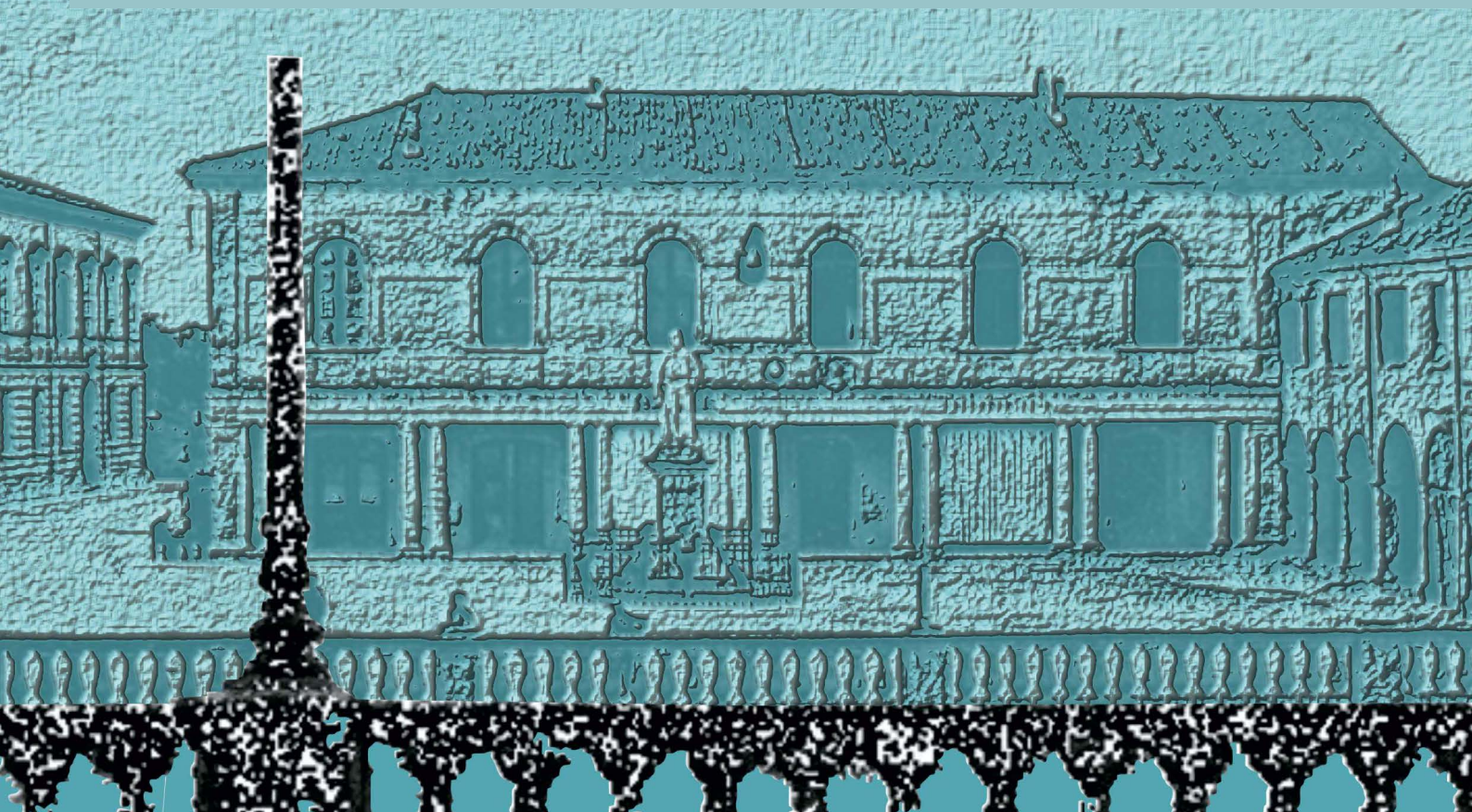




QUADERNI di GEOFISICA

Texturizzazione controllata di immagini
storiche su modelli tridimensionali



ISTITUTO NAZIONALE DI GEOFISICA E VULCANOLOGIA

203

**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

Stefano PINTORE

Pierangelo ROMANO

Paola SBARRA

Davide SCAFIDI

Maurizio SOLDANI

Sara STOPPONI

Umberto TAMMARO

David ZULIANI

REGISTRAZIONE AL TRIBUNALE DI ROMA N.174 | 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

QUADERNI di GEOFISICA

Texturizzazione controllata di immagini
storiche su modelli tridimensionali

*Controlled Texturing of Historical Images on
Three-Dimensional Models*

Arianna Pesci^{1*}, Giordano Teza², Fabiana Loddo¹, Alessandra Rossetti¹

INGV | Istituto Nazionale di Geofisica e Vulcanologia, Sezione di Bologna
Università degli Studi di Bologna, Dipartimento di Fisica e Astrofisica

**Corresponding author*

Accettato 22 giugno 2026 | Accepted 22 June 2026

Come citare | How to cite Pesci, A., Teza, G., Loddo, F., & Rossetti, A. (2026). Texturizzazione controllata di immagini storiche su modelli tridimensionali. *Quaderni di Geofisica*, 203, 70 pp. <https://doi.org/10.13127/qdg/203>

In copertina Immagine Concettuale di Arianna Pesci | Cover Conceptual Image by Arianna Pesci

203

INDICE

Riassunto	7
<i>Abstract</i>	7
Introduzione	7
1. Metodologia	9
1.1 <i>Pipeline</i>	10
1.2 Trasformazione affine	12
1.2.1 Calcolo dell'errore	14
1.3 Proiezione da 3D a piano immagine	16
2. Introduzione al software	17
2.1 PreCloud: script accessorio	17
2.2 PreImg: script accessorio	19
2.3 Historica: script principale	20
2.3.1 Alcune note tecniche	24
2.3.2 Consigli operativi	26
3. I casi di studio	26
3.1 Caso 1 - San Francesco	27
3.1.1 Analisi preliminari	29
3.1.2 <i>Texturing</i> di San Francesco	31
3.2 Caso 2 - Porto canale di Cesenatico	32
3.3 Caso 3 - Immagine termica (RGB)	34
3.4 Caso aggiuntivo	36
4. Conclusioni	37
Disponibilità di dati e codici	38
Ringraziamenti	38
Bibliografia	38
Sitografia	39
Appendice A	41
Appendice B	45
Appendice C	51

Riassunto

Il lavoro presenta una procedura guidata dall'utente per il trasferimento di informazioni cromatiche da immagini storiche a nuvole di punti 3D moderne. Il metodo, sviluppato come evoluzione di approcci precedenti, prevede l'orientamento della nuvola secondo la vista storica, la generazione di un'immagine sintetica prospettica, la selezione di punti omologhi e l'applicazione di una trasformazione affine bidimensionale, seguita dalla retro-proiezione dei valori cromatici sulla nuvola di punti.

La pipeline software, implementata in ambiente MATLAB, è modulare e composta da tre script indipendenti (PreCloud, Prelmg e Historica), consentendo un controllo esplicito delle principali fasi del processo e una facile adattabilità a differenti tipologie di dati. Inoltre, il metodo integra criteri quantitativi per la valutazione della qualità dell'allineamento, basati sull'analisi dei residui e sull'identificazione di *outlier* mediante soglie statistiche.

La procedura è pensata per garantire ripetibilità e coerenza metrica, anche in presenza di immagini storiche caratterizzate da qualità eterogenea e geometrie non note.

L'approccio consente di integrare informazioni iconografiche, fotografiche o cartografiche storiche in modelli tridimensionali moderni, rendendo tali dati confrontabili con osservazioni attuali e utilizzabili per analisi quantitative.

Abstract

This paper presents a user-guided procedure for transferring chromatic information from historical images to modern 3D point clouds. The method, developed as an evolution of previous approaches, involves the orientation of the point cloud according to the historical viewpoint, the generation of a synthetic perspective image, the selection of homologous points, and the application of a two-dimensional affine transformation, followed by back-projection of color information onto the point cloud. In addition, the method incorporates quantitative criteria for assessing alignment quality, based on residual analysis and the identification of outliers using statistical thresholds

The software pipeline is implemented in MATLAB and is modular, consisting of three independent scripts (PreCloud, Prelmg and Historica), allowing explicit control of the main processing stages and adaptability to different data types. The procedure ensures metric consistency and repeatability, even when dealing with historical images characterized by heterogeneous quality and unknown acquisition geometry.

The proposed approach enables the integration of historical iconographic, photographic, or cartographic information into modern three-dimensional models, making these data comparable with current observations and suitable for quantitative analysis.

Keywords Immagini; Modelli; Trasformazione | Images; Models; Texturing; Transformation

Introduzione

L'uso delle fotografie come sorgenti di informazione spaziale è oggi ampiamente consolidato grazie alle tecniche di fotogrammetria multi-immagine basate su Structure from Motion (SfM) e visione computazionale [Snavely et al., 2006; Remondino et al., 2014; Brunier et al., 2016; Eltner et al., 2016]. In tali contesti, immagini di natura bidimensionale possono essere

interpretate come misure, contribuendo alla ricostruzione di geometrie tridimensionali e alla descrizione metrica di oggetti, superfici e strutture. La fotografia assume quindi il ruolo di dato spaziale quantitativo. All'interno di tali *workflow*, la texturizzazione rappresenta la fase in cui l'informazione radiometrica viene trasferita sulla geometria ricostruita.

Nei processi fotogrammetrici integrati, posa delle camere, ricostruzione geometrica e *mapping* radiometrico costituiscono un sistema unitario; analogamente, anche in altri rilievi tridimensionali basati su sensori di acquisizione geometrica, l'impiego di camere calibrate consente una proiezione del colore coerente con i parametri di acquisizione [Böhler e Marbs, 2004; Guidi et al., 2009].

In questi casi, la *texture* assume il ruolo di dato scientifico, metricamente consistente e utilizzabile per analisi dei materiali, studi di degrado o modellazione di fenomeni fisici [Dostal et al., 2018; Teza e Pesci, 2019]. Accanto a tali approcci si è sviluppato un ampio insieme di pratiche orientate prevalentemente alla visualizzazione e alla fruizione immersiva. Tecniche di *UV unwrapping*, *projection mapping* semi-automatico e pipeline di derivazione grafica consentono la generazione rapida di modelli virtuali e ricostruzioni storiche [Haala & Kada, 2010; Champion, 2019]. In queste applicazioni, la texture svolge principalmente una funzione comunicativa: il risultato è spesso efficace dal punto di vista visivo, ma raramente accompagnato da una valutazione metrica o da un'analisi dell'errore, risultando pertanto poco idoneo a impieghi analitici.

Tra la *texture* intesa come misura scientifica, ovvero informazione radiometrica trasferibile e verificabile, e la *texture* intesa come superficie evocativa o puramente illustrativa, permane pertanto un evidente vuoto metodologico. Per molte applicazioni in ambito storico, architettonico e dei beni culturali risulta potenzialmente utile sfruttare il contenuto informativo di immagini esistenti, spesso reperite a posteriori, che non sono state acquisite con finalità di texturizzazione o modellazione tridimensionale. In numerosi casi, tali immagini potrebbero essere integrate con modelli 3D moderni degli oggetti di interesse, ottenuti mediante tecniche di rilievo tridimensionale. Tuttavia, esse sono generalmente prive di metadati di acquisizione, di una geometria nota o di ridondanza multi-vista, rendendo necessario un approccio specifico, controllato e verificabile per il loro impiego a fini metrici e analitici.

In questo scenario, gli strumenti oggi disponibili, sebbene di elevata qualità e basati su strategie di analisi avanzate, risultano spesso poco adeguati allo scopo. I principali software commerciali per il *texturing* e la modellazione tridimensionale, ad esempio, presuppongono *dataset* multi-immagine, geometrie già orientate o parametri di acquisizione noti; altre soluzioni automatiche tendono invece a ignorare l'origine singolare dell'immagine, assimilando il problema a un caso standard di *mapping* radiometrico ([1], [2], [3]). Ne consegue che l'immagine viene frequentemente utilizzata come semplice rivestimento visivo, senza un controllo esplicito sulla trasformazione geometrica applicata, sull'errore introdotto o sul reale contenuto informativo trasferito alla nuvola di punti.

In assenza di metadati di acquisizione o di condizioni di calibrazione ottimali, la texturizzazione può essere formulata come un problema di corrispondenza tra rappresentazioni eterogenee: un'immagine bidimensionale e una geometria tridimensionale. L'allineamento guidato tramite punti omologhi 2D-3D, combinato con opportune trasformazioni, rappresenta un'alternativa praticabile alla ricostruzione multi-vista. In tali condizioni, l'obiettivo non è la ricostruzione del punto di vista originale, spesso indeterminabile, ma la definizione di un legame coerente tra pixel e coordinate spaziali. La *texture* diventa così un dato interrogabile, mentre la distribuzione dei residui di proiezione assume un ruolo diagnostico, consentendo una valutazione locale della coerenza tra informazione iconografica e geometria contemporanea [Grün, 1985].

In questo contesto è stata sviluppata una pipeline operativa in ambiente MATLAB, originariamente derivata da strumenti per la termografia 3D [Teza et al., 2019] e successivamente adattata alla proiezione di materiali storici [Pesci et al., 2026]. Il metodo riformula il problema della texturizzazione di immagini storiche come una corrispondenza tra rappresentazioni 2D,

mediata da una proiezione sintetica generata a partire dalla nuvola di punti tridimensionale. In pratica, viene creata un'immagine prospettica derivata dal modello 3D, i punti omologhi vengono individuati tra questa immagine e l'immagine storica, e la trasformazione stimata permette di riallineare l'immagine originale. Successivamente, l'informazione radiometrica può essere trasferita in modo controllato e reversibile sulla geometria 3D. La procedura, guidata dall'utente, comprende: (i) l'orientamento della nuvola di punti secondo una vista definita dall'operatore; (ii) la generazione della proiezione 2D sintetica; (iii) l'allineamento dell'immagine storica mediante punti omologhi; (iv) il trasferimento dell'informazione radiometrica sulla geometria tridimensionale.

L'ambiente operativo è lineare, autosufficiente e organizzato in blocchi sequenziali. Sono inoltre integrati due moduli accessori: il primo consente di valutare la coerenza tra la risoluzione dell'immagine e la scala del modello 3D; il secondo analizza le immagini prospettiche generate, utilizzando la distanza dei punti dal piano di vista come indicatore del contributo geometrico dei pixel.

L'obiettivo non è la produzione di risultati visivamente ottimizzati, ma la definizione di criteri operativi per una texturizzazione scientificamente interpretabile, in cui la texture rispetti il legame tra geometria, punto di vista e risoluzione, consentendo alle immagini storiche di essere utilizzate come dati fisicamente coerenti nello spazio tridimensionale.

Rispetto alle implementazioni precedenti, l'attuale software rappresenta una riorganizzazione sostanziale del workflow, specificamente orientata al trattamento di immagini storiche. In particolare, la struttura originaria, concepita per dati termografici e condizioni radiometriche controllate, è stata semplificata e riconfigurata in una pipeline dedicata, nella quale le componenti non rilevanti per il dominio storico sono state rimosse e le fasi operative sono state rese sequenziali e più trasparenti. Tale revisione non si limita a un adattamento funzionale, ma introduce una ridefinizione dell'architettura operativa in funzione delle caratteristiche geometriche e qualitative dei dati storici.

1. Metodologia

L'obiettivo della metodologia è trasferire in modo controllato informazioni cromatiche e visive da immagini storiche a modelli tridimensionali contemporanei, o più in generale a nuvole di punti ottenute tramite tecniche di rilievo quali fotogrammetria, laser scanning o altre modalità di acquisizione 3D. La procedura integra dati eterogenei, combinando la coerenza metrica del rilievo moderno con l'informazione radiometrica delle immagini storiche, rendendo i dati utilizzabili per analisi spaziali, confronti temporali e studi applicativi in ambito storico, architettonico e dei beni culturali. È importante sottolineare che la riproducibilità del metodo si riferisce alle componenti computazionali della pipeline, mentre alcune fasi operative, come la selezione dei punti omologhi e la definizione della vista, rimangono legate a scelte guidate dall'utente. In questo senso, il processo è ripetibile a parità di condizioni operative, ma non completamente automatico *end-to-end*.

Il processo inizia con l'orientamento del modello tridimensionale secondo una vista definita dall'operatore, approssimativamente coerente con il punto di osservazione dell'immagine storica. Da questa configurazione viene generata un'immagine prospettica sintetica, in cui ciascun pixel corrisponde a una coordinata della nuvola di punti. L'immagine sintetica e la fotografia storica costituiscono così due rappresentazioni confrontabili dello stesso oggetto o scena.

Su entrambe le immagini vengono selezionati punti omologhi, distribuiti il più uniformemente possibile, utilizzati per stimare una trasformazione geometrica bidimensionale. Nel caso specifico, una trasformazione affine consente di allineare l'immagine storica alla vista sintetica derivata dal modello 3D, senza modificare direttamente la geometria tridimensionale. Una

volta ottenuto un allineamento soddisfacente, i valori cromatici dell'immagine storica vengono trasferiti sulla nuvola di punti mediante retro-proiezione, generando un modello tridimensionale texturizzato in cui l'informazione radiometrica è associata in modo esplicito e verificabile alle coordinate spaziali del rilievo.

Come in ogni procedura di proiezione tra domini di diversa dimensione, sono presenti fonti di errore. La proiezione da 3D a 2D può introdurre variazioni nella risoluzione spaziale e radiometrica, mentre la qualità dell'allineamento dipende dalla distribuzione dei punti omologhi, dalla qualità dell'immagine storica e dalla densità della nuvola di punti. Tali errori possono essere quantificati tramite l'analisi dei residui e il calcolo del RMS delle coordinate dei punti corrispondenti, e mitigati mediante scelte operative quali l'aumento della densità del modello, una selezione accurata dei punti omologhi, la pre-elaborazione delle immagini storiche e l'uso di interpolazioni appropriate nella retro-proiezione.

1.1 Pipeline

Sulla base di queste considerazioni, le fasi operative del *workflow* possono essere descritte come segue nella cascata a punti.

1. Caricamento e orientamento della nuvola di punti
 - a. Importazione della nuvola 3D (x, y, z, colore) da rilievi recenti.
 - b. Navigazione e posizionamento della vista per riprodurre il punto di osservazione della fotografia storica.
 - c. Ricostruzione della linea di vista originaria, base per la proiezione prospettica.
2. Proiezione prospettica e generazione dell'immagine sintetica
 - d. Proiezione della nuvola su un piano immagine per ottenere un *raster* 2D.
 - e. Ogni pixel mantiene il legame inverso con le coordinate 3D.
 - f. Generazione di una "fotografia moderna sintetica" coerente con la geometria 3D.
3. Identificazione dei punti omologhi
 - i. Selezione di coppie di punti corrispondenti sull'immagine sintetica e sulla fotografia storica.
 - ii. Copertura uniforme dell'intera area per definire la trasformazione geometrica.
 - iii. Calcolo della trasformazione affine
 - iv. Stima ai minimi quadrati dei parametri della trasformazione affine bidimensionale.
 - v. Correzione di scala, rotazione, traslazione e piccoli errori prospettici.
4. Trasferimento cromatico e *back-projection*
 - i. Campionamento dell'immagine storica per trasferire i valori RGB sull'immagine sintetica.
 - ii. Riattribuzione dei valori cromatici alla nuvola di punti originale tramite relazione inversa pixel- punto 3D.
 - iii. Produzione di un modello 3D metrico con cromie storiche.

Valutazione di fattori critici e strategie di mitigazione:

- a. Fonti di errore e perdita di accuratezza
 - Perdita di risoluzione nel passaggio nuvola - immagine
 - Multipli punti 3D proiettati sullo stesso pixel; effetti di quantizzazione e aliasing.
 - Aree occluse prive di informazione cromatica.
 - Interpolazione durante la *back-projection* può introdurre perdita di nitidezza o discontinuità cromatiche.
- b. Errori nei punti omologhi
 - Imprecisione nella selezione manuale.

- Variazioni morfologiche tra epoca storica e rilievo moderno.
- Deformazioni ottiche e bassa nitidezza dell'immagine d'epoca.
- Distribuzione non omogenea dei punti, con errori locali.
- RMS sulle coordinate dei punti omologhi quantifica l'errore residuo.

c. Strategie di mitigazione

- Aumento della densità della nuvola e uso di rilievi ad alta risoluzione.
- Scelta accurata e distribuita dei punti omologhi, evitando aree con elementi scomparsi o deformati.
- Filtraggio e miglioramento delle immagini storiche per nitidezza e contrasto.
- Interpolazioni bilineari o bicubiche nella *back-projection* per evitare artefatti cromatici.
- Ripetizione della selezione dei punti o ricalibrazione della trasformazione affine se necessario.

Prima di procedere, vengono introdotti alcuni sottoparagrafi di chiarimento, volti a esplicitare i concetti geometrici alla base del metodo, in particolare la trasformazione affine bidimensionale e il modello di proiezione centrale. Questi chiarimenti facilitano la comprensione delle fasi operative descritte nella pipeline. Nel presente lavoro, l'allineamento avviene tra una vista generata da una geometria 3D metrica e una fotografia storica con prospettiva simile. In questo regime di piccole variazioni geometriche, l'impiego di una trasformazione affine risulta metodologicamente adeguato e coerente.

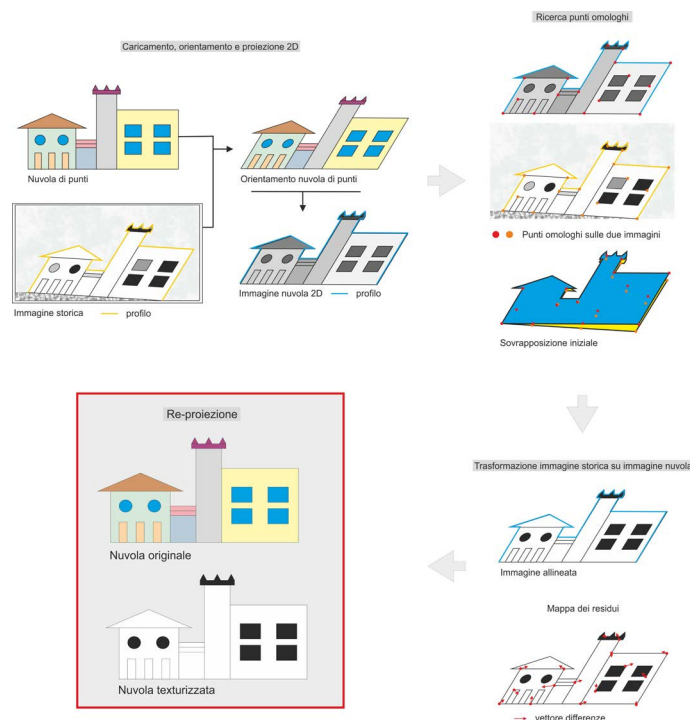


Figura 1 Schema operativo principale della pipeline: a partire dall'importazione della nuvola di punti e dell'immagine storica, viene definita la vista di lavoro e generata l'immagine 2D; la selezione di punti omologhi consente la stima della trasformazione affine e la valutazione dei residui, seguita dalla reprojezione dell'immagine storica e dal **texturing** finale della nuvola di punti.

Figure 1 Main operational scheme of the pipeline: starting from the import of the point cloud and the historical image, the working view is defined and a 2D image is generated; the selection of homologous points allows the estimation of the affine transformation and the evaluation of residuals, followed by the reprojection of the historical image and the final texturing of the point cloud.

1.2 Trasformazione affine

La procedura di co-registrazione tra due immagini, una ottenuta dalla proiezione di una nuvola di punti 3D moderna e l'altra costituita da una fotografia storica, si basa sulla stima di una trasformazione geometrica in grado di minimizzare le discrepanze tra coppie di punti omologhi selezionati nelle due immagini. In questa sezione sono illustrate le basi matematiche che regolano il processo di *texturing* adottato nel presente lavoro.

Le due immagini rappresentano lo stesso soggetto osservato da punti di vista simili: la prima è una rappresentazione sintetica derivata dalla geometria tridimensionale, la seconda è una ripresa storica acquisita con sistemi fotografici d'epoca. La procedura prevede l'individuazione di corrispondenze geometriche tra elementi riconoscibili in entrambe le immagini e il calcolo di una trasformazione che consenta il loro allineamento con la massima accuratezza possibile. Tale allineamento costituisce il presupposto per il trasferimento delle informazioni cromatiche storiche sul modello tridimensionale contemporaneo. Dal punto di vista matematico, il metodo si fonda sull'applicazione di una trasformazione affine bidimensionale, sulla stima dei suoi parametri mediante un approccio ai minimi quadrati e su una scelta accurata e ben distribuita dei punti omologhi sull'intera scena. L'attenzione alla geometria delle corrispondenze e alla stabilità numerica del sistema consente di ottenere un allineamento robusto, limitando gli errori geometrici e preservando la coerenza cromatica nella nuvola di punti texturizzata. La trasformazione affine rappresenta un modello lineare in grado di descrivere rotazione, traslazione, scale (anche anisotrope) e *shear* tra due sistemi di coordinate 2D.

Data una coppia di punti corrispondenti (x,y) nell'immagine di riferimento e (x',y') nell'immagine da trasformare, la relazione affine può essere espressa in forma matriciale come:

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & tx \\ a_{21} & a_{22} & ty \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

dove i coefficienti a_{11} , a_{12} , a_{21} , a_{22} descrivono la parte lineare (rotazione, scala, *shear*), mentre tx e ty rappresentano la traslazione. La trasformazione affine mantiene la parallelità delle linee e la proporzionalità delle distanze lungo direzioni corrispondenti, ma non garantisce la conservazione di angoli o lunghezze, risultando adeguata per deformazioni globali tra viste simili ma non identiche. Esplicitando le componenti si ottiene il sistema:

$$\begin{cases} x' = a_{11}x + a_{12}y + tx \\ y' = a_{21}x + a_{22}y + ty \end{cases}$$

È evidente che dalla matrice generale della trasformazione affine si possono ricavare una serie di trasformazioni elementari, ottenute imponendo specifiche condizioni sui parametri. Queste trasformazioni di base rappresentano casi particolari del modello affine e consentono di interpretare in modo intuitivo l'effetto geometrico dei singoli coefficienti sulla posizione e sulla forma dell'immagine (Figura 2).

La traslazione descrive uno spostamento rigido dell'immagine nel piano, senza introdurre alcuna variazione di orientamento o di scala. Essa è fondamentale per compensare differenze di origine dei sistemi di riferimento o piccoli disallineamenti globali tra le due immagini.

1. Traslazione:

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & tx \\ 0 & 1 & ty \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

Le trasformazioni di scala, isotrope o anisotrope, modificano le dimensioni dell'immagine mantenendo la parallelità delle rette. La scala isotropa preserva anche gli angoli ed è tipica di variazioni uniformi di ingrandimento o riduzione, mentre la scala anisotropa consente di modellare differenze di rapporto tra le due direzioni principali, frequentemente riscontrabili nel confronto tra immagini acquisite con sistemi e supporti differenti.

2. Scala isotropa:

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} s & 0 & 0 \\ 0 & s & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

3. Scala anisotropa:

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} sx & 0 & 0 \\ 0 & sy & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

Nel modello affine, tale comportamento è rappresentato dalla tipica condizione $a_{11} \neq a_{22}$ (qui s_x e s_y), che introduce una scala differenziale lungo le due direzioni principali dell'immagine, consentendo di descrivere deformazioni non uniformi tipiche di acquisizioni non perfettamente isotrope. Un'interpretazione intuitiva di tale comportamento può essere ottenuta facendo riferimento ai comuni strumenti di grafica digitale. In molti software di editing, infatti, le operazioni di trasformazione includono modalità di scala e rotazione che consentono di modificare manualmente la geometria di un'immagine. In particolare, la scala anisotropa corrisponde a una deformazione ottenuta agendo separatamente sui due assi principali, alterando il rapporto tra larghezza e altezza dell'immagine, mentre la rotazione consente di cambiare l'orientamento globale senza modificarne le proporzioni. La combinazione di queste operazioni produce effetti geometrici analoghi a quelli descritti dal modello affine, rendendo immediata la connessione tra formulazione matematica e strumenti operativi comunemente utilizzati.

La rotazione rappresenta un cambiamento di orientamento rigido dell'immagine, conservando lunghezze e angoli. Nel contesto della co-registrazione tra immagini storiche e immagini sintetiche, essa permette di compensare differenze di orientamento dovute alla posa della camera o all'orientamento del modello tridimensionale rispetto al piano immagine.

4. Rotazione:

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} \cos \theta & -\sin \theta & 0 \\ \sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

Le trasformazioni di taglio (*shear*) introducono una deformazione non rigida che mantiene il parallelismo delle rette ma altera gli angoli. Sebbene non rappresentino una trasformazione fisicamente rigida, esse risultano particolarmente utili per assorbire piccole discrepanze prospettiche residue, deformazioni del supporto fotografico o imprecisioni locali non modellabili tramite sola rotazione e scala.

5. Taglio/Shear orizzontale:

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & k & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

6. Taglio/Shear verticale:

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ k & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

Nel loro insieme, queste trasformazioni elementari mostrano come il modello affine sia in grado di descrivere una combinazione controllata di effetti geometrici semplici ma significativi. Quando applicate simultaneamente, attraverso la stima dei parametri affini mediante punti omologhi, consentono di ottenere un allineamento efficace tra immagini di viste prossime, operando nel regime delle piccole variazioni geometriche. Questa proprietà rende la trasformazione affine particolarmente adatta alla co-registrazione tra una vista sintetica derivata da una nuvola di punti 3D e una fotografia storica, in assenza di informazioni complete sul sistema di acquisizione originale.

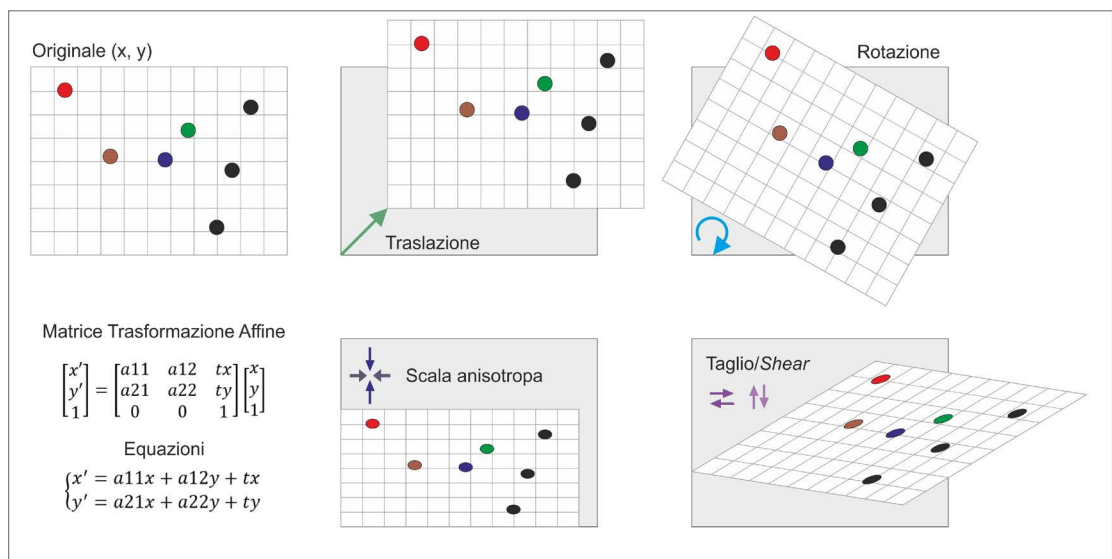


Figura 2 Trasformazione affine bidimensionale tra immagini corrispondenti. Rappresentazione concettuale della trasformazione affine utilizzata per l'allineamento tra l'immagine sintetica derivata dal modello 3D e la fotografia storica. La trasformazione combina traslazione, rotazione, scale (anche anisotrope) e taglio (*shear*), consentendo di allineare viste inizialmente simili nel regime delle piccole variazioni. I punti omologhi selezionati nelle due immagini costituiscono i vincoli geometrici per la stima dei parametri mediante il metodo dei minimi quadrati.

Figure 2 Two-dimensional affine transformation between corresponding images. Conceptual representation of the affine transformation used to align the synthetic image derived from the 3D model with the historical photograph. The transformation combines translation, rotation, scaling (including anisotropic scaling), and shear, enabling the alignment of initially similar views within the small-variation regime. Homologous points selected in the two images provide the geometric constraints for parameter estimation through a least-squares approach.

1.2.1 Calcolo dell'errore

Per stimare i sei parametri della trasformazione (a_{11} , a_{12} , a_{21} , a_{22} , tx , ty) sono necessarie almeno tre coppie di punti omologhi non allineati. In forma matriciale, il problema può essere scritto come:

$$Ap = b$$

dove:

$$A = \begin{bmatrix} x1 & y1 & 0 & 0 & 1 & 0 \\ 0 & 0 & x1 & y1 & 0 & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ xn & yn & 0 & 0 & 1 & 0 \\ 0 & 0 & xn & yn & 0 & 1 \end{bmatrix}; p = \begin{bmatrix} a11 \\ a12 \\ a21 \\ a22 \\ tx \\ ty \end{bmatrix}; b = \begin{bmatrix} x1' \\ y1' \\ \vdots \\ xn' \\ yn' \end{bmatrix}$$

Con $n \geq 3$ coppie di punti, la stima viene eseguita mediante minimi quadrati ottenendo la trasformazione che minimizza la somma dei quadrati delle differenze tra le coordinate osservate e trasformate dei punti omologhi.

$$p = (A^T A)^{-1} A^T b$$

La precisione e la robustezza della trasformazione affine dipendono fortemente dalla geometria dei punti omologhi.

Punti troppo ravvicinati o disposti lungo una stessa linea producono un sistema mal condizionato, amplificando eventuali errori di misura o di selezione. È quindi fondamentale che i punti siano distribuiti in modo ampio e omogeneo su tutta l'immagine, così da rappresentare in maniera equilibrata l'intero dominio geometrico.

Questa condizione garantisce:

- buona determinazione dei parametri lineari e di traslazione;
- stabilità numerica del sistema $p = (A^T A)^{-1} A^T b$
- residui locali non sistematici, indice di deformazioni non modellate ma distribuite in modo casuale.

Dopo l'applicazione della trasformazione, la differenza tra le coordinate trasformate e quelle dei punti target costituisce il residuo geometrico, calcolato come:

$$ri = \sqrt{(xi' - \hat{x}i')^2 + (yi' - \hat{y}i')^2}$$

dove: $(\hat{x}i', \hat{y}i')$ sono le coordinate trasformate.

La distribuzione dei residui, visualizzata con mappe cromatiche, permette di identificare:

- zone a errore casuale: trasformazione coerente;
- zone a errore sistematico: deformazione locale o prospettiva non descritta dal modello affine.

L'uso della trasformazione affine in contesti di viste vicine garantisce un allineamento semplice e controllabile, basato sulla geometria delle corrispondenze. Esistono naturalmente approcci più sofisticati, ad esempio modellazioni delle caratteristiche ottiche delle fotocamere storiche (focale, grandangolo, deformazioni radiali o prospettiche legate al formato della lastra). Tuttavia, tali informazioni sono spesso assenti e l'adozione di modelli complessi, in assenza di dati affidabili, rischierebbe di introdurre errori aggiuntivi e speculazioni ingiustificate. Per questo studio si è preferito operare con un modello affine, adeguato al regime delle piccole variazioni geometriche.

Il modello affine, proprio per la sua natura lineare, risulta un'approssimazione locale della trasformazione prospettica completa. Esso non è quindi adeguato in presenza di forti effetti prospettici, variazioni significative della profondità della scena lungo la direzione di osservazione o quando la scena non può essere approssimata come localmente planare. In tali condizioni, le distorsioni introdotte dalla proiezione centrale non possono essere correttamente descritte da

una trasformazione affine e sarebbe necessario ricorrere a modelli proiettivi più generali. Nel presente lavoro tali condizioni non si verificano, in quanto le porzioni analizzate delle facciate possono essere considerate, nel dominio di osservazione adottato, sufficientemente vicine a configurazioni planari e caratterizzate da variazioni geometriche contenute.

In scenari più complessi, caratterizzati da non planarità locale o deformazioni geometriche significative, una possibile estensione dell'approccio consiste nella suddivisione dell'immagine in regioni più piccole, all'interno delle quali l'ipotesi di planarità risulta più valida. In tali condizioni, la trasformazione affine può essere applicata localmente a ciascun blocco, riducendo l'impatto delle distorsioni prospettiche globali e migliorando la stabilità delle corrispondenze. Questo approccio consente di mantenere un modello lineare e controllabile anche in configurazioni geometricamente più articolate, seppur a fronte di una maggiore complessità operativa.

1.3 Proiezione da 3D a piano immagine

Per trasferire correttamente l'informazione radiometrica dalla nuvola di punti 3D alla fotografia storica, è necessario definire una geometria di proiezione coerente, che stabilisca come ogni punto nello spazio venga rappresentato sul piano bidimensionale dell'immagine (Figura 3).

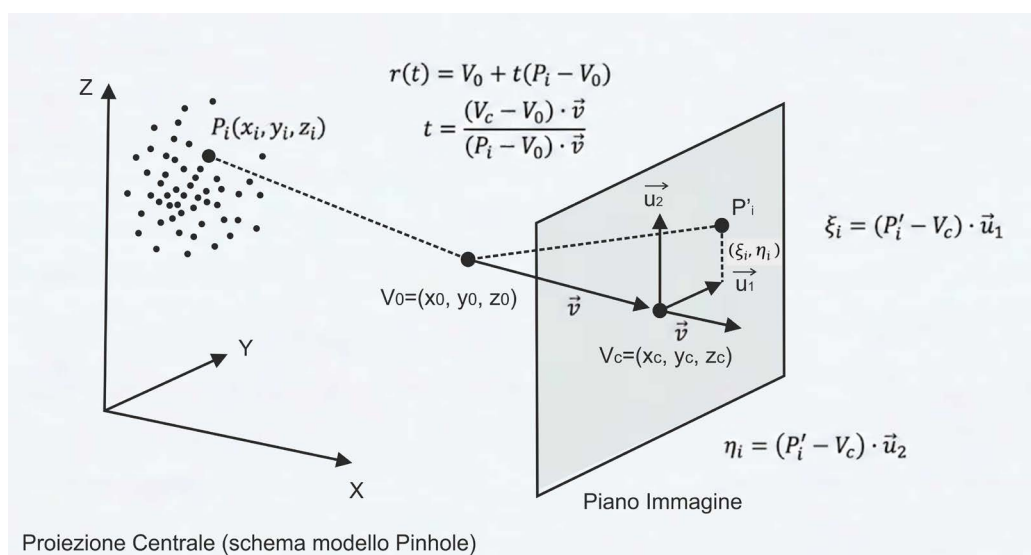


Figura 3 Schema della proiezione centrale utilizzata per generare l'immagine sintetica a partire dalla nuvola di punti 3D. Ogni punto P_i viene proiettato sul piano immagine tramite un raggio passante per il centro di proiezione V_0 ; l'intersezione con il piano, definito dal centro V_c e dalla direzione di vista v , determina le coordinate bidimensionali (ξ_i, η_i) . Il modello fornisce il legame geometrico tra spazio tridimensionale e rappresentazione 2D, base per il successivo allineamento con la fotografia storica.

Figure 3 Scheme of the central projection used to generate the synthetic image from the 3D point cloud. Each point P_i is projected onto the image plane through a ray passing through the projection center V_0 ; the intersection with the plane, defined by the center V_c and the viewing direction v , determines the two-dimensional coordinates (ξ_i, η_i) . The model provides the geometric link between three-dimensional space and its 2D representation, forming the basis for the subsequent alignment with the historical photograph.

Sia $P_i = (x_i, y_i, z_i)$ un punto della nuvola di punti, espresso in un sistema cartesiano relativo al rilievo o alla ricostruzione fotogrammetrica. La camera virtuale, che simula il punto di vista della fotografia storica, è definita da:

- il centro di proiezione $V0 = (x0, y0, z0)$, corrispondente all'occhio della camera;
- il piano immagine P con centro $Vc = (xc, yc, zc)$, perpendicolare alla linea di vista principale $v = (Vc - V0) / \|Vc - V0\|$.

La proiezione centrale collega ogni punto Pi al centro di proiezione $V0$ tramite un raggio $r(t) = V0 + t(Pi - V0)$. L'intersezione di questo raggio con il piano immagine determina la posizione (ξ_i, η_i) proiettata sulla fotografia sintetica, con: $t = [(Vc - V0) \cdot v] / [(Pi - V0) \cdot v]$. Per ottenere le coordinate effettive sul piano immagine, si definiscono due vettori ortogonali $u1$ e $u2$ sul piano, e si calcolano $\xi_i = (Pi' - Vc) \cdot u1$ e $\eta_i = (Pi' - Vc) \cdot u2$ dove Pi' è l'intersezione del raggio con il piano. Le coppie (ξ_i, η_i) costituiscono la proiezione bidimensionale della nuvola di punti e rappresentano la base geometrica per l'allineamento con la fotografia storica.

2. Introduzione al software

Il workflow di *texturing* è centrato su *Historica*, cioè il codice principale sviluppato in ambiente MATLAB, strutturato come un sistema a blocchi indipendenti eseguiti in cascata. Ciascun blocco può essere lanciato singolarmente o come parte dell'intera *pipeline*, garantendo flessibilità operativa e modularità. Il workflow combina procedure automatiche e scelte guidate dall'utente. Le operazioni di calcolo, trasformazione geometrica, proiezione e trasferimento radiometrico sono completamente automatizzate una volta definiti i parametri iniziali. L'intervento dell'utente è invece limitato a fasi specifiche di interpretazione e controllo, in particolare alla definizione della vista della nuvola di punti, alla selezione dei punti omologhi tra immagine sintetica e immagine storica e alla valutazione qualitativa dei risultati intermedi. Questa separazione consente di mantenere un controllo esplicito sulle fasi critiche del processo, pur garantendo la ripetibilità delle operazioni computazionali.

Historica esegue la registrazione della fotografia storica sulla nuvola di punti e il trasferimento dell'informazione cromatica, producendo il modello 3D texturizzato in maniera controllata e riproducibile.

A supporto di *Historica* sono disponibili due moduli accessori: *PreCloud*, che analizza la nuvola di punti e permette di definire la vista e i parametri della camera virtuale, e *Prelmg*, che valuta il contenuto informativo dell'immagine storica e stima la risoluzione utile per la proiezione. Questi strumenti forniscono indicazioni preliminari sulle scelte operative da adottare in *Historica*, senza intervenire direttamente sul *texturing* finale, e costituiscono supporti utili ma non indispensabili. Va anche sottolineato che entrambi i moduli accessori operano esclusivamente in fase preliminare e non alterano direttamente il processo di *texturing*, ma contribuiscono alla definizione di condizioni operative più riproducibili e controllate.

2.1 PreCloud: codice accessorio

Il modulo *PreCloud* costituisce una fase preliminare di calibrazione visiva della nuvola di punti tridimensionale, finalizzata alla definizione coerente del punto di vista da cui sarà simulata la fotografia storica. In ingresso, l'utente fornisce una nuvola di punti in formato testuale, contenente le coordinate spaziali (X, Y, Z) ed eventualmente un valore associato V, di colore. Il modulo mantiene inalterata la geometria originale della nuvola, applicando eventuali normalizzazioni esclusivamente a fini di visualizzazione (resa neutra in scala di grigi), in modo da facilitare la lettura spaziale senza introdurre interpretazioni cromatiche.

Il nucleo funzionale consiste in un'interfaccia grafica 3D interattiva che permette all'utente di ruotare, zoomare e navigare nella nuvola fino a ricreare una vista coerente con quella della fotografia

storica. Per ottimizzare i tempi di lavoro, PreCloud può operare su una nuvola sottocampionata, poiché l'obiettivo non è visualizzare ogni dettaglio, ma definire rapidamente la posizione della camera e l'orientamento desiderato. In questo modo, anche su sistemi non particolarmente potenti, l'utente può completare la calibrazione visiva senza rallentamenti significativi.

Una volta confermata la vista, il sistema acquisisce i parametri della camera virtuale (posizione, direzione di vista, orientamento verticale e angolo di campo) che definiscono un sistema prospettico coerente con l'osservazione desiderata. Tali parametri possono essere salvati e utilizzati direttamente in Historica, evitando di dover roto-traslare nuovamente la nuvola completa durante la fase di *texturing*.

I parametri estratti dal modulo PreCloud costituiscono quindi un insieme deterministico di valori (matrice di vista, posizione della camera e orientamento) che possono essere direttamente riutilizzati nel modulo Historica, garantendo la riproducibilità della configurazione geometrica tra sessioni diverse.

Un'ulteriore funzionalità opzionale del modulo visualizza la distanza dei punti dalla camera tramite una mappa cromatica, permettendo di identificare eventuali anomalie macroscopiche, come cluster isolati o problemi di scala. Questa verifica è pensata come controllo qualitativo sulla porzione di nuvola effettivamente utilizzata, senza la necessità di analizzare in dettaglio ogni singolo punto. Nell'esempio mostrato in Figura 4, la vista corrisponde direttamente a quella impostata dall'utente nell'interfaccia MATLAB, senza alcuna rotazione o spostamento aggiuntivo della nuvola. La mappa delle distanze riflette quindi intuitivamente la geometria della scena: le parti più alte appaiono più vicine alla camera, mentre le zone più basse risultano più lontane, confermando la correttezza del posizionamento e la coerenza prospettica della visualizzazione. L'analisi della mappa delle distanze può inoltre essere utilizzata per individuare soglie operative indicative, ad esempio per escludere regioni della nuvola caratterizzate da distanze eccessive o geometria non affidabile rispetto al punto di vista selezionato. Un'analisi attenta della figura consente di rilevare anche valori di distanza elevati, il cui significato sarà discusso nelle sezioni successive. Altri dettagli implementativi e funzioni accessorie del modulo PreCloud sono riportati nell'Appendice A, dove il codice completo è reso disponibile per consultazione, utilizzo o modifica.

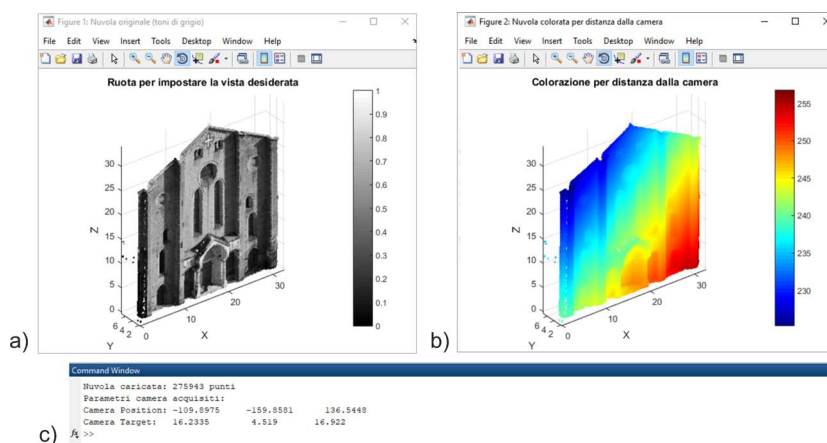


Figura 4 Output del modulo accessorio per la valutazione della distanza dei punti dalla camera. **a)** Nuvola di punti visualizzata nella vista selezionata; **b)** mappa dei colori che rappresenta la distanza di ciascun punto dal centro della camera; **c)** interfaccia MATLAB che mostra la posizione spaziale della camera rispetto alla nuvola di punti, utile per la verifica della geometria di proiezione.

Figure 4 Output of the auxiliary module for evaluating point-to-camera distance. **a)** Point cloud displayed in the selected view; **b)** color-coded map representing the distance of each point from the camera center; **c)** MATLAB interface showing the spatial position of the camera with respect to the point cloud, useful for validating the projection geometry.

2.2 PreImg: script accessorio

Il modulo *PreImg* esegue una valutazione preliminare della qualità informativa di un'immagine storica prima della sua proiezione sulla nuvola di punti. In ingresso viene fornito un file *raster*, preferibilmente non compresso o con perdita minima. Lo script converte internamente l'immagine in RGB a 8 bit, garantendo la coerenza con la pipeline di *texturing*. L'analisi inizia con la valutazione della struttura cromatica dell'immagine: viene calcolato il numero di colori unici e la loro distribuzione locale, allo scopo di individuare eventuali criticità dovute a compressione eccessiva, posterizzazione o reale carenza di informazione radiometrica. Successivamente viene eseguito un test di degradazione controllata, in cui l'immagine viene progressivamente ridotta di risoluzione (*downsampling*) e poi ricostruita tramite interpolazione bicubica (*upsampling*). La differenza rispetto all'immagine originale viene quantificata tramite l'errore quadratico medio (MSE) e l'indice di similarità strutturale (SSIM).

Il MSE quantifica la differenza media tra i valori di pixel dell'immagine originale e quelli dell'immagine degradate e ricostruita, fornendo una misura semplice di distorsione globale. L'SSIM, invece, valuta la similarità percepita tra le due immagini considerando luminanza, contrasto e struttura locale, risultando più coerente con la percezione visiva della qualità. L'uso combinato di MSE e SSIM permette di identificare eventuali perdita di informazioni dovute a compressione, posterizzazione o riduzione della risoluzione, assicurando che l'immagine impiegata per il *texturing* contenga dettagli radiometrici sufficienti [Gonzalez e Wood, 2018; Wang et al., 2004].

L'andamento di queste metriche in funzione del fattore di scala permette di individuare la risoluzione effettivamente utile per il *texturing*: valori bassi indicano immagini con pochi dettagli reali, mentre valori elevati indicano la presenza di informazione fine significativa da trasferire sulla nuvola di punti. I risultati vengono restituiti sotto forma di grafici e valori numerici e non comportano modifiche dirette all'immagine originale. Essi forniscono indicazioni operative oggettive per confermarne l'uso, suggerirne un ridimensionamento, una ottimizzazione o, eventualmente, scartarla. In pratica, l'utente può valutare preliminarmente la qualità dell'immagine e definire parametri di lavoro ottimali per il modulo principale *Historica*, evitando passaggi inutili o ridondanti. In questo modo il modulo fornisce un supporto decisionale quantitativo per stabilire se l'immagine sia idonea al *texturing*, se debba essere sottoposta a ridimensionamento o se presenti livelli di informazione insufficienti per un utilizzo affidabile nella *pipeline* principale. Un esempio applicativo è mostrato nella Figura 5, relativa a una fotografia storica della facciata della Basilica di San Francesco a Bologna, danneggiata dai bombardamenti del 1943. L'immagine, nominalmente a 350 dpi, mostra come la dimensione apparente del file non corrisponda al dettaglio effettivo utile: molte aree, come il cielo o il terreno, contengono pixel ridondanti o micro-variazioni prive di significato.

L'istogramma dell'indice cromatico combinato evidenzia pochi intervalli dominanti e picchi localizzati, coerenti con la scansione di una fotografia originariamente in scala di grigi, mentre la mappa dei pixel con vicini identici consente di visualizzare immediatamente la ridondanza cromatica. L'andamento dell'entropia media, che cresce rapidamente da circa 2.7 bit per blocchi piccoli fino a valori prossimi a 5.1 bit per blocchi di 32×32 pixel, conferma che la parte significativa dell'informazione è limitata e saturata su porzioni relativamente ampie dell'immagine. In altre parole in questo caso, pur essendo nominalmente grande, l'immagine contiene un dettaglio effettivo inferiore a quanto suggerisce la risoluzione nominale, risultando parzialmente sovracampionata rispetto all'informazione realmente disponibile. In termini operativi, le informazioni fornite da *PreImg* non sono meramente descrittive, ma influenzano direttamente le scelte esecutive del *workflow* principale. In particolare, i risultati del test di degradazione e le metriche MSE e SSIM permettono di definire in modo informato la risoluzione effettiva

da utilizzare nella proiezione e di valutare l'opportunità di includere o escludere specifiche immagini nel processo di *texturing*. In questo senso, Prelmg agisce come modulo decisionale preliminare a supporto della configurazione di Historica. Altri dettagli implementativi e funzioni accessorie del modulo Prelmg sono riportati nell'Appendice B, dove il codice completo è reso disponibile per consultazione, utilizzo o modifica.

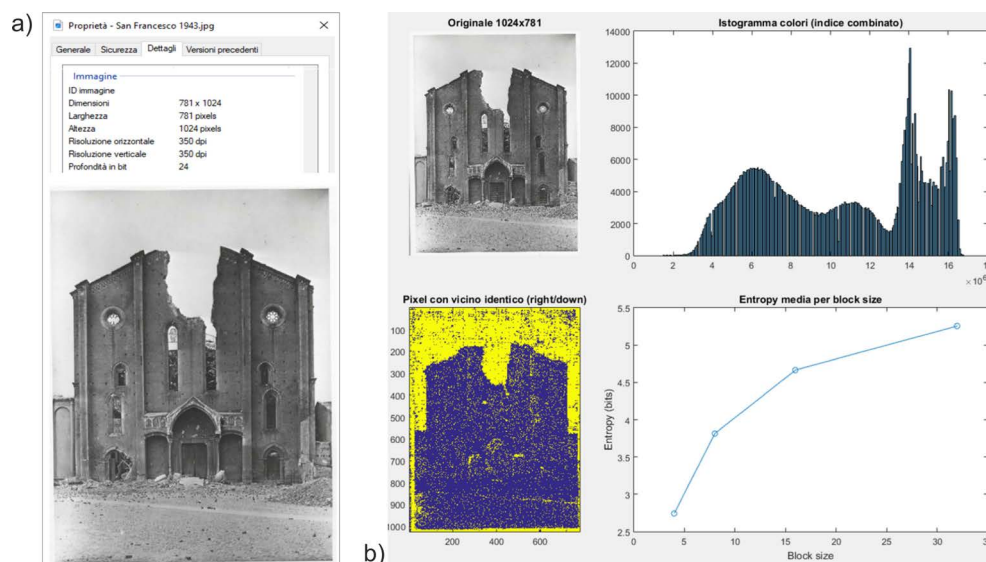


Figura 5 a) Immagine storica in formato JPG, reperita online, raffigurante la facciata della Basilica di San Francesco (Bologna) danneggiata dai bombardamenti del 1943; sono riportate le proprietà nominali del file, incluse la risoluzione di scansione (350 dpi) e le dimensioni in pixel. **b)** Output principali della pre-analisi diagnostica dell'immagine, comprendenti la mappa dei pixel con vicini identici (evidenziati in giallo), l'istogramma dell'indice cromatico combinato e l'andamento dell'entropia media calcolata su blocchi di pixel di dimensione crescente.

Figure 5 a) Historical JPG image retrieved online, depicting the façade of the Basilica of San Francesco (Bologna) damaged by the 1943 bombings; the nominal file properties are reported, including scanning resolution (350 dpi) and image size in pixels. **b)** Main outputs of the image diagnostic pre-analysis, including the map of pixels with identical neighbors (highlighted in yellow), the histogram of the combined chromatic index, and the trend of the mean entropy computed over pixel blocks of increasing size.

2.3 Historica: script principale

Questo modulo costituisce il cuore operativo della procedura e realizza la mappatura cromatica punto per punto della nuvola tridimensionale. Come input, riceve la nuvola di punti in formato testuale (.txt) e i parametri di camera definiti nei moduli precedenti. La nuvola viene inizialmente caricata e, se necessario, i valori cromatici possono essere normalizzati e la densità dei punti ridotta mediante sottocampionamento. Tali operazioni non modificano la geometria originale, ma rendono la nuvola più leggibile e facilmente gestibile, facilitando la visualizzazione e la selezione dei punti omologhi. La nuvola può essere roto-traslata manualmente fino a ottenere la posizione corrispondente alla vista utile per il confronto con l'immagine, oppure i parametri calcolati in PreCloud possono essere inseriti direttamente per velocizzare le operazioni. Infine, le coordinate 3D vengono trasformate nel sistema di riferimento della camera virtuale, permettendo all'utente di navigare nello spazio e ottenere la vista prospettica desiderata (Figura 6).

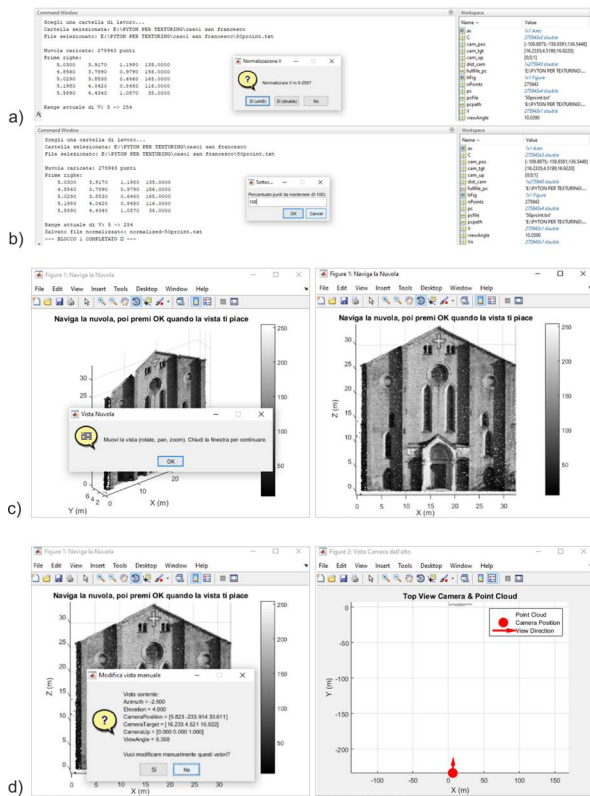


Figura 6 Prima parte del processo di elaborazione della nuvola di punti: **a)** normalizzazione dei canali colore della nuvola importata; **b)** possibile sotto-campionamento; **c)** rototraslazione per ottenere la vista desiderata; **d)** inserimento manuale dei parametri della camera e visualizzazione planimetrica della nuvola e della direzione di vista.

Figure 6 First part of the point cloud processing workflow: **a)** normalization of the color channels of the imported point cloud; **b)** optional subsampling; **c)** roto-translation to obtain the desired view; **d)** manual input of camera parameters and planimetric visualization of the point cloud and viewing direction.

Si nota che la distanza della camera virtuale mostrata nelle figure risulta “irrealistica” rispetto alla posizione reale del fotografo, probabilmente a meno di 50 m dalla facciata. Questo avviene perché il codice calcola la posizione e l’orientamento della camera unicamente per ottenere una proiezione coerente con l’immagine di riferimento, applicando i moduli di normalizzazione, rototraslazione e proiezione centrale. Sarebbe comunque possibile impostare manualmente la camera per avvicinarsi maggiormente alla distanza reale, ma ciò non apporterebbe alcun beneficio pratico, dato che il passaggio da 3D a 2D serve solo a ottenere immagini comparabili e coerenti per il *texturing* della nuvola di punti. Questo approccio è stato scelto perché permette di ottenere automaticamente una proiezione visivamente allineata senza dover definire manualmente la posizione esatta della camera o la lunghezza focale reale, semplificando il workflow anche su nuvole di punti complesse o sottocampionate.

Tecnicamente, la discrepanza nasce dal modo in cui il codice gestisce la proiezione prospettica: il parametro che definisce l’angolo di vista verticale della camera viene convertito in un fattore focale normalizzato utile per generare una proiezione coerente graficamente ma senza legame diretto con distanze fisiche reali. Prima della proiezione, la nuvola viene sottocampionata e ruotata per allinearsi alla vista desiderata, e la camera può essere automaticamente posizionata più lontano o più vicino per ottenere un allineamento visivo corretto con l’immagine di riferimento. Di conseguenza, i valori risultanti e la distanza numerica della camera non corrispondono a misure reali, ma garantiscono che gli oggetti appaiano proporzionati e frontalmente allineati nella proiezione 2D.

Successivamente alle operazioni di proiezione, si assegna una dimensione all’immagine 2D generata dalla vista selezionata della nuvola di punti. Questa scelta di larghezza e altezza in pixel non deve necessariamente corrispondere a dimensioni “reali”, **ma cerca di rispettare le proporzioni e la risoluzione dell’immagine storica utilizzata come riferimento**. La coerenza tra il render della nuvola e l’immagine storica è fondamentale per garantire un corretto allineamento dei punti, ed era già stata preliminarmente verificata tramite il modulo Prelmg, che aveva fornito indicazioni operative sulla risoluzione effettiva e sul rapporto d’aspetto dell’immagine (Figura 7).

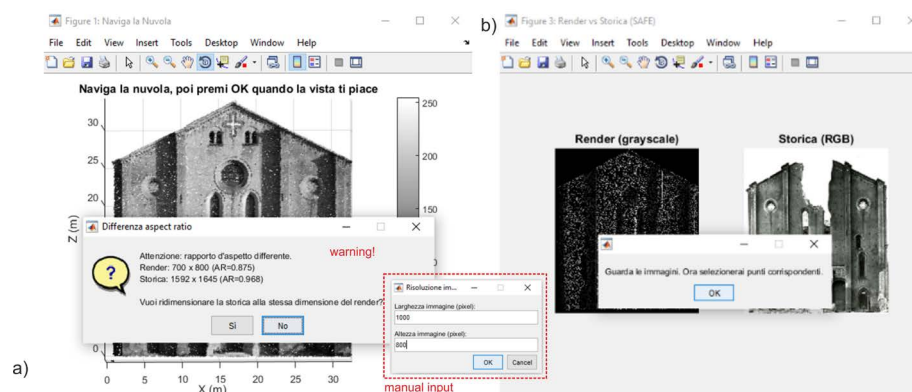


Figura 7 a) Assegnazione delle dimensioni 2D per la proiezione della nuvola di punti (riquadro rosso) e messaggio di allerta. Nota: l'immagine in figura non è la stessa utilizzata nella sezione precedente; **b)** immagine 2D della nuvola e immagine storica caricata fianco a fianco. Nota: la bassa risoluzione della nuvola di punti è per una veloce applicazione al caso pilota.

Figure 7 a) Assignment of the 2D dimensions for point cloud projection (red box) and display of a warning message.

Note: the image shown is not the same as that used in the previous section; b) 2D image of the point cloud and historical photograph loaded side by side. Note: the low resolution of the point cloud is adopted for a rapid application to the pilot case.

A questo punto viene caricata l'immagine storica corrispondente, che servirà come base per la mappatura cromatica dei punti 3D. Il passo successivo consiste nella selezione dei punti omologhi, cioè punti chiaramente riconoscibili sia nell'immagine 2D della nuvola renderizzata sia nell'immagine storica. L'utente individua almeno tre coppie di punti ben distinti, come angoli di finestre, spigoli di cornicioni o altre caratteristiche architettoniche stabili, cliccando direttamente sulle due immagini affiancate. Il software registra automaticamente le coordinate dei punti selezionati in ciascuna immagine, permettendo di calcolare la trasformazione geometrica necessaria per sovrapporre con precisione l'immagine storica alla vista della nuvola e preparare la successiva fase di trasferimento cromatico.

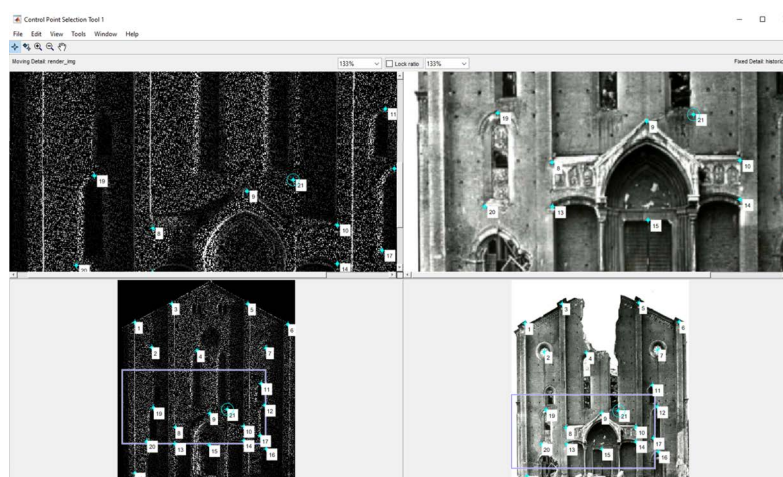


Figura 8 Interfaccia per la selezione dei punti omologhi: a sinistra, l'immagine renderizzata della nuvola di punti; a destra, l'immagine storica; i cerchi verdi indicano i punti scelti sulle due immagini, utilizzati come riferimento per l'allineamento.

Figure 8 Interface for homologous point selection: on the left, the rendered image of the point cloud; on the right, the historical image; green circles indicate the points selected on both images, used as references for the alignment.

Successivamente, la posizione di ciascun punto nel sistema di camera (X_c, Y_c, Z_c) viene proiettata sul piano immagine mediante un modello prospettico semplificato (proiezione centrale, *pinhole*). La proiezione genera le coordinate immagine (u, v), utilizzate per verificare se il punto ricade all'interno dei limiti del *raster* storico. I punti interni all'immagine ricevono il valore cromatico del pixel corrispondente, tramite campionamento del pixel più vicino o interpolazione bilineare, mentre quelli esterni vengono scartati o marcati come privi di informazione. Il risultato è una nuvola di punti arricchita di valori RGB, coerente dal punto di vista cromatico con la fotografia storica e con la prospettiva di osservazione. L'implementazione del *texturing* è lineare e deterministica, senza cicli iterativi complessi, concentrandosi sul passaggio dalle coordinate 3D a quelle immagine e sull'assegnazione dei valori radiometrici.

La selezione dei punti omologhi è stata effettuata secondo un approccio semi-interattivo guidato da criteri di riconoscibilità geometrica e stabilità strutturale. In particolare, vengono privilegiati elementi architettonici caratterizzati da elevato contrasto radiometrico e geometrico, quali spigoli, intersezioni tra superfici, cornici, aperture e discontinuità ben definite, che risultano facilmente individuabili in entrambe le rappresentazioni. Un requisito fondamentale nella scelta dei punti consiste nella loro distribuzione spaziale il più possibile uniforme sull'intera porzione di scena analizzata. Tale condizione è necessaria per garantire la stabilità numerica della trasformazione affine, evitando configurazioni degeneri o soluzioni fortemente influenzate da cluster locali di punti. In pratica, si cerca di evitare la concentrazione delle corrispondenze in una sola regione dell'immagine, privilegiando invece una copertura globale della facciata. In presenza di aree scarsamente leggibili nell'immagine storica o affette da degrado radiometrico, la selezione viene limitata agli elementi strutturalmente più affidabili, accettando una riduzione del numero locale di corrispondenze pur di mantenere la robustezza del sistema. Il numero minimo di tre coppie di punti rappresenta la condizione teorica per la determinazione della trasformazione affine, mentre in pratica vengono utilizzati un numero maggiore di corrispondenze per migliorare la stabilità della soluzione ai minimi quadrati e ridurre la sensibilità al rumore di selezione. Indipendentemente dalla disponibilità di un elevato numero di punti potenzialmente selezionabili, e considerando la natura intrinsecamente "*problem-oriented*" del repertorio di immagini storiche utilizzate, si ritiene che la fase di scelta delle corrispondenze non possa essere completamente demandata ad approcci automatici. La variabilità dei supporti, il degrado radiometrico e la presenza di distorsioni non modellate rendono infatti necessaria la supervisione diretta di un operatore esperto, in grado di interpretare correttamente il contenuto geometrico e storico delle immagini. In questa prospettiva, la selezione dei punti omologhi viene mantenuta come processo guidato dall'utente, piuttosto che automatizzato, al fine di preservare la qualità delle corrispondenze e ridurre il rischio di associazioni errate in contesti visivamente ambigui. Tuttavia, sono in corso sviluppi per introdurre un supporto semi-automatico alla selezione, basato su un approccio di tipo "*drag-based refinement*", nel quale la posizione iniziale del punto viene proposta automaticamente e successivamente raffinata mediante trascinamento interattivo da parte dell'utente. Tale strategia mira a ridurre il tempo operativo mantenendo al contempo il controllo umano sulla validazione finale delle corrispondenze.

Dopo la selezione dei punti omologhi, il software calcola automaticamente una trasformazione affine che allinea l'immagine storica alla vista 2D della nuvola renderizzata, deformandola e ridimensionandola in base ai punti di riferimento. La precisione dell'allineamento è valutata tramite la distanza residua in pixel tra i punti storici trasformati e quelli renderizzati, con statistiche che includono errore medio, RMS e massimo. Valori bassi indicano una registrazione coerente, mentre valori elevati segnalano possibili discrepanze o punti omologhi non ottimali. A partire dalla distribuzione dei residui è inoltre possibile introdurre una soglia operativa, basata su indicatori statistici quali RMS globale o percentili della distribuzione, al fine di distinguere in modo più chiaro tra residui di entità contenuta e residui elevati. In questo caso se il valore è oltre la media degli scarti per 2 volte la deviazione standard ($> 2\sigma$). Tale classificazione consente una

lettura più immediata della qualità dell'allineamento e facilita l'individuazione delle aree o delle corrispondenze che contribuiscono maggiormente all'errore complessivo. In questo contesto, la valutazione della qualità dell'allineamento non è soltanto un'analisi descrittiva a posteriori, ma costituisce un elemento operativo della pipeline. In particolare, l'errore residuo in pixel tra punti omologhi viene utilizzato come criterio quantitativo per guidare l'accettazione o la revisione della corrispondenza. La definizione di soglie statistiche basate sulla distribuzione dei residui (ad esempio media e deviazione standard) consente di introdurre una distinzione operativa tra allineamenti coerenti e situazioni in cui la presenza di *outlier* o di punti non rappresentativi compromette la stabilità della trasformazione. Tale approccio contribuisce a rendere il processo più riproducibile, riducendo la dipendenza da valutazioni esclusivamente qualitative e rendendo esplicito il legame tra scelta dei punti omologhi e qualità finale della registrazione.

Infine, i colori dell'immagine storica vengono riportati sui punti della nuvola tridimensionale, generando la versione texturizzata, confrontabile direttamente con la nuvola renderizzata originale (Figura 9). Altri dettagli implementativi e funzioni accessorie del codice Historica sono riportati nell'Appendice C, dove il codice completo è reso disponibile per consultazione, utilizzo o modifica.

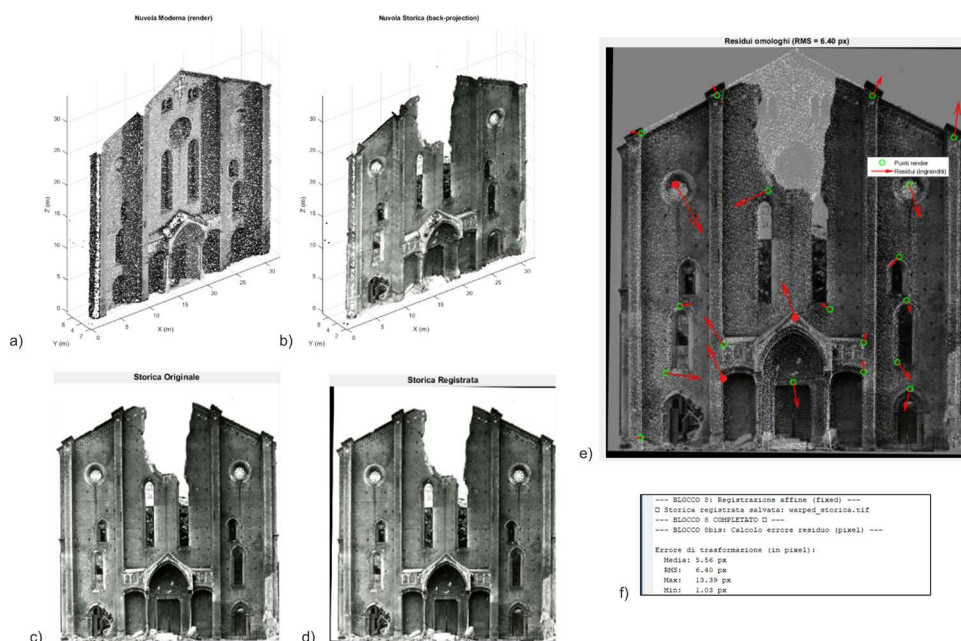


Figura 9 Risultato del *texturing* storico sulla nuvola di punti: **a)** nuvola renderizzata in scala di grigi; **b)** immagine storica allineata mediante punti omologhi; **c)** nuvola texturizzata con i colori storici proiettati sui punti sottocampionati; **d)** errore residuo e statistiche di allineamento visualizzati nell'interfaccia testuale di MATLAB. I cerchi rossi indicano i residui sopra soglia (2σ).

Figure 9 Result of the historical texturing applied to the point cloud: **a)** point cloud rendered in grayscale; **b)** historical image aligned using homologous points; **c)** textured point cloud with historical colors projected onto the subsampled points; **d)** residual error and alignment statistics displayed in the MATLAB textual interface. Red circles indicate residuals $> 2\sigma$.

2.3.1 Alcune note tecniche

1. Caricamento e parsing dei dati

La nuvola di punti viene letta come matrice Nx4 o Nx3, con le colonne X, Y, Z (e opzionale V), tramite istruzioni standard (`dload`, `load` o `importdata`). Queste colonne sono separate immediatamente per evitare ambiguità di tipo:

```
X = pts(:,1);
Y = pts(:,2);
Z = pts(:,3);
```

In modo analogo, l'immagine storica viene caricata in RGB 8-bit garantito (uint8), coerente con la pipeline del modulo PREIMG, solitamente con:

```
I = imread(img_file);
I = im2uint8(I(:,:,1:3));
```

Questa conversione elimina canali residui (alpha, CMYK) e uniforma la precisione.

2. Ricostruzione della camera utente

I parametri salvati da PreCloud (posizione camera, target, up-vector, campo visivo) vengono trasformati in un sistema di riferimento locale della camera. La rototraslazione non è implementata tramite toolbox fotogrammetrici, ma con operazioni vettoriali dirette. L'orientamento della camera è definito costruendo un sistema di riferimento ortonormale a partire dai parametri geometrici della vista. Il vettore forward è definito come la direzione tra la posizione della camera (campos) e il punto target della scena (camtarget). Il vettore right è ottenuto come prodotto vettoriale tra il forward e il vettore up iniziale (camup). Infine, il vettore true up viene ricalcolato per garantire l'ortogonalità del sistema, assicurando così la coerenza del riferimento locale della camera:

```
z_cam = normalize(camtarget - campos);
x_cam = normalize(cross(z_cam, camup));
y_cam = cross(x_cam, z_cam);
```

Questi tre assi costituiscono la matrice di rotazione R camera→mondo, invertita per proiettare i punti:

```
Pc = R' * ([X;Y;Z] - campos);
```

Il vettore risultante $P_c = [X_c, Y_c, Z_c]$ è nelle coordinate della camera.

3. Proiezione prospettica

L'algoritmo adotta un modello pinhole correlato al campo visivo verticale (camva).

Dalla geometria del FOV si ricava il coefficiente di scala:

```
f = 0.5 * H / tan(camva*pi/360);
```

dove H è l'altezza dell'immagine.

Le coordinate immagine preliminari sono quindi:

```
u = (Xc * f / Zc) + W/2;
v = -(Yc * f / Zc) + H/2;
```

con l'offset al centro del raster; il segno negativo su v mantiene il sistema immagine convenzionale (origine in alto a sinistra).

4. Controllo di validità e lettura del colore

Il codice non forza alcuna interpolazione sofisticata: se $Z_c \leq 0$, il punto è dietro alla camera e viene scartato.

Se u e v rientrano nel dominio raster,

```
if u>=1 && u<=W && v>=1 && v<=H
    Col = I(round(v), round(u), :);
else
    Col = [NaN NaN NaN];
end
```

La scelta di round() evita aliasing distribuito e rende la procedura stabile su dataset storici poco definiti.

I punti invalidi (NaN) non vengono eliminati fisicamente ma marcati, per conservare la

topologia originale della nuvola.

5. Composizione dell'output

L'output mantiene l'ordine della nuvola originale, con una matrice Nx6 (X Y Z R G B).

Questa scelta è intenzionale: evita la frammentazione della mesh e permette di esportare i dati direttamente in formati standard (ASCII + RGB, LAS/PLY con attributi) senza necessità di triangolazione.

2.3.2 Consigli operativi

Per ottimizzare la procedura, ridurre i tempi di elaborazione e garantire la coerenza dei risultati, è consigliabile seguire alcune indicazioni operative. Tali passaggi non sono obbligatori, ma rappresentano una prassi consolidata per un utilizzo corretto del pacchetto software.

1. I parametri di camera devono essere quelli ottenuti tramite il modulo PreCloud o procedure equivalenti, in quanto derivano da un'interazione diretta con la nuvola di punti.
2. Si sconsiglia la modifica manuale dei parametri di camera, poiché le fasi di trasformazione e proiezione dipendono in modo critico dalla vista effettivamente impostata sull'oggetto tridimensionale.
3. L'immagine storica utilizzata per il *texturing* deve coincidere con quella analizzata nel modulo Prelmg, così da mantenere coerenza tra valutazione qualitativa e proiezione.
4. Non è necessario applicare un downsampling preliminare all'immagine; tuttavia, qualora Prelmg individui un fattore di scala utile (ad esempio 2 o 4), è consigliabile ridimensionare l'immagine prima della fase di *texturing*, per garantire coerenza geometrica e radiometrica.
5. È preferibile evitare nuvole di punti che non descrivano in modo sufficientemente dettagliato la geometria dell'oggetto. Il concetto non è legato a una specifica soglia di densità, ma alla capacità della nuvola di preservare elementi geometrici riconoscibili e utilizzabili durante la fase di proiezione. Quando tali elementi non sono più chiaramente identificabili, la qualità del *texturing* e la stabilità della procedura possono risultare compromesse.
6. È inoltre consigliabile verificare la presenza di valori estremi lungo l'asse di profondità (Z) o di cluster fortemente separati rispetto al corpo principale della nuvola. In questo contesto, per valori estremi si intendono punti o gruppi di punti localizzati a distanze significativamente maggiori rispetto alla distribuzione principale della nuvola e quindi in grado di influenzare la geometria della proiezione. Tali configurazioni possono causare saturazioni cromatiche, una distribuzione non uniforme delle proiezioni e una concentrazione eccessiva dei punti su un numero limitato di pixel dell'immagine.
7. La presenza di cluster molto lontani dalla camera può generare saturazioni cromatiche e una concentrazione eccessiva di proiezioni su un numero limitato di pixel dell'immagine.
8. La procedura è completamente replicabile variando esclusivamente tre input: la nuvola di punti, l'immagine storica e i parametri di camera. Tutti gli altri passaggi rimangono invariati e deterministici.

3. I casi di studio

Per verificare la robustezza del *workflow* proposto e valutare il comportamento dei tre moduli software (PreCloud, Prelmg e Historica), sono stati selezionati tre casi di studio caratterizzati da condizioni geometriche e radiometriche differenti. Ciascuno presenta caratteristiche particolari che mettono in luce diversi aspetti del metodo, consentendo di mostrare la versatilità del *workflow* nell'adattarsi a immagini e geometrie differenti.

Il primo caso riguarda un manufatto architettonico, la facciata della Chiesa di San Francesco a Bologna, per il quale sono disponibili sia rilievi tridimensionali ad alta risoluzione sia fotografie d'archivio risalenti al secondo conflitto mondiale. In questo contesto la pipeline è utilizzata per proiettare il contenuto informativo delle immagini storiche sulla nuvola di punti attuale, consentendo di analizzare con precisione differenze morfologiche, lacune e deformazioni post belliche. La complessità formale dell'edificio (presenza di elementi modulari, apparato decorativo, superfici non planari) rende il confronto fra le due epoche un banco di prova significativo per la stabilità geometrica del *texturing*.

Il secondo caso, relativo al porto canale di Cesenatico, presenta caratteristiche geometriche quasi planari e una forte stabilità degli elementi antropici, che permettono di assumere riferimenti altimetrici costanti nel tempo. In questo scenario, le immagini storiche costituiscono una fonte metrica utile a ricostruire indirettamente la variazione relativa del livello marino. L'applicazione del metodo consente di trasferire le informazioni radiometriche dell'epoca su un modello tridimensionale attuale, permettendo di confrontare la quota della superficie dell'acqua con soglie, cordoli e elementi architettonici storicamente riconoscibili.

Il terzo caso è volutamente atipico, in quanto utilizza un'immagine termica al posto di una fotografia RGB. Le termografie presentano una struttura radiometrica meno dettagliata nelle frequenze spaziali elevate, ma mostrano pattern termici stabili e interpretabili, associati a giunti, materiali o anomalie di temperatura. Questo scenario è particolarmente adatto a verificare la compatibilità del *texturing* con sorgenti a basso contrasto visivo e a valutare se l'algoritmo di proiezione, basato su coordinate dirette camera-immagine, sia in grado di preservare informazioni non puramente fotometriche.

Nel complesso, i tre casi coprono condizioni geometriche e radiometriche molto diverse: superfici fortemente articolate e danneggiate (San Francesco), geometrie lineari e campionamento altimetrico su elementi rigidi (Cesenatico), distribuzioni radiometriche non visive ma termiche (immagine IR). Questa triangolazione costituisce un test di robustezza del metodo, che non dipende dalla natura dell'immagine, ma dalla corretta definizione del punto di vista (PreCloud), dalla caratterizzazione del contenuto informativo dell'immagine storica (PreImg) e dalla sua proiezione coerente sul modello tridimensionale (Historica).

3.1 Caso 1 - San Francesco

La Basilica di San Francesco a Bologna, fondata nel XIII secolo e consacrata nel 1251, rappresenta uno dei primi esempi di architettura gotica di derivazione francese in Italia. Durante la Seconda guerra mondiale, la facciata subì gravi danni a seguito dei bombardamenti, con crolli parziali e diffuse lesioni strutturali.

Per questo caso di studio è stata utilizzata un'unica fotografia storica del 1943, che ritrae la facciata immediatamente dopo gli eventi bellici. L'immagine documenta direttamente le aree di collasso e i quadri fessurativi indotti dalle esplosioni, costituendo una testimonianza visiva preziosa per l'analisi dei danni strutturali. Essa è stata scelta per applicare il metodo di *texturing* storico su nuvola di punti, con l'obiettivo di riportare spazialmente le informazioni di danno all'interno di un modello tridimensionale coerente. Particolare attenzione è rivolta alla distribuzione delle lesioni e alle zone di debolezza della facciata, il cui comportamento sotto i bombardamenti è assimilabile, per effetti strutturali, a uno scuotimento di tipo sismico.

La fotografia storica proviene dagli archivi documentari pubblici, reperibile attraverso il Catalogo Generale dei Beni Culturali (ICCD, Ministero della Cultura), e più precisamente associata alla scheda n. 0800265969, che documenta lo stato della facciata con evidenza di crolli parziali e gravi alterazioni della continuità muraria [4].

Prima dell'applicazione del metodo, l'immagine è stata sottoposta a un preprocessing mirato

esclusivamente a migliorarne la leggibilità e l'utilizzabilità per l'allineamento e il *texturing*. Sono state ritagliate le porzioni non informative, quali cielo e aree esterne all'edificio (Figura 10), e applicati interventi di miglioramento del contrasto e della dinamica radiometrica per rendere più riconoscibili i dettagli architettonici e le discontinuità strutturali, senza introdurre nuove informazioni o alterare la geometria originale. Le modalità operative di tale fase possono variare in funzione delle caratteristiche dell'immagine e degli strumenti utilizzati dall'operatore. Possono essere impiegate procedure automatiche di ottimizzazione dell'immagine oppure regolazioni manuali di contrasto, luminosità, intensità e bilanciamento cromatico. In tutti i casi, l'obiettivo rimane esclusivamente quello di migliorare la leggibilità dei dettagli esistenti, senza modificare il contenuto informativo o la struttura geometrica dell'immagine. Queste operazioni non modificano la struttura spaziale dell'immagine, ma facilitano l'interpretazione visiva e la selezione dei punti omologhi, costituendo una fase preparatoria di supporto alla procedura di proiezione e garantendo la coerenza con i dati tridimensionali di riferimento. Nel caso studio presentato in questo lavoro, tali operazioni sono state eseguite mediante gli strumenti standard di miglioramento dell'immagine disponibili in CorelDraw/Corel PHOTO-PAINT, utilizzati esclusivamente per aumentare la visibilità dei dettagli architettonici utili alle successive fasi di allineamento e proiezione.

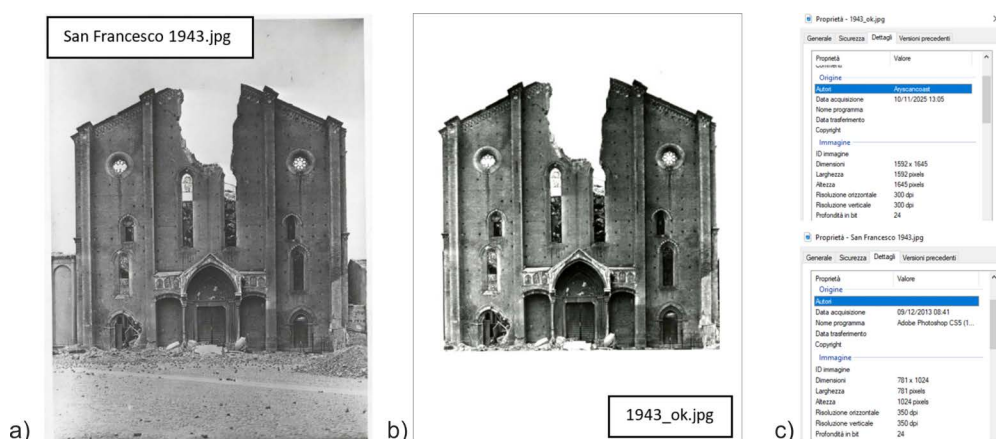


Figura 10 a) La facciata della Basilica di San Francesco (1943) gravemente danneggiata dai bombardamenti della Seconda guerra mondiale; b) l'immagine ottimizzata e preparata per la fase di *texturing*; c) i dettagli di risoluzione e dimensione delle immagini.

Figure 10 a) Façade of the Basilica of San Francesco (1943), severely damaged by World War II bombings; b) optimized image prepared for the texturing phase; c) image resolution and size details.

La campagna di rilievo fotogrammetrico, finalizzata esclusivamente a ottenere un riferimento geometrico tridimensionale per le successive fasi di *texturing*, è stata condotta nell'ottobre 2025. Sono state scattate 30 immagini con fotocamera Nikon D3300, focale 35 mm, a una distanza media di circa 30 m, seguendo un percorso a semicerchio centrato sulla porzione centrale della facciata, con sovrapposizione longitudinale e trasversale tra il 70% e l'80%.

La ricostruzione tridimensionale è stata realizzata mediante Structure-from-Motion (SfM) con Agisoft Metashape [1]. La scala del modello è stata definita in post-processing utilizzando come vincolo la larghezza complessiva della facciata, fornendo una *baseline* metrica per garantire la coerenza spaziale. La nuvola di punti risultante, metrica e coerente geometricamente, costituisce il *dataset* di supporto per l'orientamento, l'allineamento e la proiezione delle immagini storiche. La nuvola è stata infine sottoposta a ispezione morfologica, con analisi delle deviazioni rispetto a superfici di riferimento stimate mediante *fitting* sui punti della facciata, considerando sia l'intera

superficie sia porzioni selezionate. Le mappe morfologiche ottenute (Figura 11) evidenziano irregolarità locali, discontinuità e pattern deformativi, utili sia per interpretare i danni storici sia per la corretta integrazione con le immagini d'archivio. Queste ispezioni sono state realizzate mediante i moduli di Polyworks [5] ampiamente descritte in bibliografia [Pesci et al., 2013].

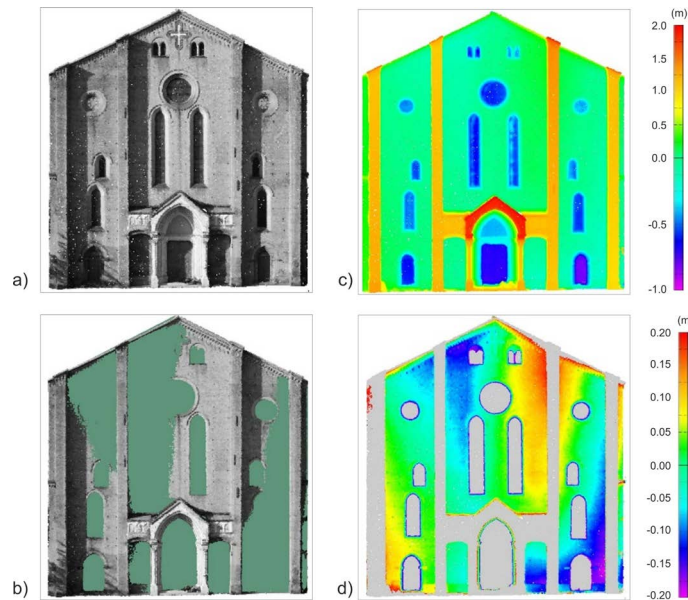


Figura 11 Dettagli della nuvola di punti della facciata di San Francesco: **a)** nuvola di punti SfM; **b)** piano di riferimento ottenuto tramite *fitting* sui punti della superficie muraria; **c)** mappa delle differenze punto-piano nell'intervallo $(-2, 1)$ m; **d)** mappa delle differenze punto-piano nell'intervallo $(-0.2, 0.2)$ m.

Figure 11 Details of the point cloud of the San Francesco façade: **a)** SfM point cloud; **b)** reference plane obtained by fitting to points on the masonry surface; **c)** point-to-plane difference map in the range $(-2, 1)$ m; **d)** point-to-plane difference map in the range $(-0.2, 0.2)$ m.

3.1.1 Analisi preliminari

Come descritto nelle sezioni precedenti e illustrato in Figura 10, l'immagine storica, opportunamente ritagliata e ottimizzata, è stata analizzata mediante il modulo accessorio Prelmg per ottenere indicazioni operative sulle dimensioni e sulla risoluzione effettiva da adottare nella generazione dell'immagine di proiezione della nuvola di punti. L'analisi ha considerato indicatori diagnostici della reale quantità di informazione radiometrica disponibile, tra cui la ridondanza locale dei pixel, l'occupazione dello spazio colore e l'andamento dell'entropia a diverse scale spaziali. I risultati hanno fornito un riferimento oggettivo sia per la scelta delle dimensioni della vista renderizzata, sia per valutare la qualità e l'affidabilità dei punti omologhi nelle fasi successive. In Figura 12 sono riportati i risultati preliminari, riferiti a un'immagine distinta, rielaborata e salvata con dimensioni in pixel e risoluzione nominale superiori rispetto ai casi precedenti.

L'analisi preliminare della nuvola di punti ha inoltre consentito di definire la configurazione di vista più coerente con l'immagine storica e di esaminare la distribuzione delle distanze dei punti dal piano di proiezione. Nel caso in esame, la facciata è osservata in modo pressoché frontale, per cui le distanze risultano uniformi, senza introdurre variazioni significative nella sensibilità metrica dei punti. In Figura 13 è mostrata la nuvola orientata interattivamente secondo la vista scelta, insieme alla mappa delle distanze dei punti rispetto alla camera virtuale, confermando la sostanziale uniformità della configurazione di osservazione.

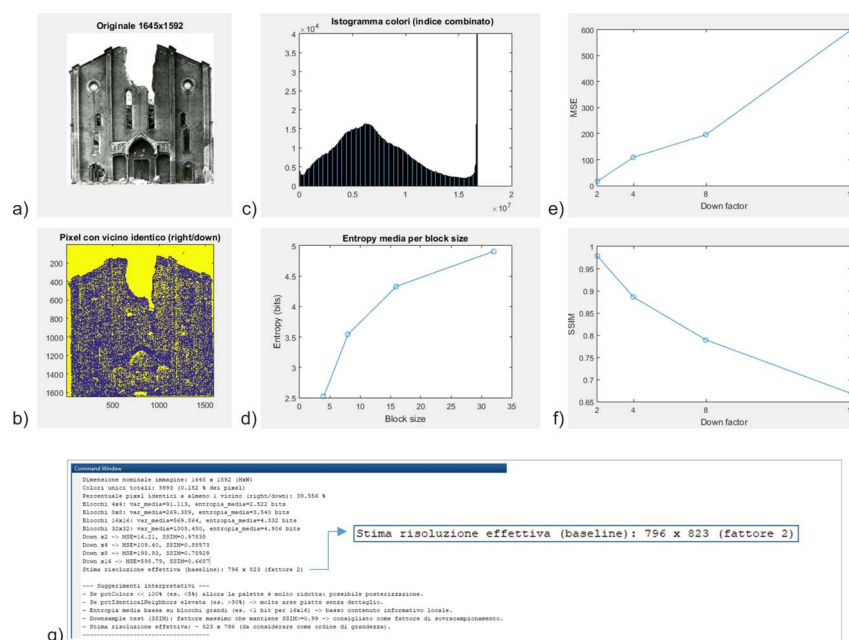


Figura 12 Analisi preliminare dell'immagine storica mediante Prelmg: **a)** immagine storica pre-processata, ottenuta tramite ritaglio delle parti non pertinenti alla facciata e ottimizzazione dei principali parametri radiometrici; **b)** mappa dei pixel che presentano almeno un vicino con caratteristiche radiometriche identiche, indicativa della ridondanza informativa locale; **c)** istogramma dei valori cromatici espresso tramite indice RGB combinato, utilizzato per valutare l'effettiva occupazione dello spazio colore; **d)** andamento dell'entropia media al variare della dimensione dei blocchi di analisi, impiegato per stimare la scala spaziale dell'informazione radiometrica utile; **e)** MSE (Mean Squared Error); **f)** SSIM (Structural Similarity Index); **g)** risultato testuale (stima risoluzione e parametri).

Figure 12. Preliminary analysis of the historical image using Prelmg: **a)** Pre-processed historical image, obtained by cropping areas not relevant to the façade and optimizing the main radiometric parameters; **b)** map of pixels exhibiting at least one neighbor with identical radiometric characteristics, indicative of local informational redundancy; **c)** histogram of chromatic values expressed through a combined RGB index, used to assess the effective occupation of the color space; **d)** trend of the mean entropy as a function of the analysis block size, employed to estimate the spatial scale of useful radiometric information; **e)** MSE (Mean Squared Error); **f)** SSIM (Structural Similarity Index); **g)** textual output (resolution estimate and parameters).

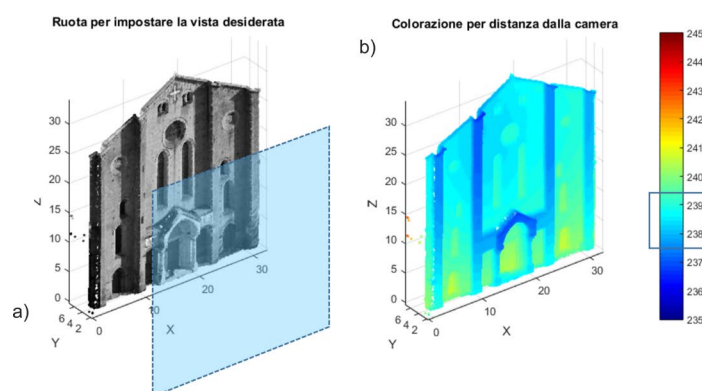


Figura13 Analisi preliminare della distanza dei punti dalla camera virtuale: **a)** nuvola di punti importata, con il piano di vista scelto visualizzato in trasparenza; **b)** nuvola di punti colorata in funzione della distanza dal centro della camera.

Figure 13 Preliminary analysis of point-to-camera distance: **a)** imported point cloud, with the selected view plane displayed in transparency; **b)** point cloud color-coded as a function of the distance from the camera center.

3.1.2 Texturing di San Francesco

Seguendo l'iter operativo descritto nella sezione 2.3, vengono qui presentati i risultati della texturizzazione della nuvola di punti della facciata tramite l'immagine storica del 1943. In Figura 14 sono illustrate le fasi principali del workflow: dalla generazione dell'immagine 2D con dimensioni e risoluzione coerenti con l'immagine storica, alla selezione dei punti omologhi, fino all'osservazione dei residui tra i pixel delle due immagini e alla produzione della nuvola di punti texturizzata.

In Figura 15 viene mostrata la nuvola con i colori originali e la stessa nuvola texturizzata, accompagnata dalla mappa delle differenze. Quest'ultima evidenzia come le porzioni di facciata crollate durante i bombardamenti, visibili in bianco e corrispondenti al cielo dietro le aree mancanti, siano correttamente collocate in zone con valori differenti rispetto alla superficie conservata. Sulla nuvola texturizzata è stato inoltre possibile tracciare con precisione i contorni delle discontinuità, integrando il dato storico con la geometria moderna e consentendo un'interpretazione puntuale dei danni osservabili.

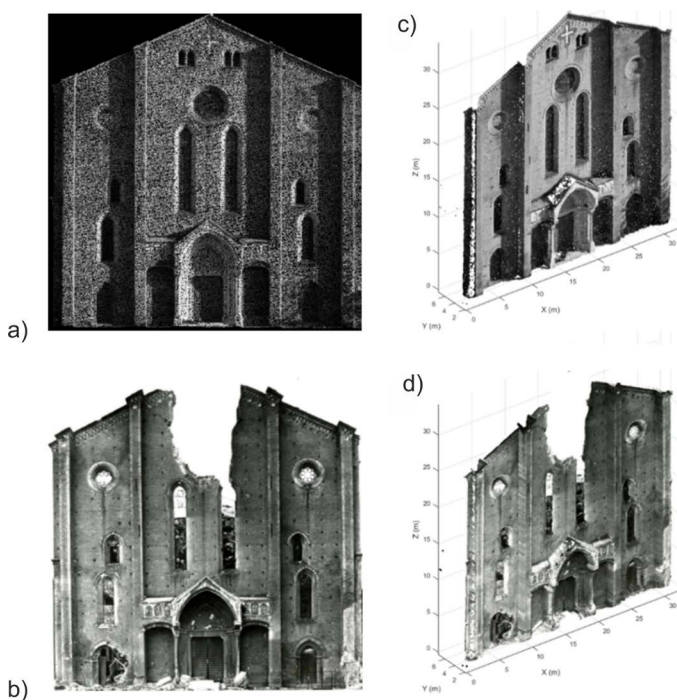


Figura 14 Fasi principali del workflow di texturizzazione: a) immagine 2D ottenuta proiettando la nuvola sul piano di vista; b) immagine storica; c) nuvola di punti originale; d) nuvola di punti texturizzata con informazioni cromatiche storiche.

Figure 14 Main stages of the texturing workflow: a) 2D image obtained by projecting the point cloud onto the view plane; b) historical image; c) original point cloud; d) point cloud textured with historical chromatic information.

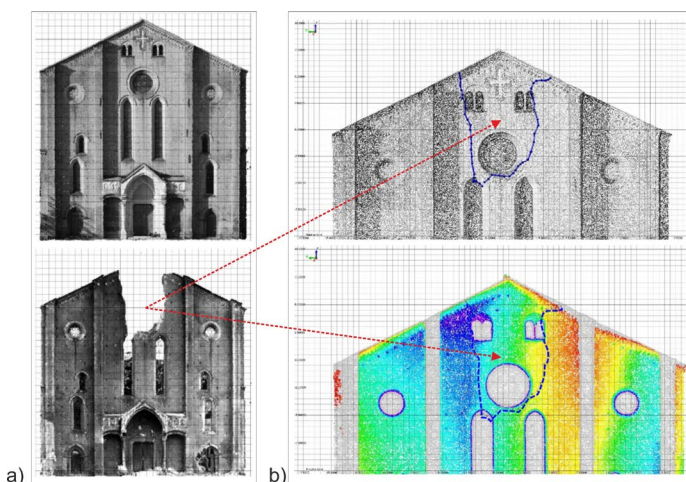


Figura 15 Risultati del texturing: a) nuvola di punti originale e texturizzata; b) polilinea di contorno dell'area bianca cioè dell'area danneggiata riportata sulla nuvola di punti e sulla mappa delle differenze punto-piano.

Figure 15 Texturing results: a) original and textured point cloud; b) polyline outlining the white area, corresponding to the damaged region, reported on both the point cloud and the point-to-plane difference map.

3.2 Caso 2 - Porto canale di Cesenatico

Il porto canale di Cesenatico rappresenta un contesto interessante per il metodo, poiché combina rilevanza storica, paesaggistica e potenziale valore scientifico. Si tratta di un canale artificiale di origine antica, tradizionalmente attribuito a un progetto leonardesco e successivamente modificato nel tempo, che comunica direttamente con il mare Adriatico. In questa configurazione, il livello dell'acqua lungo le sponde può essere assunto come indicatore del livello marino locale. Il caso di studio si basa su una fotografia storica risalente alla fine del XIX secolo o ai primi anni del Novecento, selezionata per la buona leggibilità del fronte urbano e della sponda del canale. L'immagine, tratta dal volume "Cesenatico: Immagini della Memoria" [Masacci e De Lucia, 1983] e riportata anche nella tesi di laurea di Castellucci [2010], mostra le palazzine affacciate sul canale, il parapetto e la muratura di banchina, con evidenza della fascia scura prodotta dall'azione dell'acqua marina, interpretabile come banda del livello medio del mare. L'obiettivo dell'applicazione è duplice: verificare la capacità del metodo di co-registrazione e *texturing* storico in un contesto urbano fortemente trasformato e valutare la possibilità di estrarre informazioni metriche indirette sul livello marino passato a partire da una documentazione fotografica limitata. Il porto canale costituisce un caso interessante, caratterizzato da una storia di interventi urbani e ristrutturazioni nel corso dei decenni. Per questo motivo, la selezione dei punti omologhi è stata condotta con attenzione, privilegiando dettagli architettonici e altri elementi urbanistici stabili nell'area.

La fotografia storica è stata sottoposta a un *preprocessing* mirato a migliorarne il contrasto e a rendere chiaramente individuabile la linea di separazione tra acqua e muratura. In particolare, la traccia del battente è stata enfatizzata mediante sbiancatura selettiva per compensare la perdita di definizione dovuta alla grana e all'invecchiamento del materiale fotografico.

La nuvola di punti di riferimento è stata ottenuta da un rilievo fotogrammetrico contemporaneo (SfM) eseguito nel maggio 2023. Poiché le porzioni inferiori della muratura risultano oggi sommerse e non direttamente rilevabili, la nuvola è stata estesa virtualmente verso il basso, ricostruendo il prolungamento delle superfici verticali. Questa operazione consente la proiezione coerente del livello dell'acqua storico su un modello completo, anche in assenza di riferimenti topografici originali.

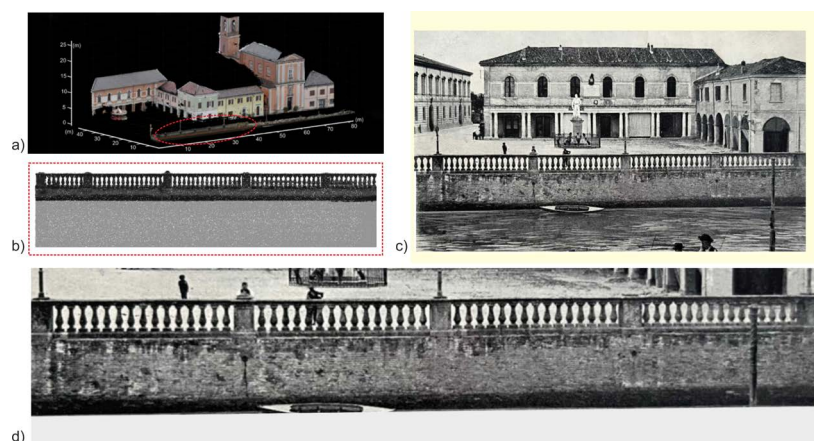


Figura 16 a) Nuvola di punti derivata dal rilievo fotogrammetrico (giugno 2023); b) porzione della nuvola con estensione virtuale della muratura sommersa; c) immagine storica (circa 1900); d) dettaglio dell'immagine storica editata e utilizzata per il confronto diretto con il rilievo.

Figure 16 a) Point cloud obtained from photogrammetric survey (June 2023); b) portion of the point cloud with virtual extension of the submerged masonry; c) historical image (circa 1900); d) detail of the edited historical image used for direct comparison with the survey.

Il *texturing* è stato applicato seguendo lo stesso *workflow* adottato per il caso precedente: dall'analisi metrica dei risultati emerge che il livello del mare all'inizio del Novecento era inferiore di circa 1.8 m rispetto a quello attuale. L'incertezza fisica associata è dominata dalle oscillazioni di marea locali (0.4-0.8 m), superiori all'errore intrinseco del metodo, stimato intorno a 0.4 m sulla base della dispersione dei punti lungo la linea di battente. Quest'ultimo rappresenta un limite inferiore dell'incertezza complessiva, mentre la variabilità naturale del livello marino costituisce la principale fonte di indeterminazione.

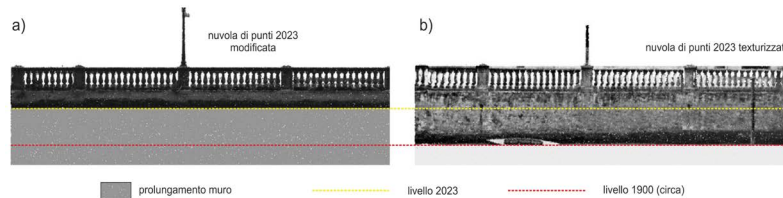


Figura 17 a) Nuvola contemporanea con estensione virtuale; b) nuvola texturizzata con immagine storica. La linea rossa tratteggiata indica il livello storico ricostruito e la sua proiezione sul modello attuale.

Figure 17 a) Contemporary point cloud with virtual extension; b) point cloud textured with the historical image. The dashed red line indicates the reconstructed historical level and its projection onto the current model.

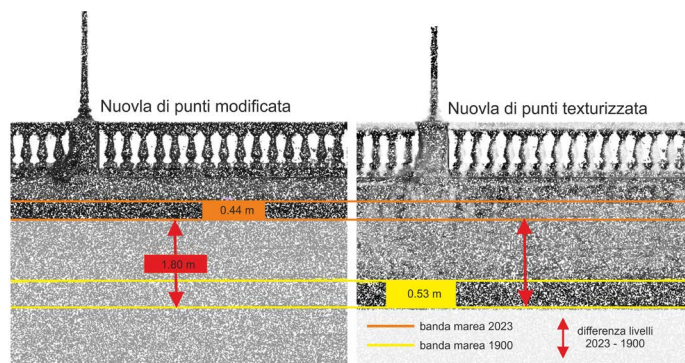


Figura 18 Nuvola di punti con colorazione moderna e storica. In arancio, il livello marino contemporaneo e la banda bagnata (spessore 0.44 m); in giallo, livello storico e banda bagnata corrispondente (0.53 m). La doppia freccia rossa indica la differenza altimetrica di 1.80 m.

Figure 18 Point cloud with modern and historical colorization. In orange, the contemporary sea level and the wet band (thickness 0.44 m); in yellow, the historical level and the corresponding wet band (0.53 m). The red double arrow indicates an elevation difference of 1.80 m.

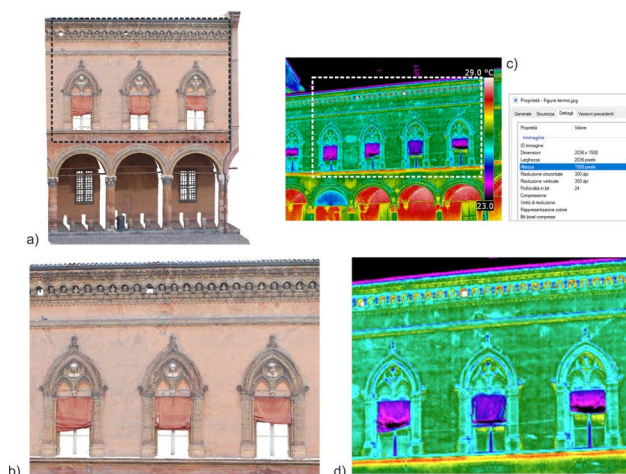
È importante sottolineare che il livello del mare nel Nord Adriatico è soggetto a variabilità naturale ben documentata, misurata da reti mareografiche come la Rete Mareografica Nazionale (RMN, [6]), la Rete Mareografica della Laguna di Venezia e stazioni storiche come Trieste - Molo Sartorio (CNR-ISMAR, [7 e 8]). Le osservazioni comprendono oscillazioni giornaliere, componenti stagionali e trend pluridecennali, disponibili anche tramite archivi internazionali come il PSMSL [9 e 10]. Nel contesto di questo caso di studio, la fascia bagnata visibile sulle murature storiche riflette l'azione combinata di marea, condizioni meteorologiche e permanenza media dell'acqua. La consapevolezza dell'ampiezza tipica delle oscillazioni giornaliere consente tentativi di interpretazione dell'incertezza associata alla stima del livello marino storico, che rappresenta un limite intrinseco della ricostruzione più che una fonte di errore metodologico. Per approfondimenti sui trend storici del livello marino, si rimanda a studi dedicati come Meli et al. [2021].

3.3 Caso 3 - Immagine termica (RGB)

L'ultimo caso di studio riguarda la texturizzazione di un'immagine derivata da rilievi termografici della facciata del palazzo di Corte Isolani a Bologna, lato Piazza Santo Stefano, realizzati nel 2012 durante realizzati nell'ambito di esperimenti di telerilevamento e integrazione dati. Le acquisizioni sono state effettuate con una termocamera FLIR T620, con risoluzione 640×480 pixel [11]. Per il presente studio, l'immagine termografica è considerata nella sua forma *raster* finale, priva delle informazioni radiometriche originali, e viene trattata analogamente a una normale immagine fotografica. In Figura 19 sono riportate la nuvola di punti originale ottenuta tramite TLS e l'immagine IR utilizzata per il *texturing*, insieme alla selezione dell'area di interesse.

Figura 19 a) Nuvola di punti TLS; b) selezione dell'area di interesse; c) immagine termografica; d) porzione selezionata per l'analisi.

Figure 19 a) TLS point cloud; b) selection of the area of interest; c) thermographic image; d) selected portion used for the analysis.



Le elaborazioni preliminari hanno permesso di valutare il contenuto radiometrico dell'immagine nella zona di interesse e di definire la dimensione in pixel più adatta per le fasi successive (Figura 20). La pre-analisi della nuvola di punti ha inoltre evidenziato variazioni di pochi metri nella distanza dei punti rispetto al centro di vista, coerentemente con la geometria relativamente semplice della facciata (Figura 21).

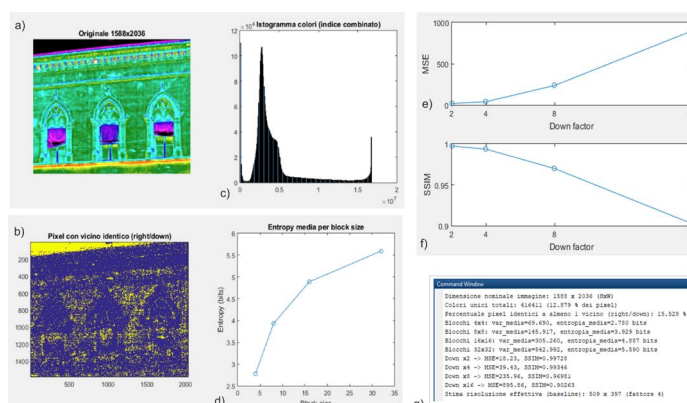


Figura 20 Analisi preliminare dell'immagine: a) immagine originale; b) mappa dei pixel identici; c) istogramma dei colori con indice combinato; d) entropia; e) MSE; f) SSIM; g) output testuale con risoluzione consigliata (circa 510×400 pixel).

Figure 20 Preliminary analysis of the image: a) original image; b) map of identical pixels; c) color histogram using the combined index; d) entropy; e) MSE; f) SSIM; g) textual output with recommended resolution (approximately 510×400 pixels).

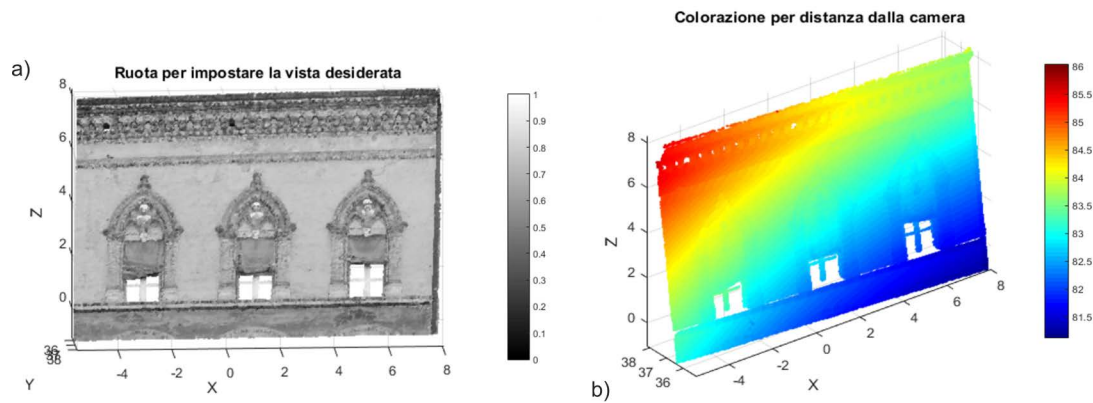


Figura 21 Analisi preliminare della nuvola di punti: **a)** vista selezionata; **b)** mappa delle distanze dal punto di osservazione.

Figure 21 Preliminary analysis of the point cloud: a) selected view; b) distance map from the observation point.

Dopo aver estratto le informazioni sulla dimensione e sulla risoluzione utile dell'immagine, nonché sulla posizione della camera e sulla distribuzione delle distanze dei punti, si è proceduto alla texturizzazione vera e propria. In questo contesto, i moduli accessori PreCloud e Prelmg risultano particolarmente utili: PreCloud permette di definire preliminarmente i parametri di vista su una nuvola fortemente sottocampionata, riducendo il carico computazionale nelle fasi successive, mentre Prelmg fornisce indicazioni operative sulla risoluzione e la dimensione dell'immagine da utilizzare nella proiezione 2D. I risultati, mostrati nelle Figure 22 e 23, evidenziano l'effetto della trasformazione affine applicata all'immagine termica, la mappa dei residui rispetto all'assetto originale e la nuvola di punti texturizzata. La qualità del metodo è riscontrabile anche qualitativamente: le principali discontinuità geometriche sono preservate, le linee architettoniche risultano coerenti con la geometria della nuvola, i cornicioni seguono in modo continuo la distribuzione cromatica e le decorazioni mantengono una corretta corrispondenza tra informazione geometrica e radiometrica.

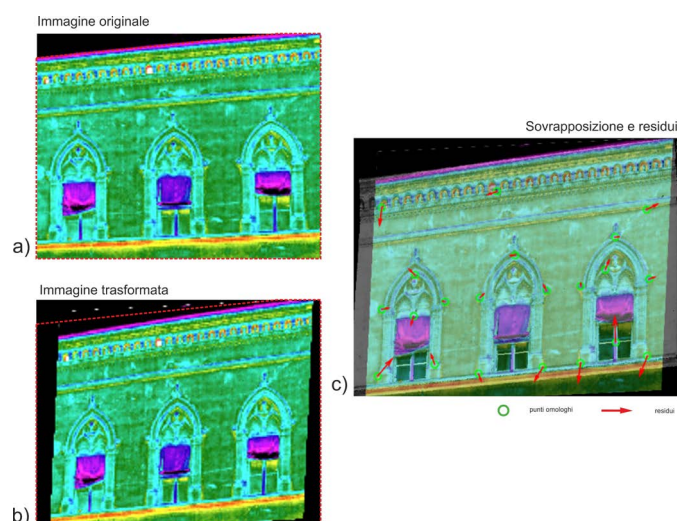


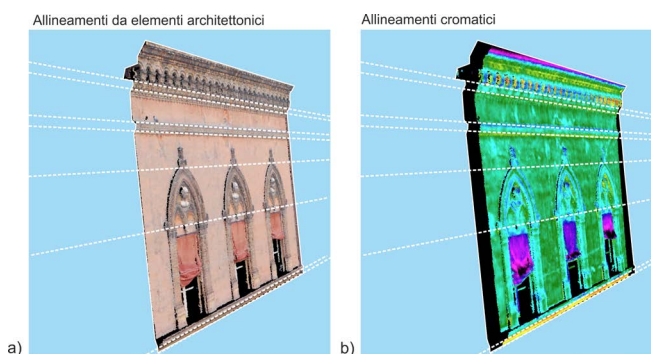
Figura 22 **a)** Immagine originale; **b)** immagine trasformata (trasformazione affine); **c)** mappa dei residui (pixel) su immagini sovrapposte.

Figure 22 a) Immagine originale; b) immagine trasformata (trasformazione affine); c) mappa dei residui (pixel) su immagini sovrapposte.

Figura 23 Confronto visivo tra la nuvola originale e quella texturizzata con l'utilizzo delle geometrie: **a)** allineamenti tracciati sulla nuvola di punti con i dettagli architettonici; **b)** osservazione del dato cromatico sugli allineamenti tracciati.

Figure 23 Visual comparison between the original and the textured point cloud using the geometries:

a) Alignments traced on the point cloud with architectural details; **b)** observation of the chromatic data along the traced alignments.



3.4 Caso aggiuntivo

Ascopodi verifica operativa del metodo, è stata condotta un'analisi utilizzando un'immagine fotografica recente, reperita da fonti online e riferita allo stato attuale dell'area di studio. L'immagine è stata trattata come dato indipendente e sottoposta alla stessa procedura di allineamento e texturizzazione descritta nei paragrafi precedenti. L'applicazione della pipeline ha prodotto un allineamento coerente tra immagine e modello tridimensionale, come evidenziato dai valori dei residui calcolati sui punti omologhi. In particolare, l'errore medio risulta pari a 3.8 pixel, con un valore RMS di 4.4 pixel, un massimo di 9.1 pixel e un minimo di 1.2 pixel. Tali valori indicano una buona qualità della registrazione nel contesto della risoluzione dell'immagine utilizzata. La Figura 24 mostra l'output della procedura, evidenziando la distribuzione dei residui e la coerenza complessiva dell'allineamento.

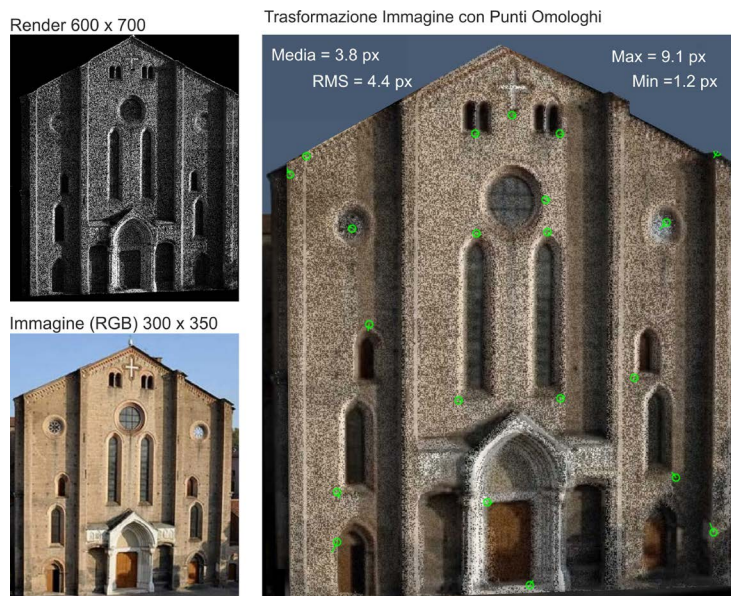


Figura 24 Figura 24. Analisi di verifica condotta utilizzando un'immagine recente reperita da fonti online. Il rendering della nuvola di punti (risoluzione 600×700 pixel) e la fotografia originale (300×350 pixel) sono affiancati all'immagine trasformata ottenuta mediante i punti omologhi, rappresentati con cerchi verdi. I residui dell'allineamento, visualizzati tramite vettori, mostrano una distribuzione contenuta, con valore RMS pari a 4.4 pixel. Nel caso analizzato non si evidenziano outlier secondo il criterio adottato.

Figure 24 Validation analysis performed using a recent image retrieved from online sources. The point cloud rendering (600×700 pixels) and the original photograph (300×350 pixels) are shown together with the transformed image obtained from the homologous points (green circles). Alignment residuals are represented by vectors and show a limited dispersion, with an RMS value of 4.4 pixels. No outliers are detected according to the adopted threshold criterion.

Il codice utilizzato per l'implementazione della procedura è riportato nelle Appendici A, B e C. Le immagini utilizzate per i test e una nuvola di punti di riferimento possono essere fornite dagli autori su richiesta, al fine di consentire verifiche indipendenti e applicazioni ad hoc del metodo.

4. Conclusioni

Il metodo proposto si è dimostrato efficace nel trasferire informazioni cromatiche provenienti da singole immagini storiche su nuvole di punti 3D moderne, attraverso una procedura ordinata, controllata e condizionatamente riproducibile. Il workflow garantisce che ogni fase, dall'orientamento della nuvola e dalla generazione dell'immagine sintetica, alla selezione dei punti omologhi fino alla *back-projection* del colore, sia tracciabile e guidata dall'utente, riducendo il rischio di errori non controllati e favorendo una piena consapevolezza delle trasformazioni applicate. In questo contesto, la riproducibilità del processo deve essere intesa come ripetibilità del *workflow* a parità di scelte operative dell'utente, data la natura parzialmente interattiva di alcune fasi chiave.

L'integrazione dei moduli di pre-analisi PreCloud e Prelmg consente una valutazione preliminare sia della qualità geometrica della nuvola di punti sia del contenuto informativo dell'immagine storica. Tali analisi permettono di anticipare criticità legate alla risoluzione effettiva, alla ridondanza radiometrica o alla distribuzione spaziale dei dati, contribuendo all'ottimizzazione del *workflow* e a una più corretta interpretazione dei risultati.

Gli errori geometrici introdotti durante la trasformazione dell'immagine storica nella prospettiva moderna possono essere facilmente rilevati mediante la visualizzazione dei residui e tramite indicatori statistici, come l'RMS. Questo approccio permette di individuare eventuali errori sistematici, valutare l'affidabilità locale del risultato e identificare rapidamente le aree che richiedono maggiore attenzione, eventualmente per ripetere la procedura con scelte più mirate. La presenza di soglie statistiche basate sulla distribuzione dei residui consente inoltre una classificazione quantitativa degli errori e una lettura più oggettiva della qualità dell'allineamento. I casi di studio presentati dimostrano la versatilità del metodo anche in condizioni particolarmente sfidanti, caratterizzate dalla disponibilità di una sola immagine per ciascun caso analizzato, mostrando come una procedura di texturizzazione apparentemente semplice possa fornire risultati informativi rilevanti. Nel caso della Basilica di San Francesco, una singola fotografia del 1943 è stata utilizzata per ricostruire e localizzare con precisione informazioni storiche relative alla facciata; nel caso del porto-canale di Cesenatico, una cartolina di fine XIX inizio XX secolo ha consentito di estrarre informazioni metriche sul livello marino storico.

Un terzo caso di studio ha riguardato l'utilizzo di un'immagine termica riferita a un dettaglio architettonico di un edificio storico di Bologna (Corte Isolani), impiegata come sorgente radiometrica alternativa per la texturizzazione della nuvola di punti. Sebbene basato su un *dataset* limitato, questo esempio mostra come il *workflow* possa essere esteso anche a dati non visibili, confermando la flessibilità del metodo rispetto alla natura e alla banda spettrale dell'immagine utilizzata.

Il presente lavoro si inserisce nelle attività del progetto di ricerca libera TEXANI (*TEXTuring Ancient Images*) e descrive l'approccio sviluppato, concepito come procedura operativa e insieme come *framework* flessibile per l'analisi metrica delle immagini storiche, supportato da moduli e codici dedicati per facilitarne l'uso, la comprensione e la riproducibilità.

Il codice e i moduli accessori derivano da esperienze precedenti [Teza et al., 2019; Teza e Pesci, 2025; Pesci et al., 2026]; rispetto a tali lavori, la procedura è stata riorganizzata e semplificata, mantenendone il rigore metrico. Il *workflow* risultante supporta non solo la texturizzazione, ma anche l'analisi quantitativa di dati storici in maniera riproducibile e controllata.

L'obiettivo a lungo termine è sviluppare un software *open-source* che offra funzionalità simili a

quelle dei sistemi commerciali, ma completamente libero e accessibile. La progettazione punta a interfacce grafiche intuitive e operazioni guidate, in modo da rendere la texturizzazione e l'analisi dei dati storici fruibili anche a utenti non specialisti. La futura migrazione verso Python consentirà di combinare performance elevate, flessibilità e gratuità, creando uno strumento modulare e facilmente estendibile, aperto a nuove funzionalità e metodologie avanzate.

Disponibilità di dati e codici

I dati alla base di questo lavoro sono disponibili su richiesta agli autori. I codici in allegato, presentati come Appendice A, B e C, possono essere liberamente utilizzati, con la sola cortesia di citare opportunamente la fonte.

Ringraziamenti

Si ringrazia il Catalogo dei Beni Culturali per la disponibilità online di immagini storiche, che hanno reso possibili le analisi presentate nello studio.

Gli autori ringraziano Marco Olivieri per aver indicato l'importanza storica e funzionale del porto-canale di Cesenatico come caso di studio nello studio delle variazioni del livello del mare. Un ringraziamento anticipato va a Daniele Randazzo, con il quale è in corso una collaborazione finalizzata allo sviluppo di un approccio più completo ed efficiente, volto alla realizzazione di un software gratuito e open-source di livello professionale, dotato di interfaccia grafica interattiva e intuitiva.

Infine, gli autori ringraziano l'INGV per le iniziative progettuali interne di ricerca libera che hanno reso possibile lo sviluppo di questa linea di ricerca, e Alessandro Carosi per l'impegno profuso in questo ambito.

Bibliografia

- Böhler, W., & Marbs, A. (2004). 3D scanning and photogrammetry for heritage recording: a comparison. In Proceedings of the 12th International Conference on Geoinformatics.
- Brunier, G., Fleury, J., Anthony, E. J., Gardel, A., & Dussouillez, P. (2016). Close-range airborne Structure-from-Motion photogrammetry for high-resolution beach morphometric surveys: Examples from an embayed rotating beach. *Geomorphology*, 261, 76-88. <https://doi.org/10.1016/j.geomorph.2016.02.025>
- Castellucci, E. (2010). *Le terme e la piazza sul mare: progetto di riqualificazione del parco di Levante a Cesenatico* [Tesi di laurea specialistica, Università di Bologna].
- Champion, E. (2019). *Virtual Heritage: A Guide*. Ubiquity Press.
- Dostal, C., & Yamafune, K. (2018). Photogrammetric texture mapping: A method for increasing the fidelity of 3D models of cultural heritage materials. *Journal of Archaeological Science: Reports*, 18, 430-436. <https://doi.org/10.1016/j.jasrep.2018.01.024>
- Eltner, A., Kaiser, A., Castillo, C., Rock, G., Neugirg, F., & Abellán, A. (2016). Image-based surface reconstruction in geomorphometry - merits, limits and developments. *Earth Surface Dynamics*, 4, 359-389. <https://doi.org/10.5194/esurf-4-359-2016>
- Gonzalez, R. C., & Woods, R. E. (2018). *Digital image processing* (4th ed.). Pearson.
- Grün, A. (1985). Adaptive least squares correlation: a powerful image matching technique. *South African Journal of Photogrammetry, Remote Sensing and Cartography*, 14(3), 175-187.
- Guidi, G., Remondino, F., Russo, M., Menna, F., Rizzi, A., & Ercolessi, R. (2009). A Multi-Resolution

- 3D Model for the Knowledge of Cultural Heritage Sites: The Castrum Novum. *Journal of Cultural Heritage*, 10, 321–330.
- Haala, N., & Kada, M. (2010). An update on automatic 3D building reconstruction. *ISPRS Journal of Photogrammetry and Remote Sensing*, 65(6), 570–580. <https://doi.org/10.1016/j.isprsjprs.2010.09.006>
- Masacci, D., & De Lucia, S. (1983). *Cesenatico. Immagini della memoria* (2a ristampa). Forlì: Nuova Itefilm.
- Meli, M., Olivieri, M., & Romagnoli, C. (2021). Sea-Level Change along the Emilia-Romagna Coast from Tide Gauge and Satellite Altimetry. *Remote Sensing*, 13(1), 97. <https://www.mdpi.com/2072-4292/13/1/97>
- Pesci, A., Bonali, E., Galli, C., & Boschi, E. (2012). Laser scanning and digital imaging for the investigation of an ancient building: Palazzo d'Accursio study case (Bologna, Italy). *Journal of Cultural Heritage*, 13(2), 215–220. <https://doi.org/10.1016/j.culher.2011.09.004>
- Pesci, A., Teza, G., Bonali, E., Casula, G., & Boschi, E. (2013). A laser scanning-based method for fast estimation of seismic-induced building deformations. *ISPRS Journal of Photogrammetry and Remote Sensing*, 79, 185–198. <https://doi.org/10.1016/j.isprsjprs.2013.02.021>
- Pesci, A., Teza, G., Rossetti, A., & Bondanelli, M. (2026). Texturing Historical Data into 3D Models: San Giorgio in Argenta Parish Church (FE). *Journal of Surveying Engineering*. <https://doi.org/10.1061/JSUED2/SUENG-1653>
- Remondino, F., Rizzi, A., Girardi, S., Benedetti, A., & Barazzetti, L. (2014). 3D surveying and modeling of heritage objects. *Graphical Models*, 76(3), 197–213.
- Snavely, N., Seitz, S., & Szeliski, R. (2006). Photo tourism: exploring image collections in 3D. *ACM Transactions on Graphics (SIGGRAPH)*, 25(3), 835–846. <https://doi.org/10.1145/1141911.1141964>
- Teza, G., & Pesci, A. (2019). Evaluation of the temperature pattern of a complex body from thermal imaging and 3D information: a method and its MATLAB implementation. *Journal Infrared Physics & Technology*, 96, 228–237. <https://doi.org/10.1016/j.infrared.2018.11.029>
- Teza, G., & Pesci, A. (2025). COMAP3/comapRid MATLAB toolbox (general scripts, functions, user's guide and tutorial data). Version 3 (2025). *Mendeley Data*. <https://doi.org/10.17632/5f6ty3zwx2.3>
- Wang, Z., Bovik, A. C., Sheikh, H. R., & Simoncelli, E. P. (2004). Image quality assessment: From error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4), 600–612. <https://doi.org/10.1109/TIP.2003.819861>

Sitografia

- [1] <https://www.agisoftmetashape.com/> (Accesso: 22/06/2026)
- [2] <https://www.weissoft.it/realityscan/> (Accesso: 22/06/2026)
- [3] <https://it.bentley.com/software/itwin-capture-modeler/> (Accesso: 22/06/2026)
- [4] <https://catalogo.cultura.gov.it/detail/PhotographicHeritage/0800265969/> (Accesso: 22/06/2026)
- [5] <https://www.polyworks.com/it-it/> (Accesso: 22/06/2026)
- [6] <https://www.isprambiente.gov.it/it/banche-dati/banche-dati-folder/mare/> (Accesso: 22/06/2026)
- [7] <https://www.venezia.isprambiente.it/meteo-mareographic-network/> (Accesso: 22/06/2026)
- [8] <https://www1.ismar.cnr.it/tematiche-di-ricerca/> (Accesso: 22/06/2026)
- [9] <https://psmsl.org/> (Accesso: 22/06/2026)
- [10] <https://psmsl.org/products/trends/> (Accesso: 22/06/2026)
- [11] <https://www.flir.com/it-it/products/t620/> (Accesso: 22/06/2026)

APPENDICE A

Codice PreCloud

```

clear; clc; close all;

%% --- 1) Caricamento nuvola XYZ(V) ---
[pcfile, pcpath] = uigetfile({'*.txt;*.xyz','File XYZV'}, 'Seleziona la nuvola di punti');
if pcfile==0, error('Nessun file selezionato. Interruzione.');
```

end

```

fullfile_pc = fullfile(pcpath, pcfile);
pc = dlmread(fullfile_pc, ' ');

X = pc(:,1); Y = pc(:,2); Z = pc(:,3); V = pc(:,4);
nPoints = length(X);
disp(['Nuvola caricata: ' num2str(nPoints) ' punti']);

%% --- 2) Normalizzazione intensità per scala di grigi ---
Vn = double(V);
Vn = (Vn - min(Vn)) / (max(Vn) - min(Vn)); % da 0 a 1

% Creiamo un colore RGB in scala di grigi (3 canali uguali)
C = [Vn Vn Vn];

%% --- 3) Visualizzazione iniziale in toni di grigio ---
hFig = figure('Name','Nuvola originale (toni di grigio)','Color','white');
scatter3(X,Y,Z,5,C,'filled');
axis equal; grid on; rotate3d on;
xlabel('X'); ylabel('Y'); zlabel('Z');
title('Ruota per impostare la vista desiderata');
colormap(gray);
colorbar;
view(3);

uiwait(helpdlg('Ruota la nuvola nella vista desiderata e premi OK','Selezione Vista'));

%% --- 4) Acquisizione parametri camera dalla figura corrente ---
ax = gca;
cam_pos = campos(ax); % punto osservazione
cam_tgt = camtarget(ax); % target
cam_up = camup(ax); % verso alto
viewAngle = camva(ax);

disp('Parametri camera acquisiti:');
disp(['Camera Position: ' num2str(cam_pos)]);
disp(['Camera Target: ' num2str(cam_tgt)]);

%% --- 5) Calcolo distanza dei punti dal punto di osservazione ---
XYZ = [X Y Z]';
dist_cam = sqrt(sum(bsxfun(@minus, XYZ, cam_pos).^2,1));

%% --- 6) Visualizzazione colorata per distanza dalla camera ---
figure('Name','Nuvola colorata per distanza dalla camera','Color','white');
scatter3(X,Y,Z,5,dist_cam,'filled');
axis equal; grid on; rotate3d on;
xlabel('X'); ylabel('Y'); zlabel('Z');
title('Colorazione per distanza dalla camera');
colorbar;
colormap(jet);

```


APPENDICE B

Codice PreImg

```
% analyse_effective_resolution.m
% MATLAB 2015 compatible
clear; clc; close all;

% --- 1) Carica immagine ---
[imgfile, imgpath] = uigetfile({'*.jpg;*.png;*.tif;*.tiff;*.bmp','Immagine storica'},
'Seleziona immagine storica');
if imgfile==0, error('Nessuna immagine selezionata'); end
I = imread(fullfile(imgpath,imgfile));

% Normalizza / forzi RGB
if ndims(I)==2
    I = repmat(I,[1 1 3]);
elseif ndims(I)==3 && size(I,3)>3
    I = I(:,:,1:3);
end
I = im2uint8(I); % garantisce uint8

[H,W,~] = size(I);
fprintf('Dimensione nominale immagine: %d x %d (HxW)\n', H, W);
Npix = H*W;

% --- 2) Numero colori unici (RGB combinato) ---
R = uint32(I(:,:,1)); G = uint32(I(:,:,2)); B = uint32(I(:,:,3));
comb = R*65536 + G*256 + B; % mappa unica per colore
combVec = reshape(comb, [], 1);
[uniqColors,~] = unique(combVec);
nColors = numel(uniqColors);
pctColors = nColors / Npix * 100;
fprintf('Colori unici totali: %d (%.3f %% dei pixel)\n', nColors, pctColors);

% --- 3) Percentuale pixel identici ai vicini (4-neighborhood) ---
% Confronta ogni pixel con right/down/left/up (evita bordi duplicati)
P = double(I);
% Compute equality with right neighbor and down neighbor to avoid double count
eqRight = false(H,W);
eqDown = false(H,W);
eqRight(:,1:W-1) = all(I(:,1:W-1,:) == I(:,2:W,:),3);
eqDown(1:H-1,:) = all(I(1:H-1,:,:) == I(2:H,:,:),3);
% union
eqAny = eqRight | eqDown;
pctIdenticalNeighbors = sum(eqAny(:)) / Npix * 100;
fprintf('Percentuale pixel identici a almeno 1 vicino (right/down): %.3f %%\n',
pctIdenticalNeighbors);

% --- 4) Patch variance & entropy (blocchi NxN) ---
blkSizes = [4 8 16 32];
meanVar = zeros(size(blkSizes));
meanEntrop = zeros(size(blkSizes));

for b=1:length(blkSizes)
    bs = blkSizes(b);
    % crop to multiple of bs
    Hc = floor(H/bs)*bs; Wc = floor(W/bs)*bs;
    Icrop = I(1:Hc,1:Wc,:);
    % convert to grayscale for var/entropy
    Ig = rgb2gray(Icrop);
    % reshape into blocks: vectorized approach
    IgBlocks = reshape(Ig, bs, [], Wc/bs);
    % a bit trickier: use blockproc if available, else manual
    % manual: compute variance per block
    varBlocks = zeros(Hc/bs, Wc/bs);
    entBlocks = zeros(Hc/bs, Wc/bs);
    rowIdx = 1;
    for r=1:bs:Hc
        colIdx = 1;
        rr = (r:(r+bs-1));
        for c=1:bs:Wc
```

```

        cc = (c:(c+bs-1));
        patch = double(Ig(rr,cc));
        varBlocks(rowIdx,colIdx) = var(patch(:));
        % entropy estimate (discrete)
        counts = histcounts(patch(:), 0:256);
        probs = counts / sum(counts);
        probs = probs(probs>0);
        entBlocks(rowIdx,colIdx) = -sum(probs .* log2(probs));
        colIdx = colIdx + 1;
    end
    rowIdx = rowIdx + 1;
end
meanVar(b) = mean(varBlocks(:));
meanEntropy(b) = mean(entBlocks(:));
fprintf('Blocchi %dx%d: var_media=%.3f, entropia_media=%.3f bits\n', bs, bs, meanVar(b),
meanEntropy(b));
end

% --- 5) Downsample - Upsample reconstruction test ---
factors = [2 4 8 16]; % provi questi fattori
mseVals = zeros(size(factors));
ssimVals = zeros(size(factors));
doSSIM = exist('ssim','file')==2; % verifica se SSIM è disponibile

for k=1:length(factors)
    f = factors(k);
    % downsample (bicubic), then upsample back
    Idown = imresize(I, 1/f, 'bicubic');
    Irec = imresize(Idown, [H W], 'bicubic');
    % MSE
    diff = double(I) - double(Irec);
    mseVals(k) = mean(diff(:).^2);
    if doSSIM
        try
            ssimVals(k) = ssim(Irec, I);
        catch
            ssimVals(k) = NaN;
        end
    else
        ssimVals(k) = NaN;
    end
    fprintf('Down %xd -> MSE=%.2f, SSIM=%s\n', f, mseVals(k), num2str(ssimVals(k)));
end

% --- 6) Stima "risoluzione effettiva" basata su threshold SSIM/MSE ---
% criterio: troviamo il max fattore f tale che SSIM>=0.99 (se disponibile)
effectiveFactor = 1;
if doSSIM && ~all(isnan(ssimVals))
    idxOk = find(ssimVals >= 0.99);
    if ~isempty(idxOk)
        effectiveFactor = factors(max(idxOk));
    else
        % se SSIM mai >=0.99, scegli fattore min con SSIM >= 0.95
        idxOk2 = find(ssimVals >= 0.95);
        if ~isempty(idxOk2)
            effectiveFactor = factors(max(idxOk2));
        else
            effectiveFactor = 1;
        end
    end
else
    % fallback su MSE: scegli fattore con MSE < (0.01 * 255^2) ~ low error
    thr = 0.01 * 255^2;
    idxOk = find(mseVals <= thr);
    if ~isempty(idxOk)
        effectiveFactor = factors(max(idxOk));
    else
        effectiveFactor = 1;
    end
end

effW = round(W / effectiveFactor);
effH = round(H / effectiveFactor);

```

```

fprintf('Stima risoluzione effettiva (baseline): %d x %d (fattore %d)\n', effW, effH,
effectiveFactor);

% --- 7) Output diagnostici / figure ---
figure('Name','Diagnostica colori e ridondanza');
subplot(2,2,1);
imshow(I); title(sprintf('Originale %dx%d', H, W));
subplot(2,2,2);
histogram(combVec, 256); title('Istogramma colori (indice combinato)'); % grafico grosso ma
informativo
subplot(2,2,3);
imagesc(eqAny); axis image; title('Pixel con vicino identico (right/down)');
subplot(2,2,4);
plot(blkSizes, meanEntropy, '-o'); xlabel('Block size'); ylabel('Entropy (bits)');
title('Entropy media per block size');

figure('Name','Downsample test');
subplot(2,1,1); plot(factors, mseVals, '-o'); xlabel('Down factor'); ylabel('MSE');
set(gca,'XTick',factors);
subplot(2,1,2); plot(factors, ssimVals, '-o'); xlabel('Down factor'); ylabel('SSIM');
set(gca,'XTick',factors);

% --- 8) Suggestimenti interpretativi (stampati) ---
fprintf('\n--- Suggestimenti interpretativi ---\n');
fprintf('- Se pctColors < 100%% (es. <5%%) allora la palette è molto ridotta: possibile
posterizzazione.\n');
fprintf('- Se pctIdenticalNeighbors elevata (es. >30%%) -> molte aree piatte senza
dettaglio.\n');
fprintf('- Entropia media bassa su blocchi grandi (es. <1 bit per 16x16) -> basso contenuto
informativo locale.\n');
if doSSIM
    fprintf('- Downsample test (SSIM): fattore massimo che mantiene SSIM>=0.99 -> consigliato
come fattore di sovracampionamento.\n');
else
    fprintf('- Downsample test (MSE fallback): usa criterio mse<thr per stimare
sovracampionamento.\n');
end
fprintf('- Stima risoluzione effettiva: ~ %d x %d (da considerare come ordine di
grandezza).\n', effH, effW);
fprintf('-----\n');

% salva risultati in una struct
results.nColors = nColors;
results.pctColors = pctColors;
results.pctIdenticalNeighbors = pctIdenticalNeighbors;
results.meanVar = meanVar;
results.meanEntropy = meanEntropy;
results.factors = factors;
results.mseVals = mseVals;
results.ssimVals = ssimVals;
results.effectiveFactor = effectiveFactor;
results.effectiveResolution = [effH effW];

save('effective_resolution_results.mat','results');
```


APPENDICE C

Codice Historica

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% -----
% SIMPLIFIED POINT CLOUD TO IMAGE - BLOCCO 1
% Setup ambiente, scelta file, import, normalizzazione + salvataggio file normalizzato
% -----
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

clear; clc; close all;

% --- 1) Scegli cartella di lavoro ---
disp('Scegli una cartella di lavoro...');
workdir = uigetdir(pwd, 'Seleziona cartella di lavoro');
if workdir == 0
    error('Nessuna cartella selezionata. Interruzione.');
```

end

```

cd(workdir);
disp(['Cartella selezionata: ' workdir]);

% --- 2) Creazione LOG ---
logfile = 'log.txt';
fid = fopen(logfile, 'a');
fprintf(fid, '\n-----\nNuova sessione: %s\n', datestr(now));
fclose(fid);

% --- 3) Seleziona file nuvola XYZV ---
[pcfile, pcpath] = uigetfile({'*.txt;*.xyz', 'File XYZV'}, 'Seleziona la nuvola di punti');
if pcfile == 0
    error('Nessun file selezionato. Interruzione.');
```

end

```

fullfile_pc = fullfile(pcpath, pcfile);
disp(['File selezionato: ' fullfile_pc]);

% --- 4) Caricamento del file con auto-separatore ---
txt = fileread(fullfile_pc);

if ~isempty(strfind(txt, ','))
    sep = ',';
elseif ~isempty(strfind(txt, ';'))
    sep = ';';
elseif ~isempty(strfind(txt, sprintf('\t')))
    sep = sprintf('\t'); % TAB
else
    sep = ' '; % Default spazio
end
clear txt;

pc = dlmread(fullfile_pc, sep);
if size(pc, 2) < 4
    error('Errore: Il file deve contenere almeno 4 colonne: X Y Z V');
```

end

```

X = pc(:, 1); Y = pc(:, 2); Z = pc(:, 3); V = pc(:, 4);

fprintf('\nNuvola caricata: %d punti\n', length(X));
disp('Prime righe:');
disp(pc(1:min(5, end), :));

% --- 5) Scelta normalizzazione V (sempre proposta) ---
minV = min(V); maxV = max(V);
disp(['Range attuale di V: ' num2str(minV) ' -> ' num2str(maxV)]);

scelta = questdlg('Normalizzare V in 0-255?', 'Normalizzazione V', ...
    'Sì (uint8)', 'Sì (double)', 'No', 'Sì (uint8)');
```

switch scelta

```

    case 'Sì (uint8)'
        V = uint8(255 * (V - minV) / (maxV - minV));
        norm_type = 'uint8';
```

```

        case 'Si (double)'
            V = 255 * (V - minV) / (maxV - minV);
            norm_type = 'double';
        otherwise
            norm_type = 'no';
    end

% --- Salvataggio file normalizzato ---
normalized_file = ['normalized-' pcfile];
pc_norm = [X Y Z double(V)];
dlmwrite(normalized_file, pc_norm, 'delimiter', ' ');

disp(['Salvato file normalizzato: ' normalized_file]);

% Aggiornamento log
fid = fopen(logfile, 'a');
fprintf(fid, 'File: %s\n', pcfile);
fprintf(fid, 'Normalizzazione: %s\n', norm_type);
fprintf(fid, 'Range finale V: %d -> %d\n', min(V), max(V));
fprintf(fid, 'File normalizzato: %s\n', normalized_file);
fclose(fid);

disp('--- BLOCCO 1 COMPLETATO ? ---');

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% -----
% BLOCCO 2 - Sottocampionamento + Visualizzazione 3D
%% -----
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Chiedi sottocampionamento in percentuale
prompt = {'Percentuale punti da mantenere (0-100):'};
dlgtitle = 'Sottocampionamento';
answer = inputdlg(prompt, dlgtitle, 1, {'100'});

if isempty(answer)
    error('Operazione annullata dall''utente.');
```

```
end
```

```

S = str2double(answer{1});
if isnan(S) || S < 0 || S > 100
    error('Percentuale non valida.');
```

```
end
```

```

% Calcolo indice per sottocampionamento sequenziale
if S == 100
    idx = 1:length(X);
else
    passo = round(100 / S);
    idx = 1:passo:length(X);
end
```

```

X_s = X(idx);
Y_s = Y(idx);
Z_s = Z(idx);
V_s = V(idx);
```

```

% ? Salviamo XYZ per blocchi successivi
XYZ = [X_s Y_s Z_s]';
```

```

% Salvataggio file sottocampionato
sampled_file = sprintf('sampled-%d.txt', S);
pc_sampled = [X_s Y_s Z_s double(V_s)];
dlmwrite(sampled_file, pc_sampled, 'delimiter', ' ');
```

```

disp(['File sottocampionato salvato: ' sampled_file]);
disp(['Punti mantenuti: ' num2str(length(idx))]);
```

```

% Scrittura nel log
fid = fopen(logfile, 'a');
fprintf(fid, 'Sottocampionamento: %d%\n', S);
fprintf(fid, 'Punti finali: %d\n', length(idx));
fprintf(fid, 'File sottocampionato: %s\n', sampled_file);
```

```

fclose(fid);

% --- Visualizzazione 3D ---
figure('Name','Naviga la Nuvola','Color','white');
scatter3(X_s, Y_s, Z_s, 3, double(V_s), 'filled');
title('Naviga la nuvola, poi premi OK quando la vista ti piace');
xlabel('X (m)'); ylabel('Y (m)'); zlabel('Z (m)');
axis equal; grid on; colormap gray;
rotate3d on;
colorbar;
view(3);

h = helpdlg('Muovi la vista (rotate, pan, zoom). Chiudi la finestra per continuare.','Vista Nuvola');
uiwait(h);

disp('--- BLOCCO 2 COMPLETATO ? ---');

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% -----
% BLOCCO 2bis - Modifica numerica manuale della vista (nuova aggiunta)
% Inserita subito dopo la rotazione manuale: permette di correggere Az/Elev/CameraPos/Target/Up/VA
%% -----
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Prendiamo l'axes corrente
ax_edit = gca;

% Lettura valori correnti
curView = get(ax_edit,'View');           % [az el]
curCamPos = campos(ax_edit);             % 1x3
curCamTgt = camtarget(ax_edit);          % 1x3
curCamUp   = camup(ax_edit);             % 1x3
curCamVA   = camva(ax_edit);             % scalar

% Messaggio riassuntivo
msg = sprintf(['Vista corrente:\nAzimuth = %.3f\nElevation = %.3f\n' ...
              'CameraPosition = [%.3f %.3f %.3f]\nCameraTarget = [%.3f %.3f %.3f]\n' ...
              'CameraUp = [%.3f %.3f %.3f]\nViewAngle = %.3f\n\n' ...
              'Vuoi modificare manualmente questi valori?'], ...
              curView(1), curView(2), ...
              curCamPos(1), curCamPos(2), curCamPos(3), ...
              curCamTgt(1), curCamTgt(2), curCamTgt(3), ...
              curCamUp(1), curCamUp(2), curCamUp(3), curCamVA);
choice = questdlg(msg, 'Modifica vista manuale', 'Si','No','No');

if strcmp(choice,'Si')
    % Preparamo default text per inputdlg
    def = {num2str(curView(1)), num2str(curView(2)), ...
           num2str(curCamPos(1)), num2str(curCamPos(2)), num2str(curCamPos(3)), ...
           num2str(curCamTgt(1)), num2str(curCamTgt(2)), num2str(curCamTgt(3)), ...
           num2str(curCamUp(1)), num2str(curCamUp(2)), num2str(curCamUp(3)), ...
           num2str(curCamVA)};
    prompt = {'Azimuth (deg)', 'Elevation (deg)', ...
              'CameraPos X', 'CameraPos Y', 'CameraPos Z', ...
              'CameraTarget X', 'CameraTarget Y', 'CameraTarget Z', ...
              'CameraUp X', 'CameraUp Y', 'CameraUp Z', ...
              'ViewAngle (deg)'};
    dlgtitle = 'Inserisci nuovi valori (lascia vuoto per mantenere il valore corrente)';
    answer = inputdlg(prompt, dlgtitle, 1, def);

    if ~isempty(answer)
        % Sostituiamo solo i campi non vuoti (compatibile R2015)
        tmp = answer; % cell array stringhe
        % Azimuth / Elevation
        newAz = str2double(tmp{1});
        newEl = str2double(tmp{2});
        if ~isnan(newAz), curView(1) = newAz; end
        if ~isnan(newEl), curView(2) = newEl; end
        set(ax_edit,'View',curView);
    end
end

```

```

        % CameraPosition
        newCPx = str2double(tmp{3}); newCPy = str2double(tmp{4}); newCPz =
str2double(tmp{5});
        if ~(isnan(newCPx) || isnan(newCPy) || isnan(newCPz))
            campos(ax_edit, [newCPx newCPy newCPz]);
        end

        % CameraTarget
        newCTx = str2double(tmp{6}); newCTy = str2double(tmp{7}); newCTz =
str2double(tmp{8});
        if ~(isnan(newCTx) || isnan(newCTy) || isnan(newCTz))
            camtarget(ax_edit, [newCTx newCTy newCTz]);
        end

        % CameraUp
        newUPx = str2double(tmp{9}); newUPy = str2double(tmp{10}); newUPz =
str2double(tmp{11});
        if ~(isnan(newUPx) || isnan(newUPy) || isnan(newUPz))
            camup(ax_edit, [newUPx newUPy newUPz]);
        end

        % ViewAngle
        newVA = str2double(tmp{12});
        if ~isnan(newVA)
            camva(ax_edit, newVA);
        end

        % Aggiorniamo le visualizzazioni
        drawnow;
        fprintf('Vista aggiornata manualmente.\n');
    else
        fprintf('Modifica vista annullata dall\'utente.\n');
    end
end

% Fine BLOCCO 2bis

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% BLOCCO 3 - Acquisizione Parametri di Vista (Camera)
%% -----
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

disp('--- BLOCCO 3: Acquisizione parametri camera ---');

% Controllo che esista una figura attiva
fig = gcf;
ax = gca;

% Estrazione parametri camera da Axes
cam_pos = ax.CameraPosition;
cam_tgt = ax.CameraTarget;
cam_up = ax.CameraUpVector;
cam_va = ax.CameraViewAngle;

disp('Parametri camera estratti:');
disp(['CameraPosition: ' num2str(cam_pos)]);
disp(['CameraTarget: ' num2str(cam_tgt)]);
disp(['CameraUpVector: ' num2str(cam_up)]);
disp(['ViewAngle: ' num2str(cam_va)]);

% Salvataggio in file camera.txt
camera_file = fullfile(workdir, 'camera.txt');
fid = fopen(camera_file, 'w');
fprintf(fid, 'CameraPosition %f %f %f\n', cam_pos);
fprintf(fid, 'CameraTarget %f %f %f\n', cam_tgt);
fprintf(fid, 'CameraUpVector %f %f %f\n', cam_up);
fprintf(fid, 'CameraViewAngle %f\n', cam_va);
fclose(fid);

% Append log
fid = fopen(logfile, 'a');
fprintf(fid, 'CameraPosition: %f %f %f\n', cam_pos);
fprintf(fid, 'CameraTarget: %f %f %f\n', cam_tgt);

```

```

fprintf(fid,'CameraUpVector: %f %f %f\n', cam_up);
fprintf(fid,'CameraViewAngle: %f\n', cam_va);
fclose(fid);

disp('--- Parametri camera salvati ? ---');
disp(['File: ' camera_file]);
disp('Procedi con il BLOCCO 4 per la proiezione ?');

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% BLOCCO 4 – Proiezione prospettica corretta (senza ribaltamenti)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

disp('--- BLOCCO 4: Proiezione sulla camera (fixed) ---');

% Rilettura parametri camera dalla figura
C = campos(ax)';      % 3x1
T = camtarget(ax)';
U = camup(ax)';
viewAngle = camva(ax);

% Direzione "in avanti" (forward)
zc = (T - C);
zc = zc / norm(zc);

% Correggiamo U: lo proiettiamo sul piano ortogonale a zc (evita ribaltamenti)
U = U - dot(U, zc) * zc;
U = U / norm(U);

% Assi ortonormali camera (convenzione destrorsa)
xc = cross(zc, U); xc = xc / norm(xc); % destra
yc = cross(xc, zc); yc = yc / norm(yc); % su

% Matrice di rotazione mondo -> camera (righe = assi della camera)
R = [xc'; yc'; zc']; % 3x3

% Costruiamo la matrice punti (3xN) dai punti sottocampionati
XYZ = [X_s'; Y_s'; Z_s']; % 3xN

% Sottrai centro camera (compatibile anche con R2015)
XYZ_shift = bsxfun(@minus, XYZ, C); % 3xN

% Trasformazione nel sistema camera
XYZc = R * XYZ_shift; % 3xN

% Protezione punti troppo vicini (evita divisioni per zero)
epsZ = 1e-9;
XYZc(3, XYZc(3,:) < epsZ) = epsZ;

% Calcolo focale da view angle (normalizzato)
f = 0.5 / tan((viewAngle*pi/180)/2);

% Proiezione prospettica
u = f * (XYZc(1,:) ./ XYZc(3,:));
v = f * (XYZc(2,:) ./ XYZc(3,:));

% Correzione orientamento verticale se serve (se l'immagine risultasse
capovolta)
% v = -v; % <--- decommenta se ti serve invertire verticale

disp('Proiezione completata ? (corretta)');

% LOG
fid = fopen(logfile,'a');
fprintf(fid,'Proiezione prospettica corretta\n');
fprintf(fid,'Camera Position: %.3f %.3f %.3f\n', C);
fprintf(fid,'Camera Target: %.3f %.3f %.3f\n', T);
fprintf(fid,'Camera Up: %.3f %.3f %.3f\n', U);
fprintf(fid,'ViewAngle: %.3f\n', viewAngle);
fclose(fid);

disp('Parametri camera salvati nel log ?');

```

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% BLOCCO 3bis – Visualizzazione top-view della camera (DOPO BLOCCO 4)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

figure('Name','Vista Camera dall''alto'); hold on; grid on;
title('Top View Camera & Point Cloud');
xlabel('X (m)'); ylabel('Y (m)'); axis equal;

% Nuvola nel piano XY (mostra i punti sottocampionati per coerenza)
plot(X_s, Y_s, '.', 'Color', [0.7 0.7 0.7], 'MarkerSize', 2);

% Punto di osservazione
plot(C(1), C(2), 'ro', 'MarkerSize', 14, 'MarkerFaceColor', 'r');

% Direzione camera (forward) dal BLOCCO 4
cam_dir = zc;

% Proiezione XY
cam_dir_xy = cam_dir;
cam_dir_xy(3) = 0;
if norm(cam_dir_xy) < 1e-9
    cam_dir_xy = [1;0;0];
else
    cam_dir_xy = cam_dir_xy / norm(cam_dir_xy);
end

% Vettore molto più lungo e visibile
bigScale = max([range(X_s) range(Y_s)]);
if bigScale == 0
    bigScale = 1;
end
bigScale = bigScale * 0.6;

quiver(C(1), C(2), cam_dir_xy(1)*bigScale, cam_dir_xy(2)*bigScale, ...
    'Color','r','LineWidth',3,'MaxHeadSize',4);

legend('Point Cloud','Camera Position','View Direction');
hold off;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% -----
% BLOCCO 5 – Rasterizzazione e creazione immagine TIFF
%% -----
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

disp('--- BLOCCO 5: Creazione immagine TIFF ---');

% Richiesta risoluzione immagine (pixel)
prompt = {'Larghezza immagine (pixel):', 'Altezza immagine (pixel):'};
dlg_title = 'Risoluzione immagine';
dims = [1 40];
definput = {'1000','800'};
answer = inputdlg(prompt, dlg_title, dims, definput);

if isempty(answer)
    error('Nessuna risoluzione immagine selezionata. Interruzione.');
```

```

for i = 1:nPoints
    xpix = u_pix(i);
    ypix = v_pix(i);

    if xpix < 1 || xpix > imgW || ypix < 1 || ypix > imgH
        continue;
    end

    zval = XYZc(3,i);

    if isnan(Zbuf(ypix, xpix)) || (zval < Zbuf(ypix, xpix))
        Zbuf(ypix, xpix) = zval;
        Vmap(ypix, xpix) = V_s(i);    % <-- usa V_s (sottocampionati)
    end
end

% Salvataggio TIFF
imgname = 'render.tif';
imwrite(Vmap, imgname);
disp(['? Immagine salvata: ' imgname]);

% LOG
fid = fopen(logfile, 'a');
fprintf(fid, 'Immagine creata: %s\n', imgname);
fprintf(fid, 'Dimensione: %d x %d pixel\n', imgW, imgH);
fclose(fid);

disp('--- BLOCCO 5 COMPLETATO ? ---');

... (rest of your original script remains unchanged) ...

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% -----
% BLOCCO 6 - Caricamento immagini: Render & Storica (SAFE LOAD)
%% -----
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

disp('--- BLOCCO 6: Caricamento immagini (safe load) ---');

% ----- 6.1 Seleziona immagine storica -----
[historic_file, img_path] = uigetfile({'*.jpg;*.png;*.tif;*.tiff;*.bmp', 'Immagini'}, ...
    'Seleziona l''immagine storica da allineare');
if historic_file == 0
    error('Nessuna immagine storica selezionata.');
```

end

```

historic_fullpath = fullfile(img_path, historic_file);
fprintf('Storica: %s\n', historic_fullpath);

% ----- 6.2 Carica render (da file creato precedentemente) -----
render_fullpath = fullfile(pwd, 'render.tif');
if ~exist(render_fullpath, 'file')
    error('Render file non trovato: %s', render_fullpath);
end
fprintf('Render: %s (carico...) \n', render_fullpath);

% ----- 6.3 Safe read helper inline (no external functions) -----
% Carica immagine e gestisce multi-page, multi-band, 16-bit, grayscale -> RGB ecc.

% --- Carica render (potrebbe essere multi-page/multi-band) ---
infoR = imfinfo(render_fullpath);
if numel(infoR) > 1
    fprintf('?? Render TIFF multipagina: prendo la prima pagina.\n');
    render_img = imread(render_fullpath, 1);
else
    render_img = imread(render_fullpath);
end

% Se render ha >3 canali prendi solo il primo (ma poi lo teniamo in grayscale)
if ndims(render_img) == 3 && size(render_img, 3) > 1
```

```

    % Se è stato salvato RGB-like (non previsto) -> converti in grayscale
    fprintf('?? Render multi-canale rilevato (%d canali) -> converto in grayscale
mantenendo primo canale.\n', size(render_img,3));
    render_img = render_img(:, :, 1); % opzione A: keep single channel
end

% Normalizza 16-bit -> 8-bit se necessario
if isa(render_img, 'uint16') || max(render_img(:)) > 255
    fprintf('?? Render 16-bit rilevato -> normalizzo verso 8-bit.\n');
    render_img = uint8(double(render_img) / double(max(render_img(:))) * 255);
end

% Forza tipo uint8 per sicurezza
if ~isa(render_img, 'uint8')
    render_img = uint8(render_img);
end

% ----- Carica storica (safe) -----
infoH = imfinfo(historic_fullpath);
if numel(infoH) > 1
    fprintf('?? Storica TIFF multipagina: prendo la prima pagina.\n');
    historic_img = imread(historic_fullpath, 1);
else
    historic_img = imread(historic_fullpath);
end

% Se la storica ha più di 3 canali, prendi i primi 3 (assumendoli R,G,B)
if ndims(historic_img) == 3 && size(historic_img,3) > 3
    fprintf('?? Storica con %d canali -> uso i primi 3 come RGB.\n',
size(historic_img,3));
    historic_img = historic_img(:, :, 1:3);
end

% Normalizza 16-bit -> 8-bit se necessario
if isa(historic_img, 'uint16') || max(historic_img(:)) > 255
    fprintf('?? Storica 16-bit rilevata -> normalizzo verso 8-bit.\n');
    historic_img = uint8(double(historic_img) / double(max(historic_img(:))) * 255);
end

% Se storica è grayscale (2D) -> replicala su 3 canali (RGB) [utente ha scelto "sì"]
if ndims(historic_img) == 2 || (ndims(historic_img)==3 && size(historic_img,3)==1)
    fprintf('Nota: immagine storica grayscale -> conversione automatica in RGB.\n');
    historic_img = repmat(uint8(historic_img), [1 1 3]);
end

% Se storica è RGB già in uint8 mantieni com'è
if ~isa(historic_img, 'uint8')
    historic_img = uint8(historic_img);
end

% ----- Controllo proporzioni e warn (utente ha scelto "sì") -----
ar_render = size(render_img,2) / size(render_img,1); % W/H
ar_hist = size(historic_img,2) / size(historic_img,1);

% soglia differenza relativa (es. 5%)
rel_diff = abs(ar_render - ar_hist) / max(ar_render, ar_hist);

if rel_diff > 0.05
    % warn e chiedi se ridimensionare la storica alla dimensione del render
    choice = questdlg(sprintf('Attenzione: rapporto d'aspetto differente.\n' ...
'Render: %d x %d (AR=%3f)\nStorica: %d x %d (AR=%3f)\n\n' ...
'Vuoi ridimensionare la storica alla stessa dimensione del render?', ...
size(render_img,2), size(render_img,1), ar_render, ...
size(historic_img,2), size(historic_img,1), ar_hist), ...
'Differenza aspect ratio', 'Si', 'No', 'No');
    if strcmp(choice, 'Si')
        fprintf('Ridimensiono storica a [%d x %d] per avere stessa risoluzione del
render.\n', size(render_img,1), size(render_img,2));
        historic_img = imresize(historic_img, [size(render_img,1), size(render_img,2)]);
    else
        fprintf('Mantengo dimensione storica originale.\n');
    end
end
end

```

```

% ----- Mostra informazioni e preview sicura -----
fprintf('Dimensioni render: %d x %d x %d\n', size(render_img,2),
size(render_img,1), ndims(render_img)==3 && size(render_img,3) || 1);
fprintf('Dimensioni storica: %d x %d x %d\n', size(historic_img,2),
size(historic_img,1), size(historic_img,3));

% Visualizzazione sicura (imshow gestisce MxN e MxNx3)
figure('Name','Render vs Storica (SAFE)');
subplot(1,2,1);
imshow(render_img); title('Render (grayscale)');
subplot(1,2,2);
imshow(historic_img); title('Storica (RGB)');
uiwait(msgbox('Guarda le immagini. Ora selezionerai punti corrispondenti.'));

% Log
fid = fopen(logfile,'a');
fprintf(fid,'Immagine storica: %s\n', historic_fullpath);
fprintf(fid,'Render: %s\n', render_fullpath);
fclose(fid);

disp('--- BLOCCO 6 COMPLETATO ? ---');

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% -----
% BLOCCO 7 - Selezione punti omologhi
%% -----
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

disp('--- BLOCCO 7: Selezione punti omologhi ---');
uiwait(msgbox('Seleziona almeno 3 punti comuni tra le due immagini (Render ?
Storica)'));

[pts_render, pts_hist] = cpselect(render_img, historic_img, 'Wait', true);

if size(pts_render,1) < 3
    error('Servono almeno 3 punti!');
end

save('control_points.mat','pts_render','pts_hist');

% LOG
fid = fopen(logfile,'a');
fprintf(fid,'Coppie punti selezionati: %d\n', size(pts_render,1));
fclose(fid);

disp('--- BLOCCO 7 COMPLETATO ? ---');

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% -----
% BLOCCO 8 - Trasformazione affine + Warping (fixed)
%% -----
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

disp('--- BLOCCO 8: Registrazione affine (fixed) ---');

% Calcolo trasformazione affine dai punti storici alla vista renderizzata
tform = fitgeotrans(pts_hist, pts_render, 'affine');

% Warping immagine storica nella stessa scala e dimensione della render_img
Rrender = imref2d(size(render_img));
warped_hist = imwarp(historic_img, tform, 'OutputView', Rrender);

% Visualizzazione e overlay diagnostico
figure('Name','Warping Storica ? Render');
subplot(1,3,1); imshow(historic_img); title('Storica Originale');
subplot(1,3,2); imshow(warped_hist); title('Storica Registrata');
subplot(1,3,3); imshowpair(render_img, warped_hist, 'blend'); title('Overlay Render vs
Warping');

```

```

% Salvataggio su disco
imwrite(warped_hist, 'warped_storica.tif');
disp('? Storica registrata salvata: warped_storica.tif');

fid = fopen(logfile,'a');
fprintf(fid,'Trasformazione affine completata\n');
fclose(fid);

disp('--- BLOCCO 8 COMPLETATO ? ---');

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% BLOCCO 8bis - Calcolo errore di trasformazione (in pixel)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

disp('--- BLOCCO 8bis: Calcolo errore residuo (pixel) ---');

% Applichiamo la trasformazione ai punti storici originali
pts_hist_transformed = transformPointsForward(tform, pts_hist);

% Calcolo differenze (residui) in pixel
difs = pts_hist_transformed - pts_render;
dists = sqrt(sum(difs.^2,2)); % distanza euclidea per punto

% Statistiche errore
mean_err = mean(dists);
rms_err = sqrt(mean(dists.^2));
max_err = max(dists);
min_err = min(dists);

% Soglia per identificazione residui elevati
thr = mean_err + 2*std(dists);

idx_high = dists > thr;
idx_low = ~idx_high;

% Visualizza risultati
fprintf('\nErrore di trasformazione (in pixel):\n');
fprintf(' Media: %.2f px\n', mean_err);
fprintf(' RMS: %.2f px\n', rms_err);
fprintf(' Max: %.2f px\n', max_err);
fprintf(' Min: %.2f px\n', min_err);

% Salvataggio nel log
fid = fopen(logfile,'a');
fprintf(fid,'Errore trasformazione (px): media=%.2f, RMS=%.2f, max=%.2f, min=%.2f\n', ...
mean_err, rms_err, max_err, min_err);
fclose(fid);

% Visualizzazione diagnostica (opzionale)
figure('Name','Residui punti omologhi');
imshowpair(render_img, warped_hist, 'blend'); hold on;

% =====
% PUNTI OMOGHI (AGGIUNTO)
% =====

% punti buoni (VERDI)
plot(pts_render(idx_low,1), pts_render(idx_low,2), 'go', ...
'MarkerSize', 8, 'LineWidth', 1.5);

% punti fuori soglia (ROSSI)
plot(pts_render(idx_high,1), pts_render(idx_high,2), 'ro', ...
'MarkerSize', 8, 'LineWidth', 1.5);

% residui bassi (verde)
for i = find(idx_low)'
plot([pts_hist_transformed(i,1), pts_render(i,1)], ...
[pts_hist_transformed(i,2), pts_render(i,2)], ...
'g-', 'LineWidth', 1.2);
end

% residui alti (rosso)
for i = find(idx_high)'

```

```

plot([pts_hist_transformed(i,1), pts_render(i,1)], ...
[pts_hist_transformed(i,2), pts_render(i,2)], ...
'r-', 'LineWidth', 1.5);
end

title(sprintf('Residui omologhi (RMS = %.2f px)', rms_err));
legend('Punti buoni', 'Punti outlier', 'Residui', 'Location', 'best');
hold off;

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

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% -----
% BLOCCO 9 – Texturing storico sulla nuvola: Backprojection (robusto)
% -----
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

disp('--- BLOCCO 9: Backprojection colori storici alla nuvola ---');

% Controllo esistenza immagine warped
if ~exist('warped_hist','var')
    error('warped_hist non trovato. Esegui BLOCCO 8 prima di BLOCCO 9.');
```

```

end

% Converti grayscale->RGB se necessario
if size(warped_hist,3) == 1
    warped_hist = repmat(warped_hist, [1 1 3]);
end

warpR = double(warped_hist(:,:,1));
warpG = double(warped_hist(:,:,2));
warpB = double(warped_hist(:,:,3));

H = size(warpR,1);
W = size(warpR,2);

% Numero punti da trattare = punti sottocampionati
nPoints = length(X_s);

% Verifica/ricostruzione u_pix,v_pix coerenti con i punti
needRecompute = false;
if ~(exist('u_pix','var') && exist('v_pix','var') && length(u_pix)==nPoints && length(v_
pix)==nPoints)
    needRecompute = true;
end

if needRecompute
    % se abbiamo u,v calcolati da XYZc li usiamo per ricostruire pixel
    if exist('u','var') && exist('v','var') && length(u)==size(XYZc,2)
        % imgW/imgH dovrebbero esistere (scelti in BLOCCO 5)
        if ~(exist('imgW','var') && exist('imgH','var'))
            imgW = size(render_img,2);
            imgH = size(render_img,1);
        end
        u_pix = round((u - min(u)) / (max(u) - min(u)) * (imgW - 1)) + 1;
        v_pix = round((v - min(v)) / (max(v) - min(v)) * (imgH - 1)) + 1;
        % Inverti Y come fatto in BLOCCO 5
        v_pix = imgH - v_pix + 1;
        % Se XYZc length matches nPoints then ok; altrimenti riduciamo
        if length(u_pix) ~= nPoints
            % ridimensioniamo la lunghezza a nPoints se possibile (primo nPoints)
            minN = min([nPoints, length(u_pix)]);
            u_pix = u_pix(1:minN);
            v_pix = v_pix(1:minN);
            nPoints = minN;
            fprintf('Nota: u_pix/v_pix ricomputati e adattati a nPoints=%d\n', nPoints);
        else
            fprintf('Nota: u_pix/v_pix ricomputati correttamente (n=%d)\n', nPoints);
        end
    else
end
else

```

```

        error('Non posso ricostruire u_pix/v_pix: mancano u,v coerenti o
dimensioni mismatch.');
```

```

    end
end

% Clamp e round sugli indici (sicurezza)
u_pix = round(u_pix);
v_pix = round(v_pix);
u_pix = min(max(u_pix,1), W);
v_pix = min(max(v_pix,1), H);

% Preallocazione colori (per i punti sottocampionati)
colorR = zeros(nPoints,1);
colorG = zeros(nPoints,1);
colorB = zeros(nPoints,1);

countInside = 0;
for i = 1:nPoints
    xpix = u_pix(i);
    ypix = v_pix(i);
    if xpix >= 1 && xpix <= W && ypix >= 1 && ypix <= H
        colorR(i) = warpR(ypix,xpix);
        colorG(i) = warpG(ypix,xpix);
        colorB(i) = warpB(ypix,xpix);
        countInside = countInside + 1;
    else
        colorR(i) = 0;
        colorG(i) = 0;
        colorB(i) = 0;
    end
end

disp(['? Punti mappati correttamente: ' num2str(countInside) '/' num2str(nPoints)]);

% --- Salvataggio su PLY con coordinate REALI (usiamo X_s,Y_s,Z_s corrispondenti) ---
plyname = 'historical_textured.ply';
fid = fopen(plyname,'w');

fprintf(fid,'ply\nformat ascii 1.0\nelement vertex %d\n', nPoints);
fprintf(fid,'property float x\nproperty float y\nproperty float z\n');
fprintf(fid,'property uchar red\nproperty uchar green\nproperty uchar blue\n');
fprintf(fid,'end_header\n');

for i = 1:nPoints
    fprintf(fid,'%6f %6f %6f %d %d %d\n', ...
        X_s(i), Y_s(i), Z_s(i), ...
        uint8(colorR(i)), uint8(colorG(i)), uint8(colorB(i)));
end
fclose(fid);

disp(['? Nuvola texturizzata salvata: ' plyname]);

fid = fopen(logfile,'a');
fprintf(fid,'Backprojection completata: %s\n', plyname);
fclose(fid);

% --- Calcolo luminanza (XYZI) ---
lum = 0.2989 * double(colorR) + ...
      0.5870 * double(colorG) + ...
      0.1140 * double(colorB);
lum = uint8(round(lum));

% =====
% Salvataggio TXT XYZ + RGB
% =====
txtname_rgb = 'historical_textured_rgb.txt';
fidtxt_rgb = fopen(txtname_rgb,'w');

for i = 1:nPoints
    fprintf(fidtxt_rgb,'%6f %6f %6f %d %d %d\n', ...
        X_s(i), Y_s(i), Z_s(i), ...
        uint8(colorR(i)), uint8(colorG(i)), uint8(colorB(i)));
end

```

```

fclose(fidtxt_rgb);

% =====
% Salvataggio TXT XYZ + luminanza (XYZI)
% =====
txtname_lum = 'historical_textured_luminance.txt';
fidtxt_lum = fopen(txtname_lum, 'w');

for i = 1:nPoints
    fprintf(fidtxt_lum, '%.6f %.6f %.6f %d\n', ...
        X_s(i), Y_s(i), Z_s(i), lum(i));
end

fclose(fidtxt_lum);

disp('--- BLOCCO 9 COMPLETATO ? --- ?');
uiwait(msgbox('? Lavoro completato: texturing storico riuscito!'));

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% -----
% BLOCCO 10 – Visualizzazione confronto nuvole (fixed)
% -----
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

disp('--- BLOCCO 10: Visualizzazione del confronto ---');

% === Figura confronto ===
figure('Name', 'Confronto Nuvole
3D', 'Color', 'white', 'Units', 'normalized', 'Position', [0.05 0.05 0.9 0.8]);

% --- Subplot 1: Nuvola moderna (render grayscale)
subplot(1,2,1);
scatter3(X_s, Y_s, Z_s, 3, double(V_s), 'filled');
title('Nuvola Moderna (render)');
xlabel('X (m)'); ylabel('Y (m)'); zlabel('Z (m)');
axis equal; grid on; colormap gray;
view(3);

% --- Subplot 2: Nuvola Storica (colori back-projected sui punti sottocampionati)
subplot(1,2,2);
cols = double([colorR(:) colorG(:) colorB(:)]) / 255;
if isempty(cols)
    warning('Nessun colore disponibile per la visualizzazione storica.');
```

```

else
    if all(cols(:)==0)
        warning('Tutti i colori storici sono zero: verifica warping e proiezione.');
```

```

    end
    scatter3(X_s, Y_s, Z_s, 8, cols, 'filled');
```

```

end
title('Nuvola Storica (back-projection)');
xlabel('X (m)'); ylabel('Y (m)'); zlabel('Z (m)');
axis equal; grid on;
view(3);

rotate3d on;

% Log
fid = fopen(logfile, 'a');
fprintf(fid, 'Visualizzazione confronto completata\n');
fclose(fid);

disp('? Confronto completato: naviga e confronta le due viste!');

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% Opzione finale – Visualizzazione residui con frecce ingrandite
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

choice = questdlg( ...
    'Vuoi visualizzare la mappa dei residui con frecce ingrandite?', ...
    'Visualizzazione residui', ...
    'Si', 'No', 'No');
```

```

if strcmp(choice,'Si')
    figure('Name','Residui punti omologhi (freccie ingrandite)');
    imshowpair(render_img, warped_hist, 'blend'); hold on;

    % Punti di riferimento
    plot(pts_render(:,1), pts_render(:,2), 'go', ...
        'MarkerSize', 8, 'LineWidth', 1.5);

    % Freccie dei residui (ingrandite)
    scaleFactor = 8; % fattore di ingrandimento freccie
    quiver(pts_render(:,1), pts_render(:,2), ...
        diffs(:,1)*scaleFactor, diffs(:,2)*scaleFactor, ...
        0, 'r', 'LineWidth', 1.5, 'MaxHeadSize', 3);

    title(sprintf('Residui omologhi (RMS = %.2f px)', rms_err));
    legend('Punti render','Residui (ingranditi)','Location','best');
    axis ij; axis equal;
    hold off;
end

disp('? Lavoro finito.');
```




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

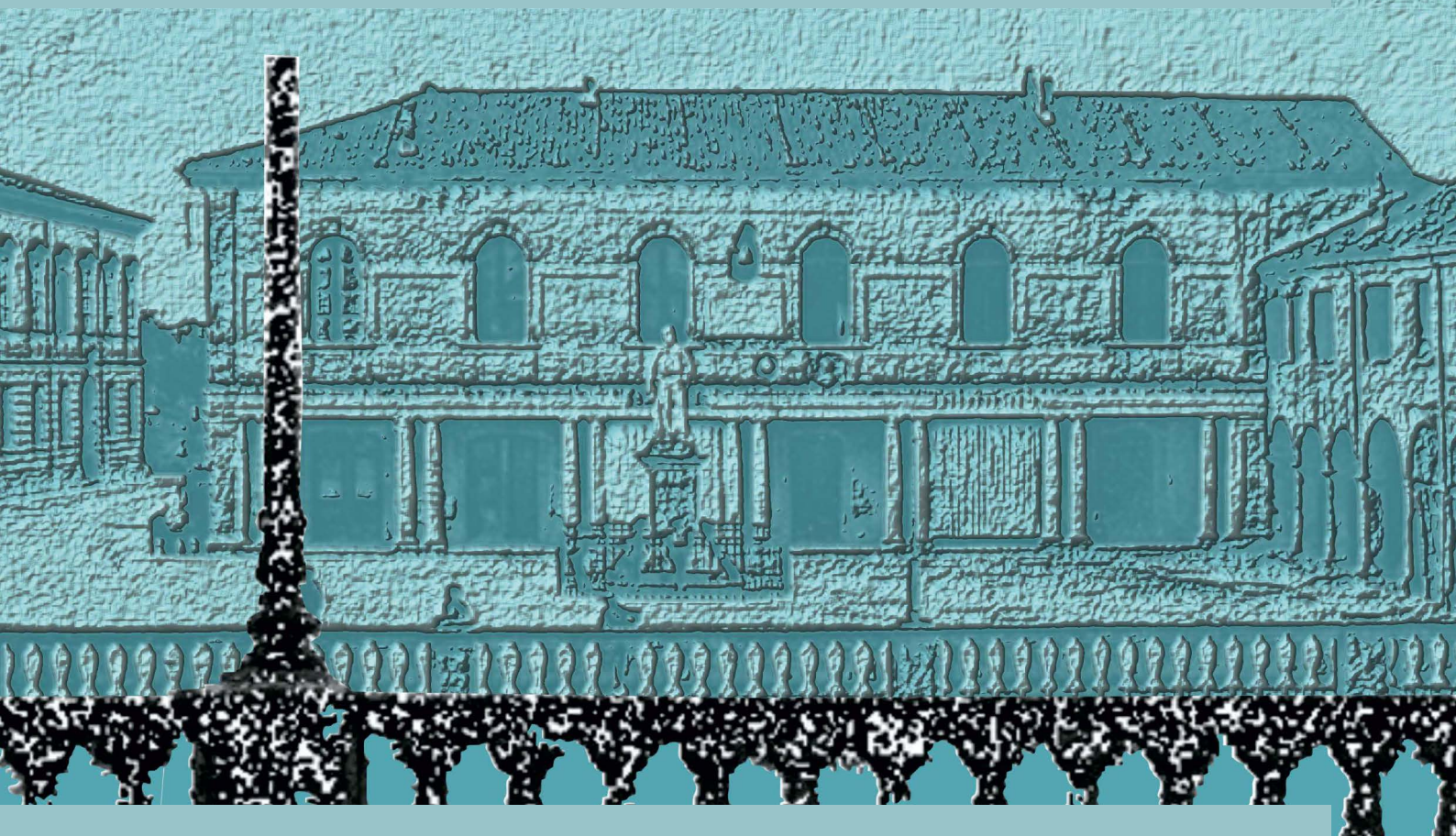
Segreteria di redazione
segreteria.collane-editoriali@ingv.it

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