

RAPPORTI TECNICI INGV

Tracce planimetriche storiche
e analisi dei movimenti: sviluppo
di un software dedicato



ISTITUTO NAZIONALE DI GEOFISICA E VULCANOLOGIA

518

**Direttore Responsabile**

Daniela VERSACE

Editor in Chief

Milena MORETTI

Editorial Board

Laura ALFONSI

Sofia BARANELLO

Carla BARNABA

Christian BIGNAMI

Simona CARANNANTE

Maria Grazia CIACCIO

Luca COCCHI

Luigi CUCCI

Lorenzo CUGLIARI

Elisabetta D'ANASTASIO

Alessia DI CAPRIO

Roberto DI MARTINO

Domenico DI MAURO

Domenico DORONZO

Filippo GRECO

Alessandro IAROCCI

Mario MATTIA

Daniele MELINI

Anna NARDI

Lucia NARDONE

Marco OLIVIERI

Arianna PESCI

Pierangelo ROMANO

Paola SBARRA

Davide SCAFIDI

Maurizio SOLDANI

Sara STOPPONI

Umberto TAMMARO

David ZULIANI

REGISTRAZIONE AL TRIBUNALE DI ROMA N.173 | 2014, 23 LUGLIO

© INGV Istituto Nazionale di Geofisica e Vulcanologia

Rappresentante legale: Presidente INGV

Sede: Via di Vigna Murata, 605 | Roma

Tutti i diritti riservati

www.ingv.it



ISTITUTO NAZIONALE DI GEOFISICA E VULCANOLOGIA

RAPPORTI TECNICI INGV

Tracce planimetriche storiche e analisi dei
movimenti: sviluppo di un software dedicato

*Historical planimetric traces and movement
analysis: development of a dedicated software*

Arianna Pesci*, Daniele Randazzo

INGV | Istituto Nazionale di Geofisica e Vulcanologia, Sezione di Bologna

*Corresponding Author

Accettato 2 marzo 2026 | Accepted 2 March 2026

Come citare | How to cite Pesci, A., Randazzo, D. (2026). Tracce planimetriche storiche e analisi dei movimenti: sviluppo di un software dedicato. *Rapporti Tecnici INGV*, 518, 22 pp. <https://doi.org/10.13127/rpt/518>

In copertina Rappresentazione concettuale della matrice delle intersezioni della traccia planimetrica storica della Torre Garisenda (1904). Le linee bianche rappresentano le serie temporali degli spostamenti nelle direzioni Est e Nord utilizzate per la ricostruzione della traccia | Cover
Conceptual representation of the intersection matrix of the historical planimetric trace of the Garisenda Tower (1904). The white lines represent the East and North displacement time series used to reconstruct the trace

518

INDICE

Riassunto	7
<i>Abstract</i>	7
Introduzione	7
1. Descrizione del software	9
1.1 Selezione della cartella di lavoro e dei file	9
1.2 Controllo dei dati	10
1.3 Visualizzazione e sovracampionamento eventuale	10
1.4 Filtraggio della traiettoria	10
1.5 Identificazione delle intersezioni	10
1.6 Visualizzazione dei risultati	11
2. Il significato degli output	11
3. Il caso di studio	12
3.1 Analisi dati	13
3.2 Risultati e confronto	15
4. Conclusioni	16
Bibliografia	17
Appendice A	19
Appendice B	29

Riassunto

In questo lavoro viene presentato un software sviluppato in MATLAB per l'analisi di tracce planimetriche e l'identificazione di punti di intersezione lungo traiettorie bidimensionali. Il programma, modulare e flessibile, consente di visualizzare i dati grezzi, applicare elaborazioni opzionali come sovra-campionamento e filtraggio, e generare rappresentazioni quantitative e grafiche dei movimenti. L'applicazione principale riguarda i dati storici del 1904 relativi alla torre Garisenda di Bologna, digitalizzati da registrazioni cartacee, permettendo un confronto con le osservazioni originali. L'analisi evidenzia sia cicli brevi, come quelli giornalieri o settimanali, sia movimenti più lunghi, dimostrando la capacità del software di valorizzare dati storici e restituirli in forma moderna e interpretabile. Questo lavoro fa parte di una serie di studi sulle serie temporali, finalizzati alla costruzione di uno strumento completo per l'analisi fine della dinamica di strutture storiche, applicabile anche all'elaborazione di serie planimetriche analoghe in altri contesti.

Abstract

This work presents a MATLAB software developed for the analysis of planimetric traces and the identification of intersection points along two-dimensional trajectories. The program, modular and flexible, allows visualization of raw data, optional processing such as oversampling and filtering, and the generation of quantitative and graphical representations of movements. The main application concerns historical 1904 data of the Garisenda Tower in Bologna, digitized from paper records, allowing comparison with the original observations. The analysis highlights both short cycles, such as daily or weekly, and longer-term movements, demonstrating the software's capability to enhance historical data and present them in a modern, interpretable form. This work is part of a series of studies on temporal series, aimed at developing a comprehensive tool for fine analysis of the dynamics of historical structures, also applicable to the processing of analogous planimetric series in other contexts.

Keywords Tracce planimetriche; Analisi delle intersezioni; Dati storici | Planimetric trace; Intersection analysis; Historical data

Introduzione

Il movimento nello spazio, anche in modo apparentemente casuale, lascia una traccia, come un'impronta del percorso. Analizzare le tracce planimetriche significa studiare come si muovono gli oggetti o le persone su una superficie bidimensionale, per comprenderne le dinamiche e identificare eventuali punti di incrocio o sovrapposizione. Questi punti, chiamati intersezioni, possono fornire informazioni preziose: ad esempio sul comportamento di più soggetti che condividono uno stesso percorso, sulla sincronizzazione di eventi o semplicemente per ricostruire mappe di movimento complesse.

L'analisi delle intersezioni nelle traiettorie è una metodologia consolidata in diversi ambiti: nello studio dei movimenti di persone o veicoli, le intersezioni aiutano a comprendere comportamenti e sincronizzazioni di eventi; nella ricerca cartografica e sulle mappe storiche, l'analisi planimetrica consente di individuare distorsioni geometriche e ricostruire percorsi accurati. Esistono studi complessi e performanti dedicati a questi temi: alcuni analizzano le

traiettorie in contesti di traffico civile per stimare tempi di incrocio e ottimizzare la viabilità [Wang, 2024; Borresen et al., 2017]; altri mirano a identificare gli algoritmi più efficienti per il rilevamento delle intersezioni [Bentley & Ottmann, 1979].

Nel nostro lavoro, invece, l'approccio è volutamente più semplice e adattato a serie storiche, come quella della Torre Garisenda, con l'obiettivo di stimare tempi e posizioni degli incroci lungo la traiettoria. Questo punto ha rappresentato una sfida fin dall'inizio, principalmente perché nelle analisi moderne di edifici alti, come torri o campanili, gli studi si concentrano prevalentemente su frequenze di oscillazione e modi vibratorii, utilizzando strumenti di monitoraggio continuo e approcci modali o basati su analisi di Fourier [Gentile et al., 2019; Elyamani et al., 2017]. Nel lavoro storico del Prof. Cavani, invece, lo studio delle tracce planimetriche e delle intersezioni aveva lo scopo di costruire un diagramma a specchio o una matrice, utile a comprendere i movimenti periodici e gli intervalli delle oscillazioni relative, analizzando direttamente i grafici degli spostamenti temporali [Cavani, 1910]. Sebbene la maggior parte della letteratura qui indicata sulla utilità delle tracce planimetriche e intersezioni spaziali riguardi contesti urbani e infrastrutturali, esistono comunque numerosi studi dedicati al monitoraggio di edifici storici, in particolare torri e strutture vincolate, mediante serie temporali di posizioni planimetriche e verticali [Barazzetti & Roncoroni, 2011; Ruccolo et al., 2021; Di Giacomo, 2022; Malcangio, 2022]. Questi studi evidenziano come le serie temporali planimetriche possano essere utilizzate per valutazioni precise dei movimenti statici e dinamici, fornendo un contesto metodologico solido per l'analisi presentata nel presente lavoro.

Strumenti come MATLAB [MathWorks, 2024] offrono funzionalità per il calcolo geometrico e l'analisi di traiettorie, ma le funzioni standard non consentono di integrare direttamente informazioni temporali. Per questo, il software sviluppato utilizza un algoritmo interno che combina dati spaziali e temporali (Tempo1, Tempo2, Delta-tempo), consentendo di gestire traiettorie con campionamenti irregolari o frequenze temporali differenti. L'elaborazione prevede la selezione dei segmenti candidati vicini, la verifica delle intersezioni mediante calcolo determinante e la registrazione di coordinate, tempi e indici dei segmenti. Questo approccio garantisce piena coerenza tra dati storici e analisi, mantenendo elevata precisione anche in presenza di segmenti quasi coincidenti o traiettorie complesse. In particolare, la riduzione della dimensione degli intervalli di ricerca lungo le traiettorie limita il numero di segmenti candidati da confrontare per l'individuazione delle intersezioni, ottimizzando i tempi di calcolo senza compromettere l'accuratezza dei risultati. Pur operando su dati bidimensionali, il metodo si dimostra robusto e adattabile a serie temporali storiche con campionamenti irregolari o frequenze temporali differenti.

Nel caso specifico descritto in questo rapporto, i dati provengono da uno studio effettuato nel 1904 che documenta gli spostamenti della Torre Garisenda di Bologna [Cavani, 1910]. Le misurazioni, effettuate quotidianamente per tutto l'anno, registravano le componenti est e nord della posizione della torre. I dati presentati come esempio applicativo sono stati digitalizzati a partire da una tavola tecnica dell'epoca e sono rappresentati in scala temporale (anni) e spaziale (millimetri). Per facilitare questo tipo di analisi, è stato sviluppato un software dedicato in ambiente MATLAB, che trasforma i dati grezzi di movimento in informazioni facilmente interpretabili. Il programma opera secondo le seguenti fasi:

- Preparazione dei dati - Selezione della cartella di lavoro e dei file contenenti le componenti del movimento lungo gli assi EST e NORD. Dopo un controllo di coerenza, il software crea un file unico con le informazioni essenziali di tempo e posizione.
- Visualizzazione della traiettoria - La traccia planimetrica può essere visualizzata in 2D, con la possibilità di evidenziare l'evoluzione temporale tramite una scala di colori. È anche possibile sovracampionare o filtrare i dati per ridurre il rumore e rendere più chiari i dettagli della traiettoria.
- Identificazione delle intersezioni - Il cuore dell'analisi consiste nel riconoscere i punti in

cui i percorsi si incrociano. Il software individua i segmenti vicini secondo una soglia di tolleranza e conferma le vere intersezioni, calcolandone coordinate e tempi associati.

- Output e visualizzazione dei risultati – Tutti i dati sulle intersezioni vengono salvati in un file dedicato e possono essere rappresentati graficamente. È possibile generare *scatter plot* e istogrammi per comprendere meglio la distribuzione dei tempi e la dinamica dei movimenti.

Lo scopo del software è quello di trasformare un insieme di coordinate in una rappresentazione chiara dei movimenti e dei punti di incontro, fornendo sia un quadro quantitativo sia una visualizzazione intuitiva delle traiettorie. Questo lavoro fa parte di una serie di studi sulle serie temporali, finalizzati alla costruzione di uno strumento completo per l'analisi fine della cinematica/dinamica di strutture storiche ma, nella sua semplicità, applicabile anche ad altre serie di movimenti planimetrici.

1. Descrizione del software

Il software sviluppato consente di studiare il movimento di un oggetto in uno spazio bidimensionale e di individuare i punti in cui il suo percorso si incrocia, le cosiddette intersezioni. Il flusso di lavoro è intuitivo: i dati vengono prima raccolti, quindi la traiettoria viene preparata e visualizzata nella sua forma originale, con la possibilità di applicare eventuali modifiche; successivamente si identificano le intersezioni e infine i risultati vengono presentati in modo chiaro e immediato.

Tra le opzioni introdotte nel software vi sono il sovracampionamento e l'applicazione di filtri, pensati per garantire maggiore flessibilità nell'analisi. Il sovracampionamento, ad esempio, permette di migliorare la continuità di una traiettoria quando la risoluzione temporale dei dati originali è bassa, ottenendo una rappresentazione più dettagliata. Allo stesso tempo, è fondamentale distinguere i dati originali da quelli modificati, in modo da comprendere l'impatto delle elaborazioni sull'interpretazione finale. L'applicazione dei filtri, utile per ridurre il rumore, deve essere valutata criticamente: la possibilità di scegliere il tipo di filtro e la larghezza della finestra consente all'utente di bilanciare precisione, regolarizzazione e fedeltà ai dati originali. Grazie a queste elaborazioni, è possibile identificare segmenti di movimento più rapidi o più lenti e analizzare la distribuzione temporale degli incroci. I grafici combinati e le heatmap trasformano infine lunghe serie di numeri in rappresentazioni intuitive, facilitando la comprensione complessiva del movimento.

Si segnala che, insieme al codice principale per l'analisi delle tracce, viene fornito un piccolo software accessorio, denominato PreDati, allegato a questo rapporto. Si tratta di un codice estremamente semplice che consente di caricare serie temporali provenienti da file Excel o CSV e di generare automaticamente il formato di input richiesto dal software principale (est.txt e nord.txt), includendo la colonna del tempo e garantendo la coerenza dei campionamenti.

1.1 Selezione della cartella di lavoro e dei file

Per iniziare, l'utente deve indicare dove si trovano i dati.

```
cartella = uigetdir(pwd, 'Scegli la cartella di lavoro');
```

Questa funzione apre una finestra in cui scegliere la cartella. Poi si selezionano i file contenenti le posizioni EST e NORD:

```
[fileEst, pathEst] = uigetfile('*.txt', 'Seleziona il file EST',  
cartella);
```

Il percorso completo al file viene costruito con *fullfile*, così il programma sa esattamente dove leggere i dati. Questo passaggio garantisce che il software lavori sul dataset corretto e riduce la possibilità di errori.

1.2 Controllo dei dati

Prima di elaborare i dati, il programma verifica che i tempi dei due file coincidano, in modo che ogni misura EST corrisponda a quella NORD:

```
if length(tEst) ~= length(tNord) || any(tEst ~= tNord)
    warning('I tempi tra EST e NORD non coincidono perfettamente.');
```

end

Viene anche controllato il passo temporale, cioè la distanza tra due misure consecutive, per assicurarsi che sia uniforme:

```
dtEst = diff(tEst);
dtNord = diff(tNord);
```

Questi controlli evitano problemi di allineamento che potrebbero falsare l'analisi delle intersezioni.

1.3 Visualizzazione e sovracampionamento eventuale

Il software mostra subito la traiettoria in 2D, colorando i punti in funzione del tempo, così da capire a colpo d'occhio come si muove l'oggetto:

```
scatter(estData, nordData, 20, tEst, 'filled');
```

Se l'utente desidera una rappresentazione più dettagliata, è possibile aumentare la densità dei punti tramite sovracampionamento e nuovi punti vengono calcolati per interpolazione lineare:

```
estNew = interp1(tEst, estData, tNew, 'linear');
nordNew = interp1(tNord, nordData, tNew, 'linear');
```

Questo permette di avere una traiettoria più "liscia" e precisa, utile per analisi più accurate.

1.4 Filtraggio della traiettoria

Per ridurre il rumore nei dati, si possono applicare diversi filtri, come quello gaussiano:

```
filtro = exp(-0.5*(x/sigma).^2)';
estFiltrata = conv(estData, filtro, 'same');
```

Il filtro gaussiano "smussa" le fluttuazioni improvvise senza alterare la tendenza principale del movimento. Viene sempre mostrata una comparazione tra traiettoria originale e filtrata, così l'utente può valutare l'effetto del filtro.

1.5 Identificazione delle intersezioni

Il cuore dell'analisi consiste nell'individuare dove i segmenti della traiettoria si incontrano. Per prima cosa si selezionano coppie di punti vicini tra loro:

```
dist = sqrt( (x(i)-x(j))^2 + (y(i)-y(j))^2 );
```

Poi si verifica matematicamente se i segmenti effettivamente si intersecano calcolando i parametri:

```
t = ((x1-x3)*(y3-y4) - (y1-y3)*(x3-x4)) / den;
```

$u = -((x1-x2)*(y1-y3) - (y1-y2)*(x1-x3)) / den;$

Se t e u sono tra 0 e 1, l'intersezione è confermata. Le coordinate, i tempi e gli indici dei segmenti vengono salvati in un file `_intersezioni.txt` per successive analisi.

1.6 Visualizzazione dei risultati

Infine, il software offre diverse modalità di visualizzazione. I punti di intersezione possono essere rappresentati rispetto a EST, NORD e tempo:

```
scatter(Tempo1, DT, 40, 'b', 'filled');
histogram(DT, 20);
imagesc(edges{1}, edges{2}, counts');
```

2. Il significato degli output

Il software richiede come dati di partenza un file contenente le misure temporali e le posizioni bidimensionali lungo gli assi EST e NORD (o in generale X e Y). Ogni file deve avere almeno due colonne: la prima con i tempi (Tempo), la seconda con i valori della componente corrispondente (EST o NORD).

La colonna del tempo rappresenta l'istante in cui è stata effettuata la misura ed è fondamentale che i tempi dei due file EST e NORD coincidano o siano allineabili, perché ogni punto della traiettoria 2D viene ricostruito accoppiando le due componenti nello stesso istante.

La colonna EST e la colonna NORD contengono le coordinate spaziali della traiettoria lungo i rispettivi assi. Queste misure costituiscono la base su cui il software costruisce la traiettoria 2D e, successivamente, analizza gli incroci tra segmenti.

Il software verifica automaticamente la coerenza dei tempi e dei passi temporali tra i due file e, se necessario, emette avvisi. Una volta controllati, i dati vengono uniti in un file unico (`traccia.txt`) con tre colonne: Tempo, EST e NORD. Questo formato standard semplifica tutte le operazioni successive, dalla visualizzazione al filtraggio e all'identificazione delle intersezioni. Il principale risultato è il file delle intersezioni, salvato con specifica estensione (`_intersezioni.txt`). Questo file raccoglie tutte le informazioni essenziali sui punti in cui la traiettoria si incrocia e sui segmenti coinvolti.

La struttura del file è organizzata in colonne chiare e significative:

- `X_inter1, Y_inter1`: coordinate della posizione dell'intersezione nello spazio bidimensionale. Permettono di localizzare esattamente il punto di incontro tra segmenti.
- `Tempo1, Tempo2`: istanti corrispondenti ai punti iniziale e finale dei segmenti che si intersecano. Queste informazioni temporali consentono di analizzare la sequenza e la durata degli incroci.
- `DT`: delta tempo tra i due punti di intersezione, utile per comprendere la dinamica del movimento (segmenti più rapidi o più lenti).
- `Idx1, Idx2`: indici dei punti originali dei segmenti coinvolti. Questi riferimenti permettono di collegare le intersezioni ai dati di partenza e di risalire facilmente alla traiettoria originale per eventuali verifiche o visualizzazioni grafiche.

Questo formato rende il file (`_intersezioni.txt`) immediatamente leggibile sia da persone sia da strumenti di analisi automatica, consentendo di integrare i risultati in ulteriori elaborazioni statistiche, grafici o mappe di movimento. L'ordinamento e la chiarezza delle colonne sono fondamentali per interpretare correttamente sia le coordinate spaziali sia la sequenza temporale delle intersezioni. Dal punto di vista di un approccio critico, le informazioni salvate nel file delle intersezioni non

servono solo a individuare punti geometrici di incrocio, ma offrono una prospettiva complessiva sul comportamento del movimento. Esse permettono di osservare ritorni della traiettoria su posizioni già percorse o su punti vicini, evidenziando schemi di oscillazione, deviazione o regressione del moto.

In particolare, grazie alle colonne temporali (Tempo1 e Tempo2) è possibile identificare i periodi in cui tali intersezioni si verificano, interpretandoli come veri e propri “chiusure” *time-dependent* della traccia su se stessa. Questo consente di analizzare fenomeni come pause, inversioni di direzione o movimenti ricorrenti in determinate aree, offrendo così una dimensione dinamica oltre a quella spaziale.

Inoltre, la presenza del delta-tempo (DT) tra i punti dei segmenti coinvolti permette di valutare la velocità relativa del ritorno: segmenti con DT piccoli indicano chiusure rapide, mentre DT maggiori evidenziano movimenti più lenti o dilatati nel tempo. Gli indici dei segmenti (Idx1, Idx2) consentono infine di collegare ciascuna intersezione ai dati originali, facilitando un’analisi critica che tenga conto delle misure effettivamente rilevate rispetto a eventuali punti ricostruiti o interpolati.

In sintesi, questi dati permettono non solo di descrivere la geometria della traiettoria, ma anche di esplorare la dinamica temporale dei ritorni, fornendo strumenti per un’interpretazione più completa e consapevole del movimento osservato. Questo approccio risulta particolarmente utile in contesti storici o sperimentali, dove la qualità e la risoluzione dei dati possono variare, e ogni intersezione diventa un indicatore del comportamento reale o della struttura del fenomeno studiato.

3. Il caso di studio

Come anticipato nell’introduzione, i dati utilizzati provengono da uno studio storico del 1904, finalizzato a monitorare gli spostamenti della Torre Garisenda di Bologna. Le relazioni tecniche dell’epoca, redatte con gli strumenti allora disponibili, si sono rivelate estremamente precise e rigorose, tanto da poter essere ancora oggi impiegate per caratterizzare la cinematica della torre. Per estrarre e analizzare questi dati, le tavole tecniche sono state digitalizzate [Pesci e Randazzo, in press] sia manualmente sia automaticamente mediante un software specificamente sviluppato [Randazzo e Pesci, submitted], che trasforma le misure (serie) originali in un formato digitale facilmente gestibile.

I dati originali invece, rilevati con la massima risoluzione consentita dal sistema di misura (composto da un sistema di filo a piombo e cannocchiali con lenti graduate), avevano una precisione di circa cinque centesimi di millimetro. Le misurazioni erano effettuate tre volte al giorno nei primi otto mesi dell’anno e una volta al giorno nei periodi successivi. I primi risultati relativi ai movimenti planimetrici furono organizzati in una matrice a specchio, che registrava le frequenze delle intersezioni e permetteva di confrontare il comportamento della torre sia all’interno di ciascun mese sia tra mesi diversi. Questa struttura permetteva di osservare i ritorni della traiettoria su posizioni già percorse, rivelando dettagli sorprendenti sulla frequenza dei passaggi, sui tempi di ritorno e sulla dinamica temporale degli spostamenti. Tutto ciò è stato ottenuto senza l’ausilio di strumenti di analisi moderni, dimostrando come le metodologie e l’accuratezza dei rilevatori dell’epoca fossero in grado di produrre dati estremamente significativi (Figura 1).

È importante sottolineare che, nel prosieguo, come caso di studio verranno presentati i risultati dell’analisi delle intersezioni derivati dalle osservazioni giornaliere, quindi con una risoluzione temporale inferiore rispetto a quella mostrata in precedenza, corrispondente alla parte della matrice successiva ad agosto. Va dunque anticipato che i risultati e l’eventuale confronto con l’analisi storica devono essere interpretati tenendo presente questo limite di risoluzione. Si anticipa tuttavia che, nonostante tali vincoli, l’analisi consentirà di preservare le informazioni storiche più significative, offrendo una lettura critica della cinematica osservata. I risultati saranno

confrontabili con le sintesi realizzate nel passato, mettendo in evidenza sia la coerenza delle osservazioni originali sia le dinamiche temporali dei ritorni della torre su posizioni già percorse.

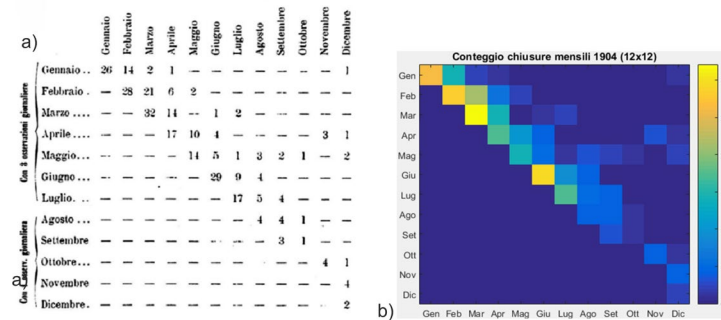


Figura 1 Diagramma delle intersezioni della traiettoria della Torre Garisenda: (a) chiusure osservate nel 1904 [Cavani, 1910]; (b) rappresentazione moderna della matrice con scala di colori. Si nota una riduzione dell'intensità dovuta al campionamento ridotto a partire dal mese di agosto.

Figure 1 Intersection diagram of the Torre Garisenda trajectory: a) closures observed in 1904 [Cavani, 1910]; b) modern representation with a color scale. A reduction in intensity is noticeable due to lower sampling starting in August.

3.1 Analisi dati

Per illustrare l'applicazione pratica del software, i dati planimetrici della torre vengono innanzitutto caricati dai file di testo *dest.txt* e *dnord.txt*, contenenti rispettivamente i valori temporali e gli spostamenti. Il software visualizza immediatamente la traccia originale come grafico planimetrico (Figura 2a), consentendo di osservare la distribuzione dei punti digitalizzati e le irregolarità presenti nel dato grezzo.

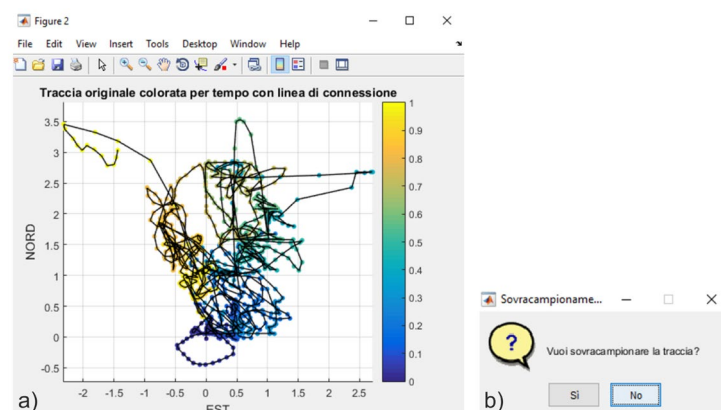


Figure 2 a) Traccia originale; b) finestra per attivare un eventuale sovracampionamento.

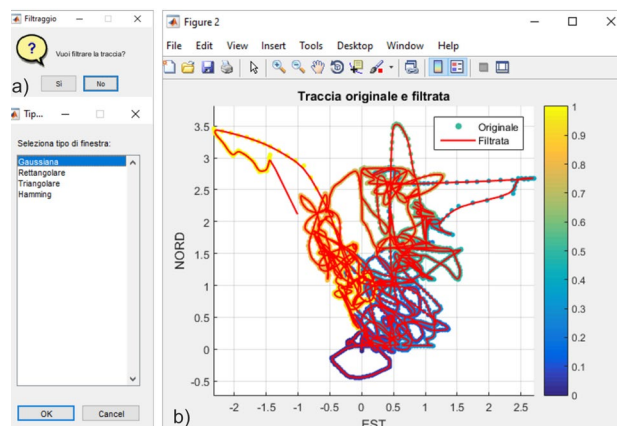
Figure 2 a) Original trace; b) window to enable optional oversampling.

A questo punto l'utente può scegliere di applicare un sovracampionamento tramite una finestra interattiva (Figura 2b). Questa operazione, del tutto opzionale, genera una linea continua più densa, utile per evidenziare i pattern di movimento senza modificare i valori originali. Il sovracampionamento può essere attivato o saltato a seconda delle necessità dell'analisi, rendendo il software flessibile rispetto ai diversi tipi di dati e alla risoluzione desiderata.

Successivamente è possibile applicare filtri selettivi alla traccia, scegliendo tra diversi tipi di finestre e parametri di *smoothing*. Anche queste operazioni sono opzionali: l'utente può confrontare immediatamente il risultato della filtratura con i punti originali (Figura 3b) per valutare quanto il rumore sia ridotto e quanto i movimenti principali della torre emergano più chiaramente.

Figura 3 a) Finestre di scelta per filtraggio, tipo di finestra e risultato (b) di traccia modificata sovrapposta ai punti originali.

Figure 3 a) Selection windows for filtering, type of window, and (b) result of the modified trace superimposed on the original points.

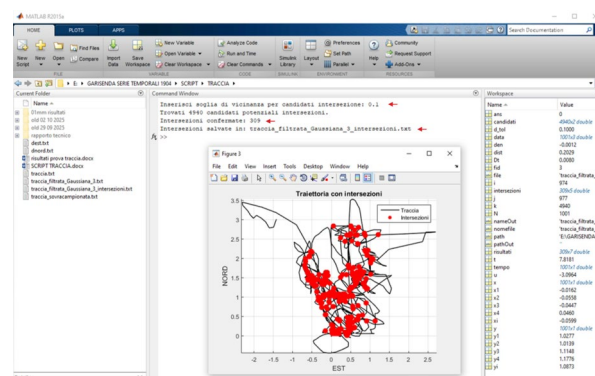


Questo approccio modulare consente di preservare la ricchezza dei dati originali, pur offrendo strumenti per migliorarne la leggibilità in termini di traiettorie planimetriche.

Una volta definita la traccia finale, il software individua i punti di intersezione lungo il percorso, utilizzando un intervallo di ricerca predefinito (nel caso specifico 0.1 mm). Le coordinate delle intersezioni vengono calcolate e visualizzate (Figura 4), mostrando le posizioni in cui la torre ritorna su punti già percorsi. Questi punti costituiscono la base per le analisi temporali, permettendo di studiare la frequenza e la durata dei cicli di ritorno.

Figura 4 Definizione intervallo ricerca (0.1 mm) e calcolo dei punti di intersezione nella traccia.

Figure 4 Definition of search interval (0.1 mm) and calculation of intersection points in the trace.



L'analisi dei tempi di ritorno si avvia selezionando il file delle intersezioni. Il software calcola gli intervalli DT tra i passaggi sugli stessi punti e li rappresenta in diverse modalità: come grafico dei tempi di ritorno rispetto ai tempi iniziali T1 (Figura 5b), come distribuzione delle frequenze dei DT (Figura 5c) e come panoramica complessiva dei dati (Figura 5d). Questa visualizzazione multipla consente di distinguere chiaramente le oscillazioni rapide dai grandi *loop* planimetrici e di mettere in evidenza cicli giornalieri, settimanali o mensili, in funzione della risoluzione temporale dei dati. L'approccio modulare adottato permette di partire dalla semplice visualizzazione dei dati grezzi, decidere quali operazioni opzionali applicare, come sovracampionamento o filtraggio, e arrivare fino all'analisi quantitativa dei movimenti. In questo modo si ottiene una rappresentazione completa delle caratteristiche planimetriche della torre, preparando i dati per confronti critici con osservazioni storiche o con studi analoghi, e facilitando l'interpretazione delle dinamiche temporali della struttura.

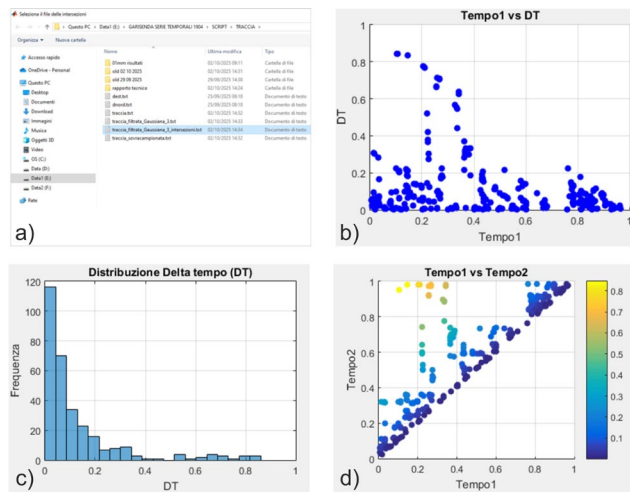


Figura 5 Analisi dei tempi di ritorno: a) selezione del file di intersezioni e panoramica dei dati; b) intervalli di tempo DT di ritorno rispetto ai tempi iniziali delle intersezioni (T1); c) distribuzione delle frequenze dei tempi di ritorno DT; d) tempi delle intersezioni e ampiezza DT.

Figure 5 Return time analysis: a) selection of the intersection file and overview of the data; b) return time intervals DT relative to the initial times of intersections (T1); c) frequency distribution of return times DT; d) intersection times and DT for amplitude.

3.2 Risultati e confronto

In ultima analisi, è stata creata una heatmap delle informazioni, rappresentata come matrice di 12×12 celle, con l'obiettivo di ricostruire in parte il dato così come era visualizzato nel diagramma a specchio realizzato nel 1904. Va sottolineato che il dato originale era molto più denso, poiché le osservazioni venivano effettuate tre volte al giorno, garantendo maggiore precisione, soprattutto per i movimenti a breve termine.

I dati utilizzati nel presente studio, invece, derivano dalle serie temporali digitalizzate degli spostamenti, considerando un solo punto al giorno, e pertanto la traccia risulta meno dettagliata. Nonostante questa limitazione, la heatmap mensile permette di evidenziare le chiusure e i ritorni della torre su posizioni già percorse, distinguendo le interazioni tra mesi consecutivi e non consecutivi e fornendo una visione sintetica ma significativa della dinamica planimetrica storica. La Figura 6 mostra i risultati per i due casi: sia le evidenze sia il numero delle intersezioni o chiusure avvenute in un dato periodo e con un determinato intervallo temporale. Si osserva un ottimo accordo con i dati storici, nonostante i limiti e le inevitabili conseguenze derivanti dal campionamento ridotto.

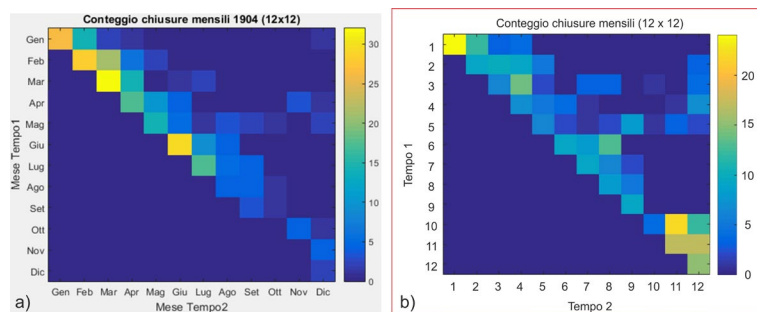


Figura 6 Comparazione tra le matrici dei risultati. a) rappresentazione degli incroci di traccia da diagramma originale; b) analisi ricavata dalla analisi dei dati giornalieri.

Figure 6 Comparison between the result matrices. a) Representation of track intersections from the original diagram; b) Analysis derived from the daily data analysis.

Confrontando la matrice storica del 1904 con quella ricavata dai dati digitalizzati emergono differenze significative dovute alla diversa frequenza di campionamento. Nella matrice originale, i primi sette mesi dell'anno (gennaio-luglio) presentano valori elevati sulla diagonale, con numerosi cicli chiusi entro lo stesso mese, grazie a tre misurazioni giornaliere (ore 8, 12 e 18). Questo consente di rilevare sia oscillazioni rapide sia micro-cicli giornalieri. Dopo agosto, la frequenza scende a una misura al giorno, con conseguente riduzione dei cicli brevi osservabili: la diagonale diventa meno densa e i valori cambiano sensibilmente.

Nella matrice moderna, ottenuta dalla traccia a sua volta ricavata dalle serie temporali con un solo dato al giorno (relativo alla misura effettuata alle ore 12), i cicli giornalieri non sono più rilevabili e i valori sulla diagonale dei primi mesi risultano notevolmente più bassi. Anche i cicli tra mesi consecutivi o distanti, corrispondenti a movimenti più lenti, risultano ridotti, soprattutto nei mesi estivi e autunnali (agosto-dicembre). Tuttavia, i grandi *loop* planimetrici della torre restano riconoscibili, sebbene con frequenze inferiori rispetto all'originale.

La differenza principale tra le due matrici riguarda appunto la risoluzione temporale: la matrice storica cattura sia i micro-cicli giornalieri sia i cicli mensili o stagionali, mentre quella moderna evidenzia soprattutto i movimenti di scala maggiore, perdendo i dettagli delle oscillazioni rapide. La dinamica complessiva della torre rimane coerente tra le due serie, ma la matrice moderna offre una rappresentazione più semplificata dei cicli planimetrici.

In via preliminare, si può osservare che il movimento della torre non è né casuale né rigido, ma combina una lenta deriva con cicli di diversa frequenza. Le oscillazioni molto rapide, corrispondenti a intervalli brevi (<0.2 anni $\approx 0-2.4$ mesi), mostrano un picco sugli intervalli più corti e sono probabilmente legate a vibrazioni, rumore o a limiti di precisione. Intervalli leggermente più lunghi ($0.2-0.4$ anni $\approx 2.4-4.8$ mesi) evidenziano oscillazioni più regolari, con frequenze distribuite in modo concentrato, riflettendo cicli strutturati di breve durata. Tra 0.5 e 0.8 anni ($\approx 6-9.6$ mesi) la distribuzione dei tempi di ritorno è più uniforme, mostrando componenti lente e meno definite, con alcuni intervalli privi di incroci.

Nel complesso, circa il 14% degli incroci evidenzia una deriva verso Nord, con oscillazioni medie quattro volte maggiori rispetto all'asse Est. Gli spostamenti variano da -0.69 mm a $+1.41$ mm, confermando che la torre subisce flessioni e deformazioni elastiche o plastiche, senza comportarsi come un pendolo rigido. Anche se i cicli dominanti non sono semplici da identificare, questa analisi permette di distinguere chiaramente le componenti veloci da quelle lente, offrendo una lettura dettagliata della dinamica planimetrica della torre.

4. Conclusioni

I dati storici, quando disponibili e accessibili, rappresentano una risorsa preziosa nello studio dei monumenti, soprattutto per strutture antiche come la Torre Garisenda, sottoposte nel tempo a sollecitazioni naturali e a interventi antropici. Essi consentono di ricostruire il comportamento passato della struttura e di interpretarne le dinamiche con un livello di dettaglio che spesso i rilievi moderni da soli non possono offrire.

I movimenti della Torre Garisenda, grazie ai dati del 1904 digitalizzati da grafici e annotazioni cartacee, sono stati analizzati tramite un codice sviluppato in ambiente MATLAB, concepito per essere semplice, modulare e versatile. Le routine implementate consentono di gestire l'intero flusso di lavoro: dalla lettura dei dati originali, alla visualizzazione della traiettoria planimetrica, alla rilevazione degli incroci, fino al calcolo degli intervalli temporali tra passaggi successivi e alla rappresentazione mediante heatmap dei cicli brevi e lunghi. La modularità e la flessibilità del metodo permettono di applicare lo stesso approccio anche ad altre serie storiche o a studi analoghi su diverse strutture.

L'analisi mostra che le osservazioni elaborate con il software sono coerenti con i dati storici

disponibili, evidenziando chiaramente cicli rapidi, settimanali e stagionali della torre. Questa linea di lavoro costituisce un tassello di un progetto più ampio, volto a costruire un quadro completo dei movimenti della Torre Garisenda e a renderli fruibili per confronti con la dinamica attuale. Le metodologie e i codici sviluppati sono modulabili e riutilizzabili, offrendo strumenti utili per l'analisi di altre serie planimetriche storiche, la verifica di modelli previsionali e lo studio comparativo delle dinamiche strutturali di edifici storici.

Il tempo e l'impegno dedicati all'estrazione dei dati di movimento da queste tavole storiche si rivelano particolarmente preziosi: una volta ottimizzati e resi fruibili tramite software dedicati, sia per lo studio delle tracce sia per l'analisi delle serie temporali, essi forniscono una rappresentazione cristallizzata dei comportamenti della Torre all'inizio del XX secolo. Questi dati costituiscono un termine di paragone fondamentale qualora si decidesse di ripetere le misure, non più con metodi tradizionali come filo a piombo o sistemi ottici di osservazione, ma con strumenti moderni quali laser, sensori fotosensibili e trasduttori, capaci di registrare un intero anno di spostamenti. In questo modo sarebbe possibile confrontare ampiezza e modalità di oscillazione attuali con quelle del passato, rilevando eventuali cambiamenti significativi nel comportamento strutturale della torre.

Bibliografia

- Barazzetti, L., Roncorononi, F. (2011). Relazione sulle misure eseguite per il controllo delle deformazioni del Duomo di Milano. *Technical report*, Politecnico di Milano.
- Bentley, J.L., & Ottmann, T.A. (1979). Algorithms for reporting and counting geometric intersections. *IEEE Transactions on Computers*, C-28(9), 643-647. <https://doi.org/10.1109/TC.1979.1675432>
- Borresen, J.L., Andersen, O., Jensen, C.S., & Torp, K. (2017). Interactive intersection analysis using trajectory data. In *Proceedings of the 25th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, Article 87, pp. 1-4. <https://doi.org/10.1145/3139958.3140041>
- Cavani, F. (1910). *Movimenti della sommità rispetto alla base nella torre Garisenda di Bologna*. Memoria letta nella seduta del 22 maggio 1910, Collegio degli Ingegneri e Architetti di Bologna, Bologna (con tavola).
- Di Giacomo, M. (2022). *Analisi multidimensionale dei dati acquisiti dal monitoraggio della torre Garisenda e relativa modellazione agli elementi finiti*. Politecnico di Torino, Corso di Laurea Magistrale in Ingegneria Civile.
- Elyamani, A., Caselles, O., Roca, P., Clapes, J. (2017). Dynamic investigation of a large historical cathedral. *Structural Control and Health Monitoring*, 24(3), e1885. <https://doi.org/10.1002/stc.1885>
- Gentile, C., Ruccolo, A., Canali, F. (2019). Continuous monitoring of the Milan cathedral: dynamic characteristics and vibration-based SHM. *Journal of Civil Structural Health Monitoring*, 9(5), 671-688. <https://doi.org/10.1007/s13349-019-00361-8>
- MathWorks (2024). MATLAB version R2024b. *The MathWorks Inc.*, Natick, Massachusetts. Available at: <https://www.mathworks.com>
- Malcangio, V.C. (2022). *Studio del comportamento strutturale di torri medievali: il caso del campanile di San Giacomo Maggiore in Bologna*. Tesi di Laurea, Università di Bologna, DISTART, Facoltà di Ingegneria.
- Pesci, A., & Randazzo, D. (2025). The Garisenda Tower in 1904: displacement time series. *Quaderni di Geofisica*, 198, 34 pp. <https://doi.org/10.13127/qdg/198>
- Randazzo, D., & Pesci, A. (in press). PyGraph Scanner: an interactive tool for digitizing historical series. *Rapporti Tecnici INGV*.

- Ruccolo A., Gentile C., Canali F. (2021). Monitoring an iconic heritage structure with OMA: the main spire of the Milan cathedral. *Smart Structures and Systems*, 27(2):305–318. <https://doi.org/10.12989/sss.2021.27.2.305>
- Wang, L., & Rui, X. (2024). A novel fuzzy inference method for urban incomplete road weight assignment. *Geo-spatial Information Science*, 27(6), 2008–2022. <https://doi.org/10.1080/10095020.2023.2261768>

APPENDICE A

Mini manuale d'uso per le analisi delle tracce planimetriche

Questo script MATLAB è stato sviluppato per analizzare i movimenti planimetrici della Torre Garisenda sulla base dei dati storici acquisiti nel 1904. Il programma consente di:

1. Caricare da file .txt le coordinate EST e NORD della torre.
2. Visualizzare le traiettorie originali in un grafico bidimensionale con rappresentazione cromatica della scala temporale.
3. Applicare procedure di sovracampionamento e filtraggio dei dati per migliorarne l'analisi.
4. Individuare i punti di intersezione lungo la traiettoria e calcolare i relativi tempi di ritorno.
5. Generare grafici e mappe di calore (heatmap) per l'analisi dei cicli di movimento a breve e lungo termine.
6. Struttura del codice

Il flusso di elaborazione si articola nelle seguenti fasi:

1. Selezione della cartella di lavoro e dei file contenenti i dati.
2. Preparazione e visualizzazione della traiettoria originale.
3. Eventuale sovracampionamento dei dati.
4. Applicazione dei filtri disponibili (Gaussiano, Rettangolare, Triangolare e Hamming).
5. Identificazione delle intersezioni e calcolo dei tempi di ritorno.
6. Generazione di grafici e heatmap, inclusa una heatmap mensile di dimensioni 12×12 .
7. File di output

Il programma genera automaticamente i seguenti file:

1. traccia.txt oppure traccia_sovracampionata.txt: dati della traiettoria elaborata;
2. traccia_filtrata_<tipoFiltro>_<larghezza>.txt: dati filtrati in funzione del filtro e della larghezza selezionati;
3. <nomefile>_intersezioni.txt: elenco delle intersezioni individuate, comprensivo di coordinate, istanti temporali e valori di ΔT .

Note operative

Il codice è strutturato in modo modulare e può essere facilmente adattato all'analisi di altre serie storiche o di strutture differenti.

Le figure e i grafici vengono generati automaticamente nelle rispettive sezioni del programma. Tutti i file di output sono salvati nella cartella di lavoro selezionata dall'utente all'inizio dell'elaborazione.

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% --- 1) Scegliere la cartella di lavoro ---
cartella = uigetdir(pwd, 'Scegli la cartella di lavoro');
if isequal(cartella,0)
    error('Nessuna cartella selezionata.');
```

end

```

cd(cartella);
fprintf('Cartella di lavoro selezionata: %s\n', cartella);

%% --- 2) Selezione dei file EST e NORD ---
[fileEst, pathEst] = uigetfile('*.txt', 'Seleziona il file EST', cartella);
if isequal(fileEst,0)
    error('Nessun file EST selezionato.');
```

end

```

[fileNord, pathNord] = uigetfile('*.txt', 'Seleziona il file NORD', cartella);
if isequal(fileNord,0)
    error('Nessun file NORD selezionato.');
```

end

```

fullpathEst = fullfile(pathEst, fileEst);
fullpathNord = fullfile(pathNord, fileNord);

%% --- 3) Controllo limiti temporali e passo ---
% Carichiamo i dati e controlliamo la corrispondenza tra EST e NORD
dataEst = load(fullpathEst);
dataNord = load(fullpathNord);
tEst = dataEst(:,1);
tNord = dataNord(:,1);

if length(tEst) ~= length(tNord) || any(tEst ~= tNord)
    warning('I tempi tra EST e NORD non coincidono perfettamente.');
```

end

```

dtEst = diff(tEst);
dtNord = diff(tNord);
if max(abs(dtEst - dtNord)) > eps
    warning('I passi temporali tra EST e NORD differiscono.');
```

end

```

fprintf('Dati caricati. Numero punti EST: %d, NORD: %d\n', length(tEst), length(tNord));

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%% --- 4) Creare traccia.txt con tempo, EST, NORD ---
% Assicuriamoci che siano vettori colonna
tEst = tEst(:);
tNord = tNord(:);
dataEst = dataEst(:,2); % estraiamo solo colonna valori
dataNord = dataNord(:,2);

% Variabili iniziali per traccia
```

```

estData = dataEst;
nordData = dataNord;

% Salvataggio file traccia originale
traccia = [tEst, estData, nordData];
outFile = fullfile(cartella, 'traccia.txt');
dlmwrite(outFile, traccia, 'delimiter', '\t', 'precision', 6);
fprintf('File traccia.txt salvato in: %s\n', outFile);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%% --- 5) Visualizzazione della traiettoria 2D ---
% a) Scatter plot semplice colorato per tempo
figure; hold on;
scatter(estData, nordData, 20, tEst, 'filled');
colorbar;
xlabel('EST'); ylabel('NORD'); title('Traccia originale colorata per tempo');
grid on; axis equal;
hold off;

% b) Scatter plot con linea di connessione
figure; hold on;
scatter(estData, nordData, 20, tEst, 'filled');
plot(estData, nordData, 'k-', 'LineWidth', 1); % linea nera
colorbar;
xlabel('EST'); ylabel('NORD');
title('Traccia originale colorata per tempo con linea di connessione');
grid on; axis equal;
hold off;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%% --- 6) Sovracampionamento della traccia (opzionale) ---
risposta = questdlg('Vuoi sovracampionare la traccia?', 'Sovracampionamento', 'SÃ', 'No',
'No');
if strcmp(risposta, 'SÃ')
    fattore = input('Inserisci il fattore di sovracampionamento (2,3,...): ');

    % Interpolazione lineare per sovracampionamento
    tNew = linspace(tEst(1), tEst(end), length(tEst)*fattore);
    estNew = interp1(tEst, estData, tNew, 'linear');
    nordNew = interp1(tNord, nordData, tNew, 'linear');

    % Aggiorniamo dati
    tEst = tNew(:);
    tNord = tNew(:);
    estData = estNew(:);
    nordData = nordNew(:);

    fprintf('Sovracampionamento applicato. Nuovo numero punti: %d\n', length(tEst));

```

```

else
    fprintf('Traccia mantenuta invariata.\n');
end

% Salvataggio file sovracampionata
traccia = [tEst, estData, nordData];
outFile = fullfile(cartella, 'traccia_sovracampionata.txt');
dlmwrite(outFile, traccia, 'delimiter','\t','precision',6);
fprintf('File traccia_sovracampionata.txt salvato in: %s\n', outFile);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%% --- 7) Filtraggio della traccia (opzionale) ---
rispostaFil = questdlg('Vuoi filtrare la traccia?', 'Filtraggio', 'SÃ-', 'No', 'No');
if strcmp(rispostaFil,'SÃ-')

    % Selezione tipo di finestra
    tipiFinestra = {'Gaussiana','Rettangolare','Triangolare','Hamming'};
    [idxTipo, ok] = listdlg('PromptString','Seleziona tipo di finestra:', ...
                           'SelectionMode','single', ...
                           'ListString', tipiFinestra, ...
                           'Name','Tipo finestra');

    if ~ok
        error('Nessun filtro selezionato. Uscita.');
```

```
end
```

```
tipoFiltro = tipiFinestra{idxTipo};
```

```
% Inserimento larghezza finestra
```

```
larghezza = input('Inserisci larghezza finestra (numero punti, dispari consigliato): ');
```

```
if mod(larghezza,2) == 0
```

```
    warning('Hai inserito un numero pari, il filtro potrebbe essere leggermente asimmetrico.');
```

```
end
```

```
% Costruzione finestra
```

```
switch tipoFiltro
```

```
    case 'Gaussiana'
```

```
        sigma = (larghezza-1)/6;
```

```
        x = linspace(-((larghezza-1)/2), ((larghezza-1)/2), larghezza);
```

```
        filtro = exp(-0.5*(x/sigma).^2)';
```

```
    case 'Rettangolare'
```

```
        filtro = ones(larghezza,1);
```

```
    case 'Triangolare'
```

```
        f = ones(ceil(larghezza/2),1);
```

```
        filtro = conv(f,f);
```

```
    case 'Hamming'
```

```
        n = (0:larghezza-1)';
```

```
        filtro = 0.54 - 0.46*cos(2*pi*n/(larghezza-1));
```

```
end
```

```
filtro = filtro / sum(filtro); % normalizzazione
```

```

% Applicazione filtro
estFiltrata = conv(estData, filtro, 'same');
nordFiltrata = conv(nordData, filtro, 'same');

% Controllo dimensioni
if length(estFiltrata) ~= length(tEst) || length(nordFiltrata) ~= length(tEst)
    error('Errore: dimensioni dei dati filtrati non corrispondono al tempo.');
```

end

```

% Salvataggio file filtrata
tracciaFiltrata = [tEst(:), estFiltrata(:), nordFiltrata(:)];
outFileFil = fullfile(cartella, ['traccia_filtrata_', tipoFiltro, '_', num2str(larghezza),
'.txt']);
dlmwrite(outFileFil, tracciaFiltrata, 'delimiter', '\t', 'precision', 6);
fprintf('File traccia filtrata salvato in: %s\n', outFileFil);

% Plot comparativo originale vs filtrata
figure; hold on;
scatter(estData, nordData, 20, tEst, 'filled'); % originale
plot(estFiltrata, nordFiltrata, 'r', 'LineWidth', 1.5); % filtrata
colorbar; xlabel('EST'); ylabel('NORD'); title('Traccia originale e filtrata');
legend('Originale', 'Filtrata'); axis equal; grid on;
hold off;
else
    fprintf('Filtraggio non applicato.\n');
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%% --- Script per trovare intersezioni nella traiettoria 2D ---
clear; clc;

% 1) Caricamento traccia pronta (tempo, EST, NORD)
[file, path] = uigetfile('*.txt', 'Seleziona il file traccia (tempo EST NORD)');
if isequal(file,0)
    error('Nessun file selezionato.');
```

end

```

data = load(fullfile(path, file));
tempo = data(:,1);
x = data(:,2); % EST
y = data(:,3); % NORD
N = length(tempo);

% 2) Parametri
d_tol = input('Inserisci soglia di vicinanza per candidati intersezione: ');

% 3) Identificazione candidati vicini
candidati = [];
for i = 1:N-2
    for j = i+2:N-1
```

```

        dist = sqrt( (x(i)-x(j))^2 + (y(i)-y(j))^2 );
        if dist < d_tol
            candidati = [candidati; i j]; %#ok<AGROW>
        end
    end
end
fprintf('Trovati %d candidati potenziali intersezioni.\n', size(candidati,1));

% 4) Verifica intersezione segmenti
intersezioni = [];
for k = 1:size(candidati,1)
    i = candidati(k,1);
    j = candidati(k,2);

    % Segmenti
    x1 = x(i); y1 = y(i);
    x2 = x(i+1); y2 = y(i+1);
    x3 = x(j); y3 = y(j);
    x4 = x(j+1); y4 = y(j+1);

    % Determinante per intersezione
    den = (x1-x2)*(y3-y4) - (y1-y2)*(x3-x4);
    if den == 0
        continue; % paralleli
    end

    % Parametri dell'intersezione
    t = ((x1-x3)*(y3-y4) - (y1-y3)*(x3-x4)) / den;
    u = -((x1-x2)*(y1-y3) - (y1-y2)*(x1-x3)) / den;

    if t >= 0 && t <= 1 && u >= 0 && u <= 1
        xi = x1 + t*(x2-x1);
        yi = y1 + t*(y2-y1);
        Dt = tempo(j) - tempo(i);
        intersezioni = [intersezioni; xi yi Dt i j]; %#ok<AGROW>
    end
end
fprintf('Intersezioni confermate: %d\n', size(intersezioni,1));

% 5) Plot intersezioni sulla traiettoria
figure; hold on;
plot(x, y, 'k-', 'LineWidth', 1.2);
if ~isempty(intersezioni)
    scatter(intersezioni(:,1), intersezioni(:,2), 80, 'r', 'filled');
end
xlabel('EST'); ylabel('NORD'); title('Traiettoria con intersezioni');
grid on; axis equal;
legend('Traccia', 'Intersezioni');

% 6) Salvataggio risultati

```

```

if ~isempty(intersezioni)
    risultati = [intersezioni(:,1:2), tempo(intersezioni(:,4)), tempo(intersezioni(:,5)+1),
intersezioni(:,3), intersezioni(:,4:5)];
    [pathOut,nameOut] = fileparts(file);
    nomefile = fullfile(pathOut, [nameOut '_intersezioni.txt']);
    fid = fopen(nomefile,'w');
    fprintf(fid, 'X_inter1\tY_inter1\tTempo1\tTempo2\tDT\tIdx1\tIdx2\n');
    fclose(fid);
    dlmwrite(nomefile, risultati, '-append','delimiter','\t','precision',6);
    fprintf('Intersezioni salvate in: %s\n', nomefile);
else
    fprintf('Nessuna intersezione trovata. Nessun file salvato.\n');
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%% --- Analisi intersezioni: caricamento file ---
[file, path] = uigetfile('*.txt', 'Seleziona il file delle intersezioni');
if isequal(file,0)
    error('Nessun file selezionato.');
```

end

```

nomefile = fullfile(path, file);
T = readtable(nomefile, 'FileType','text', 'Delimiter','\t', 'HeaderLines',0);
X1 = T.X_inter1; Y1 = T.Y_inter1;
Tempo1 = T.Tempo1; Tempo2 = T.Tempo2; DT = T.DT;

% Grafici individuali
figure; plot(Tempo1, X1, 'bo', 'MarkerFaceColor','b'); xlabel('Tempo'); ylabel('EST');
title('Intersezioni: Tempo vs EST'); grid on;
figure; plot(Tempo1, Y1, 'ro', 'MarkerFaceColor','r'); xlabel('Tempo'); ylabel('NORD');
title('Intersezioni: Tempo vs NORD'); grid on;
figure; plot(Tempo1, DT, 'ks-', 'LineWidth',1.2, 'MarkerFaceColor','k'); xlabel('Tempo1');
ylabel('DT'); title('Delta tempo vs Tempo1 (inizio segmento)'); grid on;
figure; plot(Tempo2, DT, 'ms-', 'LineWidth',1.2, 'MarkerFaceColor','m'); xlabel('Tempo2');
ylabel('DT'); title('Delta tempo vs Tempo2 (fine segmento)'); grid on;

% Grafici combinati
figure('Name','Analisi intersezioni','Position',[100 100 1000 700]);
subplot(2,2,1); scatter(Tempo1, Tempo2, 40, DT, 'filled'); colorbar; xlabel('Tempo1');
ylabel('Tempo2'); title('Tempo1 vs Tempo2'); grid on;
subplot(2,2,2); scatter(Tempo1, DT, 40, 'b', 'filled'); xlabel('Tempo1'); ylabel('DT');
title('Tempo1 vs DT'); grid on;
subplot(2,2,3); scatter(Tempo2, DT, 40, 'r', 'filled'); xlabel('Tempo2'); ylabel('DT');
title('Tempo2 vs DT'); grid on;
subplot(2,2,4); histogram(DT, 20); xlabel('DT'); ylabel('Frequenza'); title('Distribuzione
Delta tempo (DT)'); grid on;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%% --- Heatmap intersezioni: standard e mensile ---

```

```

% Caricamento dati intersezioni
[file, path] = uigetfile('*.txt', 'Seleziona il file delle intersezioni');
if isequal(file,0)
    error('Nessun file selezionato.');
```

end

```

T = readtable(fullfile(path,file), 'FileType','text', 'Delimiter','\t');
Tempo1 = T.Tempo1; Tempo2 = T.Tempo2;

% Heatmap standard con hist3
nBinX = 50; nBinY = 50;
data = [Tempo1, Tempo2];
[counts, edges] = hist3(data, 'Nbins', [nBinX nBinY]);
figure; imagesc(edges{1}, edges{2}, counts'); axis xy; colorbar;
xlabel('Tempo1'); ylabel('Tempo2'); title('Heatmap delle intersezioni Tempo1 vs Tempo2
(MATLAB2015)');
```

% Heatmap mensile 12x12

```

Tmin = min([Tempo1; Tempo2]); Tmax = max([Tempo1; Tempo2]);
t1_norm = (Tempo1 - Tmin) / (Tmax - Tmin);
t2_norm = (Tempo2 - Tmin) / (Tmax - Tmin);
nMesi = 12;
edgesMesi = linspace(0,1,nMesi+1);
idx1 = discretize(t1_norm, edgesMesi); idx2 = discretize(t2_norm, edgesMesi);
heatMatMesi = zeros(nMesi,nMesi);
for k = 1:length(idx1)
    i = idx1(k); j = idx2(k);
    if ~isnan(i) && ~isnan(j)
        heatMatMesi(i,j) = heatMatMesi(i,j) + 1;
    end
end

figure; imagesc(1:nMesi, 1:nMesi, heatMatMesi); set(gca,'YDir','reverse'); axis tight; colorbar;
xlabel('Mese Tempo2'); ylabel('Mese Tempo1'); title('Conteggio chiusure mensili (12x12)');
set(gca,'XTick',1:nMesi,'YTick',1:nMesi);
```

APPENDICE B

Codice accessorio per la preparazione e formattazione

PreDati è un programma ausiliario sviluppato in MATLAB che ha lo scopo di preparare automaticamente i dati necessari per l'esecuzione del software principale di analisi delle tracce planimetriche.

Il software semplifica le operazioni preliminari di gestione dei dati, consentendo all'utente di utilizzare file provenienti da diverse sorgenti senza dover effettuare conversioni manuali.

In particolare, il programma permette di:

1. Caricare dati da file nei formati TXT, CSV, XLS e XLSX.
2. Riconoscere automaticamente la struttura del file, sia nel caso di dati costituiti dalle sole coordinate EST e NORD, sia nel caso in cui sia presente anche l'informazione temporale.
3. Verificare la regolarità della serie temporale e, se necessario, uniformare il campionamento mediante interpolazione lineare.
4. Generare automaticamente i file standard di input richiesti dal programma principale:
5. est.txt, contenente tempo e coordinata EST;
6. nord.txt, contenente tempo e coordinata NORD.

L'utilizzo del programma è semplice e guidato. Dopo l'avvio dello script, l'utente deve selezionare il file contenente i dati da elaborare. Il software esegue automaticamente i controlli necessari e produce i file di output nella stessa cartella del file originale.

I file generati risultano immediatamente compatibili con il software principale per l'analisi delle tracce planimetriche, eliminando la necessità di ulteriori operazioni di formattazione o preparazione dei dati.

Lo scopo di PreDati è rendere più agevole e affidabile la fase di preparazione dei dati, riducendo il rischio di errori dovuti a formati non standard o a campionamenti temporali non uniformi. In questo modo l'utente può concentrarsi direttamente sull'analisi dei movimenti, senza doversi occupare delle operazioni preliminari di conversione e verifica dei dati.

```
% *****
% * PreDati.m
% *
% * Codice di pre-processing per la preparazione dei dati di input
% * del software principale di analisi delle tracce planimetriche.
% *
% * Funzionalità principali:
% * - Caricamento di serie temporali da file TXT, CSV, XLS e XLSX;
% * - Riconoscimento automatico di file con:
% *   • 2 colonne (EST, NORD)
% *   • 3 colonne (TEMPO, EST, NORD)
% * - Controllo e uniformazione della serie temporale mediante
% *   interpolazione lineare, in caso di campionamento irregolare;
% * - Creazione automatica dei file standard di input:
% *   • est.txt (TEMPO, EST)
% *   • nord.txt (TEMPO, NORD)
% *
```

```

% * Modalità d'uso:
% * - Eseguire lo script e selezionare il file dei dati richiesto;
% * - I file est.txt e nord.txt vengono generati nella stessa cartella
% *   del file di input e sono immediatamente pronti per l'analisi.
% *
% * Ambiente di sviluppo: MATLAB (compatibile con versioni 2015 e successive)
% *****

%% PreDati.m - caricamento dati per il software principale

clear; clc;

[filename, pathname] = uigetfile({'*.txt;*.csv;*.xls;*.xlsx'}, 'Seleziona il file dei dati');
if isequal(filename,0)
    disp('Nessun file selezionato. ');
    return;
end

fullFile = fullfile(pathname, filename);
[~,~,ext] = fileparts(filename);

%% Lettura dati
switch lower(ext)
    case {'*.txt'}
        data = dlmread(fullFile);
    case {'*.csv'}
        % Prova a leggere con virgola, se fallisce usa punto e virgola
        try
            data = dlmread(fullFile, ',');
        catch
            data = dlmread(fullFile, ';');
        end
    case {'*.xls', '*.xlsx'}
        [~, ~, raw] = xlsread(fullFile);
        data = cell2mat(raw);
    otherwise
        error('Formato file non supportato. ');
end

[nRows, nCols] = size(data);
fprintf('File caricato: %s\n', filename);
fprintf('Numero righe: %d, numero colonne: %d\n', nRows, nCols);

if nCols == 2
    % Assume colonne EST e NORD
    t = (0:nRows-1)'; % crea un vettore tempo fittizio se non presente
    x = data(:,1);
    y = data(:,2);
    disp('Rilevate 2 colonne: EST e NORD. Generata colonna tempo artificiale. ');
elseif nCols == 3

```

```
% Assume colonne TEMPO, EST, NORD
t = data(:,1);
x = data(:,2);
y = data(:,3);
fprintf('Rilevate 3 colonne: TEMPO, EST, NORD.\n');
fprintf('Tempo minimo: %.4f, Tempo massimo: %.4f, Intervallo medio: %.4f\n', ...
        min(t), max(t), mean(diff(t)));
else
    error('Numero colonne non supportato. Il file deve avere 2 o 3 colonne.');
```

```
end

%% Uniformazione serie temporale
dt = mean(diff(t));
t_uniform = t(1):dt:t(end);
x_uniform = interp1(t, x, t_uniform, 'linear');
y_uniform = interp1(t, y, t_uniform, 'linear');

fprintf('Serie temporale uniformata. Nuovo numero di punti: %d\n', length(t_uniform));

%% Creazione file est.txt e nord.txt
estFile = fullfile(pathname, 'est.txt');
nordFile = fullfile(pathname, 'nord.txt');

dlmwrite(estFile, [t_uniform' x_uniform'], 'delimiter','\t');
dlmwrite(nordFile, [t_uniform' y_uniform'], 'delimiter','\t');

fprintf('File generati:\n - %s\n - %s\n', estFile, nordFile);
disp('PreDati completato con successo.');
```



QUADERNI di GEOFISICA

ISSN 1590-2595

<https://collane-editoriali.ingv.it/index.php/qdg/it/index/>

QUADERNI di GEOFISICA è una rivista scientifica che pubblica dati, misurazioni e osservazioni geofisiche (anche preliminari) in italiano e inglese. L'obiettivo è una diffusione rapida e gratuita per la comunità scientifica, garantita da un processo di peer-review e da un Editorial Board multidisciplinare. La rivista è inclusa in "Emerging Sources Citation Index" di Clarivate Analytics e "Open Access Journals" di Scopus, e adotta il modello Diamond Open Access, che non prevede costi né per gli autori né per i lettori.

QUADERNI di GEOFISICA is a scientific journal that publishes geophysical data, measurements and observations, including preliminary results, in Italian and English. Its aim is to ensure rapid and free dissemination to the research community, supported by a peer-review process and by a multidisciplinary Editorial Board. The journal is included in the Emerging Sources Citation Index of Clarivate Analytics and it's included in the Open Access journals of Scopus. It follows the Diamond Open Access model, with no charges for either authors or readers.



RAPPORTI TECNICI INGV

ISSN 2039-7941

<https://collane-editoriali.ingv.it/index.php/rt/it/index>

RAPPORTI TECNICI INGV è una rivista tecnico-scientifica che mette in mostra le numerose attività tecnologiche e di ricerca condotte dai ricercatori e tecnici INGV, e non solo. Pubblicata in italiano e inglese, RAPPORTI TECNICI INGV è ideata per valorizzare anche quei lavori che offrono un contributo significativo dal punto di vista tecnico, operativo o documentale fondamentale per lo sviluppo dell'Ente. La rivista adotta il modello Diamond Open Access, che non prevede costi né per gli autori né per i lettori.

RAPPORTI TECNICI INGV is a technical-scientific journal that showcases the wide range of technological and research activities carried out by INGV researchers and technical staff, as well as by external contributors. Published in Italian and English, RAPPORTI TECNICI INGV is designed to highlight works that provide a significant contribution from a technical, operational, or documentary perspective, and that are essential to INGV development. The journal follows the Diamond Open Access model, with no charges for either authors or readers.



MISCELLANEA INGV

ISSN 2039-6651

<https://collane-editoriali.ingv.it/index.php/mi/it/index>

MISCELLANEA INGV rappresenta una tipologia di editoria non tradizionalmente catalogabile nelle ordinarie riviste scientifiche di settore. Pubblicata in lingua italiana e inglese, MISCELLANEA INGV offre uno spazio per la pubblicazione di materiali eterogenei che, pur non rientrando nei formati più strutturati delle altre Collane editoriali INGV, rivestono un interesse scientifico, tecnico o divulgativo. La rivista adotta il modello Diamond Open Access, che non prevede costi né per gli autori né per i lettori.

MISCELLANEA INGV represents a form of publishing that cannot be classified within the standard scientific journals. Published in Italian and English, MISCELLANEA INGV provides a venue for the publication of heterogeneous materials that are of scientific, technical or outreach interest, even though they do not fall within the more structured formats of the other INGV editorial series. The journal follows the Diamond Open Access model, with no charges for either authors or readers.

Progetto grafico
Barbara ANGIONI
Istituto Nazionale di Geofisica e Vulcanologia

Ufficio Editoriale
Francesca DI STEFANO | Coordinatore

Segreteria di redazione
segreteria.collane-editoriali@ingv.it

Editing | Produzione | Grafica redazionale
Barbara ANGIONI
Massimiliano CASCONI
Francesca DI STEFANO
Patrizia PANTANI

© 2026 the Author(s). Open Access.
This article is licensed under a Creative Commons Attribution 4.0 International License.



Creative Commons Attribution 4.0 International



ISTITUTO NAZIONALE DI GEOFISICA E VULCANOLOGIA