

**Università Politecnica delle Marche**

**Facoltà di Ingegneria**



**Laurea Magistrale in Ingegneria Elettronica**

Studio e sviluppo di un sistema per l'analisi del dissesto idrogeologico basato su immagini satellitari Copernicus.

Study and development of a system for the analysis of hydrogeological instability based on Copernicus satellite images.

*Relatore:*  
Prof. Paola Pierleoni

*Tesi di laurea di:*  
Nicola Bastianelli

*Correlatori:*  
Luisiana Sabbatini

Anno Accademico 2020/2021



# INDICE

INTRODUZIONE .....	3
RIEPILOGO ESECUTIVO .....	5
CAPITOLO 1.....	8
1.1    Il dissesto idrogeologico.....	8
Procedure di allertamento .....	9
1.2    Il monitoraggio idrogeologico nella Regione Marche.....	11
CAPITOLO 2.....	14
2.1    Il programma Copernicus .....	14
2.1.1    Sentinel-1.....	15
Caratteristiche dei satelliti .....	15
Orbita e copertura geografica .....	16
Strumento <i>C-SAR</i> .....	17
Prodotti disponibili [10].....	20
2.1.2    Sentinel-2.....	20
Caratteristiche dei satelliti .....	21
Orbita e copertura geografica .....	21
Strumento MSI.....	22
Prodotti disponibili [13].....	24
2.2    Accesso ai dati Copernicus.....	27
2.2.1    Copernicus Open Access Hub .....	27
Formattazione dei dati .....	28
Visualizzazione dei dati scaricati.....	29
2.3    Copernicus Emergency Management System.....	30
Attivazione del servizio .....	31

CAPITOLO 3.....	32
3.1    Utilizzo delle immagini Sentinel-1 per l’analisi del dissesto idrogeologico ...	32
3.1.1    Binarizzazione dell’immagine mediante metodi adattivi .....	33
3.2    Implementazione del codice per l’analisi delle immagini .....	34
Impostazioni di base .....	34
Startup del codice.....	35
Processamento dell’immagine Sentinel-1 .....	37
Esportazione dei dati.....	44
CAPITOLO 4.....	47
4.1    Applicazione di immagini Sentinel-1 per lo studio del dissesto idrogeologico nella Piana Reatina, Italia .....	47
4.1.1    Analisi dei risultati .....	52
Metodo Otsu e Metodo Minimo a confronto .....	53
Confronto con foto aeree in-situ .....	55
Risultati del Servizio per le Emergenze di Copernicus .....	61
4.2    Applicazione di immagini Sentinel-1 per lo studio del dissesto idrogeologico a Fermoy, Irlanda.....	62
4.2.1    Analisi dei risultati .....	65
CONCLUSIONI .....	67
BIBLIOGRAFIA .....	69

# INTRODUZIONE

Il dissesto idrogeologico costituisce un tema di particolare rilevanza per l'Italia a causa degli impatti sulla popolazione, sulle infrastrutture e sul tessuto economico e produttivo. Il forte incremento delle aree urbanizzate, verificatosi a partire dal secondo dopoguerra, spesso in assenza di una corretta pianificazione territoriale, ha portato a un considerevole aumento degli elementi esposti a frane e alluvioni e quindi del rischio. L'abbandono delle aree rurali montane e collinari ha inoltre determinato un mancato presidio e manutenzione del territorio. Il 16,6% del territorio nazionale è classificato a maggiore pericolosità; 1,28 milioni di abitanti sono a rischio frane e oltre 6 milioni di abitanti a rischio alluvioni [1]. Questi numeri impongono una maggiore attenzione da parte di tutti i soggetti sia pubblici che privati del paese, nonché un aumento di risorse da impiegare nella mitigazione dei rischi su tutto il territorio. Il continuo sviluppo di reti di monitoraggio ed allertamento ha permesso, ad esempio nella regione Marche, la creazione di una rete di monitoraggio che vanta un cospicuo numero di centraline e sensori sparsi per il territorio. Nella catena dei processi per la valutazione dello stato del territorio è in forte crescita, oggi, il telerilevamento, che utilizza immagini della superficie terrestre riprese da sensori a bordo di piattaforme satellitari, aerei o droni.

Il telerilevamento è un modo per ottenere informazioni da oggetti che si basa sulla raccolta e sull'analisi di dati senza che lo strumento usato per raccogliere i dati stessi sia in contatto diretto con l'oggetto studiato. La superficie della Terra è, in questo lavoro di tesi, l'oggetto di misura e le informazioni ottenute sono tutte necessarie ad aumentare la nostra conoscenza del pianeta.

L'imponente sviluppo scientifico e tecnico avvenuto nell'ultimo secolo ha fatto sì che venissero sviluppate delle tecnologie che fino a pochi decenni fa erano inimmaginabili; le tecniche di acquisizione dei dati e le tecnologie costruttive dei sensori sono state perfezionate ed ora è possibile accedere a prodotti con diverse risoluzioni geometriche, spettrali, temporali e radiometriche fornendo un valido supporto per monitorare e modellare caratteristiche fisiche e biologiche della Terra; una delle più interessanti applicazioni del telerilevamento satellitare è quella relativa allo studio e gestione dei disastri naturali.

Le alluvioni sono fra i più comuni disastri naturali che interessano il globo ed essere in grado di monitorare e tracciare eventi di questo tipo può facilitare il lavoro di soccorritori, tecnici e politici che dovranno intervenire con azioni mirate per fronteggiare l'emergenza. In questa tesi si propone un metodo automatico di estrazione di informazioni circa l'estensione e quindi la pericolosità di eventi quali allagamenti, inondazioni ed esondazioni che interessano il territorio provocando notevoli disagi alla popolazione e all'industria, utilizzando immagini *SAR (Synthetic Aperture Radar)* catturate da satelliti capaci di acquisire dati h24 anche in avverse condizioni meteo riuscendo a penetrare le nuvole. Si valuteranno i risultati ottenuti confrontandoli con dati provenienti da monitoraggi in-situ.

# RIEPILOGO ESECUTIVO

L'osservazione della Terra da satellite ha avuto negli ultimi anni un forte sviluppo in ambito internazionale grazie alla realizzazione di dispositivi in grado di effettuare missioni interdisciplinari. La ricerca spaziale rappresenta una delle più importanti avanguardie della Scienza e in questo contesto si collocano le tecnologie del Telerilevamento o Remote Sensing. L'Agenzia Spaziale Europea ESA, grazie all'ormai collaudato programma Copernicus, gestisce la missione Sentinel: la cattura di immagini radar e multispettrali attraverso una serie di satelliti. I satelliti del programma Copernicus rappresentano un considerevole investimento da parte dell'Agenzia Spaziale Europea, sono dotati di sensori con capacità osservative a scala globale e regionale per studi su acqua, suolo e vegetazione, ma anche per la misura di parametri fisici, atmosferici e marini.

L'obbiettivo di questa tesi è quello di dimostrare le potenzialità del telerilevamento dei satelliti Copernicus Sentinel-1 per l'analisi del dissesto idrogeologico in Italia in situazioni emergenziali come in occasione di alluvioni, esondazioni ed inondazioni.

I due satelliti Sentinel-1 sono in orbita dal 2014 e sono dotati di un sensore radar in grado di catturare immagini della superficie terrestre in ogni condizione atmosferica rilevandosi quindi estremamente efficace in circostanze climatiche avverse. Le immagini catturate e i diversi prodotti creati sono messi a disposizione della comunità gratuitamente attraverso il portale Copernicus di riferimento. [2]

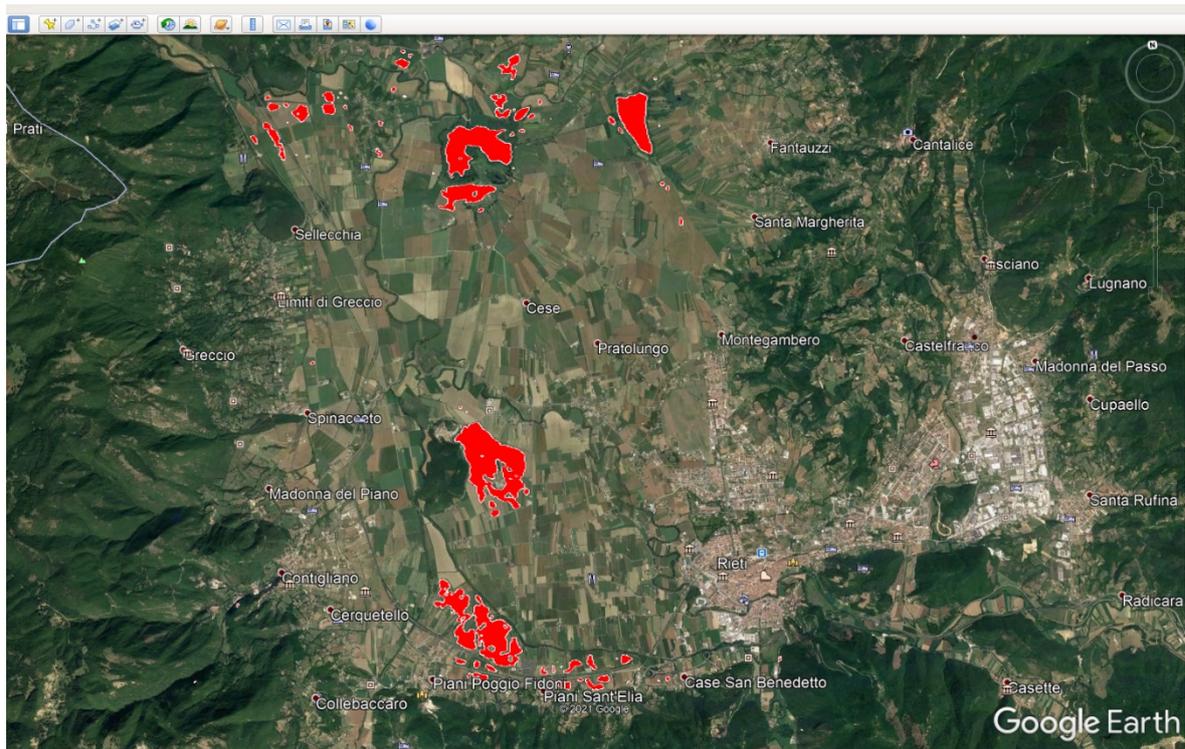
L'analisi dei prodotti *SAR* dei satelliti Sentinel ha permesso la creazione di un codice di elaborazione delle immagini per generare mappe quasi real-time dell'inondazione e quindi di potenziale supporto a soccorritori ed autorità in ambito emergenziale. A questo scopo si è scelto di utilizzare le immagini *Ground Range Detected (GRD)* acquisite con modalità *Interferometric Wide Swath (IW)* e tecnica *TOPSAR (Terrain Observation with Progressive Scans)*: il radar scansiona il terreno avanti e indietro rispetto la direzione di movimento del satellite e il fascio dell'antenna è direzionato ciclicamente in più percorsi adiacenti.

Il codice è scritto in linguaggio Python utilizzando l'ambiente di sviluppo integrato PyCharm e l'applicazione web Jupyter Notebook. L'utente può caricare in una cartella di input il prodotto scelto scaricato dal server Copernicus e un file .json con l'area di interesse per circoscrivere le operazioni di calcolo. Il codice si occupa, in automatico, di apportare una serie di correzioni all'immagine originale come la riduzione del rumore termico, il

miglioramento della calibrazione radiometrica, il filtraggio del rumore di *speckle* e le correzioni delle distorsioni geometriche del terreno. La fase più importante del processo è la binarizzazione dell'immagine e cioè l'elaborazione per individuare zone allagate o meno grazie al calcolo di una soglia di separazione del coefficiente di *backscattering* rilevato dall'antenna del satellite. A questo scopo si è scelto di utilizzare due tecniche di binarizzazione automatica e note in letteratura come Metodo Otsu e Metodo Minimo oltre che la possibilità di impostare manualmente questa soglia da parte dell'utente. Il codice è quindi in grado di generare una maschera dell'inondazione come mostrata in Figura 1. L'immagine sotto si riferisce all'allagamento della Piana Reatina dei giorni 26 e 27 Gennaio 2021 in seguito a violenti fenomeni temporaleschi; in rosso sono evidenziate le zone che dopo l'elaborazione dell'immagine di partenza risultano allagate.

Il risultato ottenuto non può essere confrontato con dati oggettivi di monitoraggi in-situ dato che non sono disponibili database per questo genere di eventi, si è quindi provveduto ad un confronto visivo tra la mappa ricavata e le fotografie aeree scattate dai Vigili del Fuoco in occasione dell'emergenza. Le zone inondate calcolate con il metodo automatico di elaborazione dell'immagine risultano confrontabili e di estensione simile a quelle delle foto aeree al netto di alcune aree, probabilmente perché le date di scatto tra le foto dall'elicottero e quelle da satellite differiscono di circa un giorno e parte dell'acqua nel suolo potrebbe essere drenata.

Quest'elaborato ha voluto dimostrare le potenzialità del telerilevamento e una possibile applicazione in un ambito di fondamentale importanza come quello della gestione delle emergenze usando dati ad accesso libero.



*Figura 1 Maschera dell'inondazione*

# CAPITOLO 1

## 1.1 Il dissesto idrogeologico

L'idrogeologia è la disciplina delle scienze geologiche che studia le acque sotterranee, anche in rapporto alle acque superficiali. Nell'accezione comune, il termine dissesto idrogeologico viene invece usato per definire i fenomeni e i danni reali o potenziali causati dalle acque in generale, siano esse superficiali, in forma liquida o solida, o sotterranee. Le manifestazioni più tipiche di fenomeni idrogeologici sono frane, alluvioni, erosioni costiere e valanghe.

Nel sistema di allertamento il rischio è differenziato e definito come:

- Il rischio idrogeologico, che corrisponde agli effetti indotti sul territorio dal superamento dei livelli pluviometrici critici lungo i versanti, dei livelli idrometrici dei corsi d'acqua della rete idrografica minore e di smaltimento delle acque piovane;
- Il rischio idraulico, che corrisponde agli effetti indotti sul territorio dal superamento dei livelli idrometrici critici (possibili eventi alluvionali) lungo i corsi d'acqua principali.

Il dissesto idrogeologico è diffuso in modo capillare e rappresenta un problema di notevole importanza. Le frane e le alluvioni sono le calamità naturali che si ripetono con maggior frequenza e causano, dopo i terremoti, il maggiore numero di vittime e di danni. Tra i fattori naturali che predispongono il nostro territorio ai dissesti idrogeologici, rientra la sua conformazione geologica e geomorfologica, caratterizzata da un'orografia (distribuzione dei rilievi) complessa e bacini idrografici generalmente di piccole dimensioni, che sono quindi caratterizzati da tempi di risposta alle precipitazioni estremamente rapidi. Il tempo che intercorre tra l'inizio della pioggia e il manifestarsi della piena nel corso d'acqua può essere dunque molto breve. Eventi meteorologici localizzati e intensi combinati con queste caratteristiche del territorio possono dare luogo a fenomeni violenti caratterizzati da cinematiche anche molto rapide (colate di fango e inondazioni improvvise).

Il rischio idrogeologico è inoltre fortemente condizionato anche dall'azione dell'uomo. La densità della popolazione, la progressiva urbanizzazione, l'abbandono dei terreni montani, l'abusivismo edilizio, il continuo disboscamento, l'uso di tecniche agricole poco rispettose dell'ambiente e la mancata manutenzione dei versanti e dei corsi d'acqua hanno

sicuramente aggravato il dissesto e messo ulteriormente in evidenza la fragilità del territorio italiano ed aumentato l'esposizione ai fenomeni e quindi il rischio stesso.

La frequenza di episodi di dissesto idrogeologico, che hanno spesso causato la perdita di vite umane e ingenti danni ai beni impongono una politica di previsione e prevenzione non più incentrata sulla riparazione dei danni e sull'erogazione di provvidenze, ma sull'individuazione delle condizioni di rischio e sull'adozione di interventi per la sua riduzione.

Nella Regione Marche provvedimenti normativi hanno imposto la sperimentazione delle aree a rischio, e si è sviluppato inoltre un sistema di allertamento e sorveglianza dei fenomeni che, assieme a un'adeguata pianificazione comunale di protezione civile rappresenta una risorsa fondamentale per la mitigazione del rischio, dove non si possa intervenire con misure strutturali. [3]

### **Procedure di allertamento**

Il Centro Funzionale regionale è inserito nella rete nazionale dei Centri Funzionali che, ai sensi della Legge 100 del 2012, concorre ad assicurare il governo e la gestione del sistema di allerta nazionale per il rischio meteo-idrogeologico ed idraulico. La rete nazionale dei Centri funzionali opera secondo criteri, metodi, standard e procedure comuni ed è componente del sistema nazionale della Protezione civile.

Il compito del Centro Funzionale è quello di fornire un servizio che sia di supporto alle decisioni delle autorità competenti per le allerte e per la gestione delle emergenze, nonché di assolvere alle necessità operative dei sistemi di protezione civile.

Il Centro Funzionale svolge le attività di previsione dirette all'identificazione degli scenari di rischio probabili, nei limiti delle conoscenze condivise dalla comunità scientifica e della strumentazione disponibile, al preannuncio, al monitoraggio, alla sorveglianza e alla vigilanza in tempo reale degli eventi e dei conseguenti livelli di rischio attesi, rivolti in particolare nell'ambito del rischio meteo-idrogeologico ed idraulico.

Il Centro Funzionale è parte della Protezione civile regionale e svolge anche le funzioni di Servizio Meteorologico Regionale e a tal fine redige quotidianamente un bollettino meteorologico, che non ha rilevanza ai fini dell'allertamento.

Il Centro Funzionale, inoltre, redige studi di climatologia e in occasione di fenomeni meteorologici significativi, anche per quanto riguarda gli effetti al suolo, specifici Rapporti

di Evento, in cui è contenuta la descrizione dell'evento meteo-idrogeologico e degli effetti prodotti dallo stesso.

Il Centro Funzionale, in fase di previsione, valuta, per ognuna delle zone di allerta in cui è suddiviso il territorio regionale, il livello di criticità atteso e il corrispondente livello di allerta. La previsione è articolata secondo la catena operativa previsionale, le cui fasi si possono sintetizzare come segue:

- acquisizione ed elaborazione dei dati meteo-idrologici, attraverso i vari sistemi di osservazione e rilevazione in dotazione al Centro Funzionale;
- previsione circa la natura e l'intensità dei fenomeni meteorologici attesi, anche attraverso l'utilizzo e la post-elaborazione di dati provenienti da sistemi modellistici previsionali;
- previsione degli effetti al suolo associati ai fenomeni previsti e dei possibili scenari d'evento, che vengono valutati attraverso opportuni livelli di criticità per le porzioni del territorio interessate.

Tali fasi si concretizzano nell'emissione dei documenti di allertamento che forniscono le informazioni riguardo gli scenari di evento atteso.

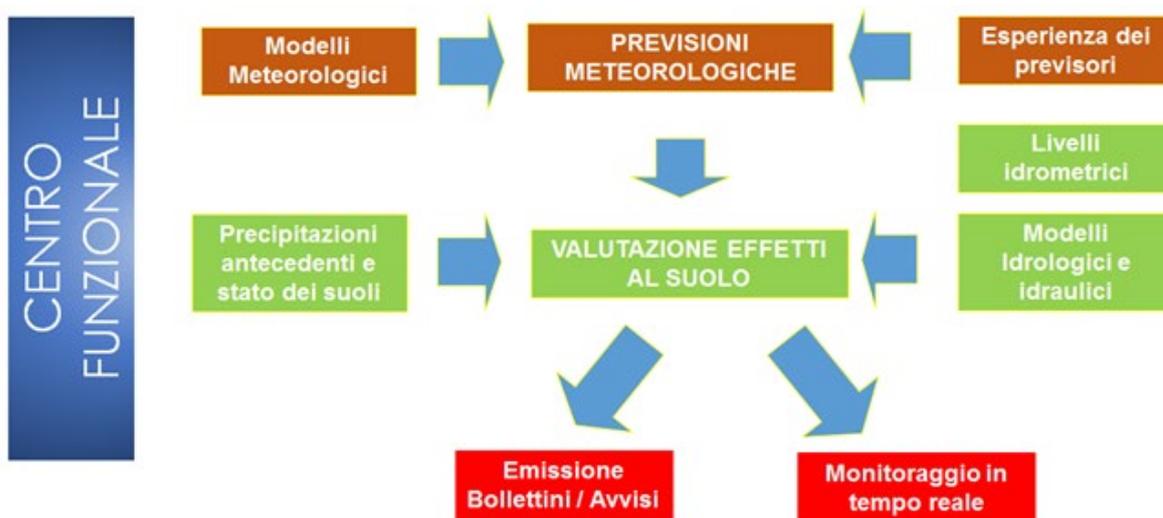


Figura 2 Catena operativa previsionale del Centro Funzionale

A seguito della definizione del livello di criticità, da parte del Centro Funzionale e del conseguente livello di allerta, deve essere attivata una fase operativa da parte dai soggetti responsabili delle pianificazioni e delle procedure ai diversi livelli territoriali dichiarata; la

Dirigenza della Struttura Regionale di Protezione civile comunica al territorio il livello di allerta e la fase operativa dichiarata per le strutture Regionali.

Le attività dell'allertamento non si esauriscono con la fase previsionale, ma continuano durante l'evento che viene monitorato per tutta la sua durata. [4]

## **1.2 Il monitoraggio idrogeologico nella Regione Marche**

Il reticolo idrografico della Regione Marche è definito a “pettine” grazie alla singolare direzione che caratterizzano i fiumi marchigiani, hanno infatti un andamento tra loro parallelo con la direzione delle aste fluviali SO-NE (con alcune eccezioni, in alcuni fiumi, nella parte iniziale montana).

In generale i fiumi marchigiani hanno una lunghezza ridotta (se si pensa che solamente alcuni risultano nell'ordine dei 100 Km) ed un regime idrografico di tipo torrentizio. Tale regime è dovuto essenzialmente dalla variabilità delle portate annue e tra i fattori che influenzano le portate annue ricordiamo il clima, caratterizzato da estati secche e piogge concentrate soprattutto nel periodo invernale. Un altro fattore rilevante è che la maggior parte dei fiumi nascono nella dorsale umbro-marchigiana, caratterizzata da acquiferi calcarei che restituiscono ai fiumi le acque meteoriche in tempi molto brevi, non consentendo un'alimentazione omogenea annua.

In generale le profondità dei corsi d'acqua sono assai ridotte e nessuno di essi risulta navigabile, se non per piccoli tratti nei pressi delle foci, dove si ritrovano in alcuni corsi d'acqua dei sistemi porto-canale.

L'attività di vigilanza nella Regione Marche avviene grazie alla Rete di Monitoraggio Idrogeologico Regionale (MIR) [5] che vanta oltre 100 stazioni di telemisura, circa 300 sensori e ponti radio SHF e UHF. La Sala Operativa Unificata Permanente (SOUP) si trova al centro del flusso informativo per l'emissione di bollettini di vigilanza sulla base dei dati acquisiti.

Tutte le comunicazioni e i trasferimenti dati da e per le stazioni periferiche avvengono attraverso un sistema radio dedicato che garantisce un elevato standard di affidabilità e la massima riduzione dei tempi di interrogazione e ricezione degli allarmi dalle stazioni. Si basa su un sistema ridondato di ponti radio; il primo, in tecnologia *spread-spectrum* a 2.4 GHz, è il sistema principale; il secondo, operante sulla banda UHF, garantisce la ridondanza

del sistema trasmissivo consentendo di mantenere attive le comunicazioni anche in caso di avaria del sistema principale.

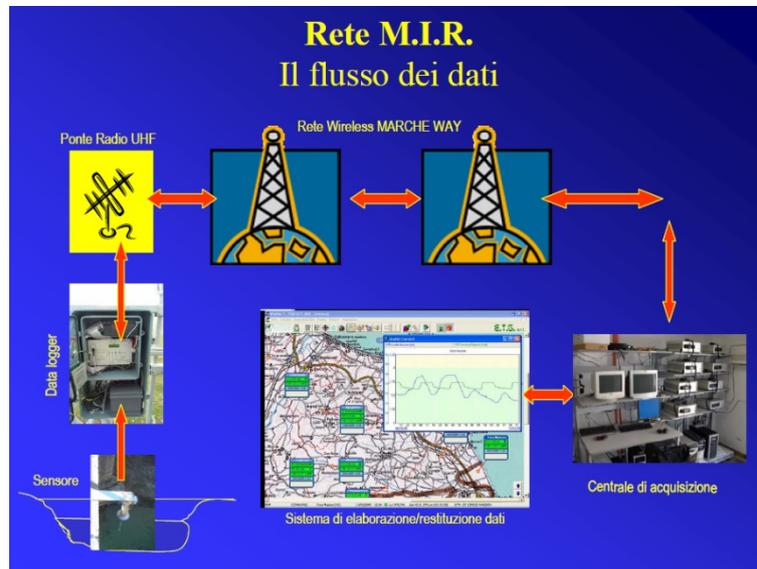


Figura 3 Schema di principio della rete MIR

La sensoristica principale si basa sull'utilizzo di idrometri, pluviometri, nivometri, sensori di temperatura, umidità, direzione e velocità del vento. I dati vengono gestiti da apposita centralina che si occupa della memorizzazione e trasmissione verso la centrale di acquisizione.

L'idrometro ad ultrasuoni in Figura 4 è uno strumento costituito da un trasmettitore-ricevitore di ultrasuoni da posizionare sopra la superficie dell'acqua, alla distanza di alcuni metri. L'altezza del pelo liquido è dedotta dal tempo impiegato dall'onda sonora per propagarsi dal trasmettitore all'acqua e per essere riflessa dall'acqua al ricevitore. I principali inconvenienti connessi all'utilizzo di tale sensore sono la variazione della temperatura e dell'umidità dell'aria e del vento, che possono influenzare la velocità di propagazione delle onde alterando la misura oppure dalla presenza di ostacoli nello spazio di misura. Gli strumenti utilizzati sono corredati di appositi sensori e software di gestione che predispongono correzioni automatiche degli strumenti in base alla variazione dei principali parametri. Sono collegati a centraline di registrazione elettroniche. Il campo di misura è tra 0 e 25 m con sensibilità di 2 cm, l'errore è  $<1$  cm.

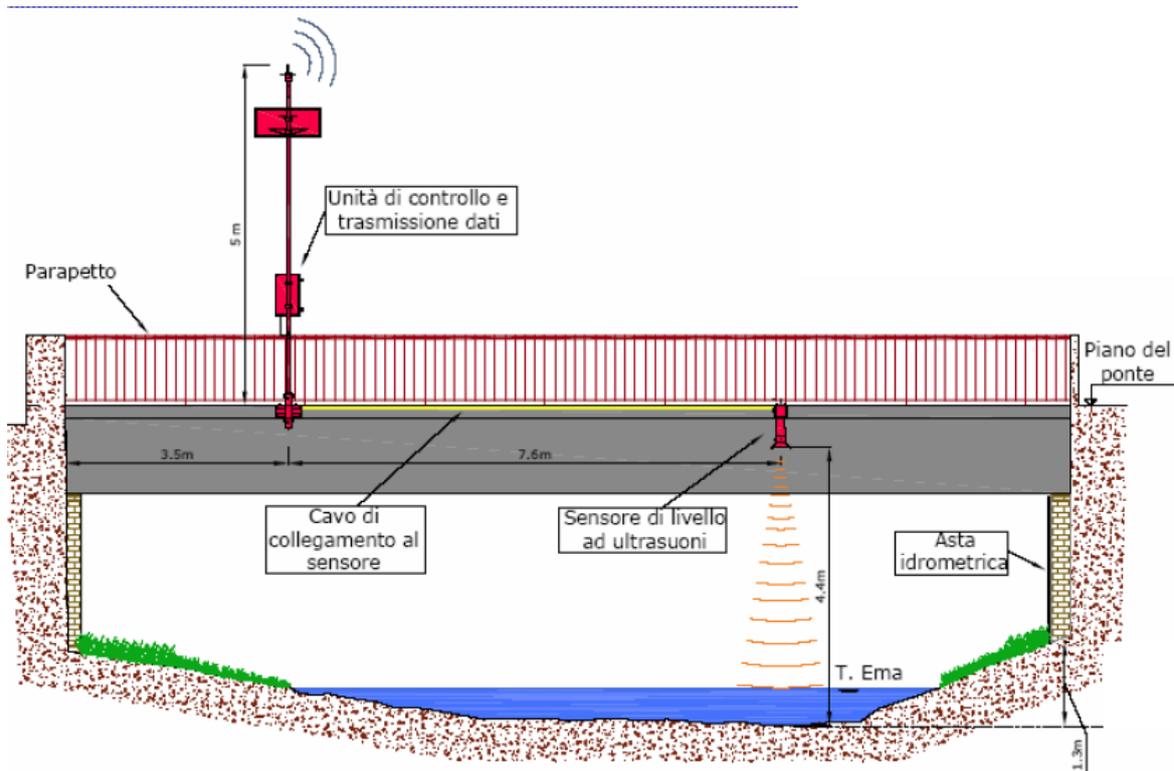


Figura 4 Idrometro ad ultrasuoni

L'idrometro ad ultrasuoni è affiancato da un'asta idrometrica che ne permette la calibrazione avendo un riferimento altimetrico immutabile nel tempo.

La sensoristica MIR si compone anche di stazioni di monitoraggio stand-alone capaci di gestire in automatico situazioni di allerta idrologica tramite l'impiego di telecamere *PTZ* (*Pan, Tilt, Zoom*) e sensori idrometrici. Le stazioni sono dotate di sensori di livello piezometrici *PLS* (*Pressure Level Sensor*) che si basano su un elemento sensibile piezoresistivo costituito da alcune resistenze in struttura a ponte diffuse su un wafer di silicio. Il trasduttore è contenuto entro una camera stagna riempita di olio e chiusa da una membrana in acciaio inox su cui esercita la pressione il liquido esterno. Il corpo è in acciaio inossidabile resistente all'immersione. Il sensore è immerso nel corpo idrico e la pressione esercitata dalla colonna di liquido sovrastante è riferita alla pressione atmosferica e viene quindi ricavata l'altezza del liquido. La telecamera *PTZ* è opportunamente configurata per catturare e trasmettere lo streaming dati video verso il centro di controllo dove avviene anche un'analisi video.

## CAPITOLO 2

Le tecniche e sensori tradizionali, esposti nel Capitolo 1, sono oggi integrati da dati ambientali tele rilevati da satellite e da aereo permettendo lo sviluppo delle applicazioni e degli strumenti di monitoraggio terrestre.

Osservare la Terra con sensori remoti è diventata una pratica diffusa e consolidata all'interno della comunità scientifica, delle pubbliche amministrazioni e del mondo delle imprese che si occupano di territorio e ambiente grazie alle opportunità offerte in ambito internazionale dai progetti per l'Osservazione della Terra promossi dall'ESA.

### 2.1 Il programma Copernicus

Copernicus [6] è un progetto nato dalla collaborazione tra la Commissione Europea ed ESA (Agenzia Spaziale Europea), dedicato a monitorare il nostro pianeta e il suo ambiente a beneficio di tutti i cittadini in maniera libera e gratuita. Il monitoraggio è reso disponibile dai satelliti chiamati Sentinel che osservano la terra e da sistemi di misurazione terrestre usati per integrare, calibrare e validare le informazioni ricevute.

I satelliti Sentinels sono suddivisi in sei famiglie e sono progettati per differenti scopi. Attualmente sono operative le seguenti missioni:

- Sentinel-1 attraverso il *SAR (Synthetic Aperture Radar)* rende possibile l'acquisizione d'immagini qualunque sia il tempo atmosferico, giorno e notte per i servizi di monitoraggio del terreno e degli oceani. La missione comprende due satelliti, 1A e 1B, in orbita polare;
- Sentinel-2 è una missione che comprende la costellazione di due satelliti in orbite polari e sfasati di 180° tra loro. Riprendono immagini multispettrali per servizi multidisciplinari;
- Sentinel-3 genera prodotti ottici e radar per lo studio dell'ambiente marino e del territorio globale, come la topografia e temperatura superficiale di mari ed oceani e garantisce un supporto affidabile per i servizi di previsione e di monitoraggio ambientali e climatico;
- Sentinel-4 mette a disposizione dati per il monitoraggio continuo della composizione atmosferica come la qualità dell'aria, i livelli di ozono, di diossido di azoto;

- Sentinel-5 è del tutto simile al Sentinel 4, si occupa di rilevazioni in atmosfera, di supporto ad applicazioni climatiche;
- Sentinel-5 Precursor è la prima missione dedicata al monitoraggio della nostra atmosfera come il rilevamento di sostanze chimiche pericolose per l'uomo;
- Sentinel-6 è una serie di due satelliti meteorologici e oceanografici.

### **2.1.1 Sentinel-1**

La missione Sentinel-1 [7] è costituita da una costellazione di due satelliti equipaggiati con sola strumentazione radar: il satellite Sentinel-1A è stato lanciato il 3 Aprile 2014, mentre l'1B il 25 Aprile 2016. Risultato di una cooperazione tra ESA e la Commissione Europea, ha diverse applicazioni:

- Oceani e ghiacci, fornisce immagini radar per generare mappe temporali dei ghiacci marini utili al passaggio delle navi nelle acque artiche oltre a provvedere ad una campionatura continua dell'oceano al fine di ottenere informazioni su venti e onde;
- Monitoraggio del territorio, permette una valutazione eccellente delle variazioni di copertura del terreno (*land cover*) fornendo importanti informazioni per applicazioni agricole e il monitoraggio di colture su larga scala e di foreste;
- Emergenze, i prodotti Sentinel-1 sono particolarmente adatti per una risposta celere alle emergenze ambientali di qualsiasi natura, specialmente per eventi meteorologici estremi, data la capacità del sensore radar di acquisire immagini anche in presenza di nubi.

#### **Caratteristiche dei satelliti**

La navicella progettata per la missione Sentinel-1 è un satellite stabilizzato a tre assi caratterizzato da sensori di sole, stelle, giroscopici e di campo magnetico con capacità di sterzo in ogni asse. È equipaggiato con un set di pannelli solari in grado di produrre 5900 W (a fine vita) accumulabili in una batteria. Il bus di Piattaforma Italiana Multi Applicativa, assicura una conoscenza accurata dell'assetto, della posizione e dell'orbita; un sistema di propulsione dedicato permette un preciso controllo dell'orbita. Il satellite ha strumentazioni relative alla gestione dei dati, la propulsione, la potenza, il controllo delle temperature, la comunicazione coi centri di controllo, l'analisi e gestione di eventuali anomalie.

La massa totale, al lancio, è di circa 2300 Kg.

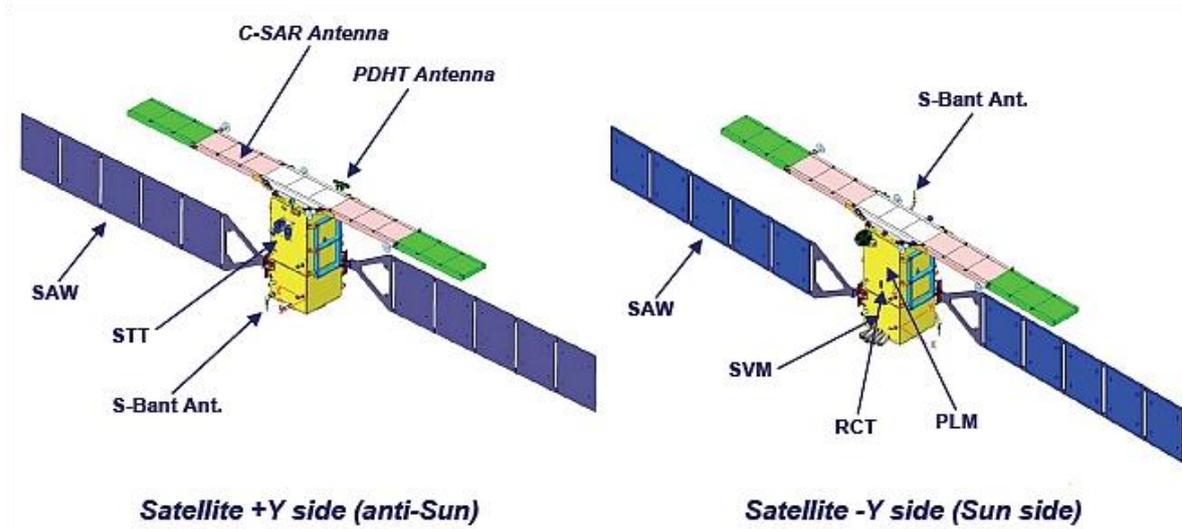


Figura 5 Rappresentazione del satellite Sentinel-1

### Orbita e copertura geografica

L'orbita è quasi polare eliosincrona con un ciclo di ripetizione di 12 giorni e 175 orbite per ciclo per ogni satellite. Sentinel 1A e 1B condividono la stessa orbita sfasati di 180°; con entrambi i satelliti, il ciclo di ripetizione è di 6 giorni.

Il controllo dell'orbita deve essere molto preciso in particolare per applicazioni interferometriche. Il controllo è definito tramite un "tubo orbitale" di 50 m di raggio, attorno al tracciato nominale.

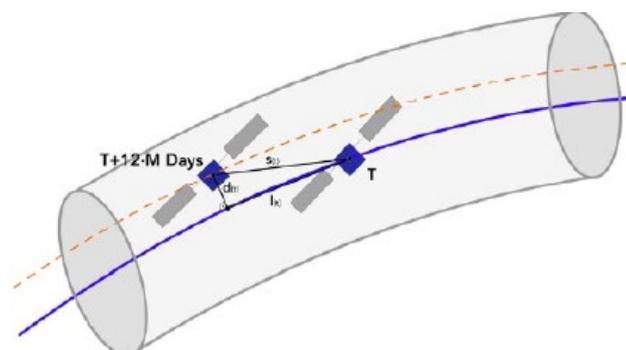
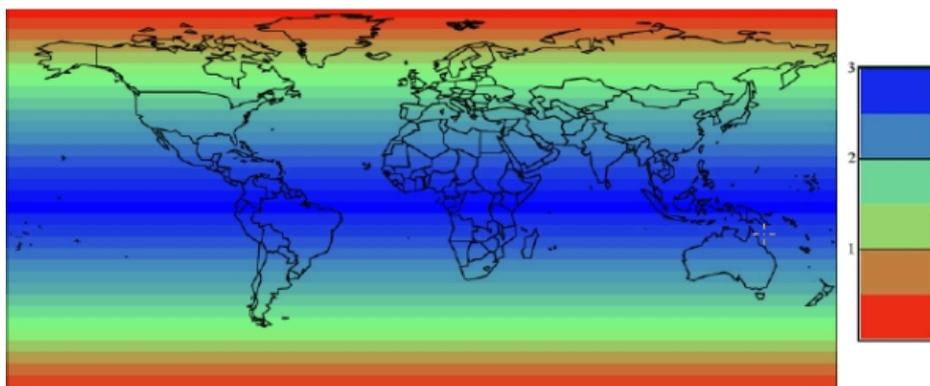


Figura 6 Tubo orbitale Sentinel 1

Un singolo satellite è in grado di mappare il globo ogni 12 giorni. La costellazione dei due satelliti permette tempi di rivisitazione più brevi in base alla latitudine, circa due giorni

nelle nostre zone. I dati radar sono disponibili nei server Copernicus generalmente il giorno dopo l'acquisizione. Un file kml disponibile sul sito [8] della missione contiene informazioni circa le acquisizioni passate e future dei satelliti.



- ✓ Two satellites in a 12 day orbit
- ✓ Repeat frequency: 6 days (important for coherence)
- ✓ Revisit frequency: (asc/desc & overlap): 3 days at the equator, <1 day at high latitudes (Europe ~ 2 days)

Figura 7 Frequenza di rivisitazione

## Strumento C-SAR

I satelliti Sentinel-1 sono equipaggiati con un radar ad apertura sintetica in banca C.<sup>1</sup> (Strumento C-SAR [9]) che supporta operazioni in singola (HH o VV) e doppia polarizzazione (HH+HV, VV+VH)<sup>2</sup>. I sensori SAR trasmette radiazione elettromagnetica grazie all'antenna verso la superficie terrestre. I sensori catturano l'energia riflessa dopo l'interazione con la superficie. La quantità di energia ricevuta è governata da parametri radar come la lunghezza d'onda, l'angolo di incidenza, la polarizzazione e da parametri del terreno come la costante dielettrica e la rugosità superficiale.

<sup>1</sup> Banda C: range di frequenze fra 4 e 8 GHz (3.75 e 7.5 cm di lunghezza d'onda).

<sup>2</sup> I radar di telerilevamento sono generalmente progettati per trasmettere radiazioni polarizzate verticalmente o orizzontalmente. Ciò significa che il campo elettrico dell'onda si trova su un piano verticale o su un piano orizzontale. Allo stesso modo, il radar può ricevere radiazioni polarizzate verticalmente o orizzontalmente e talvolta entrambe. I piani di polarizzazione trasmessa e ricevuta sono designati dalle lettere H per Orizzontale e V per Verticale. Pertanto, la polarizzazione di un'immagine radar può essere HH, per trasmissione orizzontale, ricezione orizzontale, VV per trasmissione verticale, ricezione verticale, HV per trasmissione orizzontale, ricezione verticale e viceversa (VH). Quando la polarizzazione della radiazione ricevuta è la stessa della radiazione trasmessa, si dice che l'immagine sia polarizzata in modo simile. Quando la polarizzazione della radiazione ricevuta è l'opposto della radiazione trasmessa, si dice che l'immagine sia polarizzata in modo incrociato. La polarizzazione incrociata richiede una dispersione multipla da parte del bersaglio e quindi si traduce in una retrodiffusione più debole rispetto alla polarizzazione simile. I radar satellitari generalmente usano la polarizzazione simile perché i segnali con polarizzazione incrociata sono troppo deboli per produrre una buona immagine. La polarizzazione è stabilita dall'antenna, che può essere regolata per essere diversa in trasmissione e in ricezione. La riflettività delle microonde da un oggetto dipende dalla relazione tra lo stato di polarizzazione e la struttura geometrica dell'oggetto. I possibili stati di polarizzazione, oltre a verticale e orizzontale, includono tutti gli orientamenti angolari del vettore E e gli orientamenti variabili nel tempo che portano a polarizzazioni ellittiche e circolari.

Le modalità di acquisizione sono quattro (Figura 9):

- *Stripmap (SM)*
- *Interferometric Wide swath (IW)*
- *Extra-Wide swath (EW)*
- *Wave mode (WV)*

Le modalità di acquisizione principali sono la *Interferometric Wide swath (IW)* e la *Wave mode (WV)*. Entrambe le modalità utilizzano la tecnica di acquisizione delle immagini chiamata *TOPSAR, Terrain Observation with Progressive Scans SAR*. I dati sono ottenuti direzionando ciclicamente il fascio dell'antenna in più sub-strisciate adiacenti, in ogni sub-strisciata il radar scansiona il terreno avanti e indietro rispetto la direzione di movimento del satellite (Figura 8).

La modalità *Stripmap* è usata su piccole isole e in rare circostanze per supportare servizi di emergenza.

*Extra Wide Swath* utilizza cinque sub-strisciate adiacenti a discapito della risoluzione, è usata sopra i ghiacci polari e in certe aree marittime.

*Wave mode* è la modalità di default sopra gli oceani.

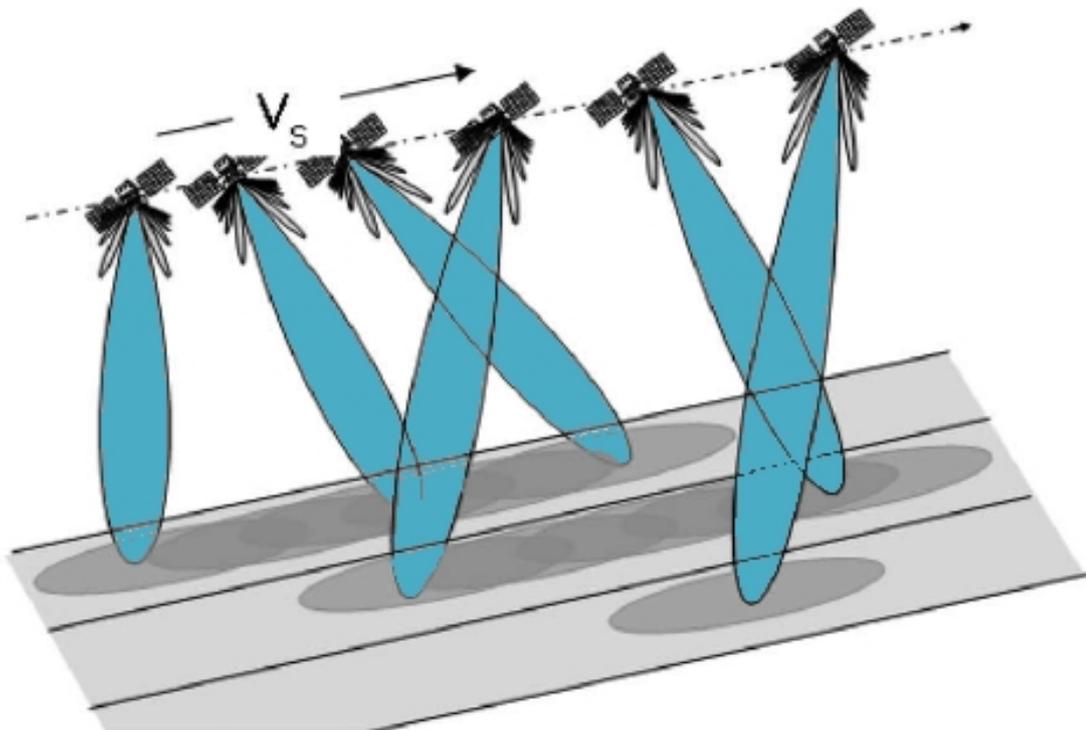


Figura 8 Acquisizione TOPSAR

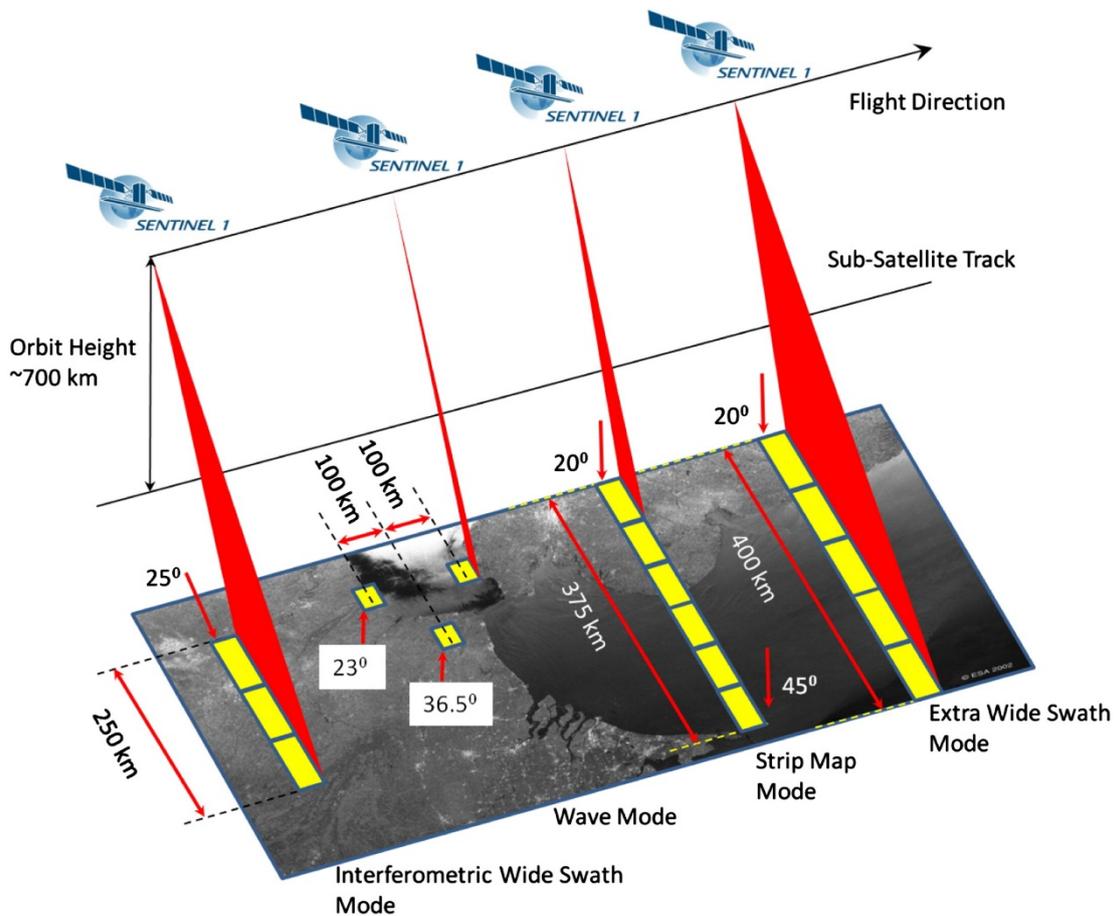


Figura 9 Modalità di acquisizione Sentinel 1

Le differenti modalità di acquisizione si caratterizzano per angolo di incidenza del raggio al terreno, risoluzione spaziale, polarizzazione e strisciata, termine con il quale si indica la larghezza della porzione di superficie terrestre acquisita dal sensore. Le caratteristiche principali sono elencate nella tabella 1.

Modalità	Angolo di incidenza	Risoluzione spaziale	Strisciata	polarizzazione
SM	20° - 45°	5x5 m	80 km	HH+HV, VV+VH, HH, VV
IW	29° - 46°	5x20 m	250 km	HH+HV, VV+VH, HH, VV
EW	19° - 47°	20x40 m	400 km	HH+HV, VV+VH, HH, VV
WV	22° - 35° 35° - 38°	5x5 m	20x20 km	HH, VV

Tabella 1 Modalità di acquisizione Sentinel 1

## Prodotti disponibili [10]

Ogni modalità può generare prodotti a diversi livelli:

- Livello 0: sono i dati grezzi, da cui provengono gli altri prodotti;
- Livello 1 *Single Look Complex (SLC)*, sono dati *SAR* calibrati e geo-referenziati usando dati di orbita e assetto provenienti dal satellite, sono dati da 8GB (in modalità IW a doppia polarizzazione); convenzionalmente si tratta di un prodotto *slant range* ottenuto con geometria zero Doppler;
- Livello 1 *Ground Range Detected (GRD) Multi-Look*, sono dati *SAR* calibrati, geo-referenziati e proiettati al suolo basandosi sul sistema di ellissoide WGS84, l'informazione di fase è persa;
- Livello 2, sono prodotti di interesse geofisico ottenuti da dati *SAR* e vengono utilizzati per applicazioni in ambito oceanico (*Level 2 Ocean, OCN*) legati a vento (*Ocean Wind Field, OWI*), onde (*Ocean Swell Spectra, OSW*) e correnti (*Surface Radial Velocity, RVL*).

Il prodotto di risoluzione maggiore è quello ottenuto con modalità SM e livello 1 GRD.

## 2.1.2 Sentinel-2

La missione Sentinel-2 [11] consiste in due piattaforme multispettrali, 13 bande, ad alta e media risoluzione spaziale a seconda della specifica banda.

I dati provenienti dai satelliti Sentinel-2 sono utili ad una vasta gamma di applicazioni, anche in combinazione con i dati radar provenienti dalla missione Sentinel-1, come:

- Cambiamento climatico, le immagini provenienti dai satelliti Sentinel-2 possono essere di grande supporto per il monitoraggio della deforestazione o della desertificazione;
- Monitoraggio del territorio, ha i medesimi campi applicativi della missione Sentinel-1 e insieme ad essa può fornire dati per la stesura di mappe di copertura e uso del suolo;
- Gestione delle emergenze, dato il breve tempo di rivisitazione (5 giorni all'equatore) i dati Sentinel-2 si prestano molto bene a questo tipo di applicazioni anche nel caso di emergenze umanitaria;

- Sicurezza, i prodotti Sentinel-2 possono essere utilizzati come supporto per il monitoraggio dei confini, delle rotte marine commerciali e migratorie, al fine di aumentare la sicurezza interna dell'Unione Europea.

### Caratteristiche dei satelliti

I satelliti pesano circa 1200 kg e sono progettati per una vita utile di circa 7 anni.



Figura 10 Rappresentazione del satellite Sentinel-2

Entrambi i satelliti sono equipaggiati con sensori *MSI (Multispectral Instrument)* sulla piattaforma lavorano passivamente raccogliendo la luce del sole riflessa dalla Terra.

### Orbita e copertura geografica

I due satelliti operano sulla medesima orbita inclinata di  $98.62^\circ$  (rispetto all'equatore), eliosincrona, ad una elevazione media di 786 km, sfasati di  $180^\circ$ . L'orbita eliosincrona minimizza l'impatto delle ombreggiature sul terreno. L'orbita è mantenuta stabile da un sistema di propulsione dedicato e tramite le misurazioni di un ricevitore.

Sono progettati affinché la frequenza di rivisitazione sia di 5 giorni all'equatore (2-3 giorni alle latitudini medie).

I Sentinel-2 acquisiscono sistematicamente dati su terra emersa e aree costiere.

## Strumento MSI

I satelliti Sentinel-2 sono equipaggiati con un sensore multispettrale, MSI [12], strumento *push-broom*, acquisisce righe di dati perpendicolarmente alla strisciata a terra (*swath*) e sfrutta il movimento in avanti del satellite per acquisire nuove stringhe di dati.

I raggi di luce incidenti vengono divisi, filtrati e concentrati su piani focali separati, uno per le bande del visibile (VIS) e dell'infrarosso vicino (*Near-Infrared NIR*) e uno per l'infrarosso a onde corte (*Short Wave Infrared SWIR*). Una sorta di otturatore evita che i sensori vengano colpiti dai raggi provenienti direttamente dal sole.

La strisciata è di 290 km e per poterlo garantire i sensori VNIR e SWIR sono composti da 12 rilevatori posti su due file orizzontali, come mostrato in Figura 11.

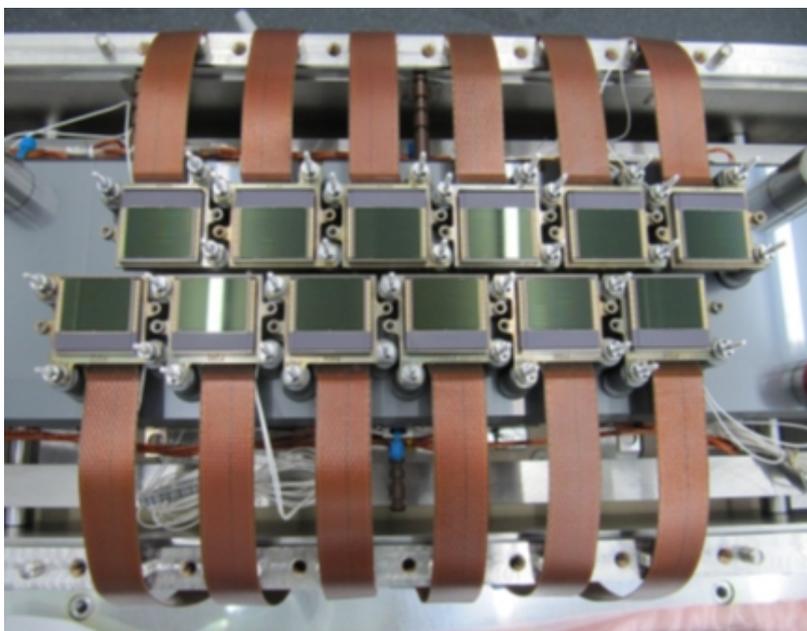


Figura 11 Piano focale VNIR

I sensori Sentinel-2 sono in grado di ricevere informazioni su 13 bande spettrali a diverse risoluzioni spaziali, come riportato in Tabella 2.

	S2A		S2B			
Band Number	Central wavelength (nm)	Bandwidth (nm)	Central wavelength (nm)	Bandwidth (nm)	Spatial resolution (m)	
1	442.7	21	442.3	21	60	Ultra blue (Coastal and Aerosol)
2	492.4	66	492.1	66	10	Blue
3	559.8	36	559.0	36	10	Green
4	664.6	31	665.0	31	10	Red
5	704.1	15	703.8	16	20	Visible and Near Infrared (VNIR)
6	740.5	15	739.1	15	20	Visible and Near Infrared (VNIR)
7	782.8	20	779.7	20	20	Visible and Near Infrared (VNIR)
8	832.8	106	833.0	106	10	Visible and Near Infrared (VNIR)
8a	864.7	21	864.0	22	20	Visible and Near Infrared (VNIR)
9	945.1	20	943.2	21	60	Short Wave Infrared (SWIR)
10	1373.5	31	1376.9	30	60	Short Wave Infrared (SWIR)
11	1613.7	91	1610.4	94	20	Short Wave Infrared (SWIR)
12	2202.4	175	2185.7	185	20	Short Wave Infrared (SWIR)

*Tabella 2 Bande spettrali per i sensori Sentinel-2*

È possibile realizzare una combinazione delle bande per ottenere una specifica informazione dall'immagine. Per esempio, ci sono combinazioni di bande che mettono in risalto caratteristiche geologiche, civili o forestali in un'immagine. L'immagine con il colore naturale è ottenuta come combinazione delle bande rossa (B4), verde (B3), blu (B2).



Figura 12 Short-Wave Infrared (B12, B8A, B4)

Usando la combinazione di bande B12, B8A e B4, si mette in risalto la vegetazione in diverse sfumature di verde a seconda della sua densità, come in Figura 12.

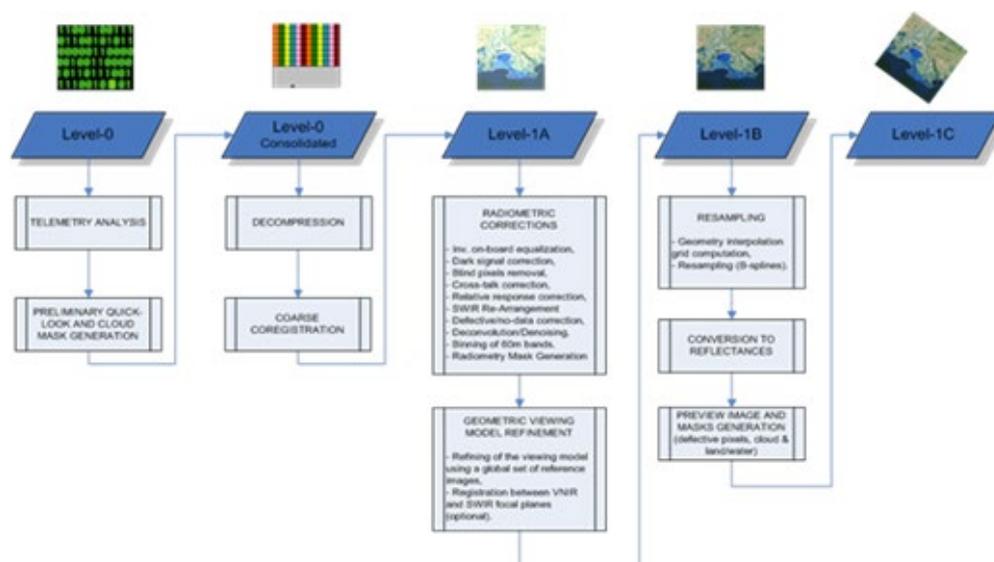


Figura 13 Processo dal Livello-0 al Livello-1C

### Prodotti disponibili [13]

I dati acquisiti dallo strumento MSI sono processati dal *Payload Data Ground Segment (PDGS)* che genera differenti livelli di prodotto: Livello-0, Livello-1A, Livello-1B, Livello-1C e Livello-2A (Figura 13). Ogni livello è composto da granuli con differenti grandezze.

I prodotti dei Livello-0 e Livello-1A sono rispettivamente immagini grezze compresse e decomprese e sono la base di partenza per la produzione dei prodotti di Livello-1.

Gli unici livelli disponibili agli utenti sono:

- Livello-1C: è il risultato di correzioni radiometriche e geometriche. Le immagini sono orto-rettificate e geo-referenziate usando un Modello di Elevazione Digitale (DEM) per proiettarle in geometria cartografica. I prodotti L1C sono campionati con *Ground Sampling Distance*<sup>3</sup> (*GSD*) costante di 10, 20, 60 m in base alla risoluzione delle differenti bande spettrali e le misure per-pixel sono ottenute secondo riflettanza *Top of Atmosphere*.

L'immagine risultante è un granulo 100x100 Km<sup>2</sup> orto-rettificato in proiezione WGS84/UTM.

- Livello-2A: all'immagine L1C è applicata una correzione atmosferica. L'algoritmo permette il calcolo della riflettanza *Bottom of Atmosphere BOA* dalla riflettanza *TOA* del livello precedente.

Sono applicate altre correzioni all'immagine come quella topografica e dei cirrocumuli.

L'immagine risultante è un granulo 100x100 Km<sup>2</sup> orto-rettificato in proiezione WGS84/UTM.

---

<sup>3</sup> *GSD*: *Ground Sampling Distance* è la distanza tra i punti centrali di due pixel vicini di un'immagine digitale, misurata sul terreno reale. Il *GSD* ti dice quanto è grande un pixel misurato sul terreno. Il *GSD* si misura in mm/pixel o cm/pixel. Quanto più piccolo è il *GSD*, tanto più precisa e definita è l'informazione contenuta nel relativo pixel.



*Figura 14 A sinistra immagine LIC TOA e alla destra immagine L2A BOA*

## 2.2 Accesso ai dati Copernicus

### 2.2.1 Copernicus Open Access Hub

Il Copernicus Open Access Hub, raggiungibile all'indirizzo web <https://scihub.copernicus.eu/dhus/#/home>, provvede al completo e libero accesso dei prodotti di Sentinel-1 e Sentinel-2 e si presenta con la seguente interfaccia (Figura 15).

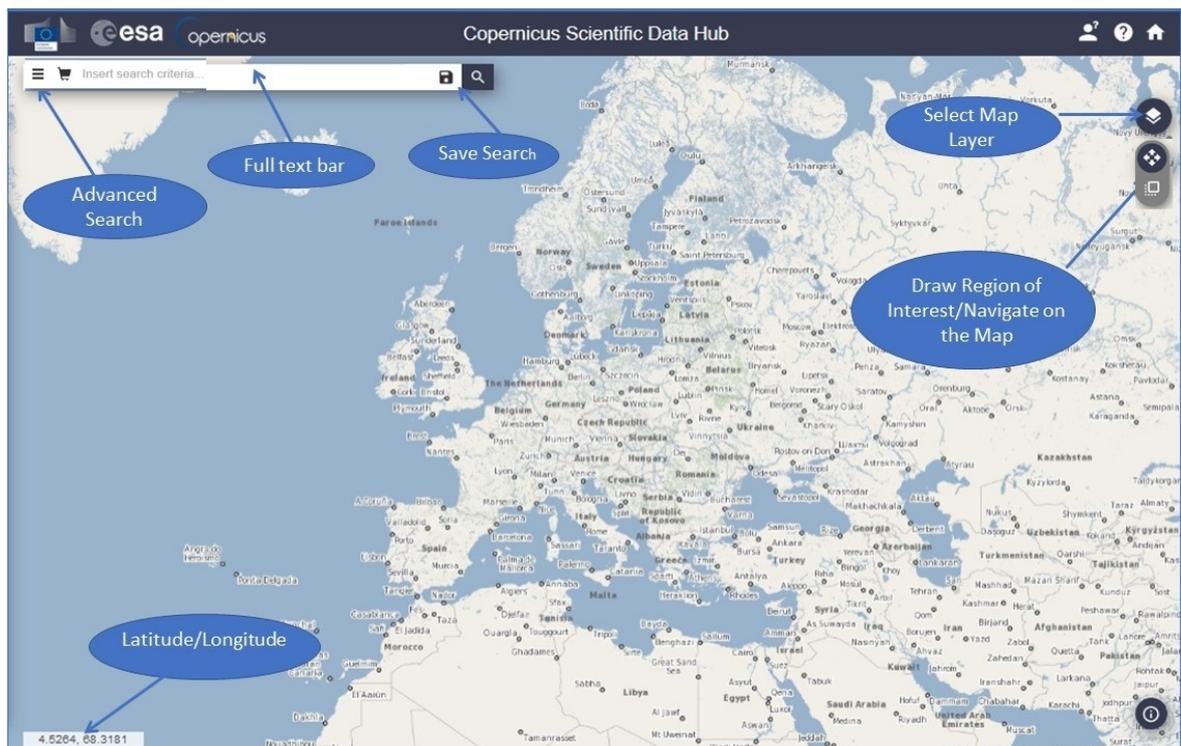


Figura 15 Interfaccia grafica Schiuh (<https://scihub.copernicus.eu/dhus/#/home>)

In questo portale è possibile selezionare l'area di interesse e svolgere una ricerca avanzata grazie alla finestra sulla sinistra dello schermo dalla quale è possibile scegliere la missione, il livello di pretrattamento del dato, modalità e periodo di acquisizione, polarizzazione e copertura nuvolosa.

Il download del file avviene dopo aver scelto il prodotto di interesse. I prodotti Sentinel sono distribuiti usando una specifica variazione del formato *SAFE* (*Standard Archive Format for Europe*).

## Formattazione dei dati

I pacchetti di dati messi a disposizione dalle missioni Sentinel sono individuati da una nomenclatura standard che li identifica in modo univoco, in Figura 16 è riportato come esempio il pacchetto *.SAFE* di un prodotto Sentinel-1.

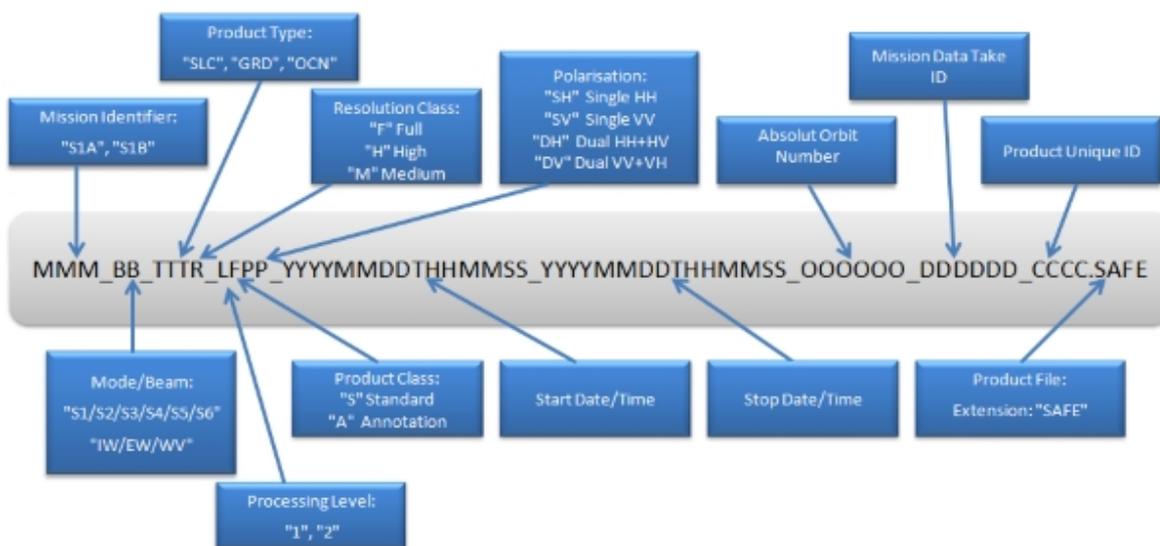


Figura 16 Nomenclatura standard dei prodotti Sentinel

La formattazione della cartella varia in base alla tipologia di prodotto scaricato. Per i dati di Sentinel-2, la formattazione della cartella principale è composta come in Figura 17, le informazioni vengono catalogate per tipologia:

- Un manifest.safe file scritto con sintassi XML che contiene i dati generali del prodotto;
- Un'anteprima dell'immagine;
- Una cartella contenente i dataset delle rilevazioni e i dati delle immagini (maschere) in formato GML-JPEG2000 (*GRANULE*);
- Un'altra cartella contenente le informazioni delle *datastrip* per i vari livelli (*DATASTRIP*);
- Una ulteriore cartella contenente dati ausiliari (*AUX\_DATA*);
- Infine, un file in formato HTML di anteprima.

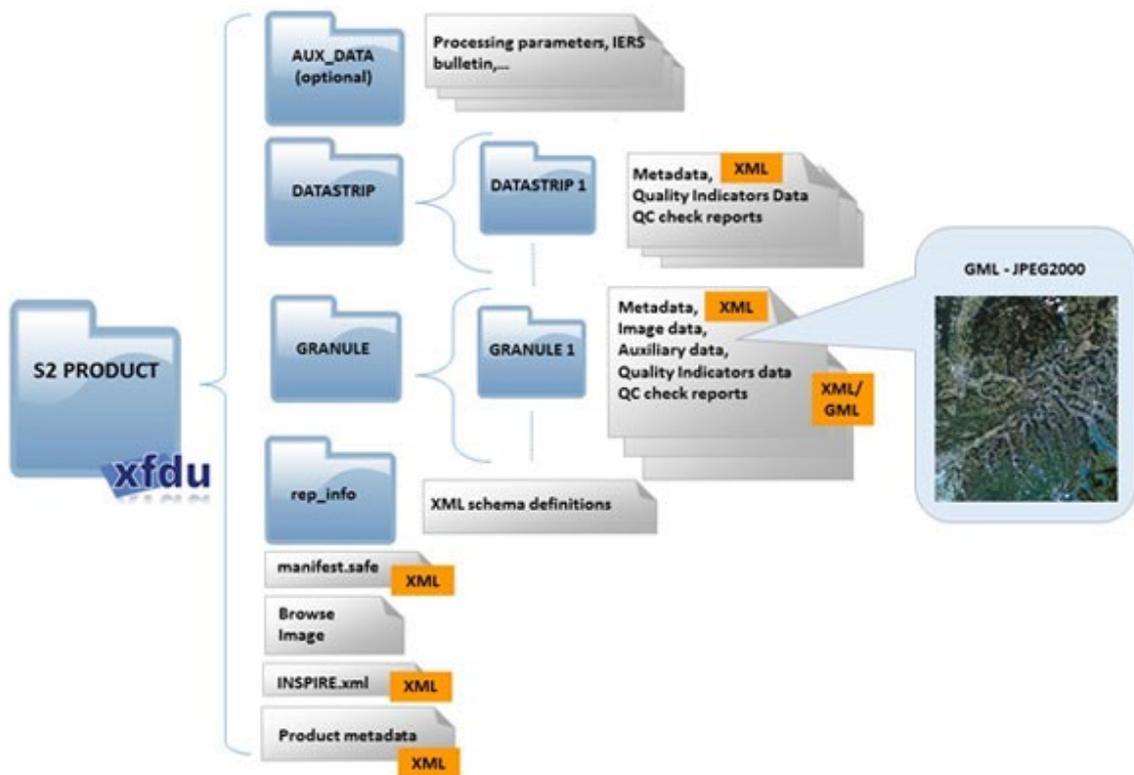


Figura 17 Formattazione della cartella .SAFE per Sentinel-2

### Visualizzazione dei dati scaricati

Sentinel Application Platform (SNAP) [14] è un software open source per PC sviluppato da ESA che permette l'analisi e il processamento di immagini provenienti da satelliti diversi come Sentinel-1 e 2. L'interfaccia grafica Windows è intuitiva e completa di tutto il necessario per visualizzare e navigare rapidamente in immagini giga-pixels del Sentinel-1. In Figura 18 è mostrata l'interfaccia della home page di SNAP in cui è aperta un'immagine SAR del centro Italia. Grazie a questo programma è possibile, ad esempio, usare il download automatico dai server Copernicus dei dati per la correzione di orbita o di calibrazione del prodotto in oggetto.

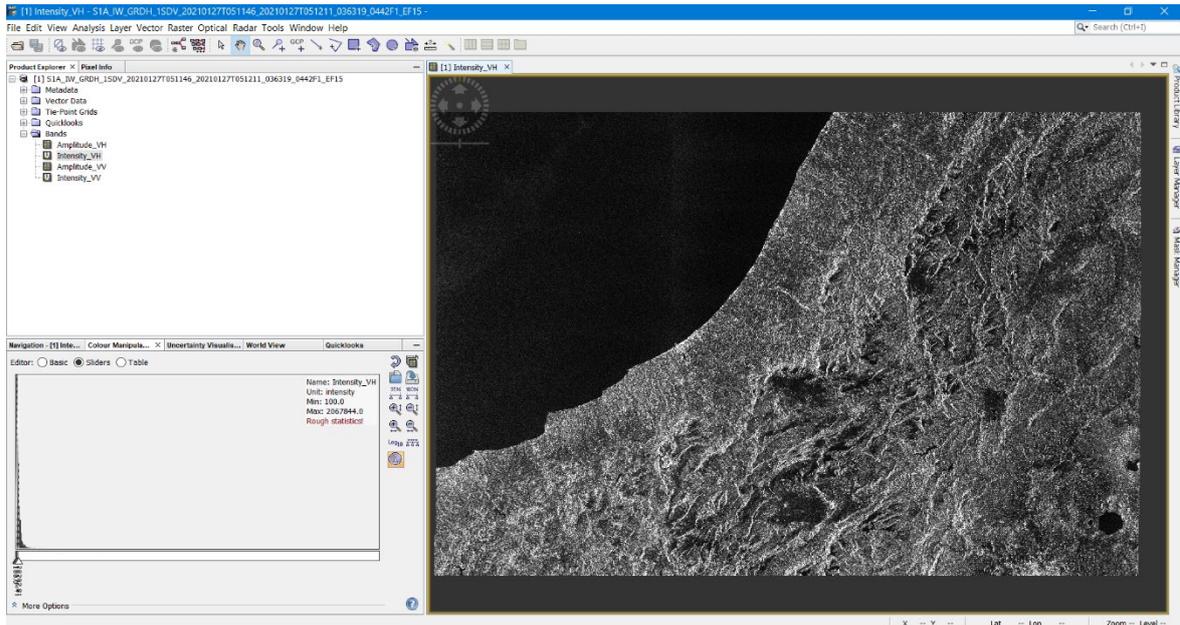


Figura 18 SNAP software per Windows

## 2.3 Copernicus Emergency Management System

Il Copernicus Emergency Management System [15] è un servizio gratuito offerto da ESA che rende disponibili mappe di porzioni di superficie terrestre interessate da disastri naturali usando immagini satellitari e vari dati geografici. La mappatura avviene su richiesta, quando un Utente Autorizzato richiede l'attivazione del servizio; la mappatura può riguardare vari scenari in situazioni di crisi come inondazioni, terremoti, incendi, eruzioni vulcaniche, crisi umanitarie.

Le mappe sono disponibili per l'intero ciclo dell'evento e divise in due categorie temporali:

- **Rapid Mapping**, disponibili in poco tempo, sono utili a diversi scopi:
  - *Reference Maps*: per avere come riferimento il territorio prima dell'evento;
  - *First Estimate Maps*: per avere una rapida panoramica dell'evento e delle aree colpite;
  - *Delineation Maps*: con certezza stabilisce i confini dell'evento;
  - *Grading Maps*: per stimare intensità e danni dell'evento.

Il servizio offre dati vettoriali e mappe pronte all'uso anche in base all'urgenza e al tipo di prodotto. *First Estimate Maps* è il prodotto che più velocemente può arrivare all'utente finale, anche dopo due ore dalla prima immagine disponibile.

- **Risk and Recovery Mapping**, disponibili per analisi di lungo periodo come prevenzione e riduzione dei rischi.

### **Attivazione del servizio**

Gli utenti sono entità e organizzazioni di livello regionale, nazionale, Europeo e internazionale attive nel settore della gestione delle crisi.

Gli **Utenti Autorizzati**, che per l'Italia sono il Presidente del Consiglio dei ministri e la Protezione Civile, possono attivare il servizio mandando una richiesta direttamente a *European Response Coordination Centre, ERCC*.

Sono **Utenti Associati** coloro che per conto degli Utenti Autorizzati possono attivare il servizio.

Gli **Utenti Generici Pubblici** non sono autorizzati ad attivare il servizio ma sono informati qualora ci fosse una richiesta di attivazione.

## CAPITOLO 3

### 3.1 Utilizzo delle immagini Sentinel-1 per l'analisi del dissesto idrogeologico

Nell'immagine radar ottenuta dal satellite Sentinel-1 l'intensità del segnale misurata dal sensore (*backscatter*) dipende da caratteristiche geometriche ed elettromagnetiche della superficie riflettente e dalle modalità di emissione/acquisizione del segnale. Ad esempio il *backscatter* è nullo nel caso di superfici perfettamente lisce poiché il segnale viene riflesso interamente nella direzione opposta a quella di provenienza; più la superficie è irregolare, maggiore sarà la probabilità che parte del segnale venga riflesso verso il sensore, quindi il *backscatter* aumenta; nel caso di superfici ad angolo (*corner reflector*) l'onda subisce una doppia riflessione per cui torna interamente nella direzione di provenienza e il *backscatter* è massimo (ciò accade ad esempio nelle aree edificate). [16]

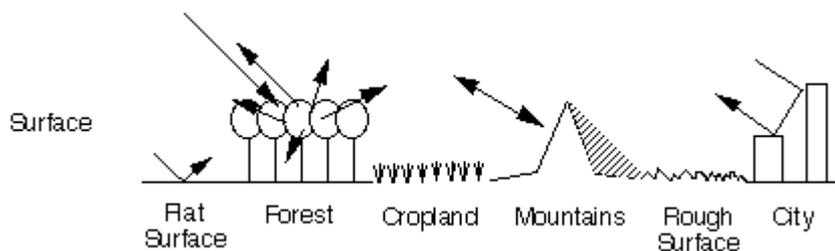


Figura 19 Backscatter e diversi tipo di superfici

Ogni pixel nell'immagine radar rappresenta il valore di *backscatter* stimato per quell'area. Le zone più scure dell'immagine denotano valori più bassi del coefficiente mentre le zone più chiare i valori più alti, nel primo caso solo una piccola parte dell'energia è riflessa. L'acqua appare come una superficie piatta e nell'immagine si contraddistingue per il colore scuro.

L'idea alla base di quest'analisi è quella di individuare un valore soglia del coefficiente di *backscatter*: i pixel i cui valori sono sotto la soglia si ipotizza che siano riferiti alla

presenza di acqua, mentre sopra la soglia al resto del paesaggio. A partire da questo valore di riferimento si crea un'immagine binaria. [17]

### 3.1.1 Binarizzazione dell'immagine mediante metodi adattivi

Nel processamento e analisi di immagini la necessità di separare due grandezze correlate come, ad esempio, terra ferma e acqua, può essere soddisfatta utilizzando tecniche adattive per calcolare il valore migliore della soglia di separazione tra le due grandezze. Per quest'analisi i metodi di binarizzazione utilizzati sono due: metodo Otsu e metodo Minimo e possono essere scelti una volta avviato il codice.

Gli algoritmi automatici di selezione della soglia generalmente comprendono i seguenti step:

- processamento dell'immagine in scala di grigi;
- calcolo dell'istogramma (distribuzione dei pixel);
- calcolo della soglia;
- sostituzione dei pixel dell'immagine di input in base alla soglia calcolata.

Il metodo Otsu [18] è largamente usato in applicazioni dove è necessario dividere l'immagine in scala di grigi in due differenti classi. Calcola automaticamente la soglia a partire dall'istogramma dell'immagine minimizzando la varianza su ciascuna delle classi. Di solito, questa tecnica produce i risultati appropriati per le immagini bimodali. L'istogramma di un tale tipo di immagine contiene due picchi chiaramente espressi, che rappresentano diversi intervalli di valori di intensità. L'idea è separare l'istogramma dell'immagine in due cluster con una soglia definita come risultato della minimizzazione della varianza ponderata di queste classi.

Nel codice è implementato grazie alla funzione:

```
threshold_otsu = skimage.filters.threshold_otsu(band_data) [19]
```

L'algoritmo del Metodo Minimo smussa ripetutamente l'istogramma finché non ci sono solo due picchi e così viene individuata la soglia, al massimo vengono fatte 10000 iterazioni.

In python è utilizzata la seguente funzione per la sua implementazione:

```
threshold_minimum = skimage.filters.threshold_minimum(band_data) [20]
```

## 3.2 Implementazione del codice per l'analisi delle immagini

Il progetto è realizzato utilizzando l'ambiente di sviluppo integrato PyCharm in linguaggio Python. Il software SNAP basato su linguaggio JAVA e discusso al Capitolo 2 è integrato in Python grazie all'API Snappy [21] che per poter funzionare necessita di SNAP installato sulla stessa macchina.

Il codice creato per l'analisi delle immagini Sentinel-1 si articola in più fasi che di seguito sono esposte e per poter funzionare, necessita di alcuni file in ingresso:

- Il file .zip scaricato come indicato nel paragrafo “1.2 Copernicus Open Access Hub” che individua il prodotto Sentinel\_1 GRD più consono alle proprie esigenze;
- Il file dell'area di interesse da analizzare in formato .json. Occorre definire l'Area di Interesse per poter circoscrivere le operazioni di analisi e ottenere migliori performance in termini di tempo di processamento e di accuratezza del risultato. Il file json può essere facilmente reperito utilizzando servizi web open source che, grazie ad una semplice ed intuitiva interfaccia grafica, ne permettono la creazione;

### Impostazioni di base

La prima fase del codice permette all'utente di scegliere se utilizzare le polarizzazioni VH, VV o entrambe dell'immagine da analizzare e se mostrare risultati e grafici durante il processo. All'utente è richiesto di inserire il percorso della cartella di input contenente il file .zip del prodotto Sentinel e il file .json dell'area di interesse; si chiede di impostare anche la cartella di output che è quella in cui verranno esportati i dati elaborati.

```
# polarizzazione da processare
polarizations = 'both'           # 'VH', 'VV', 'both'
# mostra risultati intermedi se 'True'
plotResoults = True           # 'True', 'False'
# imposta cartella input
print ("Inserisci il percorso della cartella di input con i file .ZIP e .json")
input_folder=input()
# imposta cartella output
print ("Inserisci il percorso della cartella di output")
output_folder=input()
```

## Startup del codice

Nella seguente fase si caricano i moduli Python necessari al funzionamento del codice stesso e si inizializzano funzioni di base, come quella inerente alla ricerca ed acquisizione dell'area di interesse oppure per la visualizzazione di immagini e grafici durante il processamento.

```
import sys
import matplotlib.pyplot as plt
import numpy as np
import json
import glob
import os
import ipywidgets
import time
import shutil
import ipyleaflet
import geopandas
import snappy
import jpy
import skimage.filters
import functools
import subprocess
import math
import pandas as pd
from sentinelat.sentinel import SentinelAPI, read_geojson, geojson_to_wkt
from datetime import date
from IPython.display import display
from osgeo import ogr, gdal, osr
from zipfile import ZipFile

# DEFINIZIONE FUNZIONI
# Si cerca file json per determinare AOI
def readJSONFromAOI(path):
    # cerca GeoJSON file nella sottocartella 'AOI'
    if len(glob.glob('%s/*.geojson' % path)) == 1:
        file = glob.glob('%s/*.geojson' % path)[0]
    elif len(glob.glob('%s/*.json' % path)) == 1:
        file = glob.glob('%s/*.json' % path)[0]

    # apri file JSON file e salva dati
    with open(file, 'r') as f:
        data_json = json.load(f)

    return data_json
```

```

# visualizza bande e grafici nel processamento
def plotBand(band, threshold, method, binary=False):
    # color stretch
    vmin, vmax = 0, 1
    # leggi il valore dei pixel
    w = band.getRasterWidth()
    h = band.getRasterHeight()
    band_data = np.zeros(w * h, np.float32)
    band.readPixels(0, 0, w, h, band_data)
    band_data.shape = h, w
    # color stretch
    if binary:
        cmap = plt.get_cmap('binary')
    else:
        vmin = np.percentile(band_data, 2.5)
        vmax = np.percentile(band_data, 97.5)
        cmap = plt.get_cmap('gray')
    # visualizza bande
    fig, (ax1, ax2) = plt.subplots(1,2, figsize=(16,6))
    ax1.imshow(band_data, cmap=cmap, vmin=vmin, vmax=vmax)
    ax1.set_title(band.getName())
    # visualizza e salva istogramma
    band_data.shape = h * w
    thNameVal = ""
    if method == 0 :
        thNameVal = 'otsu ' + 'threshold={}'.format(math.floor(threshold*100)/100.0)
    elif method == 1:
        thNameVal = 'minimum ' + 'threshold={}'.format(math.floor(threshold*100)/100.0)
    else:
        thNameVal = 'custom ' + 'threshold={}'.format(math.floor(threshold*100)/100.0)
    ax2.hist(np.asarray(band_data[band_data != 0], dtype='float'), bins=2048)
    ax2.axvline(x = threshold, color = 'r', label = thNameVal)
    ax2.set_title('Histogram: %s' % band.getName())
    ax2.legend()
    plt.savefig(output_folder + 'hist %s.png' % band.getName())

    for ax in fig.get_axes():
        ax.label_outer()

# directory di lavoro
directory = os.getcwd()

```

## Processamento dell'immagine Sentinel-1

La terza è la fase di processamento dell'immagine: step successivi permettono di elaborare opportunamente il file in ingresso.

I prodotti di Livello 1 Ground Range Detected (GRD) necessitano di una serie di correzioni, che in ordine di applicazione, sono di seguito esposte. [22]

- Applicazione di una precisa orbita di acquisizione:  
(S1\_Orb = snappy.GPF.createProduct('Apply-Orbit-File', parameters, S1\_Sub),  
il file di orbita nei metadata contiene informazioni sulla posizione e velocità del satellite, generalmente determinata dopo qualche giorno dalla generazione del prodotto. In automatico, è possibile effettuare il download e l'aggiornamento del vettore di stato dell'orbita nei metadata dei prodotti, informazioni sulla velocità e sulla posizione sono così più accurate. Questo step è opzionale e il codice continua se i file di orbita non sono disponibili nei server Sentinel.
- Rimozione del rumore termico di cui le immagini del Sentinel-1 sono affette:  
S1\_Tnr = snappy.GPF.createProduct('ThermalNoiseRemoval', parameters, S1\_Orb),  
l'effetto del rumore è ridotto nella texture delle sub-strisciate, normalizzando il segnale riflesso con l'intera scena riducendo le discontinuità fra sub-strisciate.
- Calibrazione radiometrica:  
S1\_Cal = snappy.GPF.createProduct('Calibration', parameters, S1\_Tnr),  
la calibrazione è la procedura che permette la conversione dei valori di intensità dei pixel in coefficiente di scattering. L'informazione per applicare la calibrazione è contenuta nel prodotto GRD.
- Filtraggio del Rumore Speckle:  
S1\_Spk = snappy.GPF.createProduct('Speckle-Filter', parameters, S1\_Cal),  
appare nelle immagini SAR come rumore granulare. È dovuto alle interferenze delle onde riflesse, il filtraggio è una procedura per aumentare la qualità dell'immagine, ma può compromettere la qualità di piccoli dettagli ripresi nella stessa.
- Correzione delle distorsioni geometriche del terreno:  
S1\_TC = snappy.GPF.createProduct('Terrain-Correction', parameters, S1\_Spk\_db),  
i dati SAR vengono generalmente rilevati con un angolo di visione variabile maggiore di 0 gradi, l'immagine presenterà una certa distorsione relativa alla geometria laterale. Le correzioni del terreno hanno lo scopo di compensare queste distorsioni in modo

che la rappresentazione geometrica dell'immagine sia il più vicino possibile alla realtà. Si utilizza un modello di elevazione digitale per correggere la posizione di ogni pixel.

- Binarizzazione:

```
S1_floodMask = binarization(GlobCover, expressions),
```

in questa fase l'istogramma è elaborato per separare le zone dell'immagine con probabile presenza di acqua dalle altre.

Il file in ingresso viene letto dalla cartella "input" e ne vengono restituite alcune informazioni come il nome, data, ora e tipo di prodotto. Il file viene visualizzato in una mappa interattiva e ne viene evidenziata anche l'area di interesse.

Questa parte di codice restituisce l'immagine elaborata assieme all'istogramma creato con il coefficiente di *backscatter* e al valore calcolato della soglia.

```
# definizione prodotto S1
```

```
def getScene(path):
```

```
    if len(files) is 1:
```

```
        file_path = path
```

```
    else:
```

```
        file_path = path.selected
```

```
    S1_source = snappy.ProductIO.readProduct(file_path)
```

```
    name, sensing_mode, product_type, polarization, height, width, band_names = ([ for i
```

```
in range(7))
```

```
    for i in files:
```

```
        sensing_mode.append(i.split("_")[1])
```

```
        product_type.append(i.split("_")[2])
```

```
        polarization.append(i.split("_")[3])
```

```
        # Read with snappy
```

```
        name.append(S1_source.getName())
```

```
        height.append(S1_source.getSceneRasterHeight())
```

```
        width.append(S1_source.getSceneRasterWidth())
```

```
        band_names.append(S1_source.getBandNames())
```

```
    df_S1_source = pd.DataFrame({'Name': name, 'Sensing Mode': sensing_mode, 'Product Type': product_type, 'Polarization': polarization, 'Height': height, 'Width': width, 'Available Bands': band_names})
```

```
    display(df_S1_source)
```

```
# leggi le coordinate geografiche dal file metadata
```

```
meta_data = S1_source.getMetadataRoot().getElement('Abstracted_Metadata')
```

```
# centra la mappa
```

```

    center = (meta_data.getAttributeDouble('centre_lat'),
meta_data.getAttributeDouble('centre_lon'))
    locations = [[{'lat' : meta_data.getAttributeDouble('first_near_lat'), 'lng' :
meta_data.getAttributeDouble('first_near_long')},
                {'lat' : meta_data.getAttributeDouble('last_near_lat'), 'lng' :
meta_data.getAttributeDouble('last_near_long')},
                {'lat' : meta_data.getAttributeDouble('last_far_lat'), 'lng' :
meta_data.getAttributeDouble('last_far_long')},
                {'lat' : meta_data.getAttributeDouble('first_far_lat'), 'lng' :
meta_data.getAttributeDouble('first_far_long')}]

# mappa interattiva
basic_map = ipyleaflet.Map(center = center, zoom = 7.5)
# polygon immagine S1
polygon_fix = ipyleaflet.Polygon(locations = locations, color='royalblue')
basic_map.add_layer(polygon_fix)
# display map
basic_map.add_control(ipyleaflet.ScaleControl(position='bottomleft'))
display(basic_map)

# AOI file convertito in JSON e visualizzato nella mappa
try:
    # leggi JSON da AOI
    data_json = readJSONFromAOI(input_folder)
    # visualizzalo nella mappa
    basic_map.add_layer(ipyleaflet.GeoJSON(data = data_json, style = {'color' :
'green'}))
    # applica il subset secondo l'AOI
    footprint = geojson_to_wkt(data_json)
    # avvia il processo
    processing(S1_source, footprint)

# se non è presente il file AOI
except:
    print('Errore nel processo o AOI non presente, per favore caricane uno.\n',
flush=True)

# calcola le soglie in base alle bande in input
# SNAP API: https://step.esa.int/docs/v6.0/apidoc/engine/
def getThreshold(S1_band):

    # leggi le bande
    w = S1_band.getRasterWidth()
    h = S1_band.getRasterHeight()
    band_data = np.zeros(w * h, np.float32)
    S1_band.readPixels(0, 0, w, h, band_data)
    band_data.shape = h * w

```

```

# calcola la soglia usando metodo Otsu
threshold_otsu = skimage.filters.threshold_otsu(band_data)

# calcola la soglia usando metodo minimo
threshold_minimum = skimage.filters.threshold_minimum(band_data)

modality = input('Scegli tra "otsu" e "minimum" o immmetti un valore:')
if modality == 'otsu':
    threshold = threshold_otsu
    method = 0
elif modality == 'minimum':
    threshold = threshold_minimum
    method = 1
else:
    threshold = float(modality)
    method = 2

return threshold, method

# calcola maschera binaria del prodotto
def binarization(S1_product, expressions):

    BandDescriptor =
    jpy.get_type('org.esa.snap.core.gpf.common.BandMathsOp$BandDescriptor')
    targetBands = jpy.array('org.esa.snap.core.gpf.common.BandMathsOp$BandDescriptor',
    len(expressions))

    # loop
    for i in range(len(expressions)):
        targetBand = BandDescriptor()
        targetBand.name = '%s' % S1_product.getBandNames()[i]
        targetBand.type = 'float32'
        targetBand.expression = expressions[i]
        targetBands[i] = targetBand

    parameters = snappy.HashMap()
    parameters.put('targetBands', targetBands)
    mask = snappy.GPF.createProduct('BandMaths', parameters, S1_product)

    return mask

# step di processo
def processing(S1_source, footprint):

    # Subset
    parameters = snappy.HashMap()
    parameters.put('copyMetadata', True)

```

```

geom = snappy.WKTReader().read(footprint)
parameters.put('geoRegion', geom)
parameters.put('sourceBands', sourceBands)
global S1_Sub
S1_Sub = snappy.GPF.createProduct('Subset', parameters, S1_source)
# status update
print("\nSubset successfully generated.\n", flush=True)

# Applica file di orbita
print('1. Apply Orbit File:      ', end="", flush=True)
start_time = time.time()
parameters = snappy.HashMap()
# continua anche se non disponibili i file
parameters.put('continueOnFail', True)
global S1_Orb
S1_Orb = snappy.GPF.createProduct('Apply-Orbit-File', parameters, S1_Sub)
print('--- %.2f seconds ---' % (time.time() - start_time), flush=True)

# Rimozione Rumore Termico
print('2. Thermal Noise Removal:  ', end="", flush=True)
start_time = time.time()
parameters = snappy.HashMap()
parameters.put('removeThermalNoise', True)
global S1_Tnr
S1_Tnr = snappy.GPF.createProduct('ThermalNoiseRemoval', parameters, S1_Orb)
print('--- %.2f seconds ---' % (time.time() - start_time), flush=True)

# Calibracione Radiometrica
print('3. Radiometric Calibration: ', end="", flush=True)
start_time = time.time()
parameters = snappy.HashMap()
parameters.put('outputSigmaBand', True)
global S1_Cal
S1_Cal = snappy.GPF.createProduct('Calibration', parameters, S1_Tnr)
print('--- %.2f seconds ---' % (time.time() - start_time), flush=True)

# Filtraggio Rumore Speckle
print('4. Speckle Filtering:      ', end="", flush=True)
start_time = time.time()
parameters = snappy.HashMap()
parameters.put('filter', 'Refined Lee')
parameters.put('windowSize', "7x7")
global S1_Spk
S1_Spk = snappy.GPF.createProduct('Speckle-Filter', parameters, S1_Cal)
print('--- %.2f seconds ---' % (time.time() - start_time), flush=True)

# Conversione in dB

```

```

global S1_Spk_db
S1_Spk_db = snappy.GPF.createProduct('LinearToFromdB', snappy.HashMap(),
S1_Spk)

# Correzione delle distorsioni geometriche del terreno
print('5. Terrain Correction:      ', end="", flush=True)
parameters = snappy.HashMap()
parameters.put('demName', 'SRTM 1Sec HGT')
parameters.put('demResamplingMethod', 'BILINEAR_INTERPOLATION')
parameters.put('imgResamplingMethod', 'BILINEAR_INTERPOLATION')
parameters.put('pixelSpacingInMeter', 10.0)
parameters.put('mapProjection', 'AUTO:42001')
parameters.put('nodataValueAtSea', False)
parameters.put('saveSelectedSourceBand', True)
global S1_TC
S1_TC = snappy.GPF.createProduct('Terrain-Correction', parameters, S1_Spk_db)
print('--- %.2f seconds ---' % (time.time() - start_time), flush=True)

# Binarizzazione
print('6. Binarization:          ', end="", flush=True)
start_time = time.time()
# Aggiungi banda GlobCover
parameters = snappy.HashMap()
parameters.put('landCoverNames', 'GlobCover')
global GlobCover
GlobCover = snappy.GPF.createProduct('AddLandCover', parameters, S1_TC)
# array vuoto per le espressioni band maths
expressions = [" for i in range(S1_TC.getNumBands())]
# array vuoto per soglie
thresholds = np.zeros(S1_TC.getNumBands())
method = np.zeros(S1_TC.getNumBands())
# loop
for i in range(S1_TC.getNumBands()):
    # calcola le soglie delle bande
    thresholds[i], method[i] = getThreshold(S1_Spk_db.getBandAt(i))
    # espressione in base alla soglia
    expressions[i] = 'if (%s < %s) then 1 else NaN' % (S1_TC.getBandNames()[i],
thresholds[i])
    # usare 'if (%s < %s && land_cover_GlobCover != 210) then 1 else NaN' per
nascondere acqua permanente.
# Binarizzazione
global S1_floodMask
S1_floodMask = binarization(GlobCover, expressions)
print('--- %.2f seconds ---' % (time.time() - start_time), flush=True)

# Secondo Filtraggio Rumore Speckle
print('7. Speckle Filtering:      ', end="", flush=True)

```

```

start_time = time.time()
parameters = snappy.HashMap()
parameters.put('filter', 'Median')
parameters.put('windowSize', "7x7")
parameters.put('filterSizeX', 5)
parameters.put('filterSizeY', 5)
# definisci flood mask
global S1_floodMask_Spk
S1_floodMask_Spk = snappy.GPF.createProduct('Speckle-Filter', parameters,
S1_floodMask)
print('--- %.2f seconds ---' % (time.time() - start_time), flush=True)

# output
if plotResults:
    print('8. Plot:          ', end="", flush=True)
    start_time = time.time()
    for i in range(S1_TC.getNumBands()):
        plotBand(S1_TC.getBandAt(i), thresholds[i], method[i])
    print('--- %.2f seconds ---' % (time.time() - start_time), flush=True)

# polarizzazioni richieste e relativo output
if polarizations == 'both':
    sourceBands = 'Amplitude_VH,Intensity_VH,Amplitude_VV,Intensity_VV'
    output_extensions = 'processed_VHVV'
elif polarizations == 'VH':
    sourceBands = 'Amplitude_VH,Intensity_VH'
    output_extensions = 'processed_VH'
elif polarizations == 'VV':
    sourceBands = 'Amplitude_VV,Intensity_VV'
    output_extensions = 'processed_VV'

# cartella del file S1 .zip
input_path = os.path.join(directory, input_folder)
files = []
# aggiungi file alla lista
for file in glob.glob1(input_path, '*.zip'):
    files.append(file)
# seleziona file input e inizia il processo se c'è solo un file
if len(files) == 1:
    input_name = files[0]
    print('Selected: %s\n' % input_name, flush=True)
    # applica subset secondo dati JSON
    getScene('%s/%s' % (input_path, input_name))
# prodotto sentinel non presente
else:
    print('Prodotto Sentinel_1 non presente in "input", caricane uno.', flush=True)

```

## Esportazione dei dati

La quarta fase del codice è quella di esportazione dei dati. Nella cartella di output vengono salvati in formato geoTiff le immagini elaborate dopo le operazioni di filtraggio e correzione assieme alla maschera *FloodMask* che rappresenta l'inondazione calcolata. Quest'ultima immagine è convertita anche nei formati Shapefile e KML per permettere un'immediata visualizzazione dei risultati.

```
print('Exporting...\n', flush=True)
# controlla cartella di uscita, se non presente creala
output_path = os.path.join(directory, output)
if not os.path.isdir(output_path):
    os.mkdir(output_path)
GeoTIFF_path = os.path.join(output_path, 'GeoTIFF')
if not os.path.isdir(GeoTIFF_path):
    os.mkdir(GeoTIFF_path)
SHP_path = os.path.join(output_path, 'SHP')
if not os.path.isdir(SHP_path):
    os.mkdir(SHP_path)
KML_path = os.path.join(output_path, 'KML')
if not os.path.isdir(KML_path):
    os.mkdir(KML_path)
FloodMask_path = os.path.join(output_path, 'FloodMask')
if not os.path.isdir(FloodMask_path):
    os.mkdir(FloodMask_path)

# output GeoTIFF
print('1. Spk_db_GeoTIFF:          ', end="", flush=True)
start_time = time.time()
snappy.ProductIO.writeProduct(S1_Spk_db, '%s/%s_%s_S1_Spk_db' % (GeoTIFF_path,
os.path.splitext(input_name)[0], output_extensions), 'GeoTIFF')
print('--- %.2f seconds ---' % (time.time() - start_time), flush=True)
#
print('2. TC_GeoTIFF:              ', end="", flush=True)
start_time = time.time()
snappy.ProductIO.writeProduct(S1_TC, '%s/%s_%s_S1_TC' % (GeoTIFF_path,
os.path.splitext(input_name)[0], output_extensions), 'GeoTIFF')
print('--- %.2f seconds ---' % (time.time() - start_time), flush=True)
#
print('3. floodMask_GeoTIFF:        ', end="", flush=True)
start_time = time.time()
snappy.ProductIO.writeProduct(S1_floodMask, '%s/%s_%s_S1_floodMask' %
(GeoTIFF_path, os.path.splitext(input_name)[0], output_extensions), 'GeoTIFF')
print('--- %.2f seconds ---' % (time.time() - start_time), flush=True)
#
print('4. GlobCover_GeoTIFF:        ', end="", flush=True)
```

```

start_time = time.time()
snappy.ProductIO.writeProduct(GlobCover, '%s/%s_%s_S1_GlobCover' %
(GeoTIFF_path, os.path.splitext(input_name)[0], output_extensions), 'GeoTIFF')
print('--- %.2f seconds ---' % (time.time() - start_time), flush=True)
#
print('5. FloodMask_Spk:          ', end="", flush=True)
start_time = time.time()
snappy.ProductIO.writeProduct(S1_floodMask_Spk, '%s/%s_%s_S1_floodMask_Spk' %
(FloodMask_path, os.path.splitext(input_name)[0], output_extensions), 'GeoTIFF')
print('--- %.2f seconds ---' % (time.time() - start_time), flush=True)

# GeoTIFF to SHP
print('6. SHP:                    ', end="", flush=True)
start_time = time.time()
gdal.UseExceptions()
open_image = gdal.Open('%s/%s_%s_S1_floodMask_Spk.tif' % (FloodMask_path,
os.path.splitext(input_name)[0], output_extensions))
srs = osr.SpatialReference()
srs.ImportFromWkt(open_image.GetProjectionRef())
shp_driver = ogr.GetDriverByName('ESRI Shapefile')
output_shp = [" for i in range(open_image.RasterCount)]
if open_image.RasterCount == 1:
    output_shp[0] = '%s/%s_processed_%s' % (SHP_path, os.path.splitext(input_name)[0],
polarizations)
else:
    VH_SHP_path = os.path.join(SHP_path, 'VH')
    if not os.path.isdir(VH_SHP_path):
        os.mkdir(VH_SHP_path)
    VV_SHP_path = os.path.join(SHP_path, 'VV')
    if not os.path.isdir(VV_SHP_path):
        os.mkdir(VV_SHP_path)
    output_shp[0] = '%s/%s_processed_VH' % (VH_SHP_path,
os.path.splitext(input_name)[0])
    output_shp[1] = '%s/%s_processed_VV' % (VV_SHP_path,
os.path.splitext(input_name)[0])
for i in range(open_image.RasterCount):
    input_band = open_image.GetRasterBand(i+1)
    output_shapefile = shp_driver.CreateDataSource(output_shp[i] + '.shp')
    new_shapefile = output_shapefile.CreateLayer(output_shp[i], srs=srs)
    new_shapefile.CreateField(ogr.FieldDefn('DN', ogr.OFTInteger))
    gdal.Polygonize(input_band, input_band.GetMaskBand(), new_shapefile, 0, [],
callback=None)
    new_shapefile.SetAttributeFilter('DN != 1')
    for feat in new_shapefile:
        new_shapefile.DeleteFeature(feat.GetFID())

```

```

    new_shapefile.SyncToDisk()
print('--- %.2f seconds ---' % (time.time() - start_time), flush=True)

# SHP to KML
print('7. KML: ', end="", flush=True)
start_time = time.time()
if open_image.RasterCount == 1:
    shp_file = gdal.OpenEx('%s/%s_processed_%s.shp' % (SHP_path,
os.path.splitext(input_name)[0], polarizations))
    ds = gdal.VectorTranslate('%s/%s_processed_%s.kml' % (KML_path,
os.path.splitext(input_name)[0], polarizations), shp_file, format='KML')
    del ds
else:
    shp_file_VH = gdal.OpenEx('%s/%s_processed_VH.shp' % (VH_SHP_path,
os.path.splitext(input_name)[0]))
    ds_VH = gdal.VectorTranslate('%s/%s_processed_VH.kml' % (KML_path,
os.path.splitext(input_name)[0]), shp_file_VH, format='KML')
    del ds_VH
    shp_file_VV = gdal.OpenEx('%s/%s_processed_VV.shp' % (VV_SHP_path,
os.path.splitext(input_name)[0]))
    ds_VV = gdal.VectorTranslate('%s/%s_processed_VV.kml' % (KML_path,
os.path.splitext(input_name)[0]), shp_file_VV, format='KML')
    del ds_VV
print('--- %.2f seconds ---' % (time.time() - start_time), flush=True)
print('Files successfully stored under %s.\n' % output_path, flush=True)

```

## CAPITOLO 4

Il codice esposto nel capitolo precedente può avere numerose applicazioni nel campo dello studio del dissesto idrogeologico. I satelliti *SAR* Sentinel-1 garantiscono acquisizioni quasi giornaliere alle nostre latitudini ed è per questo che vengono usati per il tracciamento di oggetti come navi o il monitoraggio ambientale anche in caso di eventi catastrofici come alluvioni e inondazioni. In eventi di questo tipo, per i soccorritori, può essere di aiuto un sistema di facile applicazione che, rapidamente, sia in grado di fornire informazioni circa l'estensione e i danni causati. Per questa elaborazione si è scelto di utilizzare e processare le immagini Sentinel per creare mappe di inondazione nel più breve tempo possibile rispetto l'evento catastrofico. Si ritiene molto importante, infatti, la possibilità di avere entro poche ore dall'evento una cartina che sia in grado di definire l'area interessata e metterla a disposizione delle autorità competenti per le opportune valutazioni e che sia di supporto al personale sul campo per eventuali ricerche ed analisi.

Il sistema è stato sviluppato per essere il più possibile preciso, automatico e di veloce attuazione: si è scelto di implementare in Python il codice necessario all'elaborazione delle immagini fino all'esportazione finale di una maschera che rappresenti l'area inondata.

Per questo studio verranno utilizzati prodotti GRD High Resolution in modalità IW e doppia polarizzazione VV+VH, la dimensione del pixel è di 10x10 metri e la risoluzione spaziale di 20x22 metri.

Di seguito saranno analizzati due particolari eventi che hanno colpito il centro Italia, nella zona di Rieti, Lazio e l'area di Fermoy, cittadina a sud dell'Irlanda evidenziando i risultati determinati dalla scelta del metodo di binarizzazione.

### **4.1 Applicazione di immagini Sentinel-1 per lo studio del dissesto idrogeologico nella Piana Reatina, Italia**

La Piana Reatina nei giorni del 26 e 27 Gennaio 2021 è stata colpita da intense precipitazioni che hanno portato ad esondazioni dei fiumi Velino e Turano e al conseguente allagamento di alcune aree.

Il satellite Sentinel-1A ha acquisito un'immagine della zona in oggetto il 27 alle ore 05:12 e si è provveduto a scaricare il relativo file attraverso il "Copernicus Open Access Hub". Il prodotto è di tipo IW-GRDH doppia polarizzazione VV+VH.



*Figura 20 Quicklook del prodotto Sentinel-1*

L'immagine prodotta dal Satellite Sentinel-1A (Figura 20) ricopre un'area molto vasta che ai fini dello studio occorre limitare.

Una prima analisi può essere effettuata con il software SNAP e qui sotto si riporta la vista corrispondente all'area di interesse della banda di ampiezza in polarizzazione VV.

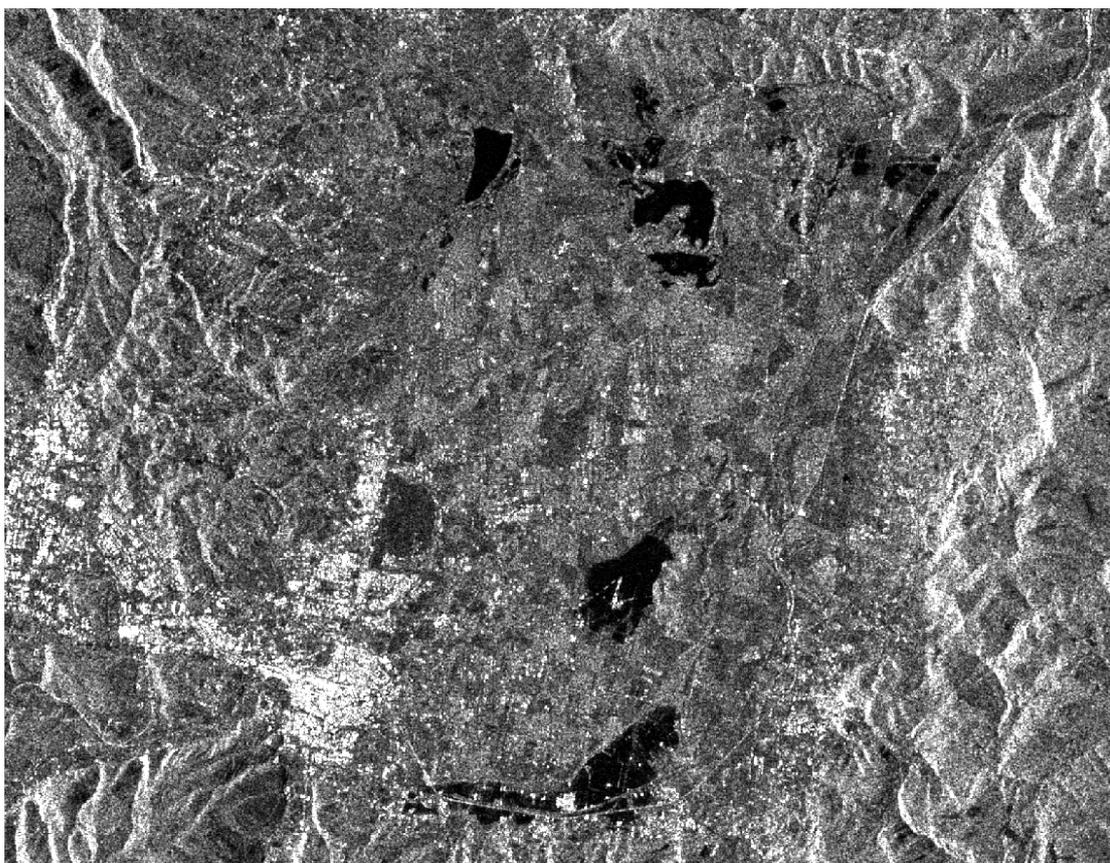


Figura 21 Banda di ampiezza VV dell'area di studio

In Figura 21 si possono notare delle aree più scure e trattandosi di un'immagine *SAR* in base a quanto descritto nel Capitolo 3, denotano un basso valore del coefficiente di *backscatter* probabilmente associabile alla presenza di acqua nel suolo.

Il codice necessita di un file JSON dell'area di interesse ed è possibile produrlo grazie ai numerosi e gratuiti servizi disponibili sul web.

Nella cartella "input" viene quindi inserito il file .zip scaricato dai server Copernicus e il file .json relativo all'area di interesse da analizzare.

Il codice è avviato grazie all'ambiente di sviluppo PyCharm e all'applicativo Jupyter in un pc notebook con processore i7 e 16GB di ram.

Si sceglie di processare le immagini *SAR* sia in polarizzazione VV che VH e vedremo che anche in base alla polarizzazione dell'immagine i risultati saranno differenti, si consiglia perciò di abilitare nel codice il processamento di entrambe le polarizzazioni.

Il codice viene inizializzato e come prima interazione con l'utente viene chiesto di impostare le cartelle di input e output, come mostrato in Figura 22.

Inserisci il percorso della cartella di input con i file .ZIP e .json

Inserisci il percorso della cartella di output

Figura 22 Impostazione delle cartelle di input e di output

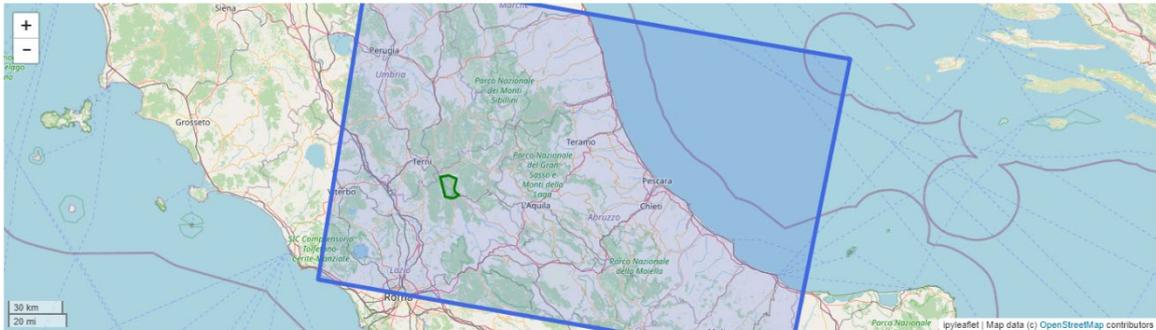
Il codice prosegue con la fase di processamento e vengono mostrate informazioni generali legate al file caricato e in una cartina interattiva ne è individuata la collocazione nello spazio; il perimetro verde delinea l'area di interesse.

L'elenco numerato tiene traccia dei processi eseguiti con il relativo tempo impiegato e durante il processamento alla fase di binarizzazione viene chiesto all'utente di scegliere la modalità di calcolo della soglia fra le due predefinite possibilità illustrate al Capitolo 3: metodo Otsu o metodo Minimo.

L'utente può anche inserire manualmente un valore di soglia che il codice utilizzerà per la binarizzazione, come si può vedere dalla Figura 23.

Selected: S1A\_IW\_GRDH\_1SDV\_20210127T051146\_20210127T051211\_036319\_0442F1\_EF15.zip

Name	Sensing Mode	Product Type	Polarization	Height	Width	Available Bands
0 S1A_IW_GRDH_1SDV_20210127T051146_20210127T051211_036319_0442F1_EF15.zip	IW	GRDH	1SDV	16670	26494	[Amplitude_VH, Intensity_VH, Amplitude_VV, Int...



Subset successfully generated.

1. Apply Orbit File: --- 0.42 seconds ---
2. Thermal Noise Removal: --- 0.05 seconds ---
3. Radiometric Calibration: --- 0.08 seconds ---
4. Speckle Filtering: --- 0.03 seconds ---
5. Terrain Connection: --- 0.34 seconds ---
6. Binarization:

Scegli tra "otsu" e "minimum" o immetti un valore:

Figura 23 Processamento dell'immagine

Una volta inserito il metodo scelto, il codice prosegue nel processamento e al termine viene mostrata l'immagine opportunamente processata e il relativo grafico con l'indicazione della soglia calcolata. In Figura 24 sono mostrati l'immagine e l'istogramma processati automaticamente con metodo Minimo e polarizzazione VV.

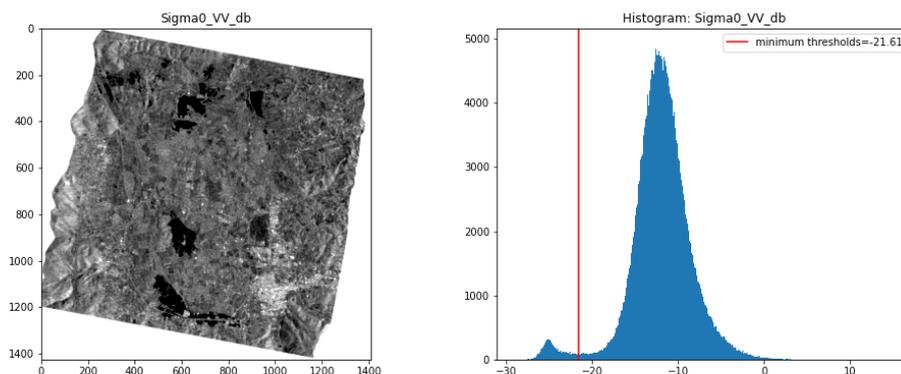


Figura 24 Immagine VV processata con il metodo Minimo e istogramma con soglia

Il processo si conclude con la quarta fase, quella di esportazione: un elenco numerato ne mostra il progresso e la cartella di salvataggio locale come si può vedere dalla Figura 25.

```
Exporting...

1. Spk_db_GeoTIFF:           --- 0.55 seconds ---
2. TC_GeoTIFF:              --- 1.28 seconds ---
3. floodMask_GeoTIFF:       --- 0.63 seconds ---
4. GlobCover_GeoTIFF:       --- 2.08 seconds ---
5. FloodMask_Spk:           --- 3.15 seconds ---
6. SHP:                      --- 0.25 seconds ---
7. KML:                      --- 0.08 seconds ---
Files successfully stored under C:\Users\output.
```

Figura 25 Processo di esportazione dei dati

### 4.1.1 Analisi dei risultati

I file esportati possono essere facilmente aperti ed analizzati: il file KML grazie all'applicazione per computer "Google Earth" [23] e la maschera dell'inondazione ottenuta è sovrapposta a quella del globo, come mostrato nella figura seguente.

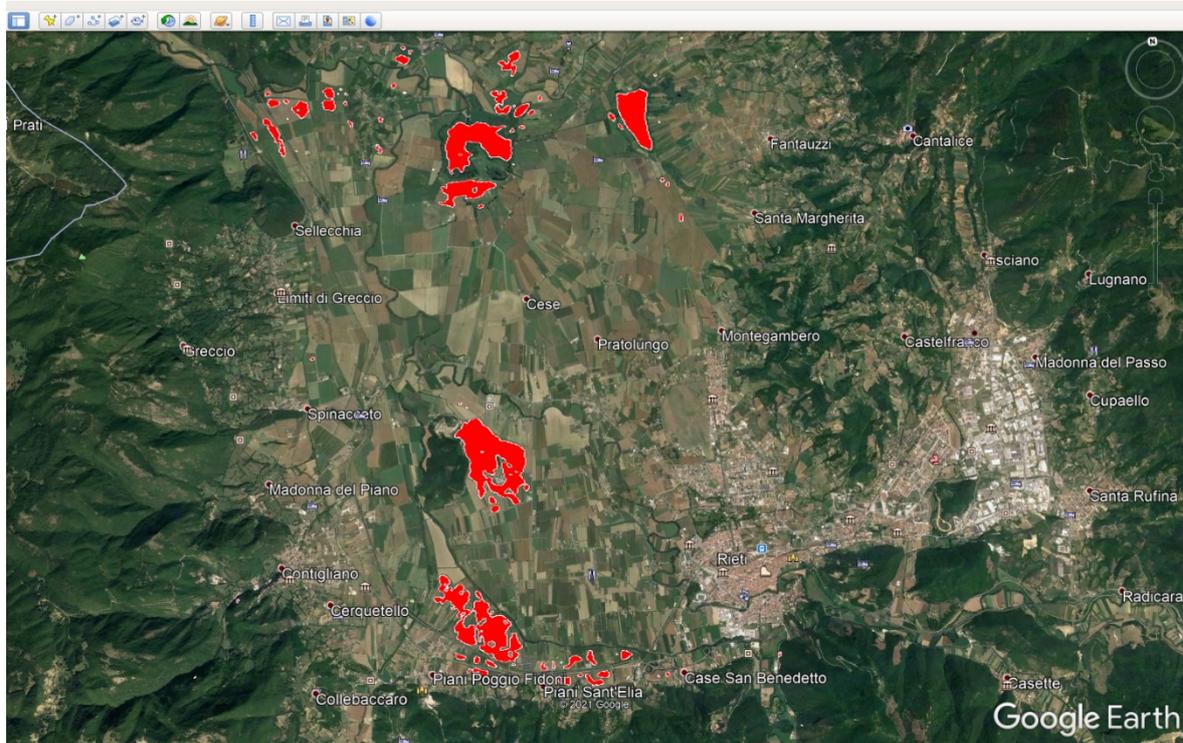


Figura 26 Maschera dell'inondazione a Rieti ricavata con metodo Minimo, polarizzazione VV

In Figura 26 le zone rosse evidenziano le aree che, a seguito del processamento dell'immagine iniziale, risultano con valori del coefficiente di *backscattering* sotto la soglia calcolata dinamicamente e quindi con potenziale presenza di acqua. A nord vengono rilevati e segnalati anche i due bacini naturali presenti in quell'area.

I risultati visti sin qui erano riferiti all'immagine in polarizzazione VV; si riportano anche i risultati relativi al processamento dell'immagine in polarizzazione VH.

In Figura 27 è riportata l'immagine VH processata con metodo Minimo e il suo istogramma dei valori di *backscatter*. In questo caso la polarizzazione VH non si presta ad un'analisi di questo tipo, il risultato ottenuto non è utilizzabile, infatti l'immagine di partenza è molto scura, i dati nel grafico sono concentrati nella parte sinistra e non si è in grado di calcolare una buona soglia di binarizzazione.

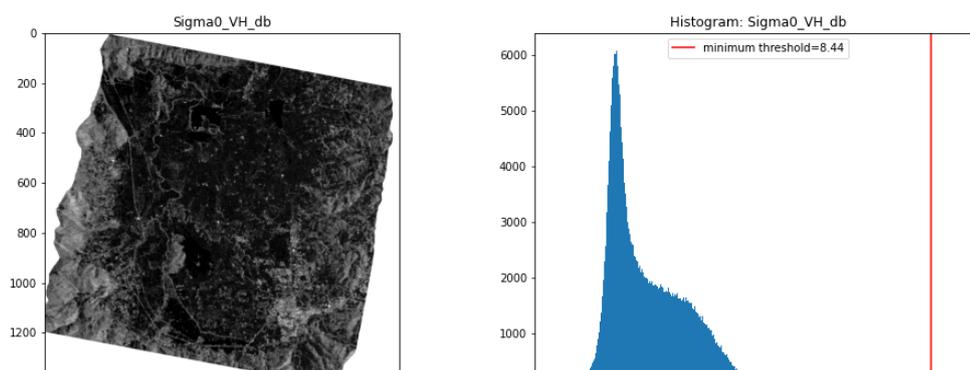


Figura 27 Immagine VH processata con metodo Minimo e istogramma con soglia

### Metodo Otsu e Metodo Minimo a confronto

Nell'analisi precedente il metodo scelto per la binarizzazione dell'immagine è stato il Metodo Minimo. La scelta dello stesso è dettata dalle caratteristiche dell'immagine da analizzare: la binarizzazione mediante il metodo Otsu da buoni risultati solo se l'istogramma è di tipo bimodale, cioè quando mostra due picchi ben definiti. Questa scelta non sarebbe quindi corretta quando l'area da analizzare ha un territorio troppo omogeneo. In Figura 28 si riporta l'istogramma con la soglia calcolata con il metodo Otsu della stessa area dell'analisi precedente.

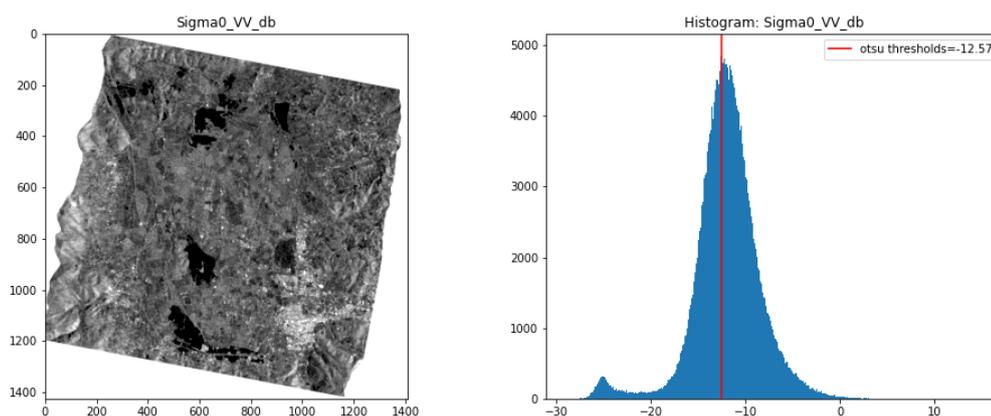


Figura 28 Immagine VV processata con il metodo Otsu e istogramma

Dall'istogramma si nota che la soglia calcolata è differente da quella ottenuta con l'altro metodo e la maschera dell'inondazione ottenuta è mostrata in Figura 29, si riporta anche il risultato della precedente analisi per un confronto fianco a fianco.

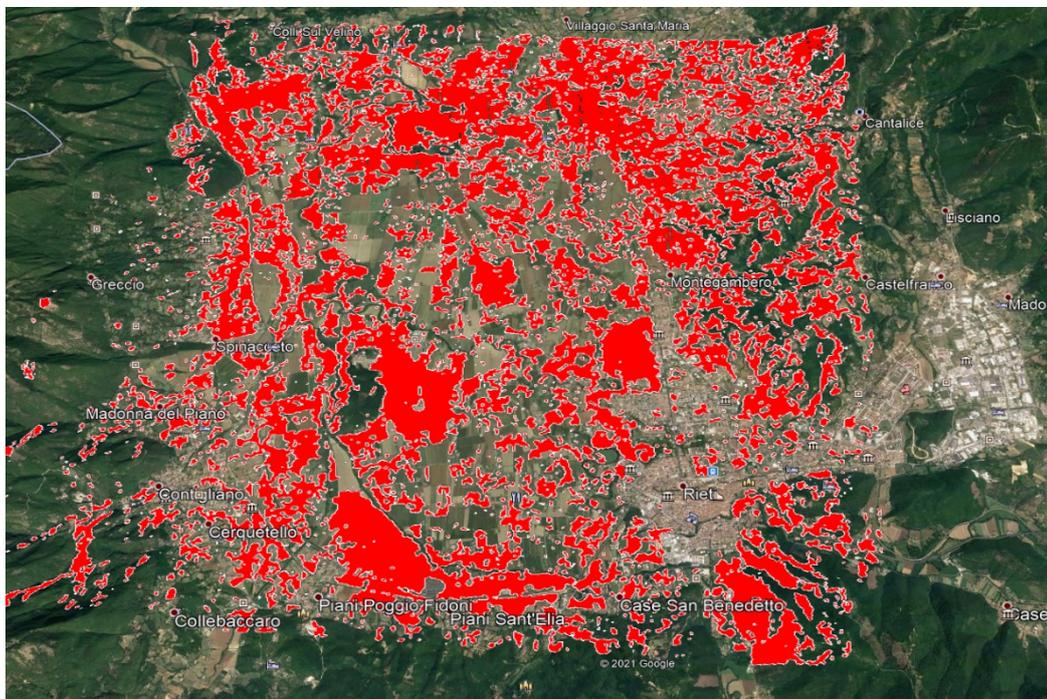


Figura 29 Maschera dell'inondazione in Google Earth con metodo Otsu

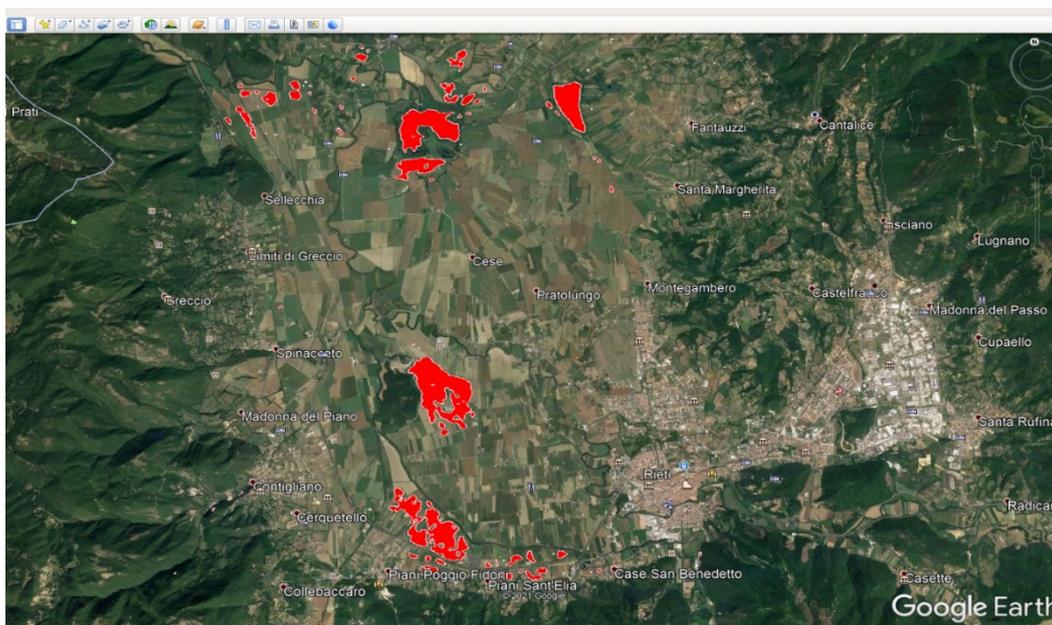


Figura 30 Maschera dell'inondazione in Google Earth con metodo Minimo

La maschera dell'inondazione in Figura 29 è completamente diversa da quella ottenuta con il metodo Minimo: la binarizzazione automatica con metodo Otsu ha calcolato una soglia

con valore troppo elevato classificando molte zone dell'area di interesse come "probabilmente inondate" risultando quindi poco credibile ed inutilizzabile.

### **Confronto con foto aeree in-situ**

I Vigili del Fuoco sono intervenuti per portare soccorso alla popolazione rimasta colpita dall'esondazione, in più punti della Piana Reatina, dei fiumi Velino e Turano dovuta all'imperversare del maltempo. Di seguito si riportano alcune foto scattate il 26/01 dall'elicottero dai Vigili del Fuoco e utilizzate per valutare i danni. Queste immagini in-situ vengono confrontate con la maschera dell'inondazione ottenuta automaticamente da immagini satellitari in polarizzazione VV e metodo Minimo come visto precedentemente.



Figura 31 Confronto in dettaglio tra foto aeree in-situ e maschera dell'inondazione ricavata con Metodo Minimo



Figura 32 Confronto in dettaglio tra foto aeree in-situ e maschera dell'inondazione ricavata con Metodo Minimo



Figura 33 Confronto in dettaglio tra foto aeree in-situ e maschera dell'inondazione ricavata con Metodo Minimo

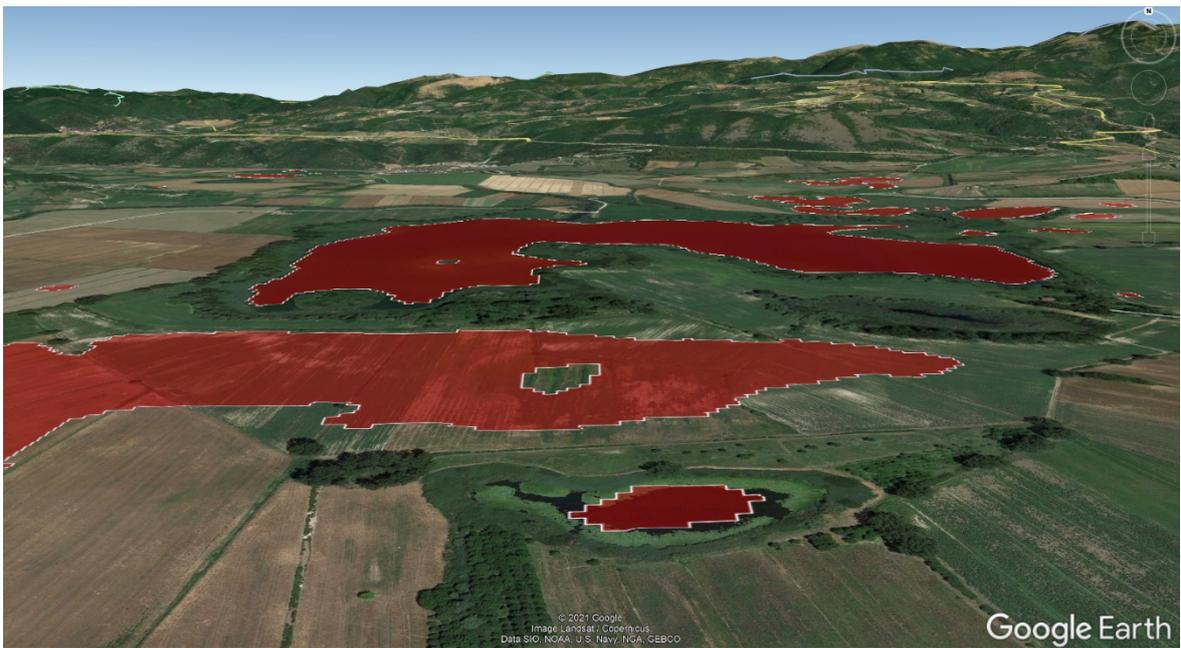


Figura 34 Confronto in dettaglio tra foto aeree in-situ e maschera dell'inondazione ricavata con Metodo Minimo

Dalle figure precedenti si può notare come la maschera dell'inondazione calcolata con Metodo Minimo mostri le zone inondate sottostimandone l'estensione se confrontata con le foto aeree. Nel confronto va tenuto conto che le foto satellitari risalgono alla giornata successiva degli scatti effettuati dai Vigili del Fuoco ed è ragionevole pensare che parte dell'acqua sia drenata e di fatto si è calcolata un'area inondata minore che potrebbe quindi non essere conseguenza di sola sottostima.

Il valore della soglia di binarizzazione può essere corretto manualmente ma si rischierebbe di ottenere numerosi falsi positivi.

## Risultati del Servizio per le Emergenze di Copernicus

La Protezione Civile il 27/01 ha richiesto l'attivazione del Servizio per le Emergenze di Copernicus [24] che il giorno seguente ha rilasciato i primi documenti di supporto all'emergenza. In Figura 35 la visione d'insieme del prodotto elaborato della loro analisi che nel dettaglio risulta in linea con quanto ricavato dalla binarizzazione con Metodo Minimo.

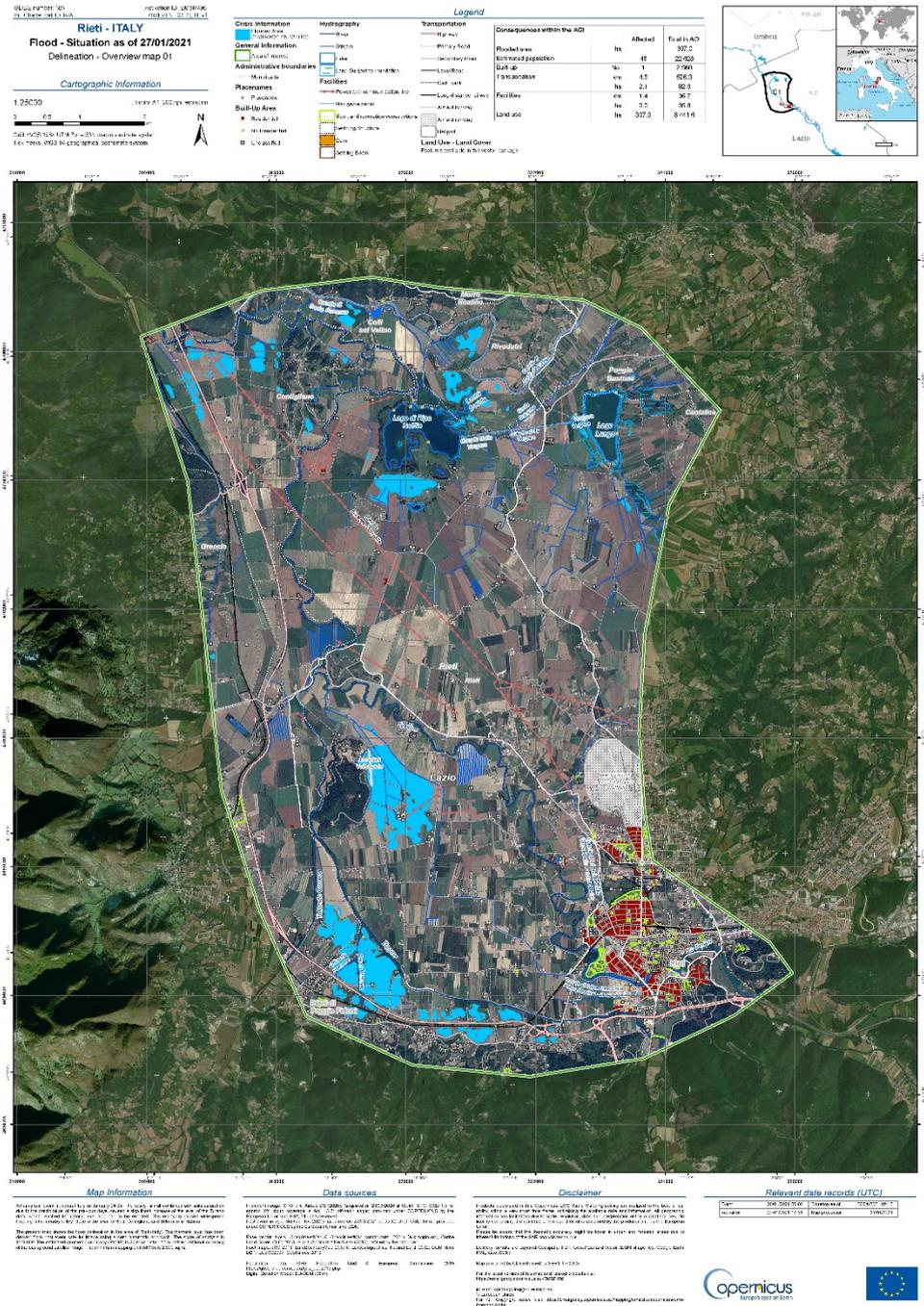


Figura 35 Cartina dell'inondazione a Rieti realizzata dal Servizio per le Emergenze di Copernicus

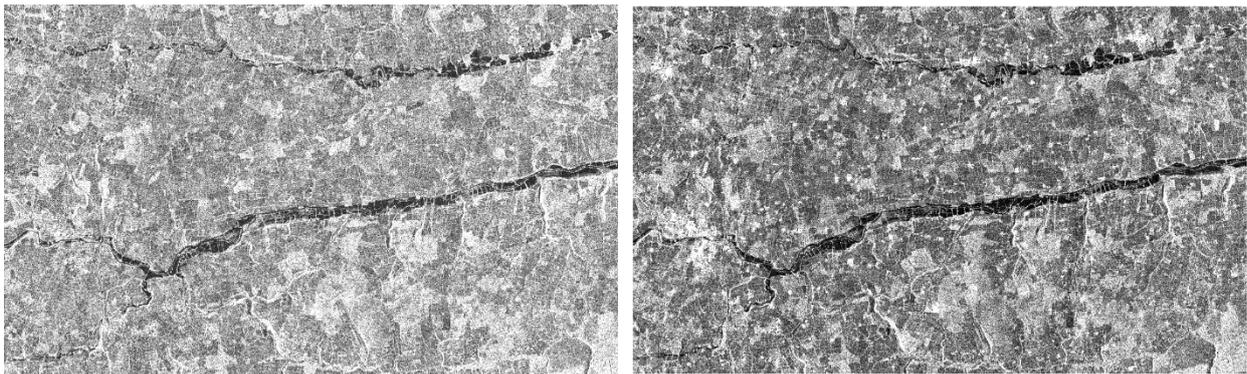
## 4.2 Applicazione di immagini Sentinel-1 per lo studio del dissesto idrogeologico a Fermoy, Irlanda

Il 23 Febbraio 2021 il culmine di una perturbazione di bassa pressione ha colpito l'Irlanda del sud provocando l'esonazione di vari corsi d'acqua sul territorio.

In particolare, il fiume "Blackwater" ha rotto gli argini in più punti del suo percorso interessando varie cittadine. Di seguito si analizza l'evento applicando quanto effettuato in precedenza.

Il prodotto scaricato dal sito Copernicus e utilizzato per l'analisi proviene dal satellite Sentinel-1A e l'immagine SAR è stata acquisita alle 18:23 del 23 Febbraio.

Una prima analisi è stata effettuata grazie all'utilizzo di Snap. In Figura 36 vengono messe a confronto le zone interessate sia con immagini in polarizzazione VV sia VH.



*Figura 36 Dettaglio dell'area di Fermoy a sinistra in polarizzazione VH e VV a destra*

Dalla figura appaiono evidenti delle aree scure attorno il naturale percorso del fiume e indicano una probabile esondazione. L'analisi viene effettuata per entrambe le polarizzazioni e nelle figure seguenti si riportano gli output del codice.

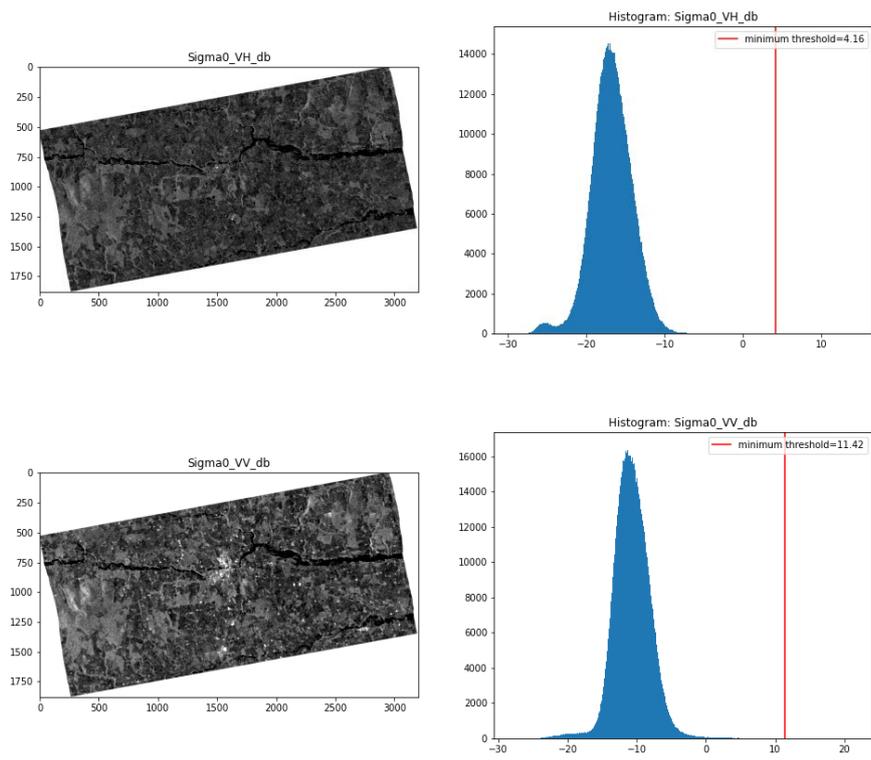


Figura 37 Immagini VH e VV processate con metodo Minimo e istogramma

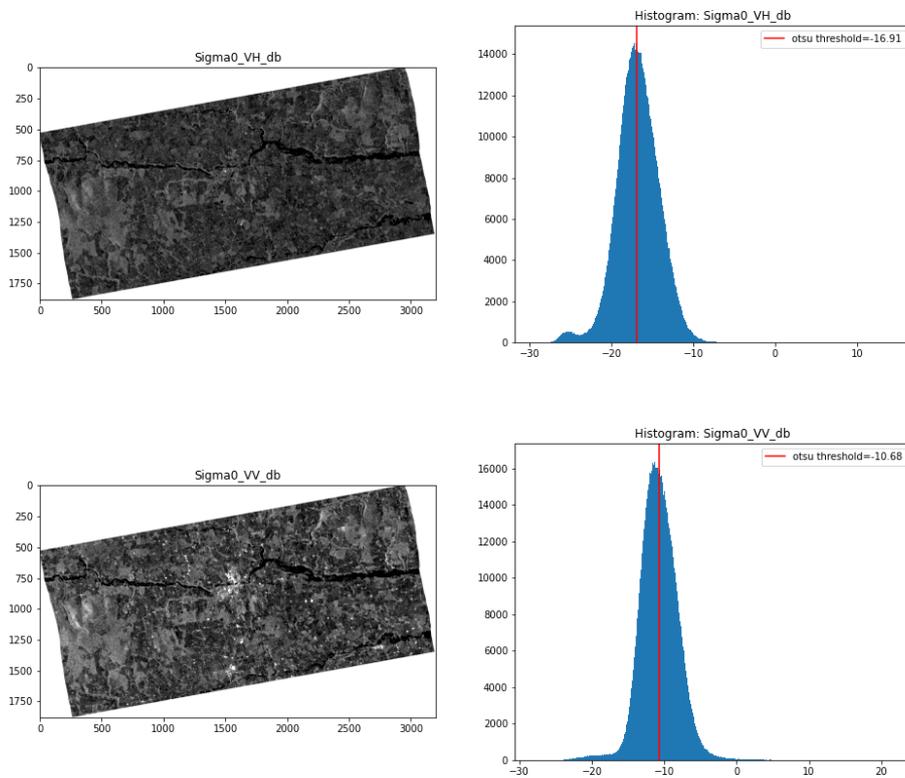
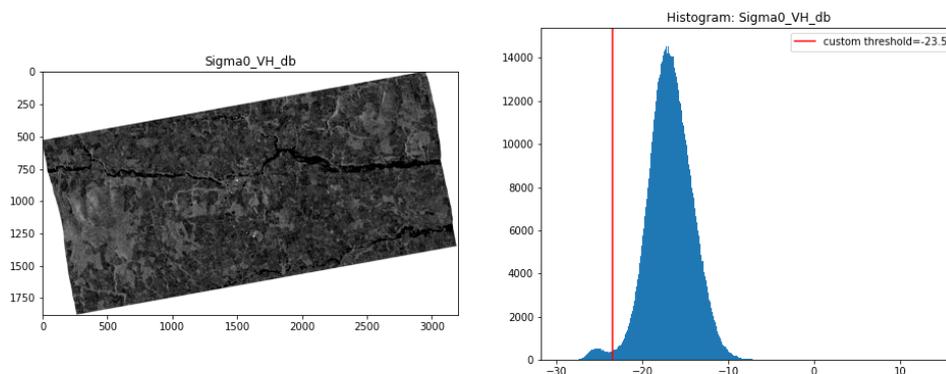


Figura 38 Immagini VH e VV processate con metodo Otsu

Per questo scenario, come emerge dalle Figura 37 e Figura 38, l'uso della binarizzazione automatica con metodo Otsu e Minimum non ha prodotto risultati soddisfacenti: la soglia calcolata in entrambi i casi è troppo elevata e non permetterebbe di separare l'acqua dal resto.

Una valutazione può comunque essere fatta inserendo manualmente nel programma la soglia di binarizzazione: l'istogramma dell'immagine VH presenta un minimo circa a -23.5dB e si usa proprio questo valore per avere dati dell'estensione dell'inondazione a Fermoy. In Figura 39 viene mostrato l'istogramma con la soglia impostata manualmente e la maschera dell'inondazione è mostrata nella figura seguente.



*Figura 39 Immagine VH processata con soglia impostata manualmente*

## 4.2.1 Analisi dei risultati

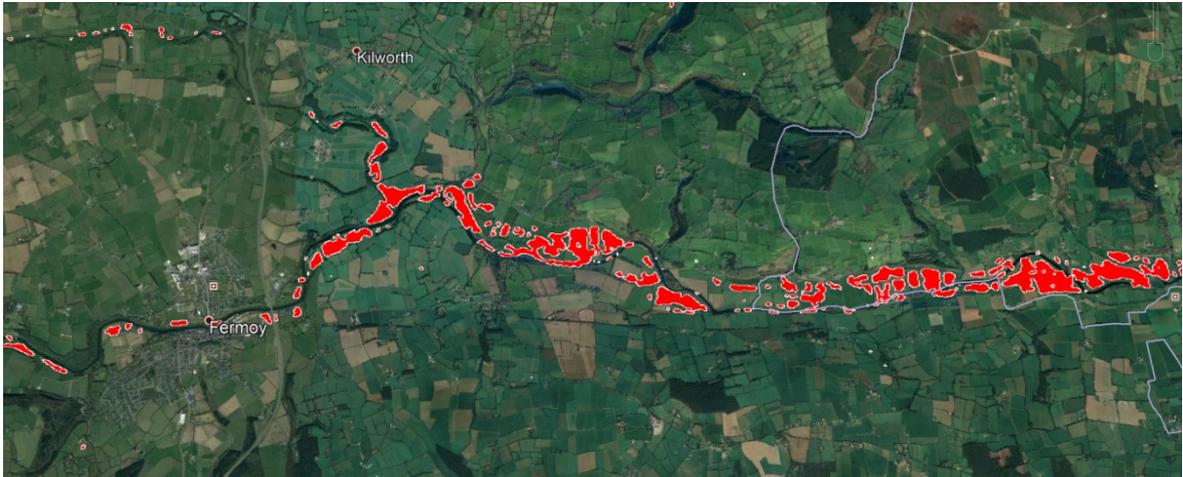


Figura 41 Maschera dell'inondazione a Fermoy con soglia impostata manualmente



Figura 40 Estratto dalla cartina dell'esondazione a Fermoy realizzata dal Servizio per le Emergenze di Copernicus

La soglia impostata a questo valore determina una rilevante esondazione lungo il percorso del fiume vicino Fermoy. La maschera dell'esondazione ottenuta può essere confrontata con quella ricavata dal Servizio per le Emergenze di Copernicus [25] e mostrata in Figura 40

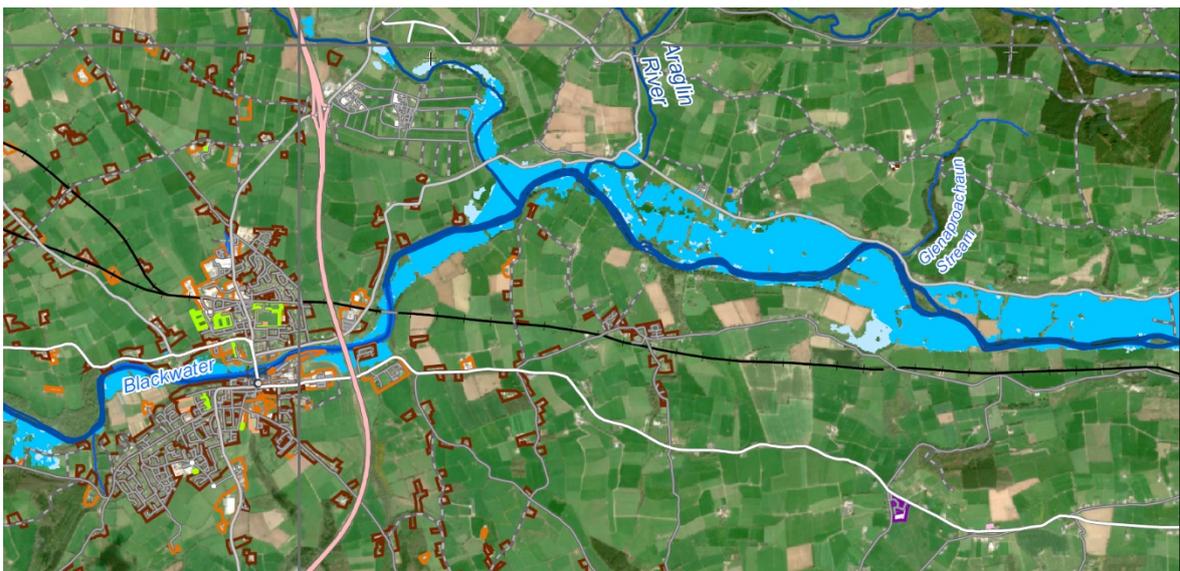


Figura 42 Esondazione a Fermoy in dettaglio

L'impostazione manuale della soglia viene fatta valutando il grafico del coefficiente di *backscatter* ricercando un minimo locale e facendo anche più tentativi; è una soluzione che permette quantomeno una prima analisi e di stimare l'entità dei danni. L'immagine in questo caso non si prestava ad essere elaborata con i metodi Otsu o Minimo probabilmente per la troppa omogeneità del terreno e ad un istogramma che mostrava un solo picco.

# CONCLUSIONI

L'analisi dimostra che Sentinel-1 ha le potenzialità per essere usato nel rilevamento di inondazioni ed esondazioni e in generale eventi caratterizzati da una presenza continua di nubi che escludono la possibilità di impiegare altri tipi di satelliti con sensori ottici.

È importante sottolineare che le immagini utilizzabili con questo metodo provengono da due satelliti in orbita che non coprono h24 l'intera superficie terrestre, potrebbero infatti non fornire dati circa l'area interessata o renderli disponibili con un non trascurabile ritardo rispetto l'evento da analizzare risultando quindi di poco aiuto.

Lo scopo era quello di generare in breve tempo dati quantitativi circa il dissesto idrogeologico manifestatosi in seguito ad eventi eccezionali come intense precipitazioni e testando più metodologie. L'identificazione della soglia di binarizzazione e quindi della maschera di inondazione è legata alla scelta dei metodi di analisi: il processo può essere automatizzato ma il controllo di un operatore è auspicabile data la sensibilità del dato fornito. Come abbiamo visto, i risultati dell'elaborazione sono influenzati oltre che dalla scelta del metodo di binarizzazione anche dalla polarizzazione dell'immagine di partenza; sono variabili che vanno ben ponderate e la peculiarità del codice è anche quella di poter essere agevolmente modificato e migliorato per far fronte a nuove esigenze e situazioni. Entrambi i metodi di binarizzazione proposti, Otsu e Minimo, si dimostrano efficaci in alcune situazioni ma anche inadeguati in altre fornendo risultati molto diversi tra loro. I metodi di binarizzazione automatici Otsu e Minimo garantiscono risultati migliori nel caso di immagini con istogramma di tipo bimodale. L'impostazione di una soglia di binarizzazione ricavata manualmente può essere utile nei casi più complessi dove non è così evidente la separazione tra zone coperte da acqua e la terra. Sarebbe importante quindi creare un algoritmo in grado di scegliere, a seconda dell'immagine sotto analisi, il metodo di binarizzazione più adeguato, visto che i due metodi hanno dei punti di forza e debolezza al variare di alcune caratteristiche dell'immagine.

La discriminazione della terra ferma dalle zone con presenza di acqua è di fondamentale importanza in termini di accuratezza del risultato e l'assenza di uno storico ufficiale circa i danni causati da questo tipo di eventi non permette un efficace confronto tra dati sperimentali e dati sul campo. L'analisi proposta è un importante passo verso lo studio di questi fenomeni e il progetto può essere migliorato prevedendo analisi e validazioni sul campo post-evento.

Le immagini *SAR* si rivelano idonee e di supporto in applicazioni necessarie ai servizi di emergenza, ad Enti Governativi e non solo; l'espansione delle costellazioni di satelliti e della loro frequenza di acquisizione di immagini permetteranno di avere a disposizione sempre più dati e in tempi rapidi rispetto l'evento da analizzare.

# BIBLIOGRAFIA

- [1] C. I. M. B. B. L. Alessandro Trigila, «Dissesto Idrogeologico in Italia: pericolosità e indicatori di rischio - Edizione 2018,» Istituto Superiore per la Protezione e la Ricerca Ambientale (ISPRA), Roma, 2018.
- [2] «Copernicus Open Access Hub,» [Online]. Available: <https://scihub.copernicus.eu/dhus/#/home>.
- [3] «Protezione Civile Marche, Rischio Idrogeologico,» [Online]. Available: <https://www.regione.marche.it/Regione-Utile/Protezione-Civile/Previsione-e-Monitoraggio/Rischio-Idrogeologico>.
- [4] «Protezione civile - Procedure di Allertamento,» [Online]. Available: <https://www.regione.marche.it/Regione-Utile/Protezione-Civile/Previsione-e-Monitoraggio/Procedure-di-Allertamento>.
- [5] «Regione Marche - Rete di Monitoraggio Idrogeologico Regionale,» [Online]. Available: <https://www.regione.marche.it/Regione-Utile/Protezione-Civile/Progetti-e-Pubblicazioni/Meteo#Presentazione>.
- [6] «Informazioni su Copernicus,» [Online]. Available: <https://www.copernicus.eu/it/informazioni-su-copernicus>.
- [7] «Sentinel-1,» [Online]. Available: <https://sentinels.copernicus.eu/web/sentinel/missions/sentinel-1>.
- [8] «Scenario di osservazione,» [Online]. Available: <https://sentinels.copernicus.eu/web/sentinel/missions/sentinel-1/observation-scenario/acquisition-segments>.
- [9] «Sentinel-1, Strumento C-SAR,» [Online]. Available: <https://sentinel.esa.int/web/sentinel/technical-guides/sentinel-1-sar/sar-instrument>.
- [10] «Sentinel-1, Prodotti Disponibili,» [Online]. Available: <https://sentinel.esa.int/web/sentinel/user-guides/sentinel-1-sar/product-types-processing-levels>.
- [11] «Sentinel-2,» [Online]. Available: <https://sentinels.copernicus.eu/web/sentinel/missions/sentinel-2>.

- [12] «Sentinel-2, Strumento MSI,» [Online]. Available:  
<https://sentinel.esa.int/web/sentinel/technical-guides/sentinel-2-msi/msi-instrument>.
- [13] «Sentinel-2, Prodotti Disponibili,» [Online]. Available:  
<https://sentinel.esa.int/web/sentinel/missions/sentinel-2/data-products>.
- [14] «SNAP,» [Online]. Available: <https://step.esa.int/main/toolboxes/snap/>.
- [15] «Copernicus Emergency Management System,» [Online]. Available:  
<https://emergency.copernicus.eu/mapping/ems/service-overview>.
- [16] «ESA earthnet online,» [Online]. Available:  
<https://earth.esa.int/handbooks/asar/CNTR1-1-2.html>.
- [17] S. M. P. Kuntla, «Development of an Automated Tool for Delineation of Flood Footprints from SAR Imagery for Rapid Disaster Response: A Case Study,» *J Indian Soc Remote Sens*, vol. 48, n. 6, pp. 935-944, 2020.
- [18] N. Otsu, «A Threshold Selection Method from Gray-Level Histograms,» *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 9, n. 1, pp. 62-66, Jan. 1979.
- [19] «Threshold Otsu,» [Online]. Available: [https://scikit-image.org/docs/dev/api/skimage.filters.html#skimage.filters.threshold\\_otsu](https://scikit-image.org/docs/dev/api/skimage.filters.html#skimage.filters.threshold_otsu).
- [20] «Threshold Minimum,» [Online]. Available: [https://scikit-image.org/docs/dev/api/skimage.filters.html#skimage.filters.threshold\\_minimum](https://scikit-image.org/docs/dev/api/skimage.filters.html#skimage.filters.threshold_minimum).
- [21] «API Python Snappy,» [Online]. Available:  
<https://senbox.atlassian.net/wiki/spaces/SNAP/pages/19300362/How+to+use+the+S+NAP+API+from+Python>.
- [22] «RUS Webinars - Flood mapping with Sentinel-1,» [Online]. Available: <https://rus-copernicus.eu/portal/rus-webinar-flood-mapping/>.
- [23] «Google Earth,» [Online]. Available: <https://www.google.com/earth/>.
- [24] «EMS: Flood in Lazio, Rieti,» [Online]. Available:  
<https://emergency.copernicus.eu/mapping/list-of-components/EMSR496>.
- [25] «EMS: Flood in Southern Ireland,» [Online]. Available:  
<https://emergency.copernicus.eu/mapping/list-of-components/EMSR502>.