de configuração, *download.conf* (figura 4.2), preenchido *à priori* com os parâmetros desejados, efetua a busca e assegura o respetivo *download* dos produtos. Para este *script* utilizou-se uma biblioteca de nome *Sentinelsat*, que facilita a pesquisa, o *download* e a recuperação de metadados de imagens Sentinel do *Copernicus Open Access Hub* Andy Hooper [64]. A utilização dos *scripts* é descrita com grande detalhe na secção B.1 do anexo B.

3.2 Processamento das imagens SAR

Nesta secção são descritos os processos, os *softwares* necessários, os *scripts* e os *batch scripts* necessários à automatização do processo. Na figura 3.2, é apresentado o fluxograma do processo que permite a construção dos interferogramas e a aplicação das respetivas correções atmosféricas, analisado em detalhe nas secções subsequentes.

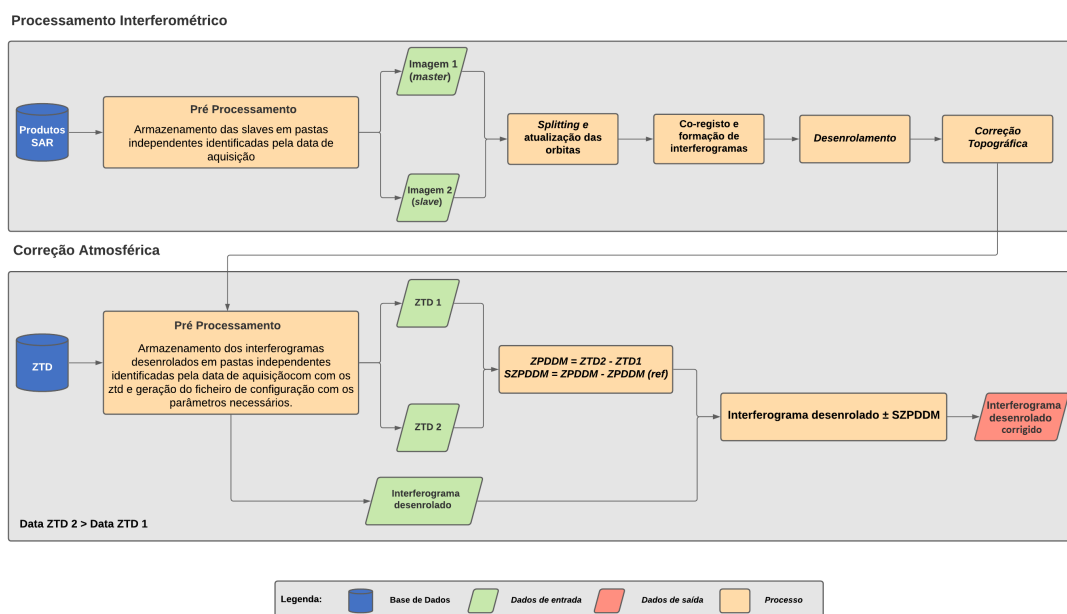


Figura 3.2: Processamento das imagens SAR.

Da diversidade de *softwares* existentes para o processamento interferométrico optou-se pelo *software* SNAP (*Sentinel Application Platform*), disponibilizado pela ESA, pelo motivo de ser de acesso livre, grátis e estar configurado para o uso das imagens disponibilizadas ao abrigo do Programa *Copernicus* e por permitir o processamento de dados a partir da linha de comandos.

De modo a automatizar o processamento das imagens recorreram-se às linguagens de programação *Python* e *MATLAB* e desenvolveram-se os *scripts* apresentados nas figuras 3.4 e 3.13, capazes de desempenhar as tarefas que antes tinham de ser feitas manualmente. Prévio ao processamento interferométrico e remoção do efeito da atmosfera, é necessário algum pré processamento, como demonstrado na figura 3.2, de modo a facilitar essa automatização. Assim, o processamento das imagens ficou dividido em oito etapas ou processos, a saber:

1. Armazenamento das imagens *slaves* em pastas independentes identificadas pela data de aquisição;
2. *Splitting* da imagem *master* e das imagens *slaves* e actualização das respectivas órbitas;
3. Co-registo e geração dos interferogramas;
4. Desenrolamento dos interferogramas e aplicação da correcção topográfica;
5. Armazenamento dos interferogramas desenrolados em pastas independentes identificadas pela data de aquisição;
6. Adição dos respectivos mapas de atraso topográfico nas pastas dos interferogramas desenrolados;
7. Geração de um ficheiro de texto com parâmetros e os caminhos dos ficheiros necessários para a remoção dos efeitos atmosféricos e adição dos respectivos *scripts* *MATLAB* devidamente configuradas;
8. Remoção dos efeitos da atmosfera dos interferogramas desenrolados.

O início da automatização deu-se com a construção e, posteriormente, a organização dos *scripts*, dos produtos SLC anteriormente obtidos, dos gráficos xml (ver subsecção 3.2.1), dos mapas de atraso topográfico (ver subsecção 3.2.2) em directórios separados e criação de um ficheiro de configuração (figura 4.3) e directórios para os quais os resultados dos processos são guardados, como demonstrado na figura 3.3:

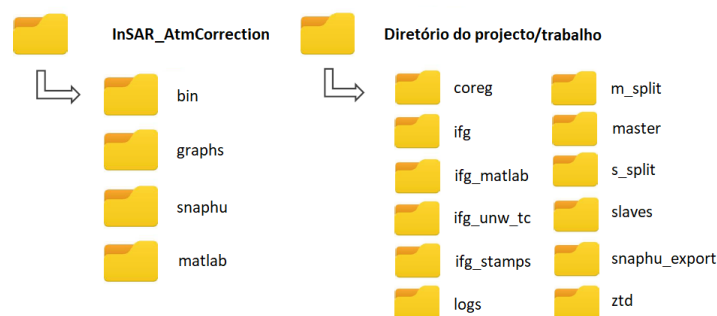


Figura 3.3: Estrutura e organização dos directórios.

Criaram-se dois directórios, o directório *InSAR_AtmosCorrection* e o directório do projecto/trabalho com vários subdirectórios cada. No directório *InSAR_AtmosCorrection* encontram-se os *Python scripts* (figura 3.4), os gráficos xml responsáveis pelo processo interferométrico (figura 3.11), o ficheiro de

configuração com localizações de diretórios e com parâmetros necessários para o processo (figura 4.3), os *scripts* responsáveis pela remoção dos efeitos da atmosfera (figura 3.13) e o executável responsável pelo desenrolamento de fase dos interferogramas (snaphu ¹). O diretório de trabalho, por sua vez, contém os produtos SLC, os mapas de atraso topográfico, os resultados dos processamentos e os *logs* desses mesmos processos.

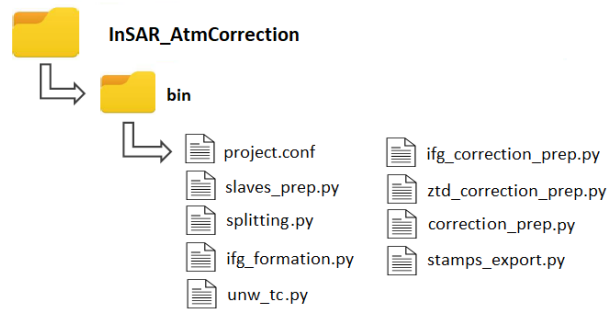


Figura 3.4: Subdirectório bin com os Python *scripts*.

3.2.1 Processamento interferométrico

No fluxograma da figura 3.5, encontram-se os "comandos" ou operadores utilizados no *software* escolhido para o processamento interferométrico (SNAP).

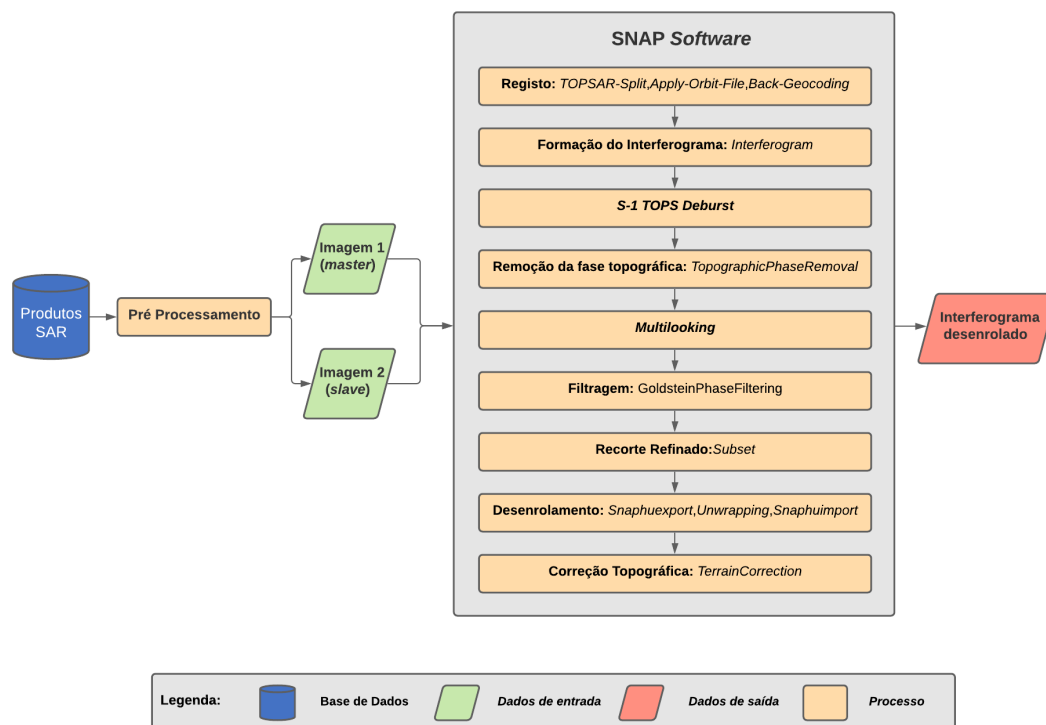


Figura 3.5: Fluxograma do processamento interferométrico.

¹SNAPHU (*Statistical-Cost, Network-Flow Algorithm for Phase Unwrapping*), é um algoritmo de desenrolamento de fase bidimensional que recupera dados de fase causado pela ambiguidade na fase Chen and Zebker [65, 66, 67]

A arquitectura SNAP fornece um *Graph Processing Framework* (GPF) que permite ao utilizador criar gráficos em formato XML, de modo a permitir o processamento em massa e de cadeias de processamento personalizadas e fazer o processamento a partir da linha de comandos, sem ter que recorrer ao *software* SNAP. Para a utilização dos gráficos, de modo a otimizar o processo e permitir processamento em cadeia, isto é, de modo a poder replicar o processo para um conjunto de imagens SAR é necessário fazer algumas alterações no código gerado, nomeadamente a identificação dos nomes das variáveis.

Na lógica da otimização dos processos, optou-se por dividir o processo, conforme ilustrado na figura 3.5, em três etapas (figura 3.6) resultando assim em quatro gráficos principais (num total de 9): *split_applyorbit.xml* (figura 3.7a), *coreg_ifg_mlflt_subset.xml* (figura 3.8), *snaphuexport.xml* (figura 3.9a) e *snaphuimport_tc.xml* (figura 3.9b). Cada processo equivale a um gráfico com excepção do terceiro e último processo, que se encontra separado em dois.

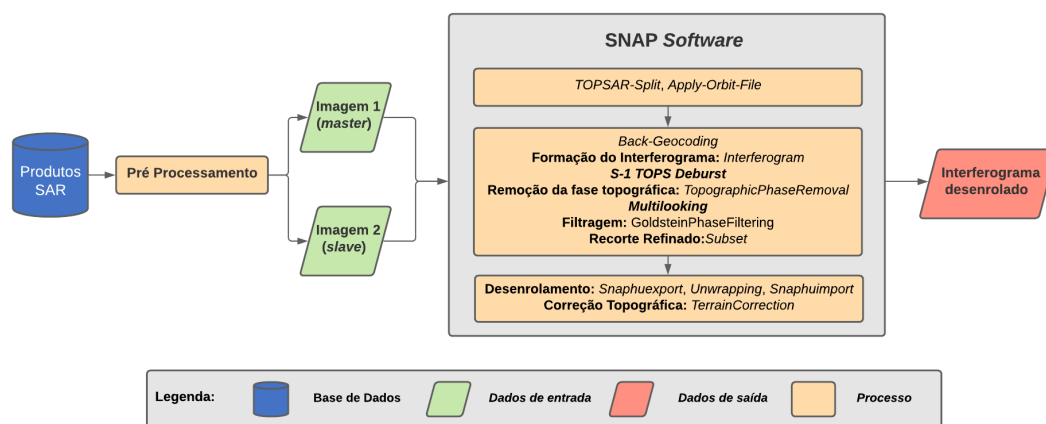


Figura 3.6: Fluxograma do processamento interferométrico condensado.

O primeiro processo, através do gráfico *split_applyorbit.xml*, é responsável, pelo *split* do *master* e do(s) *slave(s)* e pela actualização das respectivas órbitas. Caso as *slaves* cubram apenas parte da área de interesse do *master*, é possível utilizar as "fatias"(*slice*) correspondentes para garantir a completa cobertura da área com a assemblagem dos *slaves*. Neste caso, utiliza-se o gráfico *assemble_split_applyorbit.xml* (figura 3.7b), que à semelhança do *split_applyorbit.xml* faz o *split* e atualiza as orbitas. Os resultados deste processo são guardados nos subdirectórios *m_split* e *s_split* do directório de trabalho/projeto.

O segundo processo, recorrendo ao gráfico ilustrado na figura 3.8 ou ao gráfico *coreg_ifg_mlflt.xml*, caso não seja preciso o recorte refinado (*subset*), forma e disponibiliza os interferogramas, no directório *ifg*. Caso seja necessário a exportação para o STAMPS², o *script* responsável por este processo, através da configuração do devido parâmetro no ficheiro *project.conf a priori*, é capaz de seleccionar outro gráfico (figura 3.10a), muito idêntico ao *coreg_ifg_mlflt_subset.xml* que, para além de fornecer os interferogramas acima referidos, fornece no directório *coreg*, os co-registos dos produtos SLC e no directório *ifg_stamps*, os interferogramas sem "filtragem". A exportação é feita com o auxílio do gráfico

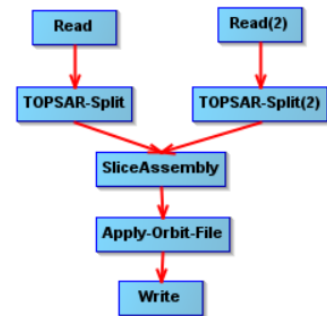
²STAMPS - *Stanford Method for Persistent Scatterers*, é um software que implementa um método de InSAR *persistent scatterer* (PS) que funciona mesmo em terrenos desprovidos de estruturas feitas pelo homem e/ou que sofrem deformações não constantes Marcel Wille [68]

da figura 3.10b.

O terceiro e último processo encontra-se dividido em três partes e, como referido acima, utiliza dois gráficos xml, *snaphuexport.xml* e *snaphuimport.tc.xml*. O gráfico *snaphuexport.xml* é responsável pela exportação de modo a poder efectuar o desenrolamento com o SNAPHU, enquanto que o gráfico *snaphuimport.tc.xml* é responsável pela importação dos dados desenrolados e pela aplicação da correcção topográfica. Esta divisão deu-se devido à impossibilidade de efectuar o desenrolamento com o auxílio do GPF, ou seja, devido à inexistência de um operador capaz de efectuar o desenrolamento. Trata-se de um algoritmo externo ao SNAP que, entretanto já possui um *plugin* (ESA [69]) que permite a sua realização manualmente. Para agilizar e automatizar o processo, fez-se o download do executável, *snaphu*, que se encontra no directório *InSAR_AtmCorrection* (figura 3.4), que com o auxílio do *script unw.tc.py* executa, em cadeia, os desenrolamentos dos interferogramas que se encontrarem no subdirectório *ifg* do directório de trabalho/projeto. Os resultados deste processo são guardados no subdirectório *ifg_unw.tc*.



(a) split_applyorbit.xml



(b) assemble_split_applyorbit.xml

Figura 3.7: Gráficos xml referente *Splitting* das imagens SAR e actualização de orbitas - split_applyorbit.xml, assemble_split_applyorbit.xml.

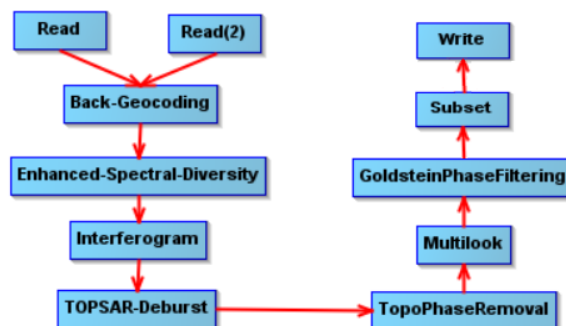
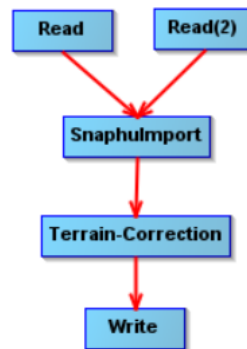


Figura 3.8: Gráfico xml referente ao Co-registo e formação dos interferogramas - coreg_ifg_mlflt_subset.

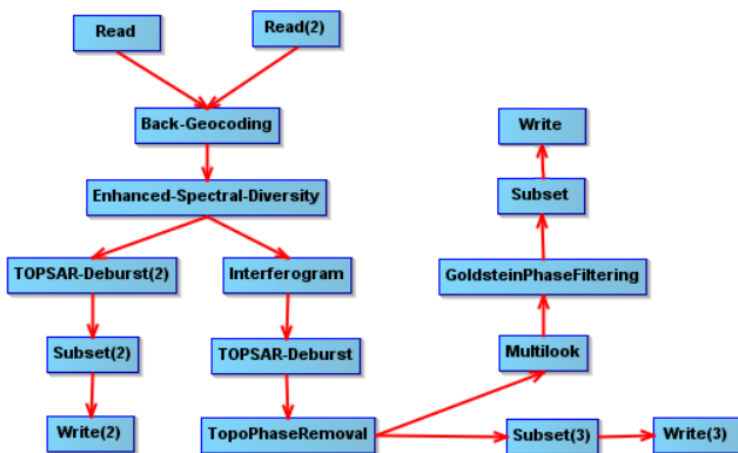


(a) snaphuexport.xml

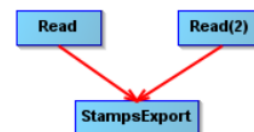


(b) snaphuimport.tc.xml

Figura 3.9: Gráficos xml referente ao Desenrolamento de fase e correcção topográfica - snaphuexport.xml, snaphuimport.tc.xml.



(a) coreg_ifg_mlflt_subset_export.xml



(b) stampsexport.xml

Figura 3.10: Gráficos xml referente ao Co-registo, formação dos interferogramas e exportação para o STAMPS - coreg_ifg_mlflt_subset_export.xml, stampsexport.xml.

Os gráficos encontram-se no subdirectório *graphs* como se pode observar na figura 3.11. No anexo B encontra-se um exemplo das alterações efectuadas no código de um dos gráficos de modo permitir o processamento em cadeia.

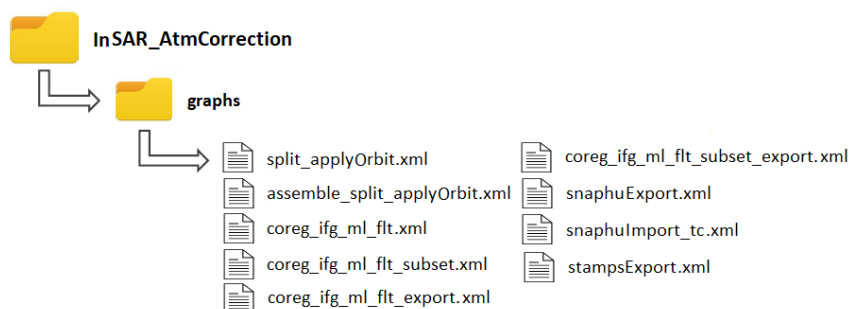


Figura 3.11: Gráficos xml do processo interferométrico.

3.2.2 Correção atmosférica

Como visto anteriormente no capítulo 2, dos factores mais contribuintes para a fase do interferograma encontram-se as condições atmosféricas. Este efeito é especialmente perturbador quando se tem por objetivo comparar interferogramas, visto que as condições climáticas são naturalmente diferentes todos os dias.

Para mitigar esse efeito, recorreu-se a um serviço *online* de correção atmosférica, denominada *Generic Atmospheric Correction Online Service for InSAR - GACOS* (www.gacos.net), desenvolvido na Universidade de Newcastle em 2017. Esta ferramenta, através de um modelo de Decomposição Troposférica Iterativa (ITD), gera mapas, num formato binário, de atraso total zenital de alta resolução espacial para corrigir medições InSAR Yu et al. [70, 71, 72], GACOS [73].

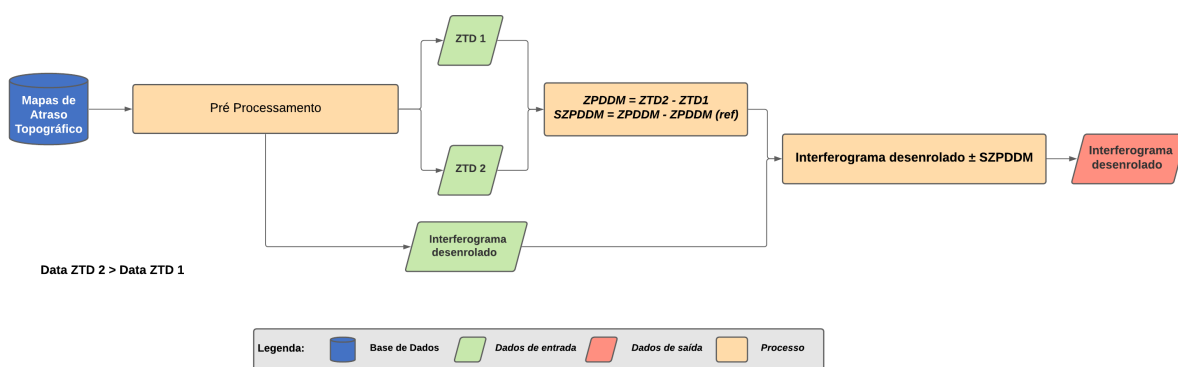


Figura 3.12: Fluxograma da correção atmosférica.

De modo a efetuar essa correção deve-se, primeiramente, solicitar os mapas na plataforma *online* do GACOS. Para a solicitação dos mapas de atraso troposférico é necessário preencher um formulário onde se tem em conta a área de estudo em questão e as respectivas datas de interesse. Será necessário a introdução de um email ao qual serão enviados os mapas, após a submissão do formulário.

Após solicitar os mapas de atraso troposférico, estes devem ser movidos ou copiados para o subdirectório *ztd* (figura 3.3) do directório de trabalho.

A correção troposférica é aplicada aos interferogramas, subtraindo ou somando os atrasos da atmosfera diferenciada (diferença entre os mapas de atraso referentes à duas datas), consoante a escolha das imagens master e slave na geração dos interferogramas.

A plataforma GAGOS, além de fornecer os mapas de atraso troposférico, fornece também uma *toolbox* do software *MATLAB*, *GACOS_UTILITY*, que aplica a correção. É necessário algum préprocessamento de modo a ter os dados dos interferogramas gerados anteriormente, disponíveis para esse processamento. Deste modo, houve a necessidade de desenvolver um *script*, *atm_correction.m*, capaz de ler os dados de um interferogramas após ser gerado com o auxílio do SNAP e, então, aplicar a correção. No anexo B, secção B.4, encontra-se o respectivo *script*, juntamente com o código fornecido pelo GACOS e outras considerações importantes referentes ao mesmo.

Para além do *atm_correction.m*, criou-se outro *script*, *main_auto_atm_correction.m*, que através de um ficheiro de texto, *atm_correction.txt*, gerado durante o pré processamento (figura 3.12), com parâmetros

e caminhos de parâmetros necessários, permite efectuar o processamento em cadeia, isto é, de vários interferogramas em simultâneo ao contrário do *atm_correction.m*. Estes *scripts* encontram-se no subdirectório *matlab* (figura 3.13) e durante o pré processamento, são copiados para o subdirectório *matlab_ifg* do directório do projecto (figura 3.3).

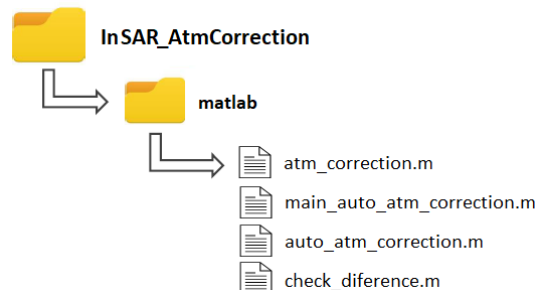


Figura 3.13: Subdirectório *matlab* com os *scripts* MATLAB.

3.3 Comparação entre os Interferogramas corrigidos

O *script*, *main_auto_atm_correction.m*, recorrendo à função, *check_diference.m*, que se desenvolveu, efectua uma comparação percentual, calcula o índice de similaridade estrutural (*Structural Similarity Index* - SSIM) e o erro quadrático médio (*Means Squared Error* - MSE) entre o interferograma em processamento e o processado no instante anterior. Tratam-se de comparações grosseiras que permitem perceber o comportamento ao longo do tempo dos interferogramas processados. A comparação percentual é feita a nível dos pixels, isto é, é feita uma subtração entre os interferogramas corrigidos onde contabilizam-se o numero de pixels diferentes que posteriormente é dividido pelo numero total de pixels. A similaridade estrutural é calculada recorrendo à função do Matlab *ssim* que se baseia no cálculo de três termos, luminância, contraste e estrutura Zhou Wang et al. [74]. O erro quadrático médio é calculado recorrendo a função do Matlab *immse*.

Esses dados são calculados e guardados num ficheiro de texto de nome *ifg_comparison.txt* no directório *ifg_matlab* durante a correção atmosférica.

3.4 Visualização dos dados espaciais

De modo a visualizar os interferogramas após a remoção do efeito da atmosfera, guarda-se todos os dados do processo e uma imagem com os interferogramas antes e depois da correção. Também se efectua a exportação dos interferogramas corrigidos para GeoTIF³ durante o processamento. Os ficheiros encontram-se nos subdirectórios *correction_data* referentes a cada interferograma do directório *ifg_matlab* e os GeoTIF encontram-se no subdirectório *TIFF_images* do mesmo directório. Assim sendo, para visualizar os dados, apenas é preciso uma aplicação capaz de ler os ficheiros acima referidos,

³GeoTIF (*Geo Tagged Image File Format*) - metadados de domínio público que possui as informações de georreferenciamento embutidas no arquivo de imagem, de forma que não há necessidade de um arquivo *.tifw* de acompanhamento.

podendo esta aplicação ou plataforma ser construída de base a partir de uma lista de requisitos previamente identificados ou uma já existente.

Capítulo 4

Caso de estudo

No presente capítulo são apresentadas as áreas que foram utilizadas para os testes de processamento no âmbito desta dissertação, assim como as correspondentes séries temporais de imagens SAR utilizadas.

Foram seleccionadas áreas urbanas com diferentes tipos de estruturas, desde estradas, pontes, a estruturas de grandes dimensões como estádios, arenas e palácios. Para a escolha das mesmas, teve-se em consideração tanto a disponibilidade das imagens SAR, como o potencial de aplicação ao estudo de casos concretos com interesse para algumas entidades.

4.1 Enquadramento Geográfico

A área em estudo corresponde aos concelhos de Lisboa, Oeiras e Cascais, pertencentes à Área Metropolitana de Lisboa (Figura 4.1), numa dimensão total de 243 km^2 (tabela 4.1). O período temporal considerado para o estudo está compreendido entre Julho de 2019 e Junho de 2020. Apenas serão considerados dados de um dos satélites Sentinel-1 e de uma única direção de aquisição dessas informações, por questões relacionadas com capacidade de processamento.

Concelhos	Freguesias	Área [Km^2]	Habitantes
Lisboa	53	100.05	547 733 (2011)
Oeiras	10	45.88	172 120 (2011)
Cascais	6	97.40	206 479 (2011)

Tabela 4.1: Características da área de estudo.

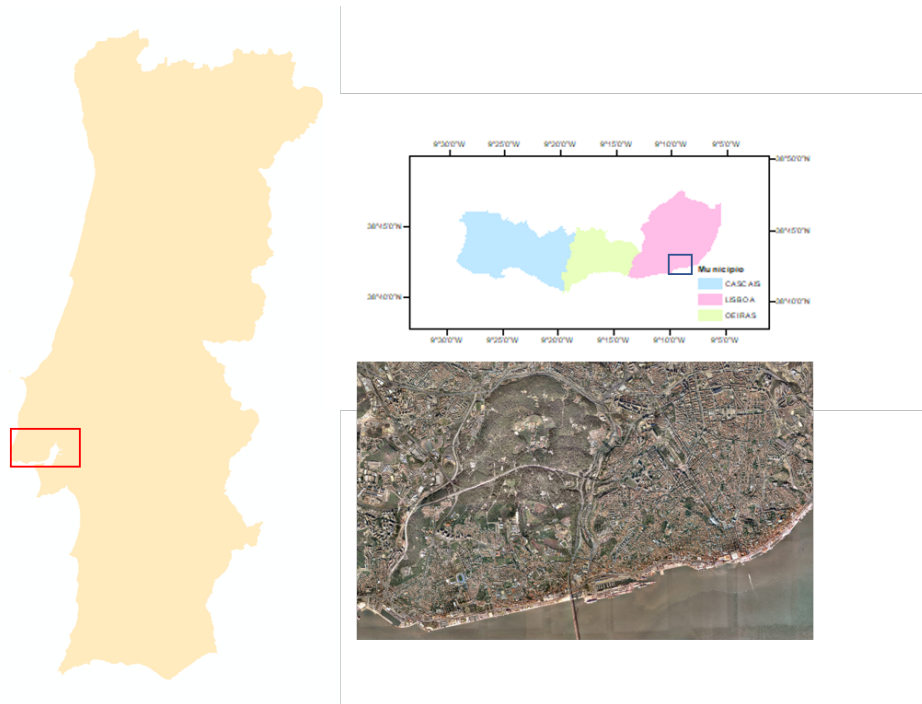


Figura 4.1: Área de estudo.

4.2 Processamento das Imagens SAR

4.2.1 Requisitos de sistema

A geração de interferogramas é um processo que consome muitos recursos computacionais, pelo que se recomenda no mínimo uma CPU quad-core com 16 GB de RAM. De modo a poder processar o equivalente a um ano de dados espaciais mantendo o mesmo master é necessário ainda cerca de 140 GB de espaço de armazenamento livre em disco.

São considerados três tipos de espaço de armazenamento:

- Armazenamento temporário (0.5 GB) de arquivos localmente durante o processamento que são descartados no final das etapas de processamento.
- Armazenamento intermediário (136 GB) de dados e produtos intermediários que são necessários durante um ciclo completo de processamento.
- Armazenamento permanente (3.5 GB) para dados auxiliares estáticos e produtos finais, que podem ser mantidos indefinidamente.

Na prática, apenas o armazenamento permanente é obrigatório, enquanto o armazenamento temporário e intermediário pode ser escalado dependendo de fatores como o orçamento, a implementação algorítmica e da quantidade de dados que se deseja processar.

Existem soluções de processamento para pesquisa e desenvolvimento de serviços provenientes da ESA, como o *ESA Research and Service Support* que oferece a possibilidade de obter uma Máquina Virtual com recursos definidos pelo usuário para processamento temporal (<https://wiki.services.eoportal.org/tiki-index.php?page=RSS+CloudToolbox+Service>).

4.2.2 Imagens SAR

Na tabela 4.2, podemos encontrar as características principais ou necessárias relativa às imagens SAR, de modo a poder efectuar o processamento dos interferogramas tendo em conta a área de interesse.

Sensor	Modo aquisição	Direcção	Nível	Tipo	Órbita Relativa
Sentinel-1A	IW	Ascendente	L1	SLC	45

Tabela 4.2: Características das imagens SAR da área de estudo.

De modo a obter os produtos SLC referentes ao período de um ano com as características referidas na tabela acima, a partir do *script download.py*, tem-se, primeiramente, que preencher o ficheiro de configuração, *download.conf*, com os devidos parâmetros, como ilustrado na figura 4.2.

```
##### CONFIGURATION FILE #####
#####
# DOWNLOAD DEFINITION #
#####
USERNAME=*****
PASSWORD=*****
DOWNLOADFOLDER=C:\Users\Administrador\Desktop\SLC_products
LOGFOLDER=C:\Users\Administrador\Desktop\SLC_products
#-----PROGRESSDISPLAY-----#
# 1-True 0-False
SHOWPROGRESS=1
#####
# PARAMETERS (Only one option **) #
#####
##### Option 1 #####
COMPLETEQUERY=
##### Option 2 #####
#-----INTEREST AREA-----#
LONMIN=-9.487535829905955|
LATMIN=38.67412870285344
LONMAX=-9.075262172839683
LATMAX=38.74132698016754
#---- SENSING DATE ----#
SSTART=
SEND=
#--- INGESTION DATE ----#
ISTART=2019-07-08T00:00:00.000Z
IEND=2020-07-08T23:59:59.999Z
#-----#
PLATFORMNAME=Sentinel-1
FILENAME=S1A_*
PRODUCTTYPE=SLC
SENSOROPERATIONALMODE=IW
ORBITNUMBER=
RELATIVEORBITNUMBER=45
POLARISATIONMODE=
CLOUDCOVERPERCENTAGE=
ORBITDIRECTION=Ascending
SWATHIDENTIFIER=
#####
# ** format specified in https://scihub.copernicus.eu/userguide/FullTextSearch
```

Figura 4.2: Ficheiro de configuração para download de imagens SAR - *download.conf*.

Para obter os mesmos produtos SLC, através do *dhusget script*, deve-se utilizar como *query* o seguinte excerto de código:

```
(footprint: "Intersects(POLYGON((-9.508171819895166 38.67099357178023, -9.039679074194092
38.67099357178023, -9.039679074194092 38.785511063988366, -9.508171819895166
38.785511063988366, -9.508171819895166 38.67099357178023)))") AND (ingestionDate
:[2019-07-08T00:00:00.000Z TO 2020-07-08T23:59:59.999Z]) AND ((platformname:Sentinel
-1 AND filename:S1A_* AND producttype:SLC AND sensoroperationalmode:IW AND
relativeorbitnumber:45))
```

É de realçar que é possível utilizar o mesmo excerto de código com o *download.py*, definindo o parâmetro *COMPLETEQUERY* do ficheiro de configuração (figura 4.2) como sendo esse excerto.

O Anexo A.2 apresenta informação sobre as séries temporais de imagens SAR usadas nesta dissertação, adquiridas pelos *scripts* desenvolvidos.

4.2.3 Processamento Interferométrico e correção atmosférica

Para dar início ao processamento das imagens, primeiramente, é necessário ter disponível os produtos SLC e os mapas de atraso topográfico nos respectivos subdirectórios (figura 3.3).

Foi escolhida a imagem de 26 de junho de 2020 como principal (*master*), preencheu-se o ficheiro de configuração com os respectivos caminhos e parâmetros necessários, como demonstrado no exemplo da figura abaixo:

```
##### CONFIGURATION FILE #####
#####
# PROJECT DEFINITION #
#####
PROJECTFOLDER=C:\Project
GRAPHSFOLDER=C:\Users\Administrador\Desktop\InSAR_AtmosCorrection\graphs
#####
# INSAR PROCESSING #
#####
MASTER=S1A_IW_SLC__1SDV_20200626T183528_20200626T183556_033192_03D858_AC12.zip
IWX=IW2
FIRSTBURST=4
LASTBURST=5
POLARIZATION=VV
#--APPLYORBITFILE---#
# 0-No or 1-Yes
CONTINUEONFAIL=0
#-----SUBSET-----#
LONMIN=-9.520502090454102
LATMIN=38.69184494018555
LONMAX=-9.05436897277832
LATMAX=38.827796936035156
#-----UNWRAPPING-----#
SNAPHUFOLDER=C:\Users\Administrador\Desktop\InSAR_AtmosCorrection\snaphu
#--EXPORTTOSTAMPS---#
# 0-No or 1-Yes
STAMPSEXPORT=0
#-----SNAP GPT-----#
GPTBIN_PATH=C:\Program Files\snap\bin\gpt
#####
# COMPUTING RESOURCES TO EMPLOY
#####
CPU=8
CACHE=16G
#####
# ATM CORRECTION #
#####
MATLABFOLDER=C:\Users\Administrador\Desktop\InSAR_AtmosCorrection\matlab
ZTDFOLDER=C:\Produtos\ztd
LASTCORRECTEDIFGDATE=
LASTCORRECTEDIFGFOLDER=
#####
```

Figura 4.3: Ficheiro de configuração do processamento de imagens - *project.conf*.

Depois de ajustar o ficheiro de configuração dá-se início ao processo interferométrico, utilizando os *scripts* mencionados no capítulo 3 (figura 3.4).

Como descrito anteriormente no capítulo anterior, subsecção 3.2.2, são "gerados" dois *scripts* Matlab que permitem tanto a correcção individual (*atm_correction.m*) de um interferograma bem como a correcção em cadeia (*main_auto_atm_correction.m*) dos interferogramas disponíveis no sub directório ifg_matlab. Ambos os *scripts* encontram-se no mesmo subdirectório (matlab.ifg) acompanhados do ficheiro de texto *atm_correction.txt*, com os caminhos dos interferogramas e parâmetros que servem de auxílio ao processamento em cadeia.

Na figura 4.4, podemos observar cinco interferogramas, antes e depois de aplicar a correcção, referentes aos últimos interferogramas da série temporal processada.

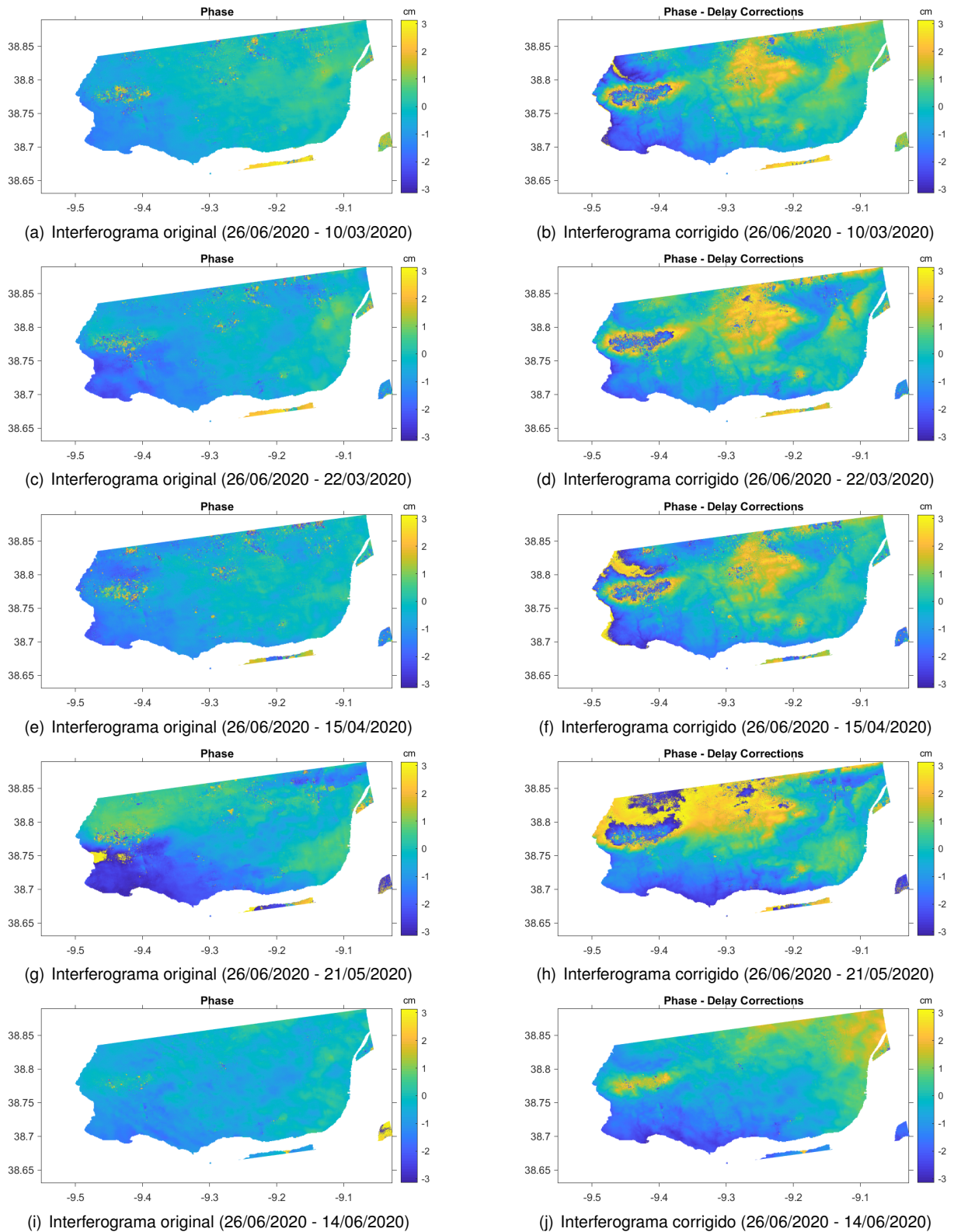


Figura 4.4: Interferogramas antes e após a correção atmosférica referentes aos últimos produtos SAR processados. Esquerda: Interferograma original. Direita: Interferograma após a correção atmosférica.

4.3 Comparação entre os Interferogramas corrigidos

Do ficheiro gerado (*ifg_comparison.txt*) referente às comparações entre os interferogramas corrigidos sequencialmente, isto é, entre um interferograma em processamento com o interferograma corrigido no instante anterior obteve-se os dados ilustrados nas figuras 4.5 e 4.6.

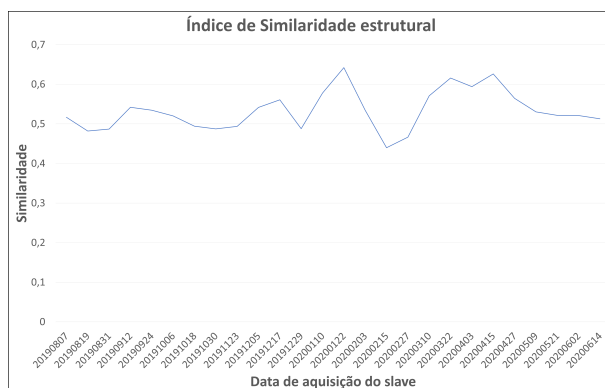


Figura 4.5: Índice de Similaridade estrutural entre os interferogramas corrigidos sequencialmente.



Figura 4.6: Erro quadrático médio entre os interferogramas corrigidos sequencialmente.

Do gráfico 4.5 relativo ao Índice de Similaridade Estrutural pode-se observar que os valores encontram-se, maioritariamente, acima de 0.5 o que revela uma alta taxa perceptual de semelhança ou de não degradação entre os interferogramas.

Do gráfico 4.6 relativo ao Erro Quadrático Médio pode-se observar que houve uma diminuição do erro com o avançar do processamento e diminuição da linha de base temporal.

Esses valores e variações podem ser em parte justificados com a linha de base temporal longa e linha de base perpendicular "curta" dos produtos utilizados, visto que uma das etapas cruciais para geração de um mapa de subsidência ou de um modelo digital de elevação (*Digital Elevation Model - DEM*) bem sucedida é a seleção do par de imagens com as propriedades adequadas. O tempo entre as imagens deve ser o mais curto possível para reduzir o risco de de-correlação temporal da fase e a distância entre as posições dos satélites no momento da aquisição da imagem deve estar entre 150 e 300 metros Ferretti et al. [32].

4.4 Visualização dos dados espaciais

Para visualização dos interferogramas obtidos após a remoção do efeito da atmosfera recorreu-se a uma plataforma de gestão de dados espaciais *online*, uma *WebGIS*, o *iMSEP* (*internet-based information Management System for Environmental Protection and Disaster Risk Management*) de modo a permitir ao público, em geral, visualizar, analisar e compartilhar as informações geográficas.

4.4.1 Plataforma iMSEP

iMSEP (<https://imsep.gis.lu.se/>) é um Sistema de Gestão de Informação baseado na Internet para Proteção Ambiental e Gestão de Risco de Desastres Naturais que fornece aos indivíduos ou instituições, cujo papel seja planejar e tomar decisões, acesso a dados espaciais e análises. Trata-se de um sistema desenvolvido na *Lund University*, ao abrigo do projeto europeu EMME (*International MSc Educational Programme in Environmental Management and Modelling*) do qual o IST-UL é parceiro, sob coordenação e gestão do Professor Associado Dr. Ali Mansourian. O *iMSEP* e *emGeo* (*environmental management Geoportal*) foram integrados para fornecer aos usuários funções GIS e IDE num único sistema.

Na figura 4.7 podemos observar alguns dos resultados obtidos ao fim do processamento:

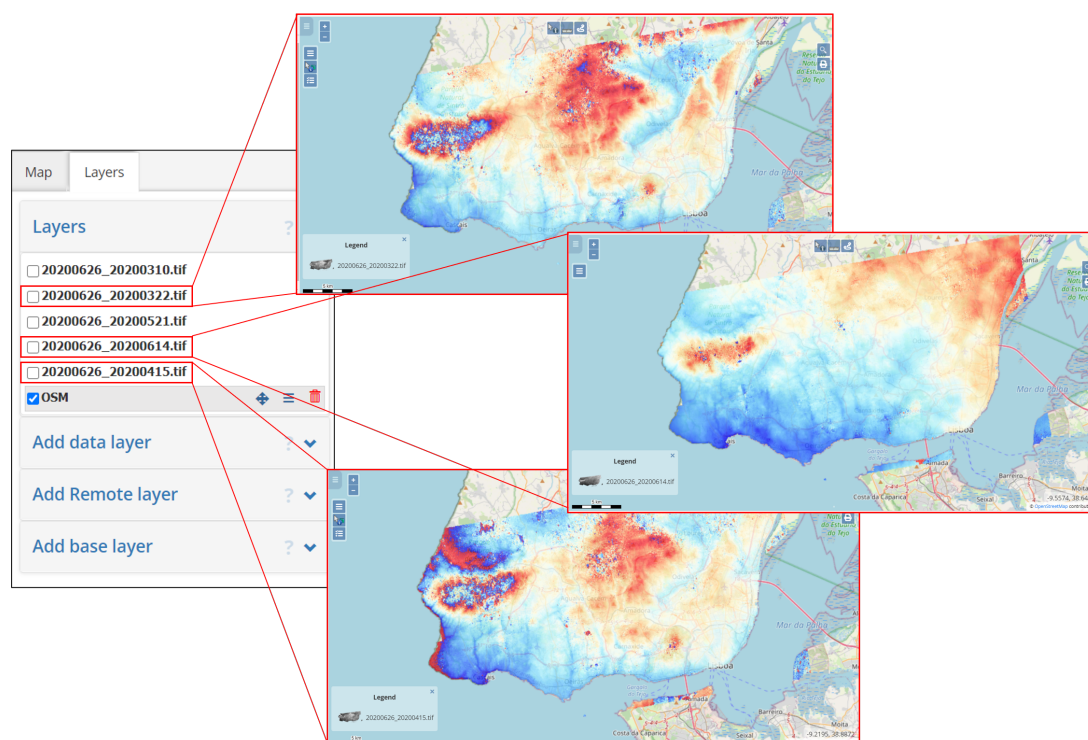


Figura 4.7: Visualização parcial dos interferogramas corrigidos na plataforma iMSEP.

Para visualizar a totalidade dos resultados obtidos, consultar a plataforma *iMSEP*, <https://imsep.gis.lu.se/>.

Capítulo 5

Conclusão

Os arquivos de dados referentes à observação da Terra estão a expandir a uma velocidade sem precedentes, tanto a nível de tamanho quanto a nível de variedade. Cada vez mais, os sensores de mapeamento remoto existentes e os em desenvolvimento, fornecem ou fornecerão dados de observação da Terra, caracterizados por grandes coberturas, elevadas resoluções espaciais e altas frequências de revisitação. Em paralelo, o poder da computação tem crescido, permitindo assim o processamento desses dados em escala global, representando uma oportunidade única no estudo e o conhecimento integrado de fenómenos espaciais à escala de toda a superfície terrestre.

Nesta dissertação, começou-se por apresentar os aspetos básicos dos sistemas SAR e algumas missões espaciais que utilizam esses radares, com maior ênfase ao programa Copernicus - Sentinel-1. Foram apresentados os conceitos relacionados com a interferometria com recurso a imagens SAR e abordados de forma sucinta as diferentes etapas que compõem o processo interferométrico. Igualmente merecedores de atenção foram os aspetos referentes às infraestruturas de dados espaciais, que são infraestruturas de interoperabilidade, pela capacidade de diferentes aplicativos que usam diferentes linguagens ou conceitos comunicarem entre si. Analisaram-se as componentes relacionadas com a arquitetura destes sistemas e foram exemplificadas duas plataformas (uma plataforma InSAR e um Geoportal), com algumas das suas referências mais técnicas.

Na sequência do anteriormente dito, com o intuito de contribuir para a promoção do uso de imagens SAR nos serviços de monitorização da superfície terrestre, recorrendo às técnicas interferométricas, criou-se uma linha de processamento versátil capaz de processar dados SAR, produzir interferogramas e remover o efeito da atmosfera na fase dos mesmos recorrendo a uma plataforma online, para posteriormente serem comparados e visualizadas numa plataforma online. Como caso de estudo, foram criados e corrigidos interferogramas referentes ao período entre Julho de 2019 e Junho de 2020 a sensivelmente um ano civil, da região contemplada pelos concelhos de Lisboa, Oeiras e Cascais. Estes resultados encontram-se disponíveis para consulta na plataforma de visualização de dados espaciais iMSEP (<https://imsep.gis.lu.se/>), disponibilizada no âmbito do projeto europeu EMME (*International MSc Educational Programme in Environmental Management and Modelling*).

Na produção da cadeia interferométrica considerou-se a imagem de 26 de junho de 2020 como

master e no total foram calculados e disponibilizados 27 interferogramas. Para todos foram aplicadas as correções atmosféricas disponibilizadas pela plataforma GACOS, que embora satisfatórias à escala global, na medida em que se observam correções relacionadas com a correlação dos efeitos atmosféricos com a orografia do terreno, não são suficientemente detalhadas para uma análise mais fina. Trata-se de um facto esperado, dado que o GACOS é uma solução elaborada para estudos de grandes áreas (por exemplo, para uma faixa interferométrica inteira). No sentido de agilizar a interpretação dos resultados foi criado um *script* que compara pixel a pixel interferogramas sequenciais. Nesta apreciação foram utilizados as variáveis relativas aos índices de similaridade estrutural e o erro médio quadrático, apresentados nas figuras 4.5 e 4.6.

5.1 Perspectivas de desenvolvimentos futuros

As pesquisas realizadas no âmbito deste trabalho conducentes à definição da metodologia e aplicação ao caso de estudo, oferecem boas perspectivas no interesse pela implementação deste tipo de soluções, por parte das entidades responsáveis pela gestão e análise de informação espacial. A solução apresentada assegura a exequibilidade da cadeia de processamento interferométrico automatizado, muito embora seja ainda possível otimizar processos e implementar outras abordagens de processamento interferométrico. Assim como trabalho futuro são apresentados os seguintes pontos:

- Avaliar a possibilidade de escalar o processamento a nível do país (continental), o qual requereria mais capacidade de processamento e armazenamento. Para um ano de imagens processadas de um dos satélites Sentinel-1 com uma única direção de aquisição de informações, seriam precisos cerca de 860 GB de espaço de armazenamento de modo a poder efetuar o processamento referente a todo o território português (continental);
- Identificação e utilização de outras técnicas de remoção do efeito da atmosfera, especialmente mais detalhadas e com recurso a dados in-situ. Idealmente na cadeia de processamento interferométrico a escolha do modelo deveria ser oferecida como opção ao utilizador;
- Automatização de outras técnicas interferométricas multi-temporais. Permitir que o utilizador, em função do objetivo, possa escolher a técnica interferométrica mais indicada, promovendo o interesse por parte do utilizador;
- Identificação de outros critérios de comparação de interferogramas e emissão de alertas em caso de diferenças superiores a um determinado valor limite;
- Desenvolver uma aplicação web que permita, aos interessados, visualizar os dados produzidos.

Bibliografia

- [1] R. Grandin, M.-P. Doin, L. Bollinger, B. Pinel-Puysségur, G. Ducret, R. Jolivet, and S. N. Sapkota. Long-term growth of the himalaya inferred from interseismic insar measurement. *Geology*, 40(12): 1059–1062, 2012.
- [2] P. J. González, M. Bagnardi, A. J. Hooper, Y. Larsen, P. Marinkovic, S. V. Samsonov, and T. J. Wright. The 2014–2015 eruption of fogo volcano: Geodetic modeling of sentinel-1 tops interferometry. *Geophysical Research Letters*, 42(21):9239–9246, 2015.
- [3] E. Chaussard, S. Wdowinski, E. Cabral-Cano, and F. Amelung. Land subsidence in central mexico detected by alos insar time-series. *Remote sensing of environment*, 140:94–106, 2014.
- [4] DLR. Terrasar-x - germany's radar eye in space, 2020. URL <https://www.dlr.de/content/en/articles/missions-projects/terrasar-x/terrasar-x-earth-observation-satellite.html>. Acedido: 2020-09-05.
- [5] ASI. Cosmo-skymed, 2020. URL <https://www.asi.it/en/earth-science/cosmo-skymed/>. Acedido: 2020-09-05.
- [6] JAXA. About alos-palsar, 2020. URL <https://www.eorc.jaxa.jp/ALOS/en/about/palsar.htm>. Acedido: 2020-09-05.
- [7] CSA. What is radarsat-1, 2020. URL <https://www.asc-csa.gc.ca/eng/satellites/radarsat1/what-is-radarsat1.asp>. Acedido: 2020-09-05.
- [8] M. I. Skolnik. Radar handbook second edition. *McGrawHill*, 1990.
- [9] ESA. História da observação da terra, 2010. URL https://www.esa.int/SPECIALS/Eduspace_PT/SEMAB965P1G_0.html. Acedido: 2020-09-09.
- [10] T. Farr and M. Kobrick. Shuttle radar topography mission produces a wealth of data. *EOS Transactions*, 81:583–585, 11 2000. doi: 10.1029/EO081i048p00583.
- [11] P. A. Rosen, S. Hensley, K. Wheeler, G. Sadowy, T. Miller, S. Shaffer, R. Muellerschoen, C. Jones, H. Zebker, and S. Madsen. Uavsar: a new nasa airborne sar system for science and technology research. In *2006 IEEE Conference on Radar*, pages 8 pp.–, 2006. doi: 10.1109/RADAR.2006.1631770.

- [12] O. Monserrat, M. Crosetto, and G. Luzi. A review of ground-based sar interferometry for deformation measurement. *ISPRS Journal of Photogrammetry and Remote Sensing*, 93:40 – 48, 2014. ISSN 0924-2716. doi: <https://doi.org/10.1016/j.isprsjprs.2014.04.001>. URL <http://www.sciencedirect.com/science/article/pii/S0924271614000884>.
- [13] F. Di Traglia, T. Nolesini, A. Ciampalini, L. Solari, W. Frodella, F. Bellotti, A. Fumagalli, G. De Rosa, and N. Casagli. Tracking morphological changes and slope instability using spaceborne and ground-based sar data. *Geomorphology*, 300:95–112, 2018.
- [14] R. Bamler. Principles of synthetic aperture radar. *Surveys in Geophysics*, 21(2-3):147–157, 2000.
- [15] A. P. Falcão. *Interferometria RADAR*. PhD thesis, Instituto Superior Técnico, 11 2000.
- [16] D. P. Lusch. Introduction to environmental remote sensing. *Center for Remote Sensing and GIS Michigan State University: East Lansing, MI, USA*, 1999.
- [17] A. Moreira, P. Prats-Iraola, M. Younis, G. Krieger, I. Hajnsek, and K. P. Papathanassiou. A tutorial on synthetic aperture radar. *IEEE Geoscience and Remote Sensing Magazine*, 1(1):6–43, 2013.
- [18] ESA. Satellite frequency bands, 2019. URL https://www.esa.int/Applications/Telecommunications_Integrated_Applications/Satellite_frequency_bands. Acedido: 2020-08-09.
- [19] UNAVCO. Synthetic aperture radar (sar) satellites, 2014. URL <https://www.unavco.org/instrumentation/geophysical/imaging/sar-satellites/sar-satellites.html>. Acedido: 2020-08-28.
- [20] D. Giuli. Polarization diversity in radars. *IEEE Proceedings*, 74:245–269, Feb. 1986.
- [21] N. R. Canada. Polarization in radar systems, Nov 2014. URL <https://www.nrcan.gc.ca/earth-sciences/geomatics/satellite-imagery-air-photos/satellite-imagery-products/educational-resources/9567>. Acedido: 2020-10-28.
- [22] J. Sanden and J. Johannes. Radar remote sensing to support tropical forest management /. *Prof.dr. R.A. Feddes, Prof.dr.ir. R.A.A. Oldeman (supervisors); Dr.ir. D.H. Hoekman (co-supervisor). Wageningen Agricultural University, The Netherlands (1997) 330 pp. ISBN 90-5485-778-1., 01 1997.*
- [23] J. J. van Zyl, B. D. Chapman, P. Dubois, and J. Shi. The effect of topography on sar calibration. *IEEE Transactions on Geoscience and Remote Sensing*, 31(5):1036–1043, 1993.
- [24] A. Braun. *Radar satellite imagery for humanitarian response. Bridging the gap between technology and application*. PhD thesis, Universität Tübingen, 08 2019.
- [25] N. R. Canada. Radar image distortions, Nov 2015. URL <https://www.nrcan.gc.ca/maps-tools-publications/satellite-imagery-air-photos/remote-sensing-tutorials/microwave-remote-sensing/radar-image-distortions/9325>. Acedido: 2020-10-28.

- [26] ESA. Radar course 3. URL https://earth.esa.int/web/guest/missions/esa-operational-eo-missions/ers/instruments/sar/applications/radar-courses/content-3/-/asset_publisher/mQ9R7ZVkJg5P/content/radar-course-3-slant-range-ground-range. Acedido: 2020-10-28.
- [27] ESA. Sentinel-1 overview, September 2013. URL <https://sentinel.esa.int/web/sentinel/missions/sentinel-1/overview>. Acedido: 2020-08-22.
- [28] ESA. Overview, 2017. URL <https://scihub.copernicus.eu/userguide/WebHome>. Acedido: 2020-08-22.
- [29] ESA. *SAFE Specification*, 2013. URL <https://sentinel.esa.int/web/sentinel/user-guides/sentinel-1-sar/data-formats/safe-specification>. Acedido: 2020-08-22.
- [30] EOS. *Sentinel 1*, 2020. URL <https://eos.com/sentinel-1/>. Acedido: 2020-09-18.
- [31] ESA. Mission summary, 2013. URL <https://sentinel.esa.int/web/sentinel/missions/sentinel-1/overview/mission-summary>. Acedido: 2020-10-31.
- [32] A. Ferretti, A. V. Monti-Guarnieri, C. Prati, F. Rocca, D. Massonnet, et al. *INSAR Principles B*. ESA publications, 2007.
- [33] T. Lillesand, R. W. Kiefer, and J. Chipman. *Remote sensing and image interpretation*. John Wiley & Sons, 2004.
- [34] F. Rocca. An overview of sar interferometry. In *Proc. 3re ERS Symp, Florence, 1997*, 1997.
- [35] R. Hanssen. *Radar Interferometry: Data Interpretation and Error Analysis*. Remote Sensing and Digital Image Processing. Springer Netherlands, 2001. ISBN 9780792369455. URL <https://books.google.pt/books?id=bqNkJUk4wtMC>.
- [36] I. H. Woodhouse. *Introduction to microwave remote sensing*. CRC press, 2005.
- [37] R. M. Goldstein and C. L. Werner. Radar interferogram filtering for geophysical applications. *Geophysical research letters*, 25(21):4035–4038, 1998.
- [38] Y. Huang and J. Van Genderen. Comparison of several multi-look processing procedures in insar processing for ers-1&2 tandem mode. In *ERS SAR Interferometry*, volume 406, page 215, 1997.
- [39] R. M. Goldstein, H. A. Zebker, and C. L. Werner. Satellite radar interferometry: Two-dimensional phase unwrapping. *Radio science*, 23(4):713–720, 1988.
- [40] H. A. Zebker, P. A. Rosen, R. M. Goldstein, A. Gabriel, and C. L. Werner. On the derivation of coseismic displacement fields using differential radar interferometry: The landers earthquake. *Journal of Geophysical Research: Solid Earth*, 99(B10):19617–19634, 1994.
- [41] H. A. Zebker and Y. Lu. Phase unwrapping algorithms for radar interferometry: residue-cut, least-squares, and synthesis algorithms. *JOSA A*, 15(3):586–598, 1998.

- [42] A. Braun. Dem generation with sentinel-1 workflow and challenges, January 2020. URL <http://step.esa.int/docs/tutorials/S1TBX%20DEM%20generation%20with%20Sentinel-1%20IW%20Tutorial.pdf>. Acedido: 2020-09-05.
- [43] N. R. Council, M. S. Committee, et al. *Toward a coordinated spatial data infrastructure for the nation*. National Academies Press, 1993.
- [44] A. Rajabifard and I. P. Williamson. Spatial data infrastructures: concept, sdi hierarchy and future directions. 2001.
- [45] A. Phillips, I. Williamson, and C. Ezigbalike. Spatial data infrastructure concepts. *Australian Surveyor*, 44(1):20–28, 1999.
- [46] N. R. Budhathoki, Z. Nedovic-Budic, et al. Reconceptualizing the role of the user of spatial data infrastructure. *GeoJournal*, 72(3-4):149–160, 2008.
- [47] A. Rajabifard, F. Escobar, and I. P. Williamson. Hierarchical spatial reasoning applied to spatial data infrastructures. *Cartography*, 29(2):41–50, 2000.
- [48] I. Masser. All shapes and sizes: the first generation of national spatial data infrastructures. *International Journal of Geographical Information Science*, 13(1):67–84, 1999.
- [49] D. J. Maguire and P. A. Longley. The emergence of geoportals and their role in spatial data infrastructures. *Computers, environment and urban systems*, 29(1):3–14, 2005.
- [50] M. G. Tait. Implementing geoportals: applications of distributed gis. *Computers, Environment and Urban Systems*, 29(1):33–47, 2005.
- [51] C. Cipolloni. *Architecture of Spatial Data Infrastructure (SDI) (DRAFT)*, 2018.
- [52] L. P. S. Fortes and V. O. H. de Araújo. Permanent committee for geospatial data infrastructure of the americas (pc-idea), 2013.
- [53] L. Bernard and M. Craglia. Sdi-from spatial data infrastructure to service driven infrastructure. In *Research Workshop on Cross-Learning Between Spatial Data Infrastructures and Information Infrastructures, Enschede, The Netherlands*. Citeseer, 2005.
- [54] W3C. Extensible markup language (xml), 2016. URL <https://www.w3.org/XML/>.
- [55] W3C. Web services architecture requirements, 2004. URL <https://www.w3.org/TR/wsa-reqs/>.
- [56] S. Vinoski. Web services interaction models, part 1: Current practice. *IEEE Internet Computing*, 6(3):89, 2002.
- [57] M. Klopfer. Interoperability & open architectures: an analysis of existing standardisation processes & procedures. *OGC White Paper. Consortium, OG, Open Geospatial Consortium: 26p*, 2005.
- [58] NGU. Insar norway, 2018. URL <https://www.ngu.no/en/topic/insar-norway>. Acedido: 2020-09-14.

- [59] A. Hooper. A multi-temporal insar method incorporating both persistent scatterer and small baseline approaches. *Geophysical Research Letters*, 35(16), 2008.
- [60] CLMS. European ground motion service: Service implementation plan and product specification document, 2020.
- [61] NG. The national geoportal, 2017. URL <http://www.pcn.minambiente.it/mattm/en/the-national-geoportal/>. Acedido: 2020-09-14.
- [62] R. Rossi. Considerazioni sui nuovi servizi di conversione di coordinate del geoportale nazionale. *GEOmedia*, 16(6), 2012.
- [63] ESA. Copernicus open access hub, batch scripting - dhusget script, 2020. URL https://scihub.copernicus.eu/twiki/do/view/SciHubUserGuide/BatchScripting#dhusget_script. Acedido: 2020-08-22.
- [64] K. S. Andy Hooper, David Bekaert. *Sentinelsat Documentation*. Sentinelsat, 2019. URL https://sentinelsat.readthedocs.io/_/downloads/en/stable/pdf/.
- [65] C. W. Chen and H. A. Zebker. Network approaches to two-dimensional phase unwrapping: intractability and two new algorithms. *JOSA A*, 17(3):401–414, 2000.
- [66] C. W. Chen and H. A. Zebker. Two-dimensional phase unwrapping with use of statistical models for cost functions in nonlinear optimization. *JOSA A*, 18(2):338–351, 2001.
- [67] C. W. Chen and H. A. Zebker. Phase unwrapping for large sar interferograms: Statistical segmentation and generalized network models. *IEEE Transactions on Geoscience and Remote Sensing*, 40(8):1709–1719, 2002.
- [68] K. C. Marcel Wille. *StaMPS/MTI Manual.*, 2013. URL https://homepages.see.leeds.ac.uk/~earahoo/stamps/StaMPS_Manual_v3.3b1.pdf.
- [69] ESA. Phase unwrapping with snaphu, 2019. URL <https://step.esa.int/main/third-party-plugins-2/snaphu/>. Acedido: 2020-11-01.
- [70] C. Yu, Z. Li, N. T. Penna, and P. Crippa. Generic atmospheric correction model for interferometric synthetic aperture radar observations. *Journal of Geophysical Research: Solid Earth*, 123(10): 9202–9222, 2018.
- [71] C. Yu, Z. Li, and N. T. Penna. Interferometric synthetic aperture radar atmospheric correction using a gps-based iterative tropospheric decomposition model. *Remote Sensing of Environment*, 204: 109 – 121, 2018. ISSN 0034-4257. doi: <https://doi.org/10.1016/j.rse.2017.10.038>. URL <http://www.sciencedirect.com/science/article/pii/S0034425717305011>.
- [72] C. Yu, N. T. Penna, and Z. Li. Generation of real-time mode high-resolution water vapor fields from gps observations. *Journal of Geophysical Research: Atmospheres*, 122(3):2008–2025, 2017. doi:

- 10.1002/2016JD025753. URL <https://agupubs.onlinelibrary.wiley.com/doi/abs/10.1002/2016JD025753>.
- [73] GACOS. Gacos, 2017. URL <http://ceg-research.ncl.ac.uk/v2/gacos/>. Acedido: 2020-08-28.
- [74] Zhou Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4):600–612, 2004. doi: 10.1109/TIP.2003.819861.
- [75] ESA. *Acquisition Modes*, 2013. URL <https://sentinel.esa.int/web/sentinel/user-guides/sentinel-1-sar/acquisition-modes>. Acedido: 2020-09-15.
- [76] ESA. *Stripmap*, 2013. URL <https://sentinel.esa.int/web/sentinel/user-guides/sentinel-1-sar/acquisition-modes/stripmap>. Acedido: 2020-09-15.
- [77] F. De Zan and A. M. Guarnieri. Topsar: Terrain observation by progressive scans. *IEEE Transactions on Geoscience and Remote Sensing*, 44(9):2352–2360, 2006.
- [78] ESA. *Interferometric Wide Swath*, 2013. URL <https://sentinel.esa.int/web/sentinel/user-guides/sentinel-1-sar/acquisition-modes/interferometric-wide-swath>. Acedido: 2020-09-15.
- [79] ESA. *Extra Wide Swath*, 2013. URL <https://sentinel.esa.int/web/sentinel/user-guides/sentinel-1-sar/acquisition-modes/extra-wide-swath>. Acedido: 2020-09-15.
- [80] ESA. *Wave*, 2013. URL <https://sentinel.esa.int/web/sentinel/user-guides/sentinel-1-sar/acquisition-modes/wave>. Acedido: 2020-09-15.
- [81] ESA. *Product Types and Processing Levels*, 2013. URL <https://sentinel.esa.int/web/sentinel/user-guides/sentinel-1-sar/product-types-processing-levels>. Acedido: 2020-09-15.
- [82] GNU. Gnu wget 1.20 manual, 2018. URL <https://www.gnu.org/software/wget/manual/>. Acedido: 2020-11-02.
- [83] F. Totir. Envi file reader/writer. Acedido: 2020-08-22, 2010. URL <https://www.mathworks.com/matlabcentral/fileexchange/27172-envi-file-reader-writer>.

Apêndice A

Imagens SAR

A.1 Modos de aquisição de imagens SAR e respectivos níveis

A.1.1 Modos de aquisição

Os modos SM, IW e EW funcionam num ciclo de 25 minutos por cada órbita. Utilizam polarização única, HH ou VV, e polarização dupla, HH + HV e VV + VH. O modo WV, por sua vez, funciona num ciclo de 75 minutos por órbita e apenas suporta polarizações simples em HH ou VV [75].

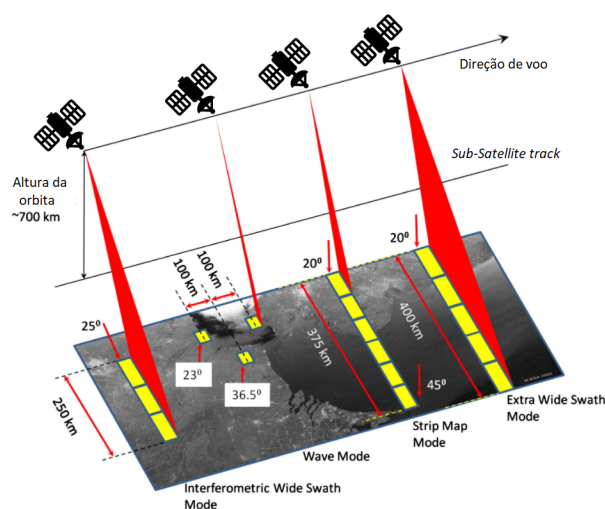


Figura A.1: Modos de aquisição das imagens dos satélites Sentinel 1 [75].

Stripmap Mode (SM)

O modo fornece uma cobertura constante com uma resolução geométrica de 5 m por 5 m e com uma largura de faixa de 80 km e pode operar com uma de seis elevações de feixe pré-definidas, cada uma com um ângulo incidente distinto. No total, essas seis faixas sobrepostas cobrem uma extensão total de 375 km. É um modo que destina-se apenas a casos excepcionais, como apoio a ações de gestão de emergência [76].

Interferometric Wide Swath (IW)

IW é um modo com uma largura de faixa de 250 km com uma resolução moderada de 5 m por 20 m. É o modo mais usado para análises e monitorizações sobre a terra, pois ajuda a localizar um alvo no solo. Este adquire as imagens em três sub-faixas implementando um modo ScanSAR, recentemente formulado, denominado *Terrain Observation with Progressive Scans SAR*, (TOPSAR¹) ESA [78].

Extra Wide Swath Mode (EW)

O modo EW assemelha-se ao modo IW, utiliza técnicas TOPSAR que permitem cobrir uma área mais ampla de até 400 km, usando cinco sub-faixas. Possui também uma resolução média de até 20 m por 40 m de distância do solo. É um modo principalmente utilizado em zonas marítimas, sobre gelo e zonas polares onde é necessário ter uma ampla cobertura e um curto tempo de revisita ESA [79].

Wave Mode (WV)

O modo WV assemelha-se aos modos ERS e Evisat, porém com uma melhor resolução espacial e vinhetas maiores. Este modo adquire os dados em vinhetas de 20 km por 20 km a cada 100 km ao longo da órbita. O resultado do uso deste consiste em várias vinhetas, sendo cada uma, uma imagem separada. É utilizado sobretudo sobre os oceanos ESA [80].

A.1.2 Níveis

Os produtos Sentinel-1 adquiridos através dos modos de aquisição descritos na subsecção A.1.1 são distribuídos em três níveis de processamento, nível-0, nível-1 e nível-2. Como se pode visualizar na figura A.2, os produtos são identificados com base no seu modo de aquisição, tipo de produto e resolução.

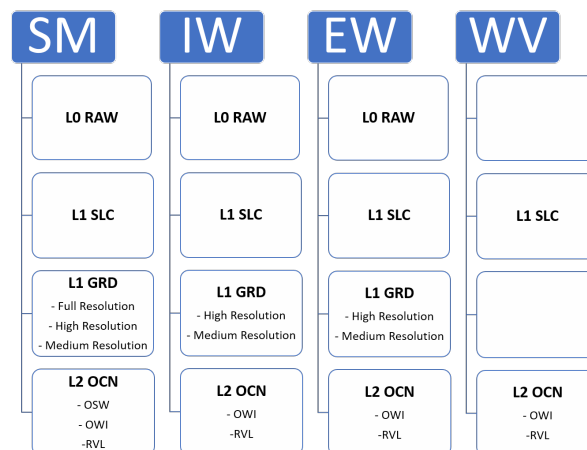


Figura A.2: Níveis dos produtos Sentinel 1 ESA [81].

¹TOPSAR corresponde a uma técnica Scanning SAR (ScanSAR), em que os dados são adquiridos em *bursts* através da alternância cíclica do feixe da antena entre múltiplas subfaixas adjacentes De Zan and Guarnieri [77]

Nível 0

Os produtos do Nível-0 consistem em dados SAR brutos comprimidos e não focados, incluem ruído, calibração interna, pacotes de fonte de eco e informações de órbita e atitude. Encontram-se disponíveis para os modos SM, IW e EW. São os produtos base a partir da qual os produtos dos níveis acima são produzidos. Para os dados poderem ser utilizados, estes têm de ser descomprimidos e processados ESA [81].

Nível 1

O Nível 1 caracteriza-se por fornecer produtos de dois tipos, *Slant-range single-look complex products* (SLC) e *Ground range Detected geo-referenced products* (GRD) , a partir dos produtos de nível 0 após processamento.

Os SLC consistem em dados SAR focados, georreferenciados utilizando dados de órbita e atitude do satélite, e proporcionados em geometria de alcance inclinado em zero-Doppler. Estes produtos consistem em amostras complexas que preservam a informação de fase. Cada pixel da imagem é representado por valores de magnitude I e Q. Esses produtos são feitos para todos os modos de imagem. Nos modos IW e EW cada subfaixa consiste numa série de *bursts*, em que cada *bursts* é processado separadamente como uma imagem SLC única.

Os produtos GRD usam dados focados, que são projectados para o alcance do solo, usando um modelo de elipsóide da Terra corrigido com a altura do terreno especificada que mantém a direcção do caminho do satélite original e inclui informações de georreferenciamento completas. O produto final tem resolução espacial quase quadrada com redução de speckle, degradando a resolução geométrica. Estes produtos são proporcionados em três resoluções: *Full Resolution*, *High Resolution* e *Medium Resolution* ESA [81].

Nível 2

Os produtos de Nível-2 consistem em produtos geofísicos derivados dos produtos de Nível-1. Os produtos Nível-2 Ocean (OCN) podem ser OWI (*Ocean Wind Field*), OSW (*Ocean Swell spectra*) e RVL (*Surface Radial Velocity*). A disponibilidade destas três componentes é dependente do modo de aquisição ESA [81].

A.2 Séries Temporais imagens SAR

Produtos	Identificador
S1A_IW_SLC__1SDV_20200626T183528_20200626T183556_033192_03D858_AC12	3f515721-27d9-4078-806b-3aca58ac44bf
S1A_IW_SLC__1SDV_20200614T183528_20200614T183556_033017_03D307_6F8A	e841be0f-d281-42b4-9214-68c7f6fd1b91
S1A_IW_SLC__1SDV_20200602T183527_20200602T183555_032842_03CDD5_D43C	b7a706f6-8ae5-4bea-91b1-b68187843f65
S1A_IW_SLC__1SDV_20200521T183526_20200521T183554_032667_03C894_56C6	05400f9c-2058-4844-8e35-9dbccad07f25e
S1A_IW_SLC__1SDV_20200509T183526_20200509T183553_032492_03C34B_1CAA	d254f22e-9ae8-4000-90d5-c9dc42040999
S1A_IW_SLC__1SDV_20200427T183525_20200427T183553_032317_03BD64_913F	8306861d-fe75-4893-9688-a2165826a4b2
S1A_IW_SLC__1SDV_20200415T183524_20200415T183552_032142_03B73F_EAE7	845b0afb-fb7d-4c6e-b79a-d4fa8813614f
S1A_IW_SLC__1SDV_20200403T183524_20200403T183552_031967_03B115_EF69	9405cfa-4202-47de-b72d-6f8bb41273c3
S1A_IW_SLC__1SDV_20200322T183524_20200322T183552_031792_03AAED_8AC1	7850253a-d017-4cc2-bf61-0186a1a71e19
S1A_IW_SLC__1SDV_20200310T183524_20200310T183551_031617_03A4C8_D446	bfd6ab73-c059-40f5-93a7-453f5cd869930
S1A_IW_SLC__1SDV_20200227T183523_20200227T183551_031442_039EB3_E1B2	bc1dd810-9ed9-45ef-a580-4c144a43aeb2
S1A_IW_SLC__1SDV_20200215T183524_20200215T183552_031267_0398AF_225B	3802b4a8-fd5c-4ba8-a3b6-83b4b1151ffc
S1A_IW_SLC__1SDV_20200203T183524_20200203T183551_031092_0392A5_C64A	72c53ddd-b01e-475b-ae92-5afc7d901834
S1A_IW_SLC__1SDV_20200122T183524_20200122T183551_030917_038C80_3318	c5545716-76c6-4c47-a1ac-c09c4b93ec6a
S1A_IW_SLC__1SDV_20200110T183525_20200110T183553_030742_038658_6539	b0eef1ce-9dba-406e-8618-eb74fcbdcc4
S1A_IW_SLC__1SDV_20191229T183525_20191229T183553_030567_038049_391C	c5afc60c-f0dd-41b0-8386-afc5a8e45bf1
S1A_IW_SLC__1SDV_20191217T183526_20191217T183554_030392_037A46_5DE0	a898f5e1-68fd-4edc-af60-eat1e9080edf9
S1A_IW_SLC__1SDV_20191205T183526_20191205T183554_030217_037438_7C3E	1df635cb-13a5-496a-9062-fc72d04a45b8
S1A_IW_SLC__1SDV_20191123T183527_20191123T183555_030042_036E2A_286C	5d921a61-6481-4b5e-9f3b-c0c82499eedc
S1A_IW_SLC__1SDV_20191030T183527_20191030T183555_029692_0361FD_07B5	f07e79e7-0f7d-40d3-9315-46a699c84529e
S1A_IW_SLC__1SDV_20191018T183527_20191018T183555_029517_035BE4_D59A	05167311-4bfd-4412-a687-9361bc407348
S1A_IW_SLC__1SDV_20191006T183527_20191006T183555_029342_0355DF_225A	49dca39d-e472-4237-9a06-e3b8841c5ed9
S1A_IW_SLC__1SDV_20190924T183527_20190924T183555_029167_034FD4_A7A5	c27c41ca-8e1a-4567-8fe9-f3f42cf3304
S1A_IW_SLC__1SDV_20190912T183526_20190912T183554_028992_0349DB_7820	d1b0c4fba-137b-4d70-a65a-2dce443a2ab0
S1A_IW_SLC__1SDV_20190831T183526_20190831T183554_028817_0343C8_3D9E	9cc41037-93eb-42e8-aeca-eb8edd8467dd
S1A_IW_SLC__1SDV_20190819T183525_20190819T183553_028642_033DAA_94DC	88d84461-74ea-415d-88a3-6440168724fe
S1A_IW_SLC__1SDV_20190807T183524_20190807T183552_028467_03379C_53DB	37dd1517-38f1-498c-b909-9f099e5b267e2
S1A_IW_SLC__1SDV_20190726T183524_20190726T183552_028292_033242_7B7C	32289659-7f4d-435c-9e1a-ce5736289207

Tabela A.1 : Série Temporal de imagens SAR utilizados para a geração de interferogramas.

Apêndice B

Códigos e *Scripts*

B.1 *Download* imagens SAR

B.1.1 *dhusget script*

Para a utilização do *script*, é necessário ter um ambiente *Linux* com *wget*¹ instalado, visto ser um *script shell* do *Unix* (.sh).

Para executar o processo a partir da linha de comandos é necessário a introdução do seguinte comando: `dhusget.sh [OPÇÕES LOGIN]... [OPÇÕES DE PESQUISA]... [OPÇÕES RESULTADO PESQUISA]... [OPÇÕES DOWNLOAD]...`

Exemplo: `dhusget.sh -d https://scihub.copernicus.eu/dhus -l 100 -P 1 -C product.list.csv -u username -p password -o product -O sentineldata -R failed.txt -F 'query'`

Para mais informações em relação ao *script* e as opções de pesquisa, consultar o site da ESA ESA [63].

B.1.2 *download.py*

Para efectuar a busca e o *download* de imagens SAR Sentinels, deve-se preencher o ficheiro de configuração com as características desejadas e, posteriormente, executar na linha de comandos o seguinte: `python download.py download.conf`. Os produtos SLC são guardadas no directório indicado no ficheiro de configuração.

```
1 ### Python script to download Sentinels products
2 # Author Steven Edward Coelho dos Santos
3 # Date: 27/11/2020
4
5 # connect to the API
6 from sentinelsat import SentinelAPI
```

¹Wget é uma ferramenta criada pelo GNU Project, que permite o *download* de arquivos da Internet a partir da linha de comandos GNU [82]

```

7
8 import sys
9 import datetime
10 from itertools import islice
11 import os
12 inputfile = sys.argv[1]
13
14 now = datetime.datetime.now()
15 bar = '\n#####\n'
16
17 # Getting configuration variables from inputfile
18 try:
19     in_file = open(inputfile, 'r')
20     ...
21     #Missing part of code
22     ...
23 finally:
24     in_file.close()
25
26 parameters=["AREA","SDATE","IDATE","PLATFORMNAME","FILENAME","PRODUCTTYPE","
27             SENSOROPERATIONALMODE","ORBITNUMBER","RELATIVEORBITNUMBER",
28             "POLARISATIONMODE","CLOUDCOVERPERCENTAGE","ORBITDIRECTION","SWATHIDENTIFIER"]
29
30 # AREA OF INTEREST
31 if LONMIN != "" and LONMAX != "" and LATMIN != "" and LATMAX != "":
32     AREA='POLYGON((' + LONMIN + ' ' + LATMAX + ', ' + LONMAX + ' ' + LATMAX + ', ' + LONMAX + ' ' + LATMIN + ', ' +
33     LONMIN + ' ' + LATMIN + ', ' + LONMIN + ' ' + LATMAX + '))'
34     print("AREA: " + AREA)
35 else:
36     AREA=""
37
38 # SENSING DATE
39 SDATE=""
40 if SSTART != "":
41     if SEND != "":
42         SDATE='beginposition:[' + SSTART + ' TO ' + SEND + '] AND endposition:[' + SSTART
43         + ' TO ' + SEND + ']'
44     else:
45         SDATE='beginposition:[' + SSTART + ' TO NOW] AND endposition:[' + SSTART + ' TO
46         NOW]'
47 else:
48     if SEND != "":
49         SDATE='beginposition:[* TO ' + SEND + '] AND endposition:[* TO ' + SEND + ']'
50 print("Sensing date: " + SDATE)
51
52 # INGESTION DATE
53 IDATE=""
54 if ISTART != "":
55     if IEND != "":

```

```

52     IDATE='ingestiondate:[' + ISTART + ' TO ' + IEND + ']'
53     else:
54         IDATE='ingestiondate:['+ ISTART + ' TO NOW]'
55 else:
56     if IEND != "":
57         IDATE='ingestiondate:[* TO ' + IEND + ']'
58 print("Ingestion date: " + IDATE)
59
60 # PROGRESS
61 if SHOWPROGRESS == '0':
62     SHOWPROGRESS = "false"
63 else:
64     SHOWPROGRESS = "true"
65
66 #####
67 ## Products download ##
68 #####
69
70 api = SentinelAPI(USERNAME, PASSWORD, 'https://scihub.copernicus.eu/dhus', SHOWPROGRESS)
71
72 logfolder = LOGFOLDER
73 if not os.path.exists(logfolder):
74     os.makedirs(logfolder)
75
76 outlog=logfolder+'/download_stdout.log'
77 out_file = open(outlog, 'a')
78
79 #----header-----#
80 print(bar)
81 out_file.write(bar)
82 message='## Products download started: ' + str(now.year) + '/' + str(now.month) + '/' +
83         str(now.day) + ' ' + str(now.hour) + ':' + str(now.minute) + ':' + str(now.second)
84 print(message)
85 out_file.write(message)
86 print(bar)
87 out_file.write(bar)
88 #----/header-----#
89
90 if COMPLETEQUERY != "":
91     ...
92     #Missing part of code
93     ...
94 else:
95     checklist=[]
96     for parameter in parameters:
97         if eval(parameter)!="":
98             checklist.append(parameter)
99     print("checklist: " + str(checklist))
100     keywords=""

```

```

100     if checklist[0] == "AREA":
101         ...
102         #Missing part of code
103         ...
104     else:
105         ...
106         #Missing part of code
107         ...
108     if len(checklist) > 1:
109         for parameter in islice(checklist, 1, len(checklist) - 1):
110             ...
111             #Missing part of code
112             ...
113
114     keywords = keywords + ')'
115     print("\nQuery: " + keywords)
116     products = api.query(raw=str(keywords))
117     numprod = api.count(raw=str(keywords))
118     print("PRODUCTS FOUND: " + str(numprod))
119     out_file.write("\nQUERY: " + str(keywords) + '\n')
120     out_file.write("PRODUCTS FOUND: " + str(numprod) + '\n')
121
122 # download all results from the search
123 success,b,failed = api.download_all(products,DOWNLOADFOLDER)
124
125 #raw_input("Press Enter to continue...")
126 out_file.write("\nDOWNLOADED:\n")
127 for prodid in success:
128     print("A: " + str(prodid) + '\n')
129     out_file.write(prodid + '\n')
130
131 out_file.write("\nFAILED:\n")
132 for prodid in failed:
133     print("C: " + str(prodid) + '\n')
134     out_file.write(prodid + '\n')
135
136 print(bar)
137 out_file.write(bar)
138 out_file.close()

```

B.2 SNAP Graph Builder

O Graph Builder, no SNAP-Desktop, permite ao utilizador construir os gráficos a partir de uma lista de operadores já existente conectando os nós dos operadores uns aos outros. Após a construção dos gráficos, para que fosse possível repetir o processo com os demais pares de produtos SLC foi necessário efectuar algumas alterações nos códigos xml. Deste modo, foi necessário abrir os ficheiro XML num editor de texto, substituir alguns valores nos gráfico por variáveis que podem ser importadas de um

arquivo de texto ou substituídas através de *scripts*. Na figura B.1, pode-se observar uma das alterações efectuadas, neste caso, no operador *Read* do ficheiro *coreg_ifg_ml_ft_subset.xml* para permitir a leitura do produto SLC *master*:

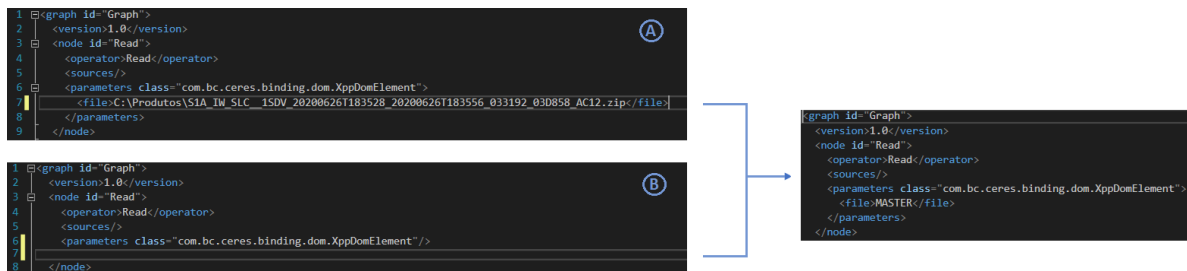


Figura B.1: Alteração no operador *Read* do ficheiro *coreg_ifg_ml_ft_subset.xml*.

Caso a construção do gráfico tenha sido feita com produtos abertos no SNAP, deve-se substituir os caminhos com as *tags* `<file>` pela variável *MASTER* (figura B.1 A). Caso contrário, adiciona-se após a *tag* `<parameters class = "com.bc.ceres.binding.dom.XppDomElement">` o excerto de código: `<file>MASTER</file></parameters>` (figura B.1 B).

B.3 Processamento interferométrico

Para efectuar o processo interferométrico, executa-se, apenas, na linha de comandos `python <script name> <configuration file>` a sequência abaixo indicada:

1. `python slaves_prep.py project.conf`
2. `python splitting.py project.conf`
3. `python ifg_formation.py project.conf`
4. `python unw_tc.py project.conf`
5. `python ifg_correction_prep.py project.conf`
6. `python ztd_correction_prep.py project.conf`
7. `python correction_prep.py project.conf`

Caso for pretendido a exportação para o STAMPS, só são precisos realizar as 3 primeiras sequências acima e por último `python stamps_export.py project.conf`.

Após a execução dos comandos acima, os interferogramas desenrolados ficam disponíveis no sub-directório `ifg_unw_tc` do directório de trabalho, bem como os ficheiros necessários para a próxima etapa do processamento, no sub directório `ifg_matlab` do mesmo directório.

B.3.1 *slaves_prep.py*

```
1 ### Python script to prepare the slaves to use on InSAR process
2 # Author Steven Edward Coelho dos Santos
3 # Date: 08/11/2020
4
5 import os
6 import shutil
7 import sys
8 import datetime
9
10 now = datetime.datetime.now()
11 inputFile = sys.argv[1]
12
13 # Configuration variables from inputfile
14 try:
15     config_file = open(inputFile, 'r')
16     ...
17     #Missing part of code
18     ...
19 finally:
20     config_file.close()
```

```

21
22 separator = '\n#####\n'
23
24 #####
25 ## Slaves sortering in folders ##
26 #####
27
28 directory=PROJECT+'/slaves'
29
30 #----header-----#
31 print(separator)
32 message='## Sort Slaves in folders started: ' + str(now.year) + '/' + str(now.month) + '/'
    ' + str(now.day) + ' ' + str(now.hour) + ':' + str(now.minute) + ':' + str(now.
        second)
33 print(message)
34 print(separator)
35 #----/header-----#
36
37 for filename in os.listdir(directory):
38     if filename.endswith(".zip") :
39         ...
40         #Missing part of code
41         ...
42
43         # Moving files
44         source = os.path.join(directory, filename)
45         destination = os.path.join(subdirectory, tail)
46         print('Moving '+ source + ' to ' + destination + '\n')
47         shutil.move(source,destination)
48     else:
49         continue
50
51 print(separator)

```

B.3.2 *splitting.py*

```

1  ### Python script to splitting the SAR products
2  # Author Steven Edward Coelho dos Santos
3  # Date: 08/11/2020
4
5  import os
6  import sys
7  import glob
8  import subprocess
9  import time
10 import datetime
11
12 inputFile = sys.argv[1]
13 now = datetime.datetime.now()
14 separator = '\n#####\n'

```

```

15
16 # Getting configuration variables from config file
17 try:
18     config_file = open(inputFile, 'r')
19     ...
20     #Missing part of code
21     ...
22
23 finally:
24     config_file.close()
25
26 #####
27 ## TOPSAR Splitting (Assembling) and Apply Orbit ##
28 #####
29
30 masterFolder = PROJECT+'/master'
31 slaveFolder = PROJECT+'/slaves'
32 m_splitFolder = PROJECT+'/m_split'
33 s_splitFolder = PROJECT+'/s_split'
34 logFolder = PROJECT+'/logs'
35
36 if not os.path.exists(m_splitFolder):
37     os.makedirs(m_splitFolder)
38 if not os.path.exists(s_splitFolder):
39     os.makedirs(s_splitFolder)
40 if not os.path.exists(logFolder):
41     os.makedirs(logFolder)
42
43 if CONTINUEONFAIL == '1':
44     CONTINUEONFAIL = "true"
45 else:
46     CONTINUEONFAIL = "false"
47     print("NOTE: The process will stop if new orbit file is not found otherwise change the
48         config file\n")
49
50 auxGraph = GRAPH+'/splitgraph_aux.xml'
51
52 # Log file
53 logName = logFolder+'/split_applyOrbit.log'
54 LogFile = open(logName, 'a')
55
56 #-----header-----#
57 print(separator)
58 LogFile.write(separator)
59 message='## TOPSAR Splitting and Apply Orbit: ' + str(now.year) + '/' + str(now.month) +
60     '/' + str(now.day) + ' ' + str(now.hour) + ':' + str(now.minute) + ':' + str(now.
    second)
61 print(message)
62 LogFile.write(message)

```

```

61 print(separator)
62 LogFile.write(separator)
63 #----/header-----#
64
65 #####
66 ## Split master ##
67 #####
68
69 message = "Starting master split:\n"
70 print(message)
71 LogFile.write('\n' + message + '\n')
72 master_file = MASTER
73 message = "MASTER: " + os.path.join(masterFolder, master_file) + '\n'
74 print("MASTER: " + os.path.join(masterFolder, master_file))
75 LogFile.write(message)
76 head, tail = os.path.split(master_file)
77
78 mastersplitfolder = m_splitFolder + '/' + tail[17:25]
79 if not os.path.exists(mastersplitfolder):
80     os.makedirs(mastersplitfolder)
81
82 outputname = tail[17:25] + '_' + IW + '.dim'
83 graphxml = GRAPH + '/split_applyorbit.xml'
84
85 # Read in the file
86 ...
87 #Missing part of code
88 ...
89
90 # Write the file out again
91 with open(auxGraph, 'w') as file:
92     file.write(filedata)
93
94 args = [ GPT, auxGraph, '-c', CACHE, '-q', CPU]
95 print(args)
96 LogFile.write(str(args)+'\n')
97
98 # launching the process
99 ...
100 #Missing part of code
101 ...
102
103 if process.returncode != 0 :
104     message = 'Error splitting master ' + os.path.join(masterFolder, master_file) + '\n'
105     print('Error splitting master ' + os.path.join(masterFolder, master_file))
106 else:
107     message = 'Split master ' + os.path.join(masterFolder, master_file) + ' successfully
108     completed.\n'
109     print('Split master ' + os.path.join(masterFolder, master_file) + ' successfully

```

```

    completed.')
```

109

```

110 LogFile.write(message)
111 print("-----\n")
112 LogFile.write("\n-----\n")
113
114 #####
115 ## Split slave(s) ##
116 #####
117
118 message = "Starting slaves split:\n"
119 print(message)
120 LogFile.write('\n' + message+ '\n')
121
122 i=0
123 for slaveFolderData in sorted(os.listdir(slaveFolder)):
124     i=i+1
125     print('['+str(i)+'] Folder: '+ slaveFolderData)
126     LogFile.write('['+str(i)+'] Folder: '+ slaveFolderData+'\n')
127     print(os.path.join(slaveFolder, slaveFolderData))
128     LogFile.write(str(os.path.join(slaveFolder, slaveFolderData))+'\n')
129     files = glob.glob(os.path.join(slaveFolder, slaveFolderData) + '/*.zip')
130     print(files)
131     LogFile.write(str(files)+'\n')
132     head, tail = os.path.split(os.path.join(str(files)))
133
134     splitslavefolder = s_splitFolder+'/' +tail [17:25]
135     if not os.path.exists(splitslavefolder):
136         os.makedirs(splitslavefolder)
137
138     outputname=tail [17:25]+'_'+IW+'.dim'
139
140     if len(files) == 1 :
141         graphxml=GRAPH+' /split_applyorbit.xml'
142         # Read in the file
143         ...
144         #Missing part of code
145         ...
146
147         # Replace the target string
148         ...
149         #Missing part of code
150         ...
151
152     if len(files) > 1 :
153         ...
154         #Missing part of code
155         ...
156
```

```

157     args = [ GPT, auxGraph, '-c', CACHE, '-q', CPU]
158     print(args)
159     LogFile.write(str(args)+'\n')
160     # launching the process
161     ...
162     #Missing part of code
163     ...
164
165     if process.returncode != 0 :
166         message='Error splitting slave '+str(files)
167     else:
168         message='Split slave '+str(files)+' successfully completed.\n'
169     print(message)
170     LogFile.write(message)
171     print("-----\n")
172     LogFile.write("\n-----\n")
173
174 print(separator)
175 os.remove(auxGraph)
176 LogFile.write(separator)
177 LogFile.close()

```

B.3.3 ifg_formation.py

```

1  ### Python script for interferogram formation
2  # Author Steven Edward Coelho dos Santos
3  # Date: 08/11/2020
4
5  import os
6  import sys
7  import glob
8  import subprocess
9  import time
10 import datetime
11
12 inputFile = sys.argv[1]
13 now = datetime.datetime.now()
14 separator = '\n#####\n'
15
16 # Configuration variables from config file
17 try:
18     config_file = open(inputFile, 'r')
19
20     ...
21     #Missing part of code
22     ...
23
24 finally:
25     config_file.close()
26

```

```

27 if LONMIN != "" and LATMIN != "" and LONMAX != "" and LATMAX != "":
28     polygon='POLYGON (('+LONMIN+' '+LATMAX+', '+LONMAX+' '+LATMAX+', '+LONMAX+' '+LATMIN+',
        '+LONMIN+' '+LATMIN+', '+LONMIN+' '+LATMAX+'))'
29 else:
30     polygon = ""
31 #####
32 ## TOPSAR Coregistration and Interferogram formation ##
33 #####
34
35 masterFolder = PROJECT + '/master'
36 slaveSplittedFolder = PROJECT + '/s_split'
37 masterSplittedFolder = PROJECT + '/m_split'
38 coregFolder = PROJECT + '/coreg'
39 ifgFolder = PROJECT + '/ifg'
40 ifgExportFolder = PROJECT + '/ifg_stamps'
41 logFolder = PROJECT + '/logs'
42
43 if not os.path.exists(ifgFolder):
44     os.makedirs(ifgFolder)
45 if not os.path.exists(logFolder):
46     os.makedirs(logFolder)
47
48 # Log file
49 logName = logFolder+'ifg_formation.log'
50 logFile = open(logName, 'a')
51
52 # graph choice
53 if STAMPSEXPOR == 1:
54     if polygon!="":
55         graphxml = GRAPH + '/coreg_ifg_mlflt_subset_export.xml'
56     else:
57         graphxml = GRAPH + '/coreg_ifg_mlflt_export.xml'
58     if not os.path.exists(ifgExportFolder):
59         os.makedirs(ifgExportFolder)
60     if not os.path.exists(coregFolder):
61         os.makedirs(coregFolder)
62 else:
63     STAMPSEXPOR = 0
64     if polygon!="":
65         graphxml = GRAPH + '/coreg_ifg_mlflt_subset.xml'
66     else:
67         graphxml = GRAPH + '/coreg_ifg_mlflt.xml'
68 print("USING GRAPH: " + graphxml)
69 auxGraph = GRAPH + '/coreg_ifg_aux.xml'
70
71 #-----header-----#
72 print(separator)
73 logFile.write(separator)
74 message='## Coregistration and Interferogram computation started: ' + str(now.year) + '/'

```



```

    + str(now.month) + '/' + str(now.day) + ' ' + str(now.hour) + ':' + str(now.
minute) + ':' + str(now.second)
75 print(message)
76 logFile.write(message)
77 print(separator)
78 logFile.write(separator)
79 #----/header-----#
80
81 head, m_tail = os.path.split(os.path.join(masterFolder, MASTER))
82
83 #Master's date
84 for master_file in glob.iglob(masterSplittedFolder + '/*/*'+IW+'.dim'):
85     head, tailm = os.path.split(os.path.join(masterSplittedFolder, master_file))
86     ...
87     #Missing part of code
88     ...
89
90 print("MASTER FILE: " + master_file + '\n')
91 logFile.write("\nMASTER FILE: " + master_file + '\n\n')
92
93 #ifg formation
94 i=0
95 for dimfile in glob.iglob(slaveSplittedFolder + '/*/*'+IW+'.dim'):
96     print("SLAVE FILE: " + dimfile)
97     i=i+1
98     head, tail = os.path.split(os.path.join(slaveSplittedFolder, dimfile))
99     message='['+str(i)+'] Processing slave file :'+tail+'\n'
100    print(message)
101    logFile.write(message)
102
103    ...
104    #Missing part of code
105    ...
106
107    if STAMPSEXPORT == 1:
108        ...
109        #Missing part of code
110        ...
111
112    # Write the file out again
113    with open(auxGraph, 'w') as file:
114        file.write(filedata)
115
116    args = [ GPT, auxGraph, '-c', CACHE, '-q', CPU]
117    # Launch the processing
118    ...
119    #Missing part of code
120    ...
121

```

```

122     if process.returncode != 0 :
123         message='Error computing with coregistration and interferogram generation of
splitted slave '+str(dimfile)
124     else:
125         message='Coregistration and Interferogram computation for data '+str(tail)+'
successfully completed.\n'
126
127     print(message)
128     logFile.write(message)
129     print("_-----\n")
130     logFile.write("\n-----\n")
131
132 print(separator)
133 logFile.write(separator)
134 os.remove(auxGraph)
135 logFile.close()

```

B.3.4 *unw.tc.py*

```

1  ### Python script for unwrapping and terrain correction
2  # Author Steven Edward Coelho dos Santos
3  # Date: 07/11/2020
4
5  import os
6  import shutil
7  import sys
8  import datetime
9  import subprocess
10 import time
11
12 now = datetime.datetime.now()
13 inputFile = sys.argv[1]
14
15 separator = '\n#####\n'
16
17 # Getting configuration variables from config file
18 try:
19     config_file = open(inputFile, 'r')
20     ...
21     #Missing part of code
22     ...
23 finally:
24     config_file.close()
25
26 ifgFolder = PROJECT + '/ifg'
27 exportFolder = PROJECT + '/snaphu_export'
28
29 logFolder = PROJECT+'logs'
30 if not os.path.exists(logFolder):
31     os.makedirs(logFolder)

```

```

32 if not os.path.exists(exportFolder):
33     os.makedirs(exportFolder)
34
35 #####
36 ## Snaphu export ##
37 #####
38
39 graphxml = GRAPH + '/snaphuexport.xml'
40 print("USING GRAPH: " + graphxml)
41 auxGraph = GRAPH+'/'snaphuexport_aux.xml'
42
43 # Log file
44 logName = logFolder + '/snaphu_export.log'
45 LogFile = open(logName, 'a')
46
47 #----header-----#
48 print(separator)
49 LogFile.write(separator)
50 message='## Export computation started: ' + str(now.year) + '/' + str(now.month) + '/' +
        str(now.day) + ' ' + str(now.hour) + ':' + str(now.minute) + ':' + str(now.second
        )
51 print(message)
52 LogFile.write(message)
53 print(separator)
54 LogFile.write(separator)
55 #---/header-----#
56
57 i=0
58 for dimfile in os.listdir(ifyFolder):
59     if dimfile.endswith(".dim"):
60         print("FILE: " + dimfile)
61         i=i+1
62         head, tail = os.path.split(os.path.join(ifyFolder, dimfile))
63         message='['+str(i)+'] Processing file :'+tail+'\n'
64         print(message)
65         LogFile.write(message)
66
67         with open(graphxml, 'r') as file :
68             filedata = file.read()
69
70         # Replace the target string
71         ...
72         #Missing part of code
73         ...
74
75         # Write the file out again
76         with open(auxGraph, 'w') as file:
77             file.write(filedata)
78

```

```

79     args = [ GPT, auxGraph, '-c', CACHE, '-q', CPU]
80     # Launch the processing
81     ...
82     #Missing part of code
83     ...
84
85     if process.returncode != 0 :
86         message='Error computing with export '+str(dimfile)
87     else:
88         message='Export for data '+str(tail)+' successfully completed.\n'
89
90     print(message)
91     LogFile.write(message)
92     print("-----\n")
93     LogFile.write("\n-----\n")
94
95 print(separator)
96 LogFile.write(separator)
97 os.remove(auxGraph)
98 LogFile.close()
99
100
101 #####
102 ### Unwrapping and Terrain Correction ###
103 #####
104
105 outputFolder = PROJECT + '/ifg_unw_tc'
106 if not os.path.exists(outputFolder):
107     os.makedirs(outputFolder)
108
109 masterfolder = PROJECT + '/master'
110 snaphuFolder = os.path.join(SNAPHUFOLDER, 'bin')
111 snaphuEXE = os.path.join(snaphuFolder, 'snaphu')
112
113 graphxml = GRAPH + '/snaphuimport_tc.xml'
114 print("USING GRAPH: " + graphxml)
115 auxGraph = GRAPH + '/snaphuimport_tc_aux.xml'
116
117 # Log file
118 logName = logFolder + '/unwrapping_tc.log'
119 logFile = open(logName, 'a')
120
121 #----header-----#
122 print(separator)
123 logFile.write(separator)
124 message='## Unwrapping started: ' + str(now.year) + '/' + str(now.month) + '/' + str(now
        .day) + ' ' + str(now.hour) + ':' + str(now.minute) + ':' + str(now.second)
125 print(message)
126 logFile.write(message)

```

```

127 print(separator)
128 logFile.write(separator)
129 #----/header-----#
130
131 k=0
132 for filename in os.listdir(exportFolder):
133     k=k+1
134     message = "\nFOLDER: " + os.path.join(exportFolder, filename) + '\n'
135     print(message)
136     logFile.write(message)
137
138     # Copying all the files needed on the process
139     for fileInFolder in os.listdir(os.path.join(exportFolder, filename)):
140         message = "FILE: " + os.path.join(filename, fileInFolder)
141         print(message)
142         logFile.write('\n' + message + '\n')
143
144         if fileInFolder.endswith(".hdr") or fileInFolder.endswith(".img") or fileInFolder.
145         endswith(".conf"):
146             ...
147             #Missing part of code
148             ...
149
150             if fileInFolder.endswith(".conf"):
151                 fp = open(os.path.join(exportFolder, filename, fileInFolder))
152                 for i, line in enumerate(fp):
153                     if i == 6:
154                         command = line[30:78]
155                 fp.close()
156
157                 print("COMMAND: " + command)
158                 logFile.write("COMMAND: " + command + '\n')
159                 args = snaphuEXE + ' -f ' + os.path.join(snaphuFolder, 'snaphu.conf') + ' ' + os.path
160                 .join(snaphuFolder, command)
161
162                 # Launch the processing
163                 ...
164                 #Missing part of code
165                 ...
166
167                 if process.returncode != 0 :
168                     message='Error computing unwrapping '+str(filename)
169                 else:
170                     message='Unwrapping computation for data '+str(filename)+' successfully completed
171                     .\n'
172                 print(message)
173                 logFile.write(message)
174
175     # ----- Import and TC correction -----

```

```

173     for snaphufile in os.listdir(snaphuFolder):
174         if snaphufile.startswith("UnwPhase_") and snaphufile.endswith(".hdr"):
175             break
176
177     message = "Snaphu import and terrain correction started:\n"
178     print(message)
179     logFile.write('\n' + message)
180     message = "IFG FILE: " + os.path.join(ifgFolder, filename + '.dim')
181     print(message)
182     logFile.write('\n' + message)
183     message = "UNW FASE FILE: " + os.path.join(snaphuFolder, snaphufile) + '\n'
184     print(message)
185     logFile.write('\n' + message)
186
187     with open(graphxml, 'r') as file :
188         filedata = file.read()
189         # Replace the target string
190         ...
191         #Missing part of code
192         ...
193
194         # Write the file out again
195         with open(auxGraph, 'w') as file:
196             file.write(filedata)
197
198         args = [ GPT, auxGraph, '-c', CACHE, '-q', CPU]
199         # Launch the processing
200         ...
201         #Missing part of code
202         ...
203
204         if process.returncode != 0 :
205             message='['+str(k)+'] Error computing with import and terrain correction '+str(
filename)
206         else:
207             message='['+str(k)+'] Import and terrain correction computation for data '+str(
filename)+' successfully completed.\n'
208
209         print(message)
210         logFile.write(message)
211         # ----- /Import and TC correction -----
212
213         # Removing previously copied files
214         ...
215         #Missing part of code
216         ...
217
218         print(" _-----\n")
219         logFile.write("\n _-----\n")

```

```

220
221 print(separator)
222 logFile.write(separator)
223 os.remove(auxGraph)
224 logFile.close()

```

B.3.5 ifg_correction_prep.py

```

1  ### Python script to prepare the ifgs for the matlab's atmosphere correction processing
2  # Author Steven Edward Coelho dos Santos
3  # Date: 08/11/2020
4
5  import os
6  import shutil
7  import sys
8  import datetime
9
10 now = datetime.datetime.now()
11 inputFile = sys.argv[1]
12
13 # Configuration variables from config file
14 try:
15     config_file = open(inputFile, 'r')
16     ...
17     #Missing part of code
18     ...
19 finally:
20     config_file.close()
21
22 separator = '\n#####\n'
23
24 #####
25 ## Copying the unwrapped ifgs for atm correction computation ##
26 #####
27
28 directory = PROJECT + '/ifg_unw_tc'
29 matlabdirectory = PROJECT + '/ifg_matlab'
30 if not os.path.exists(matlabdirectory):
31     os.makedirs(matlabdirectory)
32
33 #----header-----#
34 print(separator)
35 message='## Copying the unwrapped ifgs for atm correction computation started: ' + str(
36     now.year) + '/' + str(now.month) + '/' + str(now.day) + ' ' + str(now.hour) + ':' +
37     str(now.minute) + ':' + str(now.second)
38 print(message)
39 print(separator)
40 #----/header-----#
41
42 for filename in os.listdir(directory):

```

```

41     if filename.endswith(".data"):
42         print("IFG: " + os.path.join(directory, filename))
43         head, tail = os.path.split(os.path.join(directory, filename))
44
45         destinationdirectory = matlabdirectory + '/' + tail[9:17]
46         if not os.path.exists(destinationdirectory):
47             os.makedirs(destinationdirectory)
48
49         if os.path.exists(os.path.join(destinationdirectory, tail)):
50             shutil.rmtree(os.path.join(destinationdirectory, tail))
51
52         for file in os.listdir(os.path.join(directory, filename)):
53             ...
54             #Missing part of code
55             ...
56     else:
57         continue
58
59 print(separator)

```

B.3.6 *ztd_correction_prep.py*

```

1  ### Python script to prepare the ztds for the matlab's atmosphere correction processing
2  # Author Steven Edward Coelho dos Santos
3  # Date: 07/11/2020
4
5  import os
6  import shutil
7  import sys
8  import datetime
9
10 now = datetime.datetime.now()
11 inputFile = sys.argv[1]
12
13 # Configuration variables from config file
14
15 try:
16     config_file = open(inputFile, 'r')
17     ...
18     #Missing part of code
19     ...
20 finally:
21     config_file.close()
22
23 separator = '\n#####\n'
24
25 #####
26 ### Copying the ZTDs for atm correction computation ##
27 #####
28

```



```

29 ifgFolder = PROJECT + '/ifg_matlab'
30 masterFolder = PROJECT + '/master'
31
32 #----header-----#
33 print(separator)
34 message='## Copying the ZTDs for atm correction computation started: ' + str(now.year) +
    '/' + str(now.month) + '/' + str(now.day) + ' ' + str(now.hour) + ':' + str(now.
    minute) + ':' + str(now.second)
35 print(message)
36 print(separator)
37 #----/header-----#
38
39 head, tail = os.path.split(os.path.join(masterFolder, MASTER))
40 mastertail = tail[17:25]
41 print("\nMASTER DATE: " + mastertail + '\n')
42
43 masterztd = []
44
45 # Find the master's ztd
46 for filename in os.listdir(ZTDFOLDER):
47     ...
48     #Missing part of code
49     ...
50
51 # Copying the ztds
52 for filename in os.listdir(ZTDFOLDER):
53     if filename.endswith(".ztd") or filename.endswith(".rsc") or filename.endswith(".jpg"
54     ):
55         ...
56         #Missing part of code
57         ...
58     else:
59         continue
60 print(separator)

```

B.3.7 *correction_prep.py*

```

1 ### Python script to prepare the atmospheric correction matlab script
2 # Author Steven Edward Coelho dos Santos
3 # Date: 08/11/2020
4
5 import os
6 import sys
7 from shutil import copyfile
8 import datetime
9
10 now = datetime.datetime.now()
11 inputFile = sys.argv[1]
12

```

```

13 # Configuration variables from config file
14 try:
15     config_file = open(inputFile, 'r')
16     ...
17     #Missing part of code
18     ...
19 finally:
20     config_file.close()
21
22 separator = '\n#####\n'
23
24 #####
25 ## Atmospheric correction pre-processing ##
26 #####
27
28 ifgFolder = PROJECT + '/ifg_matlab'
29 logFolder = PROJECT + '/logs'
30 matlabScript = MATLABFOLDER + '/atm_correction.m'
31 TIFF_images = ifgFolder + '/TIFF_images'
32
33 if not os.path.exists(logFolder):
34     os.makedirs(logFolder)
35 if not os.path.exists(ifgFolder):
36     os.makedirs(ifgFolder)
37 if not os.path.exists(TIFF_images):
38     os.makedirs(TIFF_images)
39
40 # Log file
41 logName = logFolder + '/atm_correction_prep.log'
42 LogFile = open(logName, 'a')
43
44 # Copy of the matlab's atmospheric correction script whit the necessary parameters
45 #for run with an ifg individually if necessary
46 auxScript = MATLABFOLDER + '/atm_correction_aux.m'
47 copyfile(matlabScript, auxScript)
48
49 # Atmospheric correction config file for run with more than one ifg at the same time
50 atm_correction_conf_folder = PROJECT + '/ifg_matlab'
51 atm_correction_conf = atm_correction_conf_folder + '/atm_correction.txt'
52 atm_correction_conf_file = open(atm_correction_conf, 'w')
53
54 atm_correction_conf_file.write(TIFF_images + '\n') #To save the geotiff images
55
56 if LASTCORRECTEDIFGDATE != "":
57     atm_correction_conf_file.write(LASTCORRECTEDIFGDATE + '\n')
58 else:
59     atm_correction_conf_file.write("No reference" + '\n')
60 if LASTCORRECTEDIFGFOLDER != "":
61     atm_correction_conf_file.write(LASTCORRECTEDIFGFOLDER + '\n\n')

```

```

62 else:
63     atm_correction_conf_file.write("No reference" + '\n\n')
64
65 #---header-----#
66 print(separator)
67 LogFile.write(separator)
68 message='## Preparations of matlab scrip started: ' + str(now.year) + '/' + str(now.month
        ) + '/' + str(now.day) + ' ' + str(now.hour) + ':' + str(now.minute) + ':' + str(
        now.second)
69 print(message)
70 LogFile.write(message)
71 print(separator)
72 LogFile.write(separator)
73 #---/header-----#
74
75 #--- Variables
76 ifg_Unw_Phase_hdr = ""
77 ifg_Unw_Phase_img = ""
78 coordinate_matrices = ""
79 ztd_master = ""
80 ztd_slave = ""
81 #--- /Variables
82
83 head, mtail = os.path.split(MASTER)
84
85 processcheckflag = "true"
86
87 for Dir in os.listdir(igfFolder):
88
89     if os.path.isdir(igfFolder + '/' + Dir) and Dir != "TIFF_images" :
90
91         ...
92         #Missing part of code
93         ...
94
95         #--- /Aux Variables
96
97         for subDir in os.listdir(igfFolder + '/' + Dir):
98             if os.path.isdir(igfFolder + '/' + Dir + '/' + subDir) and (subDir == "
atmosphere" or subDir.endswith(".data")):
99                 message = "CURRENT SUBDIR: " + subDir
100                 print(message)
101                 LogFile.write(message + '\n')
102
103                 if subDir == "atmosphere":
104                     ...
105                     #Missing part of code
106                     ...
107

```

```

108         if subDir[0:8] == mtail[17:25]:
109             ...
110             #Missing part of code
111             ...
112
113     # Create a folder for the result's datas of the atmospheric correction
114     resultfolder = ifgFolder + '/' + Dir + '/correction_data'
115     if not os.path.exists(resultfolder):
116         os.makedirs(resultfolder)
117
118     with open(matlabScript, 'r') as file :
119         filedata = file.read()
120
121     if ifg_Unw_Phase_hdr != "" and ifg_Unw_Phase_img != "" and ztd1 != "" and ztd2 !=
122     "" and coordinate_matrices != "":
123         missingfilesflag = "false"
124
125         # Replace the target string
126         ...
127         #Missing part of code
128         ...
129
130         # Write the file out again
131         with open(auxScript, 'w') as file:
132             file.write(filedata)
133
134         file.close()
135
136         if os.path.exists(ifgFolder + '/' + Dir + '/' + 'atm_correction.m'):
137             os.remove(ifgFolder + '/' + Dir + '/' + 'atm_correction.m')
138
139         # Copy the file to the ifg folder to run individually if necessary
140         copyfile(auxScript, ifgFolder + '/' + Dir + '/' + 'atm_correction.m')
141
142         # Create the Atmospheric correction text file
143         ...
144         #Missing part of code
145         ...
146
147     else:
148         missingfilesflag = "true"
149         processcheckflag = "false"
150
151         if ifg_Unw_Phase_hdr == "" or ifg_Unw_Phase_img == "":
152             print("WARNING: Not found UNW_PHASE file(s) in this directory")
153             LogFile.write("WARNING: Not found UNW_PHASE file(s) in this directory\n")
154         if ztd1 == "" or ztd2 == "" or coordinate_matrices == "":
155             print("WARNING: Not found ZTD file(s) in this directory")
156             LogFile.write("WARNING: Not found ZTD file(s) in this directory")

```

```

156
157     print('\n')
158
159 if processcheckflag == "false":
160     message='\n\nError computing the preparation (Missing files).\n'
161     print('Error computing the preparation (Missing files).\n')
162 else:
163     ...
164     #Missing part of code
165     ...
166
167 os.remove(auxScript)
168 LogFile.write(message)
169 print(separator)
170 LogFile.write(separator)
171 LogFile.close()
172 atm_correction_conf_file.close()

```

B.3.8 *stamps_export.py*

```

1  ### Python script to export to STAMPS
2  # Author Steven Edward Coelho dos Santos
3  # Date: 08/11/2020
4
5  import os
6  from pathlib import Path
7  import sys
8  import glob
9  import subprocess
10 import time
11 import datetime
12
13 now = datetime.datetime.now()
14 inputFile = sys.argv[1]
15
16 # Getting configuration variables from config file
17 try:
18     config_file = open(inputFile, 'r')
19     ...
20     #Missing part of code
21     ...
22 finally:
23     config_file.close()
24
25 separator = '\n#####\n'
26
27 #####
28 ## Export to STAMPS ##
29 #####
30

```

```

31 coregFolder=PROJECT+'/coreg'
32 ifgFolder=PROJECT+'/ifg_stamps'
33 head, tail = os.path.split(MASTER)
34 exportFolder=PROJECT+'/INSAR_'+tail[17:25]
35 logFolder=PROJECT+'/logs'
36
37 if not os.path.exists(exportFolder):
38     os.makedirs(exportFolder)
39 if not os.path.exists(logFolder):
40     os.makedirs(logFolder)
41
42 # Log file
43 logName=logFolder+'/stamps_export.log'
44 logFile = open(logName, 'a')
45
46 graphxml=GRAPH+'/stampsexport.xml'
47 auxGraph=GRAPH+'/stampsexport_aux.xml'
48
49 #----header-----#
50 print(separator)
51 logFile.write(separator)
52 message='## StaMPS PSI export started: ' + str(now.year) + '/' + str(now.month) + '/' +
53     str(now.day) + ' ' + str(now.hour) + ':' + str(now.minute) + ':' + str(now.second)
54 print(message)
55 logFile.write(message)
56 print(separator)
57 logFile.write(separator)
58 #----/header-----#
59
60 i=0
61 for dimfile in glob.iglob(coregFolder + '/*'+IW+'.dim'):
62     print("COREG FILE: " + dimfile)
63     head, tail = os.path.split(os.path.join(coregFolder, dimfile))
64     i=i+1
65     message='['+str(i)+'] Exporting pair: master-slave pair '+tail+'\n'
66     ifgdim = Path(ifgFolder+'/'+tail)
67     print(ifgdim)
68     if ifgdim.is_file():
69         print(message)
70         logFile.write('\n' + message)
71         with open(graphxml, 'r') as file :
72             filedata = file.read()
73
74     # Replace the target string
75     ...
76     #Missing part of code
77     ...
78
79 if process.returncode != 0 :

```

```

79     message='Error exporting '+str(tail)+'\n'
80 else:
81     message='Stamps export of '+str(tail)+' successfully completed.\n'
82
83     print(message)
84     logFile.write(message)
85     print(separator)
86     logFile.write(separator)
87
88 logFile.close()
89 os.remove(auxGraph)

```

B.4 Correção atmosférica

De modo poder utilizar os *scripts* presentes nesta secção é necessário para além da *toolbox* fornecida pelo GACOS, efectuar o download de uma outra *toolbox* que permite a leitura e a gravação de dados ENVI e *header files* Totir [83]

B.4.1 main_auto_atm_correction.m

```

1 fileID = fopen('atm_correction.txt');
2 fid = fopen('ifg_comparation.txt','a');
3
4 if fileID ~= -1
5     out_ifgFolder = fgetl(fileID);
6     date1 = fgetl(fileID);
7     last_correction_folder = fgetl(fileID);
8     fgetl(fileID);
9     while ~feof(fileID)
10        date2 = fgetl(fileID);
11        ifg_Unw_Phase_hdr = fgetl(fileID);
12        disp(ifg_Unw_Phase_hdr);
13        ifg_Unw_Phase_img = fgetl(fileID);
14        disp(ifg_Unw_Phase_img);
15        coordinate_matrices = fgetl(fileID);
16        disp(coordinate_matrices);
17        ztd_master = fgetl(fileID);
18        disp(ztd_master);
19        ztd_slave = fgetl(fileID);
20        disp(ztd_slave);
21        correction_folder = fgetl(fileID);
22        disp(correction_folder)
23        operation = fgetl(fileID);
24        disp(operation);
25        auto_atm_correction(ifg_Unw_Phase_hdr, ifg_Unw_Phase_img, coordinate_matrices,
26        ztd_master, ztd_slave, correction_folder, date2, operation, out_ifgFolder);
27        check_diference(last_correction_folder, date1, correction_folder, date2, fid);
28        last_correction_folder = correction_folder;

```

```

28     date1 = date2;
29     fgetl(fileID);
30     disp(newline);
31 end
32 fclose(fid);
33 fclose(fileID);
34 end

```

B.4.2 auto_atm_correction.m

```

1 function auto_atm_correction(ifg_Unw_Phase_hdr, ifg_Unw_Phase_img, coordinate_matrices,
2     ztd_master, ztd_slave, correction_folder, date, operation, out_ifgFolder)
3 addpath('ENVI');
4 addpath('GACOS');
5 %----- READ THE SNAP INTERFEROGRAM -----%
6 %=====
7 %Read the header file.
8 info = envihdrread(ifg_Unw_Phase_hdr);
9 %Read the binary file.
10 data = envidataread(ifg_Unw_Phase_img, info);
11 %Create the coordinate matrices.
12 [Xifg,Yifg] = meshgrid(info.x,info.y);
13 %----- Pre processing GACOS -----%
14 %=====
15 %Read the header of GACOS data and Create the coordinate matrices.
16 [width,len,xfirst,yfirst,xstep,ystep]=read_header(coordinate_matrices);
17
18 ...
19 %Missing part of code
20 ...
21
22 %Read the GACOS grids.
23 fi = fopen(ztd_master,'r'); ztd1 = fread(fi,[ width len],'float');fclose(fi);
24 fi = fopen(ztd_slave,'r'); ztd2 = fread(fi,[ width len],'float');fclose(fi);
25
26 %Interpolate the different grids on the same footprint.
27 f1 = scatteredInterpolant(Xgacos(:),Ygacos(:),ztd1(:),'linear','none');
28 f2 = scatteredInterpolant(Xgacos(:),Ygacos(:),ztd2(:),'linear','none');
29 f3 = scatteredInterpolant(Xifg(:),Yifg(:),data(:),'linear','none');
30
31 %Select the final comun grid and we interpolate the data.
32 ...
33 %Missing part of code
34 ...
35
36
37 %Finally, Save the grid for GACOS.
38 fi = fopen(strcat(correction_folder,'/ztd1_int.tmp'),'w');
39 fwrite(fi,ztd1_int','float'); fclose(fi);

```



```

40 fi = fopen(strcat(correction_folder, '/ztd2_int.tmp'), 'w');
41 fwrite(fi, ztd2_int, 'float'); fclose(fi);
42 fi = fopen(strcat(correction_folder, '/unw_int.tmp'), 'w');
43 fwrite(fi, unw_int, 'float'); fclose(fi);
44
45 %Create the .rsc file. Only one file will be written and it will be copied.
46 fi = fopen(strcat(correction_folder, '/rsc.tmp'), 'w');
47 fprintf(fi, 'WIDTH\t%d\n', size(ztd1_int, 2));
48 fprintf(fi, 'FILE_LENGTH\t%d\n', size(ztd1_int, 1));
49 fprintf(fi, 'XMIN\t%d\n', 0);
50 fprintf(fi, 'XMAX\t%d\n', size(ztd1_int, 2));
51 fprintf(fi, 'YMIN\t%d\n', 0);
52 fprintf(fi, 'YMAX\t%d\n', size(ztd1_int, 1));
53 fprintf(fi, 'X_FIRST\t%f\n', min(Xcom(:)));
54 fprintf(fi, 'Y_FIRST\t%f\n', max(Ycom(:)));
55 n = gradient(Xcom(1, :));
56 fprintf(fi, 'X_STEP\t%f\n', mode(n(:)));
57 n = gradient(Ycom(:, 1));
58 fprintf(fi, 'Y_STEP\t%f\n', mode(n(:)));
59 fprintf(fi, 'X_UNIT\t%s\n', 'degrees');
60 fprintf(fi, 'Y_UNIT\t%s\n', 'degrees');
61 fprintf(fi, 'Z_OFFSET\t%d\n', 0);
62 fprintf(fi, 'Z_SCALE\t%d\n', 1);
63 fprintf(fi, 'PROJECTION\t%s\n', 'LATLON');
64 fprintf(fi, 'DATUM\t%s\n', 'WGS84');
65 fprintf(fi, 'TIME_OF_DAY\t%s\n', 'NONE');
66 fclose(fi);
67
68 ...
69 %Missing part of code
70 ...
71
72 %----- GACOS -----%
73 %=====
74 %Functions to make corrections from GACOS
75 % GACOS MATLAB code: http://ceg-research.ncl.ac.uk/v2/gacos/
76 %plot inline -s 1200,750
77
78 %%mandatory parameters
79 %dir where contains all the data and outputs
80 dir = correction_folder;
81 %interferogram filename
82 fln_phase='unw_int.tmp';
83 %ztd1 filename, Note: Unit in meters
84 fln_ztd1='ztd2_int.tmp';
85 %ztd2 filename, Note: Unit in meters
86 fln_ztd2='ztd1_int.tmp';
87 %elevation angle file
88 fln_elev='NONE';

```

```

89 %if elevation angle file not found, use this value instead, unit in degree
90 elev=28;
91 %interferogram unit,choose from p,m,c for phase, meter, centimeter
92 unit_phase='p';
93 %phase wavelength,S1=0.055165,
94 wavelength=0.055165;
95 %output unit,choose from p,m,c for phase, meter, centimeter
96 unit_out='c';
97
98 %%optional parameters
99 %if remove planar trend, 1 for yes, 0 for no
100 isplanar=1;
101 %if rewrap when plotting, note that the result file will not be wrapped
102 %set 0 for not wrapping
103 %set > 0 to wrap the image to (-iswrap,iswrap), unit as unit_out
104 iswrap=pi;
105 %set reference point, give 0 to use the defaule points with mean phase
106 %give longitude and latitude in degree, not row and col number!
107 ref_lat=38.722; %Lisbon Lat and Lon
108 ref_lon=-9.139;
109 %value for void points, choose from nan or any float numbers
110 void_value=nan;
111
112 %call functions
113 make_correction(dir,fln_phase,fln_ztd1,fln_ztd2,fln_elev,elev,unit_phase,wavelength,
    unit_out,isplanar,iswrap,ref_lat,ref_lon,void_value, date, operation, out_ifgFolder);
114 end

```

B.4.3 check_diference.m

```

1 function check_diference(dir1,date1,dir2,date2,file_log)
2     addpath('GACOS');
3     if dir1 ~= "No reference" && date1 ~= "No reference"
4         dir1=[dir1, '/'];
5         dir2=[dir2, '/'];
6         [wid,len,~,~,~,~] = read_header([dir1,'phase-ztd','.rsc']);
7
8         %read phase and convert to meters
9         fid1 = fopen([dir1,'phase-ztd'],'rb');
10        last_corrected = fread(fid1,[wid, len],'float');
11        fid2 = fopen([dir2,'phase-ztd'],'rb');
12        now_corrected = fread(fid2,[wid, len],'float');
13
14        ...
15        %Missing part of code
16        ...
17
18        fprintf('The ifgs %s and %s are %f%% equals\n',date1, date2, percentMatching);
19        fprintf('The Structural similarity is %f and the Mean-squared error is %f\n',
    ssimval, err);

```

```
20     fprintf(file_log,'The ifgs %s and %s are %f%% equals\n',date1, date2,
percentMatching);
21     fprintf(file_log,'The Structural similarity is %f and the Mean-squared error is %
f\n\n', ssimval, err);
22     fclose(fid1);
23     fclose(fid2);
24     end
25 end
```

