

Alma Mater Studiorum – Università di Bologna – Campus di Cesena

—
Dipartimento Informatica – Scienza e Ingegneria (DISI)

Corso di Laurea in Ingegneria e Scienze Informatiche

DS4H Image Alignment tool: implementazione e validazione di algoritmi per allineamento automatico elastico di immagini multimodali

Tesi in: Web Semantico

Relatore

Prof.ssa Antonella Carbonaro

Presentata da

Matteo Iorio

Correlatori

Prof. Filippo Piccinini

Dr. Nicola Normanno

Prof. Gastone Castellani

Anno Accademico 2025 / 2026

*“A cosa posso paragonare
il mondo e la vita dell’uomo?
Al riflesso della luna
nella goccia di rugiada, scossa
dal becco di una gru.”*

Dōgen

Parole Chiave

Istologia

Microscopia

Allineamento multimodale

Registrazione automatica

Open-source tool

Abstract (English version)

Many cancer studies begin with the analysis of histological samples, that is, tissue samples taken from the patient. Through the use of specific markers, namely selective dyes applied to the section to be analyzed, specific parts of the sample are examined and more information is obtained from the sample itself. However, these markers cannot always be applied simultaneously and are often used sequentially after a washing step of the sample. To conduct colocalization studies, the images obtained must therefore be aligned by simulating the parallel acquisition of the different signals. Unfortunately, elastic-type deformations may occur in the samples under observation due to repeated washing, causing difficulties in aligning these images. The use of algorithms for the correction of elastic deformation is therefore essential to address this problem. In particular we developed *DS4H Image Alignment*, an open-source, user-friendly plug-in implemented for *ImageJ/Fiji* that not only allows multi-modal image alignment with minimal user interaction, but also enables the application of algorithms for elastic deformation correction. In particular, two different elastic registration algorithms, called *bUnwarpJ* and *BigWarp*, have been implemented, and a series of experiments have been conducted to validate them in different application contexts.

Abstract (versione italiana)

Molti degli studi sul cancro iniziano con l'analisi di campioni istologici, cioè campioni di tessuto prelevati dal paziente. Attraverso l'uso di marcatori specifici, ovvero coloranti selettivi applicati alla sezione da analizzare, vengono studiate parti specifiche del campione ed ottenute più informazioni dal campione stesso. Tuttavia, questi marcatori non possono sempre essere applicati contemporaneamente e spesso vengono utilizzati in sequenza dopo un lavaggio del campione. Per poter condurre studi di colocalizzazione, le immagini ottenute devono quindi essere allineate simulando l'acquisizione parallela dei diversi segnali. Purtroppo, può capitare di applicare delle deformazioni di tipo elastico sui campioni sotto osservazione, dovuto ai continui lavaggi, provocando delle difficoltà nell'allineamento di queste immagini. L'utilizzo di algoritmi per la correzione della deformazione elastica risulta quindi fondamentale per poter correggere questo tipo problema. Abbiamo sviluppato *DS4H Image Alignment*, un plug-in open-source, user-friendly implementato per *ImageJ/Fiji* che permette oltre ad allineare immagini multimodali con una richiesta di minima iterazione da parte dell'utente, anche di applicare algoritmi per la correzione della deformazione elastica. In particolare, sono stati implementati due diversi algoritmi di registrazione elastica, denominati *bUnwarpJ* e *BigWarp*, e sono stati condotti una serie di esperimenti per validarli in diversi contesti applicativi.

Indice

Introduzione	9
1 Tool per allineamento elastico	15
1.1 Deformazioni di tipo elastico	15
1.1.1 Algoritmi di deformazione elastica	16
1.2 Competitors	18
1.2.1 BigWarp	19
1.2.2 Correlia	19
1.2.3 Elastix	20
1.2.4 ITK	20
1.2.5 Elastic Alignment e Montage	21
1.2.6 Moving Least Squares	21
1.2.7 BigStitcher	21
1.2.8 TransformJ	22
1.2.9 ImageJ	22
1.2.10 UnwarpJ	23
1.2.11 bUnwarpJ	23
1.3 Limiti dei principali competitors	23
1.3.1 BigWarp	23
1.3.2 Correlia	24
1.3.3 Elastix	24
1.3.4 ITK	24
1.4 DS4H Image Alignment tool	25
1.4.1 Presentazione generale	25
1.5 Modalità d'uso dell'applicazione DS4H	26
1.5.1 Caricamento delle immagini	26
1.5.2 Caricamento delle immagini tramite voce File	27
1.5.3 Caricamento delle immagini tramite voce Project	27
1.5.4 Impostazioni del plugin	28
1.5.5 Scelta immagine target	30
1.5.6 Rimozione immagine	30
1.5.7 Gestione corner points	31
1.5.8 Aggiunta dei corner points	31
1.5.9 Selezione singola o multipla dei corner points	32
1.5.10 Copia dei corner points	33
1.5.11 Rimozione dei corner points	34
1.5.12 Modifica dei corner points	35
1.5.13 Esportare il progetto	36

1.5.14	Allineamento manuale rigido	36
1.5.15	Allineamento manuale elastico	38
1.5.16	Salvataggio delle immagini	38
1.5.17	Riuso delle immagini	39
1.5.18	Applicazione algoritmo di deformazione elastica dopo allineamento rigido	39
1.5.19	Allineamento automatico rigido	40
1.5.20	Allineamento automatico di tipo elastico	40
2	Supporto alle architetture ARM	41
2.1	Computer Vision	41
2.2	OpenCV	41
2.2.1	OpenCV Extra Features Library	42
2.2.2	Compilazione di OpenCV	42
3	Implementazione di Algoritmi di deformazione Elastica	44
3.1	Deformazioni elastiche su immagini	44
3.1.1	Libreria per la deformazione elastica di immagini	45
3.1.2	Utilizzo della libreria BigWarp	45
3.2	Descrizione di Thin Plate Splines	46
3.2.1	Formulazione matematica di Thin Plate Spline	46
3.3	Implementazione algoritmi di deformazione elastica in DS4H	48
3.3.1	Design delle classi	48
3.3.2	Algoritmo elastico manuale	49
3.3.3	Algoritmo elastico automatico	51
4	Risultati sperimentali	53
4.1	Tool interno per misurazione di similarità	53
4.1.1	Deformazioni applicate	53
4.1.2	Metriche di misurazione	57
4.1.3	Dataset	58
4.2	Risultati e osservazioni	59
5	Conclusioni e sviluppi futuri	64

Introduzione

Questo progetto di Tesi è stato sviluppato all'interno del gruppo di ricerca "*Data Science for Health*" (DS4H), composto da ricercatori e professori dell'Università di Bologna (UniBo) e dell'*Istituto Romagnolo dei Tumori Dino Amadori* (IRST) di Meldola (FC). Il gruppo DS4H si propone di coordinare professionisti e risorse provenienti dai settori dell'informatica, dell'ingegneria dell'informazione, della fisica, nonché dalle discipline biomediche, della biologia, della chimica e della medicina. In particolare, il progetto di tesi intitolato "*DS4H Image Alignment tool: implementazione e validazione di algoritmi per allineamento automatico elastico di immagini multimodali*" si basa sui lavori pubblicati in una serie di Tesi precedenti che hanno portato alla creazione delle prime versioni del software chiamato "*DS4H Image Alignment*". L'obiettivo di questo progetto è sviluppare e applicare algoritmi di deformazione elastica per l'allineamento di immagini multimodali bidimensionali acquisite tramite microscopia a campo largo. Tale procedura consente di eseguire la registrazione accurata delle immagini, finalizzata allo studio della colocalizzazione dei segnali intracellulari, migliorando la qualità e l'affidabilità della valutazione dei campioni istologici analizzati.

In particolare, *DS4H Image Alignment* risponde alle richieste originariamente formulate da medici e biologi dell'IRST. Per comprendere il contesto operativo, è importante spiegare come i professionisti dell'IRST valutano tipicamente i tumori dei pazienti quando dispongono di una biopsia del tessuto. Di solito, si inizia creando provini istologici dai campioni di tessuto prelevati dal paziente, a cui vengono applicati marcatori specifici per identificare le cellule con caratteristiche tumorali particolari. Questi marcatori sono tipicamente sonde fluorescenti visibili attraverso un microscopio a fluorescenza. Inoltre, per ottenere maggiori informazioni dal campione, vengono frequentemente utilizzati più marcatori. Tuttavia, questi marcatori non possono sempre essere applicati contemporaneamente e vengono spesso utilizzati in sequenza dopo un lavaggio del campione. Tali lavaggi possono introdurre delle deformazioni di tipo elastico, rendendo più complessa l'operazione di allineamento delle varie immagini campionate. Dopo l'applicazione dei coloranti, vengono solitamente ottenute immagini fluorescenti utilizzando microscopi a campo largo. L'allineamento di tali immagini ed è soggetto a possibili errori. Un software potrebbe rendere il processo più rapido e affidabile, consentendo il riconoscimento delle aree comuni e l'allineamento dei segnali, aiutando inoltre a correggere le possibili deformazioni elastiche causate dai vari risciacqui dei campioni. Da questa necessità è stato necessario introdurre all'interno del plug-in *DS4H Image Alignment* nuovi algoritmi per la correzione delle deformazioni elastiche, con l'obiettivo di colmare questa lacuna

tecnica e automatizzare il processo nell'ambito dell'istologia. L'obiettivo finale è quello di ottenere un'analisi rapida e precisa dei campioni istologici attraverso uno strumento di allineamento delle immagini acquisite con diverse tecniche e colorazioni (multi-modalità).

Il riconoscimento del problema e le prime ipotesi di soluzione sono stati formulati dalla Prof.ssa Antonella Carbonaro e dal Prof. Filippo Piccinini, successivamente sviluppate utilizzando immagini ottenute da campioni istologici di pazienti reali. Queste immagini sono state fornite dal Dr. Nicola Normanno, direttore scientifico dell'IRST di Meldola, e dal Prof. Gastone Castellani, Professore Ordinario di Fisica, direttore del gruppo di ricerca *BioPhysics Group* dell'Università di Bologna.

In questo progetto di tesi sono stati svolti i seguenti tasks:

- A) Compilazione della libreria *OpenCV* per fornire il supporto alle piattaforme *ARM*;
- B) Studio dei tools stato dell'arte;
- C) Implementazione di algoritmi di deformazione elastica in modalità automatica;
- D) Implementazione di algoritmi di deformazione elastica in modalità manuale
- E) Validazione del tool usando dataset di test;
- F) Risoluzione di bug presenti;
- G) Release codice sorgente e *standalone* per *Windows*, *Mac* e *Linux*.

In particolare, sono stati sviluppati 3 moduli principali:

- (I) Il primo modulo ha consentito di risolvere un importante problema presente sulle piattaforme di tipo *ARM (Apple)*. In particolare, non era possibile utilizzare alcuni algoritmi di allineamento automatico di tipo rigido.
- (II) Il secondo modulo riguarda l'implementazione di un nuovo algoritmo di deformazione elastica seguendo due modalità principali di utilizzo: modalità manuale e modalità automatica. In questo modulo verranno illustrate le varie modalità attraverso la quale questi di deformazione elastica sono stati implementati.
- (III) Il terzo modulo è dedicato alla valutazione del comportamento del nuovo algoritmo di deformazione elastica. Il comportamento è stato valutato tramite l'utilizzo di diverse metriche specifiche per l'analisi di immagini.

Nei prossimi capitoli di questo documento vengono descritte le soluzioni specifiche e dettagliate proposte per lo sviluppo dei moduli sopra menzionati.

1 Provini istologici: importanza e preparazione

L'istologia è un settore della biologia che si occupa dell'analisi della struttura microscopica e ultramicroscopica dei tessuti animali e vegetali. Un campione di tessuto utilizzato per queste analisi è chiamato provino istologico. In questo documento, con il termine "*provino istologico*" ci riferiamo a campioni prelevati da pazienti con lo scopo di monitorare e studiare patologie, nonché identificare malattie per determinare il trattamento più appropriato.

Nel corso degli anni, le tecniche di studio dei campioni istologici sono state perfezionate, con l'adozione di metodologie che consentono una migliore e più duratura conservazione dei campioni stessi, oltre a una resa informativa ottimale per le analisi. In particolare, un provino istologico deve essere preparato seguendo cinque fasi fondamentali successive, chiamate:

1. Fissazione
2. Disidratazione
3. Inclusione
4. Sezionamento
5. Colorazione

Durante il processo di *fissazione*, si creano condizioni ottimali per il provino istologico, simili a quelle fisiologiche, al fine di conservare il tessuto e rafforzare la struttura cellulare attraverso una cross-correlazione irreversibile delle proteine. La fissazione può essere eseguita immergendo il campione in una sostanza biologica come la formalina o in un fluido per il congelamento. Successivamente, il campione sarà estratto da tale materiale per il sezionamento.

Successivamente, si procede con la disidratazione, un processo finalizzato alla rimozione dell'acqua dal campione sostituendola tipicamente con molecole di etanolo. Una volta completata la disidratazione, si utilizza lo xilene per rimuovere l'etanolo dal tessuto, in modo che possa essere preparato per il taglio al microtomo. Questa fase prevede l'uso di alcoli con una crescente intensità e volume per evitare danni alle cellule.

In seguito, si procede con *l'inclusione*, che è una fase in cui il tessuto viene solidificato utilizzando sostanze chimiche specifiche, di solito paraffina. Questo processo implica l'esposizione del campione

al calore seguito dal congelamento per prevenire il deterioramento. Poiché la paraffina e l'etanolo non sono miscibili tra loro, viene utilizzato un fluido che può mescolarsi con entrambi per "pulire" il campione prima dell'inclusione.

Nella fase successiva, chiamata *sezionamento*, il tessuto viene tagliato utilizzando un microtomo o un criostato, che sono strumenti di precisione appositamente progettati per questa operazione. L'obiettivo è ottenere sezioni sottili sufficienti, di solito intorno ai cinque micrometri, affinché la luce del microscopio possa attraversarle agevolmente per consentirne l'analisi.

Infine, si procede con *l'applicazione di diverse coloranti* al fine di evidenziare parti specifiche e proprietà del tessuto, aumentandone il contrasto e consentendo di distinguere tali parti dagli altri componenti del materiale biologico. Prima dell'applicazione dei coloranti, il campione viene sottoposto a un processo di deparaffinazione e reidratazione, in modo che possa assorbire le tinture. Successivamente, il campione viene risciacquato e disidratato una volta completata la procedura. Questa fase è nota come *colorazione* e può alterare la sezione del campione, richiedendo quindi un'ulteriore fase di lavaggio o la sua sostituzione prima di ripetere la procedura per lo studio di diverse parti utilizzando coloranti diversi [1][2].

Le seguenti illustrazioni mostrano i risultati ottenuti acquisendo immagini del medesimo campione utilizzando un microscopio a campo largo. Il tessuto del campione proviene dal rene di un modello murino ed è stato sottoposto a tutte le fasi descritte in precedenza. In particolare, sono state acquisite immagini utilizzando la tecnica del *Differential Interference Contrast* (DIC) ed immagini fluorescenti dopo l'applicazione di tre diverse sonde, precisamente DAPI, Cy3 e FITC.

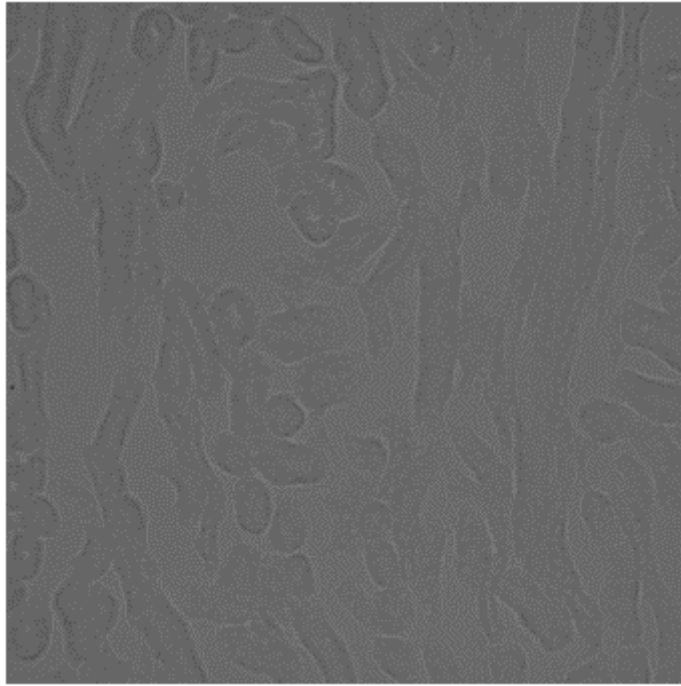


Figura 1.1: sezione di rene di topo con ingrandimento 20x in modalità DIC.

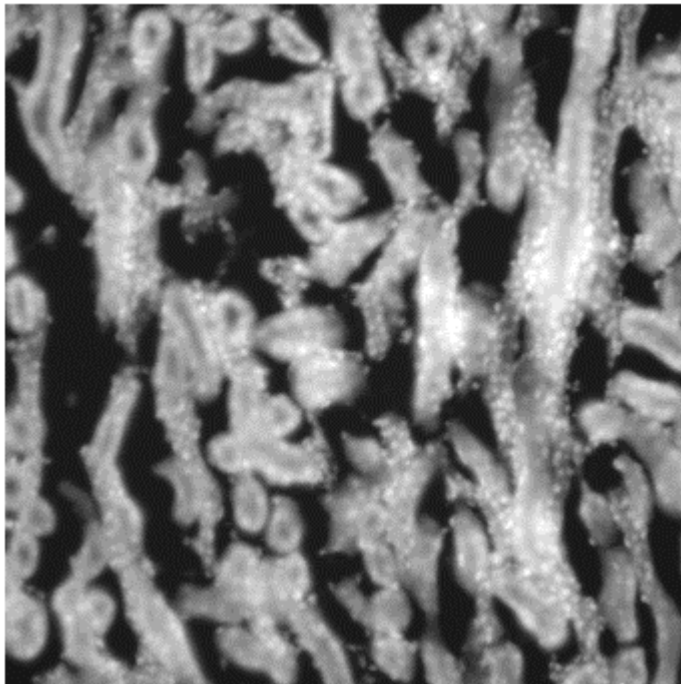


Figura 1.2: sezione di rene di topo con ingrandimento 20x e colorante DAPI.

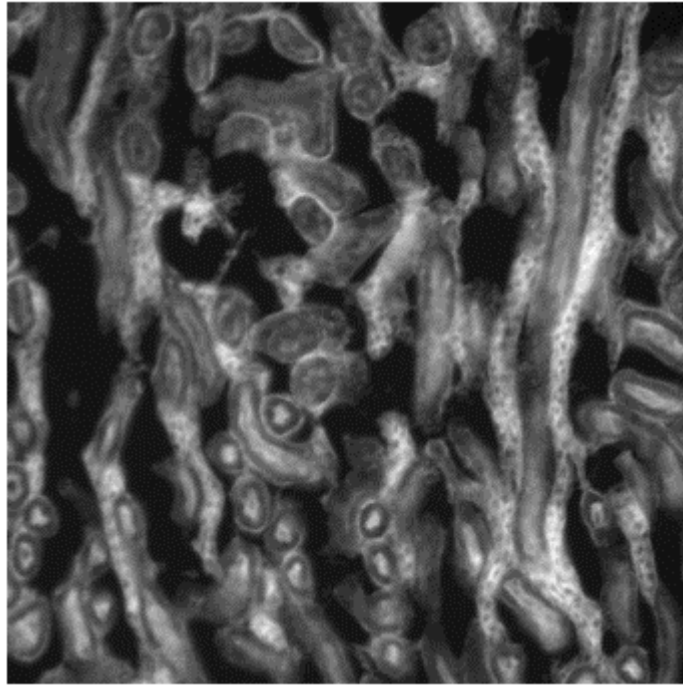


Figura 1.3: sezione di rene di topo con ingrandimento 20x e colorante Cy3.

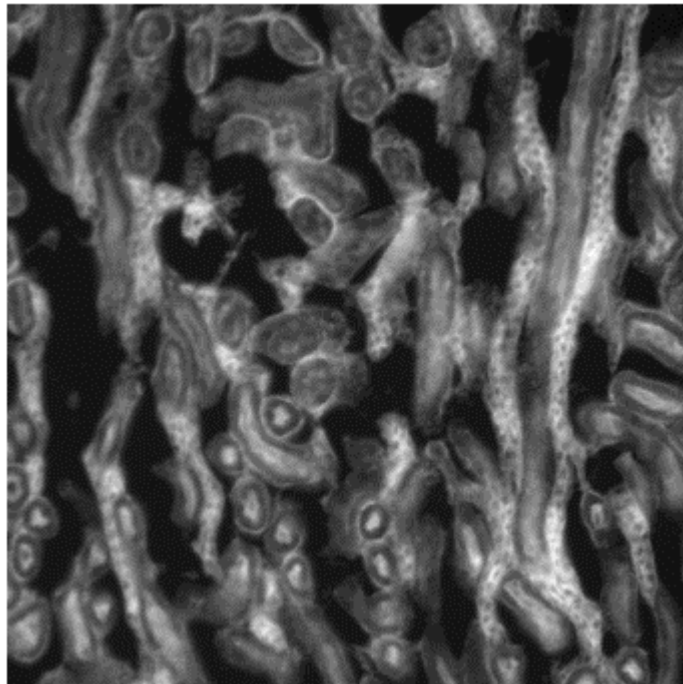


Figura 1.4: sezione di rene di topo con ingrandimento 20x e colorante FITC.

1 Tool per allineamento elastico

Come menzionato in precedenza, si parla di *colocalizzazione* quando più immagini vengono allineate sulla base di una correlazione spaziale non casuale. Ciò significa che le immagini forniscono informazioni diverse sullo stesso oggetto in fase di studio e l'osservazione combinata di tali immagini può fornire una quantità maggiore di informazioni rispetto alle analisi condotte individualmente sui singoli segnali.

Quando si lavora con campioni istologici, è comune utilizzare coloranti al fine di evidenziare diverse parti subcellulari del campione. Tuttavia, per un'analisi più approfondita, spesso è necessario utilizzare più marcatori che non possono sempre essere applicati contemporaneamente e devono essere utilizzati in serie, con un lavaggio del campione tra l'applicazione di ciascun marcatore, provocando possibili deformazioni di tipo elastico. Tali deformazioni possono complicare l'operazione di allineamento finale dei diversi campioni. È quindi necessario sottoporre le immagini a un algoritmo di deformazione elastica, per poi applicare un algoritmo di registrazione, manuale o automatico, che consenta lo studio della colocalizzazione a livello di singola cellula, simulando un'acquisizione simultanea dei vari segnali. Attualmente, tuttavia, non è disponibile una procedura standard per l'allineamento delle immagini ottenute con questo approccio.

1.1 Deformazioni di tipo elastico

Nel presente capitolo, viene approfondito il fenomeno delle *deformazioni di tipo elastico*, un comportamento indotto su materiali sottoposti ad azioni esterne che permette loro di modificare la propria forma, e dimensione. Le deformazioni elastiche si possono considerare in *deformazione longitudinale*, osservabile come un allungamento o accorciamento del corpo, sia la *deformazione trasversale*, responsabile di variazioni delle dimensioni laterali.

Durante lo studio di provini istologici, vengono applicati una batteria di coloranti. Questo comporta continui risciacqui del campione sotto osservazione, per l'applicazione di nuovi coloranti, provocandone delle deformazioni nella forma del campione.

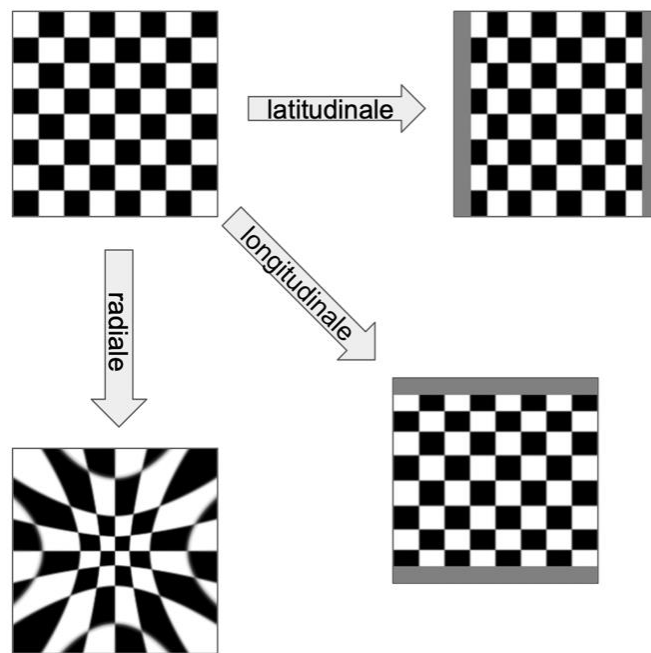


Figure 1-1 Trasformazioni latitudinali, longitudinali e radiali

1.1.1 Algoritmi di deformazione elastica

Qui di seguito verrà descritta la modalità di funzionamento degli algoritmi di deformazione elastica. Questi algoritmi per funzionare si basano su un'immagine di partenza (*reference*) e le varie immagini deformate da modificare. In questi algoritmi, si cerca di riportare le immagini deformate alla corrispondenza dell'immagine di riferimento.

Gli algoritmi di deformazione elastica calcolano una *mappa di trasformazione spaziale* ($T(x)$) che porta ogni punto dell'immagine deformata nella posizione corrispondente dell'immagine di riferimento. L'obiettivo è quello di allineare le caratteristiche delle immagini acquisite in condizioni diverse (postura diversa e modalità diversa dovuti a vari risciacqui dei campioni), in modo che corrispondenti strutture si sovrappongano il più possibile. Questa mappa è spesso rappresentata come un campo di spostamento (*displacement field*) definito a ogni pixel.

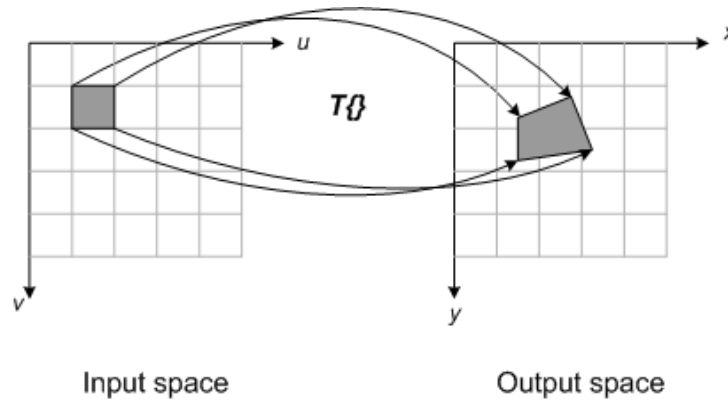


Figure 1-2 Mappa di trasformazione spaziale [3]

Gli algoritmi moderni si differenziano per il modello di trasformazione (*parametrico* o *non parametrico*), la misura della similarità usata per valutare l'allineamento, la strategia di regolarizzazione per rendere la soluzione “plausibile” e l'algoritmo di ottimizzazione. La combinazione di questi componenti determina se l'algoritmo è veloce, stabile e se preserva la topologia. Esistono diverse modalità con la quale si può portare l'immagine deformata alla forma originale:

- **Demons** – algoritmo iterativo basato su forze locali che “spingono” l'immagine deformata verso l'immagine di riferimento, utilizzato principalmente per la registrazione intensità-basata. Questo tipo di deformazioni garantiscono trasformazioni invertibili (utili per riportare la deformata all'originale senza perdita topologica) (Figure 1-3)[4];

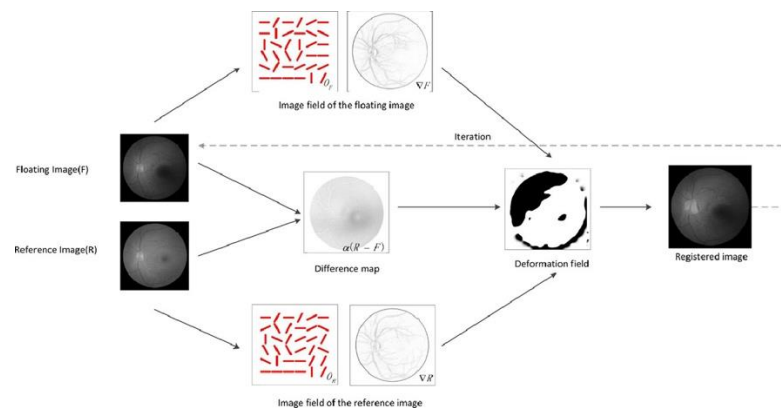


Figure 1-3 Demons algorithm example

- **Free-Form Deformation con B-spline** – rappresenta la trasformazione tramite una griglia di nodi di controllo (B-spline), offre un controllo multi-scala e buone proprietà di regolarità. È usata in ambito medico e permette di ottenere campi di spostamento lisci e interpretabili. Per riportare l'immagine deformata all'immagine di riferimento, si valuta il campo e si applica la trasformazione inversa o si stima direttamente la mappa inversa (Figure 1-4 [5]);

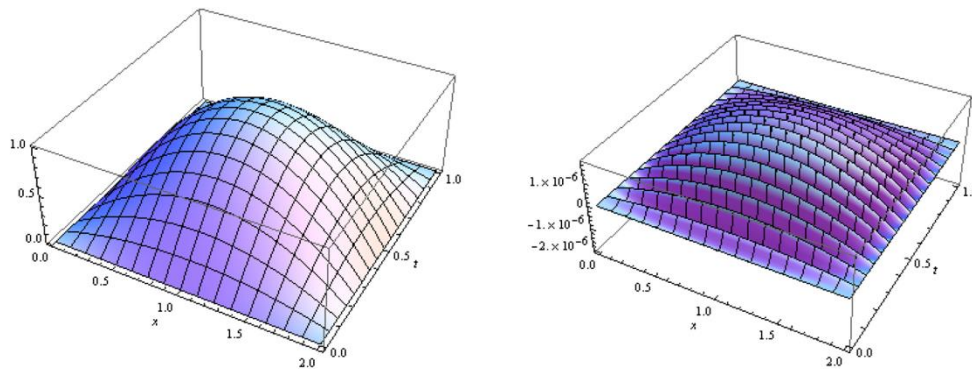


Figure 1-4 B-Spline deformation

- **Diffeomorphic methods** – costruiscono trasformazioni che sono continuamente invertibili e differenziabili; questo è cruciale quando si vuole riportare l'immagine deformata in modo fisicamente coerente. Utilizzati per la registrazione di alta dimensione;
- **Metodi fisici / FEM (elastic registration)** – modellano direttamente l'elasticità lineare/non lineare del materiale usando equazioni di equilibrio e risolvono per il campo di spostamento con tecniche di elementi finiti; utili quando si desidera che la trasformazione segua leggi meccaniche;
- **Optical Flow** – basati su un approccio variazionale che minimizza una funzione energia (termine di similarità + termine di regolarizzazione), spesso usati per spostamenti piccoli e continui e adattabili per diversi tipi di similarità (Figure 1-5 [6]).



Figure 1-5 Optical flow by using OpenCV

1.2 Competitors

Un algoritmo di deformazione elastica combinato ad un allineamento di tipo manuale o automatico richiede tempo, diversi software tools ed è soggetto a potenziali errori. L'utilizzo di un unico software potrebbe rendere il processo più rapido e affidabile. Attualmente, esistono vari strumenti disponibili per la registrazione l'applicazione di algoritmi rigidi e non rigidi su immagini. In questa sezione, viene fornita una breve descrizione delle loro caratteristiche principali.

1.2.1 BigWarp

BigWarp è un plugin per *ImageJ/Fiji* che permette l'allineamento elastico e manuale di immagini attraverso l'uso di punti di riferimento interattivi. È stato descritto nel 2016. Questo plugin consente agli utenti di definire manualmente le corrispondenze tra i punti di riferimento tramite un'interfaccia intuitiva, consentendo loro di posizionare coppie di punti di riferimento e visualizzare in tempo reale gli effetti della deformazione. Il modello di registrazione utilizzato da *BigWarp* si basa sull'utilizzo della tecnica *Thin Plate Spline*, che permette di definire una trasformazione deformabile uniforme che mappa in modo preciso i punti di riferimento. È possibile scegliere tra diversi modelli di registrazione, come il modello affine (trasformazione lineare con traslazione, rotazione, scale indipendenti e taglio), il modello di similarità (trasformazione lineare con traslazione, rotazione e un parametro di scala), il modello di rotazione (noto anche come modello rigido, una trasformazione lineare con traslazione e rotazione) o il modello di traslazione più semplice (solo spostamenti in X e Y). È inoltre possibile esportare e importare i punti di riferimento da file di testo. Una volta ottenuta l'immagine deformata, è possibile allinearla facilmente all'immagine di riferimento sfruttando le funzionalità offerte da *ImageJ/Fiji* (gli autori del plugin hanno fornito diversi video tutorial per mostrare come integrare in modo efficace *BigWarp* e *ImageJ/Fiji*) [7].

1.2.2 Correlia

Correlia è un plug-in open-source per *ImageJ/Fiji*, che funziona in modo indipendente dalla piattaforma e ha la capacità di gestire dati di microscopia 2D di vario tipo. È stato specificamente progettato per la co-registrazione di set di dati di microscopia multimodale 2D. *Correlia* è stato sviluppato presso "ProVIS - Center for Correlative Microscopy" (Helmholtz-Centre for Environmental Research, UFZ, Germania) ed è stato inizialmente concepito per soddisfare le esigenze della microscopia chimica, coinvolgendo diverse micrografie e mappe chimiche con diverse risoluzioni e campi visivi. Il plug-in è accompagnato da una documentazione esaustiva e da esempi di dati. Da un lato, il software offre diverse opzioni di registrazione manuale e automatica ed è integrato direttamente con *bUnwarpJ* (un altro popolare plug-in *ImageJ/Fiji*) per eseguire la co-registrazione elastica. D'altra parte, l'utilizzo di *Correlia* può essere complesso e richiede spesso la definizione di più parametri. Il plug-in offre diverse modalità di visualizzazione, ma le opzioni di esportazione sono limitate. Ad esempio, richiede la definizione di un frame di riferimento per ritagliare tutte le immagini da registrare e non consente l'esportazione delle immagini allineate come uno stack multi-frame a piena risoluzione [8].

1.2.3 Elastix

Elastix è un software open source basato su *ITK* che fornisce una raccolta di algoritmi ampiamente utilizzati per risolvere problemi di registrazione delle immagini, principalmente nel campo medico. La sua architettura modulare consente agli utenti di configurare, testare e confrontare diversi metodi di registrazione per adattarli alle proprie esigenze specifiche. *Elastix* offre una varietà di opzioni, tra cui metodi di ottimizzazione, schemi multi-risoluzione, interpolatori, modelli di trasformazione e funzioni di costo. Il codice sorgente in linguaggio C++ può essere compilato su diversi sistemi operativi, come *Windows XP*, *Linux* e *Mac OS X*. Mentre *Elastix* non ha una GUI ufficiale, esistono alcuni plugin che offrono un'interfaccia grafica per coloro che preferiscono utilizzare le funzionalità di *Elastix* in modo visuale. Ad esempio, *SlicerElastix* è un'estensione che integra *Elastix* all'interno del software *3D Slicer*. Inoltre, *Elastix* è supportato da *ITKElastix*, che rende il software disponibile in *Python*, e da *SimpleElastix*, che offre supporto per molti linguaggi di programmazione come *Java*, *R*, *Ruby*, *C#* e *Lua*. Gli autori hanno anche sviluppato una versione di *Elastix* per *ImageJ/Fiji*, sebbene l'installazione e la registrazione delle immagini con successo possano risultare complesse [9].

1.2.4 ITK

The *National Library of Medicine Insight Toolkit (ITK)* è un sistema open source multiplatforma progettato per l'elaborazione di immagini mediche. Questo toolkit fornisce ai ricercatori nel campo dell'imaging medico una vasta gamma di algoritmi all'avanguardia per la registrazione, la segmentazione, l'analisi e la quantificazione dei dati medici. Inizialmente concepito nel 1999 per supportare il progetto specifico chiamato "*The Visible Human Project*", *ITK* si è evoluto nel corso degli anni diventando la tecnologia di base per molti prodotti commerciali di analisi dell'immagine medica in tutto il mondo. Nel 2005, la comunità *ITK* ha fondato *l'Insight Journal*, una rivista scientifica dedicata alla pratica del metodo scientifico. In questa rivista, tutti gli articoli devono includere la serie completa di codice sorgente, dati e parametri necessari per riprodurre le scoperte degli autori. Grazie all'*Insight Journal*, il repository *ITK* oggi contiene milioni di righe di codice sorgente ed è molto popolare. Tuttavia, va notato che *ITK* è stato progettato principalmente per scienziati e sviluppatori informatici, pertanto potrebbe risultare complesso per biologi o medici senza una formazione specifica in informatica. Per semplificarne l'utilizzo, gli autori hanno creato *SimpleITK*, un'interfaccia di programmazione semplificata che mette a disposizione gli algoritmi e le strutture dati di *ITK*. *SimpleITK* supporta diverse interfacce per diversi linguaggi di programmazione, andando oltre il C++ originale. Tuttavia, al momento non esiste una GUI (interfaccia grafica)

ufficiale per *ITK* o *SimpleITK*, quindi l'utilizzo di questi strumenti richiede competenze di programmazione [10].

1.2.5 Elastic Alignment e Montage

I plugin "*Elastic Alignment*" e "*Montage*" sono funzionalità incorporate nel software "*TrakEM2*". Questi plugin sono utilizzati quando si lavora con ampie serie di sezioni multi-tile tramite il menu di allineamento del software. Il plugin "*Elastic Alignment*" si occupa di allineare serie di sezioni deformate, mentre il plugin "*Montage*" crea mosaici combinando immagini sovrapposte che presentano deformazioni non lineari. L'obiettivo è deformare le immagini in modo da ottenere sovrapposizioni ottimali tra di esse. Ogni deformazione viene calcolata in modo da ridurre al minimo la deformazione complessiva del mosaico finale. Entrambi i plugin funzionano esclusivamente con stacks di immagini, consentendo un'elaborazione coerente e integrata delle immagini all'interno di *TrakEM2* [11].

1.2.6 Moving Least Squares

Il plugin *Moving Least Squares* utilizza il metodo dei minimi quadrati ponderati. Questa tecnica è utile per deformare un'immagine singola utilizzando una serie di punti di riferimento. Tuttavia, non si occupa dell'allineamento tra immagini. A differenza del plugin *bUnwarpJ*, che utilizza deformazioni elastiche, le trasformazioni implementate nel plugin *Moving Least Squares* sono considerate rigide. Gli autori del plugin hanno voluto dimostrare che è possibile ottenere deformazioni utilizzando trasformazioni geometricamente simili, ovvero combinando trasformazioni rigide con ridimensionamenti isometrici. Questo approccio permette di evitare la complessità delle minimizzazioni non lineari [12].

1.2.7 BigStitcher

"*BigStitcher*" è la versione successiva di "*Image Stitching*" [13] e mantiene le stesse funzionalità. Tuttavia, presenta un importante miglioramento: è in grado di elaborare e visualizzare immagini di qualsiasi dimensione, grazie alla sua invarianza di dimensione. Attualmente, il software è in grado di effettuare un'ottimizzazione globale che permette di allineare insiemi di dati con connessioni sparse, dove il contenuto dell'immagine è separato da aree in cui lo sfondo è quasi costante. Tuttavia, è importante notare che il software si trova ancora nella fase di beta test e non è considerato completo. Oltre a queste funzionalità, "*BigStitcher*" [14] offre anche altre caratteristiche, tra cui:

- La registrazione multi-view dove si possono importare immagini provenienti il cui orientamento è diverso;
- L'elaborazione dei dati come la fusione o la deconvoluzione delle immagini allineate, anche per le risoluzioni ridotte dell'immagine prodotta sia per selezioni di parte di essa;
- Scelta della miglior direzione della luce per ogni immagine;
- Supporto di griglie non regolari a cui ad esempio mancano delle parti.

1.2.8 TransformJ

TransformJ è un plugin avanzato sviluppato per *ImageJ* con l'obiettivo di eseguire trasformazioni geometriche di elevata qualità su immagini bidimensionali e tridimensionali. A differenza degli strumenti di trasformazione più basilari inclusi nativamente nel software, *TransformJ* si distingue per l'impiego di modelli di interpolazione basati sulle spline, che garantiscono una maggiore precisione nella manipolazione dell'immagine e riducono le distorsioni che normalmente si verificano con metodi interpolativi tradizionali. Questa caratteristica lo rende particolarmente adatto per applicazioni in cui è essenziale preservare la qualità del segnale, come nella registrazione di immagini, nell'allineamento di stack volumetrici o nella correzione geometrica di dati acquisiti mediante microscopie avanzate.

1.2.9 ImageJ

ImageJ è un software open-source ampiamente utilizzato in ambito scientifico per l'elaborazione e l'analisi di immagini digitali. Sviluppato originariamente dal *National Institutes of Health* (NIH), si basa sul linguaggio *Java* e risulta pertanto compatibile con i principali sistemi operativi. La sua diffusione nella comunità accademica è dovuta alla combinazione tra semplicità d'uso, versatilità e un elevato grado di estendibilità. *ImageJ* è infatti progettato per operare su immagini bidimensionali e multidimensionali, incluse sequenze temporali e stack tridimensionali, permettendo così la gestione di dataset complessi tipici della ricerca biologica, medica.

1.2.10 UnwarpJ

UnwarpJ è un *plugin* Java per *ImageJ* che implementa una registrazione elastica basata su *spline cubiche*. L'idea è quella di distorcere in direzione opposta un'immagine sorgente per farla somigliare a un'immagine target. Permette tre tipi di modalità di funzionamento ovvero automatica, interattiva e mista. Questo tipo di software viene solo ed esclusivamente utilizzato per deformazioni elastiche,

non ha supporto per allineamenti di tipo rigido. Il progetto è completamente *open source*, ma non sembra essere un plugin con sviluppo molto attivo. [15]

1.2.11 bUnwarpJ

Il plugin *bUnwarpJ* è un'evoluzione di *UnwarpJ*, anch'essa come plugin per *ImageJ/Fiji* ma con caratteristiche avanzate come registrazione elastica bidirezionale, ovvero viene calcolata la deformazione dall'immagine A a B e la deformazione inversa, ovvero da B a A, garantendo una certa pseudo invertibilità della trasformazione finale. Questo software viene utilizzato solo ed esclusivamente per effettuare trasformazioni di tipo elastico, non fornisce alcun supporto per le deformazioni di tipo rigido. Il software è *open-source* ed incluso come default all'interno delle versioni di *Fiji*. [16]

1.3 Limiti dei principali competitors

Lo studio della letteratura fatto per individuare i competitors ha evidenziato la presenza di sei competitors principali:

- *BigWarp*;
- *Correlia*;
- *Elastix*;
- *ITK*.

Per ognuno dei *software* qui sopra elencati si definiscono i seguenti limiti qui a seguito descritti:

1.3.1 BigWarp

BigWarp ha alcune limitazioni che includono il fatto di non essere in grado di effettuare allineamenti rigidi di qualsiasi tipo manuale o elastico. Questa libreria nasce solo ed esclusivamente per effettuare deformazioni manuali di tipo elastico, di un utente che utilizza questo software dovrebbe continuamente alternare diversi programmi per effettuare trasformazioni rigide e trasformazioni non rigide.

1.3.2 Correlia

Correlia ha alcune limitazioni significative da considerare: come la maggior parte dei competitor elencati, richiede all'utente di specificare le trasformazioni da applicare durante la fase di

allineamento automatico, il che rende il processo particolarmente lento. Altri aspetti negativi da tenere in considerazione includono l'uso di algoritmi obsoleti e la dipendenza da *bUnwarpj*. *Correlia* presenta anche altre limitazioni, come l'incapacità di gestire immagini multicanale e il fatto che ogni volta che viene eseguita un'operazione di allineamento, le immagini di output perdono le proprietà originali, come la profondità di ogni pixel o eventuali LUT. Inoltre, la dimensione di ogni immagine viene impostata a una grandezza prefissata a priori, con il rischio di perdere la qualità dell'immagine originale.

1.3.3 Elastix

Elastix presenta sicuramente come svantaggio la sua complessità d'uso. La procedura di installazione può essere soggetta a errori in quanto richiede il download sia del plugin tramite l'updater di *Fiji*, che comprende solo la GUI, sia del vero e proprio programma dal repository di *GitHub*, con la necessità di specificare il percorso della cartella durante l'esecuzione. Inoltre, l'elenco dei parametri è lungo e poco intuitivo per gli utenti. *Elastix*, come molti altri, non supporta le deformazioni elastiche, non è in grado di gestire immagini multicanale e non conserva le proprietà originali dell'immagine dopo aver eseguito le operazioni di allineamento.

1.3.4 ITK

Uno dei principali difetti di *ITK* è rappresentato dalla sua complessità d'uso. Questo toolkit richiede una conoscenza approfondita della programmazione e una buona comprensione dei concetti matematici sottostanti. Le sue molteplici funzionalità e le numerose opzioni di configurazione possono risultare intimidatorie per gli utenti meno esperti, rendendo necessario un lungo periodo di apprendimento per utilizzarlo efficacemente. Un altro difetto di *ITK* è la configurazione complessa. L'utente deve affrontare una serie di parametri di allineamento e opzioni di configurazione che possono risultare poco intuitivi. La scelta delle impostazioni corrette richiede una conoscenza approfondita delle tecniche di allineamento e un'attenta sperimentazione, rendendo il processo di configurazione laborioso e potenzialmente suscettibile a errori.

1.4 DS4H Image Alignment tool

In questo capitolo viene presentata la versione più recente del plugin *DS4H Image Alignment* che, grazie al primo e secondo modulo sviluppati descritti nell'introduzione:

- a. Include una funzionalità di allineamento automatico di tipo elastico completamente funzionante;

- b. Include una funzionalità di allineamento manuale di tipo elastico completamente funzionante;
- c. Supporta qualsiasi tipo di piattaforma. Questo permette di poter utilizzare il plugin senza dover dipendere da una piattaforma specifica.

1.4.1 Presentazione generale

Il plugin presenta un'interfaccia grafica composta di finestre modali mostrate a seconda della funzionalità richiesta. Le finestre più informative sono:

- *Finestra principale* - rappresenta l'area in cui vengono visualizzate tutte le immagini in uso, oltre a consentire la visualizzazione delle immagini, offre diverse funzionalità, tra cui la modifica degli algoritmi di allineamento automatico e manuale (rigido e elastico), la configurazione dei parametri per il plugin *bUnwarpJ*, la gestione dell'intero progetto di lavoro tramite le opzioni di "import", "export" o "clear", la possibilità di selezionare il target tra tutte le immagini e rimuovere eventuali immagini non necessarie e la possibilità di poter modificare la tipologia di algoritmo elastico;
- *Finestra di preview* – consente di visualizzare un'immagine singola e offre la possibilità di aggiungere o rimuovere punti per l'allineamento e inoltre gestire la dimensione ed il colore dei vari punti specificati;
- *Finestra di output* – visualizza il risultato finale dell'allineamento, offrendo la possibilità di salvare le immagini, riutilizzarle per un successivo allineamento o applicare l'algoritmo di deformazione elastica *bUnwarpJ* a tutte le immagini.

1.5 Modalità d'uso dell'applicazione DS4H

Di seguito vengono illustrate tutte le funzionalità disponibili per gli utenti. Le schermate mostrate in questa sezione sono state catturate utilizzando un computer *MacBook Pro* con il sistema operativo *MacOS Tahoe 26.1* e processore *ARM M2 Pro*.

1.5.1 Caricamento delle immagini

Dopo il lancio del nostro plugin, la prima finestra che si aprirà sarà la finestra principale (Figura 1-5), è possibile osservare la struttura di questa finestra. Per effettuare il caricamento delle immagini si può fare affidamento a due voci presenti nel menu in alto:

- "File";
- "Project".

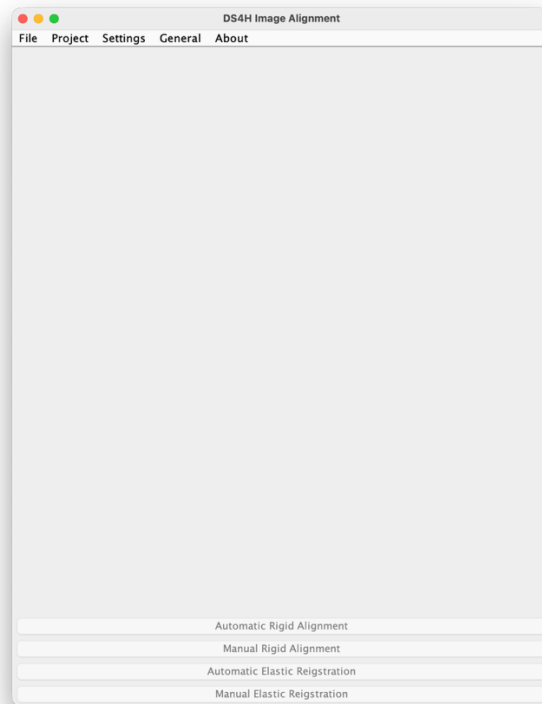


Figure 1-6 Schermata principale

1.5.2 Caricamento delle immagini tramite voce File

Dopo aver selezionato la voce "File" nella barra dei menu, verrà aperta un'altra finestra per il caricamento delle immagini (Figura 1-6), sarà possibile esplorare le cartelle del sistema e selezionare un gruppo di immagini da caricare all'interno del plugin.

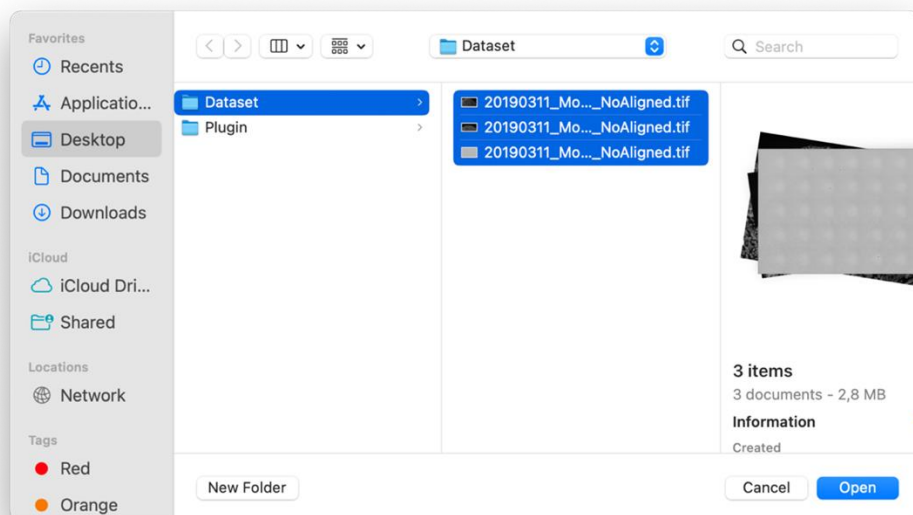


Figure 1-7 Selezione multipla di immagini tramite "File"

1.5.3 Caricamento delle immagini tramite voce Project

In seguito ad aver cliccato su "Project", verrà visualizzato un sottomenu dal quale sarà necessario selezionare "Import". Successivamente, si aprirà una finestra (Figura 1-7), dalla quale sarà possibile scegliere una cartella. Tutte le immagini presenti nella cartella selezionata verranno importate automaticamente.

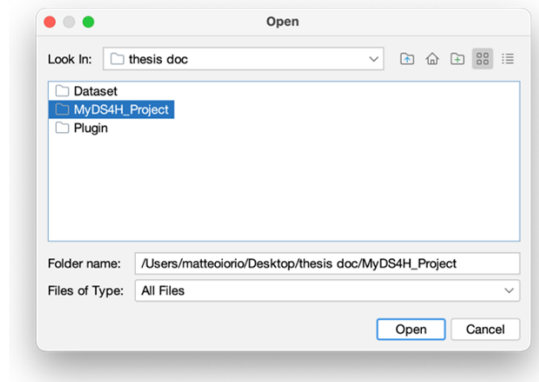


Figure 1-8 Caricamento di immagini tramite "Project > Import"

1.5.4 Impostazioni del plugin

Nel nostro software, diamo anche all'utente la possibilità di personalizzare le impostazioni degli algoritmi automatici, manuali e di deformazione elastica. Se si clicca sulla voce "Settings" nella barra dei menu, si aprirà un sottomenu con tre opzioni:

- 1) *"Manual Rigid Alignment"*: selezionando questa voce, il menù dell'allineamento manuale si aprirà (Figura 1-8) dando possibilità all'utente di configurare:
 - ☐ "Algorithm", per selezionare il tipo di algoritmo di allineamento manuale che si andrà ad utilizzare, varia tra *"Traslativo"*, *"Affine"* e *"Proiettivo"*;
 - ☐ si potrà scegliere se applicare *"Traslazione"*, *"Scala"* e *"Rotazione"*;
 - ☐ "Point Overload", selezionare come gestire eventuale abbondanza di punti nel caso in cui questi non fossero necessari, selezionando *"First point available"*, *"RANSAC"* o *"Minimum Least Square"*.

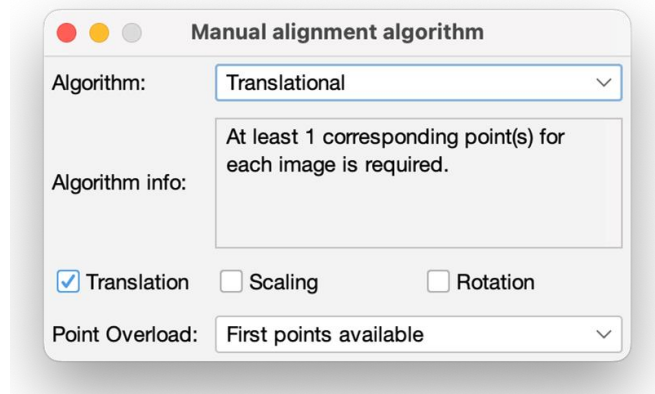


Figure 1-9 Menù di configurazione algoritmo manuale rigido

2) “Automatic Rigid Alignment”: cliccando su tale voce, si aprirà un menu (Figura 1-9) in cui è possibile impostare:

- ☐ “Detector”, il tipo di algoritmo automatico per individuare automaticamente i punti per l’allineamento;
- ☐ “Threshold factor”, per selezionare anche eventuali punti “non ottimi”;
- ☐ “Scaling factor”, ci permette di selezionare di quante volte ogni singola immagine va scalata;
- ☐ “Algorithm”, quale algoritmo si andrà ad utilizzare una volta che si saranno individuati i punti automaticamente, questi algoritmi sono gli stessi del metodo manuale;
- ☐ Se applicare “Traslazione”, “Scala” e “Rotazione”.

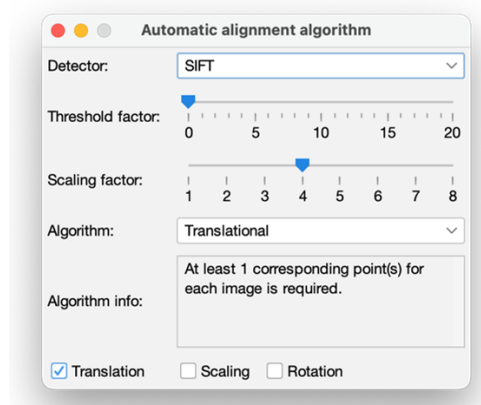


Figure 1-10 Menù di configurazione algoritmo automatico rigido

3) “bUnwarpJ”: selezionando questa voce, sarà possibile configurare i parametri da utilizzare durante l’algoritmo di deformazione elastica (Figura 1-10), tra i tanti parametri che si possono impostare mi soffermo sul più rilevante, “Mode”, in particolare questo ci permette di scegliere tra tre diverse voci:

- “*Mono*”, fa sì che il programma esegua solo una registrazione unidirezionale, cioè dallo sorgente alla destinazione
- “*Accurate*” e “*Fast*”, prevedono la registrazione bidirezionale e influiscono sui criteri di arresto utilizzati internamente dal programma.

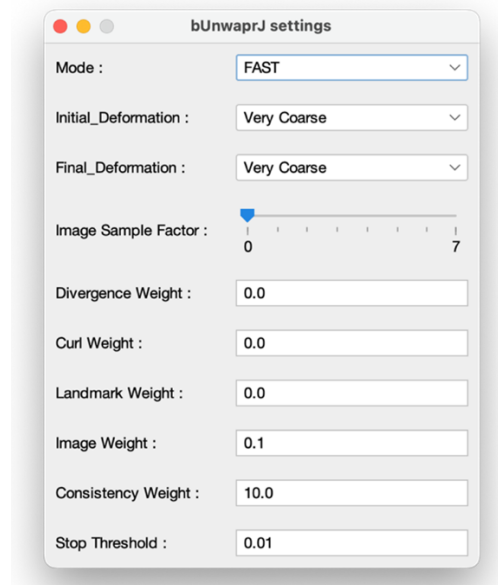


Figure 1-11 Menù di configurazione bUnwarpJ

- 4) “*Elastic Algorithm Definition*”: cliccando su questa voce, verrà data la possibilità all’utente di selezionare il tipo di algoritmo *elastico* da utilizzare durante l’allineamento in modalità manuale o automatica (Figure 1-11). Viene inoltre fornita la possibilità di selezionare il numero massimo di *landmarks* da utilizzare durante la modalità di allineamento elastico in modalità manuale.

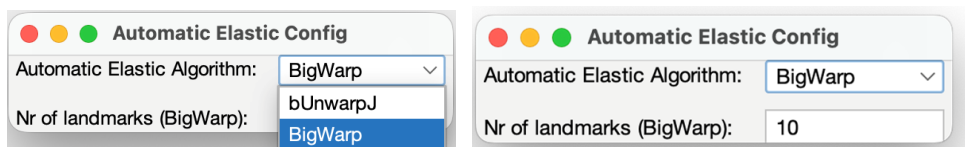


Figure 1-12 Menu di configurazione algoritmi elastici

1.5.5 Scelta immagine target

Dopo aver caricato correttamente le immagini nel nostro plugin, per eseguire l’allineamento manuale ed automatico è necessario selezionare tra tutte le immagini caricate, quale sia l’immagine “Target”. Questa immagine rappresenta la base sulla quale gli allineamenti dovranno essere eseguiti, più nello specifico tutte le immagini verranno alterate in base a questa immagine. Per impostare il “Target” basterà semplicemente cliccare sul bottone “TARGET” (Figura 1-12).

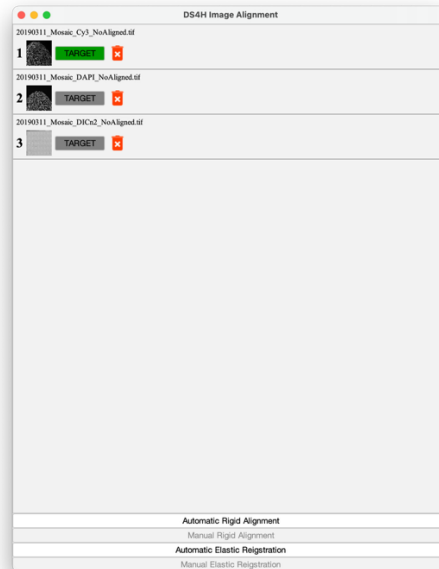


Figure 1-13 Finestra principale con immagini caricate

1.5.6 Rimozione immagine

Nel caso in cui dopo aver caricato correttamente le immagini nel nostro plugin, l'utente nota di aver caricato inconsapevolmente un'immagine non necessaria, potrà rimuoverla dal progetto cliccando sull'icona del cestino rosso vicino al bottone "TARGET" (Figura 1-12). In questo modo l'immagine verrà rimossa completamente dal progetto.

1.5.7 Gestione corner points

Dopo aver caricato correttamente le immagini nel nostro plugin, per eseguire l'allineamento manuale è necessario indicare lo stesso numero di corner points su ciascuna immagine caricata. Il numero minimo di punti richiesti varia a seconda dell'algoritmo selezionato. Per *impostare* i corner point su ciascuna immagine, sarà necessario fare clic sull'immagine desiderata nella finestra principale. Successivamente, si aprirà la *finestra di preview* (Figura 1-13), in cui sarà possibile lavorare sull'immagine selezionata.

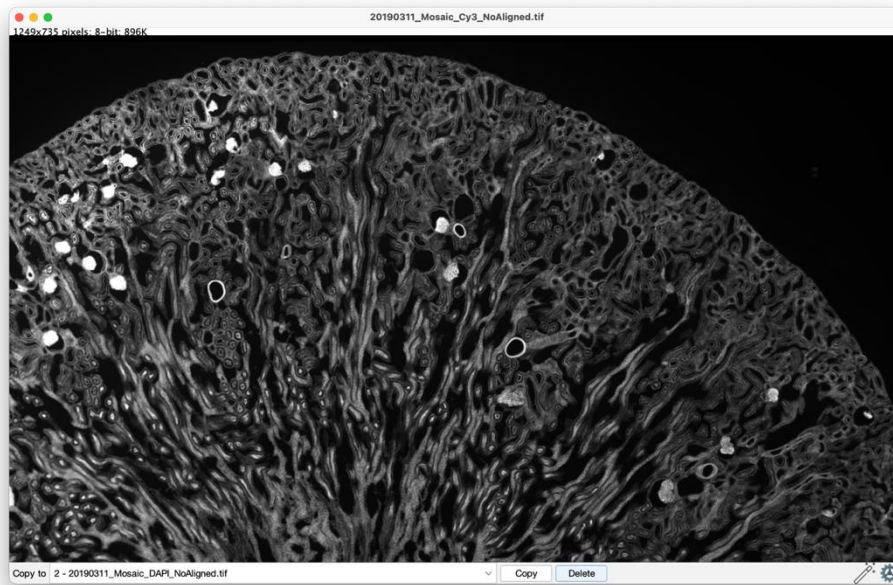


Figure 1-14 Finestra di preview.

1.5.8 Aggiunta dei corner points

Se si preme due volte sul tasto sinistro del mouse all'interno dei limiti dell'immagine corrente, verrà creato un *corner point* nel punto esatto in cui è posizionato il puntatore (Figura 1-14). Si può vedere il risultato di questa operazione, con la creazione di tre corner points. L'ordine di creazione dei punti è evidenziato grazie alla numerazione automatica.

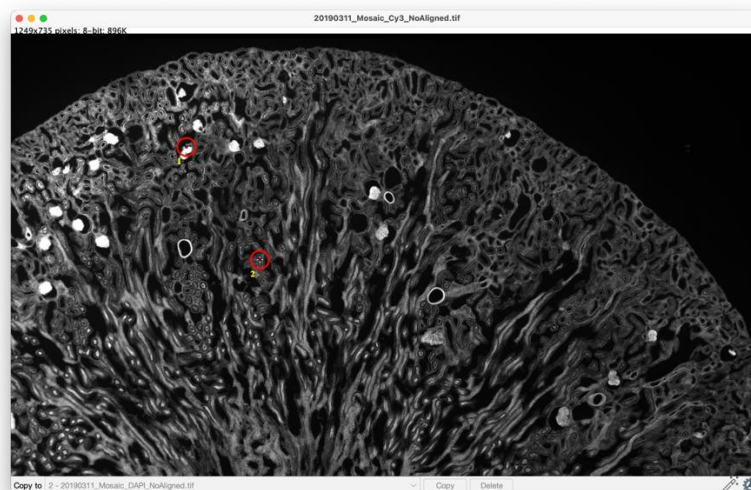


Figure 1-15 Finestra di preview in cui sono stati aggiunti 3 corner

1.5.9 Selezione singola o multipla dei corner points

Come detto precedentemente è possibile inserire anche più corner all'interno della stessa immagine. È possibile selezionarli singolarmente (Figura 1-15) o in modo multiplo (Figura 1-16). Quando si

selezionano più corner, è possibile spostarli tutti insieme, lo spostamento dei corner points è limitato nei confini dell'immagine, così facendo l'utente non potrà inserire i corner points all'esterno dei confini dell'immagine su cui si sta lavorando.

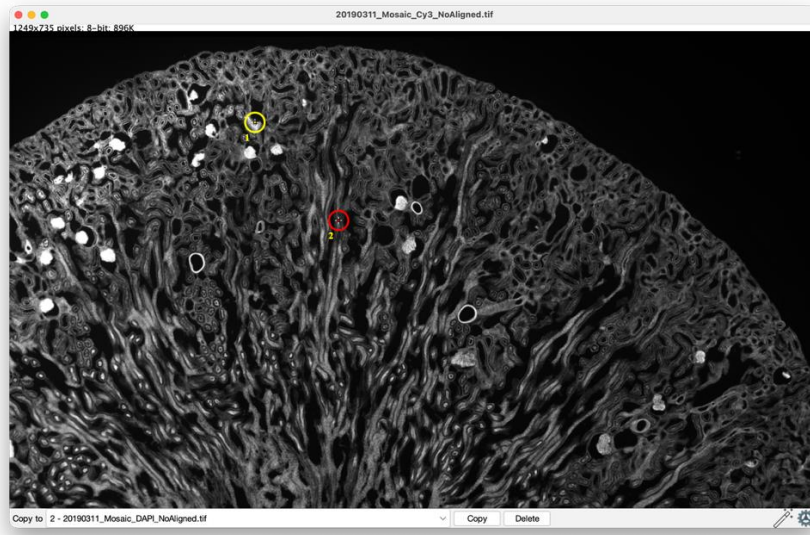


Figure 1-16 Esempio selezione singola

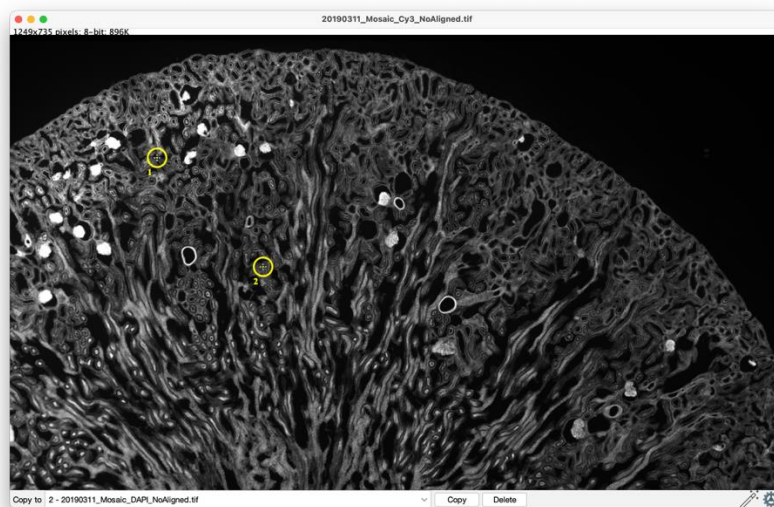


Figure 1-17 Esempio selezione multipla

1.5.10 Copia dei corner points

Al fine di facilitare l'uso del nostro plugin, abbiamo implementato anche la funzionalità di copiare i corner points inseriti su un'immagine e incollarli su un'altra. Per iniziare, è necessario selezionare almeno un corner point all'interno dell'immagine. Successivamente, sarà possibile scegliere l'immagine di destinazione in cui si desidera incollare i corner points, selezionandola dal menù a tendina nella voce "copy to" (Figura 1-17). Infine, cliccando sul pulsante "Copy", l'operazione verrà

completata. Nel caso in cui alcuni dei corner point copiati dovessero finire al di fuori dell'immagine, verrà notificato all'utente (come mostrato nella Figura 1-18) e il corner point non verrà inserito nell'immagine di destinazione.

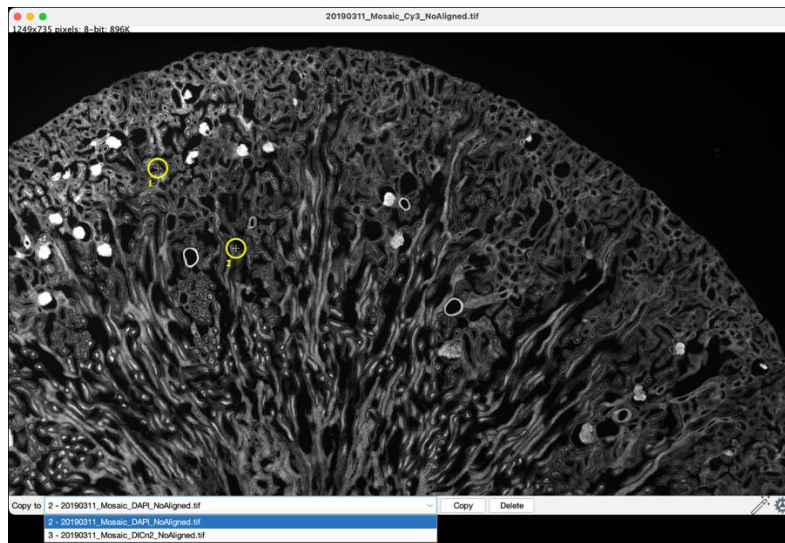


Figure 1-18 Esempio di selezione copia di corner point.

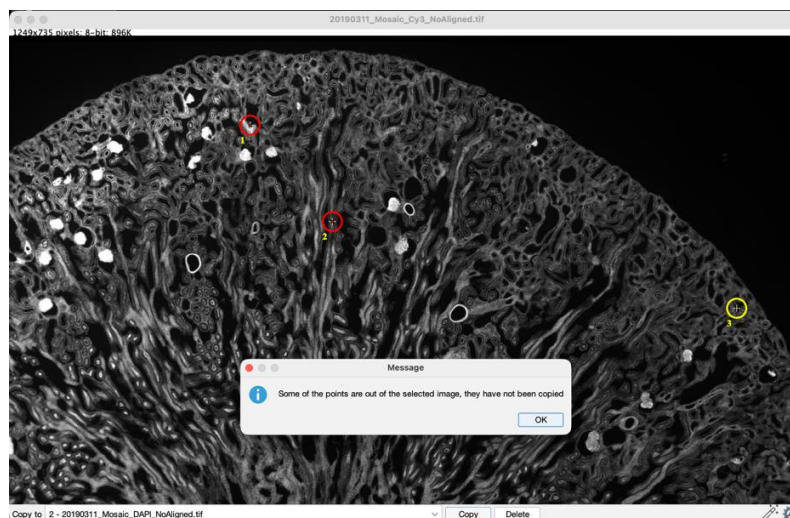


Figure 1-19 Esempio di copia di corner point fuori dall'immagine

1.5.11 Rimozione dei corner points

Dopo aver selezionato uno o più corner points, è possibile rimuoverli completamente dall'immagine premendo il pulsante "Delete" (Figura 1-19) con il cursore. Ciò comporterà la completa eliminazione dei corner point selezionati dall'immagine (Figura 1-20) e automaticamente gli indici dei corner points verranno scalati per adattarsi alla nuova configurazione.

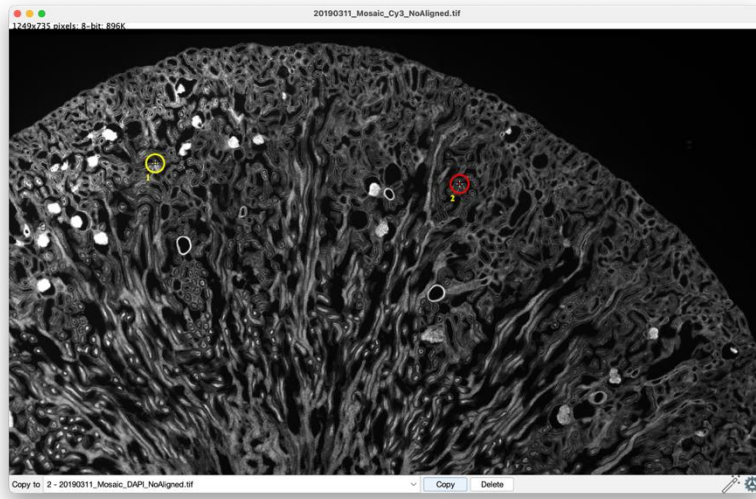


Figure 1-20 Esempio selezione corner point 1 per la rimozione

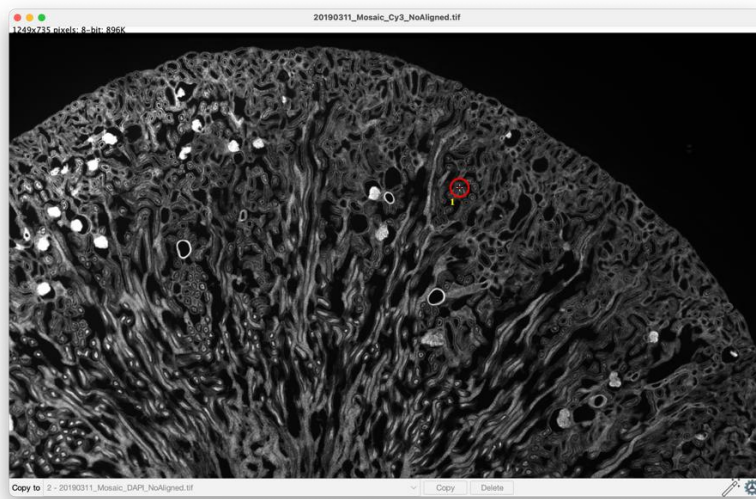


Figure 1-21 Risultato dopo aver eliminato il punto 1

1.5.12 Modifica dei corner points

Nel caso in cui il colore dei corner points produce un contrasto poco significativo o se si desidera modificare l'ordine tra due corner points inseriti, è possibile apportare modifiche sia all'aspetto e sia all'ordine di ciascun corner point. Ciò può essere fatto facendo clic sull'icona delle impostazioni, che aprirà la finestra delle impostazioni per i corner points inseriti (Figura 1-21). Attraverso questa finestra è possibile:

- ☐ Variare il colore del corner;
- ☐ Cambiare il colore di selezione del corner;
- ☐ Alterare il colore dell'indice del corner;
- ☐ Modificare la dimensione del corner;
- ☐ Sostituire il numero dell'indice di un corner point con quello di un altro corner inserito

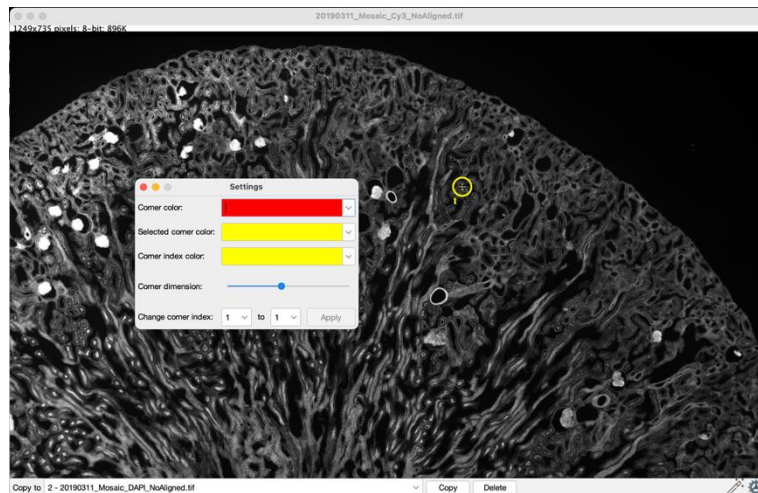


Figure 1-22 Finestra di settings per i punti impostati

1.5.13 Esportare il progetto

Una volta che l'utente ha caricato le immagini, può eseguire l'operazione di "Project > Export" per esportare l'intero progetto di lavoro. In particolare, utilizzando questa funzione, l'utente può salvare tutte le immagini con i rispettivi punti impostati su ciascuna immagine, oltre al target selezionato, in una qualsiasi cartella del proprio sistema. Nel momento in cui si clicca la voce "Export", verrà chiesto all'utente (Figura 1-22) in che percorso effettuare il salvataggio del progetto, automaticamente, verrà creata una cartella chiamata "*YYYYMMDD_HHMM_DS4H_Project*" (con *YYYYMMDD* indicante l'anno il mese e il giorno del salvataggio, *HHMM* indicante l'ora ed il minuto) contenente tutte le immagini e un singolo file in formato JSON contenente tutte le informazioni sul "target" e sui corner points.

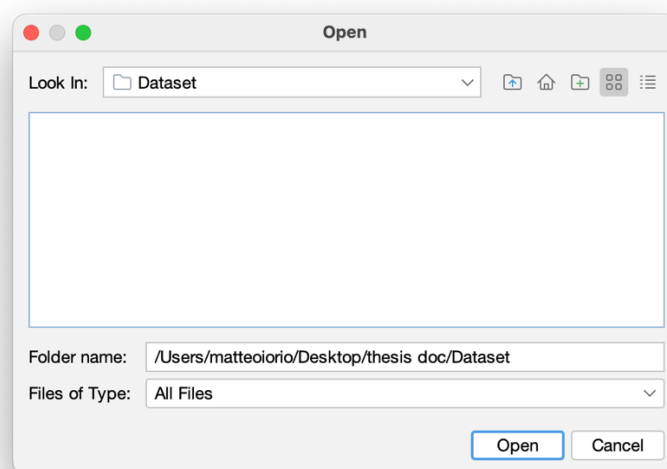


Figure 1-23 Esempio di export di un progetto

1.5.14 Allineamento manuale rigido

Una volta che su ogni immagine sono stati specificati i corner point, sarà possibile allinearle cliccando sul tasto “Manual Rigid Alignment” presente nella *finestra principale*. Ora che l’allineamento è partito verrà visualizzata una finestra che ci permetterà di capire a che punto siamo con il nostro allineamento (Figura 1-23).



Figure 1-24 Finestra di loading

Quando l’allineamento sarà terminato, verrà visualizzata la *finestra di output*, la quale contiene uno stack scorribile con tutte le immagini allineate in base al target selezionato (Figura 1-24).

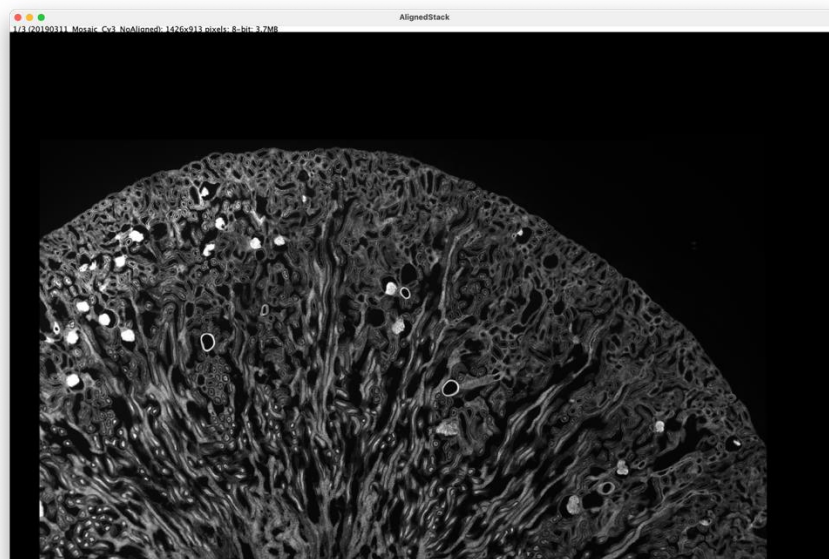


Figure 1-25 Finestra di output

Nel caso in cui volessimo vedere le nostre immagini sovrapposte, dopo averle allineate, ci basterà adoperare *ImageJ/Fiji*, così facendo potremo ottenere all'interno dello stesso stack le varie immagini allineate sovrapposte (Figura 1-25).

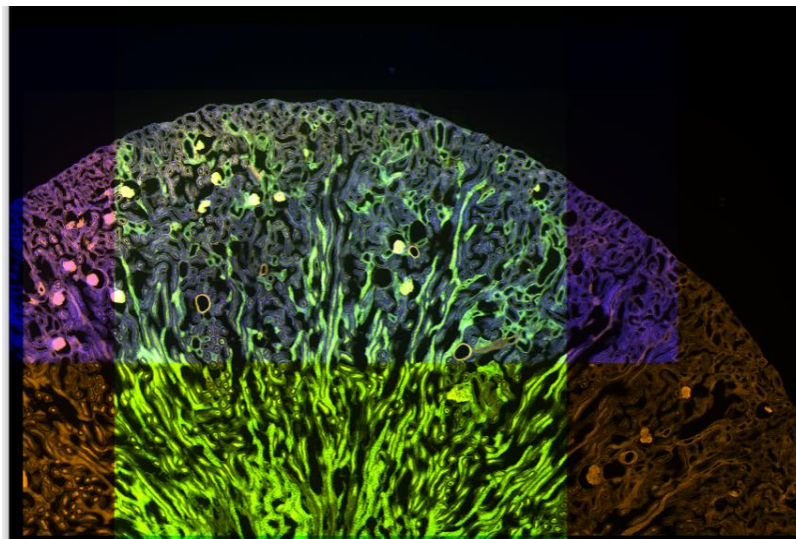


Figure 1-26 Finestra di output, con immagini allineate sovrapposte

1.5.15 Allineamento manuale elastico

Come descritto in precedenza è possibile utilizzare anche un allineamento manuale di tipo elastico sulle immagini caricate all'interno del progetto. Prima di poter avviare questa procedura è necessario che l'utente individui un insieme di corner point per ogni immagine, successivamente poi il bottone “*Manual Elastic Alignment*” si attiverà rendendo possibile l'utilizzo di questa procedura. Una volta che l'allineamento manuale elastico viene avviato, verrà visualizzata una finestra di caricamento che mostrerà lo stato dell'allineamento (Figura 1-23). Quando l'allineamento sarà terminato, verrà visualizzata la *finestra di output*, la quale contiene uno stack scorribile con tutte le immagini allineate in base al target selezionato (Figura 1-24). Questa procedura, dunque, permette di correggere eventuali errori dovuti a deformazioni elastiche, in modo tale da poter riutilizzare le stesse immagini per gli allineamenti di tipo rigido, andando ad aumentare la qualità dell'output.

1.5.16 Salvataggio delle immagini

Una volta che abbiamo ottenuto il nostro stack allineato, possiamo effettuare una serie di operazioni, tra cui il *salvataggio* delle immagini, per fare ciò basterà semplicemente cliccare sulla voce “Save” nel menu bar e selezionare quali tra le immagini di output si vorranno salvare (Figura 1-26). Dopo aver scelto quali immagini salvare verrà chiesto in che directory salvare le immagini, in questo modo

nel percorso specificato troveremo una cartella denominata “YYYYMMDD_HHMM_DS4H_AlignedImages”.

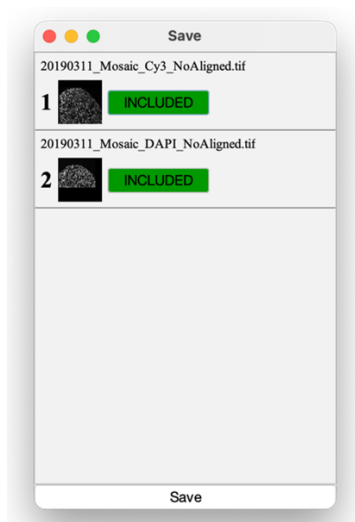


Figure 1-27 Finestra per il salvataggio delle immagini

1.5.17 Riutilizzo delle immagini

Dopo aver allineato le immagini, oltre al salvataggio, si può eseguire un'ulteriore operazione che consiste nell'utilizzare tutte o alcune delle immagini per ulteriori allineamenti utilizzando l'opzione "Reuse" nel menu bar. Una volta selezionata questa voce, verrà richiesto di selezionare le immagini che si desiderano utilizzare per questa operazione (Figura 1-27).

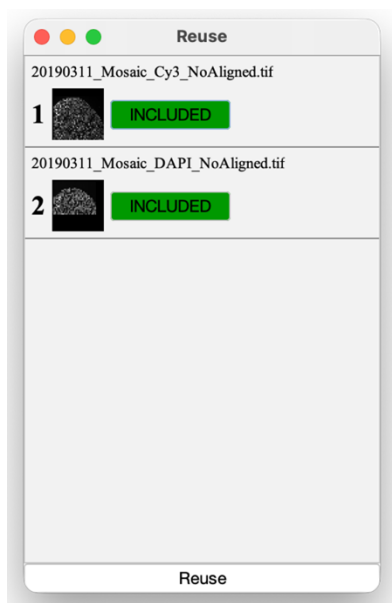


Figure 1-28 Finestra per il riutilizzo delle immagini

1.5.18 Applicazione algoritmo di deformazione elastica dopo allineamento rigido

Dopo aver allineato le immagini, si può eseguire l'ultima operazione che consiste nell'applicare l'algoritmo di deformazione elastica *bUnwarpJ*. Per farlo, si deve semplicemente cliccare sull'opzione "*Elastic*" nel menu. In automatico, il nostro programma utilizzerà le impostazioni configurate nella finestra principale come parametri di input per l'algoritmo.

1.5.19 Allineamento automatico rigido

Dopo aver caricato correttamente le immagini nel nostro plugin, si può facilmente utilizzare la funzionalità di allineamento automatico cliccando sul pulsante "Automatic Rigid Alignment". Verrà visualizzata immediatamente una finestra di caricamento che mostrerà lo stato di avanzamento dell'allineamento (Figura 1-23). Una volta completato con successo l'allineamento, verrà mostrata la *finestra di output* (Figura 1-24).

1.5.20 Allineamento automatico di tipo elastico

Successivamente al caricamento delle immagini da parte dell'utente, è possibile utilizzare, senza dover inserire alcun corner point, l'algoritmo automatico di deformazione elastica, tramite un semplice click sulla voce "Automatic Elastic Alignment". Una volta eseguita questa operazione, verrà visualizzata la finestra di caricamento che mostrerà lo stato di avanzamento dell'allineamento (Figura 1-23). Dopo che l'allineamento avrà terminato la propria esecuzione, il risultato verrà mostrato tramite la *finestra di output* (Figure 1-24).

2 Supporto alle architetture ARM

In questo capitolo viene presentato il primo modulo di sviluppo della tesi, dedicato alla compilazione della libreria *OpenCV* con supporto per architetture di tipo *ARM*. La scelta di utilizzare *OpenCV* nasce dalla sua natura di libreria di *computer vision* tra le più complete e ampiamente adottate in ambito accademico e industriale. Nel seguito del capitolo verranno illustrati, in modo dettagliato, tutti i passaggi e i comandi necessari per abilitare il supporto alle architetture *ARM*, così da garantire una base solida per i moduli di sviluppo successivi.

2.1 Computer Vision

La Computer Vision è una disciplina all'interno del campo dell'Intelligenza Artificiale (AI) che permette alle AI di prendere decisioni basate su ciò che "vedono". L'obiettivo principale della Computer Vision è quello di imitare il nostro senso della vista. Analogamente a noi esseri umani, che fin dalla nostra infanzia abbiamo imparato a classificare e distinguere informazioni attraverso retine, nervi ottici e la corteccia visiva, la Computer Vision utilizza algoritmi per raggiungere lo stesso scopo [17].

2.2 OpenCV

OpenCV è disponibile in forma precompilata per *Android*, *iOS* e *Windows*, ma per qualsiasi altra piattaforma (o quando si desidera utilizzare la libreria *OpenCV Contrib*) è necessario compilare il codice sorgente personalmente. Per implementare il modulo di allineamento automatico delle immagini, ho dovuto scaricare entrambi i sorgenti e generare un file *JAR* e le relative librerie per le piattaforme di esecuzione, ad esempio i file *.dll per *Windows*, i file *.so per *Linux* e i file *.dylib per *MacOS X* con architettura *ARM*. Per questo processo, è necessario avere una distribuzione di *Java* installata nel sistema operativo, scaricare il programma "*CMake*" e utilizzare un terminale. Dopo aver scaricato le librerie *OpenCV* e *OpenCV Contrib* in formato zip e averle estratte, è utile mantenere entrambe le cartelle risultanti all'interno di una cartella padre per comodità. Successivamente, è necessario aprire il terminale e impostare la cartella padre come percorso corrente. Il nostro plugin include già le librerie *OpenCV* precompilate e le caricherà automaticamente in una cartella temporanea del sistema operativo. In questo modo, sarà possibile utilizzare *OpenCV* senza incontrare problemi, poiché la libreria specifica per il sistema operativo corrente verrà gestita in modo automatico.

2.2.1 OpenCV Extra Features Library

OpenCV include numerosi moduli, tutti utilizzabili senza alcuna licenza, come gli algoritmi: *AKAZE*, *KAZE*, *BRISK* e *ORB*, mentre altri algoritmi sono soggetti ad alcune eccezioni. Queste eccezioni sono racchiuse nella libreria chiamata "*OpenCV Contrib*", che contiene moduli con funzionalità che potrebbero non essere stabili o che richiedono una licenza a pagamento per l'uso commerciale, ma non per prodotti successivamente resi open-source. All'interno di questa libreria sono presenti due degli algoritmi utilizzati in questo lavoro di Tesi per creare il modulo di allineamento automatico delle immagini: *SURF* e *SIFT*. [18]

2.2.2 Compilazione di OpenCV

Come prima cosa è stato necessario scaricare i sorgenti di "*OpenCV*" e "*OpenCV contrib*" da *GitHub* tramite il comando mostrato in Codice 2-1.

```
git clone -b 4.5.5 git@github.com:opencv/opencv_contrib.git  
git clone -b 4.5.5 git@github.com:opencv/opencv.git
```

Codice 2-1 Clonazione del repository OpenCV e OpenCV contrib

L'utilizzo del sorgente "*OpenCV contrib*" è necessario per l'utilizzo di alcune funzionalità extra come *SIFT*, non presenti nella distribuzione principale. Una volta scaricati i sorgenti, bisogna creare una directory destinata a ospitare tutti gli output generati in fase di compilazione. Prima di procedere con la configurazione dell'ambiente, è stato necessario installare la corretta versione di *Java*, così da consentire la generazione dei file ".jar" necessari al funzionamento del plugin. Per questo scopo è stato utilizzato *jabba*, un tool esterno per gestione di varie *JVM*, attraverso il quale è stata installata la versione di *Java* (Codice 2-2) [19]:

```
jabba install  
1.8.0=tgz+https://download.bell-sw.com/java/8u452+11/bellsoft-jdk8u452+11-macos-aarch64.tar.gz
```

Codice 2-2 Codice per scaricare Java

L'utilizzo della corretta versione della *JVM* garantisce la generazione di artefatti *OpenCV* perfettamente compatibili con il plugin sviluppato. Successivamente si è passati alla fase di configurazione tramite *Cmake*, specificando esplicitamente – tramite l'apposito flag [20] – il percorso dei moduli aggiuntivi di *OpenCV Contrib*, come mostrato nel Codice 2-3 .

```

cmake -DCMAKE_BUILD_TYPE=Release \
-DWITH_PYTHON=OFF \
-DPYTHON_EXECUTABLE=NOTFOUND \
-DCMAKE_INSTALL_PREFIX=$HOME/Desktop/opencv/output \
-DOPENCV_EXTRA_MODULES_PATH=../opencv_contrib/modules \
-DBUILD_opencv_java=ON \
-DBUILD_opencv_java_bindings_generator=ON \
-DOPENCV_ENABLE_NONFREE=ON \
-DBUILD_SHARED_LIBS=ON \
-DCMAKE_CXX_FLAGS="-O3 -march=native" \
-DCMAKE_C_FLAGS="-O3 -march=native" \
-DBUILD_TESTS=OFF \
-DBUILD_ZLIB=OFF \
-DBUILD_PERF_TESTS=OFF \
-DBUILD_EXAMPLES=OFF \
-DBUILD_opencv_python3=OFF \
-DBUILD_opencv_python_bindings_generator=OFF \
-DCMAKE_JAVA_HOME=$JAVA_HOME \
-DJAVA_AWT_INCLUDE_PATH=$JAVA_HOME/include \
-DJAVA_AWT_LIBRARY=$JAVA_HOME/lib/libjawt.dylib \
-DJAVA_INCLUDE_PATH=$JAVA_HOME/include \
-DJAVA_INCLUDE_PATH2=$JAVA_HOME/include/darwin \
-DJAVA_JVM_LIBRARY=$JAVA_HOME/lib/server/libjvm.dylib \
-DBUILD_PNG=OFF \
../opencv

```

Codice 2-3 cmake command

Completata la configurazione dei sorgenti è stata eseguita mediante il comando *make*, sfruttando sette *thread* per velocizzare il processo (Codice 2-4):

make -j7

Codice 2-4 make con 7 thread

L'intera procedura ha permesso non solo di generare correttamente i file jar necessari, ma anche di garantire il supporto ai dispositivi basati su architettura *ARM*. Tale problematica era stata precedentemente segnalata anche da un utente esterno tramite l'apertura di una issue su *GitHub*, confermando quindi la rilevanza della soluzione adottata.

3 Implementazione di Algoritmi di deformazione Elastica

Il seguente capitolo rappresenta il modulo *core* di tutta la mia tesi, ovvero il modulo responsabile dell'implementazione di algoritmi di deformazione elastica di tipo manuale e automatico all'interno del plugin. Nel seguente capitolo verrà esplorato il tipo di algoritmo utilizzato e le modalità con la quale è stato possibile realizzare la modalità automatica e manuale. Questo capitolo il particolare tratta gli algoritmi di *deformazione elastica* impiegati per riportare un'immagine deformata su una immagine di riferimento (registrazione non rigida), il capitolo spiega i passi implementativi per introdurre questo tipo di algoritmo all'interno del plugin.

3.1 Deformazioni elastiche su immagini

Gli algoritmi di deformazione elastica rappresentano una categoria di metodi utilizzati per allineare due immagini che presentano differenze locali di *forma*. A differenza delle trasformazioni rigide o affini, limitate a traslazioni e scalamenti, le deformazioni elastiche permettono di modellare variazioni più complesse e realistiche, adattando un'immagine *mobile* ad una di *riferimento*. L'obiettivo è ottenere una corrispondenza più accurata tra strutture omologhe nelle due immagini, garantendo al tempo stesso la *coerenza geometrica* della trasformazione. Ad un livello astratto, la deformazione elastica può essere vista come una funzione che associa a ogni punto uno spostamento che lo porta verso la posizione più coerente nell'immagine di riferimento. Il processo di spostamento può essere suddiviso in tre fasi principali:

- 1) *Misurazione della differenza* tra le due immagini: viene adottata una metrica di *similarità* che quantifica quanto un pixel differisca dalla corrispondente posizione nell'altra immagine;
- 2) *Calcolo del campo di deformazione*: l'algoritmo cerca la trasformazione che minimizza tali differenze. Tale trasformazione non è rigida: può assumere forme complesse, adattandosi localmente, come se l'immagine fosse costituita da un materiale elastico capace di deformarsi;
- 3) *Regolarizzazione della deformazione*: per evitare soluzioni irrealistiche, come distorsioni troppo brusche, il campo di deformazione viene vincolato con un modello matematico che ne garantisce la *plausibilità fisica*. Questo passaggio rende la deformazione *elastica*, l'immagine si deforma come un materiale elastico, con resistenze intrinseche a stiramenti o contorsioni eccessive.

Questa tipologia di algoritmo è particolarmente utile nel campo della *istologia*, dal momento in cui è possibile che durante dei campionamenti si possano andare a creare delle variazioni strutturali

significative sul campione sotto osservazione. La capacità di adattarsi localmente rende questo approccio più preciso rispetto ai modelli rigidi, soprattutto in presenza di deformazioni non uniformi.

3.1.1 Libreria per la deformazione elastica di immagini

Come evidenziato in precedenza, il plugin è sviluppato utilizzando la versione di *Java* 8, questo ha reso più complesso la ricerca (un possibile sviluppo) di un algoritmo di deformazione elastica, dal momento in cui le principali registrazioni elastiche sono più comuni nei linguaggi *C++* e *Python*. Qui di seguito vengono elencate alcune delle librerie/strumenti *Java* per effettuare questi tipi di deformazioni:

- *UnwarpJ*: plugin per *ImageJ* scritto completamente in *Java*, permette le registrazioni di tipo elastico usando *spline cubiche*;
- *bUnwarpJ*: questo plugin rappresenta l'evoluzione di *UnwarpJ*, sempre per *ImageJ/Fiji*, implementa deformazioni elastiche rappresentate da **B-Spline** e assicura una certa *invertibilità* della trasformazione tramite restrizioni di consistenza;
- *SimpleElastix* (tramite *SimpleITK*): non è puramente scritto in *Java* ma *SimpleITK* permette di fornire i binding per *Java*, permette di usare diversi algoritmi di registrazione, compresi quelli deformabili (elastici) grazie all'uso della libreria *Elastix*.
- *BigWarp*: plugin per *ImageJ/Fiji* implementa deformazioni di tipo elastic

3.1.2 Utilizzo della libreria BigWarp

BigWarp è un plugin per le piattaforme *Fiji/ImageJ* progettato per la registrazione *non lineare* di immagini, con una grande attenzione ai dataset di grandi dimensioni (tipici di imaging biologico e biomedico). È stato sviluppato con l'obiettivo di fornire uno strumento interattivo, scalabile e ad alte prestazioni per l'allineamento elastico basato su punti di controllo.

A differenza di altri strumenti di registrazione di tipo *automatica*, *BigWarp* si basa su un approccio interattivo guidato dall'utente, consentendo di definire corrispondenze anatomiche o morfologiche attraverso coppie di *landmark* posizionati manualmente. Questo risulta affidabile quando la registrazione automatica risulta inaffidabile. La registrazione elastica è ottenuta tramite l'uso di **Thin Plate Spline**, un modello di interpolazione morbido che consente di stimare un campo di deformazione a partire dai punti di controllo definiti nella coppia di immagini. Proprio per questi motivi è stata scelta questa libreria per implementare l'algoritmo di deformazione elastica.

3.2 Descrizione di Thin Plate Splines

Le **Thin Plate Splines (TPS)** sono una modalità di interpolazione dei dati basata su tecniche *spline*. Questo algoritmo costituisce uno dei modelli più utilizzati per la stima di deformazioni non lineari a partire da corrispondenze geometriche, in particolare nella registrazione di immagini biomediche. Si tratta di un approccio interpolativi basato su una formulazione che garantisce deformazioni lisce e globalmente coerenti, risultando particolarmente adatto a contesti in cui la deformazione deve riprodurre spostamenti morfologici realistici o anatomici (Figura 3-1) [21].

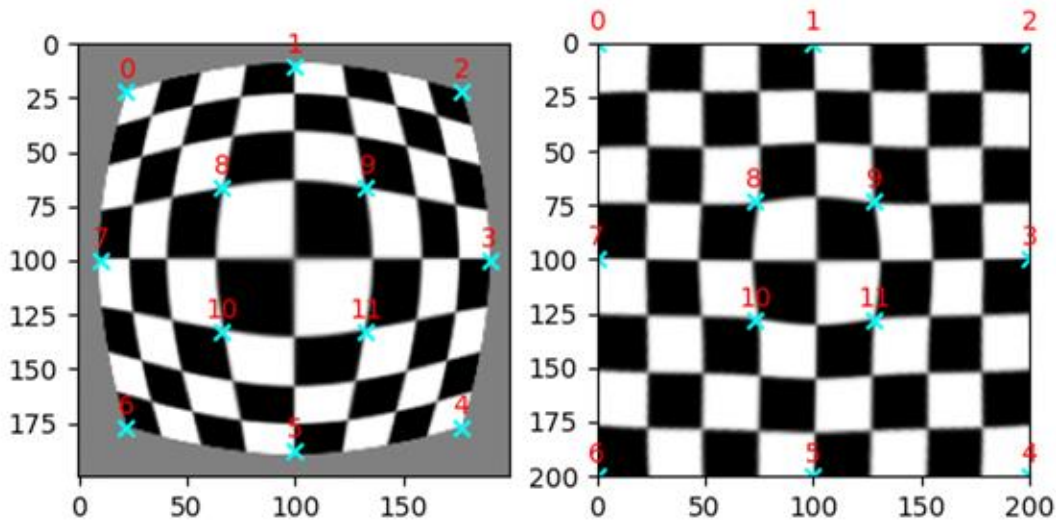


Figure 3-29 Deformazione elastica tramite TPS

3.2.1 Formulazione matematica di Thin Plate Spline

Qui di seguito viene spiegato nel dettaglio la formulazione delle **Thin Plate Spline** [22], le quali rappresentano una classe di trasformazioni non lineari regolarizzate impiegate per modellare deformazioni lisce tra due domini *Euclidei*. Formalmente le **Thin Plate Spline** possono essere definite come la soluzione al problema variazionale di trovare la mappatura più liscia che interpoli un insieme di landmark sparsi tra punti sorgente e punti destinazione, Siano dati:

- Un insieme di **punti sorgente**

$$\{X_i\}_{i=1}^N, \text{ con } x_i \in R^2$$

- Un insieme di **punti destinazione**

$$\{Y_i\}_{i=1}^N, \text{ con } y_i \in R^2$$

Tali che si desideri determinare una mappa: $f: R^2 \rightarrow R^2$ che soddisfi:

$$f(x_i) = y_i, \forall i = 1, \dots, N$$

Tra tutte le funzioni che soddisfano le condizioni di interpolazione, le **TPS** selezionano quella che *minimizza l'energia di curvatura* definito sul dominio come:

$$\Sigma[f] = \iint_{R^2} \left(\frac{\partial^2 f}{\partial x^2} \right)^2 + 2 \left(\frac{\partial^2 f}{\partial x \partial y} \right)^2 + \left(\frac{\partial^2 f}{\partial y^2} \right)^2 dx dy$$

Il funzionale penalizza le variazioni di curvatura della mappa, privilegiando deformazioni globalmente lisce. La minimizzazione di tale funzione sotto vincoli di interpolazione porta a una soluzione analitica in forma chiusa, che risulta essere della forma:

$$f(x) = a_0 + A_x + \sum_{i=1} w_i U(\|x - x_i\|)$$

Dove:

- $a_0 \in R^2$ rappresenta la traslazione
- $A \in R^{2 \times 2}$ è la componente
- $w_i \in R^2$ sono i coefficienti dei nuclei radiali,
- $U(r)$ è la funzione di base radiale TPS definita come:

$$U(r) = r^2 \log(r^2)$$

Questa specifica funzione di base deriva come soluzione fondamentale dell'operatore biarmonico in due dimensioni, ed è ciò che conferisce alle *TPS* la caratteristica di modellare lisce e minimizzare l'energia di curvatura.

3.3 Implementazione algoritmi di deformazione elastica in DS4H

Qui di seguito viene fornita la modalità attraverso la quale è stato possibile implementare gli algoritmi di deformazione elastica in modalità manuale ed automatica.

3.3.1 Design delle classi

Per introdurre il nuovo meccanismo di allineamento elastico all'interno del plugin, è stato necessario effettuare un *refactor* generale delle varie classi che costituiscono le immagini. Questo *refactor* è stato necessario per facilitare l'introduzione nei nuovi meccanismi per le deformazioni elastiche. In particolare, sono andati a sviluppare tre tipi di interfacce diverse:

- **PointRepository**: questa interfaccia definisce un'astrazione per la gestione strutturata di punti bidimensionali rilevati all'interno di un processo di analisi delle immagini. Essa fornisce le operazioni fondamentali per l'inserimento, la consultazione e la manipolazione dell'insieme dei punti memorizzati, garantendo un accesso coerente e indipendente dall'implementazione concreta del repository;
- **DataImage**: l'interfaccia *DataImage* rappresenta un livello astratto di accesso ai dati grezzi e ai metadati associati a un'immagine digitale. Essa è progettata per fornire un punto di ingresso uniforme alle operazioni di pre-processing dell'immagine;
- **AnalyzableImage**: questa interfaccia estende le interfacce *PointRepository* e *DataImage*, combinando in un'unica struttura le funzionalità di gestione dei punti rilevati e l'accesso alle informazioni. Essa rappresenta quindi una componente centrale nei modelli di analisi, poiché fornisce un'astrazione completa dell'immagine come entità analizzabile.

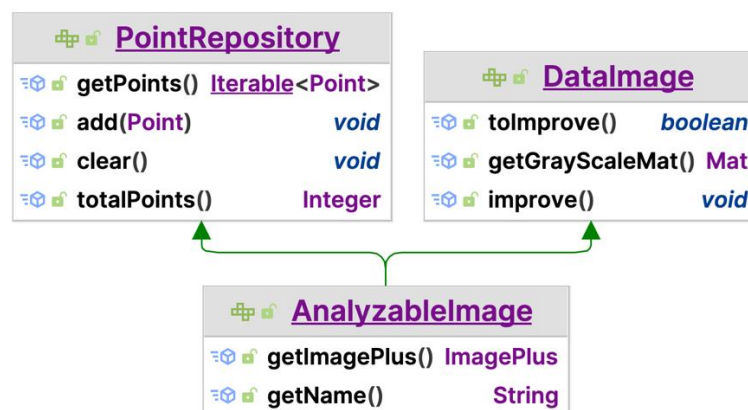


Figure 3-30 Interfacce principali di un'immagine del plugin

3.3.2 Algoritmo elastico manuale

All'interno di questo sotto capitolo viene descritto come utilizzare l'algoritmo di deformazione elastica in modalità manuale.

3.3.2.1 Numero di punti di controllo necessari per la deformazione elastica

L'allineamento elastico manuale delle immagini mediante *landmark based registration* richiede la definizione di una serie di punti di controllo corrispondenti tra immagine di riferimento e immagine da trasformare. La qualità e la stabilità della deformazione dipendono in maniera diretta sia dal numero di punti inseriti, sia dalla loro distribuzione spaziale:

- L'inserimento di un solo punto di controllo consente esclusivamente una traslazione locale dell'immagine, in questo caso l'algoritmo di deformazione elastica non è in grado di stimare una trasformazione geometrica complessa, non permettendo rotazioni, né cambiamenti di scala;
- L'utilizzo di due punti di controllo permette di definire una trasformazione che oltre alla traslazione, può includere anche una rotazione e una variazione di scala lungo l'asse definito dai due punti; tuttavia, non è sufficiente per una vera deformazione elastica;
- L'utilizzo di tre punti di controllo permette di avere una trasformazione completa *affina* nel piano 2D, non rappresenta comunque una deformazione elastica accurata, specialmente in presenza di deformazioni non uniformi.

Una deformazione elastica richiede un numero maggiore di punti di controllo, in genere almeno 5-10 punti, distribuiti omogeneamente nell'immagine. Con un numero sufficiente di landmarks, è possibile costruire una trasformazione non lineare in grado di adattarsi localmente alle differenze tra le due immagini.

3.3.2.2 Come utilizzare l'algoritmo di deformazione elastica

La prima modalità di applicazione dell'algoritmo di deformazione elastica è stata quella in modalità *manuale*. Questo tipo di modalità rappresenta un approccio interattivo particolarmente utile nei contesti in cui le differenze tra le immagini sono tali da rendere inaffidabili o insufficienti le tecniche di allineamento elastico di tipo *automatico*. In questo paradigma, l'utente guida il processo di deformazione definendo un insieme di corrispondenze spaziali (landmark) tra l'immagine sorgente e l'immagine di riferimento. L'idea alla base di questo metodo è di consentire all'utente di intervenire direttamente nella rilevanza, dove le relazioni geometriche possono essere complesse o difficili da catturare automaticamente. Una volta specificare le coppie di punti, il sistema calcola una deformazione continua e liscia che interpola fedelmente i vincoli imposti, propagando la trasformazione anche alle aree circostanti mediante un modello elastico.

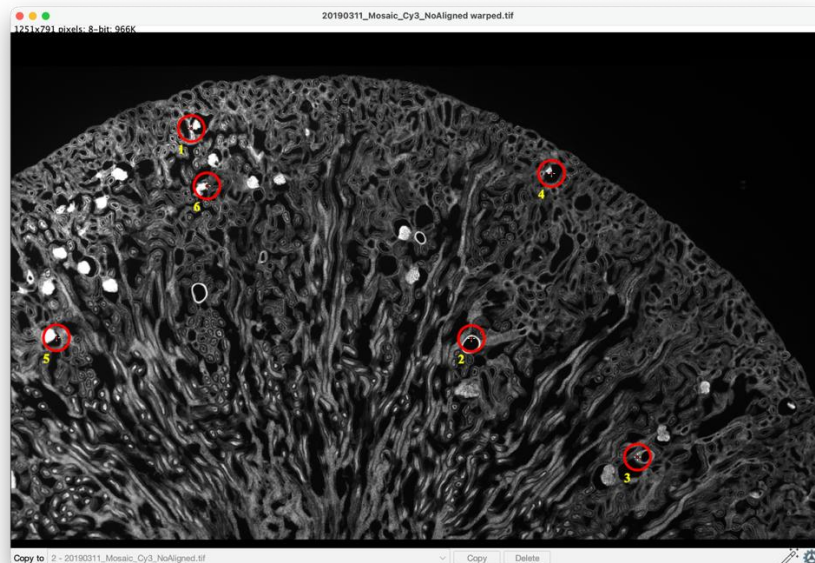


Figure 3-31 Landmark su immagine sorgente

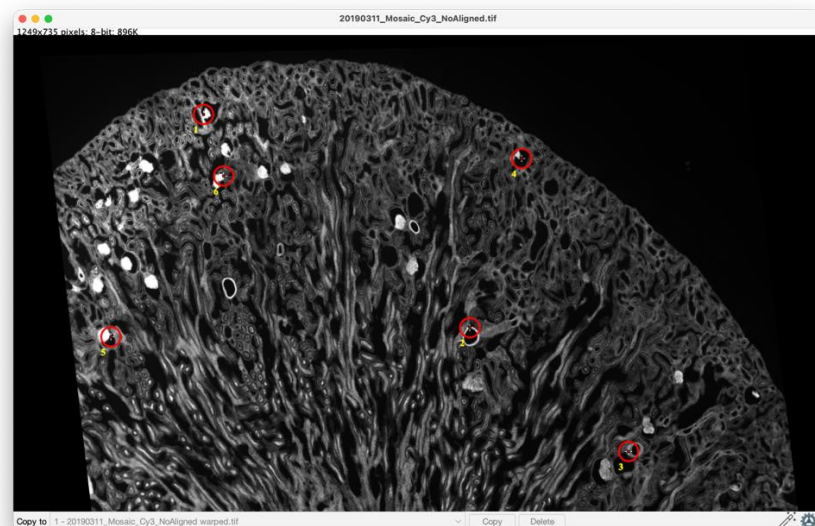


Figure 3-32 Landmark su immagine warped

Come mostrato in Figura 3-2, tramite l'interfaccia grafica del nostro plugin è possibile selezionare un insieme di landmarks sull'immagine di riferimento e sull'immagine distorta. Dopo aver selezionato questi landmarks, è possibile avviare l'algoritmo di deformazione elastica tramite il bottone: “*Manual Elastic Registration*” e successivamente se i vari landmark sono stati definiti in maniera corretta si può ottenere una correzione della deformazione elastica più o meno forte. Questo approccio permette quindi una registrazione altamente personalizzabile nella quale l'operatore mantiene un controllo completo sulle regioni chiave dell'immagine.

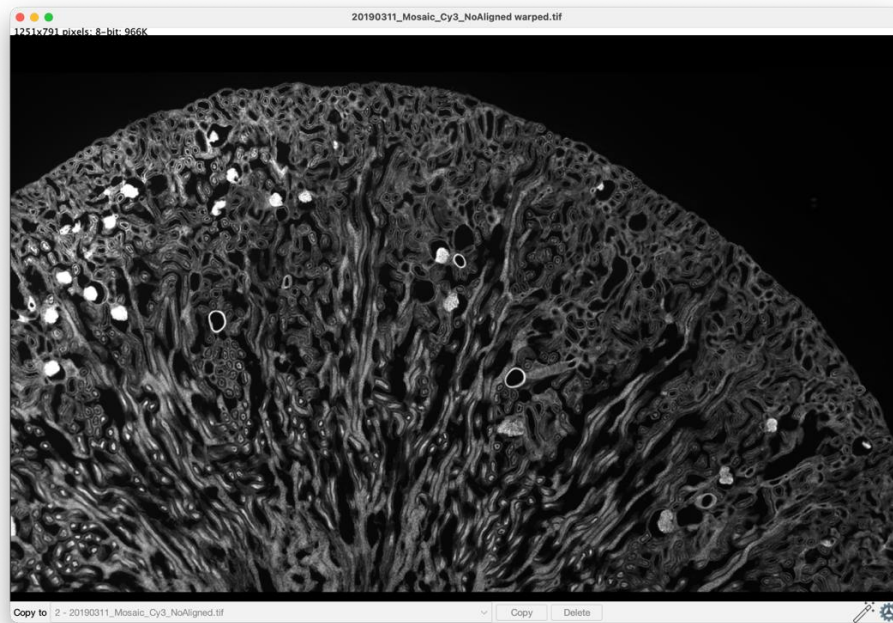


Figure 3-33 Output dopo deformazione elastica

3.3.3 Algoritmo elastico automatico

La seconda modalità di applicazione dell'algoritmo di deformazione elastica è quella in modalità *automatica*. Questo tipo di modalità rappresenta un approccio che non necessita della definizione di landmarks da parte dell'utente, questo fa sì che la deformazione possa essere altamente flessibile. A differenza dell'allineamento manuale, in cui l'utente definisce esplicitamente le corrispondenze tra punti omologhi, l'allineamento automatico si basa su algoritmi in grado di individuare corrispondenze in modo completamente autonomo a partire dai criteri interni all'immagine. In questo paradigma, i landmark vengono individuati in maniera automatica tramite l'interfaccia *PointDetector*, successivamente dopo aver memorizzato all'interno di ogni immagine una serie di landmark, l'immagine deformata viene automaticamente sottoposta all'algoritmo di deformazione elastica prendendo come immagine di riferimento l'immagine target (Figure 3-6).

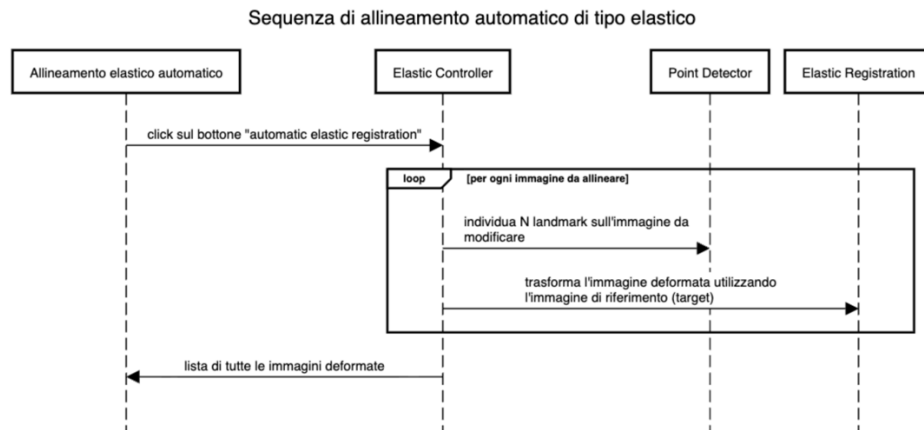


Figure 3-34 Diagramma di sequenza per la procedura automatica

Questo approccio va impiegato facendo attenzione dal momento in cui non è sempre possibile ottenere un risultato affidabile tramite la procedura automatica. L'allineamento automatico elastico offre un vantaggio sostanziale nei contesti in cui si devono registrare grandi quantità di dati o quando un intervento manuale sistematico non è praticabile. Questo però può introdurre una fase successiva di validazione manuale.

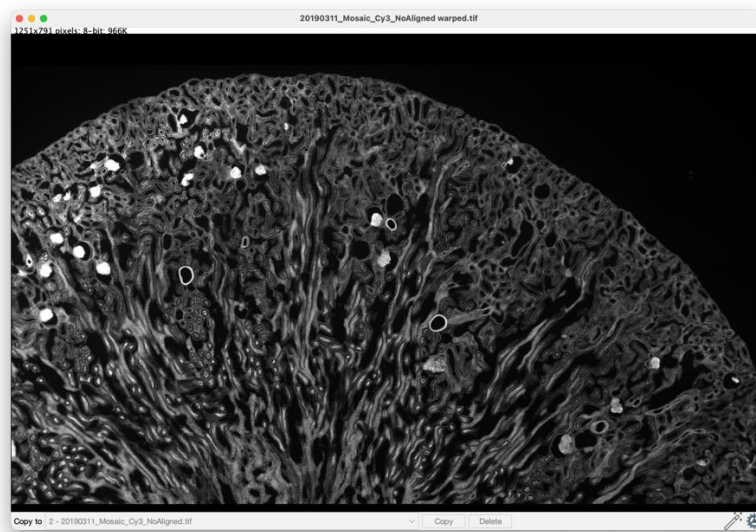


Figure 3-35 Output allineamento automatico elastico

4 Risultati sperimentali

All'interno di questo capitolo viene presentata la parte finale del mio lavoro, in particolare dopo aver introdotto la possibilità di effettuare registrazioni di tipo elastico in modalità automatica e manuale, è stato necessario verificare la validità degli output prodotti dalla deformazione elastica. A tal fine è stato sviluppato un software in linguaggio *Python3* per verificare la validità degli output prodotti dal plugin. Il codice è scaricabile dal seguente link: <https://github.com/MatteoIorio11/similarity-checker>.

4.1 Tool interno per misurazione di similarità

Il software sviluppato ha l'obiettivo di valutare in modo sistematico il grado di similarità tra due immagini, tipicamente la versione originale e quella ottenuta dopo il processo di allineamento elastico. L'applicativo acquisisce in ingresso le immagini selezionate dall'utente e le elabora affinché abbiano la stessa dimensione, condizione necessaria per garantire la correttezza delle analisi successive. Una volta processate, le immagini vengono sottoposte a diverse metriche di confronto, ciascuna implementata attraverso una specifica classe dedicata. Il programma utilizza tali strategie per calcolare il livello di similarità secondo diversi approcci, restituendo per ognuno un valore numerico che quantifica quanto le due immagini risultino coerenti tra loro dopo la deformazione applicata. In questo modo il software fornisce un sistema flessibile e modulare per misurare la qualità dell'allineamento, consentendone di verificare l'affidabilità delle tecniche di registrazione implementate.

4.1.1 Deformazioni applicate

Nel contesto delle registrazioni di immagini, le trasformazioni geometriche rivestono un ruolo centrale, poiché permettono di mettere in corrispondenza due acquisizioni che rappresentano la stessa scena ma presentano differenze dovute a cambiamenti di posizione, prospettiva o deformazioni locali. Tra le trasformazioni più comuni rientrano quelle *affini*, *proiettive* e *traslative* ciascuna caratterizzata da un differente livello di complessità e flessibilità, fino ad arrivare alle deformazioni elastiche, impiegata quando è necessario modellare variazioni non uniformi e localizzate. Le trasformazioni sono state applicate a delle specifiche *ROI* di immagini più complesse (Figure 4-1, Figure 4-2).

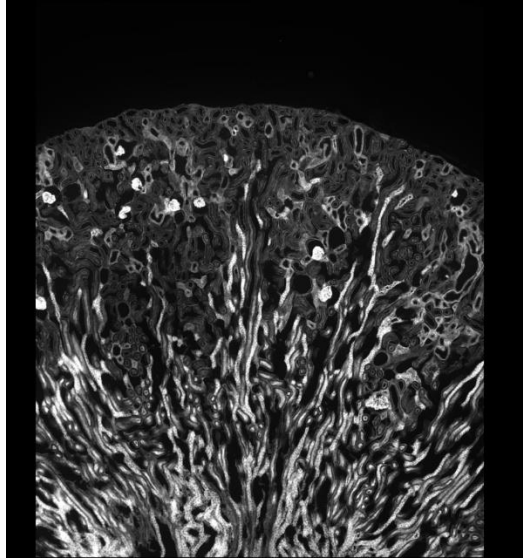


Figure 4-1 Immagine del dataset

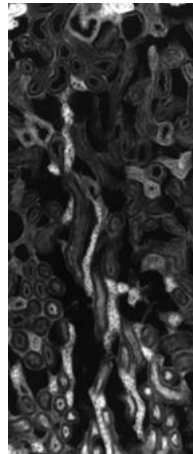


Figure 4-2 ROI del campione

La trasformazione **affine** consente di descrivere modifiche globali dell'immagine che preservano parallelismo e linearità, come rotazioni traslazioni, scalatura e inclinazioni. Pur offrendo un buon equilibrio tra semplicità e capacità descrittiva, questa trasformazione resta vincolata a variazioni omogenee sull'intera immagine.

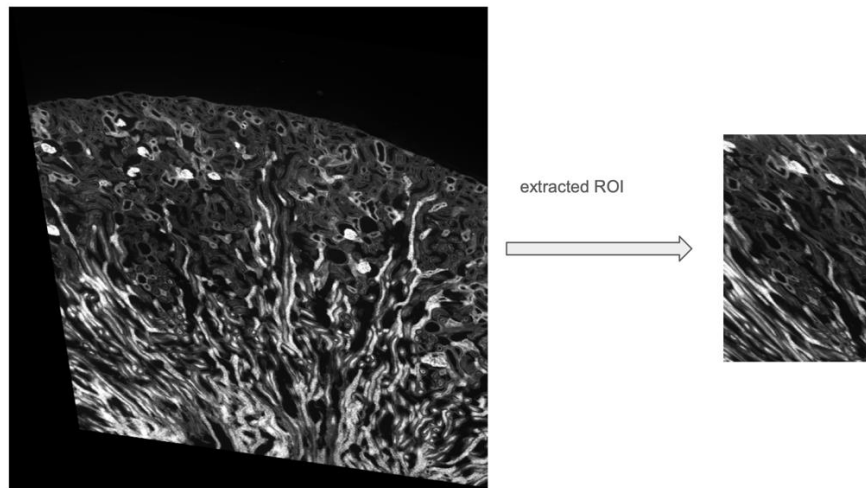


Figure 4-3 Trasformazione Affine e ROI affine

La trasformazione **proiettiva** estende ulteriormente il modello affine, permettendo di rappresentare cambiamenti dovuti alla prospettiva. Essa è in grado di adattare l'immagine in modo tale da simulare variazioni del punto di vista, deformando lo spazio in modo non uniforme mantenendo una coerenza geometrica complessiva. Risulta quindi utile quando l'acquisizione subisce effetti di inclinazione o profondità.

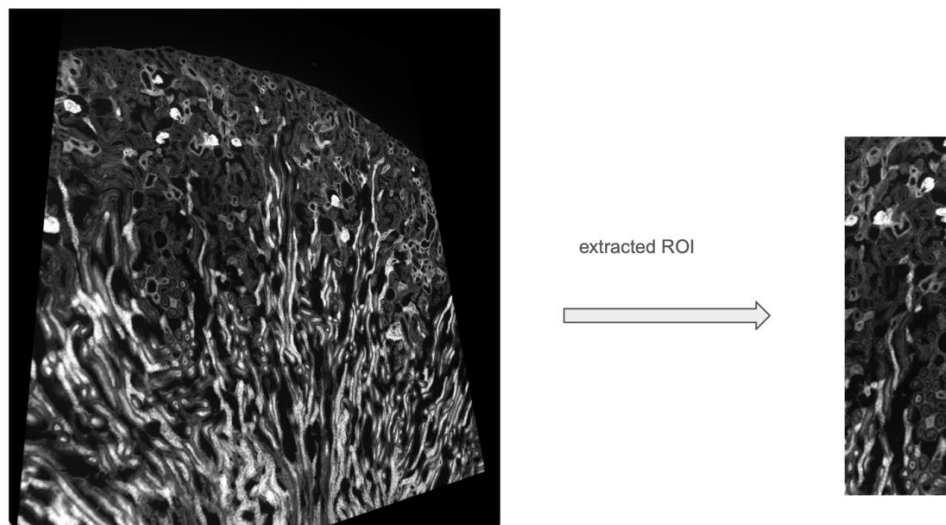


Figure 4-4 Trasformazione Proiettiva e ROI proiettiva

La trasformazione **traslativa** rappresenta il caso più semplice ed elementare, in cui un'immagine viene spostata rigidamente nello spazio senza subire alcuna modifica in forma o scala. Essa permette di correggere differenze di posizione lineare tra due immagini.

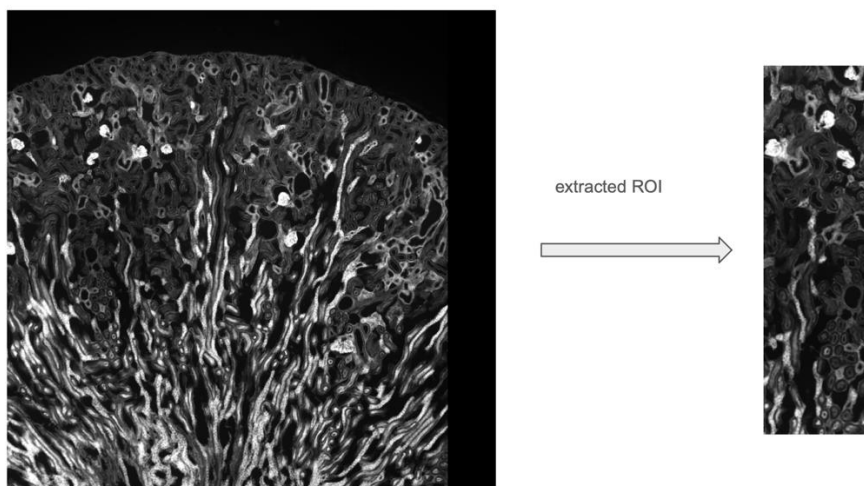


Figure 4-5 Trasformazione traslativa e ROI traslativa

Infine, la deformazione elastica costituisce un approccio molto più flessibile e potente, poiché consente di modellare variazioni locali e non uniformi. A differenza dei modelli rigidi o globali, questa trasformazione permette a ogni zona dell'immagine di muoversi in modo indipendente, pur mantenendo una continuità complessiva. Viene utilizzata quando l'allineamento richiede di compensare distorsioni complesse dovute a fenomeni fisici. In questo caso, ho usato questa trasformazione per applicare delle deformazioni pesanti sull'immagine.



Figure 4-6 Trasformazione elastica pesante

4.1.2 Metriche di misurazione

Durante lo sviluppo di questo tool interno si è scelto di integrare diverse metriche di valutazione al fine di misurare con maggiore accuratezza la qualità dei risultati ottenuti dopo il processo di

allineamento. L'impiego di più indicatori consente infatti di analizzare la similarità tra immagini da prospettive complementari, riducendo il rischio che una singola misura possa risultare fuorviante in presenza di specifiche caratteristiche o artefatti. Le metriche adottate nel software sono **Gabor**, **GLCM**, **UQI** e **SSIM**, ciascuna caratterizzata da un proprio modello di analisi.

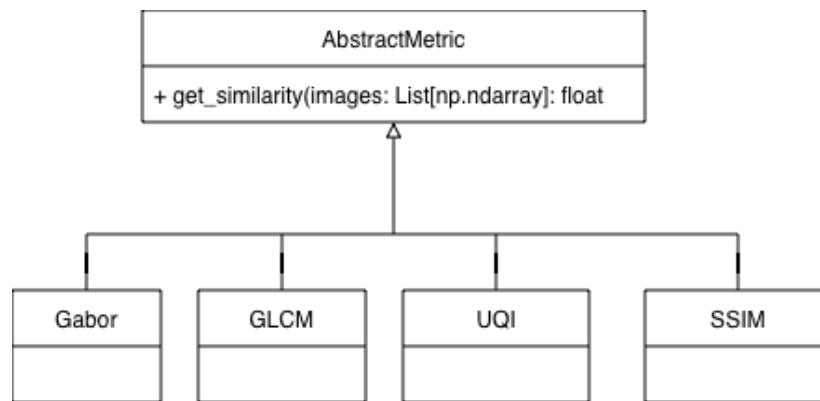


Figure 4-7 UML Delle metriche implementate

La metrica di **GLCM** (Gray Level Co-occurrence Matrix) analizza la distribuzione delle co-occorrenze dei livelli di grigio, permettendo di rappresentare in modo quantitativo la texture dell'immagine, questa metrica valuta la somiglianza o differenze legate alla disposizione e alla ripetizione dei pattern locali



Figure 4-8 Esempio di matrici di co occorrenza

La metrica di **Gabor** si basa sull'utilizzo di filtri di *Gabor*, che permettono di estrarre informazioni locali di frequenze e orientazione. Essa risulta particolarmente efficace nell'evidenziare pattern, texture e strutture direzionali, offrendo una misura di similarità sensibile alle variazioni strutturali presenti nelle immagini [23].

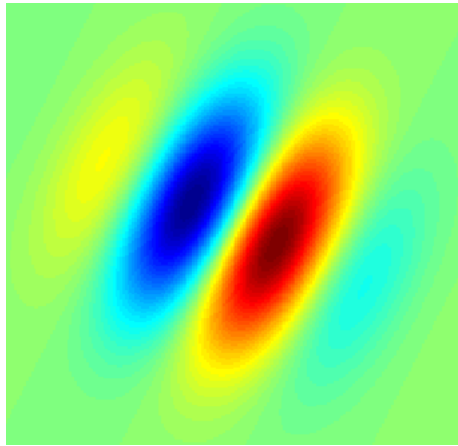


Figure 4-9 Filtri di Gabor

Le metriche **UQI** (Universal Quality Index) fornisce una misura sintetica della qualità dell'immagine comparando tre componenti fondamentali: correlazione, contrasto e luminanza. L'obiettivo è quantificare quanto la struttura complessiva dell'immagine modificata si discosti da quella originale, offrendo un indicatore compatto ma informativo. Ed infine la metrica **SSIM** (Structural Similarity Index Measure) si focalizza sulla percezione della qualità dell'immagine, valutando la similarità in termini di luminanza, contrasto e struttura locale. A differenza di misure puramente numeriche, *SSIM* è progettata per essere coerente con la percezione visiva umana e risulta particolarmente affidabile nel confronto tra immagini oggetto di trasformazione complesse.

4.1.3 Dataset

Il dataset utilizzato per le attività di sperimentazione è stato costruito selezionando quattro immagini eterogenee, scelte con l'obiettivo di valutare il comportamento dell'algoritmo di registrazione elastica introdotto all'interno del nostro plugin. Il primo dato impiegato è un'immagine satellitare, caratterizzata da strutture geografiche complesse e da variazioni di texture su larga scala, ideale per testare la capacità delle trasformazioni di gestire pattern estesi. La seconda immagine appartiene all'ambito astronomico e presenta tipicamente intensità luminose e dettagli debolmente contrastati, permettendo di analizzare la robustezza delle tecniche in presenza di segnali poco definiti.

Completano il dataset due immagini di cellule, selezionate per rappresentare un contesto microscopico con strutture minute e irregolari. Queste immagini sono particolarmente utili per valutare la precisione delle trasformazioni locali, in quanto contengono dettagli sensibili alle deformazioni e richiedono un allineamento accurato per poter essere confrontati correttamente.

La scelta di un dataset così diversificato ha permesso di verificare le prestazioni degli strumenti sviluppati in scenari con caratteristiche visive, scale e livelli di complessità differenti, garantendo una valutazione più completa e affidabile dell'intero processo di registrazione (Figure 4-10).

Le immagini utilizzate sono disponibili al seguente link:

<https://github.com/MatteoIorio11/evaluator-thesis/tree/main/dataset>.

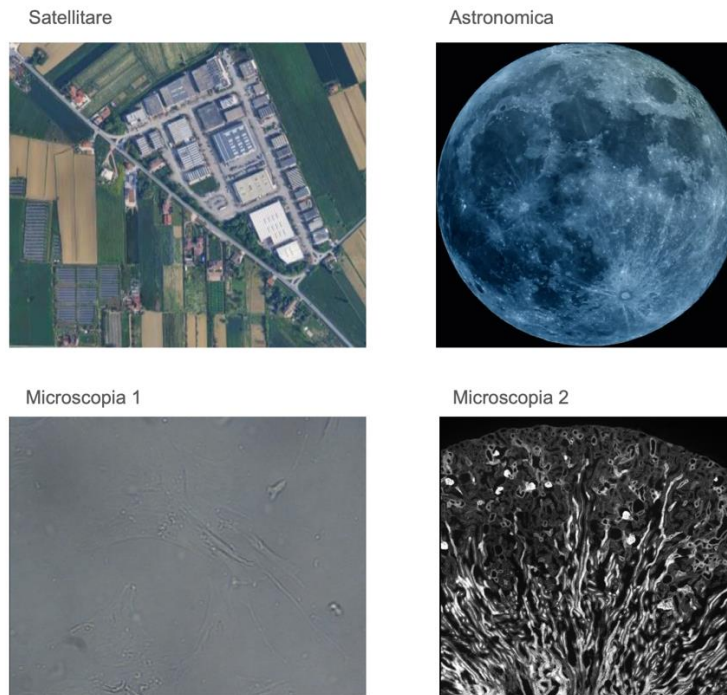


Figure 4-10 Dataset

4.2 Risultati e osservazioni

In questo capitolo vengono presentati e analizzati i risultati ottenuti dalle osservazioni effettuate sulle immagini del dataset dopo l'applicazione delle deformazioni elastiche, eseguite sia in modalità automatica e sia in modalità manuale. Le misurazioni raccolte sono state organizzate in due tabelle distinte, ciascuna contenente i valori delle metriche di similarità calcolate sulle coppie di immagini allineate. Tale approccio ha permesso di valutare in modo oggettivo l'efficacia delle due differenti modalità di registrazione.

Le deformazioni elastiche **manuali** sono state eseguite intervenendo direttamente sui punti di controllo e regolando localmente l'intensità delle trasformazioni, con l'obiettivo di ottenere un allineamento quanto più preciso possibile. Al contrario le deformazioni **automatiche** si basano su un algoritmo che calcola in autonomia i landmark necessari per effettuare la deformazione elastica,

producendo quindi risultati maggiormente dipendenti dalle caratteristiche dell'immagine e dalla robustezza dell'algoritmo stesso.

Image	Score Before: SSIM	Score Before: Gabor	Score Before: GLCM	Score After: SSIM	Score After: Gabor	Score After: GLCM	Error SSIM	Error Gabor	Error GLCM
Cell - Affine	0,05750 276664	0,6329 243818	211,98 97918	0,2024 244719	0,8207 83753	5,19974 1887	0,144921 7053	0,1878593 712	206,790 05
Cell - Projective	0,10998 6715	0,8341 725035	28,633 89797	0,3860 535621	0,9420 711591	12,8917 0036	0,276066 8471	0,1078986 556	15,7421 9761
Cell - Translation	0,56368 79638	0,9946 595711	84,543 38858	0,6182 218784	0,9971 331711	65,5433 8858	0,054533 91457	0,0024736 00007	19
Cell - Elastic	0,10544 70777	0,9073 691846	81,036 35617	0,4107 903119	0,9938 750807	43,1645 7177	0,305343 2342	0,0865058 9613	37,8717 8439
Satellite - Affine	0,46851 15346	0,9560 552288	5,3540 10823	0,6701 005145	0,9983 11362	3,73993 2198	0,201588 9799	0,0422561 3318	1,61407 8625
Satellite - Projective	0,38572 7636	0,9329 885807	4,8273 26623	0,4510 125466	0,9778 610101	3,80262 8731	0,065284 91063	0,0448724 2945	1,02469 7893
Satellite - Translation	0,46851 15346	0,9760 552288	1,5749 90751	0,7760 430332	0,9991 471313	0,87775 86743	0,307531 4986	0,0230919 0251	0,69723 20766
Satellite - Elastic	0,32496 05088	0,8983 620537	6,4233 49809	0,6182 74139	0,9946 204543	5,42215 0798	0,293313 6301	0,0962584 0059	1,00119 9011

Mo on - Af- fin e	0,50981 4035	0,9318 274139	2,0478 12872	0,6724 668632	0,9963 798793	1,04795 7667	0,162652 8282	0,0645524 6537	0,99985 52046
Mo on - Pro ject ive	0,51770 40333	0,9245 803699	6,5169 29883	0,6159 207184	0,9608 664535	2,16745 5107	0,098216 68511	0,0362860 8353	4,34947 4776
Mo on - Tra nsl ati on	0,58435 10083	0,9970 668956	0,5169 298826	0,6847 505068	0,9973 629967	0,03209 810348	0,100399 4985	0,0002961 010853	0,48483 17791
Mo on - Ela stic	0,37379 69958	0,7889 498132	33,941 26345	0,4333 566837	0,8435 660234	15,8098 4974	0,059559 68792	0,0546162 1022	18,1314 1371
Cel l 2 - Af- fin e	0,84509 80251	0,9979 716321	1,8232 83872	0,8478 567071	0,9984 078161	2,87924 5724	0,002758 681968	0,0004361 839549	1,05596 1852
Cel l 2 - Pro ject ive	0,64592 62445	0,9060 136019	21,539 06356	0,9136 698645	0,9999 91721	0,79664 14312	0,267743 62	0,0939781 1905	20,7424 2212
Cel l 2 - Tra nsl ati on	0,93045 65043	0,9982 907276	0,6287 695155	0,8704 195446	0,9979 242344	0,67210 95741	0,060036 95972	0,0003664 932213	0,04334 005853
Cel l 2 - Ela stic	0,75885 30008	0,9971 716292	4,5561 9095	0,8301 733228	0,9972 622463	1,15550 7555	0,071320 32199	0,0000906 1706189	3,40068 3395

Nel caso della deformazione elastica in modalità **automatica**, i landmark necessari alla registrazione elastica sono stati rilevati tramite un algoritmo di ricerca di *keypoints* come ad esempio l'algoritmo

SIFT. Successivamente questi *keypoints* sono utilizzati per deformare le immagini, i risultati ottenuti sono i seguenti:

Im ag e	Score Before: SSIM	Score Before: Gabor	Score Before: GLCM	Score After: SSIM	Score After: Gabor	Score After: GLCM	Error SSIM	Error Ga- bor	Erro GLCM
Cel l - Af- fine	0,05750 276664	0,63292 43818	211,989 7918	0,38828 43464	0,98924 77683	50,9968 0263	0,330781 5798	0,356323 3865	160,992 9892
Cel l - Pro jec- tive	0,10998 6715	0,83417 25035	28,6338 9797	0,04331 787696	0,28223 57327	293,108 1375	0,066668 83808	0,551936 7707	264,474 2395
Cel l - Tra nsl atio n	0,56368 79638	0,99465 95711	84,5433 8858	0,09259 1282	0,81241 94625	81,9145 6849	0,471096 6818	0,182240 1086	2,62882 009
Cel l - Ela stic	0,10544 70777	0,90736 91846	81,0363 5617	0,72995 10952	0,99746 98594	63,3750 4766	0,624504 0175	0,090100 67485	17,6613 0851
Sa- tel- lite - Af- fine	0,46851 15346	0,95605 52288	5,35401 0823	0,52281 11969	0,99656 47765	4,57073 657	0,054299 6623	0,040509 5477	0,78327 42533
Sa- tel- lite - Pro jec- tive	0,38572 7636	0,93298 85807	4,82732 6623	0,40672 02536	0,92615 88595	3,96410 1244	0,020992 61758	0,006829 721176	0,86322 53795
Sa- tel- lite - Tra nsl atio n	0,46851 15346	0,97605 52288	1,57499 0751	0,60983 60615	0,99686 87827	3,28339 7847	0,141324 5268	0,020813 55394	1,70840 7096
Sa- tel- lite - Ela stic	0,32496 05088	0,89836 20537	6,42334 9809	0,38271 96594	0,18664 90659	59,2598 5321	0,057759 1506	0,711712 9878	52,8365 034

Mo on - Af- fine	0,50981 4035	0,93182 74139	2,04781 2872	0,65738 71727	0,99807 14959	1,73606 4065	0,147573 1377	0,066244 08197	0,31174 88069
Mo on - Pro jec- tive	0,51770 40333	0,92458 03699	6,51692 9883	0,51420 70628	0,91139 43689	3,88351 4833	0,003496 970562	0,013186 00104	2,63341 505
Mo on - Tra nsl atio n	0,58435 10083	0,99706 68956	0,51692 98826	0,81102 41569	0,99913 053	0,72288 85314	0,226673 1486	0,002063 634415	0,20595 86488
Mo on - Ela stic	0,37379 69958	0,78894 98132	33,9412 6345	0,23333 86775	0,56585 74178	42,2981 3025	0,140458 3183	0,223092 3954	8,35686 6799
Cel l 2 - Af- fine	0,84509 80251	0,99797 16321	1,82328 3872	0,81238 49988	0,99814 07633	9,81238 4999	0,032713 02627	0,000169 131157	7,98910 1126
Cel l 2 - Pro jec- tive	0,64592 62445	0,90601 36019	21,5390 6356	0,78357 18685	0,99699 98375	1,55392 1659	0,137645 6241	0,090986 2356	19,9851 419
Cel l 2 - Tra nsl atio n	0,93045 65043	0,99829 07276	0,62876 95155	0,81891 37264	0,99742 12806	1,03791 8112	0,111542 778	0,000869 4469675	0,40914 85969
Cel l 2 - Ela stic	0,75885 30008	0,99717 16292	4,55619 095	0,81857 28634	0,99733 11432	0,85811 14515	0,059719 86266	0,000159 5140158	3,69807 9499

Dall'analisi comparativa dei risultati emerge che la modalità **manuale** tende a produrre un allineamento complessivo più accurato. Su un totale di dodici campionamenti, sette presentano valori migliori rispetto alla corrispondente deformazione automatica, mentre solo cinque casi mostrano *performance* superiori nella modalità automatica. Questo risultato evidenzia come l'intervento umano, guidato da un'osservazione consapevole evidenzia come l'intervento umano, quando guidato

da un'osservazione consapevole delle strutture da allineare, sia ancora in grado di garantire una maggiore precisione rispetto all'automatizzazione completa del processo.

La modalità *automatica*, pur mostrando in alcuni casi prestazioni competitive o addirittura migliori, risulta più sensibile alla variabilità delle immagini e alle loro caratteristiche locali. La presenza di pattern complessi, texture irregolari o rumore può influenzare significativamente il comportamento dell'algoritmo compromettendo in parte la qualità dell'allineamento.

In conclusione, i risultati ottenuti suggeriscono che, sebbene le deformazioni automatiche forniscano una soluzione rapida e potenzialmente scalabile, le deformazioni manuali mantengono un vantaggio in termini di accuratezza in gran parte delle osservazioni. Tale evidenza conferma l'importanza del controllo diretto nei processi di registrazione elastica, specialmente quando si lavora con immagini eterogenee o caratterizzate da dettagli complessi.

5 Conclusioni e sviluppi futuri

Questa Tesi ha portato allo sviluppo della versione 1.1 del tool *DS4H Image Alignment*, disponibile gratuitamente come software open-source al link: <https://github.com/UniBoDS4H/IA>. In prima persona mi sono occupato dello studio dei tool competitor, ho realizzato le strutture per gli algoritmi di deformazione elastica in modalità automatica e manuale ed infine mi sono occupato del problema riguardante l'impossibilità di utilizzare la libreria *OpenCV* sulle piattaforme di tipo *ARM*.

DS4H Image Alignment è stato creato per soddisfare alcune richieste provenienti da medici e biologi che dovevano registrare immagini multimodali acquisite tramite microscopi a campo largo, al fine di condurre studi sulla colocalizzazione dei segnali intracellulari e migliorare l'analisi di provini istologici.

Il software sviluppato consente:

1. All'utente di eseguire vari tipi di allineamento, partendo da una forma manuale ad una completamente automatica:
 - L'allineamento manuale è realizzato mediante l'inserimento di un insieme di punti di riferimento (corner points) in ciascuna immagine. Questo consente l'utilizzo di specifiche trasformazioni geometriche per allineare le immagini caricate.
 - La funzionalità di allineamento automatico viene realizzata tramite la selezione di uno dei vari algoritmi implementati.
2. È inoltre possibile affrontare anche il problema della deformazione elastica in modalità automatica e manuale:
 - L'allineamento manuale è realizzato mediante l'inserimento di un insieme di landmark in ciascuna immagine, questo consente di perfezionare il risultato ottenuto dalla deformazione elastica.
 - La funzionalità di allineamento automatico viene realizzata mediante un algoritmo per individuare automaticamente i *keypoints* all'interno di un'immagine, i quali poi vengono impiegati per effettuare la deformazione di tipo elastico.
3. Il plugin sviluppato gestisce anche immagini multicanale;
4. Tutti gli allineamenti effettuati tramite questo software preservano le informazioni contenute in ogni immagine e possono essere utilizzati con tutti i tipi di immagini senza incorrere in errori.

Un'evoluzione significativa del plugin prevede un'estesa attività di *refactoring* del codice, finalizzata a migliorarne la qualità complessiva, la manutenibilità e le prestazioni. In primo luogo, sarà necessario ottimizzare gli algoritmi attualmente sviluppati, intervenendo sia sulla loro complessità computazionale sia sulla gestione delle risorse. Tale processo consentirà di ridurre il carico operativo dell'applicazione, migliorando la reattività e garantendo un comportamento più efficiente anche in scenari ad elevata intensità di utilizzo.

Un secondo intervento riguarda l'eliminazione del meccanismo di polling, che rappresenta una soluzione poco efficiente e potenzialmente dispendiosa in termini di consumi. La sua sostituzione con un modello basato su notifiche o eventi permetterà di migliorare il flusso di comunicazione interno, riducendo le operazioni superflue e promuovendo un approccio maggiormente reattivo. Infine, un ulteriore obiettivo consiste nell'adozione di pattern architetturali adeguati al contesto applicativo, così da strutturare il codice secondo principi solidi e riconosciuti. L'impiego di pattern appropriati consentirà non solo di rafforzare la modularità e l'estendibilità del plugin, ma anche di facilitare l'integrazione di nuove funzionalità future.

In conclusione, *DS4H Image Alignment* attualmente rappresenta la soluzione gratuita più completa per biologi e medici che necessitano di registrare immagini 2D ottenute tramite acquisizioni al microscopio. In particolare, il tool è già stato presentato alla comunità scientifica in due conferenze:

- F. Piccinini, M. Iorio, F. Vincenzi, M.E. Duma, M. Tazzari, M.M. Tumedei, J.C. Pyun, G. Martinelli, G. Castellani, A. Carbonaro. User-friendly open-source tools for aligning multimodality 2D microscopy images and performing single-cell co-localization analysis. Cells & Extracellular Templates (CET) University Niccolò Cusano-Roma, Rome, Italy, June 7-9, 2023.
- F. Piccinini, F. Vincenzi, M. Iorio, M. Tazzari, M.M. Tumedei, J.C. Pyun, E. Giampieri, G. Martinelli, G. Castellani, A. Carbonaro. DS4H Image Alignment (DS4H-IA), an open-source ImageJ/Fiji plugin for aligning multimodality 2D microscopy images. Straub Conference 2023, Biological Research Centre (BRC), Szeged, Hungary, May 25-26, 2023.

Attualmente è gratuitamente disponibile online ed è quotidianamente utilizzato presso l'*Istituto Romagnolo per lo Studio dei Tumori "Dino Amadori"* IRCCS IRST di Meldola (FC, Italy) ed all'interno della comunità del *Centro Nazionale di Ricerca* CNR (Italy).

Bibliografia

- [1]. <https://qima-lifesciences.com/en/sample-and-slide-preparation/>
- [2]. <https://www.ncbi.nlm.nih.gov/books/NBK557663/>
- [3]. <https://blogs.mathworks.com/steve/2006/04/28/spatial-transforms-forward-mapping/>
- [4]. <https://www.sciencedirect.com/science/article/abs/pii/S0167865519301175>
- [5]. <https://www.mdpi.com/2075-1680/7/3/46>
- [6]. https://docs.opencv.org/3.4/d4/dee/tutorial_optical_flow.html
- [7]. <https://imagej.net/plugins/bigwarp>
- [8]. <https://pubmed.ncbi.nlm.nih.gov/32492178/>
- [9]. <https://elastix.dev>
- [10]. <https://itk.org>
- [11]. <https://imagej.net/plugins/elastic-alignment-and-montage>
- [12]. <https://imagej.net/plugins/moving-least-squares>
- [13]. <https://imagej.net/plugins/image-stitching>
- [14]. <https://imagej.net/plugins/bigstitcher/>
- [15]. <https://bigwww.epfl.ch/thevenaz/UnwarpJ/>
- [16]. <https://imagej.net/plugins/bunwarpj/>
- [17]. <https://www.ibm.com/think/topics/computer-vision>
- [18]. https://docs.opencv.org/4.x/db/d05/tutorial_config_reference.html
- [19]. <https://stackoverflow.com/questions/62745195/opencv-with-python-and-java-support-using-cmake-command-line>
- [20]. <https://stackoverflow.com/questions/30657774/surf-and-sift-algorithms-doesnt-work-in-opencv-3-0-java/35266046#35266046>
- [21]. https://en.wikipedia.org/wiki/Thin_plate_spline
- [22]. https://adamdjellouli.com/articles/numerical_methods/6_regression/thin_plate_spline_interpolation
- [23]. https://it.wikipedia.org/wiki/Filtro_di_Gabor

Ringraziamenti

Vorrei subito ringraziare la Prof.ssa Antonella Carbonaro dell'Università di Bologna per avermi seguito nel ruolo di relatrice durante la preparazione e lo svolgimento di questa Tesi, con prontezza, professionalità e disponibilità, dandomi l'opportunità di lavorare a questo progetto multidisciplinare.

Ringrazio il Dr. Nicola Normanno (permesso di utilizzo dati pazienti oncologici) per aver consentito l'utilizzo di dati acquisiti presso l'Ospedale IRST IRCCS Meldola (FC), ed il Prof. Gastone Castellani (accesso alle infrastrutture informatiche) per aver accesso alle infrastrutture con cui testare l'algoritmo sviluppato.

Desidero esprimere la mia gratitudine al Prof. Filippo Piccinini per avermi introdotto nel mondo della microscopia e dell'oncologia. Invece di essere semplicemente un correlatore, il Professor Piccinini si è distinto per la sua gentilezza e disponibilità, permettendomi di vivere un'esperienza eccezionale. È stato sempre pronto a discutere le mie soluzioni, offrendomi suggerimenti per migliorare il software. Ha sostenuto e incoraggiato le mie idee, facendomi sentire sempre competente in questo ruolo e credendo nelle mie potenzialità. Sono grato per tutto ciò che ha fatto per me.

Inoltre, desidero esprimere un profondo e sentito ringraziamento ai miei genitori, che mi sono sempre stati accanto nonostante tutte le difficoltà e i momenti di incertezza. La loro presenza costante, il loro sostegno incondizionato e la loro capacità di credere in me anche quando io stesso faticavo a farlo sono stati fondamentali lungo tutto il mio percorso.

È grazie ai loro sacrifici e alla loro fiducia se sono riuscito a portare a termine ogni mio obiettivo. Non mi sono mai sentito solo: il loro incoraggiamento mi ha accompagnato in ogni scelta, aiutandomi a sentirmi all'altezza dei traguardi che desideravo raggiungere.

A loro devo tutti i miei risultati raggiunti.

Desidero anche rivolgere un ringraziamento speciale ai miei amici, che hanno rappresentato una presenza preziosa in ogni fase di questo percorso. La loro vicinanza, il loro sostegno spontaneo e la loro capacità di strapparmi un sorriso anche nei momenti più impegnativi hanno fatto davvero la differenza.

Con loro ho condiviso non solo le difficoltà, ma anche le soddisfazioni, i piccoli progressi quotidiani e le pause necessarie per ritrovare energia e motivazione.

Infine, desidero ringraziare la mia lei, che mi ha accompagnato in ogni difficoltà, sostenendomi con pazienza e facendomi sentire sempre all'altezza. La sua presenza, il suo affetto e la sua capacità di valorizzarmi anche nei momenti più complessi mi hanno dato la forza di proseguire con fiducia.