Tesi

# bozza della tesi

## plugin in imagej

I componenti principali di ImageJ, che gli fanno anche da base, sono i Plugin che estendono e arricchiscono il Tool in modo semplice e efficace. Ogni elemento, anche l’interfaccia con cui accoglie l’utente, è creata attraverso i Plugin. Un Plugin si può sviluppare in Macro e Java ma in questo elaborato ci concentreremo sulle tipologie scritte con il secondo linguaggio di programmazione nominato. Qualsiasi utente può iniziare la stesura di un Plugin personalizzato. Attraverso il menu principale, scegliendo la voce “Plugins”, può iniziare la scrittura attraverso il comando new e scegliendo ilPlugin base prescelto. Il particolare ImageJ ne offre quattro:

1. Plugin
2. Plugin Filter
3. Plugin Frame
4. Plugin Tool

Per concentrarsi sulla individuazione e calcolo delle misure richieste per l’elaborato i plugin utilizzati sono della tipologia Plugin e Plugin Filter. In termini tecnici esse sono interfacce. PlugIn riporta un solo metodo chiamato run che prende come argomento un oggetto di tipologia String. Il metodo viene chiamato quando viene caricato il Plugin che si serve di questa interfaccia. L’argomento di classe String può essere anche vuoto, ovvero “”. Plugin Filter è leggermente più corposo. Al suo interno vi è il metodo setup. Quest’ultimo viene chiamato quando il Plugin viene caricato. Presenta due argomenti passati uno ti tipologia String che può essere vuoto e un oggetto di tipo ImagePlus che indica l’immagine attualmente attiva. Restituisce un valore di tipo int che indica se il metodo è andato a buon fine. Presenta anche un metodo run (medesimo nome della interfaccia citata sopra) ma con un obbiettivo diverso; utilizzano il metodo per elaborare l’immagine e presenta una serie di flag di controllo, pubbliche. Prende in ingresso un oggetto della classe ImageProcessor. Quest’ultima è volta ad applicare alcuni metodi base per manipolare immagini 2D.

## Plugin dell’elaborato

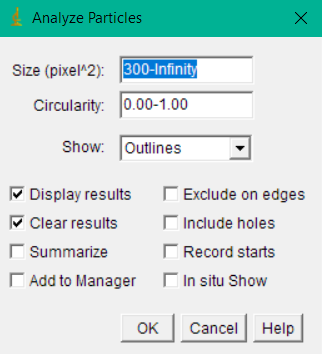
### introduzione

Nonostante ImageJ sia ricco di Plugin per poter trattare e calcolare molte features data in ingresso una immagine, nessun plugin prende tutte le misure necessarie all’obbiettivo proposto, o comunque non le raggruppa nello stesso Plugin. L’idea sarebbe poter sfruttare al meglio sui Plugin base su cui si basa ImageJ e estrarre le misure richieste relative ad immagini biomediche. In particolare, il Plugin che si avvicina di più a queste esigenze è “ParticlesAnalyzer”, avviabile dal menù “Analyze” con la voce “Analyze Particles…”. L’elemento più importante di ParticlesAnalyzer è la possibilità di individuare più oggetti in una immagine e calcolare le features su ognuno di essi. Questo avviene attraverso un Threshould manuale, definizione delle aree di interesse attraverso il plugin “Threshould” accessibile dal menu sotto le voci Image, Adjust e “Threshould”. Se si tratta di immagini in RGB, ovvero a colori, si usa invece “Color Threshould” accessibile da medesimo menù. Particles Analyzer implementa PlugInFilter e Measurements. Measuments è una interfaccia che lo collega ad un menù in cui si possono settare alcune misure ma ai fini dell’elaborato non sarà rilevante. Date le proprietà di questo Plugin la decisione presa per implementare e raggruppare le misure già esistenti è quella di creare un nuovo Plugin che estendesse Particles Analyzer e basarsi su di esso.

### Sviluppo e storia del plugin

Il Plugin prodotto è stato aggiornato volta per volta basandosi su tre stadi; il primo stadio prevedeva la implementazione di misure in B&W, sia già calcolate sia da calcolare; il secondo stadio prevedeva la implementazione delle misure in grey; il terzo stadio e ultimo prevedeva la implementazione delle misure in RGB. Durante la stesura del codice e laricerca di metodologie da applicare per inserire le formule, per la maggior parte note, all’interno del Plugin, si è voluto curare anche l’interfaccia in modo tale da poter scegliere se implementare tutte le misure proposte o solo alcune. Inoltre struttura del Plugin stesso è stato modificato più volte a causa, o grazie, a metodologie scoperte e applicate in corso d’opera che rendevano il codice comprensibile e logico.

### Struttura del Plugin

“Catch\_Parasite3” (??) è il nome attuale del Plugin implementato. Esso implementa l’interfaccia PlugIn e al suo interno vi è solo un metodo chiamato run, imposto dalla interfaccia un vettore di ImagePlus di nome impVector. All’interno del metodo run viene avviato il plugin e se l’immagine è a colori viene trattata per elaborarla con i canali RGB e le nuove tre immagini derivanti dal trattamento vengono inserite all’interno del vettore impVector che verrà utilizzato per estrarle e calcolare alcune misure. Viene avviata una inner class; questa classe interna ha il nome Parasite (??) ed estende ParticlesAnalyzer ricavandone le proprietà necessarie per poter lavorare su più oggetti nell’immagine. Vengono dichiarate una serie di variabili booleane necessarie per selezionare solo le misure scelte dall’utente. Seguono i metodi necessari per aggiungere le misure implementate nel plugin; molti di essi sono Override dei metodi della super classe. Il primo metodo importante è showDialog che non prende argomenti in ingresso e restituisce un valore di tipo intero. Esso è un Override e richiama il metodo della super.classe. Questo visualizzerà un menu iniziale già implementato da Particles Analyzer. Nel seguente menù, l’utente può decidere la grandezza minima dell’oggetto analizzato e a seguire quella massima e la sua circolarità. Attraverso il menu a tendina si può scegliere il trattamento della immagine finale. Le scelte sono: nothing, per cui l’immagine non viene trattata, outlines per cui l’immagine viene trattata e come risultato si ha una seconda immagina bianca dove si vedono solo ed esclusivamente i contorni degli oggetti con una etichetta che sta ad indicare il numero dell’oggetto, bare outlines per cui si vedono solo i contorni degli oggetti, ellipses per cui l’immagine risultante è a sfondo bianco con delle ellipsi orientate al posto degli oggetti individuati, mask che restituisce l’immagine della maschera in bianco e nero, nero per gli oggetti e bianco lo sfondo, count mask (??), overlay che riprendendo l’immagine originale etichetta gli oggetti e infine overlay masks che riprende l’immagine originale e nelle zone dove si trova la maschera individuata sostituisce con un colore celeste chiaro. Le altre opzioni che si possono settare attraverso checkbox sono display results per visualizzare appunti i risultati, clear results per cancellare i risultati precedenti e summarize, Add to Manager, Exclude on Edges, Include holes, records starts, in situ Show.

|  |
| --- |
|  |