

UNIVERSITÀ DEGLI STUDI DI NAPOLI “FEDERICO II”
FACOLTÀ DI INGEGNERIA

DIPARTIMENTO DI SCIENZA
ED INGEGNERIA DELLO SPAZIO
“L. G. NAPOLITANO”

TESI DI
LAUREA IN
INGEGNERIA AEROSPAZIALE

**Ottimizzazione dell’algoritmo PGA
per radar ad apertura sintetica da aereo**

Relatore:

Ch.mo Prof. Antonio Moccia

Candidato:

Antonio Fortunato

Matr. 347/886

Correlatore:

Ing. Ph.D. Ciro Caramiello

ANNO ACCADEMICO 2005 - 2006

Dum loquimur invida aetas fugerit,
carpe diem, quam minimum credula postero.

Orazio, Odi, I, XI, 7-8

A Marcella

Indice

Introduzione	1
1 Il progetto MINISAR	4
1.1 La missione	4
1.2 Il sistema	6
1.2.1 Le caratteristiche generali	6
1.2.2 La piattaforma	9
1.2.3 L'hardware	11
1.2.4 Il software	14
1.3 La motion compensation	17
2 L'algoritmo PGA	18
2.1 Aspetti generali	18
2.2 Struttura	20
2.2.1 Centrazione	22
2.2.2 Finestratura	24
2.2.3 Stima	25
2.2.4 Correzione	26

2.3	Implementazione	27
2.3.1	L'implementazione originale	28
2.3.2	L'evoluzione	33
3	Risultati	41
3.1	I test	41
3.1.1	Aspetti generali	41
3.1.2	Il software di elaborazione	44
3.1.3	Valutazione dei risultati	45
3.2	TOPSAR Data Set	49
3.2.1	Il Vesuvio: Raw	50
3.2.2	Il Vesuvio: Range Perturbed	55
3.2.3	Il Vesuvio: Azimuth Perturbed	58
3.2.4	Somma Vesuviana: Raw	61
3.2.5	Somma Vesuviana: Range Perturbed	66
3.2.6	Somma Vesuviana: Azimuth Perturbed	69
3.3	MINISAR Data Set	72
3.3.1	Stabilimento industriale: Raw	73
3.3.2	Stabilimento industriale: Range Perturbed	78
3.3.3	Stabilimento industriale: Azimuth Perturbed	81
3.3.4	Il Voltorno: Raw	84
3.3.5	Il Voltorno: Range Perturbed	89
3.3.6	Il Voltorno: Azimuth Perturbed	92
3.4	Conclusioni	95

A	L'algoritmo PGA: le funzioni MATLAB	97
A.1	L'implementazione originale	97
A.2	L'evoluzione	100
B	Il pacchetto <i>minisar_proc</i>: le chiamate all'algoritmo PGA	106
B.1	Range	106
B.2	Azimuth	108
	Ringraziamenti	110

Elenco delle figure

1.1	Il P92-JS	9
1.2	L'hardware MINISAR	12
1.3	Il software MINISAR	15
2.1	Schema a blocchi del PGA	20
2.2	Immagine degradata da errore polinomiale	22
2.3	Centratura	23
2.4	Finestratura	24
2.5	Errore totale e criterio di convergenza	25
2.6	Immagine dopo la correzione	27
2.7	Funzione di riferimento	29
2.8	Finestra unitaria	30
2.9	Applicazione del PGA originale in azimuth e range	32
2.10	Peso computazionale della FFT	36
2.11	Finestra gaussiana	37
2.12	Dimensione della finestra in funzione delle iterazioni	38
2.13	Errore totale in funzione delle iterazioni	39

3.1	Errore polinomiale dell'ottavo ordine	42
3.2	Collocazione classica del PGA	43
3.3	Collocazione del PGA per MINISAR/TOPSAR	44
3.4	L'interfaccia grafica utente di SLCCompare	46
3.5	Il programma FlashTest	47

Elenco delle tabelle

1.1	Le caratteristiche del sistema MINISAR	7
1.2	Le caratteristiche del sistema MINISAR (cont.)	8
1.3	P92-JS: le caratteristiche tecniche	10

Introduzione

L'immagine radar è il risultato di un complesso processo di elaborazione dei segnali che traduce informazioni di distanza e velocità relativa in un prodotto di più immediata interpretazione. Alla base di questa trasformazione ci sono un certo numero di ipotesi il cui livello di soddisfacimento determina la qualità del risultato finale. In particolare, si suppone che il moto della piattaforma da cui il radar è operato sia rettilineo ed uniforme. Il discostamento da questa condizione nominale può portare ad un notevole degrado dell'immagine e costituisce un problema particolarmente sentito per i radar ad apertura sintetica da aereo.

Le tecniche sviluppate nel corso degli anni per consentire la formazione di immagini radar ad alta risoluzione anche in presenza di perturbazioni sono numerose e globalmente conosciute come *motion compensation*.

Le soluzioni classiche prevedono l'uso di una piattaforma INS/GPS le cui letture sono usate per la correzione dei dati prima della focalizzazione. Questo metodo presenta il vantaggio di essere relativamente semplice ed intuitivo ma, per contro, affida la qualità del risultato all'accuratezza degli strumenti

di misura e, per sua stessa natura, non è in grado di rimuovere i termini di defocalizzazione legati a meccanismi diversi dalle perturbazioni del moto.

In alternativa, si possono usare tecniche che sfruttino unicamente i dati delle antenne. L'idea di base è rilevare gli effetti della defocalizzazione direttamente nell'immagine, senza formulare alcuna ipotesi sulle cause che li hanno determinati. I vantaggi di quest'approccio sono interessanti e consentono nel non dover disporre, in prima approssimazione, di una piattaforma INS/GPS, e nell'essere in grado di ripulire gli effetti legati a cause diverse dal moto.

Il più efficiente degli algoritmi di questa classe è il PGA (Phase Gradient Autofocus), una tecnica di correzione della focalizzazione ad ampio spettro in grado di operare anche su immagini prive di scatteratori forti ed isolati.

Il presente lavoro di tesi, sviluppato presso il CO.RI.S.T.A. (COnsorzio di RIcerca sui Sistemi di Telesensori Avanzati), ha avuto come obiettivo l'ottimizzazione dell'algoritmo PGA in relazione alle condizioni operative di MINISAR, un radar ad apertura sintetica da aereo progettato e realizzato presso lo stesso CO.RI.S.T.A. per il monitoraggio delle zone ad alto dissesto idrogeologico.

Principalmente, ci si è indirizzati verso l'analisi di tutti i parametri in grado di influenzare il problema della *motion compensation* ed, in quest'ottica, sono state passate in rassegna le caratteristiche di MINISAR e quelle del

suo software di elaborazione (Capitolo 1). L'algoritmo PGA è stato, quindi, studiato nel dettaglio con l'attenzione rivolta agli sviluppi proposti in letteratura negli ultimi anni (Capitolo 2). Sulla base delle informazioni raccolte è stata realizzata una nuova implementazione che fosse la più idonea all'operatività con piattaforme aeree di piccole dimensioni.

Numerosi test di focalizzazione sono stati dedicati alla definizione delle prestazioni del nuovo algoritmo. In questa fase, si è sviluppato un confronto operativo tra questa nuova implementazione e quella già presente nel software di elaborazione di MINISAR (Capitolo 3). I risultati sono stati più che soddisfacenti ed hanno mostrato un sensibile miglioramento delle prestazioni soprattutto in relazione alle scene rurali a basso numero di scatteratori prominenti.

Quest'evoluzione rappresenta un'interessante punto di partenza verso il processing in tempo reale di dati relativi a disastri naturali e verso lo sviluppo di sistemi che integrino le potenzialità del PGA con quelle delle tecniche classiche di *motion compensation*.

Capitolo 1

Il progetto MINISAR

1.1 La missione

Lo sconsiderato sfruttamento delle risorse naturali e l'edificazione incontrollata in zone inadatte agli insediamenti urbani ha generato nel corso degli anni una pesante alterazione degli equilibri geologici. Le proporzioni di questo fenomeno sono tali da rendere frane ed alluvioni tristemente ricorrenti in numerose regioni italiane: la cronaca giornalistica non manca ogni anno di sottolineare gli enormi costi economici ed umani di un tale stato di cose [1].

La risposta più efficace a questi problemi è in provvedimenti che impediscano sul nascere il sopraggiungere di condizioni di rischio, nonchè in una continua ed accurata sorveglianza delle zone in cui il pericolo è già manifesto. Quest'ultima esigenza ha spinto negli ultimi anni allo sviluppo di sistemi aerospaziali per il telerilevamento ritagliati sulle particolari caratteristiche dei fenomeni in gioco.

In generale, affinché si percepisca un'effettiva utilità devono essere soddisfatti i seguenti vincoli:

- il sistema deve essere flessibile: i dati devono pervenire sia con cadenza regolare sia quando se ne ha specificamente bisogno;
- il prodotto finale deve presentare sia natura *qualitativa* che *quantitativa*;
- i dati di carattere quantitativo devono essere accurati: in alcuni casi si può avere la necessità di valutare spostamenti dell'ordine dei centimetri;
- il tempo meteorologico non deve pesare eccessivamente sul funzionamento del sistema: presumibilmente è proprio in condizioni di pioggia insistente che si ha maggiore bisogno di informazioni sulle zone a rischio;
- l'effettiva utilizzabilità su larga scala della soluzione dipende dalla sua economicità, intesa sia come spesa iniziale che come costo di esercizio;

In virtù di queste condizioni, possono essere sviluppate alcune considerazioni. In primo luogo, le piattaforme spaziali devono essere escluse a favore di quelle aeronautiche. I vincoli imposti dalla meccanica orbitale, infatti, pregiudicano le esigenze di flessibilità ed i costi, contemporaneamente, determinano un profilo economico decisamente poco favorevole. Dal punto di vista della modalità di acquisizione dei dati, i sistemi nelle microonde si presentano certamente più adeguati rispetto a quelli elettroottici, troppo pesantemente legati alle condizioni di illuminazione e visibilità della scena. Per

di più, nell'ambito delle microonde il requisito di accuratezza può essere brillantemente soddisfatto con l'uso di soluzioni interferometriche [2].

In definitiva, il profilo ideale è quello di un radar ad apertura sintetica aviotrasportato, con antenne in configurazione bistatica.

Il progetto MINISAR - ideato e realizzato presso il centro di ricerca CORISTA - costituisce la prima risposta interamente italiana alla sfida della sorveglianza di zone ad intenso dissesto idrogeologico [3]. I concetti chiave che hanno ispirato lo sviluppo di questo sistema sono quelli indicati in precedenza e l'implementazione proposta costituisce un deciso passo in avanti verso strumenti di controllo economici, efficienti e facilmente gestibili a livello locale.

1.2 Il sistema

1.2.1 Le caratteristiche generali

Le tabelle 1.1 e 1.2 riportano i principali parametri del sistema MINISAR. E' significativo notare come siano previste due modalità operative, narrow e wide, che realizzano ampiezze di strisciata diverse con l'uso di una differente quota di volo. Allo scopo di mantenere rapporti segnale/rumore confrontabili, sono previste differenti lunghezze d'impulso, a parità di valore della potenza di picco.

	Narrow	Wide
Operative frequency	X-band - 9.65 MHz	
Polarization	Linear HH	
Transmitted bandwidth	4 x 70 MHz	
Sampling frequency	300 MHz	
Nominal aircraft velocity	70 m/s	100 m/s
PRF	210 Hz	300 Hz
Nominal off-nadir angle	45°	
Antenna elevation angle (3dB)	22°	
Antenna azimuth angle (3dB)	2.2°	
Nominal altitude	3000 m	5000 m
Nominal slant swath	3000 - 5365 m	6100 - 8900 m
Ground swath dimension	2200 m	3900 m

Tabella 1.1: Le caratteristiche del sistema MINISAR

Dal punto di vista delle prestazioni interferometriche, l'accuratezza nel valutare l'altezza del terreno è principalmente funzione delle componenti della baseline, degli angoli di assetto, dello slant range, della quota di volo e della differenza interferometrica di fase. Supponendo cause di errore indipendenti, ognuno dei parametri appena menzionati contribuisce all'incertezza totale con la propria varianza moltiplicata per la derivata dell'altezza rispetto al parametro. Quantitativamente, in narrow mode è possibile pervenire ad un'incertezza massima nell'ordine dei cinque metri, mentre in wide mode altri due metri devono essere aggiunti. In entrambi i casi l'accuratezza è compatibile con la realizzazione di carte topografiche standard in scala 1:25.000.

Dal momento che una baseline di 1.5 m risulta problematica a causa alle ridotte dimensioni della piattaforma, è stata implementata una modalità di trasmissione di tipo ping-pong. Questa configurazione prevede che ogni an-

	Narrow	Wide
Baseline length	1.5 m (physical 75 cm)	
Transmitted power	80 W	
Pulse duration	18 μs	30 μs
Range resolution (1 Look)	0.85 m	
Azimuth resolution (1 Look)	0.5 m	
Nominal interferometric resolution	0.85 x 1.5 m	2.5 x 3 m
	3 Look	18 Look
SNR	≥ 10 dB	
ASR	< 20 dB	
Signal dynamic	20 dB	
Number of bit per sample	8 bit	
Data rate	≤ 50 Mb/s	≤ 118 Mb/s
Data storage	7.3 Gb	12.2 Gb

Tabella 1.2: Le caratteristiche del sistema MINISAR (cont.)

tenna trasmetta e riceva un impulso alternativamente, così come accade per l'interferometria multi-pass da satellite. In questo modo, si riescono ad avere le medesime performance interferometriche con la metà della baseline.

Altra caratteristica peculiare del sistema è che la larghezza di banda di 280 MHz è raggiunta con l'uso di quattro chirp da 70 MHz ciascuno. Questi sono generati consecutivamente e, quindi, portati in banda X a quattro valori di frequenza leggermente diversi. Questa scelta è giustificata principalmente da due ragioni:

- si aggiunge flessibilità al sistema, consentendo una facile estensione della larghezza di banda;
- si possono utilizzare dispositivi per la generazione del chirp molto ac-

curati, capaci di garantire un alto grado di linearità della fase.

Questi vantaggi sono in parte compensati dalle maggiori accuratezze richieste per i sistemi di sincronizzazione.

1.2.2 La piattaforma

La piattaforma aerea scelta per il progetto MINISAR è il velivolo ultraleggero P92-JS delle industrie aeronautiche TECNAM.



Figura 1.1: Il P92-JS

Le caratteristiche di questo aeromobile, riassunte nella tabella 1.3, consentono di disporre durante le missioni di una notevole flessibilità e semplicità operativa. Si considerino, in particolare, i seguenti aspetti:

	P92-JS 80hp	P92-JS 100hp
DIMENSIONS		
Wing Span	9.30 m	
Length over-all	6.30 m	
Height over-all	2.45 m	
Width cabin	1.10 m	
Wing area	13.2 m^2	
Aspect ratio	6.55	
Dihedral angle	1.5°	
Tailplane span	2.90 m	
Wheel track	1.80 m	
Wheel tyres	5.00-5	
ENGINE AND PROPELLER		
Power max	Rotax 912A	Rotax 912S
Propeller	GT wooden two-blade	
Fuel	EURO SUPER	
Tanks	70 l	
WEIGHT		
Max T.O.	550 kg	
Useful Load	240 kg	
Wing loading	41.7 kg/m^2	
Basic empty weight	310 kg	
Load ultimate factor	+3.8 -1.5	
PERFORMANCE		
Max speed s/l	210 km/h	232 km/h
Cruising speed (75%)	185 km/h	
Rate of climb	4.35 m/s	5 m/s
Stalling speed (with flap)	67 km/h	
Service ceiling	4100 m	4300 m
Max range (without reserve) 65%	740 km	833 km
Take off to 15 m	330 m	290 m
Take off run	165 m	145 m
Landing run	100 m	
Landing distance	265 m	

Tabella 1.3: P92-JS: le caratteristiche tecniche

- per i decolli e gli atterraggi è sufficiente una ridotta aviosuperficie: questa possibilità, unita alla notevole autonomia di volo, consente di operare per periodi relativamente lunghi anche in zone lontane da aeroporti;
- l'acquisizione dei dati può essere comandata direttamente dal pilota: non è necessario disporre a bordo di un operatore dedicato.

In merito alle dimensioni del P92-JS, l'hardware di bordo - antenne, elettronica, piattaforma INS/GPS, ecc. - è stato realizzato per adattarsi ai ridotti volumi disponibili: il tutto è reso quanto più compatto e modulare possibile. Di fatto, la scelta di questo aeromobile costituisce una soluzione intermedia verso una più spinta miniaturizzazione dei sistemi, la cui estrema evoluzione consisterà nell'adozione degli UAV¹ quali piattaforme privilegiate.

1.2.3 L'hardware

Per quanto riguarda la configurazione hardware, il sistema è stato progettato modularmente allo scopo di poterlo facilmente utilizzare su altre piattaforme di piccole dimensioni. Le antenne sono state progettate dall'Università della Calabria e presentano una configurazione basata su un array multistrato uniforme di 3x3 elementi. Il loro sottosistema di interfaccia (I/F) è costituito da guidelines, circolatori ed interruttori in ferrite per l'implementazione della modalità ping-pong.

¹Acronimo per *Unmanned Aerial Vehicle*.

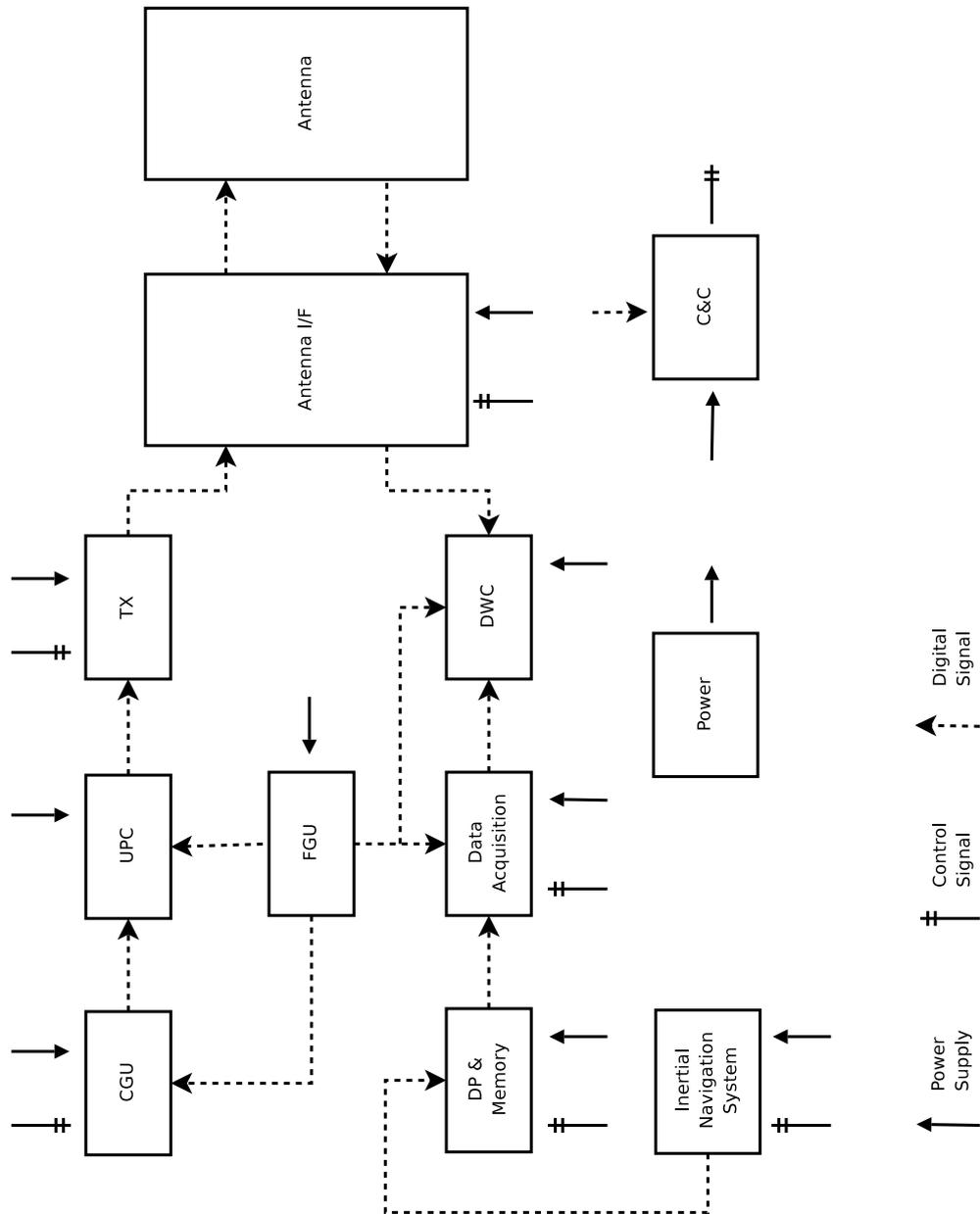


Figura 1.2: L'hardware MINISAR

I chirp sono generati digitalmente da una Chirp Generation Unit (CGU), basata su un sintetizzatore ed un convertitore digitale/analogico estremamente avanzati. Le caratteristiche di questi elementi consentono larghezze di banda fino 150 Mhz.

La Frequency Generation Unit (FGU) è il cuore del sistema e rappresenta la sorgente di rumore di fase che maggiormente influenza le performance interferometriche del radar. Il suo scopo è fornire le frequenze ai vari sottosistemi digitali e principalmente a quelli che si occupano della conversione dei chirp. A causa della modalità di funzionamento stepped chirp, la FGU deve essere in grado di fornire quattro valori di frequenza diversi in ogni PRI.

Le unità di Up- e Down-Conversion (UPC, DWC) sono responsabili della traslazione in frequenza dei chirp prima della trasmissione e dopo la ricezione, realizzando internamente anche le operazioni di amplificazione e filtraggio necessarie.

Il sistema di trasmissione è costituito da un piccolo TWT commerciale che può amplificare fino 120 W il segnale in banda X prima di passarlo all'antenna. Il baso livello di potenza trasmesso costituisce una delle caratteristiche peculiari del sistema.

L'unità di processamento dei dati e memoria (DP & Memory) si occupa della formattazione e salvataggio dei dati provenienti dalle antenne e dalla piattaforma inerziale. Questo sottosistema è supporta data rate fino a 160

Mbyte/sec.

MINISAR è anche equipaggiato con un sistema di navigazione inerziale (INS) e con un ricevitore GPS, allo scopo di misurare con un alto livello di accuratezza l'assetto e la posizione dell'antenna. I dati forniti da questi sensori sono usati per georeferenziare le immagini del radar.

Dal punto di vista della potenza elettrica, MINISAR è completamente autonomo grazie ad un sistema di batterie che gli assicurano circa mezz'ora di autonomia.

1.2.4 Il software

Per l'elaborazione dei dati MINISAR è stato sviluppato presso il CORISTA un completo software di SAR processing interferometrico, il cui scopo è realizzare immagini SLC e Digital Elevation Model a partire dai dati raw [4] [5].

Come si può evincere dal diagramma di flusso riportato in figura 1.3, l'architettura del sistema può essere analizzata a blocchi considerando una serie di procedure consecutive ciascuna delle quali risulta essere chiaramente individuabile all'interno del processamento complessivo. Ciascun blocco, considerando un metaprodotto in ingresso, fornisce un metaprodotto in uscita caratterizzato da un ben preciso livello di processamento.

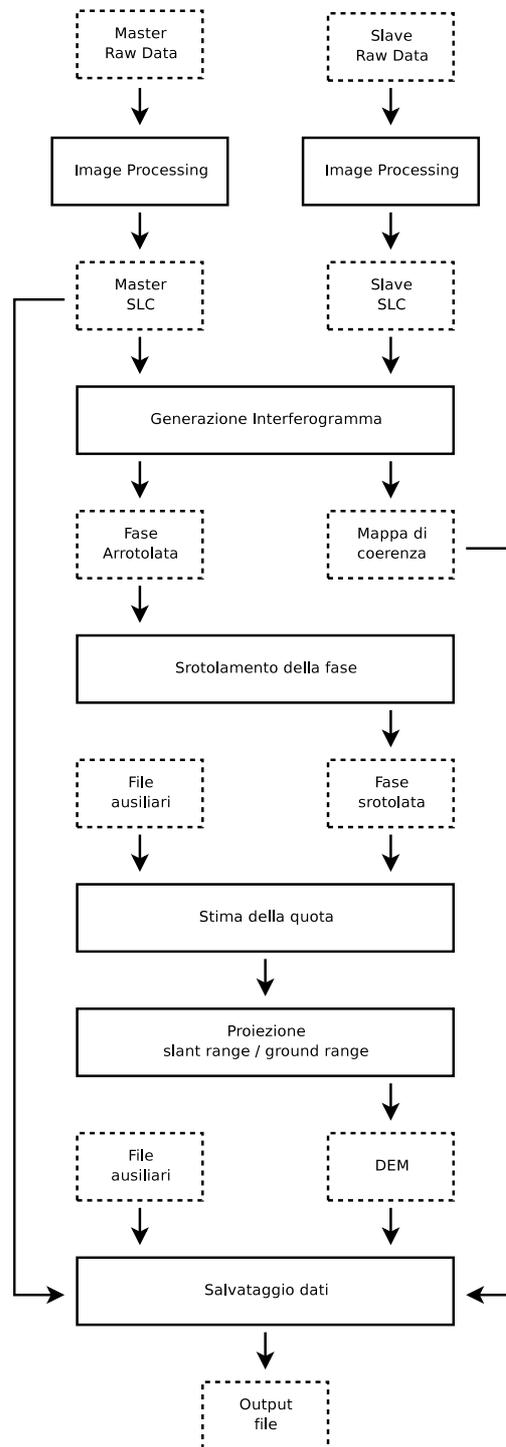


Figura 1.3: Il software MINISAR

Per metaprodotto si intende, in senso lato, un prodotto intermedio non necessariamente utilizzabile come dato finale né necessariamente avente un preciso significato fisico dato che, nell'economia complessiva dello schema architetturale, ciascuna delle procedure in cui si articola il processing può prevedere, se possibile, più metaprodotto.

A monte dell'algoritmo di focalizzazione, si colloca il blocco di deformazione che consente di sviluppare l'intero processo indipendentemente dal formato dei dati d'ingresso.

Per quanto attiene la focalizzazione, ovvero la generazione di immagini SLC, è stato implementato l'algoritmo chirp scaling, il cui nome deriva dal fatto che esso permette di scalare l'immagine direttamente mentre la si comprime consentendo, in fase di processamento, una coregistrazione tra due immagini SAR ed evitando ulteriori elaborazioni prima della fase di generazione dell'interferogramma. Il chirp scaling agisce per mezzo di una serie di FFT e moltiplicazioni. Ciò significa che i filtri in range ed azimut sono delle funzioni nel dominio trasformato [6].

A partire dalle immagini SLC, occorre considerare una riduzione del rumore di speckle sulla fase interferometrica. Una volta avute le SLC a basso speckle, occorre generare l'interferogramma e la mappa di coerenza. L'interferogramma è, quindi, srotolato e viene effettuata una stima delle altezze. A questo punto si ha a disposizione un DEM in slant range. Il passaggio in ground range viene effettuato per mezzo di un riposizionamento dei pun-

ti mediante considerazioni di tipo vettoriale. I dati prodotti vengono infine salvati in un file.

1.3 La motion compensation

Una delle maggiori difficoltà insite nell'adozione per il telerilevamento di piattaforme aeree di piccole dimensioni consiste nelle notevoli perturbazioni di volo cui sono soggette. Nella formazione dell'immagine radar si ipotizza una traiettoria di volo rettilineo uniforme, per cui ogni discostamento da questa condizione comporta un degrado nella qualità della focalizzazione². Questo stato di cose rende necessaria l'adozione di algoritmi di compensazione che riducano l'impatto delle condizioni di acquisizione sul prodotto finale.

Il presente lavoro di tesi si occupa dello studio di questo problema in relazione allo scenario operativo di MINISAR ed individua nell'algoritmo PGA, la tecnica di motion compensation più promettente.

²In particolare, interviene nello spettro di frequenze una perturbazione del termine di fase.

Capitolo 2

L'algoritmo PGA

2.1 Aspetti generali

L'algoritmo PGA¹, sviluppato presso i Sandia National Laboratories tra la fine degli anni '80 e l'inizio degli anni '90, rappresenta la soluzione di maggior successo al problema dell'autofocus [8]. La sua applicazione nel corso degli anni è stata ininterrotta ed ha riguardato principalmente il settore dei radar e sonar ad apertura sintetica e quello dell'interferometria stellare.

L'enorme popolarità di questo algoritmo è riconducibile a due fattori:

- non è necessario conoscere i meccanismi che determinano la defocalizzazione delle immagini;
- la ridondanza dell'errore di fase è riconosciuta ed opportunamente sfruttata.

¹Acronimo per *Phase Gradient Autofocus*.

Il primo aspetto riguarda il problema della modellazione degli errori di fase. Alcune tecniche di autofocus molto usate in passato, tra cui il metodo del contrasto, estraggono dalle immagini i termini di degrado ipotizzando a priori una forma per la curva di errore: il problema è ricondotto alla determinazione dei soli parametri che servono a particolarizzare tale curva. Il funzionamento di questi algoritmi è, quindi, limitato ai casi in cui sussiste un'effettiva congruenza tra il modello adottato e la realtà dei dati.

Il PGA è il più illustre rappresentante del filone alternativo all'approccio appena descritto. La stima dell'errore di fase è completamente svincolata da qualsiasi apriorismo e questo consente all'algoritmo di affrontare un'ampia gamma di problematiche, a partire da quella della motion compensation fino alle distorsioni delle onde elettromagnetiche nei mezzi ionizzati [9].

Il secondo punto fa riferimento alla possibilità di sfruttare ai fini della stima la ripetitività dell'errore di fase nello spettro di frequenze delle immagini. Questo aspetto rappresenta la vera forza dell'algoritmo PGA ed il motivo della sua eccezionale diffusione. Questa ripetitività, infatti, consente di adottare un'approccio statistico al problema della stima e rende possibile affrontare con successo la focalizzazione di scene del tutto prive di scatteratori molto forti ed isolati. In tal senso, in letteratura sono testimoniate prestazioni estremamente significative [10] [11]. Risultati parimenti positivi sono stati ottenuti nell'ambito di questo lavoro di tesi.

2.2 Struttura

Le versioni del PGA disponibili in letteratura sono numerose e strettamente legate alle specifiche applicazioni di riferimento. Si fornisce nel seguito un quadro di massima degli elementi strutturali comuni a tutte. Si intende discutere in questa sede i soli aspetti concettuali, rimandando alla sezione successiva la trattazione di dettaglio.

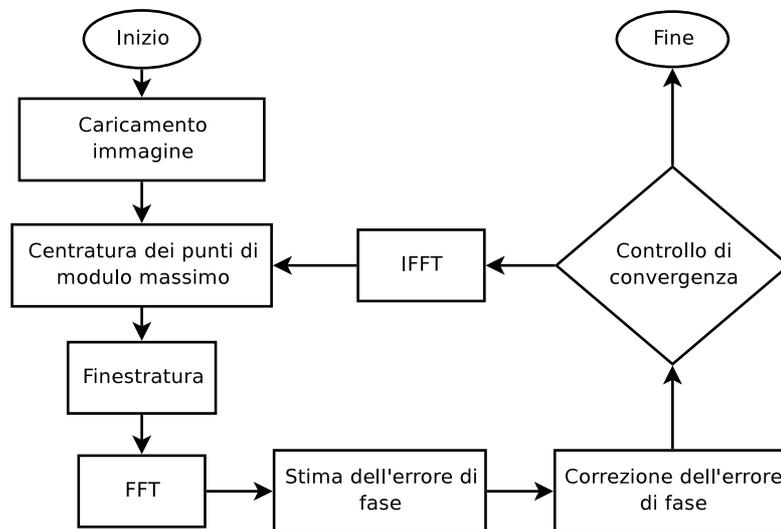


Figura 2.1: Schema a blocchi del PGA

La matrice complessa che rappresenta l'immagine radar completamente formata costituisce il punto di partenza del PGA. Lo scopo dell'algoritmo è fornire stime strettamente monodimensionali dell'errore di fase: quando lo si vuole sfruttare sia in azimuth sia in range, si deve prevedere un'applicazione ripetuta.

Nell'ottica della motion compensation, le caratteristiche dell'errore di fase consentirebbero di operare lungo il solo azimuth. Nonostante questo, per il trattamento dei dati MINISAR e TOPSAR si è scelto di prendere in considerazione entrambe le direzioni, ottenendo miglioramenti dell'immagine con la rimozione di errori anche non strettamente legati al moto della piattaforma.

Essendo le modalità di applicazione del PGA le stesse in entrambe le direzioni, d'ora in avanti si ipotizzerà di effettuare la stima sulle righe della matrice, senza specificare se esse rappresentino gli azimuth oppure i range.

In pre-processing, si può alleggerire il peso computazionale dell'algoritmo limitando il numero di righe e colonne su cui si effettua la stima. A tal fine, è necessario stabilire quali righe contengono le informazioni più significative e qual è il numero minimo di colonne che determina un buon rapporto tra la risoluzione nel dominio delle frequenze e la velocità di calcolo. Una rassegna dei documenti specialistici trovati in letteratura ha mostrato come tipicamente ci si affidi a criteri empirici per la scelta di questi parametri [10].

Il PGA opera in quattro passi, ripetuti iterativamente fino al soddisfacimento di un assegnato criterio di convergenza. Le operazioni in gioco sono quelle di centratura, finestrazione, stima e correzione. Nei paragrafi che seguono si analizzano i dettagli di ciascuno di questi punti.

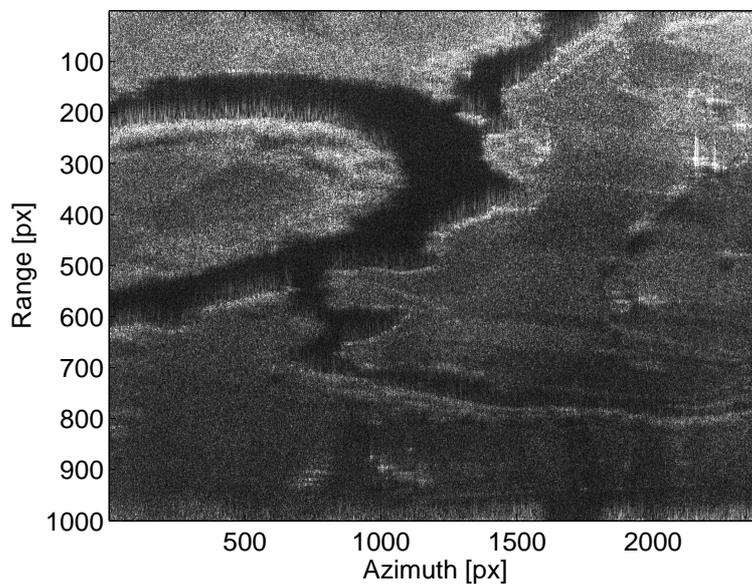


Figura 2.2: Immagine degradata da errore polinomiale

2.2.1 Centrazione

Il primo passo del PGA consiste nel porre al centro di ogni riga i punti di modulo massimo, effettuando un'operazione di *circular shifting*. Lo scopo è mettere a registro le zone di immagine che presentano le migliori proprietà di contrasto, dal momento che da esse è più semplice estrarre l'informazione sulla risposta impulsiva defocalizzata del sistema. La condizione ideale è costituita dall'incolonnamento di tutti scatteratori puntiformi, ma lo stimatore del PGA funziona altrettanto bene su zone con deboli variazioni di luminosità.

Le modalità di realizzazione della centrazione suggeriscono che per radar

ad apertura sintetica stripmap non è consigliabile utilizzare il PGA sulla globalità dell'immagine. In tal caso, infatti, nell'applicare l'algoritmo in azimuth, si andrebbero a mettere a registro punti relativi a posizioni molto distanti lungo la traiettoria, con il risultato di mediare nel processo di stima su porzioni di immagine degradate da errori di fase diversi.

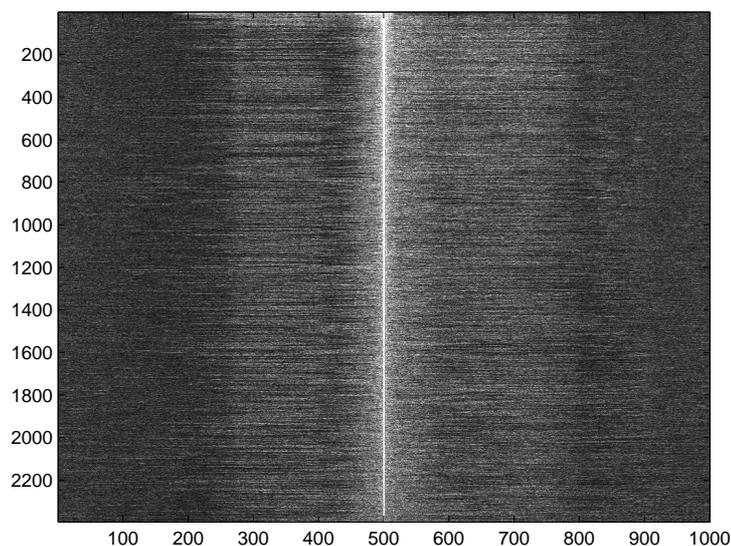


Figura 2.3: Centratura

L'uso del PGA per il caso stripmap può essere garantito in vari modi [13] [12]. Nello specifico del presente lavoro di tesi ci si è riferiti a [14] e si è provveduto a ripartire l'immagine in blocchi di azimuth di dimensione opportuna. La loro estensione dipende dal tempo di apertura sintetica del radar e dal livello di perturbazione manifestatosi durante il volo.

2.2.2 Finestratura

Durante la seconda fase, tutte le righe sono moltiplicate per una finestra centrata sui punti di modulo massimo. Lo scopo di quest'operazione è preservare l'informazione relativa ad un intorno di tali punti, eliminando tutto ciò che non fornisce contributo. La corretta scelta di forma e dimensioni della finestra rappresenta uno dei problemi più delicati dell'intero PGA. Il modo in cui si filtrano i dati in questa fase, infatti, determina ciò su cui concretamente si effettua la stima: dimensioni troppo piccole o troppo grandi possono portare a risultati del tutto inaccurati.

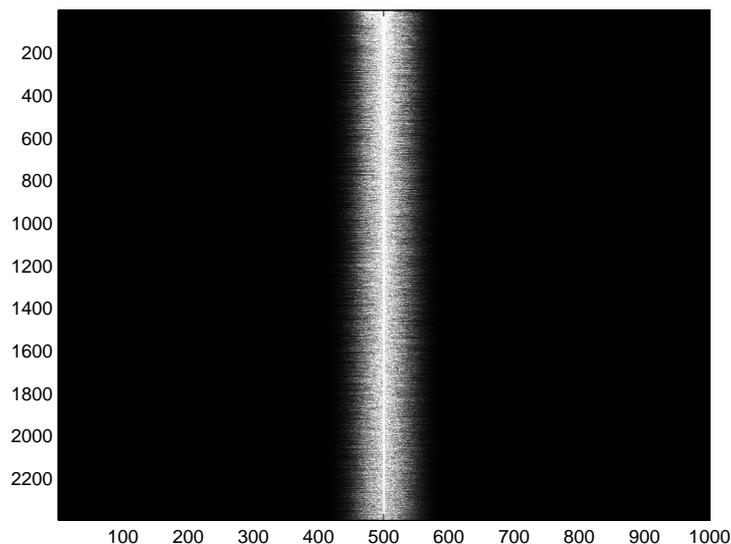


Figura 2.4: Finestratura

Generalmente, le dimensioni della finestra sono determinate automatica-

mente ad ogni passo in base all'informazione contenuta nell'immagine. Quest'operazione può essere effettuata in vari modi [10] [15] e le diverse soluzioni si distinguono principalmente per la capacità di gestire scene a basso rapporto segnale/clutter. In effetti, non è sempre possibile affidarsi a metodi automatici ed, in alcuni casi, per valori limite di tale rapporto si prevede l'uso di finestre che assumono ad ogni iterazione dimensioni predeterminate.

2.2.3 Stima

La stima dell'errore si basa su un'analisi della fase nel dominio delle frequenze e rappresenta il cuore dell'algoritmo PGA. Le possibili scelte documentate in letteratura sono molteplici. Soluzioni molto comuni sono quella dello stimatore lineare non polarizzato a varianza minima (LUMV) [8] e quella dello stimatore di massima verosimiglianza (ML) [16].

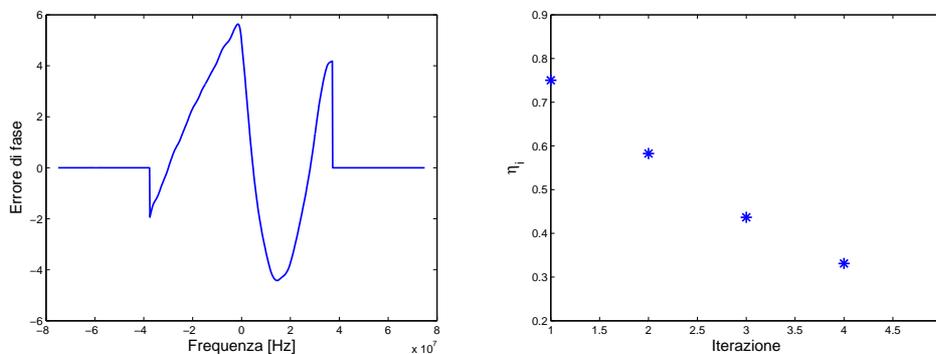


Figura 2.5: Errore totale e criterio di convergenza

La grandezza calcolata dallo stimatore è il gradiente dell'errore di fase,

per cui si rende necessaria un'integrazione per ricavare l'errore vero e proprio. A valle di quest'operazione, bisogna sottrarre dalla stima il termine lineare perchè non contribuisce alla defocalizzazione dell'immagine ma, per le ben note proprietà della trasformata di Fourier, è unicamente responsabile di uno spostamento dei dati nel dominio antitrasformato.

2.2.4 Correzione

Le operazioni di correzione presenti all'interno del PGA sono due. La prima è effettuata all'interno di ciascuna iterazione e serve a ripulire i dati utilizzati per la stima. La seconda è realizzata in chiusura di algoritmo ed ha lo scopo di rimuovere l'errore di fase complessivo dall'immagine originale.

In entrambi i casi, i dati sono corretti nel dominio delle frequenze moltiplicando ciascuna riga per l'esponenziale complesso dell'errore stimato. Formalmente, se la trasformata di Fourier della generica riga è $F(u)$, il termine corretto è

$$F_c(u) = F(u)e^{j\phi(u)} \quad (2.1)$$

Per quanto riguarda la fase di correzione interna al ciclo, tipicamente sono necessarie 4 - 5 iterazioni perchè i dati risultino correttamente focalizzati.

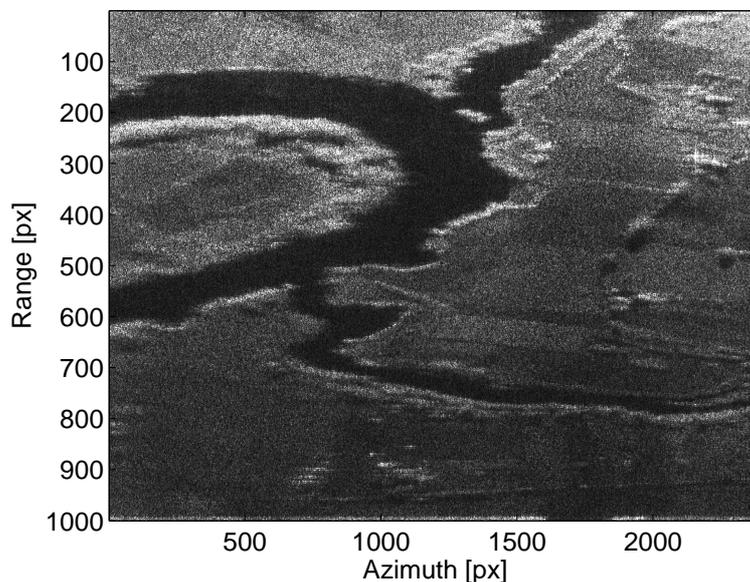


Figura 2.6: Immagine dopo la correzione

2.3 Implementazione

Il software di elaborazione di MINISAR include un'implementazione della versione del PGA comparsa per prima in letteratura [8]. Questa soluzione, giustificata dalla mole di documentazione disponibile, non sfrutta gli sviluppi proposti nel corso degli ultimi anni e, in particolare, non fa alcun riferimento a specifici accorgimenti per ridurre il peso computazionale e migliorare il funzionamento dell'algoritmo su scene a basso rapporto segnale/clutter. Nell'ambito di questa tesi, sono state individuate le possibilità di sviluppo più promettenti e le si è testate in riferimento ad un assegnato set di dati. Nel seguito, si descrivono l'algoritmo originale, i suoi limiti e le modifiche ad esso apportate nella nuova implementazione.

2.3.1 L'implementazione originale

L'algoritmo

Il primo passo del PGA di MINISAR consiste in un'operazione di pre-processo in cui si selezionano i dati su cui far lavorare l'algoritmo.

Se N è il numero di righe che si vogliono estrarre dalla matrice dell'immagine, il criterio di scelta prevede

1. il calcolo del modulo massimo di ogni riga;
2. la realizzazione di una lista dei moduli massimi e rispettivi indici di riga, ordinata per valori decrescenti del primo parametro;
3. la selezione delle prime N righe della lista.

Le righe sono prese nella loro piena lunghezza, senza effettuare alcun tipo di selezione sulle colonne. Il risultato di quest'operazione rappresenta l'ingresso delle fasi canoniche di centratura, finestatura, stima e correzione.

Il passo successivo consiste nell'effettuare le operazioni di centratura e finestatura. La forma prescelta per la finestra è quella della funzione unitaria: nella parte centrale, il valore è ovunque uno; in quelle periferiche, zero. L'ampiezza di questo filtro è determinata tramite la funzione

$$s(x) = \frac{1}{N} \sum_{i=1}^N |f_i(x)|^2 \quad (2.2)$$

ove x è l'indice di colonna, i quello di riga ed $f_i(x)$ rappresenta il valore della matrice complessa dei dati sottoposti a centratura, nel punto (i, x) .

In virtù di come è stata calcolata, questa funzione presenta un massimo assoluto nel centro e decresce verso l'esterno: i due punti, a destra ed a sinistra, in cui si registra una caduta di 10dB dal massimo definiscono le ascisse di taglio della finestra.

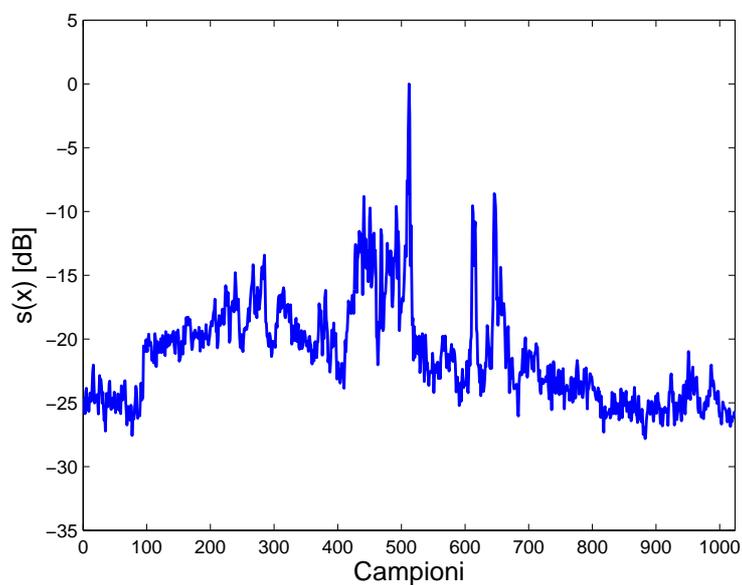


Figura 2.7: Funzione di riferimento

La fase di stima è realizzata trasformando ciascuna riga nel dominio delle frequenze ed applicando ad esse lo stimatore lineare non polarizzato a varianza minima (LUMV). Formalmente, la sua espressione è data da

$$\dot{\phi}_{lumv}(u) = \frac{\sum_{i=1}^N \text{Im} \{ G_i^*(u) \dot{G}_i(u) \}}{\sum_{i=1}^N |G_i(u)|^2} \quad (2.3)$$

ove u è l'indice delle frequenze di riga e $G_i(u)$ è la trasformata di Fourier della i -sima riga centrata e finestrata.

Il gradiente dell'errore di fase così ottenuto è integrato con il metodo dei trapezoidi e ripulito dei termini fuori banda. A questo punto, lo si sottrae dalla fase dei dati per realizzare la correzione e lo si accumula nel vettore di errore complessivo.

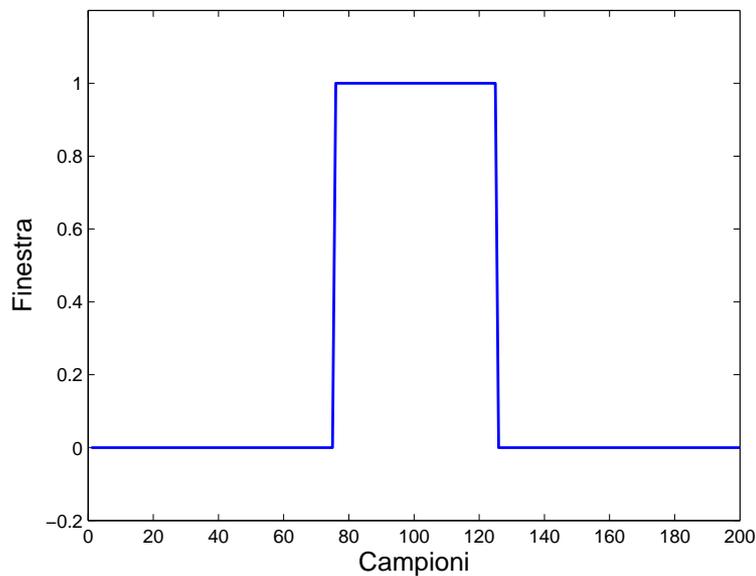


Figura 2.8: Finestra unitaria

In chiusura di iterazione, si calcola la deviazione standard del valore assoluto della fase stimata e la si confronta con una soglia assegnata. Il risultato definisce se si è pervenuti a convergenza o sono necessari altri passaggi.

Il soddisfacimento della condizione di arresto fa terminare l'esecuzione vera e propria dell'algoritmo e rende disponibile al software di elaborazione il valore dell'errore di fase complessivo. Questa grandezza è utilizzata per correggere l'immagine, usando esattamente la stessa procedura adottata all'interno dell'algoritmo per effettuare le correzioni alla fine di ogni ciclo.

Limiti

I test condotti sui dati TOPSAR e MINISAR mostrano un funzionamento complessivamente buono dell'algoritmo, ma ne sottolineano al contempo alcuni importanti limiti.

In primo luogo, è stato verificato come per ottenere una corretta estrazione dell'errore di fase sia necessario calibrare, a seconda dell'applicazione, in modo diverso i parametri in gioco. In particolare, la soglia di 10dB per la determinazione automatica della larghezza della finestra non è parimenti adeguata per l'applicazione del PGA in azimuth ed in range. Usando lo stesso valore in entrambi i casi, si può pervenire ad una stima completamente sbagliata dell'errore di fase in una delle due direzioni e, conseguentemente, ad una severa perdita di risoluzione dell'immagine (vedi figura 2.9).

In alcune condizioni, si è anche verificato che utilizzare un numero relati-

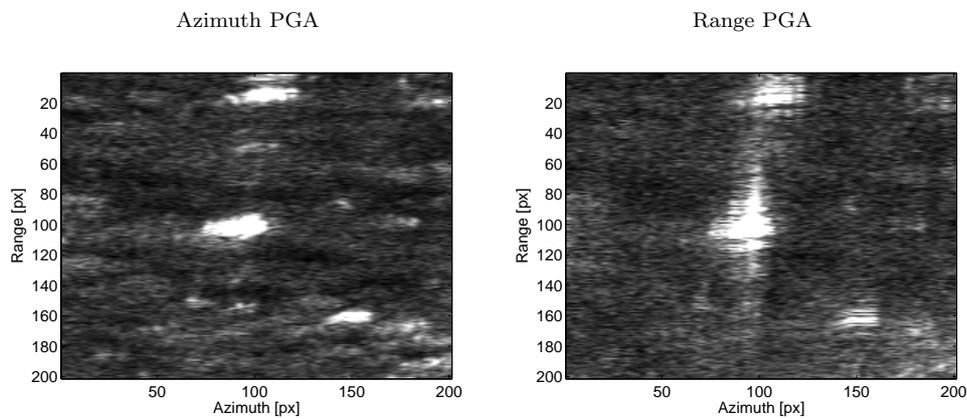


Figura 2.9: Applicazione del PGA originale in azimuth e range

vamente alto di righe per effettuare la stima può portare ad una valutazione completamente errata dell'errore di fase. Questo fenomeno è contrario a ciò che normalmente ci si aspetterebbe aumentando le dimensioni di un campione. Le ragioni sono diverse ma, principalmente, vanno ricercate nella scarsa efficienza con cui la finestrazione rimuove i dati non significativi.

Altro aspetto rilevante è la gestione del carico computazionale complessivo. Questo problema è radicato sia nel modo in cui si realizzano le singole iterazioni, sia nella determinazione del loro numero.

Per quanto riguarda il primo aspetto, c'è da riscontrare che la presenza di un'operazione di finestrazione rende inappropriato l'uso della FFT sull'intera lunghezza di riga. La maggior parte delle colonne, infatti, a valle del filtraggio, assume valore nullo e non contribuisce alla costruzione dello spettro di

frequenze. In tal senso, in molti casi, si allungano inutilmente i tempi di calcolo.

Il secondo aspetto riguarda la scelta del criterio di convergenza. Di fatto, la condizione di arresto è stabilita rispetto ad una grandezza assoluta e questo non fornisce sufficienti garanzie di corretto funzionamento in tutte le situazioni. Ci possono essere casi in cui questo approccio, non potendosi verificare la condizione di convergenza, porta all'esecuzione dell'algoritmo fino al raggiungimento del numero massimo di iterazioni, con un'inutile aumento del peso computazionale.

2.3.2 L'evoluzione

I problemi riscontrati con l'implementazione originale del PGA di MINISAR hanno indotto allo sviluppo di una nuova versione dell'algoritmo che includesse alcuni dei risultati più rilevanti degli ultimi anni [15]. Da un punto di vista metodologico, l'attenzione è stata concentrata solo sugli aspetti critici, lasciando inalterate la struttura e le scelte realizzative originali ove opportuno. Le innovazioni più interessanti sono state introdotte nella fase di selezione dei dati ed in quella di finestratura, oltre che, in misura minore, in relazione al criterio di convergenza.

Selezione dei dati

La procedura di selezione dei dati è stata modificata introducendo la possibilità di prendere per ciascuna riga della matrice solo un numero limitato

di colonne. All'atto della scelta delle righe su cui effettuare il PGA, solo un opportuno intorno dei punti di modulo massimo è effettivamente considerato. Questo accorgimento è inteso ad alleggerire il peso computazionale dell'algoritmo, migliorandone al contempo il risultato.

Per quanto concerne il primo aspetto, si deve considerare che le operazioni più ricorrenti ed impegnative dell'intero processo iterativo sono le FFT. Di fatto, ad ogni ripetizione della catena di centratura, finestatura, stima e correzione sono effettuate ben tre trasformate monodimensionali su matrici complesse dell'ordine delle centinaia di migliaia di elementi. Il peso computazionale di ciascuna di queste operazioni è decisamente rilevante e va, per quanto possibile, minimizzato. In tal senso, si può osservare che a valle della fase di finestatura, essendo la maggior parte dei punti posto a zero, solo una frazione ridotta dei dati effettivamente contribuisce alla determinazione dello spettro di frequenze. In questa situazione, considerare l'intera estensione delle righe ha il solo risultato di realizzare implicitamente enormi interpolazioni [17].

Quantitativamente, si può considerare che

- per vettori di lunghezza pari a potenze di due il peso computazionale della FFT ha andamento lineare in funzione del numero di campioni (vedi Figura 2.10);
- più di tre quarti dell'estensione di ogni riga è filtrato dalla finestra.

In virtù di questo, scegliendo una frazione del numero totale di colonne

ed avendo l'accortezza di selezionare un numero di campioni pari ad una potenza di due, si può ridurre sensibilmente il tempo impiegato per calcolare le FFT. Questo risultato è, per altro, accompagnato da un miglioramento della bontà della stima, dovuto al fatto di lavorare con una proporzione migliore tra dati reali e dati interpolati.

Guadagni teoricamente maggiori potrebbero essere ottenuti effettuando le trasformate di Fourier solo sugli elementi non nulli [15]. In questo modo, si avrebbero tempi di esecuzione delle singole FFT molto bassi e, soprattutto, decrescenti al diminuire delle dimensioni della finestra. Questa soluzione non è stata attuata per i seguenti motivi:

- al di sotto di 500 / 1000 punti la risoluzione in frequenza è troppo bassa per una buona stima dell'errore di fase;
- ad ogni iterazione si avrebbero vettori di errore di lunghezza diversa: si dovrebbe continuamente ricampionare per accumularli nel vettore di errore totale.

Ad ogni modo, si deve considerare che utilizzare un numero di colonne diverso dalla dimensione effettiva della matrice impone il ricampionamento dell'errore di fase totale al termine dell'esecuzione del PGA. Si ha la necessità, infatti, di adattare tale vettore all'immagine originale per la correzione finale. Quest'operazione in parte limita i vantaggi insiti nella nuova tecnica di selezione, ma c'è da considerare che essa è eseguita una sola volta, mentre le FFT sono all'interno di ogni singolo ciclo.

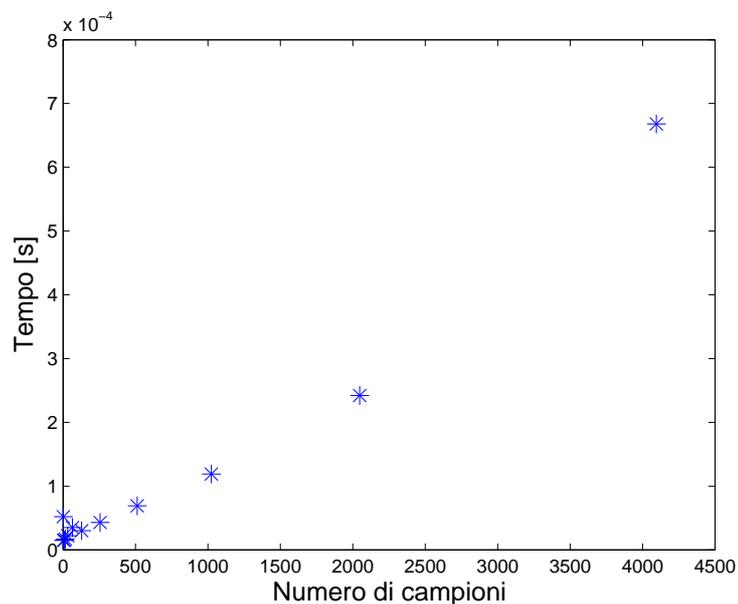


Figura 2.10: Peso computazionale della FFT

Allo stato attuale, il ricampionamento è realizzato con interpolatori cubic spline, ma è anche possibile utilizzare curve di regressione.

Finestratura

La fase di finestratura è stata completamente rivoluzionata rispetto alla versione originale dell'algoritmo. Seguendo le indicazioni in [15], è stata sostituita la tecnica classica a favore di un metodo più radicalmente basato sui dati. In particolare, si è intervenuto sulla forma della finestra e sul modo in cui sono determinate le sue dimensioni, lasciando invariata l'espressione della funzione $s(x)$ di riferimento.

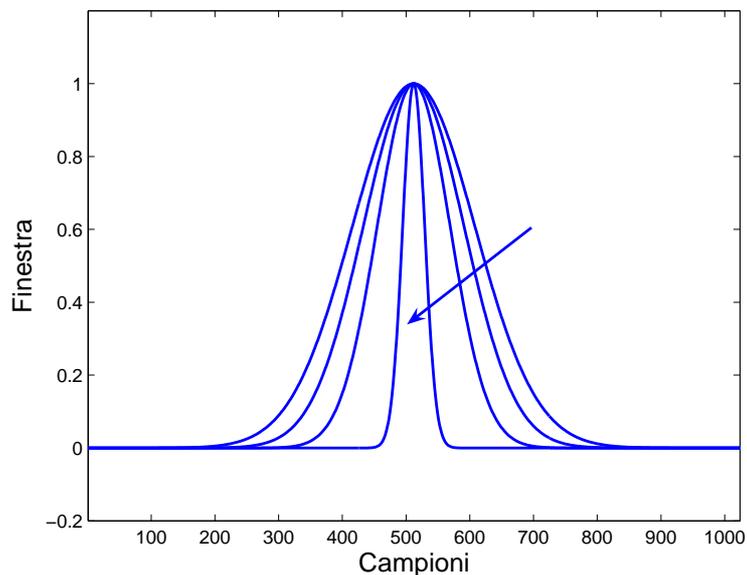


Figura 2.11: Finestra gaussiana

Per quanto riguarda il primo parametro, è stata scelta la funzione gaussiana. Il taglio dei dati risulta più naturale e contemporaneamente si ha un utile smoothing dello spettro di frequenze. Quest'ultima proprietà discende dal teorema della convoluzione e dal fatto che la gaussiana coincide con la propria trasformata.

$$F\{e^{-\alpha t}\} = \frac{1}{\sqrt{2\alpha}} e^{-\frac{\omega^2}{2\alpha}} \quad (2.4)$$

La dimensione della finestra nell'ambito di questo nuovo metodo è determinata considerando i punti (x_l, y_l) e (x_r, y_r) in cui la funzione di riferimento $s(x)$ attraversa la sua media. La gaussiana è costruita fissandone

il valor medio all'ascissa di centro riga x_c ed imponendo che per $y = 0.01$ sia $|x - x_c| = \max\{|x_l - x_c|, |x_r - x_c|\}$. Questa soluzione consente l'adattamento della soglia di taglio ai dati, ed evita la scelta di un valore fisso che, necessariamente, risulterebbe adeguato solo in certe condizioni.

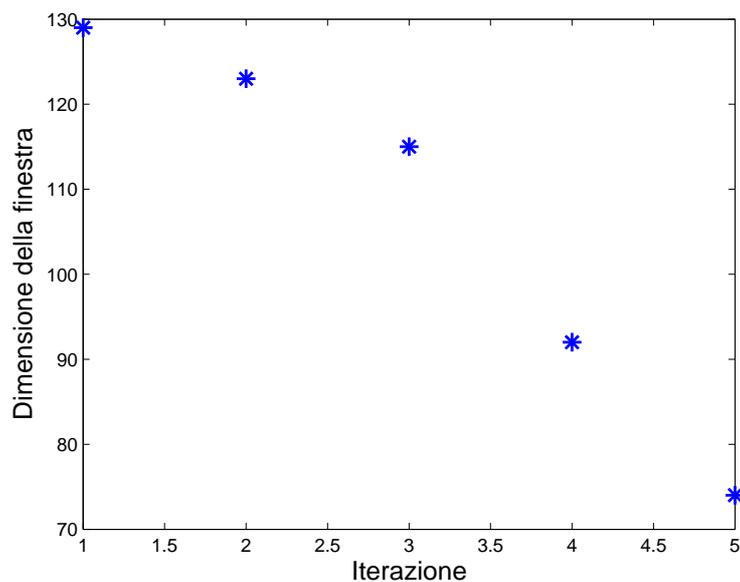


Figura 2.12: Dimensione della finestra in funzione delle iterazioni

C'è da rilevare che, in alcuni casi, si sono verificate oscillazioni nella dimensione della finestra all'aumentare del numero di iterazioni. Per gestire opportunamente questi casi, si è imposto che la curva dimensioni/iterazione sia monotona decrescente. In concreto, detta \hat{d}_i la dimensione determinata con il metodo automatico al passo i e d_{i-1} quella usata effettivamente al passo $i - 1$, si ha che

$$\hat{d}_i > d_{i-1} \Rightarrow d_i = 0.8d_{i-1} \quad (2.5)$$

Imporre questo restringimento evita di considerare dati non rilevanti, migliorando la capacità di correzione dell'algoritmo.

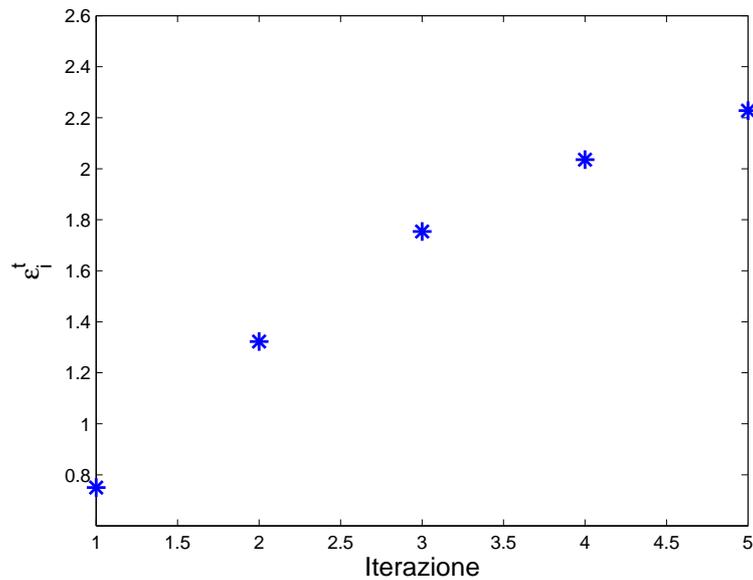


Figura 2.13: Errore totale in funzione delle iterazioni

Criterio di convergenza

La condizione di arresto non è più basata su un errore assoluto, ma sulla differenza percentuale tra lo RMS dell'errore totale all'iterazione precedente e quello all'iterazione attuale. Formalmente, si impone che alla convergenza la grandezza

$$\eta_i = \left| \frac{\epsilon_{i-1}^t - \epsilon_i^t}{\epsilon_{i-1}^t} \right| \quad (2.6)$$

sia minore di una soglia assegnata. Naturalmente, si assume che

$$\epsilon_i^t = \sqrt{\frac{1}{N} \sum_{k=1}^N x_k^2} \quad (2.7)$$

ove x_k è l'elemento di posto k -simo del vettore che contiene l'errore complessivo al passo i ed N il suo numero di elementi totali.

Questa scelta ha consentito una più semplice determinazione del valore di soglia ed ha consentito una maggiore uniformità di valutazione su una casistica più ampia.

Capitolo 3

Risultati

3.1 I test

3.1.1 Aspetti generali

La nuova implementazione del PGA sviluppata nell'ambito di questo lavoro di tesi è stata diffusamente testata su dati reali di radar ad apertura sintetica.

I dati della prima campagna di volo MINISAR e della missione TOPSAR hanno costituito la materia prima per l'esecuzione di tutte le prove di focalizzazione. Le scene utilizzate sono state le più disparate ed hanno spaziato da zone agricole a città fino a regioni completamente prive di qualsiasi segno di urbanizzazione. Nello specifico, si è fatto riferimento a quattro scene:

- il cratere del Vesuvio;
- la città di Somma Vesuviana;

- uno stabilimento industriale isolato;
- una zona agricola sulle sponde del fiume Volturno.

Ciascuna di queste scelte presenta caratteristiche che condizionano in modo peculiare il funzionamento dell'algoritmo PGA. La prima ed ultima scena, in particolare, essendo quasi completamente prive di scatteratori forti, costringono a valutare l'errore di fase prevalentemente su deboli contrasti e rappresentano le situazioni di maggiore criticità per il processo di stima. Condizioni diverse si registrano per la seconda e terza scena in cui sono presenti un gran numero di zone molto luminose, ma per lo più relative ad oggetti estesi.

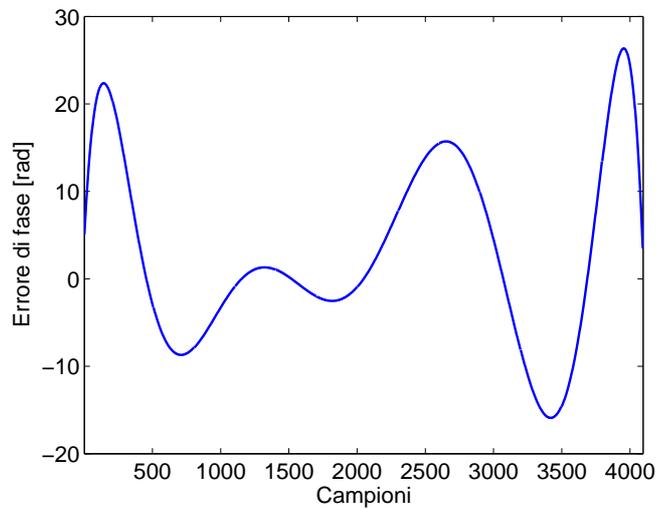


Figura 3.1: Errore polinomiale dell'ottavo ordine

Alla varietà di scene è associata anche varietà dell'errore di fase. I due set

di dati sono naturalmente caratterizzati da un degrado molto diverso, dovuto alle profonde differenze nelle piattaforme dei rispettivi radar. MINISAR e TOPSAR utilizzano, infatti, soluzioni che si collocano ad estremi opposti, essendo il primo montato a bordo di un P92-JS ed il secondo di un DC-8. A questo si aggiunga che si è provveduto in alcuni test a degradare ulteriormente le immagini con errori di fase assegnati. In particolare, sono state usate funzioni di tipo polinomiale e rumore bianco a varie intensità e combinazioni, al fine di valutare le condizioni limite di applicazione del PGA. I risultati riportati in questo testo si riferiscono all'errore polinomiale in figura 3.1 e sono decisamente rappresentativi anche delle altre situazioni.

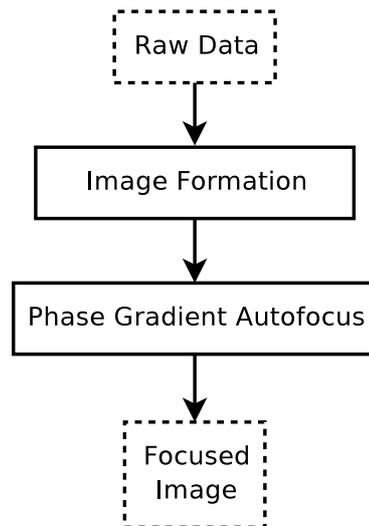


Figura 3.2: Collocazione classica del PGA

Questo approccio è inteso a garantire condizioni effettivamente significative della maggior parte delle situazioni prevedibili per un sistema SAR.

3.1.2 Il software di elaborazione

Le prove di focalizzazione relative sia ai dati TOPSAR sia a quelli MINISAR sono state realizzate utilizzando *minisar_proc*, un codice MATLAB originariamente pensato per coprire l'intero segmento di SAR processing di MINISAR. Alcune modifiche sono state necessarie per inserire la nuova implementazione dell'algoritmo PGA (vedi Appendice A.2) e prevedere la defocalizzazione dei dati con un errore di fase assegnato.

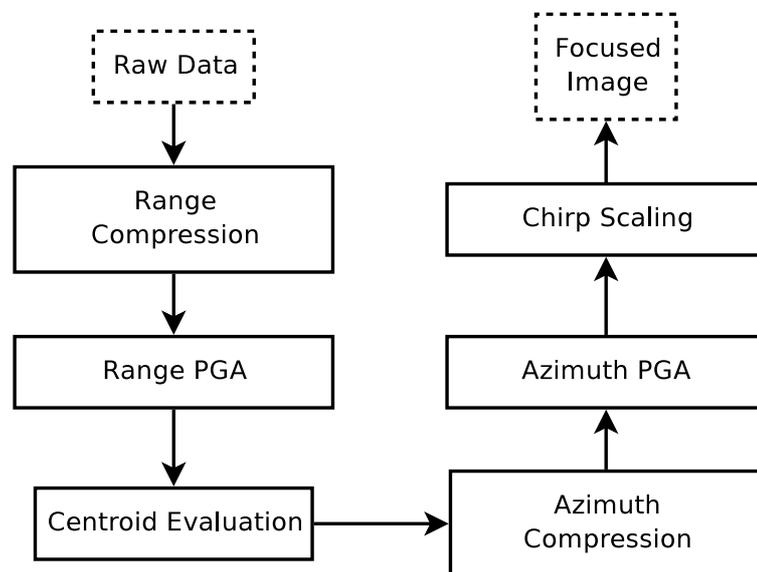


Figura 3.3: Collocazione del PGA per MINISAR/TOPSAR

A proposito delle caratteristiche specifiche di questo pacchetto, è opportuno puntualizzare che in letteratura generalmente si esegue il PGA a valle della formazione dell'immagine radar. In questo modo, ci si svincola completamente dai dettagli implementativi della focalizzazione, relegando il re-

cupero dell'errore di fase esclusivamente al post-processing.

Per l'elaborazione dei dati MINISAR e TOPSAR si è seguito un criterio diverso. Le compressioni in azimuth e range sono effettuate in due tempi. In una prima fase, si realizza una pre-focalizzazione dell'immagine con un classico algoritmo Range-Doppler e si applica contestualmente il PGA nelle due direzioni. In un secondo tempo, si esegue un'accurata focalizzazione mediante l'algoritmo di Chirp Scaling. In virtù di questa scelta, il secondo processo può sfruttare una matrice dei dati completamente esente da errore, nonché una valutazione dei centroidi doppler basata su dati già parzialmente ripuliti. Questa scelta può portare in certe condizioni a prestazioni migliori.

3.1.3 Valutazione dei risultati

I risultati dei test di focalizzazione sono stati analizzati tramite *SLCCompare*, un codice scritto in MATLAB dedicato allo scopo. Come si può evincere dalla figura 3.4, l'interfaccia grafica utente di questo pacchetto consente di selezionare il percorso delle immagini da confrontare ed il punto su cui calcolare la risoluzione spaziale. A fronte di queste informazioni, si visualizzano le due immagini ed il dettaglio del punto selezionato, insieme ad una stima dell'entropia e della risoluzione in azimuth e range.

La scelta dei parametri in base ai quali valutare la bontà della focalizzazione è stata dettata da considerazioni di carattere teorico, cui si è trovato esplicito riferimento in letteratura [11].

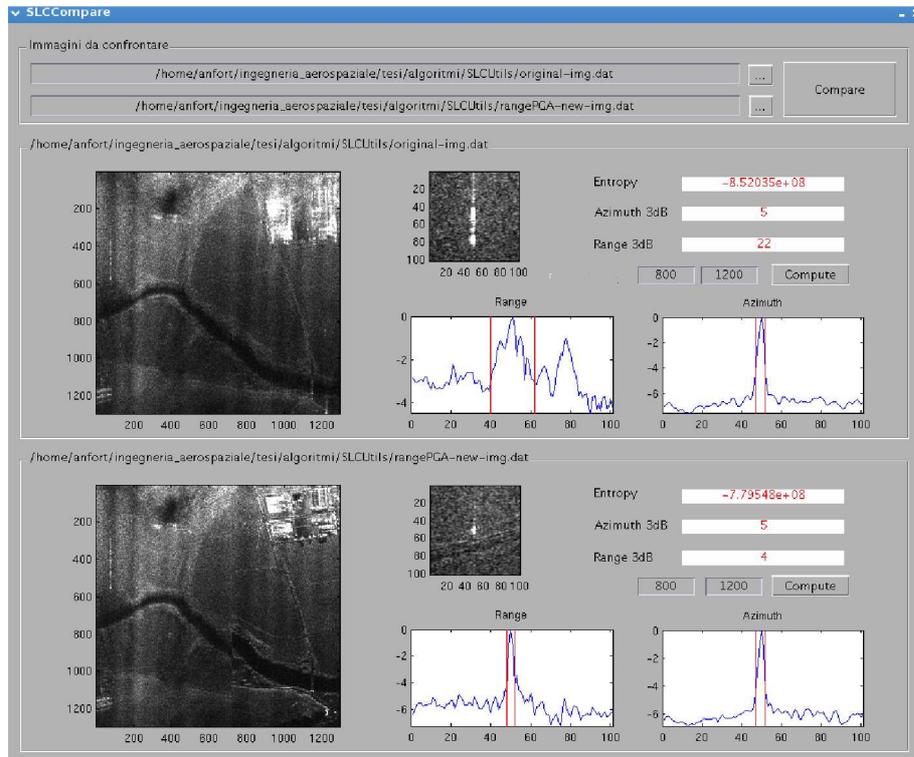


Figura 3.4: L'interfaccia grafica utente di SLCCCompare

In particolare, l'utilizzo dell'entropia è stato mutuato dalle tecniche di autofocus per ottimizzazione del contrasto, che usano la minimizzazione del valore assoluto di questo parametro come indicazione dell'avvenuta convergenza [19]. Formalmente, per entropia si intende

$$H = - \sum_{i=1}^{N_{row}} \sum_{j=1}^{N_{col}} I_{i,j} \log I_{i,j} \quad (3.1)$$

ove $I_{i,j}$ è il modulo della matrice dell'immagine nella cella di posto i, j ,

mentre N_{row} e N_{col} sono, rispettivamente, il numero di righe ed quello di colonne totali.

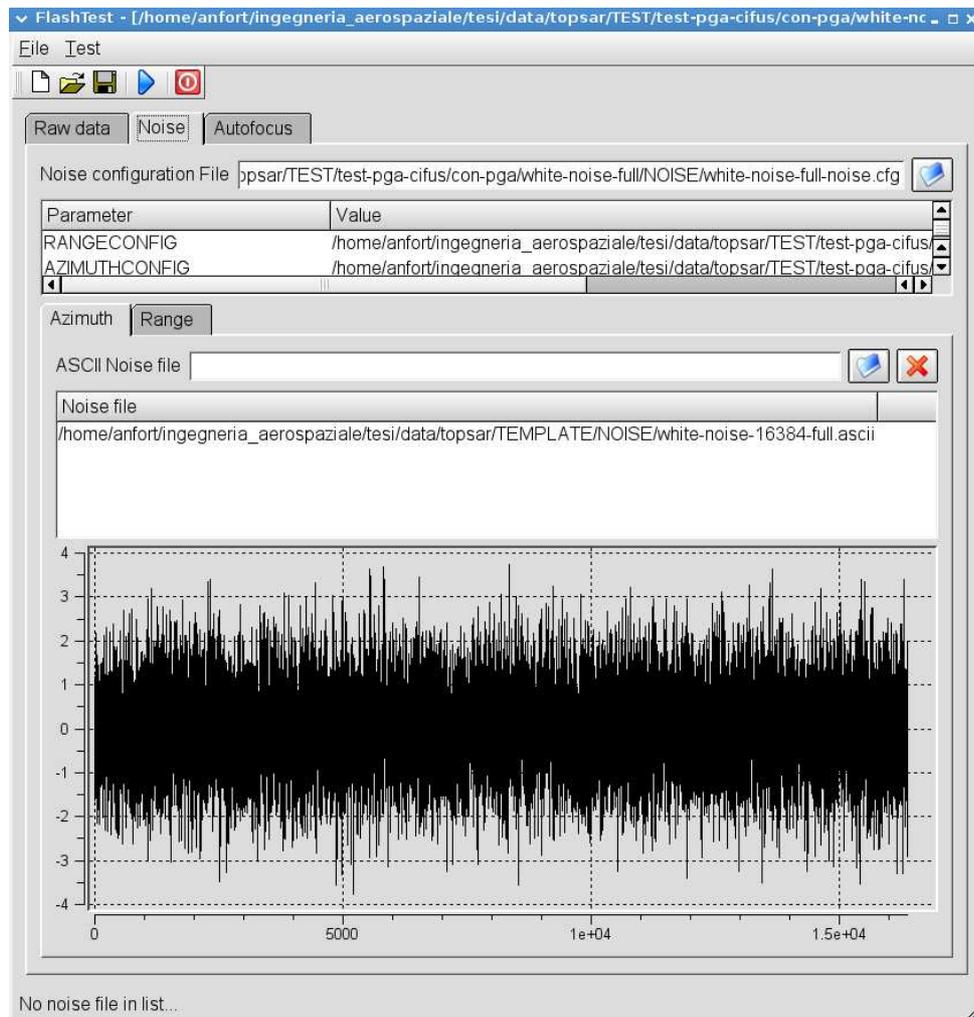


Figura 3.5: Il programma FlashTest

La scelta della risoluzione spaziale costituisce una soluzione classica, in grado di dare immediata misura del livello di dettaglio minimo riscontrabile all'interno dell'immagine. La determinazione di questo parametro è effettuata considerando, in range ed azimuth, i due punti, a destra ed a sinistra, di caduta a 3dB dal massimo nella risposta impulsiva del sistema. Ovviamente, avendo lavorato con immagini reali e non con simulazioni, non si è avuta a disposizione questa informazione e ci si è dovuti accontentare solo delle risposte che hanno approssimato meglio quella impulsiva. In tal senso, la misura di risoluzione è da considerarsi un'indicazione relativa utile al solo scopo di confrontare sulla medesima immagine gli effetti di processi diversi.

Si sottolinea, infine, che nell'ambito del presente lavoro di tesi è stato profuso uno sforzo particolare per rendere le operazioni di verifica degli algoritmi di autofocus le più semplici ed uniformi possibile. In particolare, sono stati sviluppati diversi programmi in MATLAB e C++ per gestire l'esecuzione dei test.

L'idea di base è utilizzare una singola interfaccia grafica utente (vedi figura 3.5) per gestire contemporaneamente i parametri del processo di formazione dell'immagine, quelli dell'algoritmo di autofocus e l'errore di fase imposto. In questo modo, i risultati del test e la configurazione usata per realizzarlo sono impacchettati insieme all'interno della medesima cartella, rendendo possibile riprodurre in un secondo momento le esatte condizioni originali. Un tale concetto dimostra in pieno la propria utilità al crescere della complessità delle verifiche e, quindi, del numero di parametri da monitorare.

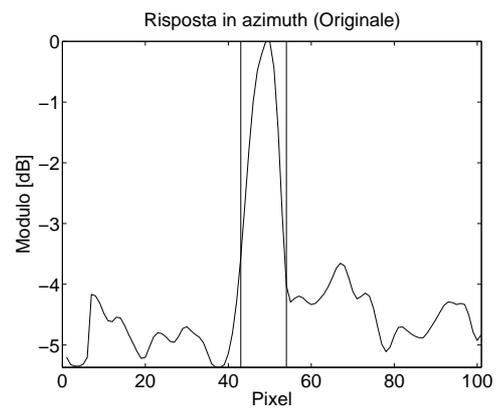
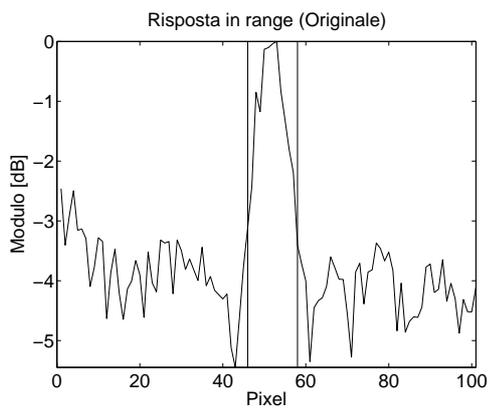
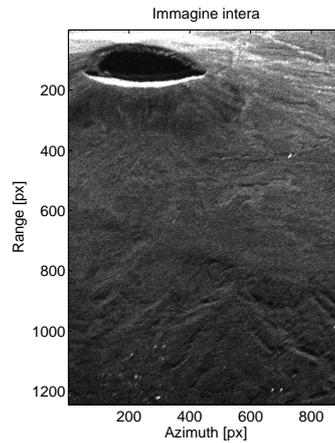
3.2 TOPSAR Data Set

TOPSAR è un radar ad apertura sintetica interferometrico della NASA che, all'inizio degli anni '90, ha volato a bordo di un DC-8 sulla provincia di Napoli [20] [21]. La strisciata utilizzata per le prove di focalizzazione parte dal Monte Faito e si estende per 20 km in direzione Nord. Da essa sono state scelte le due scene relative al cratere del Vesuvio ed alla città di Somma Vesuviana.

Le schede delle pagine seguenti presentano in forma schematica i risultati ottenuti su queste scene applicando il PGA all'immagine originale ed a quelle perturbate con errore polinomiale rispettivamente in azimuth e range. Per ciascuno dei tre gruppi, in prima pagina sono mostrate le condizioni dell'immagine non elaborata, con specifico riferimento all'andamento del modulo in range ed azimuth, alla risoluzione spaziale nelle due direzioni ed all'entropia. Nelle pagine successive, le stesse informazioni, arricchite dall'errore stimato e dal dettaglio del punto su cui si è calcolata la risoluzione, sono riportate per le immagini a valle dell'applicazione del PGA. Per convenzione, nella colonna di sinistra sono indicati i risultati dell'implementazione originale, mentre quella di destra è riservata alla nuova versione.

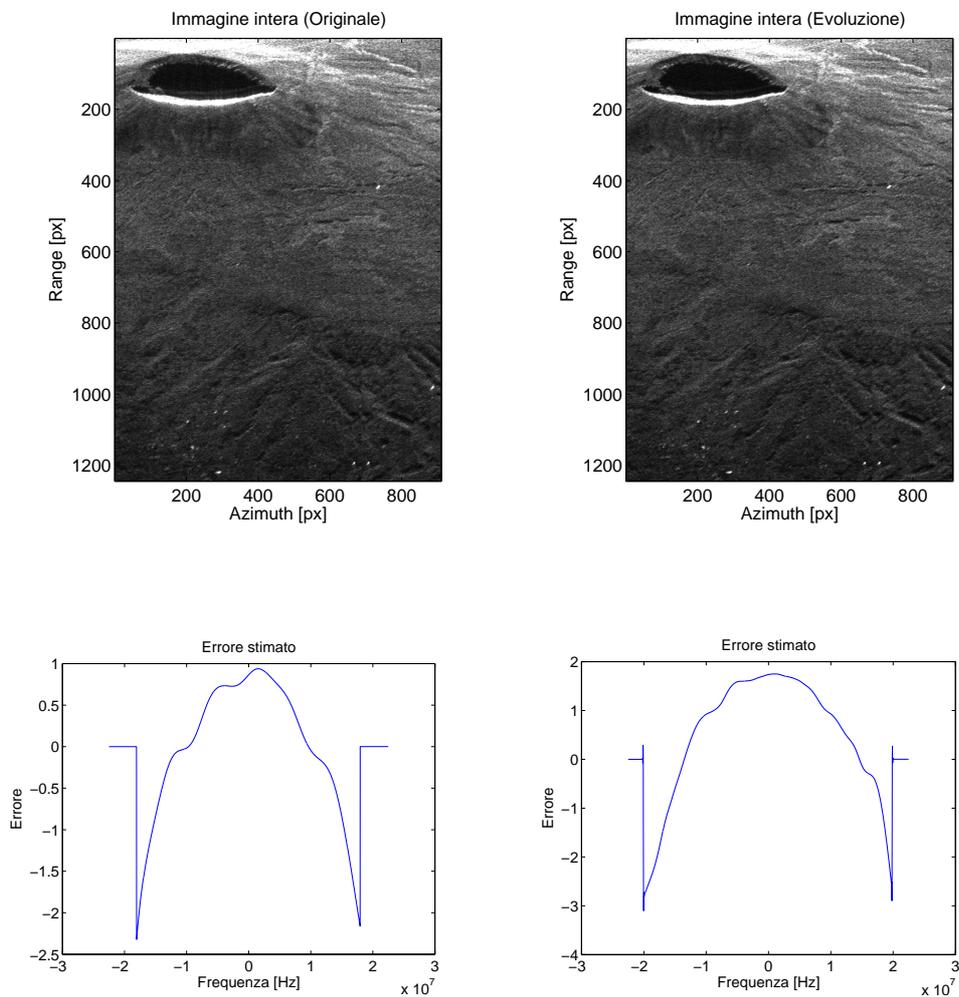
3.2.1 Il Vesuvio: Raw

Immagine non elaborata

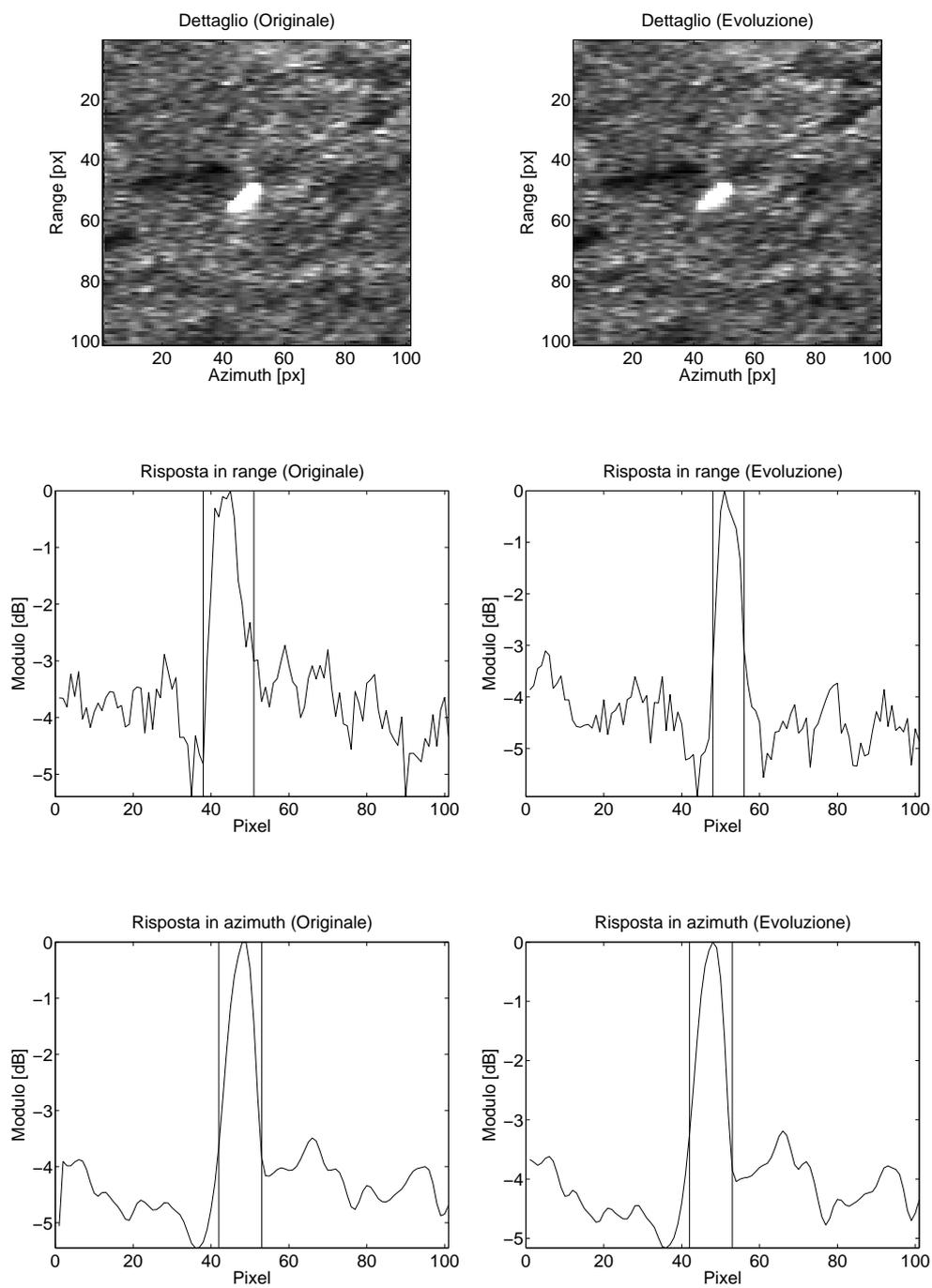


Entropia	-6.86523e+08
Risoluzione in range [px]	12
Risoluzione in azimuth [px]	11

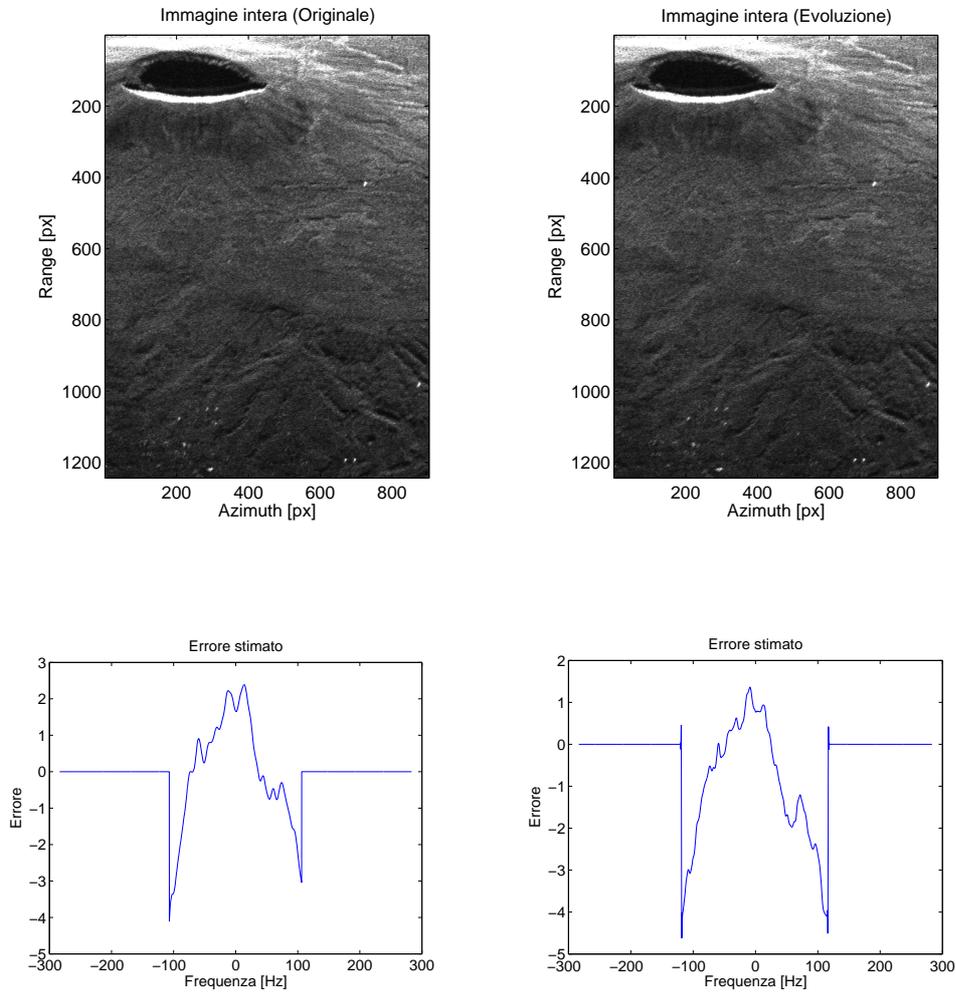
Range PGA



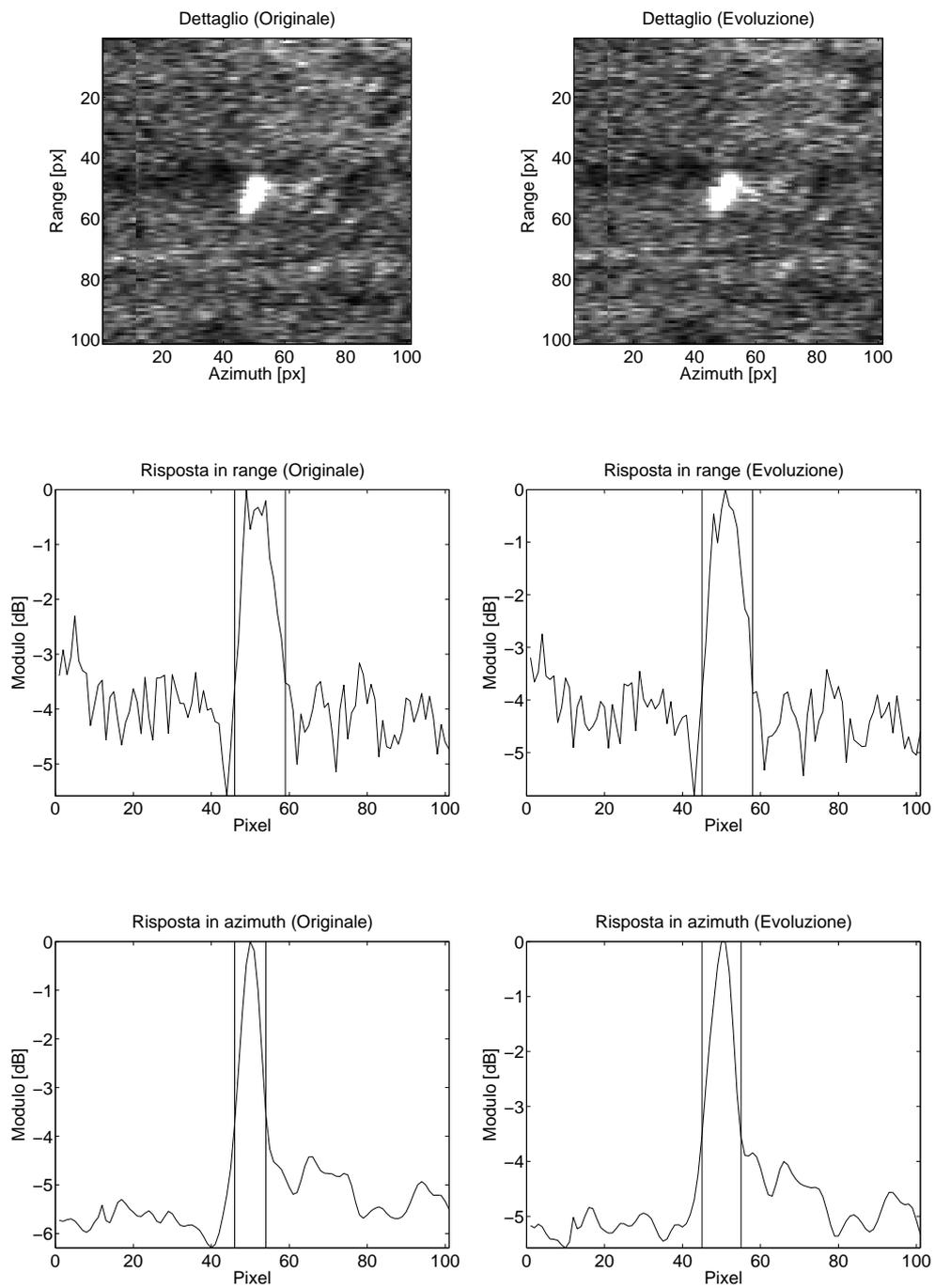
	Originale	Evoluzione
Entropia	-6.88141e+08	-6.8469e+08
Risoluzione in range [px]	12	8
Risoluzione in azimuth [px]	11	11
Tempo di esecuzione [s]	10.00	7.21



Azimuth PGA

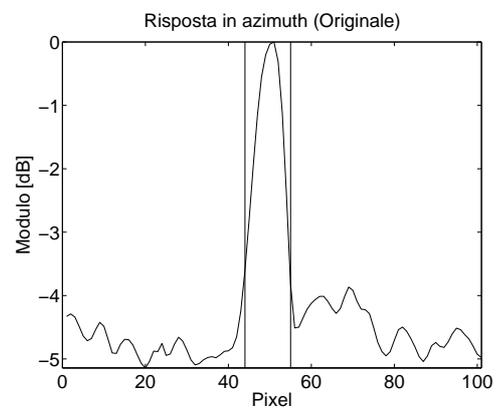
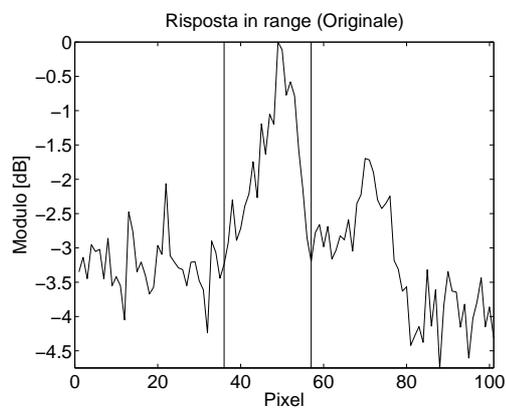
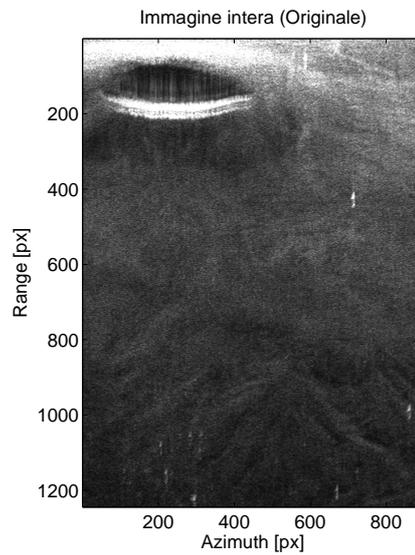


	Originale	Evoluzione
Entropia	-6.84733e+08	-6.85711e+08
Risoluzione in range [px]	13	13
Risoluzione in azimuth [px]	8	10
Tempo di esecuzione [s]	27.50	7.23



3.2.2 Il Vesuvio: Range Perturbed

Immagine non elaborata



Entropia

-7.04468e+08

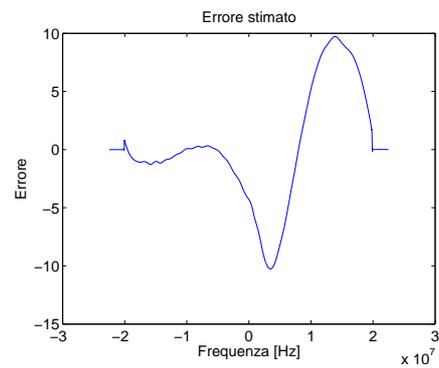
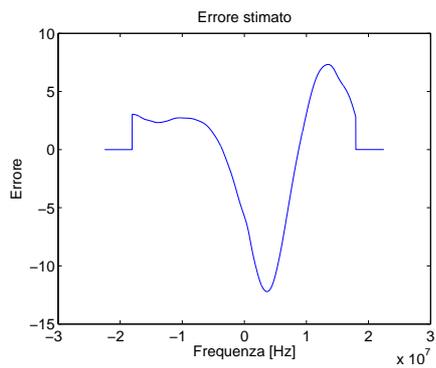
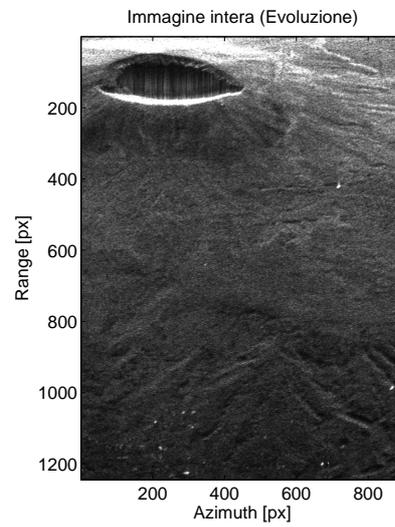
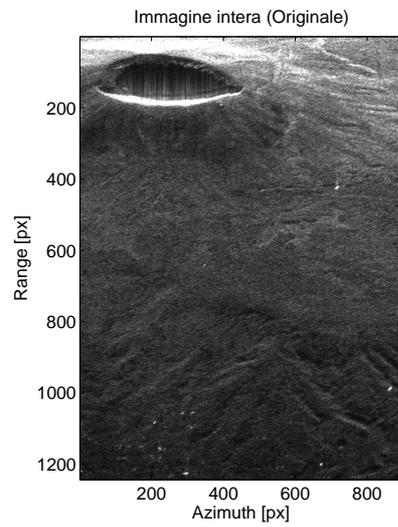
Risoluzione in range [px]

21

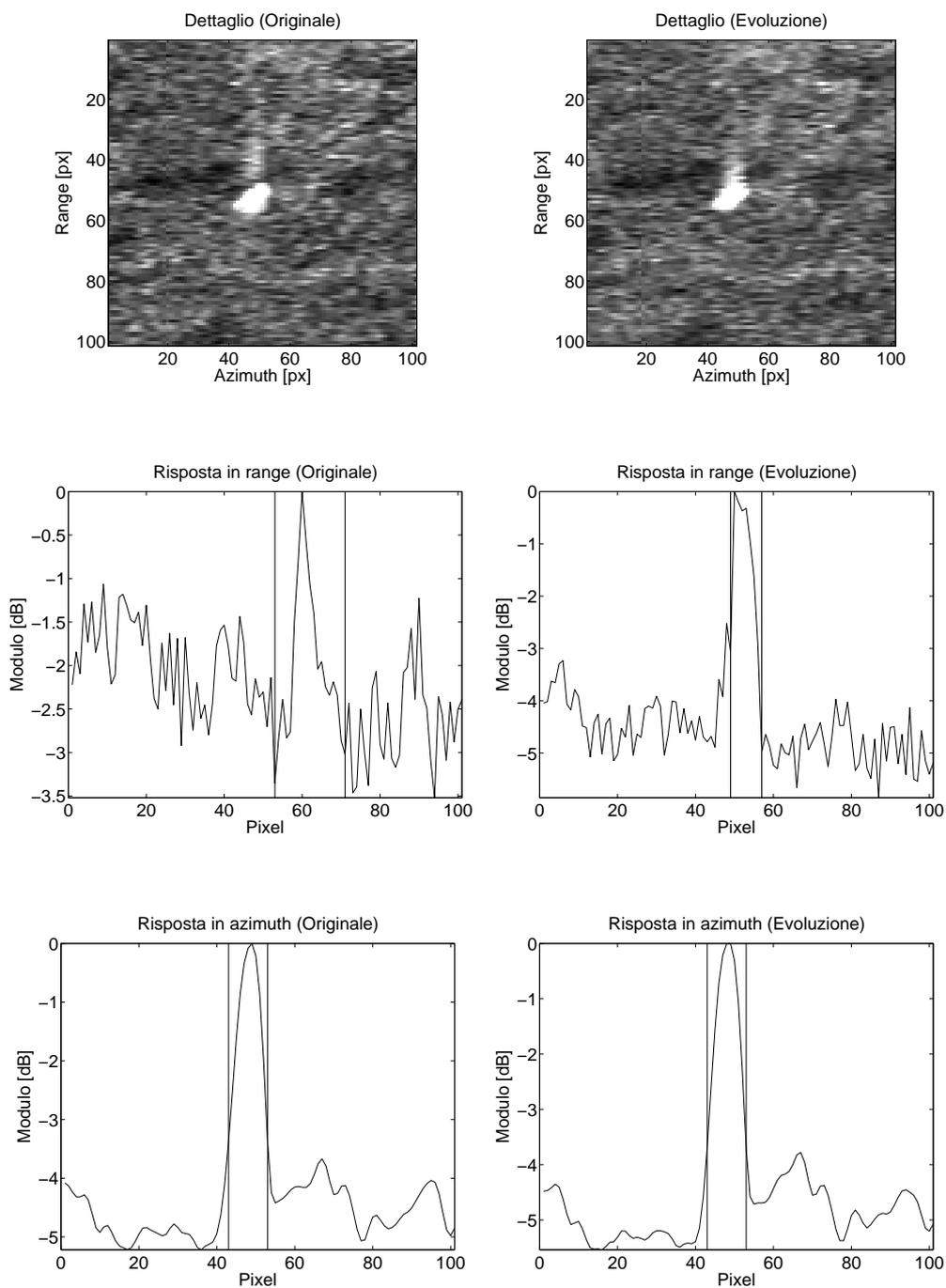
Risoluzione in azimuth [px]

11

Range PGA

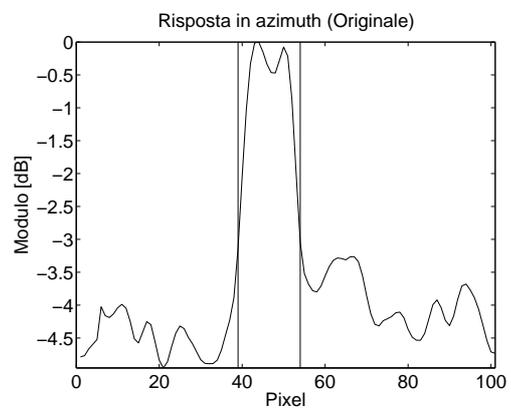
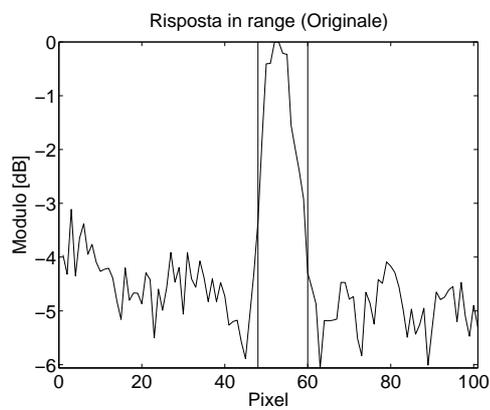
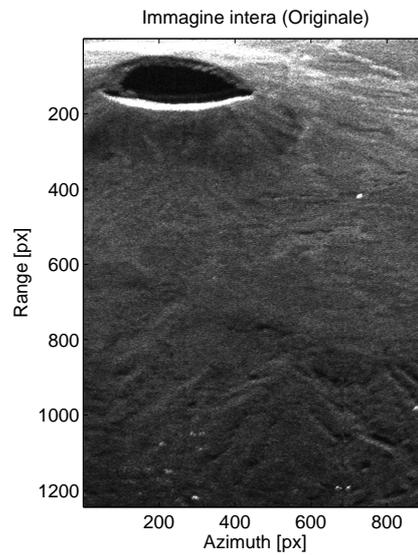


	Originale	Evoluzione
Entropia	-6.90882e+08	-6.87106e+08
Risoluzione in range [px]	10	8
Risoluzione in azimuth [px]	10	10
Tempo di esecuzione [s]	15.28	7.99



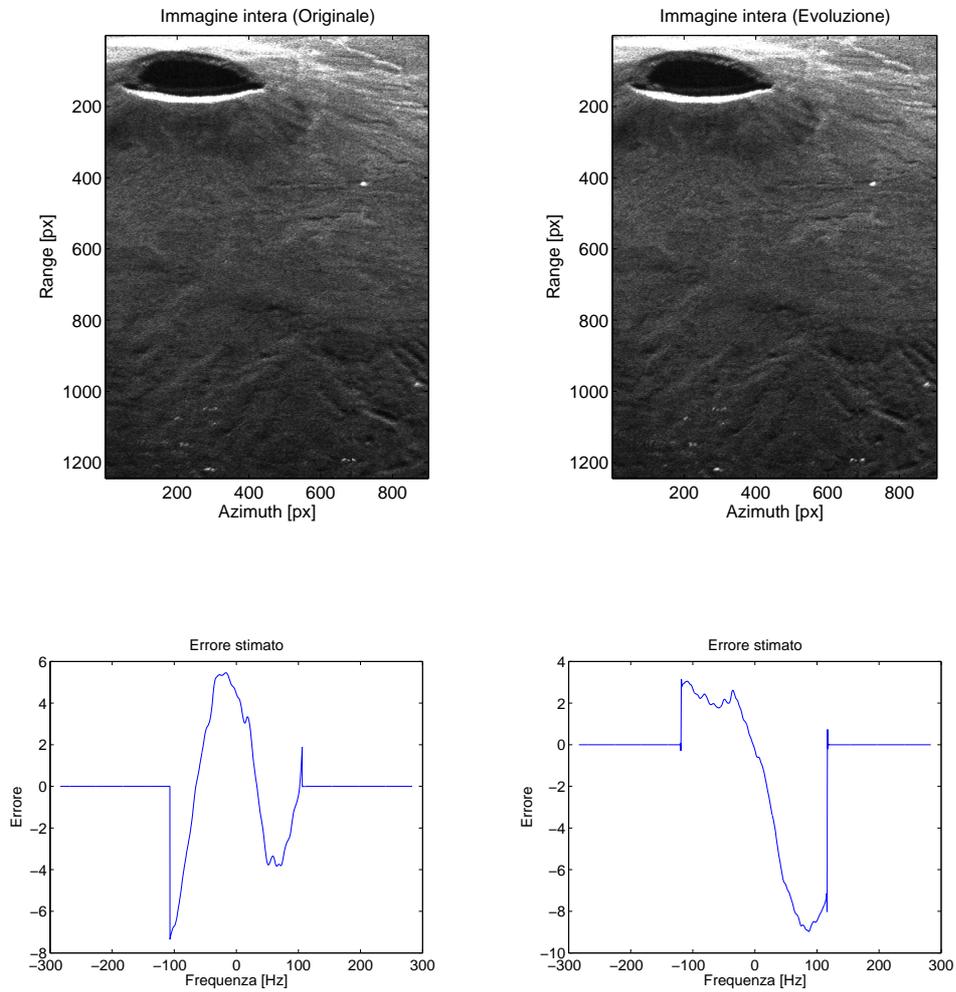
3.2.3 Il Vesuvio: Azimuth Perturbed

Immagine non elaborata

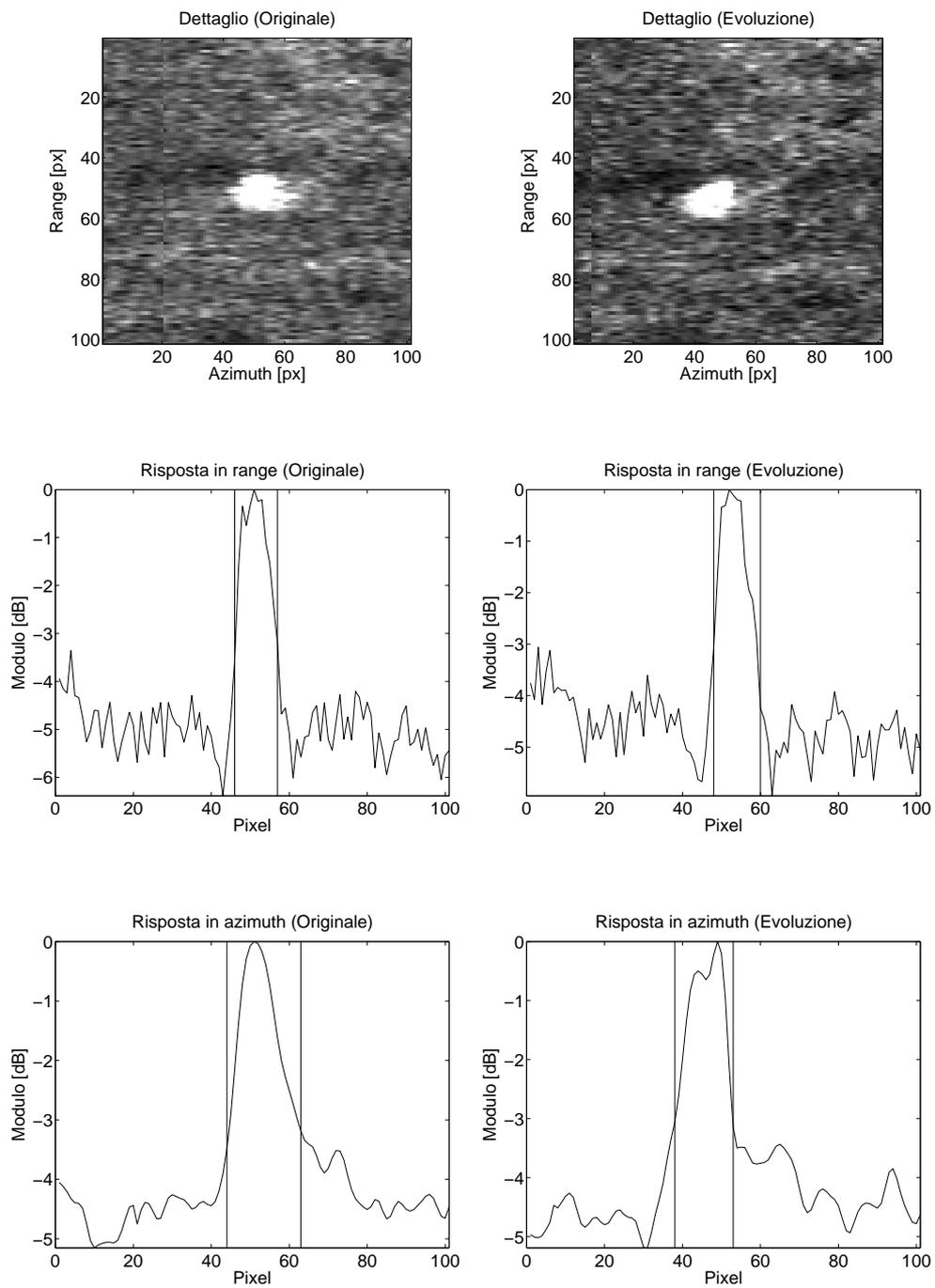


Entropia	-6.88767e+08
Risoluzione in range [px]	12
Risoluzione in azimuth [px]	15

Azimtuh PGA

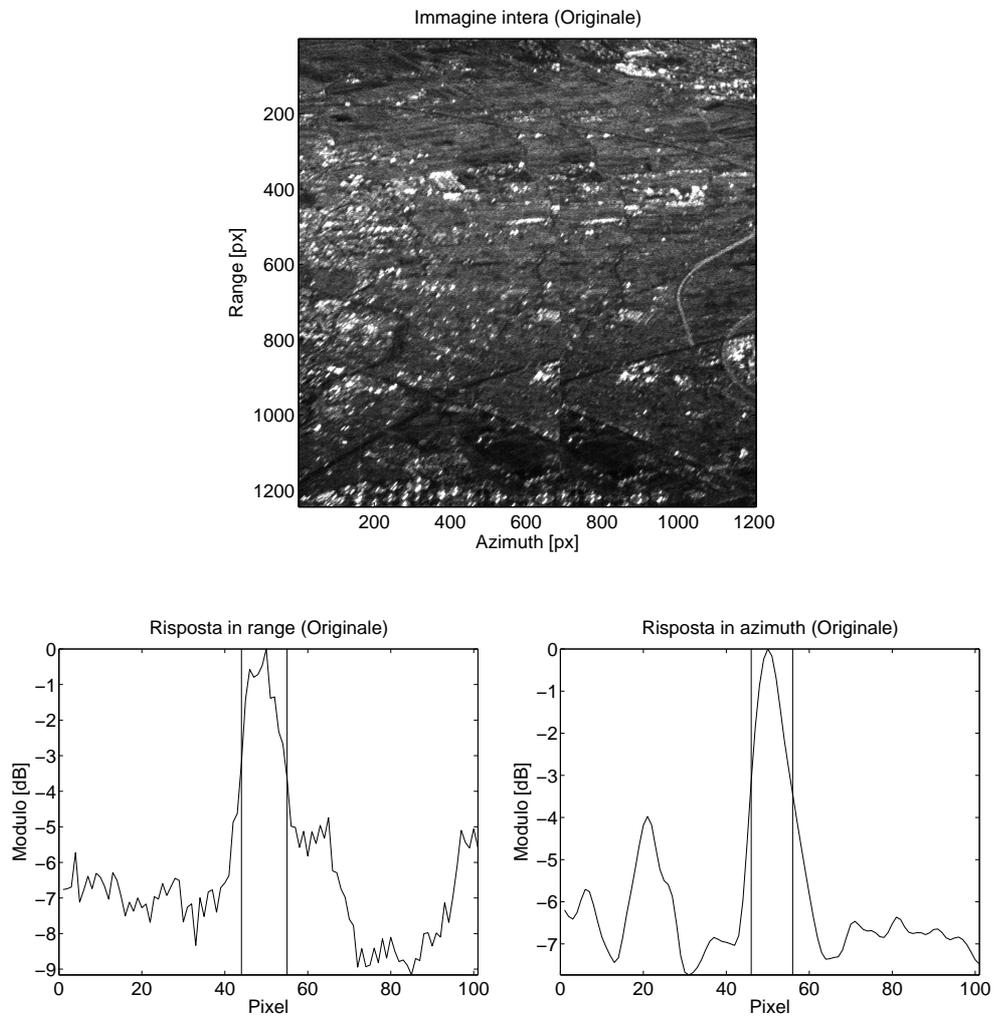


	Originale	Evoluzione
Entropia	-6.81883e+08	-6.87843e+08
Risoluzione in range [px]	11	12
Risoluzione in azimuth [px]	19	15
Tempo di esecuzione [s]	29.46	5.99



3.2.4 Somma Vesuviana: Raw

Immagine non elaborata



Entropia

-1.40034e+08

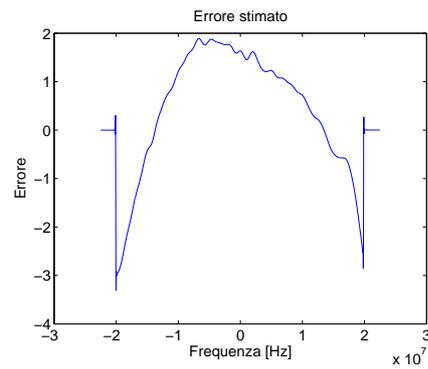
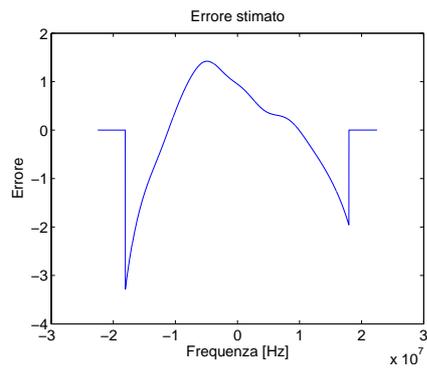
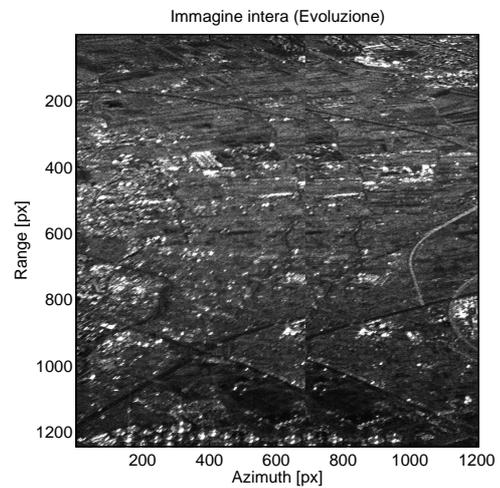
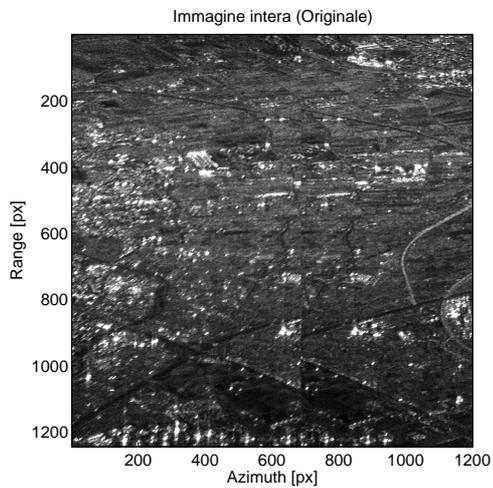
Risoluzione in range [px]

11

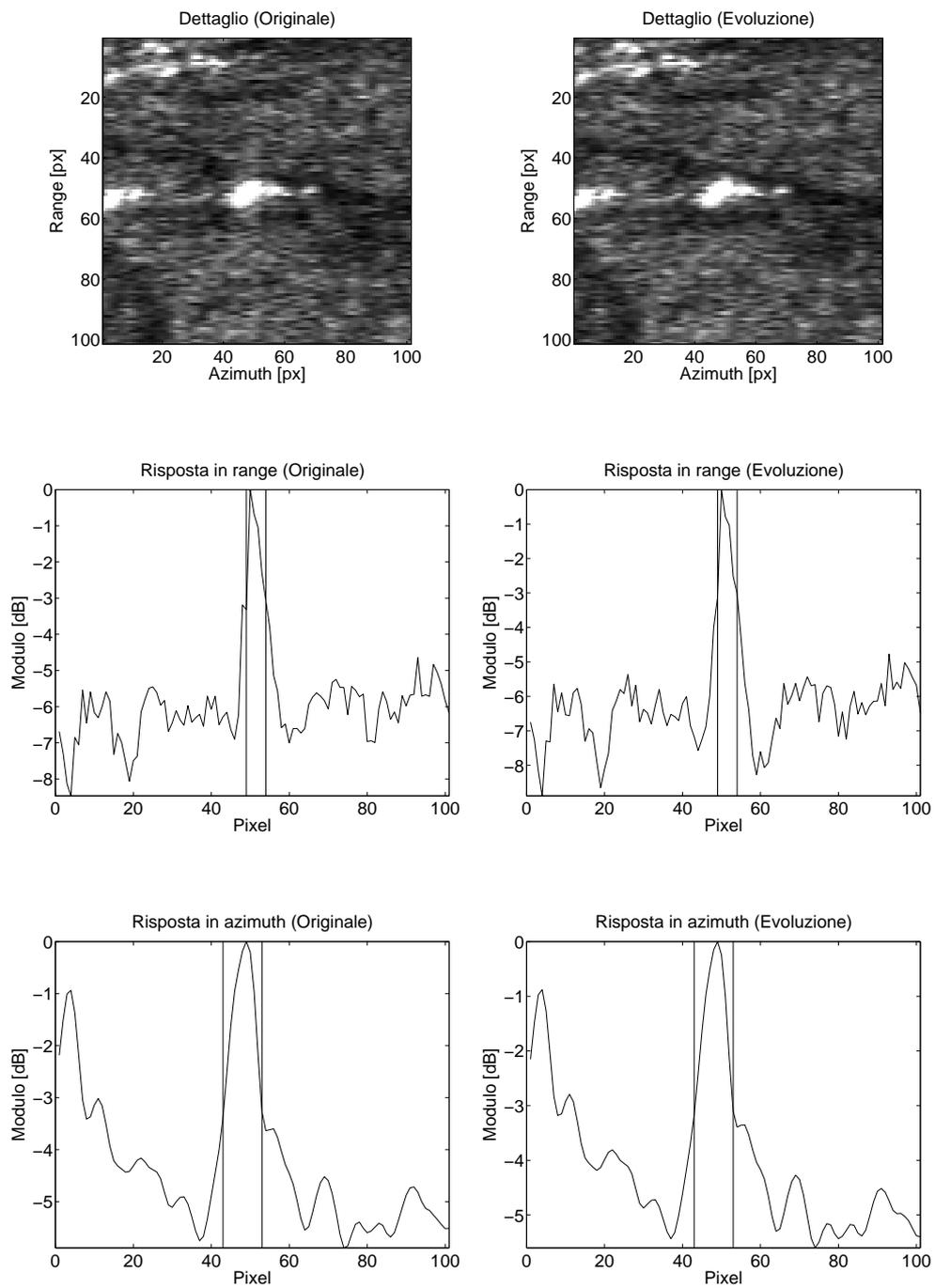
Risoluzione in azimuth [px]

10

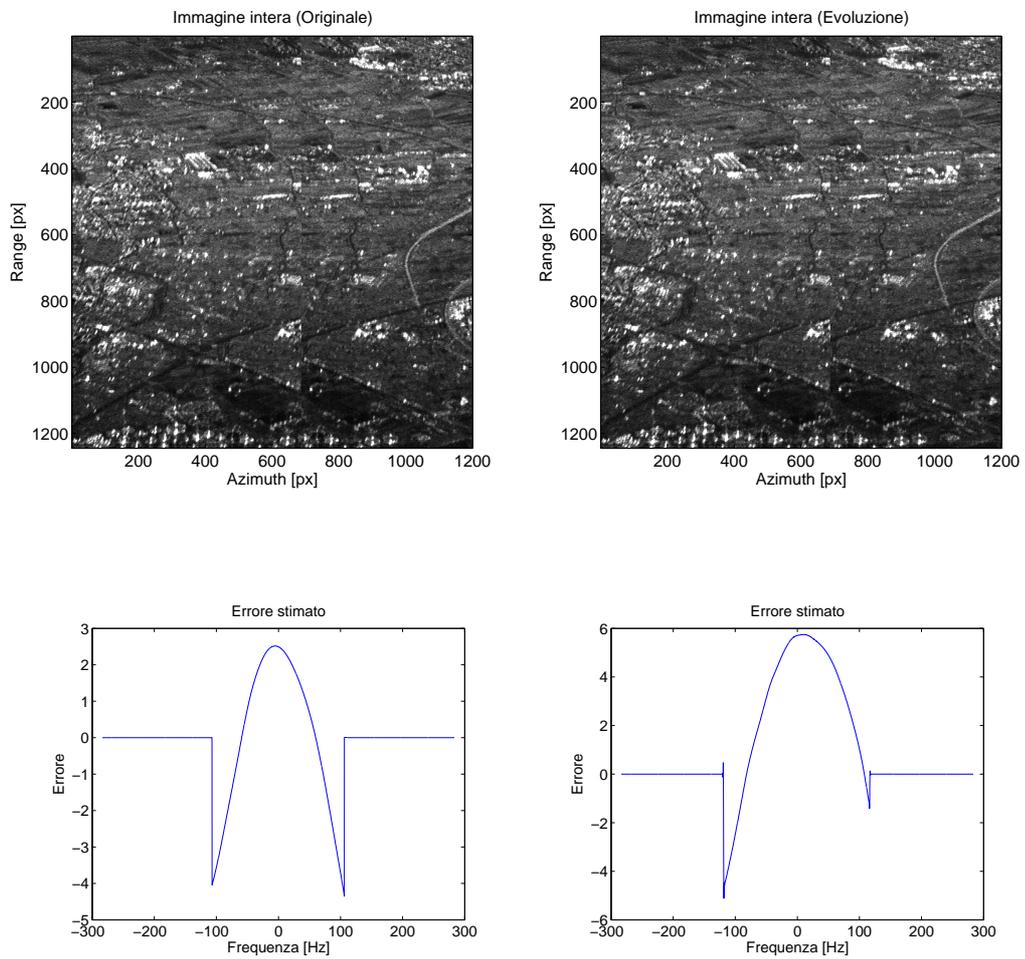
Range PGA



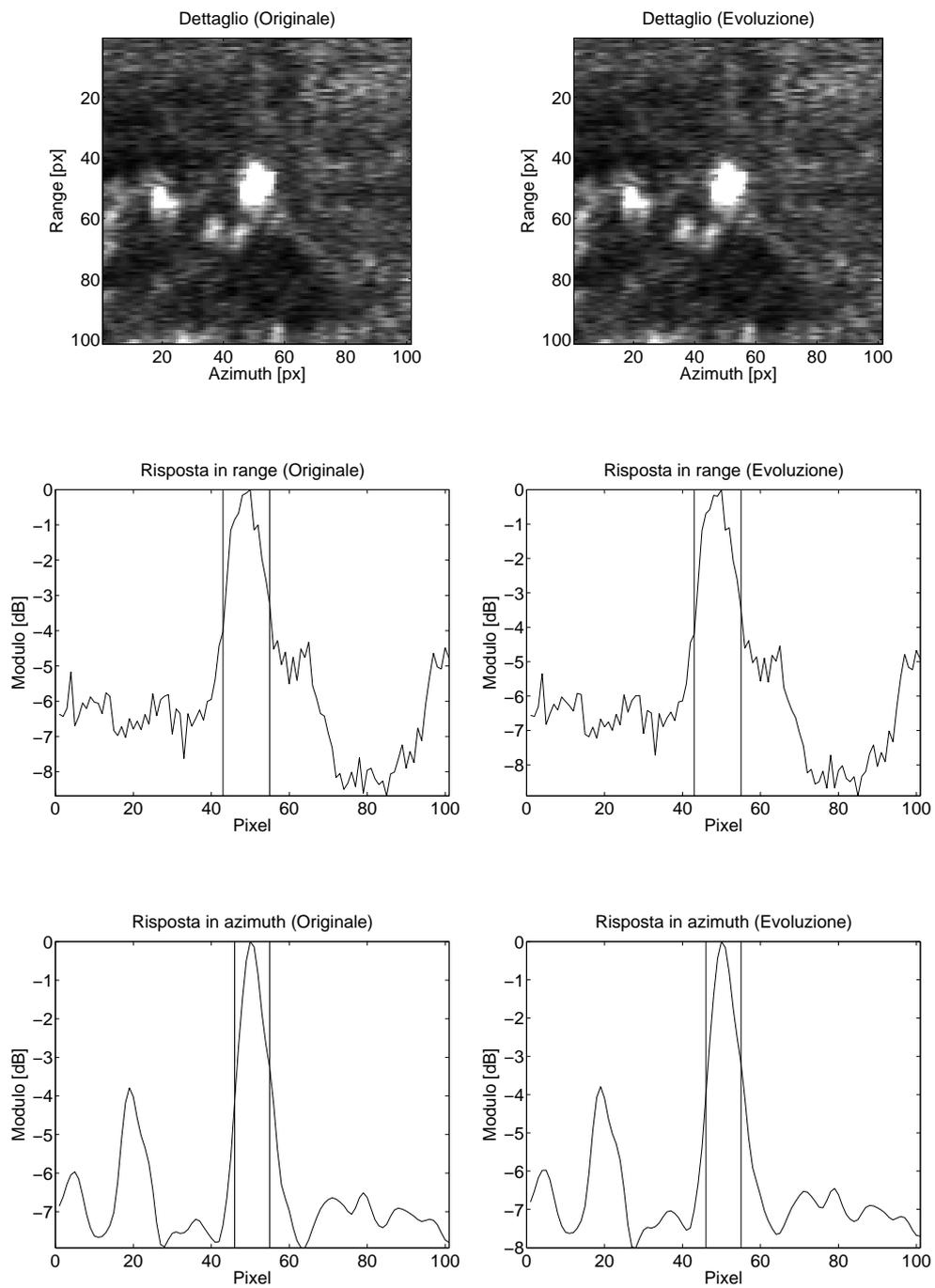
	Originale	Evoluzione
Entropia	-1.39354e+09	-1.37101e+09
Risoluzione in range [px]	5	5
Risoluzione in azimuth [px]	10	10
Tempo di esecuzione [s]	5.76	9.03



Azimuth PGA

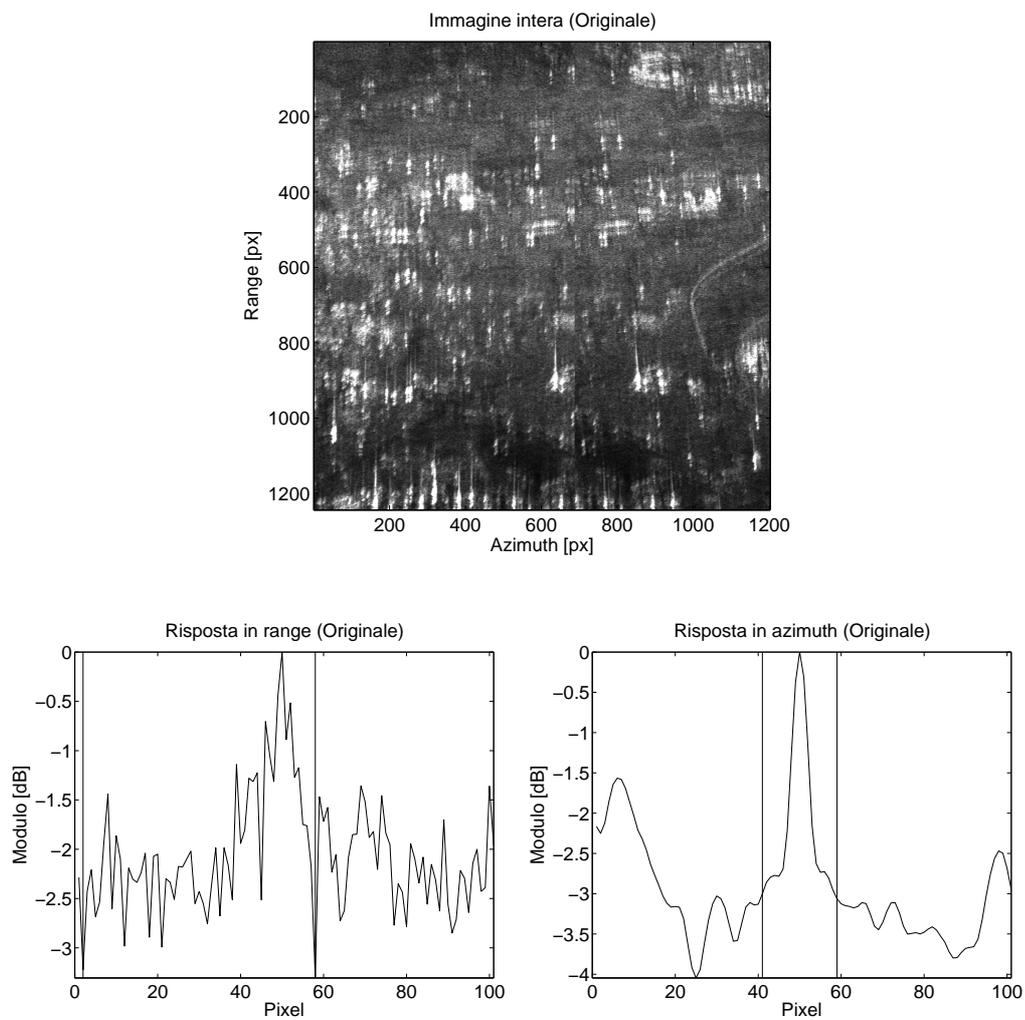


	Originale	Evoluzione
Entropia	-1.386e+09	-1.39496e+09
Risoluzione in range [px]	12	12
Risoluzione in azimuth [px]	9	9
Tempo di esecuzione [s]	12.59	8.91



3.2.5 Somma Vesuviana: Range Perturbed

Immagine non elaborata



Entropia

-1.81934e+09

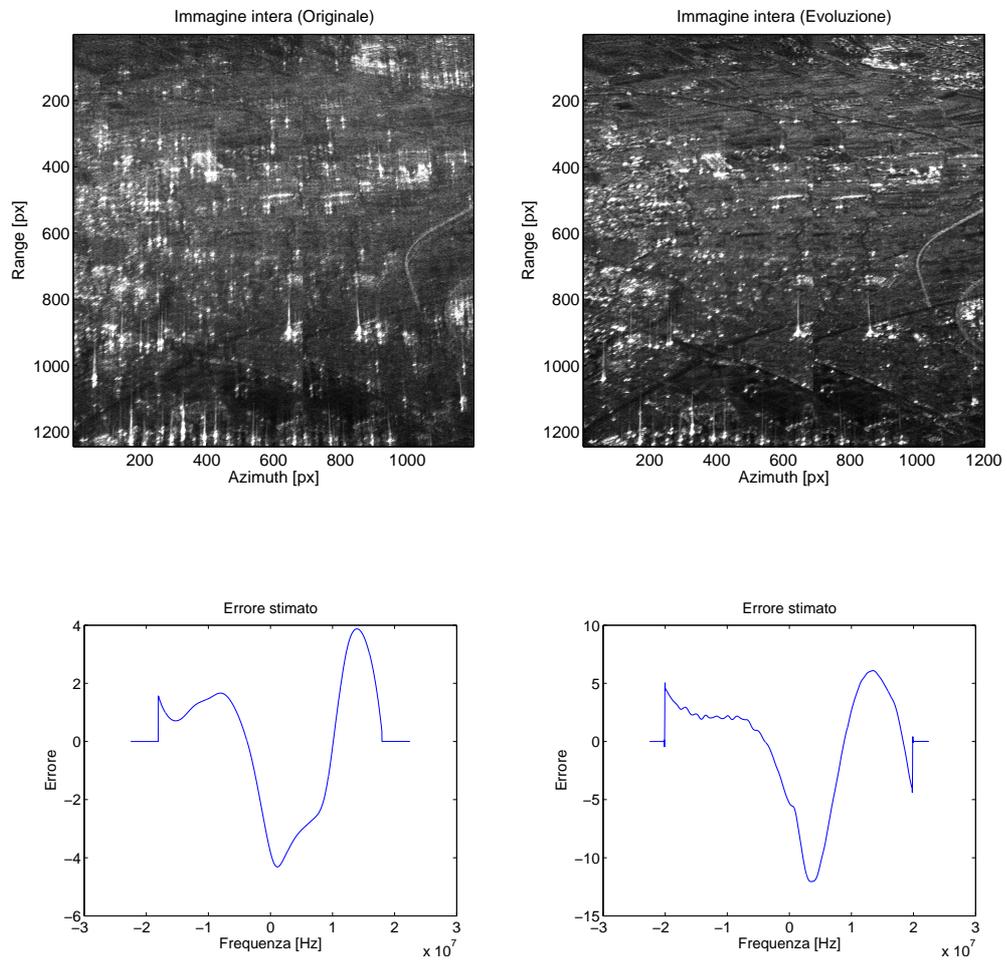
Risoluzione in range [px]

56

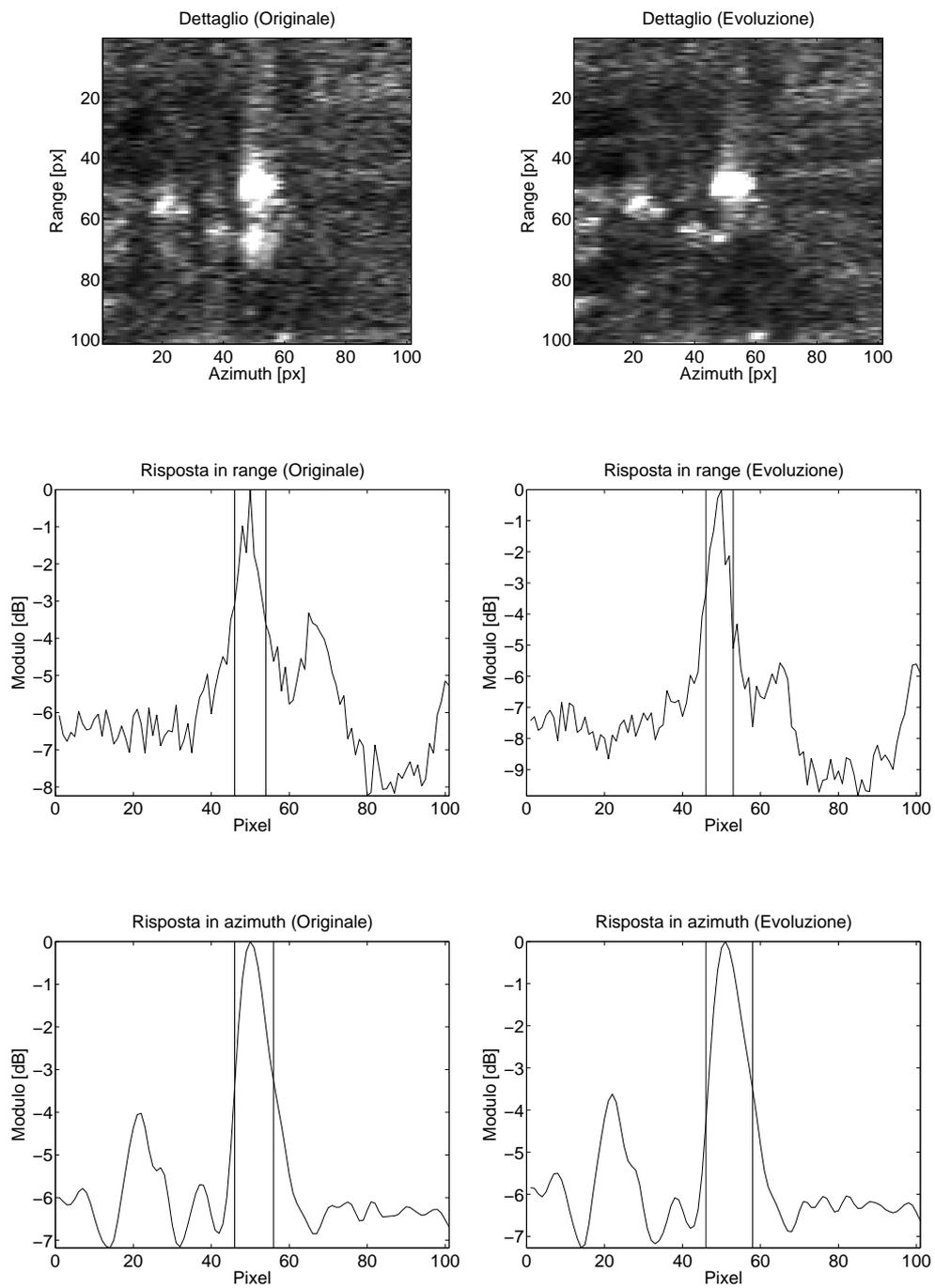
Risoluzione in azimuth [px]

18

Range PGA

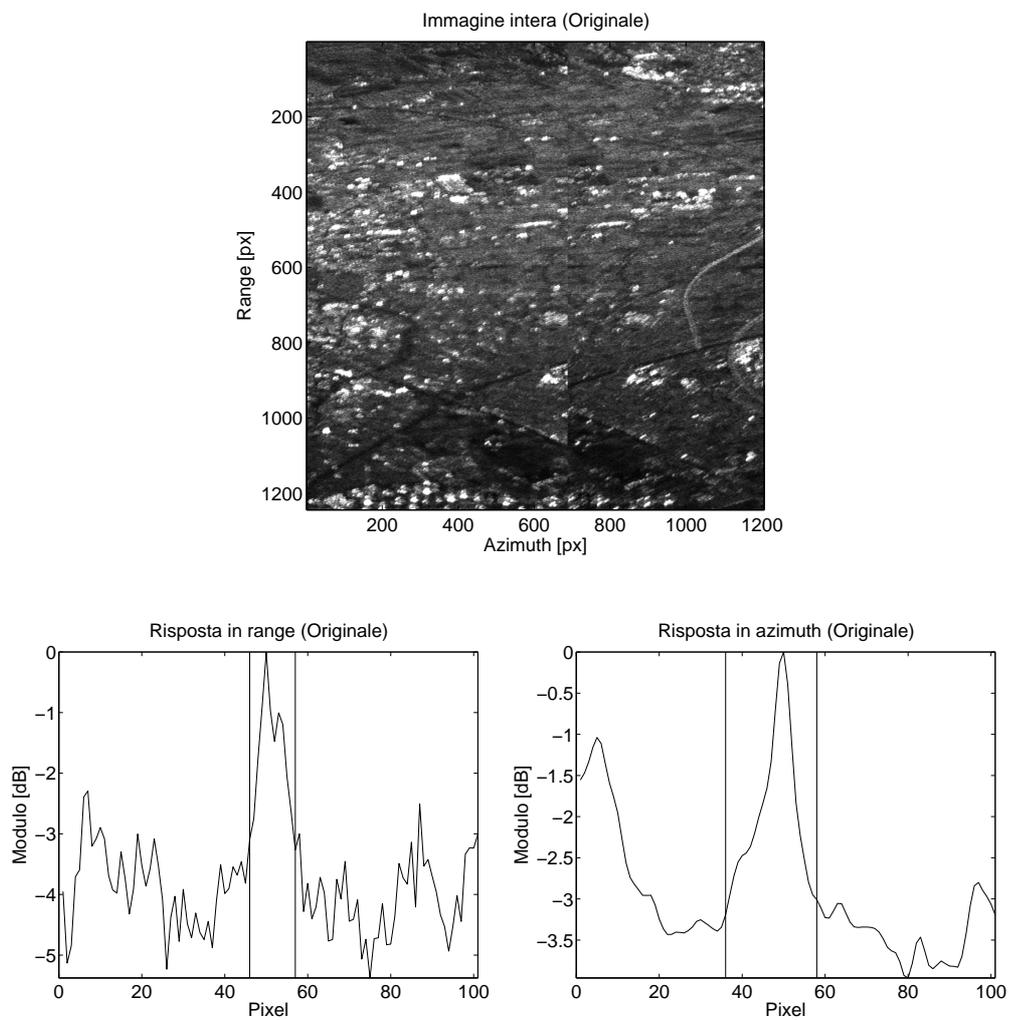


	Originale	Evoluzione
Entropia	-1.47997e+09	-1.4157e+09
Risoluzione in range [px]	8	7
Risoluzione in azimuth [px]	10	12
Tempo di esecuzione [s]	19.90	7.04



3.2.6 Somma Vesuviana: Azimuth Perturbed

Immagine non elaborata



Entropia

-1.44e+09

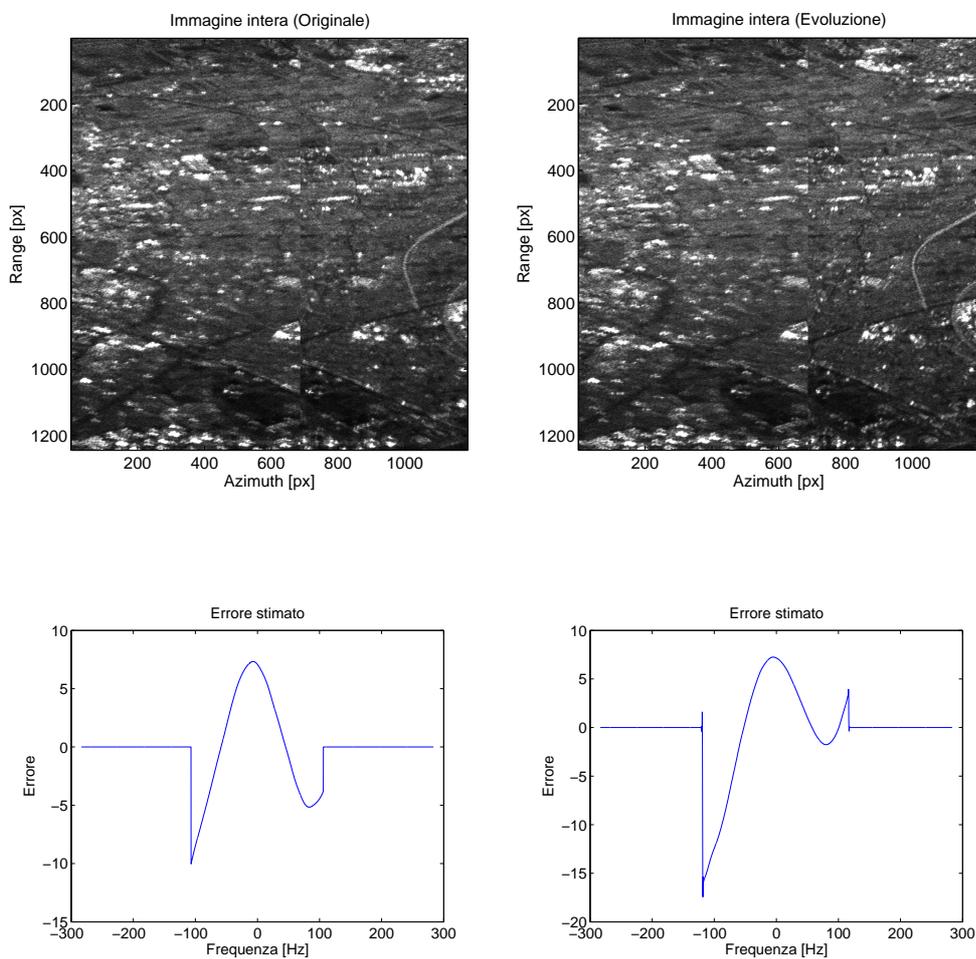
Risoluzione in range [px]

11

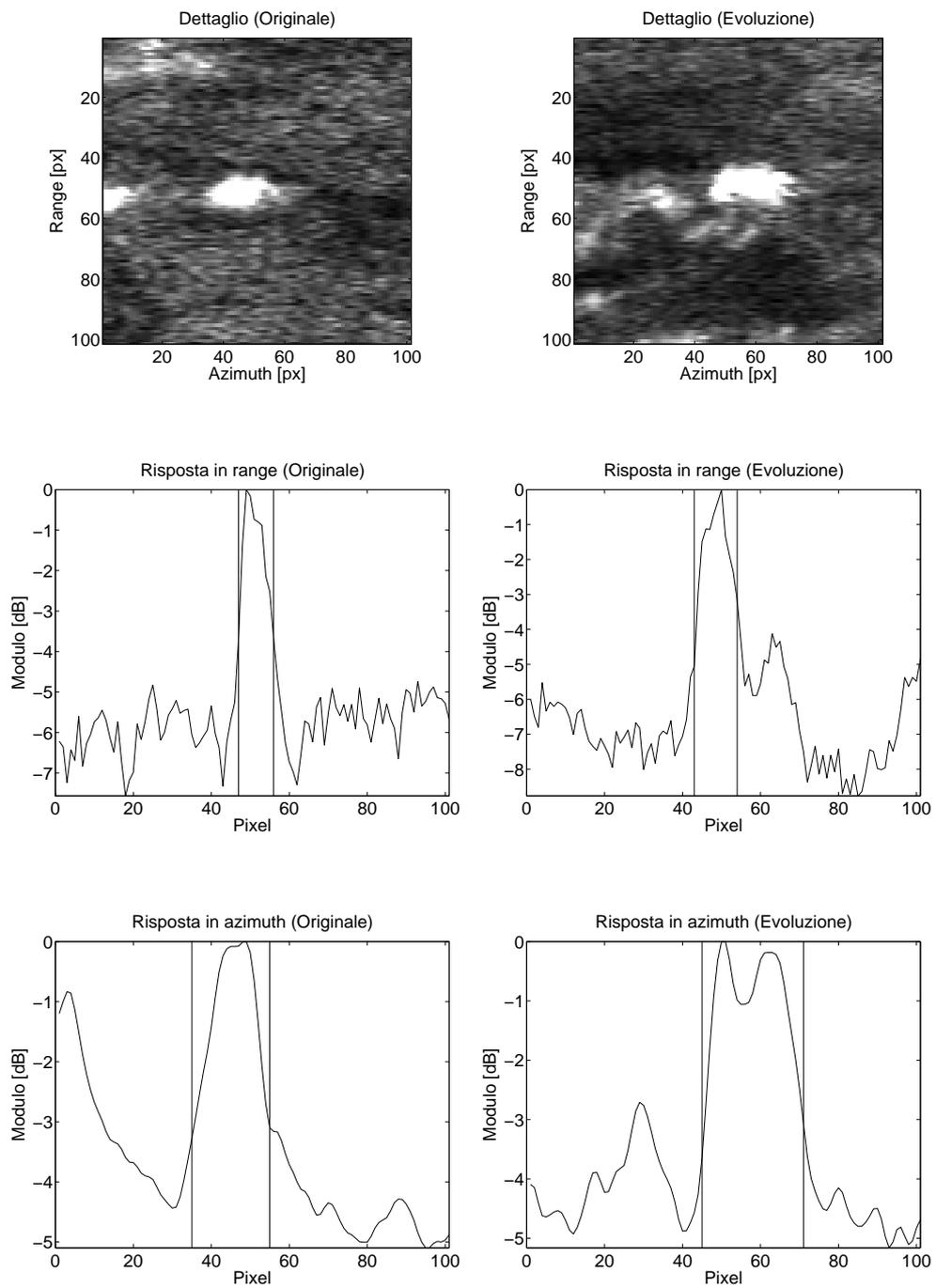
Risoluzione in azimuth [px]

22

Azimutuh PGA



	Originale	Evoluzione
Entropia	-1.45605e+09	-1.45891e+09
Risoluzione in range [px]	9	11
Risoluzione in azimuth [px]	20	26
Tempo di esecuzione [s]	25.33	7.39



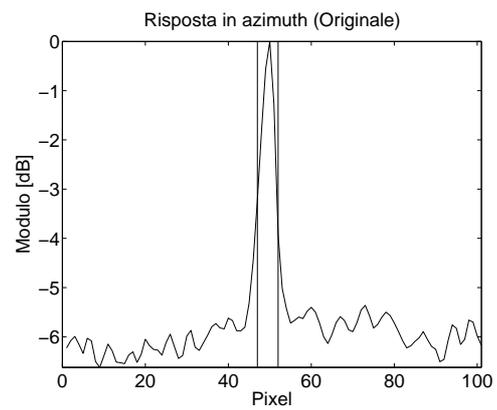
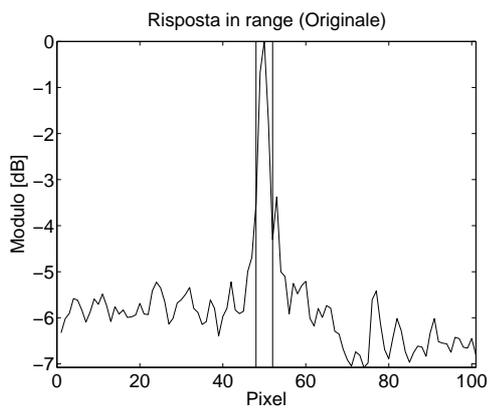
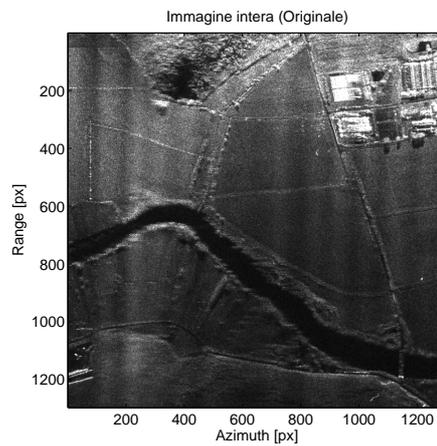
3.3 MINISAR Data Set

I dati MINISAR utilizzati per la realizzazione dei test di focalizzazione fanno riferimento al volo del 31 Marzo 2006 sulla piana di Caiazzo. La scansione originale è lunga circa 4 km e si estende lungo il Fiume Volturno in direzione Est-Ovest. Le scene estratte riguardano uno stabilimento industriale immediatamente a ridosso del fiume ed alcune coltivazioni agricole nelle vicinanze.

Le schede delle pagine seguenti presentano in forma schematica i risultati ottenuti su queste scene applicando il PGA all'immagine originale ed a quelle perturbate con errore polinomiale rispettivamente in azimuth e range. Per ciascuno dei tre gruppi, in prima pagina sono mostrate le condizioni dell'immagine non elaborata, con specifico riferimento all'andamento del modulo in range ed azimuth, alla risoluzione spaziale nelle due direzioni ed all'entropia. Nelle pagine successive, le stesse informazioni, arricchite dall'errore stimato e dal dettaglio del punto su cui si è calcolata la risoluzione, sono riportate per le immagini a valle dell'applicazione del PGA. Per convenzione, nella colonna di sinistra sono indicati i risultati dell'implementazione originale, mentre quella di destra è riservata alla nuova versione.

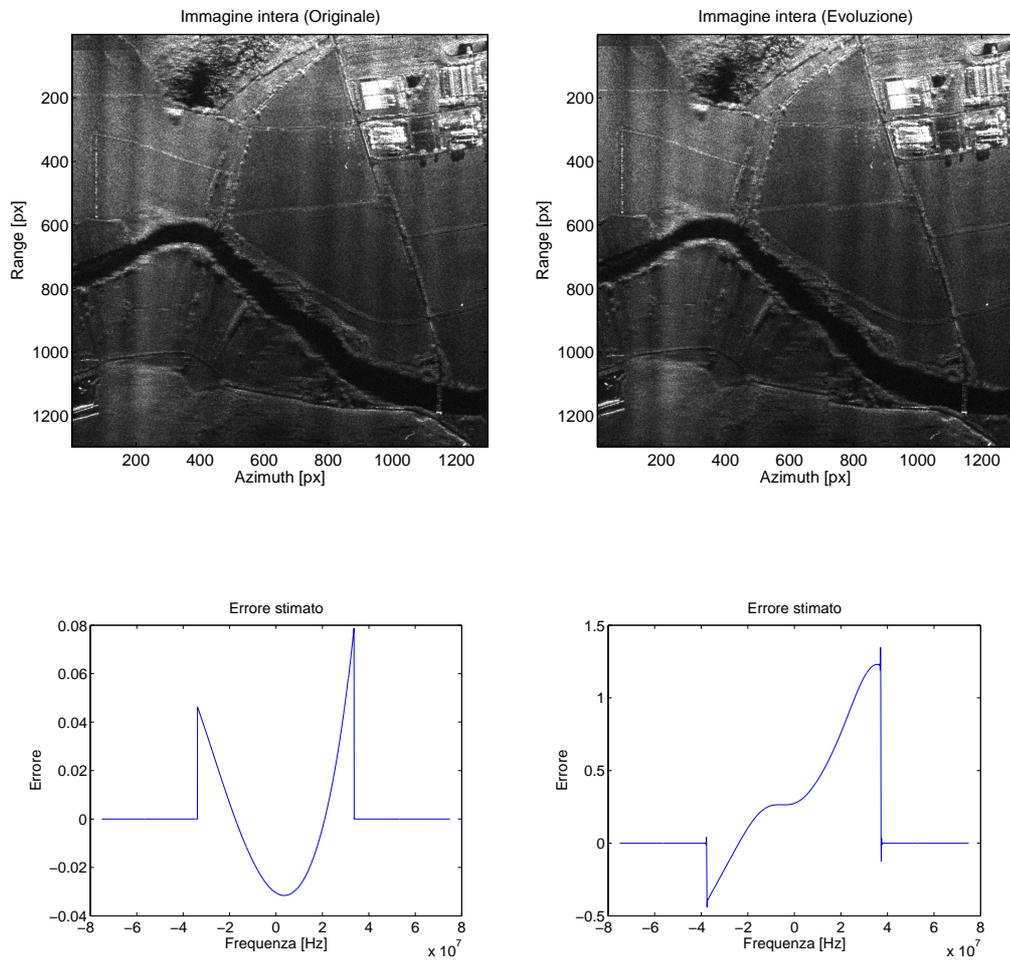
3.3.1 Stabilimento industriale: Raw

Immagine non elaborata

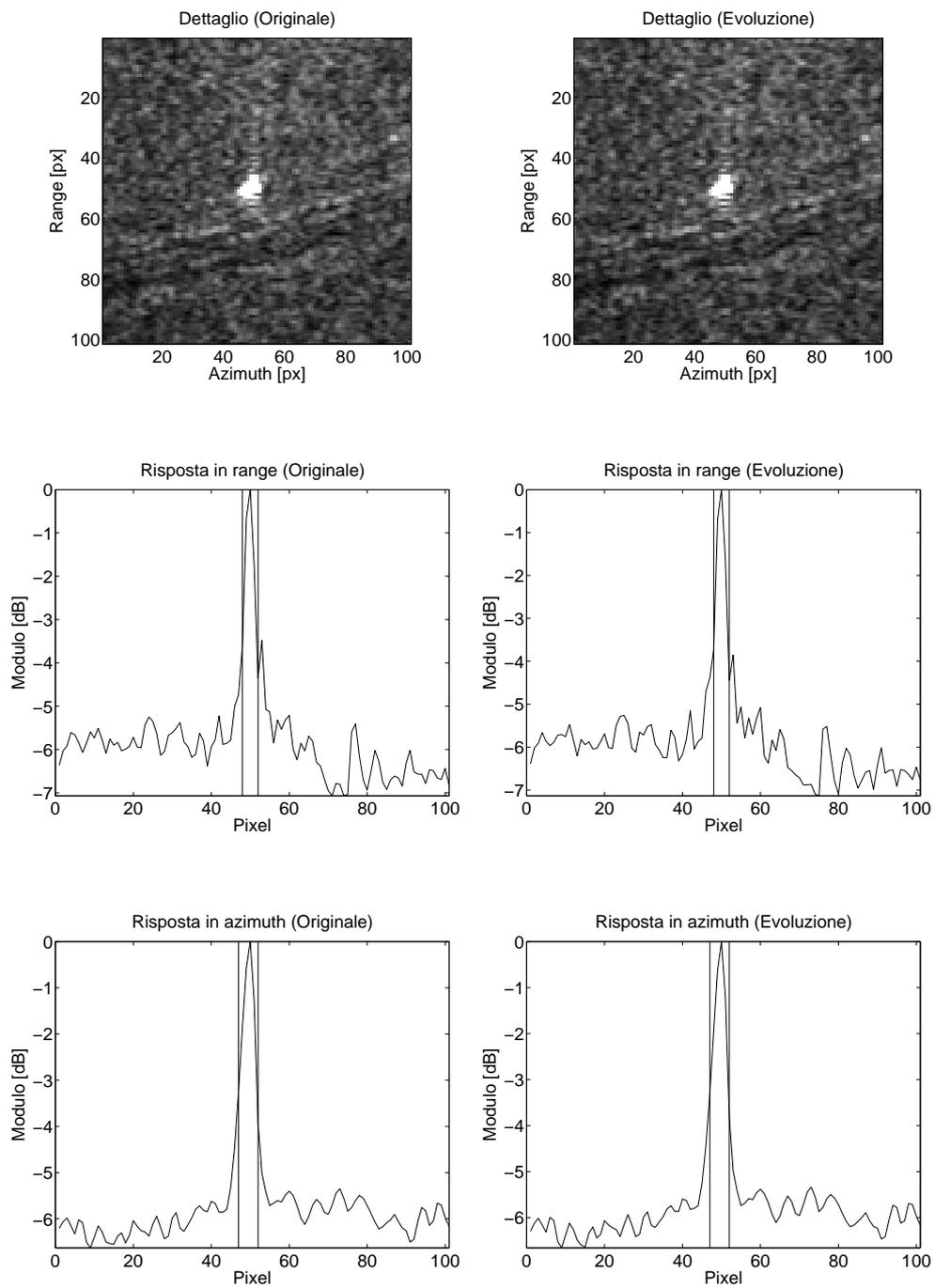


Entropia	-8.19177e+08
Risoluzione in range [px]	4
Risoluzione in azimuth [px]	5

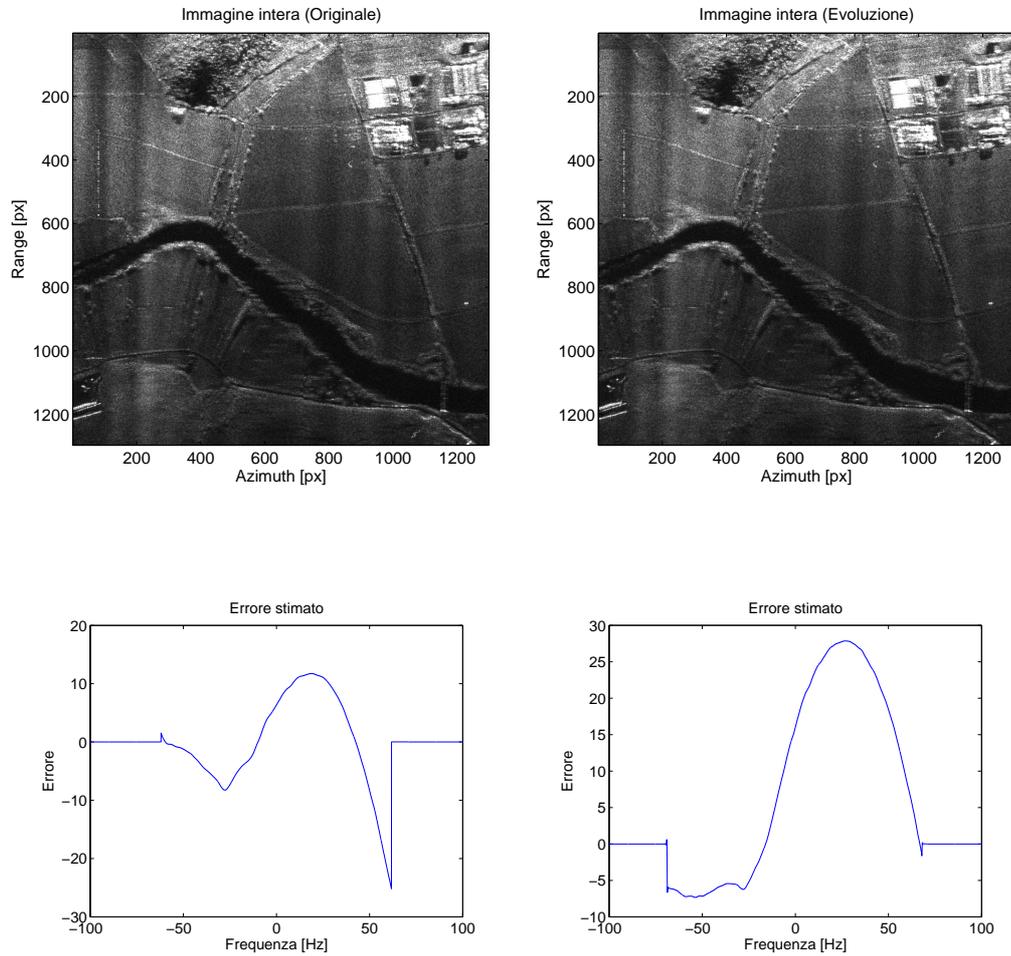
Range PGA



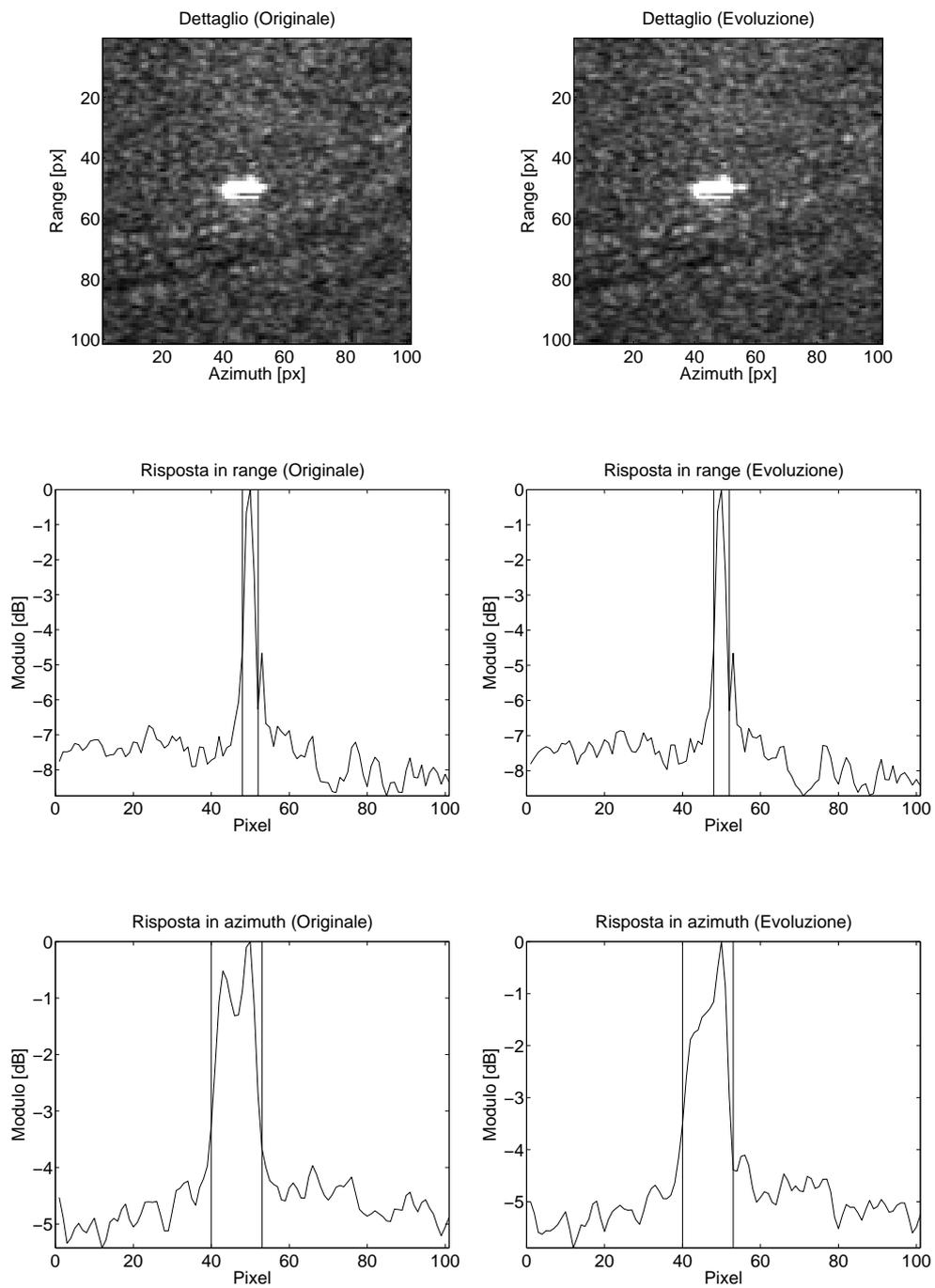
	Originale	Evoluzione
Entropia	-8.191e+08	-8.14208e+08
Risoluzione in range [px]	4	4
Risoluzione in azimuth [px]	5	5
Tempo di esecuzione [s]	3.95	9.40



Azimuth PGA

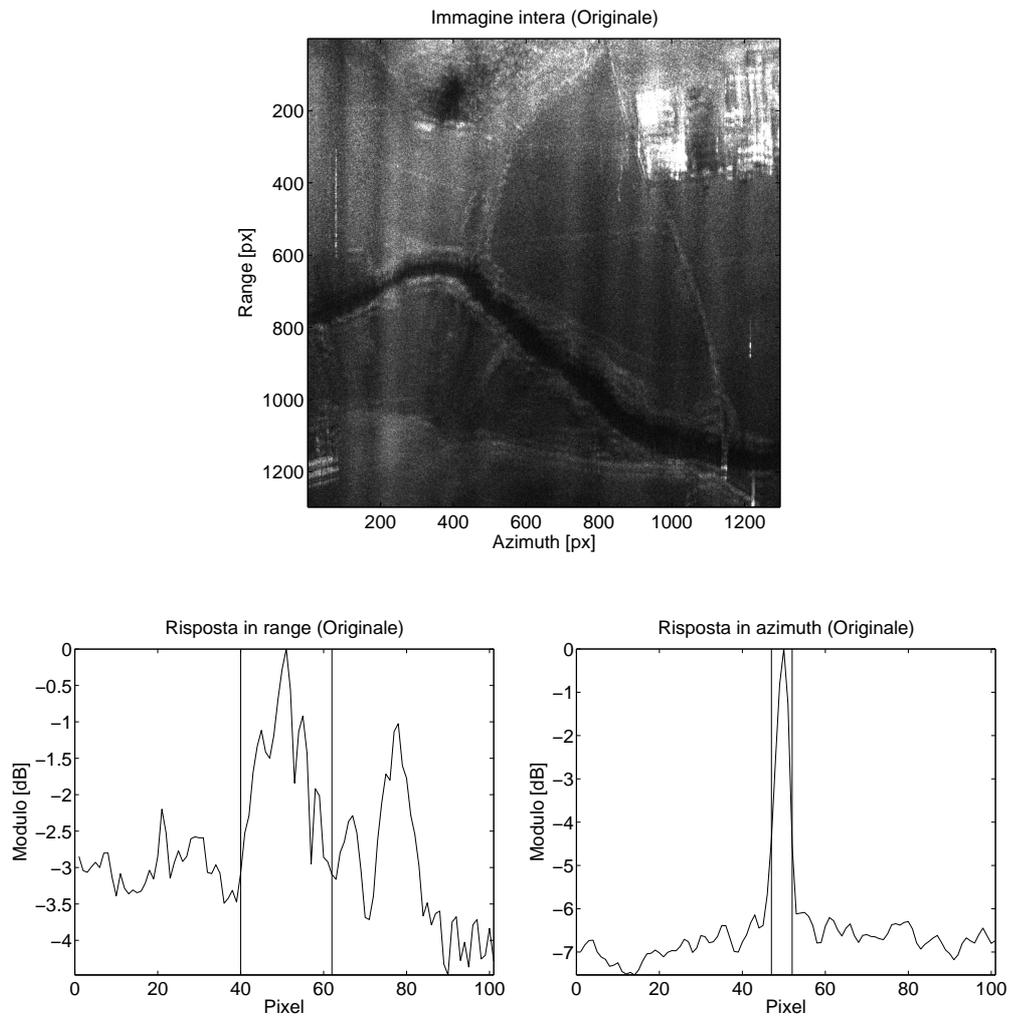


	Originale	Evoluzione
Entropia	-7.99827e+08	-8.05607e+08
Risoluzione in range [px]	13	13
Risoluzione in azimuth [px]	4	4
Tempo di esecuzione [s]	26.62	9.18



3.3.2 Stabilimento industriale: Range Perturbed

Immagine non elaborata



Entropia

-8.85035e+08

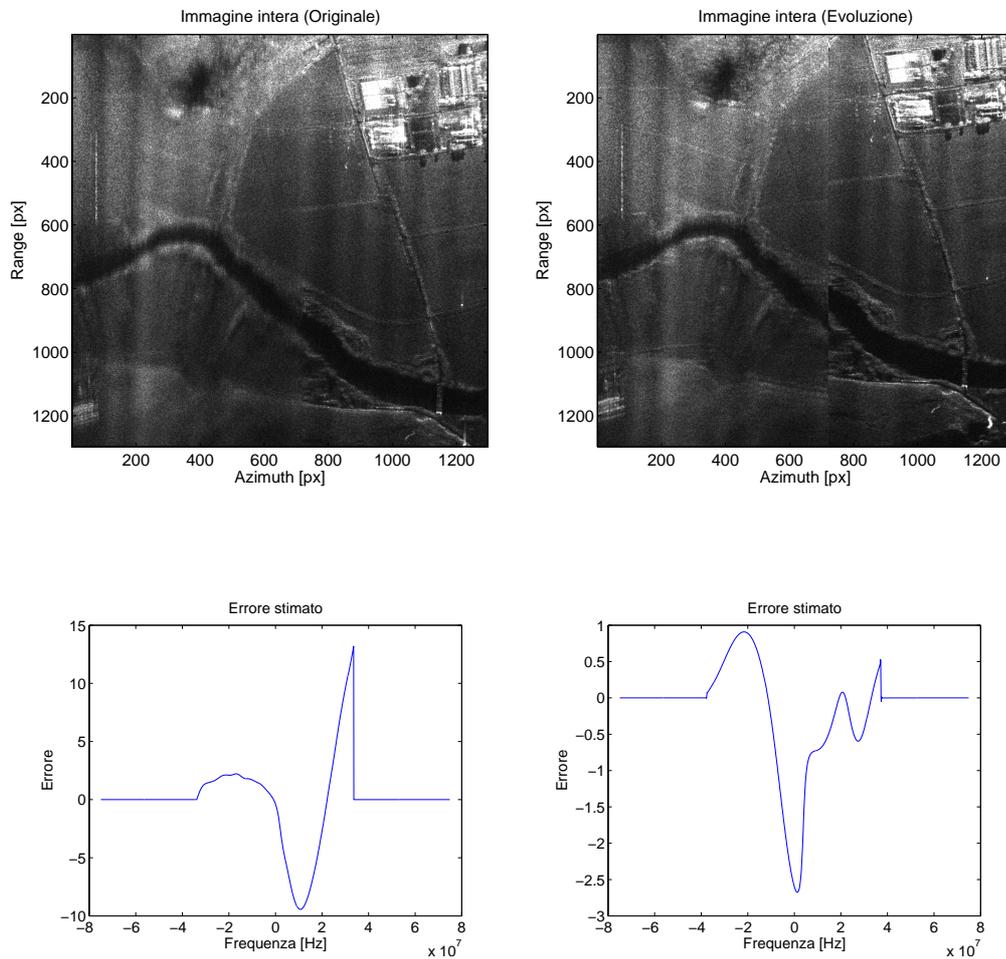
Risoluzione in range [px]

22

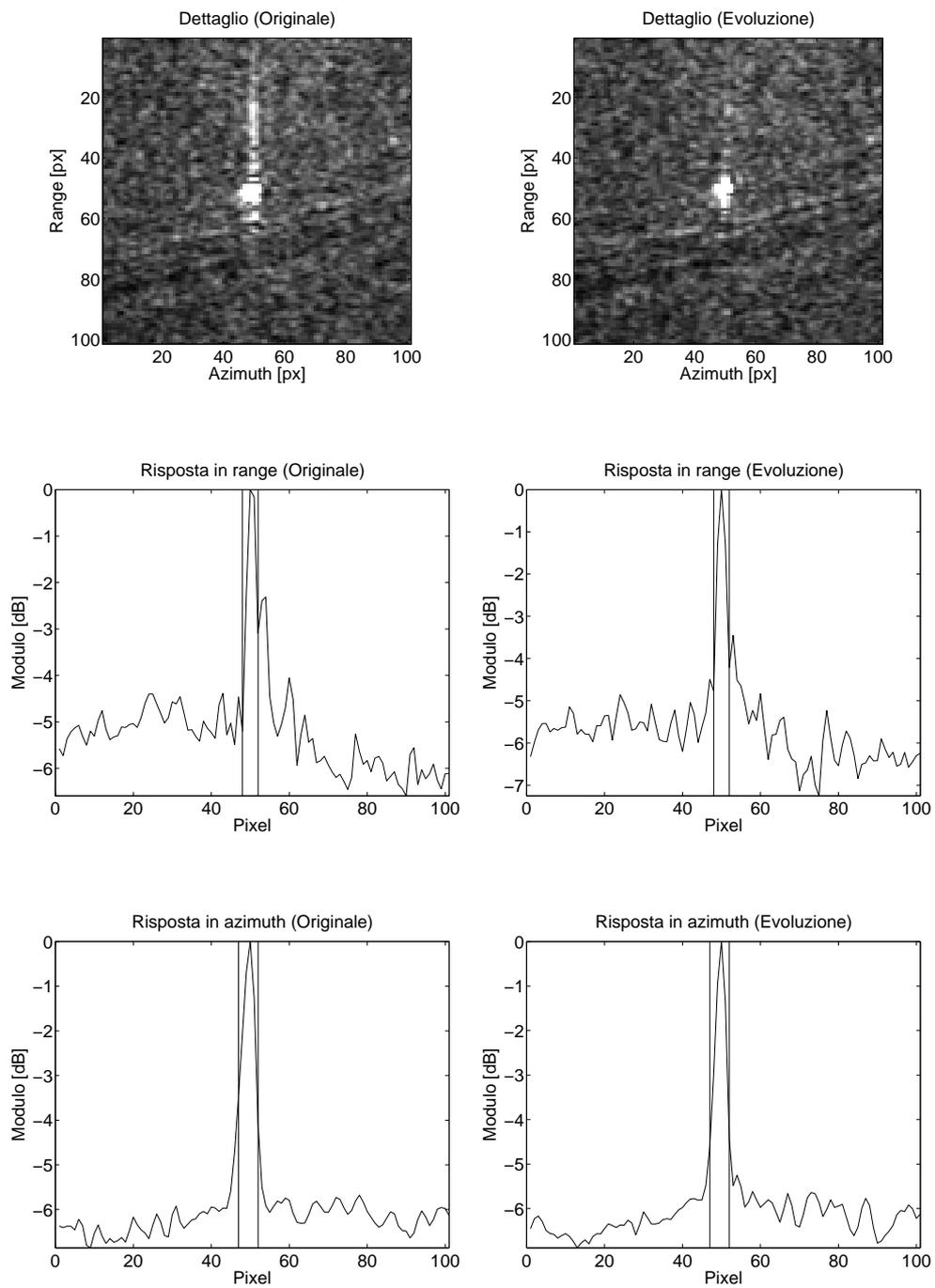
Risoluzione in azimuth [px]

5

Range PGA

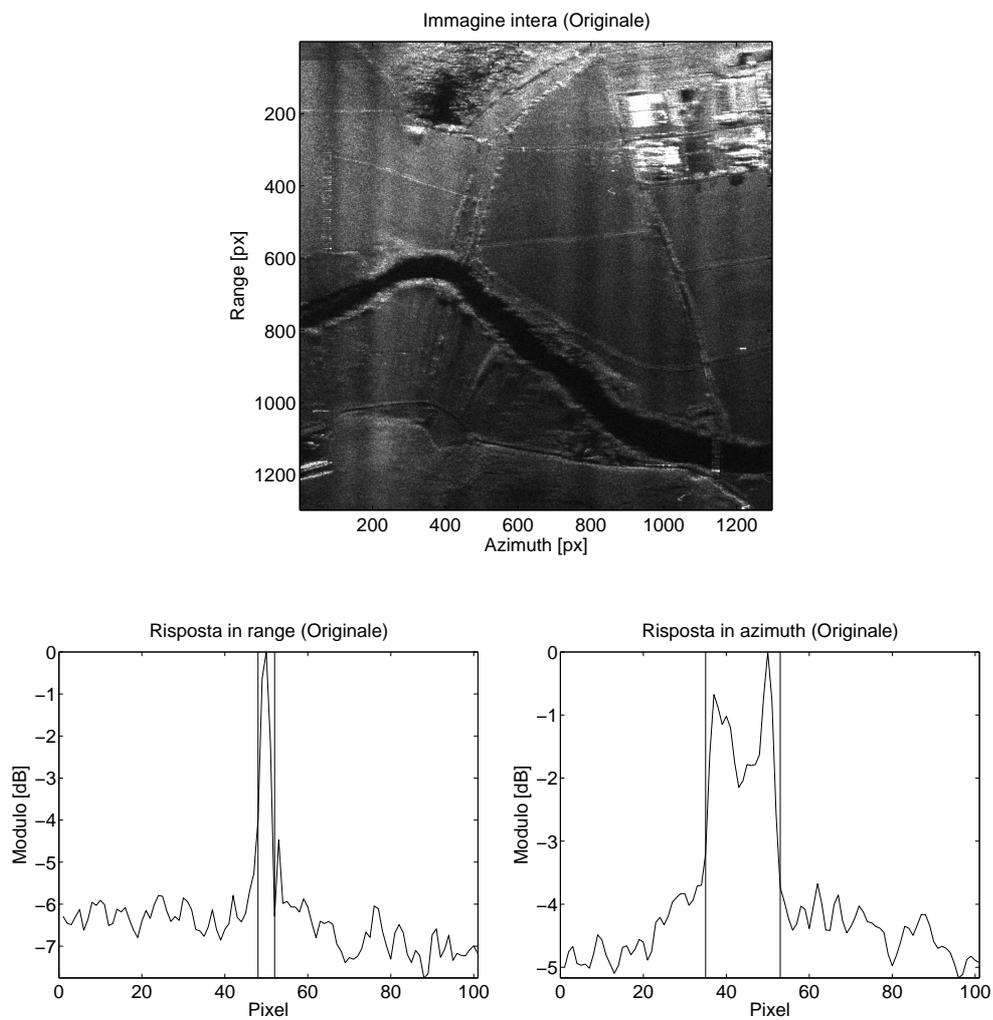


	Originale	Evoluzione
Entropia	-8.34764e+08	-7.79548e+08
Risoluzione in range [px]	4	4
Risoluzione in azimuth [px]	5	5
Tempo di esecuzione [s]	15.86	14.87



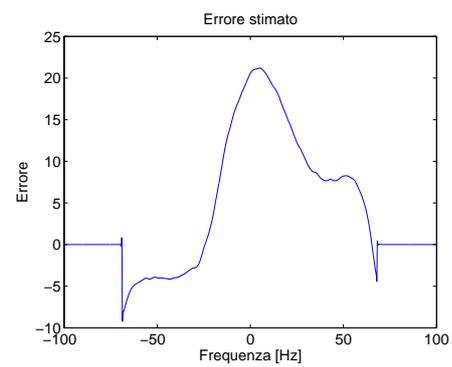
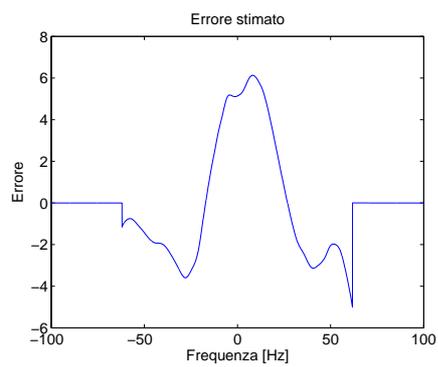
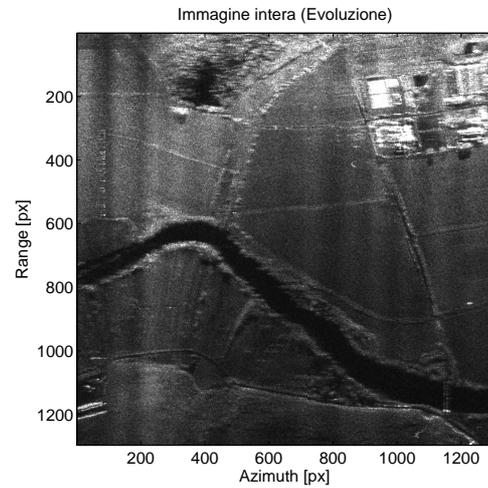
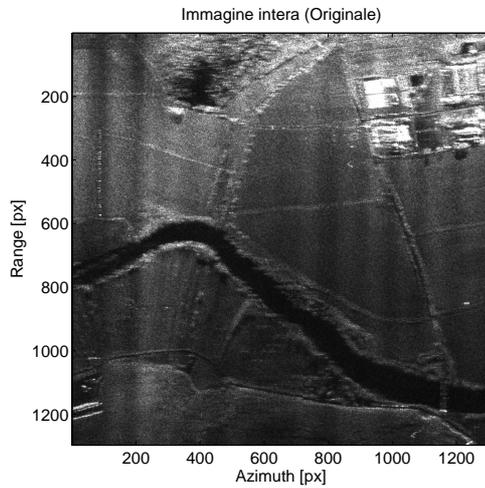
3.3.3 Stabilimento industriale: Azimuth Perturbed

Immagine non elaborata

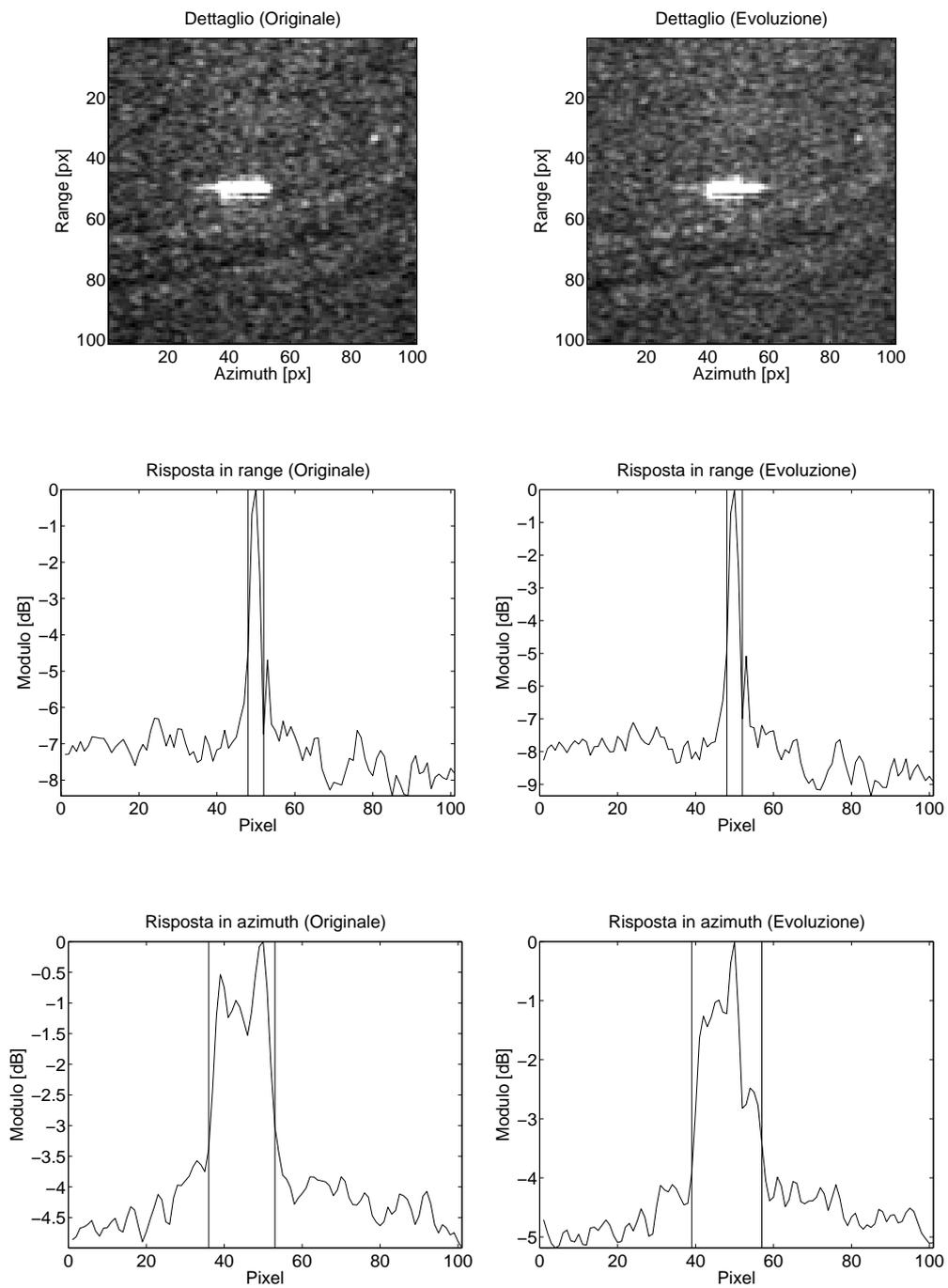


Entropia	-8.28654e+08
Risoluzione in range [px]	4
Risoluzione in azimuth [px]	18

Azimtuh PGA

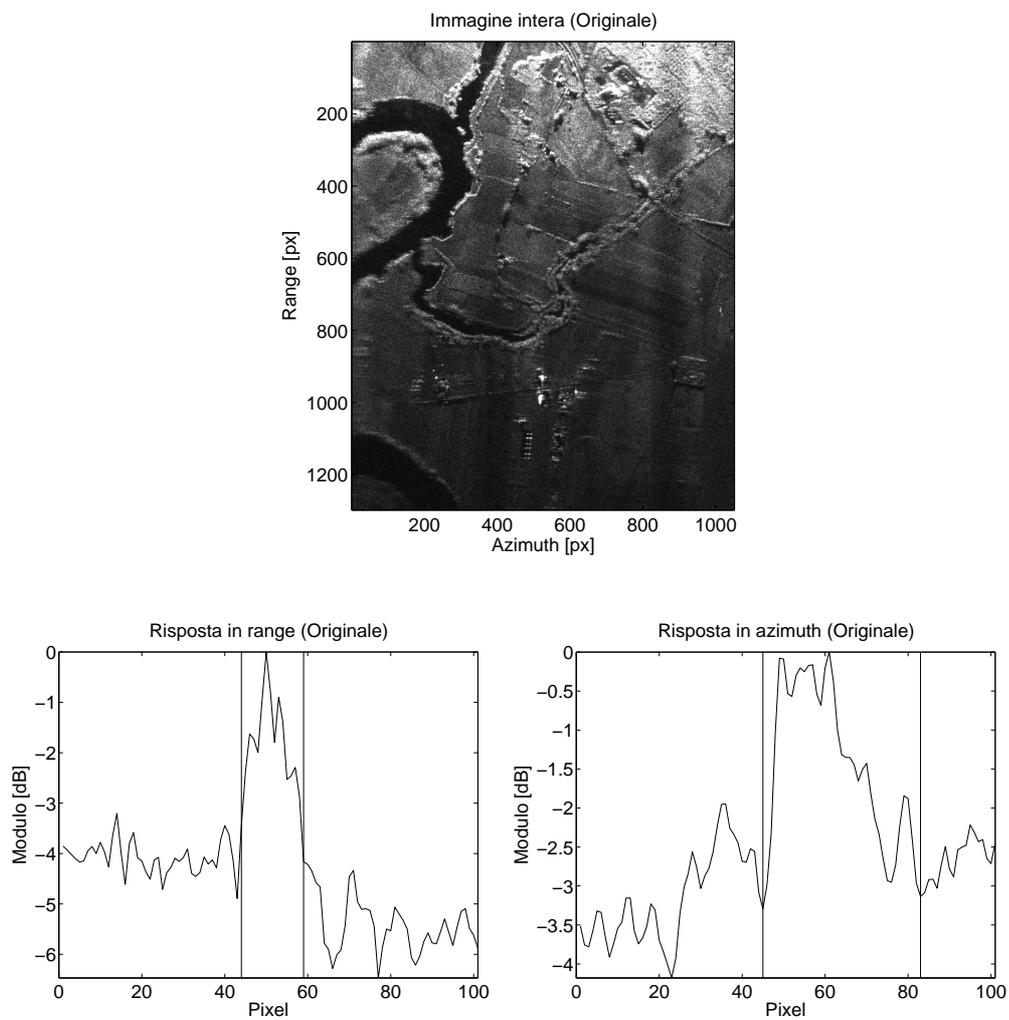


	Originale	Evoluzione
Entropia	-8.13277e+08	-8.06432e+08
Risoluzione in range [px]	4	4
Risoluzione in azimuth [px]	17	18
Tempo di esecuzione [s]	41.16	8.19



3.3.4 Il Volturno: Raw

Immagine non elaborata



Entropia

-5.95271e+08

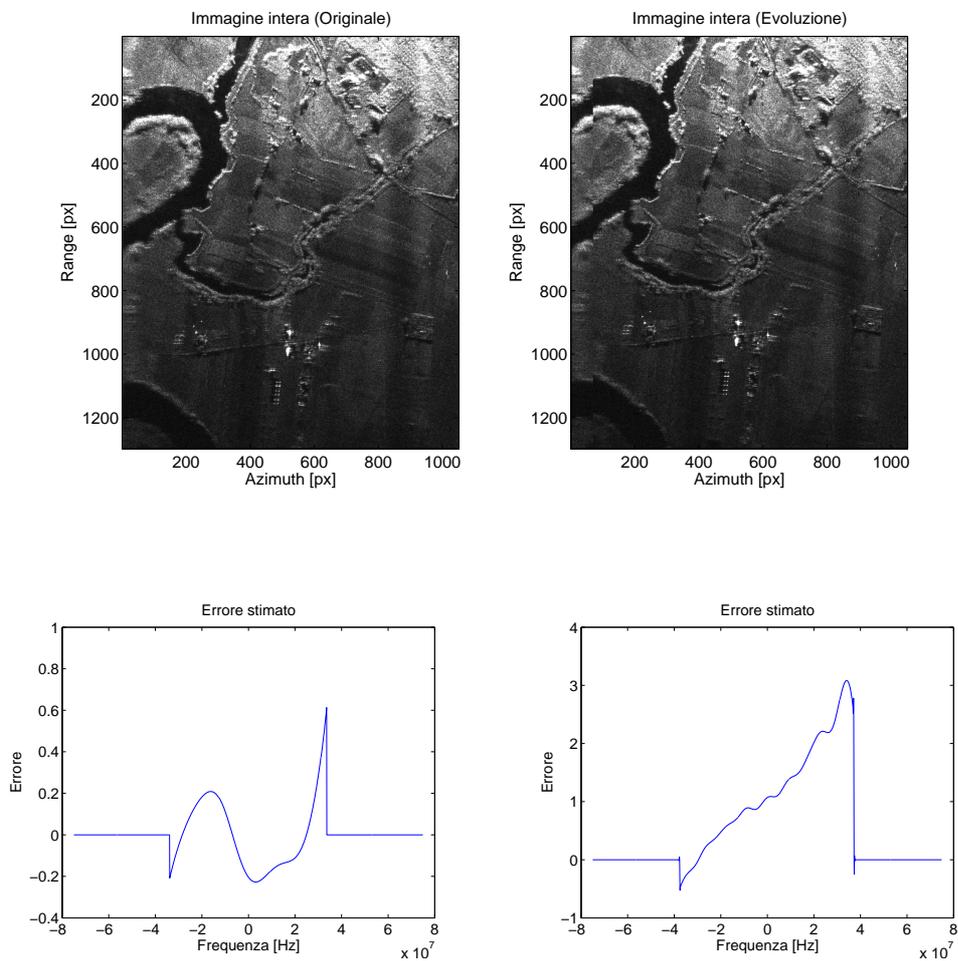
Risoluzione in range [px]

15

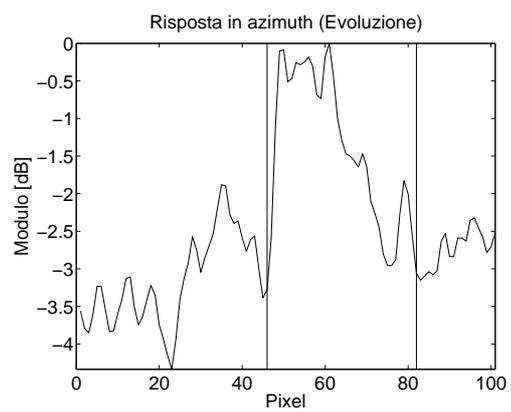
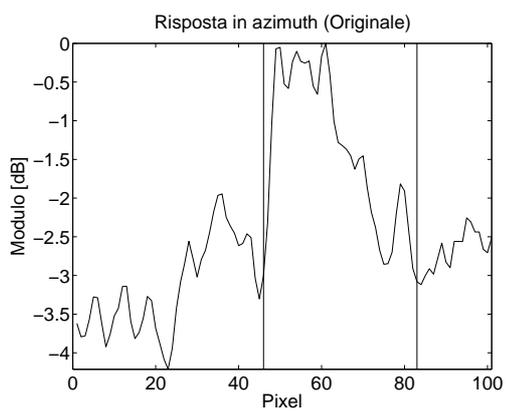
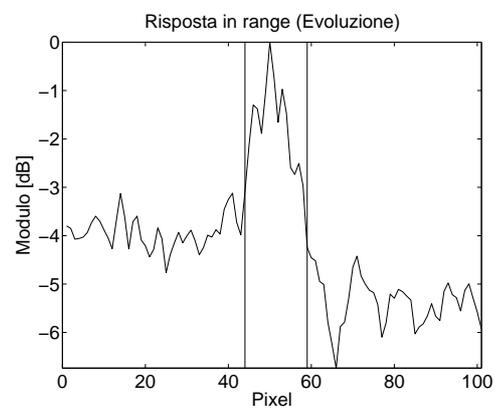
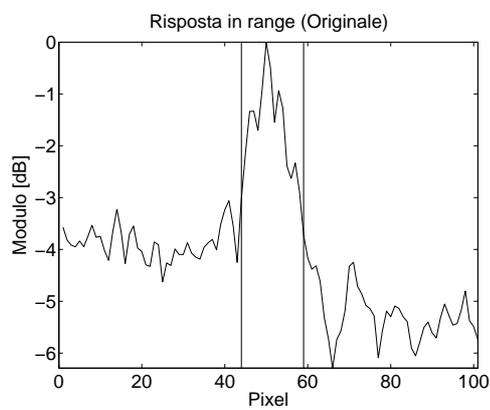
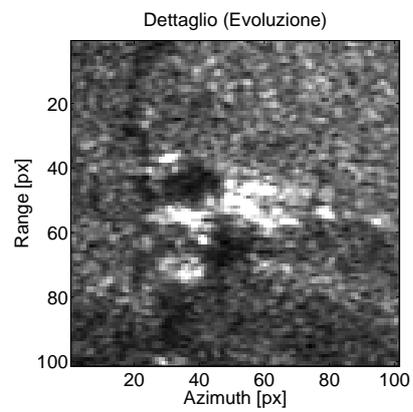
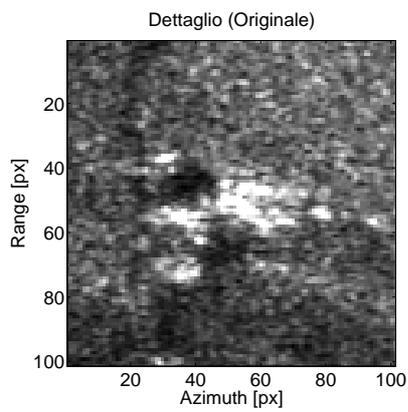
Risoluzione in azimuth [px]

38

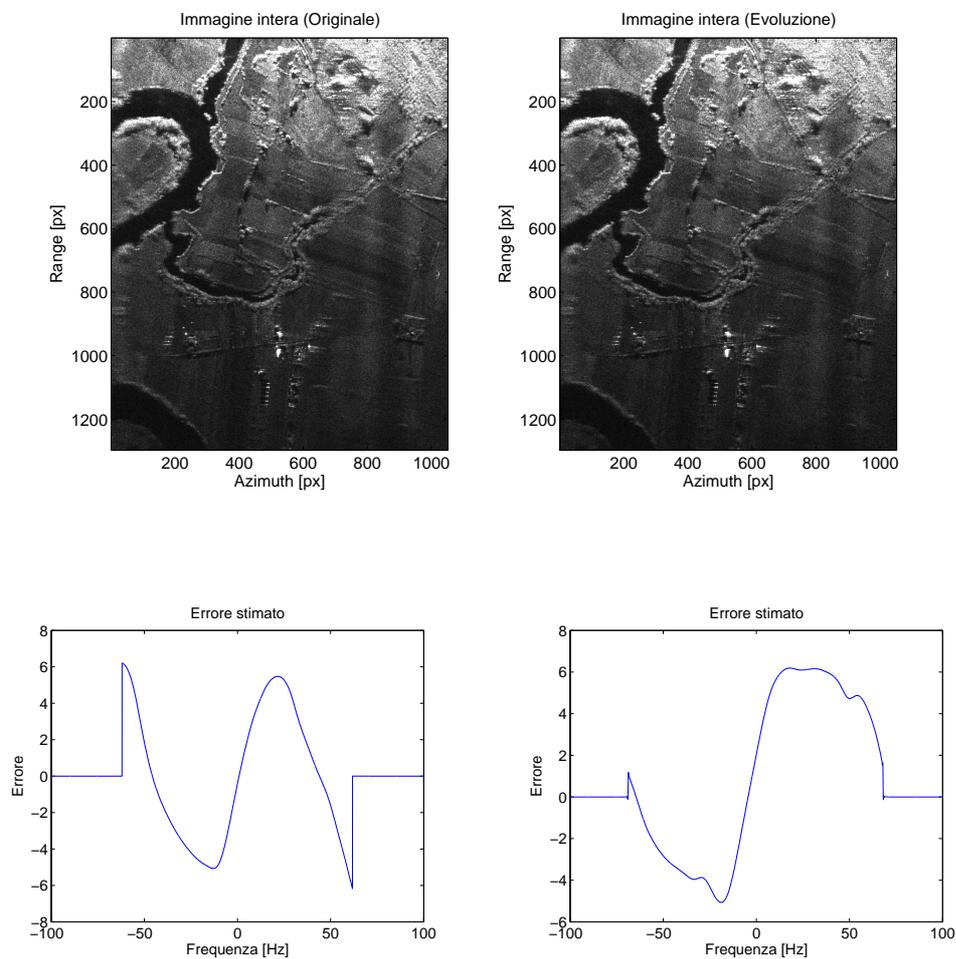
Range PGA



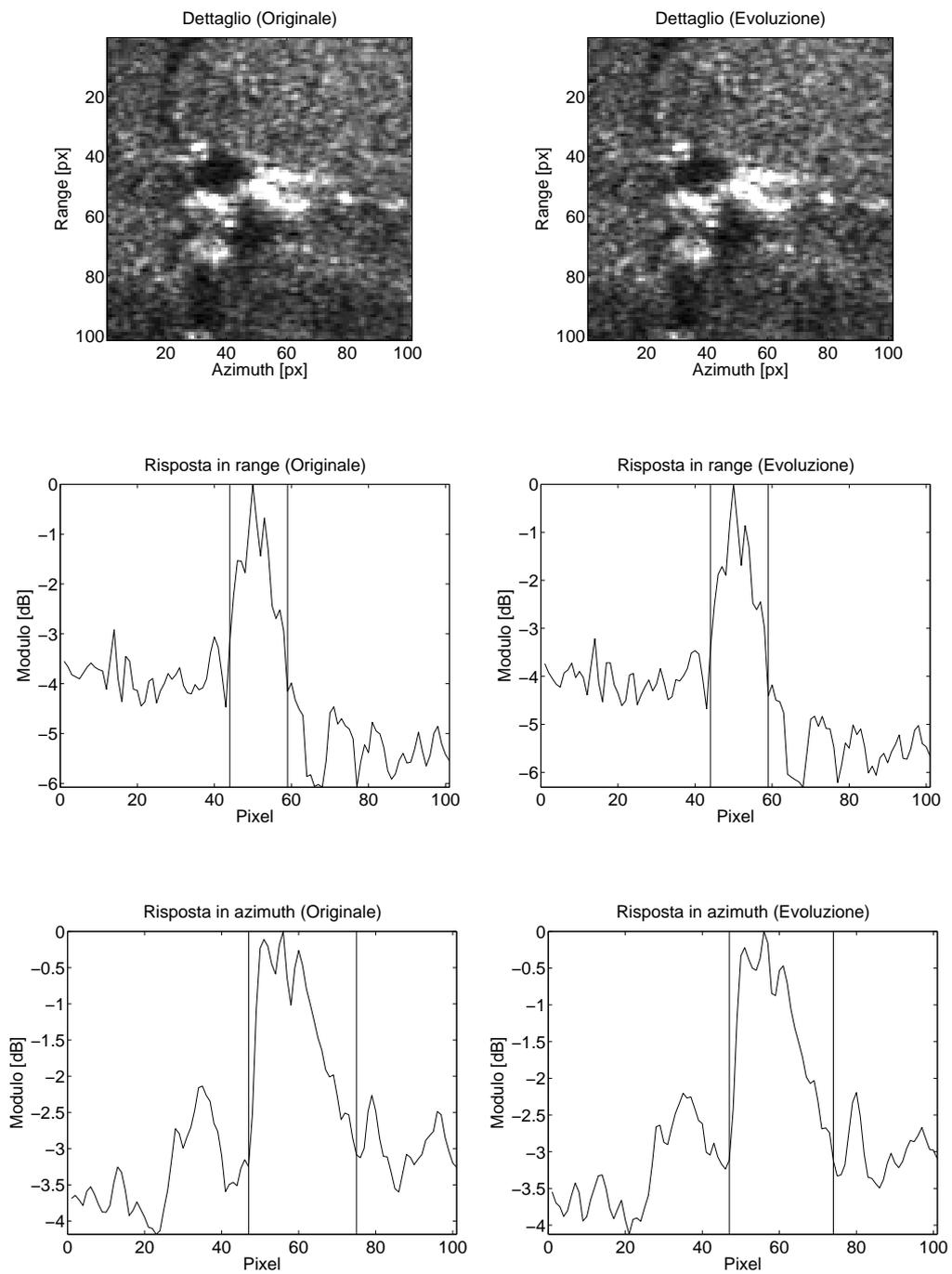
	Originale	Evoluzione
Entropia	-5.16943e+08	-5.71578e+08
Risoluzione in range [px]	15	15
Risoluzione in azimuth [px]	37	36
Tempo di esecuzione [s]	7.00	9.60



Azimuth PGA

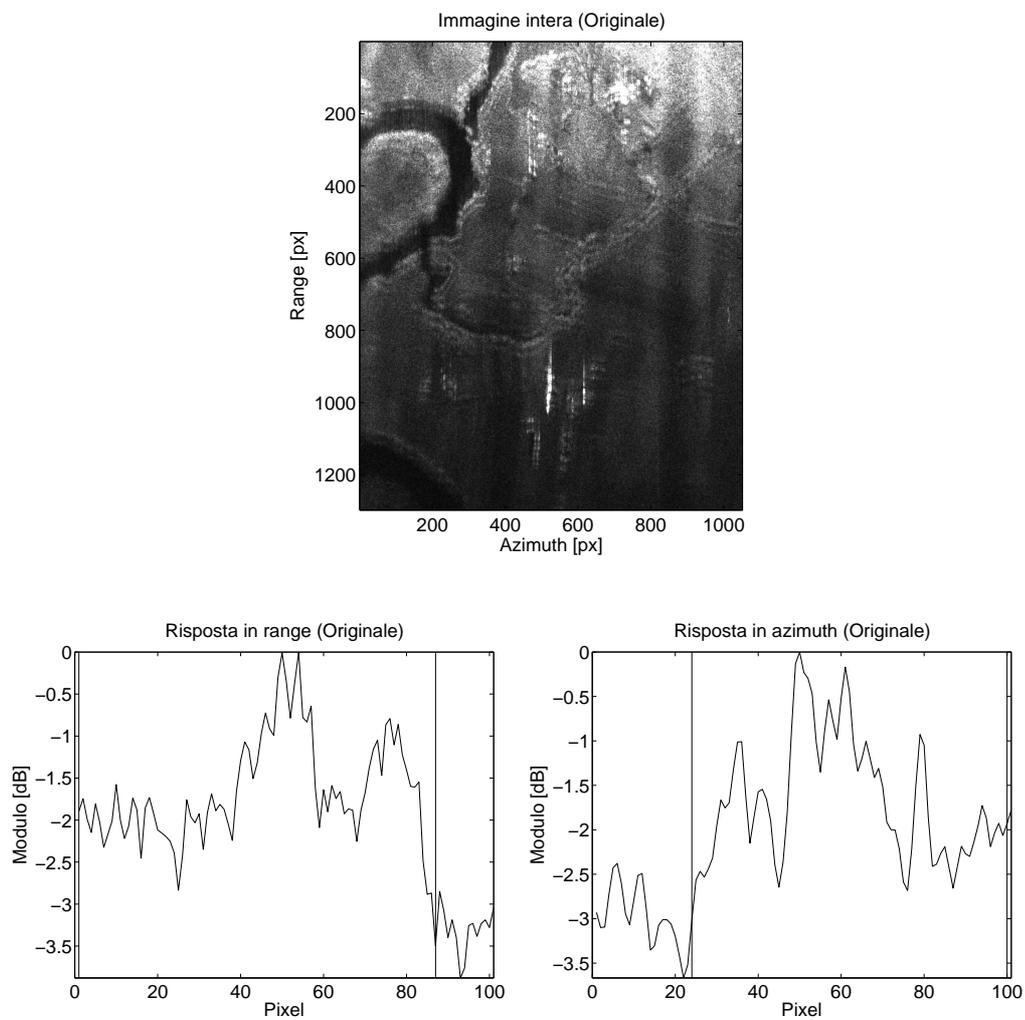


	Originale	Evoluzione
Entropia	-5.92186e+08	-5.92599e+08
Risoluzione in range [px]	15	15
Risoluzione in azimuth [px]	28	27
Tempo di esecuzione [s]	29.28	8.28



3.3.5 Il Volturino: Range Perturbed

Immagine non elaborata



Entropia

-6.2082e+08

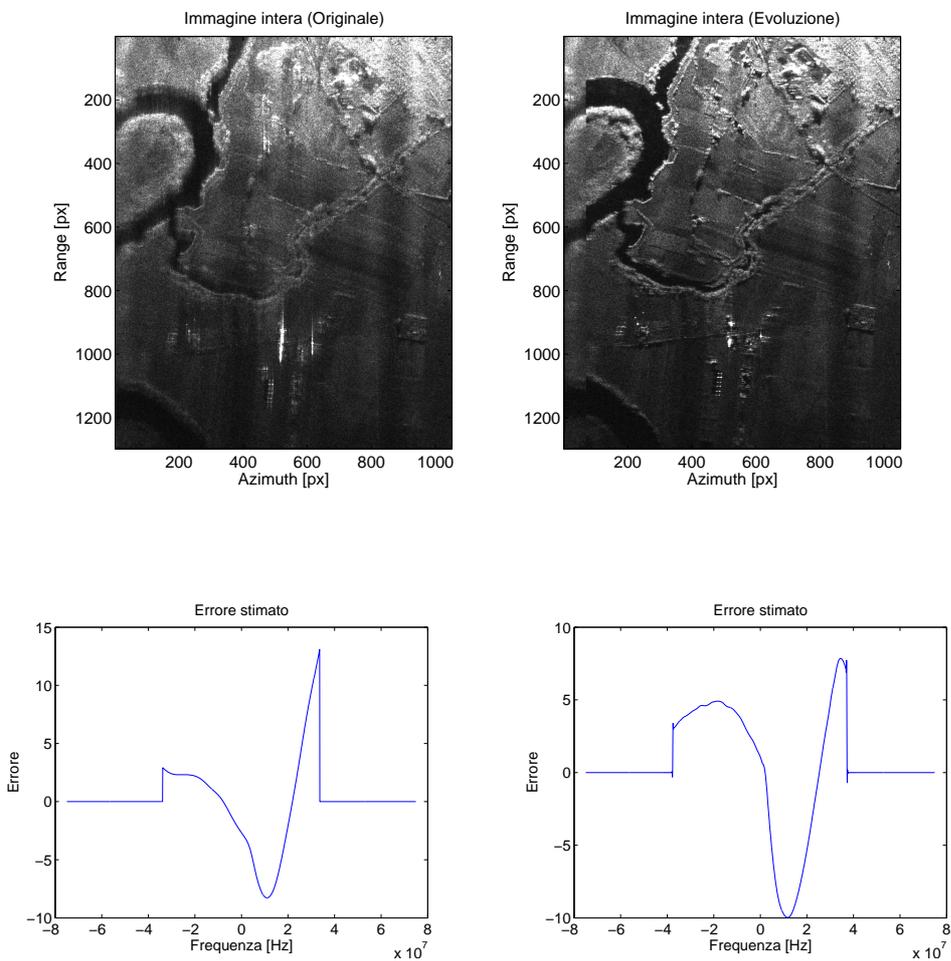
Risoluzione in range [px]

86

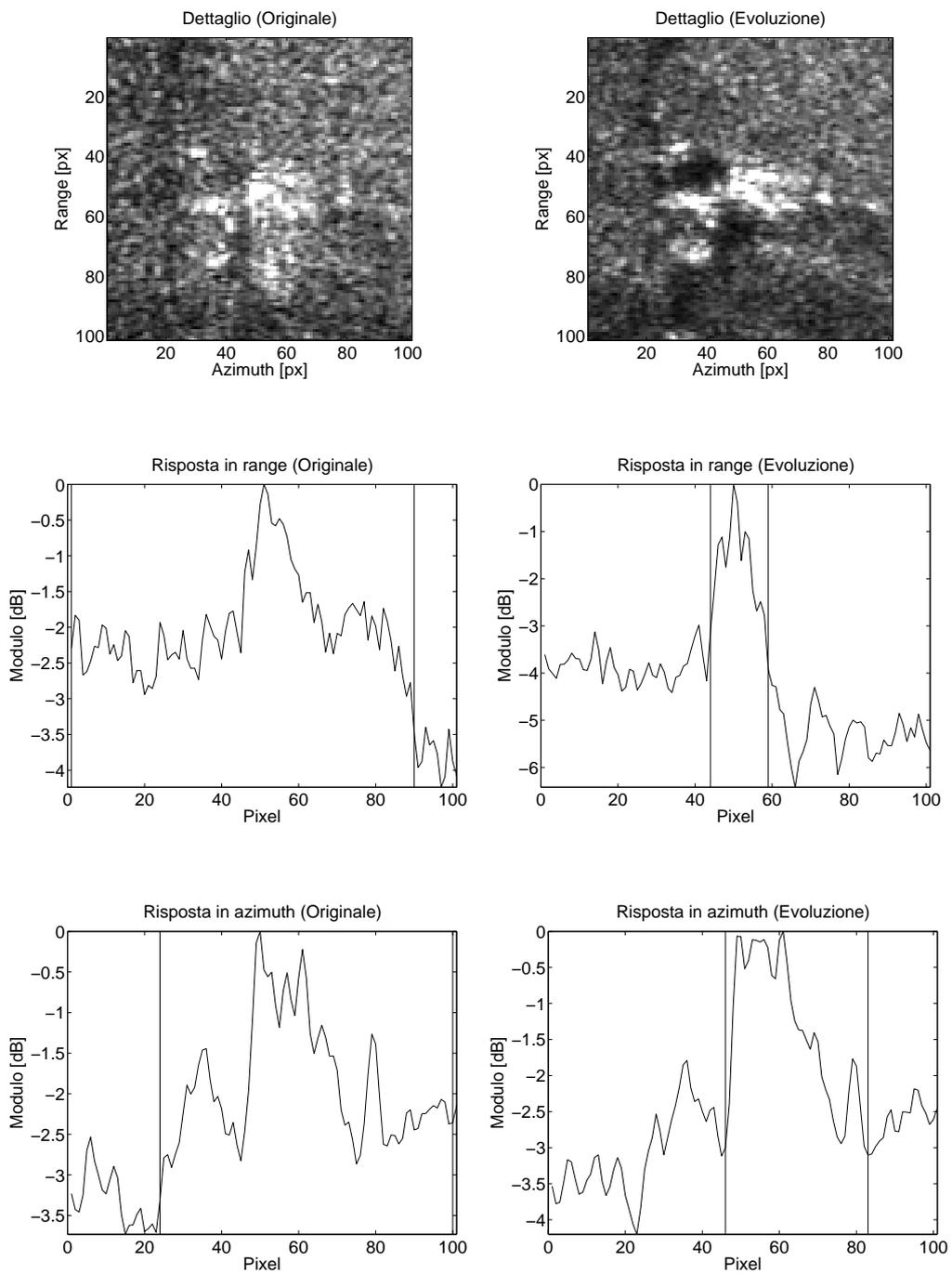
Risoluzione in azimuth [px]

76

Range PGA

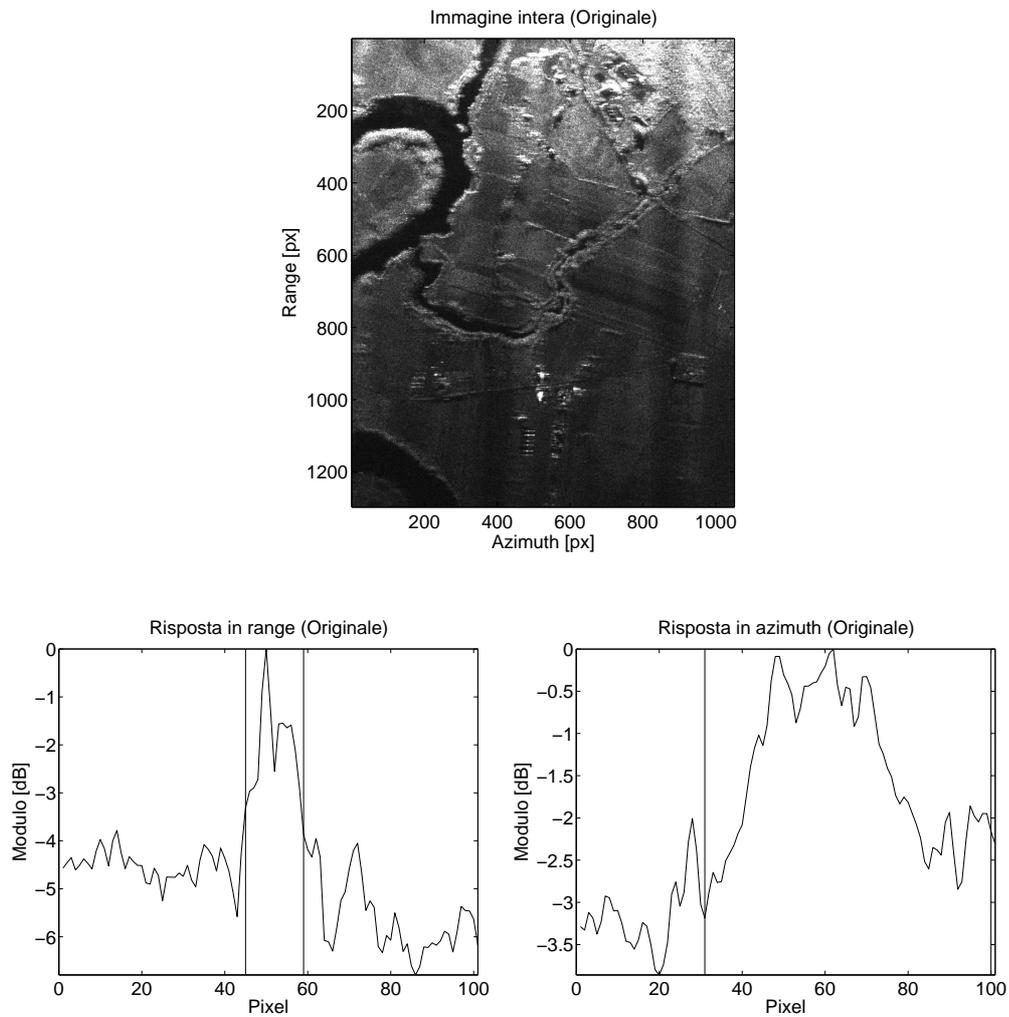


	Originale	Evoluzione
Entropia	-6.12648e+08	-5.73531e+08
Risoluzione in range [px]	89	15
Risoluzione in azimuth [px]	76	37
Tempo di esecuzione [s]	21.85	10.06



3.3.6 Il Volturno: Azimuth Perturbed

Immagine non elaborata



Entropia

-6.01219e+08

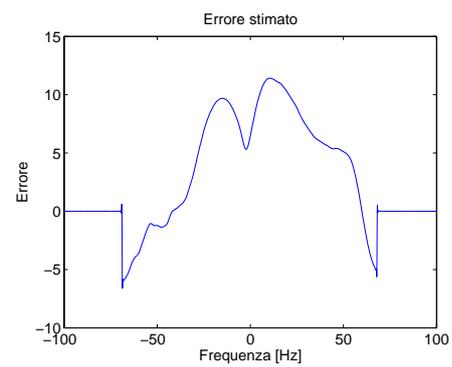
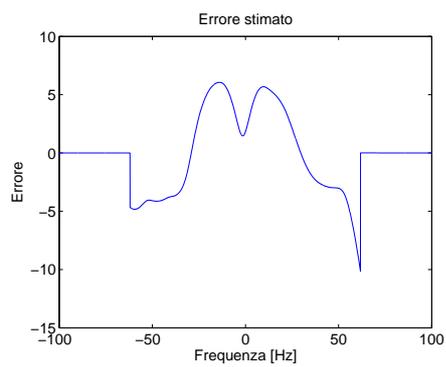
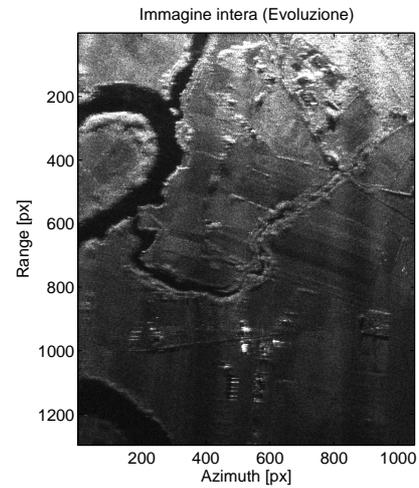
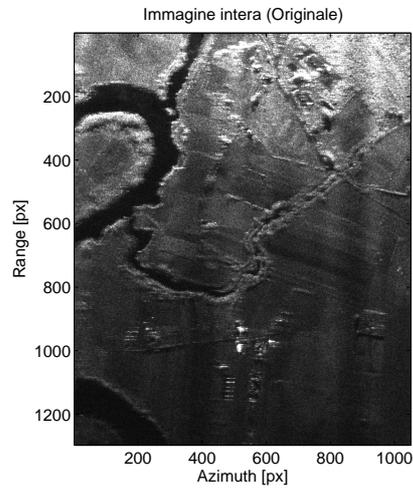
Risoluzione in range [px]

14

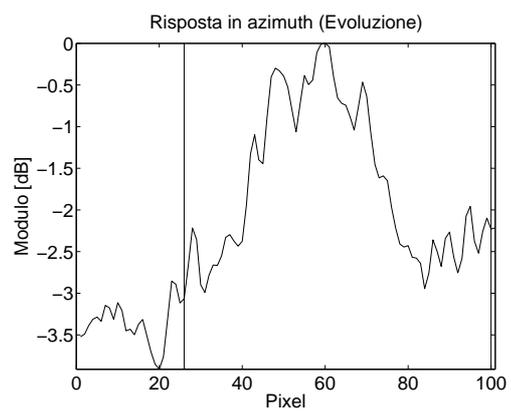
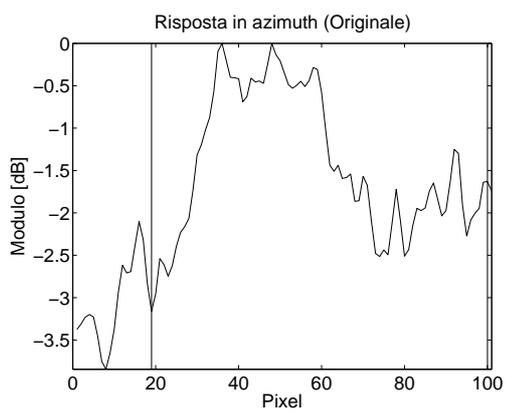
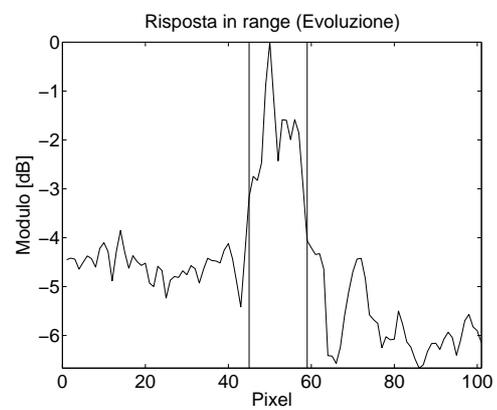
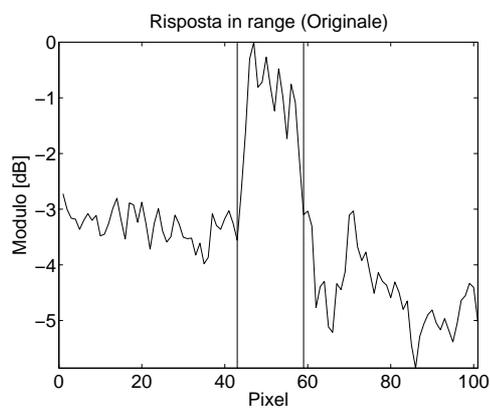
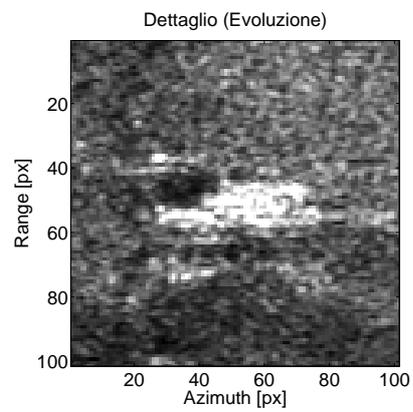
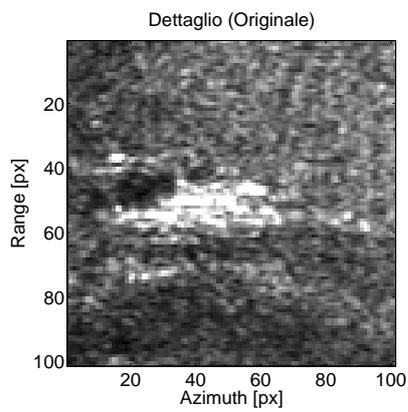
Risoluzione in azimuth [px]

69

Azimtuh PGA



	Originale	Evoluzione
Entropia	-6.02079e+08	-6.01199e+08
Risoluzione in range [px]	16	14
Risoluzione in azimuth [px]	81	74
Tempo di esecuzione [s]	34.47	9.00



3.4 Conclusioni

La nuova implementazione dell'algoritmo PGA ha mostrato prestazioni di assoluto rispetto nel corso dei test presentati nelle pagine precedenti: nella quasi totalità delle situazioni *la stima dell'errore di fase è stata adeguata al conseguimento di un sensibile miglioramento della focalizzazione, ad un costo computazionale estremamente ridotto.*

E' importante notare che i migliori risultati sono stati registrati sulle immagini più problematiche. Nelle situazioni in cui c'è stata prevalenza di risposte clutter-like in associazione ad intensità notevoli dell'errore di fase, la risoluzione ed il tempo di calcolo della nuova implementazione si sono dimostrati apprezzabilmente migliori di quelli della vecchia (vedi 3.3.5). Questo aspetto è particolarmente interessante alla luce delle applicazioni del sistema MINISAR ed, in particolare, del controllo di quelle zone rurali in cui l'assenza di manufatti priva l'immagine di risposte forti ed isolate.

In tutti gli altri casi, sotto l'aspetto del risultato finale, si è registrata notevole concordanza tra i due algoritmi. Il loro funzionamento è uniformemente buono ovunque, tranne che in presenza di condizioni particolari per cui l'uno o l'altro falliscono completamente nel loro compito (vedi 3.2.5 e 3.3.1).

Dal punto di vista della prestazione normalizzata rispetto al peso computazionale, invece, è notevole il miglioramento tra il nuovo ed il vecchio. I dati

mostrano chiaramente che la nuova implementazione ha un comportamento uniforme al variare del contenuto della scena: i tempi di esecuzione, inclusi delle necessarie operazioni di ricampionamento, si mantengono costantemente su valori estremamente contenuti. Per contro, la versione originale del PGA mostra buone prestazioni solo in presenza di scatteratori molto forti, presentando normalmente tempi di esecuzione fino a quattro volte superiori rispetto alla nuova.

In conclusione, è importante sottolineare che il PGA è un algoritmo pensato originariamente per fungere da motion compensation del secondo ordine. Tipicamente non è utilizzato per rimuovere *tutto* l'errore di fase presente in un'immagine: il suo scopo è ripulire quanto resta a valle di una motion compensation classica, insieme a tutto ciò che non è riconducibile al moto della piattaforma. In virtù di questo, l'ampiezza degli errori con cui si ha generalmente a che fare è relativamente limitata.

Nel caso specifico, invece, i set di dati utilizzati per effettuare i test non hanno subito alcun tipo di elaborazione preliminare: la rimozione dell'errore di fase è stata completamente affidata al PGA. Questo ha comportato un utilizzo dell'algoritmo molto più spinto di quanto non si faccia in genere ed, in tal senso, ha consentito di maturare una notevole fiducia nelle sue prestazioni. Tale fiducia spinge a ridimensionare l'uso delle sole tecniche classiche ed invita alla validazione di metodi ibridi in cui si tenga conto delle letture delle piattaforme INS/GPS nella struttura del PGA.

Appendice A

L'algoritmo PGA: le funzioni MATLAB

A.1 L'implementazione originale

```
function [fas_err,var_err]=PGA(risp_imp,fc,B,n_chirp,NFFT,N_cen)

var_err=100;
fas_err=zeros(1,NFFT);
it=1;
f=0:fc/NFFT:fc-1/NFFT;

[nc nr]=size(risp_imp);
risp_imp_defoc_shift=zeros(n_chirp,NFFT);

NB=(NFFT/fc)*min(B);
i_B=[floor(NFFT/2-NB/2):floor(NFFT/2-NB/2)+NB-1];

while min(var_err)>1e-2&it<30

    risp_imp_defoc=risp_imp.'
```

```

sti=zeros(1,NFFT);
Gnum=zeros(1,NFFT);
Gden=zeros(1,NFFT);

for ii=1:n_chirp
    [max_def i_max]=max(abs(risp_imp_defoc(ii,:)));
    help=circshift(risp_imp_defoc(ii,:),[0 (floor(NFFT/2)-i_max)]);
    risp_imp_defoc_shift(ii,1:length(help))=help;
end
risp_mean=mean(abs(risp_imp_defoc_shift).^2,1);

[x_max i_max]=max(risp_mean);
i_win_min=i_max;
i_win_max=i_max;
while (risp_mean(i_win_min)>=x_max*0.01) && (i_win_min>1)
    i_win_min=i_win_min-1;
end

while (risp_mean(i_win_max)>=x_max*0.01) && (i_win_max<nc)
    i_win_max=i_win_max+1;
end

G=zeros(1,NFFT);
for ii=1:n_chirp
    G=circshift(fft(risp_imp_defoc_shift(ii,:)).*
-->    window1(NFFT,i_win_min,i_win_max),NFFT,[0 NFFT/2-N_cen(ii)]);
    DER=gradient(G,fc/NFFT);
    Gnum=Gnum+imag((conj(G)).*DER);
    Gden=Gden+abs(G).^2;
end
sti=Gnum./Gden;
INT=cumtrapz(sti).*fc/NFFT;

INT(i_B)=INT(i_B)-polyval(polyfit(i_B,INT(i_B),1),i_B);

INT(1:i_B(1)-1)=0;
INT(i_B(length(i_B))+1:NFFT)=0;

```

```
fas_err=fas_err+INT;

for ii=1:n_chirp
    INT_matrix(:,ii)=circshift(INT,[0 -NFFT/2+N_cen(ii)]);
end

risp_imp = ifft( fft(risp_imp,NFFT,1) .* exp(j.*INT_matrix),NFFT,1);

var_err(it)=sqrt(var(abs(INT)));
it=it+1;
end
```

A.2 L'evoluzione

```
function [PhaseError RmsVsIteration] = PGA_NEW(Data,fc,B,nPGA_r,nPGA_c,Direction,Centroid)
% La routine PGA effettua la rimozione dell'errore di fase in Data.
% L'algoritmo usato e' una versione leggermente modificata del PGA
% classico di Eichel, Jakowatz e Chiglia.
%
% Gli argomenti passati in ingresso hanno i seguenti significati:
% Data Dati defocalizzati
% fc Frequenza di campionamento
% B Larghezza di banda
% nPGA_r Numero di righe a cui applicare il PGA
% nPGA_c Numero di colonne da considerare in ogni riga
% Direction Direzione a cui va applicato il PGA (1: colonne (default), 2: righe)
% Centroid Posizione degli eventuali centroidi

% DIREZIONE A CUI APPLICARE IL PGA
% L'organizzazione interna dell'algoritmo prevede l'applicazione del
% PGA alle righe della matrice. E' richiesta una trasposizione
% preliminare quando non si specificano Direction o quando lo si
% ponga uguale a 1.
if nargin >= 6
    if Direction == 1
        Data = Data.';
    end
else
    Data = Data.';
end

[nR nC] = size(Data);

% VERIFICA DI CONSISTENZA DEL NUMERO DI COLONNE INDICATO
% Si verifica che non sia stato inserito un numero di colonne maggiore
% del numero di elementi della generica riga dei dati.
if nPGA_c > nC
    nPGA_c = nC;
end
```

```
% COSTANTI
% Si definiscono le costanti in uso nell'algoritmo
MAX_ITERATION = 30;           % Numero massimo di iterazioni
CONVERGENCE_THRESHOLD = 1e-1; % Soglia di convergenza
CUT_THRESHOLD = 0.1;         % Soglia di taglio
NUMBER_OF_ELEMENTS_SQUARED = sqrt(nPGA_c); % Radice quadrata del numero
% di elementi del vettore
% errore di fase stimato

% Selezione dei punti interni alla banda
NB = (nPGA_c/fc)*min(B);
i_B = [floor(nPGA_c/2 - NB/2):floor(nPGA_c/2 - NB/2)+NB-1];

% CENTROIDI
% L'organizzazione interna dell'algoritmo prevede che gli eventuali
% centroidi siano forniti in un vettore colonna. Se Centroid non
% rispetta questa convenzione si effettua una trasposizione. Inoltre,
% si riscalanano linearmente i centroidi per renderli coerenti con le
% lunghezze usate nel PGA.
if nargin == 7
    Centroid = Centroid(:);
else
    Centroid = zeros(nR);
end

Centroid = round(Centroid.*(nPGA_c/nC));

% ESTRAZIONE DEI DATI
% In questa fase si estrae dalla matrice Data un numero di righe e
% colonne sufficiente all'applicazione del PGA. Le righe sono scelte
% in base al valore del modulo dello scatteratore più forte. Per le
% colonne si estrae un intorno opportuno di questi massimi. I centroidi
% sono adattati di conseguenza.

[MaxValue MaxIndex] = max(abs(Data), [], 2);

for i = 1:nR
    Data(i,:) = circshift(Data(i,:), [0 (floor(nC/2)-MaxIndex(i))]);
```

```
end

[MaxValueSort MaxIndexSort] = sort(MaxValue,'descend');

if mod(nPGA_c,2) == 0
    InfCut = floor(nC/2) - nPGA_c/2 + 1;
    SupCut = floor(nC/2) + nPGA_c/2;
else
    InfCut = floor(nC/2) - floor(nPGA_c/2);
    SupCut = floor(nC/2) + floor(nPGA_c/2);
end

Centroid = Centroid(MaxIndexSort(1:nPGA_r));
A = Data(MaxIndexSort(1:nPGA_r),InfCut:SupCut);

clear Data

% INIZIALIZZAZIONE
PhaseError = zeros(1,nPGA_c);
RmsVsIteration = zeros(1, MAX_ITERATION);
RmsTotVsIteration = zeros(1, MAX_ITERATION);
RmsTotRateVsIteration = zeros(1, MAX_ITERATION);

Iteration = 0;
ConvergenceSwitch = 1;

% PGA
while ConvergenceSwitch
    Iteration = Iteration + 1;

    % INIZIALIZZAZIONE DEL CICLO
    clear MaxValue MaxIndex;

    % CIRCULAR SHIFTING
    % Si effettua uno shifting lineare dei dati per portare lo scatteratore
    % più forte di ogni riga al centro.
    [MaxValue MaxIndex] = max(abs(A), [], 2);
```

```
for i = 1:nPGA_r
    A(i,:) = circshift(A(i,:), [0 (floor(nPGA_c/2)-MaxIndex(i))]);
end

% WINDOWING
% Finestratura delle righe intorno al punto di massimo.

% DETERMINAZIONE AUTOMATICA DELLE DIMENSIONI DELLA FINESTRA
% La dimensione della finestra intorno al massimo è calcolata individuando
% per una funzione opportuna i punti di attraversamento della media.
s = mean(abs(A).^2).*nPGA_r;

s_mean = mean(s);
[sMaxValue sMaxIndex] = max(s);

LeftWinOffset = 0;
LeftWinIndex = sMaxIndex;
while (s(LeftWinIndex) >= s_mean) && (LeftWinIndex > 1)
    LeftWinIndex = LeftWinIndex - 1;
    LeftWinOffset = LeftWinOffset + 1;
end

RightWinOffset = 0;
RightWinIndex = sMaxIndex;
while (s(RightWinIndex) >= s_mean) && (RightWinIndex > nPGA_c)
    RightWinIndex = RightWinIndex + 1;
    RightWinOffset = RightWinOffset + 1;
end

Offset = ceil(max(LeftWinOffset, RightWinOffset));
WinWidth(Iteration) = 2*Offset + 1;

% CONTROLLO DI CONVERGENZA DELLA FINESTRA
% Si impone che le dimensioni della finestra vadano a diminuire con le iterazioni.
if Iteration > 1
    if WinWidth(Iteration) > WinWidth(Iteration-1)
        WinWidth(Iteration) = ceil((WinWidth(Iteration-1).*8));
        Offset = ceil((WinWidth(Iteration)-1)/2);
    end
end
```

```
    end
end

% COSTRUZIONE DELLA FINESTRA
% Si costruisce per i dati una finestra gaussiana.
i = 1:nPGA_c;
Variance = abs(Offset^2/(2*log(CUT_THRESHOLD)));
Win = exp(-((i - sMaxIndex).^2)/(2*Variance));

% STIMA DEL GRADIENTE DELL'ERRORE DI FASE
% La stima del gradiente dell'errore di fase è effettuata mediante lo stimatore LUMV
EstNum = zeros(1,nPGA_c);
EstDen = zeros(1,nPGA_c);
for i = 1:nPGA_r
    G = fft(A(i,:).*Win);
    G = circshift(G, [0 floor(nPGA_c/2)-Centroid(i)]);

    % CALCOLO DI NUMERATORE E DENOMINATORE DELLO STIMATORE
    GGrad = gradient(G,fc/nPGA_c);

    EstNum = EstNum + imag(conj(G).*GGrad);
    EstDen = EstDen + abs(G).^2;
end

% CALCOLO DELLO STIMATORE
Est = EstNum./EstDen;

% INTEGRAZIONE ED ELIMINAZIONE DEI TERMINI COSTANTE E LINEARE
EstInt = cumtrapz(Est).*fc/nPGA_c;

C = polyfit(1:nPGA_c,EstInt,1);
EstInt = EstInt - polyval(C, 1:nPGA_c);

% ELIMINAZIONE DEI TERMINI FUORI BANDA
EstInt(1:min(i_B)) = 0;
EstInt(max(i_B):end) = 0;

PhaseError = PhaseError + EstInt;
```

```
% CORREZIONE DEI DATI
for i = 1:nPGA_r
    EstIntShifted = circshift(EstInt, [0 -floor(nPGA_c/2)+Centroid(i)]);
    A(i,:) = ifft(fft(A(i,:)).*exp(j.*EstIntShifted));
end

% RMS DELL'ERRORE DI FASE
RmsVsIteration(Iteration) = norm(EstInt)/NUMBER_OF_ELEMENTS_SQUARED;
RmsTotVsIteration(Iteration) = norm(PhaseError)/NUMBER_OF_ELEMENTS_SQUARED;

% VERIFICA DEL CRITERIO DI CONVERGENZA
% Come criterio di convergenza si stabilisce che la variazione
% percentuale dello RMS dell'errore di fase complessivo sia minore
% di una certa soglia e che il numero di iterazioni non superi un
% certo limite.
if Iteration == 1
    if MAX_ITERATION == 1
        ConvergenceSwitch = 0;
    else
        ConvergenceSwitch = 1;
    end
else
    RmsTotRateVsIteration(Iteration) = abs((RmsTotVsIteration(Iteration)-
--> RmsTotVsIteration(Iteration))/RmsTotVsIteration(Iteration-1));
    ConvergenceSwitch = (RmsTotRateVsIteration(Iteration)
--> > CONVERGENCE_THRESHOLD) & (Iteration <= MAX_ITERATION);
end
end
```

Appendice B

Il pacchetto *minisar_proc*: le chiamate all'algoritmo PGA

B.1 Range

```
if PGA_Range_flg==1
    tic
    fprintf(1,'\n\n Range PGA')
    if PGAType == 'N'
fprintf(1, ': New Algorithm');

    [fas_err, var_err] = PGA_NEW(matrice_dati_rgc,fc,B,n_PGA_r,n_PGA_c,2);

    % Ricampionamento da n_PGA_c a NFFTR
    if ReSampling == 'reg'
        % Regressione ad un polinomio di ottavo grado
        x_old = 0:fc/length(fas_err):fc-1/length(fas_err);
        x_new = 0:fc/NFFTR:fc-1/NFFTR;

        poly_coeff = polyfit(x_old,fas_err,12);
        fas_err = polyval(poly_coeff, x_new);
```

```

        % Cancellazione dei termini fuori banda
        NB = (NFFTR/fc)*min(B);
        i_B = [floor(NFFTR/2 - NB/2):floor(NFFTR/2 - NB/2)+NB-1];

        fas_err(1:min(i_B))=0;
        fas_err(max(i_B):end)=0;
        elseif ReSampling == 'int'
            fas_err = interp1(0:fc/length(fas_err):fc-1/length(fas_err),
-->                fas_err,0:fc/NFFTR:fc-1/NFFTR,'spline');
        end
        elseif PGAType == '0'
            fprintf(1, ' : Old Algorithm');
            [x_max ii_max]=max(abs(matrice_dati_rgc.'));
            [x_max_sort ii_max_sort]=sort(x_max,'descend');

            [fas_err,var_err]=PGA(matrice_dati_rgc(ii_max_sort(1:n_PGA_r)
-->                ,1:nout_rg).',fc,B*0.9,n_PGA_r,NFFTR,zeros(1,n_PGA_r));
        end
        fprintf(1,'\n\t[PGA] '); toc;

        fwrite(FF_aux,length(fas_err),'int');
        fwrite(FF_aux,fas_err,'float');
        fwrite(FF_aux,length(var_err),'int');
        fwrite(FF_aux,var_err,'float');

        fprintf(1,'\n  Range data correction')
        for ii=1:N_read
            matrice_dati(ii,:)=ifft(fft(matrice_dati(ii,:),NFFTR).*
-->                exp(j.*fftshift(fas_err)),NFFTR);
        end

        matrice_dati_rgc = RangeCompression(fft(matrice_dati,[],2)
-->                ,fft(chirp_r,NFFTR),NFFTR,nout_rg);
    end

```

B.2 Azimuth

```

if(PGA_Azimuth_flg==1)

    fprintf(1,'\n  Azimuth compression')
    matrice_dati_azc = AzimuthCompression(matrice_dati_rgc_fft,
-->    fft(chirp_az(1:N_read,:), [], 1), N_read, N_read, N_cen);
    clear matrice_dati_rgc_fft;

    fprintf(1,'\n\n  Azimuth PGA')

    tic
    if PGAType == 'N'
    fprintf(1, ' : New Algorithm');
        [fas_err, var_err]=PGA_NEW(matrice_dati_azc, PRF, Bd, n_PGA_a, n_PGA_c, 1, N_cen);

        % Ricampionamento da n_PGA_c a N_read
        if ReSampling == 'reg'
            % Regressione ad un polinomio di ottavo grado
            x_old = 0:PRF/length(fas_err):PRF-1/length(fas_err);
            x_new = 0:PRF/N_read:PRF-1/N_read;

            poly_coeff = polyfit(x_old, fas_err, 8);
            fas_err = polyval(poly_coeff, x_new);

            % Cancellazione dei termini fuori banda
            NB = (N_read/PRF)*min(Bd);
            i_B = [floor(N_read/2 - NB/2):floor(N_read/2 - NB/2)+NB-1];

            fas_err(1:min(i_B))=0;
            fas_err(max(i_B):end)=0;
        elseif ReSampling == 'int'
            fas_err = interp1(0:PRF/length(fas_err):PRF-1/length(fas_err),
-->            fas_err, 0:PRF/N_read:PRF-1/N_read, 'spline');
        end
    elseif PGAType == 'O'
    fprintf(1, ' : Old Algorithm');
        [x_max ii_max]=max(abs(matrice_dati_azc));

```

```

[x_max_sort ii_max_sort]=sort(x_max,'descend');

[fas_err,var_err]=PGA(matrice_dati_azc(1:N_read,ii_max_sort(1:n_PGA_a)),
--> PRF,Bd(ii_max_sort(1:n_PGA_a))*0.9,n_PGA_a,N_read,N_cen(ii_max_sort(1:n_PGA_a)));
end
fprintf(1,'\n\t[PGA] '); toc;

fwrite(FF_aux,length(fas_err),'int');
fwrite(FF_aux,fas_err,'float');
fwrite(FF_aux,length(var_err),'int');
fwrite(FF_aux,var_err,'float');

NB=floor(N_read/PRF*min(Bd)*0.9);
i_B=[floor(N_read/2-NB/2):floor(N_read/2-NB/2)+NB-1];
pp=polyfit(ni(i_B)/1e6,fas_err(i_B),2);

clear matrice_dati_azc;

fprintf(1,'\n Reference velocity = %f',Velocity)
Velocity_est = sqrt(pi/2*lambda*Rref/abs(pi/2*lambda*Rref/Velocit^2+pp(1)/1e12));
fprintf(1,'\n Estimated velocity = %f',Velocity_est)

fas_err(i_B)=fas_err(i_B)-polyval(pp,ni(i_B)/1e6);
fcen=fcen/Velocit*Velocity_est;

fprintf(1,'\n Azimuth data correction')
for ii=1:NFFT
    matrice_dati(1:N_read,ii)=ifft(fft(matrice_dati(1:N_read,ii),
--> N_read).*exp(j.*circshift(fas_err',[0 -floor(N_read/2)+N_cen(ii)])),N_read);
end
clear fas_matrix;
else
clear matrice_dati_rgc_fft;
Velocity_est = Velocity;
fprintf(1,'\n Reference velocity = %f',Velocity_est)
end
end

```

Ringraziamenti

A conclusione di questo lavoro, voglio ringraziare il centro di ricerca CORISTA per avermi consentito di realizzare la tesi di laurea in un ambiente stimolante e produttivo. Un sentito ringraziamento lo rivolgo, in particolare, al prof. Moccia, all'Ing. Alberti ed all'Ing. Caramiello per la preziosa guida umana, prima ancora che professionale, che mi hanno saputo dare nel corso di questi ultimi mesi.

Sento il dovere, inoltre, di ringraziare tutti quelli che mi hanno aiutato e supportato durante gli anni dell'università.

Un primo speciale pensiero è destinato alla mia famiglia ed, in particolare, ai miei genitori: il loro continuo sostegno e la fiducia che hanno sempre riposto in me sono stati l'ingrediente fondamentale per studiare serenamente, senza mai perdere di vista le priorità fondamentali.

Un enorme “grazie” va a Marcella, una persona eccezionale, che mi ha saputo sempre incoraggiare e spingere avanti nei momenti di maggiore difficoltà, aiutandomi a superare l'innata pigrizia che da sempre mi contraddistingue.

Voglio ringraziare i *panda@castagneto*: la vostra compagnia e la carica di scanzonata allegria che siete in grado di trasmettere sono stati uno degli elementi più belli degli ultimi anni. Un particolare ringraziamento lo rivolgo a Luca e Dania per la vicinanza ed il sorriso che mi hanno regalato durante il complesso periodo della tesi.

Uno speciale ringraziamento lo devo, infine, a tutte le persone con cui ho condiviso il percorso universitario ed, in particolare, a tre loschi figure noti come lo Smilzo, il Virus ed il Foggiano (al secolo Roberto, Andrea ed Ettore). La strada fatta insieme è stata straordinariamente piacevole e produttiva: mi auguro che continui ancora a lungo.

Bibliografia

- [1] Legambiente: *La difesa del suolo in Italia*, Marzo 2006
- [2] P. A. Rosen, S. Hensley, I. R. Joughin, F. K. Li, S. N. Madsen, E. Rodriguez R. M. Goldstein: *Synthetic Aperture Radar Interferometry*, Proceedings of the IEEE, vol. 88, pp. 333-382, 2000.
- [3] G. Alberti, L. Citarella, L. Ciofaniello, R. Fusco, G. Galiero, A. Minoliti, A. Moccia, M. Sacchettino e G. Salzillo: *Current status about the development of an Italian airborne SAR system (MINISAR)*, Proceedings of SPIE, vol. 5236, 2004.
- [4] L. Citarella: *Definizione del software di elaborazione*, MRS-COR/NT/03/05, 2003.
- [5] C. Caramiello: *Elaborazione dati MINISAR*, MRS-COR/NTE/06/050, 2006.
- [6] A. Moreira and Yonghong Huang: *Airborne SAR processing of highly squinted data using a chirp scaling approach with integrated motion compensation*, IEEE Transactions on Geoscience and Remote Sensing, vol. 32, 1994

-
- [7] J. C. Kirk: *Motion Compensation for SAR*, IEEE Transactions on Aerospace and Electronic Systems, May 1975.
- [8] P. H. Eichel, D. C. Chiglia and C. V. Jakowatz: *Speckle processing method for synthetic aperture radar phase correction*, Optics Letters, vol. 14, January 1989.
- [9] D. Laurenza: *Algoritmi di compensazione degli effetti ionosferici su segnali radar in atmosfera marziana*, Tesi di laurea presso la Seconda Università degli Studi di Napoli, 2005.
- [10] D. E. Wahl, P. H. Eichel, D. C. Chiglia and C. V. Jakowatz: *Phase Gradient Autofocus - A robust tool for high resolution SAR phase correction*, IEEE Trans. Aerosp. Electron. Syst., vol. 30, pp. 827-834, July 1994.
- [11] V. C. Koo, T. S. Lim and H. T. Chua: *A comparison of autofocus algorithms for SAR imagery*, Electromagnetics Research Symposium 2005, August 2005.
- [12] D. G. Thompson, J. S. Bates, D. V. Arnold and D. G. Long: *Extending the Phase Gradient Autofocus for low altitude stripmap mode SAR*, Proceedings of the 1999 IEEE Radar Conference, pp. 36-40, April 1999.
- [13] D. E. Wahl, C. V. Jakowatz and P. A. Thompson: *New approach to stripmap SAR autofocus*, Sixth IEEE Digital Signal Processing Workshop, pp. 53-56, October 1994.

-
- [14] W. W. Bonifant: *Interferometric synthetic aperture sonar processing*, Master's thesis at Georgia Institute of Technology, July 1999.
- [15] M. J. Hicks: *On the convergence of the Phase Gradient Autofocus Algorithm for Synthetic Aperture Radar Imaging*, Sandia Report SAND95-2364, January 1996.
- [16] D. E. Wahl and C. V. Jakowatz: *Eigenvector method for maximum-likelihood estimation of phase errors in synthetic aperture radar imagery*, Journal of the Optical Society of America, p. 10, December 1993.
- [17] S. W. Smith: *The scientist and engineer's guide to Digital Signal Processing*, California Technical Publishing, 1999.
- [18] H. L. Chan and T. S. Yeo: *Noniterative Quality Phase Gradient Autofocus (QPGA) Algorithm for Spotlight SAR Imagery*, IEEE transactions on Geoscience and Remote Sensing, Vol. 36, No. 5, pp. 1531-1539, 1998.
- [19] R. L. Morrison, M. N. Do and D. C. Munson: *SAR autofocus by sharpness optimization: a theoretical analysis*, IEEE Transactions on Image Processing, 2005.
- [20] H. A. Zebker, S. N. Madsen, J. Martin, K. B. Wheeler, T. Miller, Y. Lou, G. Alberti, S. Vetrella and A. Cucci: *The TOPSAR interferometric radar topographic mapping instrument*, IEEE Transactions on Geoscience and Remote Sensing, vol. 30, September 1992.

- [21] G. Alberti, S. Vetrella, A. Moccia and S. Ponte: *Analysis and Results of the TOPSAR Experiments in Southern Italy*, EARSeL Advances in Remote Sensing, vol. 3, pp. 2-18, 1994.