



Università
Ca' Foscari
Venezia

Università degli Studi di Venezia
Dipartimento di Scienza Ambientali, Informatiche e
Statistiche

Corso di Laurea in Informatica

Tesi di laurea

Titolo

Sottotitolo

Candidato:
Matteo Scarpa
Matricola 845087

Relatore:
Claudio Silvestri

Anno Accademico 2015–2016

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

Lorem ipsum dolor sit amet, consectetur adipiscing elit.

— Oscar Wilde

Dedicata a tutti quelli che mi sono stati vicini nelle giornate no.

Indice

1	Descrizione approfondita dell'argomento	3
1.1	Problema da risolvere	3
1.2	QGis	3
1.2.1	Descrizione dei tipi di dati supportati	4
1.2.2	La struttura di QGis	4
1.2.3	Sistema dei plugin	5
2	Realizzazione	7
2.1	Divisione in sottoproblemi	7
2.1.1	Interfaccia grafica	7
2.1.2	Gestione input-output	9
2.1.3	Gestione dati	9
2.1.4	Gestione output	9
A	Appendice	11
A.1	Strutturare un plugin	11
	Bibliografia	13

Elenco delle figure

Elenco delle tabelle

Introduzione

In questa tesi viene spiegato come è stato realizzato un plugin per QGis per il calcolo del terreno eroso avendo dei dati geomorfici legati al territorio scelto. Questi dati comprendono la conformazione del terreno, le precipitazioni registrate nell'area, il tipo di copertura della vegetazione e il numero di anni per cui questa previsione viene eseguita.

Questo plugin è stato scritto in Python e implementa il plugin ipotizzato nella tesi [\[Zen13\]](#) che, anche se la tesi è legata all'area geografica del Veneto, risulta essere indipendente e utilizzabile per il calcolo su una qualunque area geografica di cui si hanno i dati richiesti.

Capitolo 1

Descrizione approfondita dell'argomento

1.1 Problema da risolvere

Lo scopo di questa tesi è quello di calcolare in modo corretto l'erosione del territorio nell'area geografica scelta. Questo è stato fatto utilizzando un modello RUSLE basato sull'equazione¹

$$A = R * K * LS * C * P \quad (1.1)$$

in cui

- **A** corrispondente alla perdita di suolo annua
- **R** erosività delle precipitazioni
- **K** erodibilità del suolo
- **LS** rapporto lunghezza pendenza
- **C** fattore di copertura del suolo
- **P** misure di prevenzione dell'erosione

Questo modello è stato quindi applicato utilizzando i dati in formato di layer raster attraverso l'applicativo QGis. Questo viene fatto in quanto lo standard *de facto* per questa tipologia di dato è il layer raster.

In particolare questo modello viene utilizzato per calcolare anno per anno l'erosione del territorio interessato avendo variazioni di dati in base all'anno o assumendo come invarianti negli anni i dati inseriti in input.

1.2 QGis

QGis si definisce come "Un Sistema di Informazione Geografica Libero e Open Source"[[Sita](#)] ovvero è un software che gestisce, elabora e visualizza dati geomorfici

¹Il modello proposto è quello indicato in [[Zen13](#), p. 37]

e georeferenziati, completamente gratuito e modificabile in ogni sua parte (è limitato dalla licenza utilizzata nello sviluppo del software [Sib]).

In particolare sono state utilizzate le componenti di gestione input e output dei raster, il motore di calcolo e il sistema di rendering video. In oltre il programma supporta anche un sistema di plugin e scripting che aumenta le funzionalità del programma utilizzando codice esterno sviluppato dalla community.

Altro dettaglio molto utile di QGIS è il fatto che un programma multi piattaforma. Questo vuol dire che il programma è stato distribuito per più sistemi operativi² e obbliga gli sviluppatori a scrivere plugin e script che siano anch'essi multiplatforma. Questo permette l'utilizzo dei plugin e degli script per la creazione di funzioni automatizzate utilizzando le componenti già presenti in QGIS o aggiungerne di nuove.

In oltre, QGIS supporta la quasi totalità dei formati open source utilizzati per la codifica per dati geomorfici e georeferenziati in modo nativo mentre i formati di file che non supporta sono utilizzabili attraverso plugin gratuiti disponibili online.

1.2.1 Descrizione dei tipi di dati supportati

I formati di dati possono essere divisi in tre macrocategorie in base a come vengono forniti:

- Vettoriali
- Raster
- DBRMS con estensione spaziale

Questi tre formati permettono solo alcune operazioni possibili legate alla composizione dei dati nel formato di input. Quindi, per alcune operazioni, QGIS converte i dati nel formato appropriato in modo che sia sempre possibile fare tutte le operazioni disponibili, anche se queste può portare ad approssimazioni dei dati nella conversione.

1.2.2 La struttura di QGIS

QGIS è un programma avanzato con molte funzionalità che possono essere suddivise in alcune macro-categorie che sono indipendenti dal tipo di dato o file utilizzato. Le funzionalità standard di QGIS sono gestite attraverso dei moduli che costituiscono l'installazione di base del programma e le funzionalità aggiuntive vengono implementate attraverso plugin dipendenti da questi moduli di base per poter funzionare correttamente. Solitamente i moduli sono scritti in C o C++ mentre i plugin e gli script sono in Python.

Rendering grafico dei dati

QGIS ha la capacità di realizzare svariate visualizzazioni grafiche dei dati inseriti o calcolati. Questo permette la realizzazioni di immagini ad alta definizione dei dati inseriti e ne permetta anche l'esportazione nei formati più comuni di immagini.

²In questo caso per Android, Mac OS, Windows e Linux

Realizzazione mappe

In presenza di dati georeferenziati QGIS permette di sovrapporre la rappresentazione grafica dei dati alla mappa corrispondente alle coordinate georeferenziali. Questo permette di creare mappe fisiche contenenti la visualizzazione di aree di erosione attraverso aree colorate o altri effetti applicabili alla mappa sottostante.

Creazione, elaborazione e conversione dati

QGIS presenta due moduli per l'elaborazioni di dati:

- **Class** modulo standard di QGIS per i calcoli aritmetici elementari e alcuni calcoli statistici di base
- **Grass** modulo aggiuntivo di QGIS per i calcoli avanzati e regressioni. Contiene anche modelli previsionali e probabilistici.

Analisi dati

Quando si lavora con un insieme di dati non è detto che questi siano tutti corretti o non abbiano un livello di approssimazione. All'interno di QGIS sono disponibili, di base, un insieme di funzionalità atte a calcolare la bontà delle approssimazioni o dei dati stessi presi in esame dal programma. Questo viene fatto di base con algoritmi generici per la maggior parte dei formati ma se si usa solo un tipo specifico o si vuole una approssimazione o un modello preciso è possibile ottenere il risultato desiderato attraverso plugin di terze parti.

1.2.3 Sistema dei plugin

Per aumentare le funzionalità di QGIS e scriptare alcune operazioni sono stati creati i plugin. Sono dei programmi dipendenti da QGIS che implementano funzionalità non presenti nel programma di base o estendono il supporto delle funzioni esistenti o i tipi di file supportati.

Plugin per formati di file

Vengono utilizzati per elaborare dati grezzi altrimenti difficilmente manipolabili. Un esempio possono essere i dati grezzi ottenuti da degli strumenti di misurazione con dati in output non standard. Normalmente questi file sono leggibili solo col programma abilitato alle modifiche ma, con l'utilizzo del plugin corretto, è possibile leggerlo e elaborarlo direttamente in QGIS.

Plugin che alterano o aumentano le funzionalità

La reale forza di QGIS. Permettono di ampliare le funzioni del programma semplificando o aumentando le funzionalità pre esistenti. Alcuni esempi sono i plugin della famiglia GRASS che aggiungono operazioni matematiche o gli strumenti di creazione timelapse partendo da una serie di raster temporizzati.

Capitolo 2

Realizzazione

Il progetto è stato realizzato attraverso la divisione in subproblemi. Questo permette la sostituzione di una componente con un'altra in modo rapido ed sicuro in quanto viene a essere modificata solo la parte del codice interessata dall'operazione e non tutte le altre componenti, che rimangono invariate.

Questo ha portato anche alla ricerca di alcune librerie per eseguire le operazioni necessarie alle funzioni scelte.

2.1 Divisione in sottoproblemi

Per semplificare lo sviluppo di questa applicazione e il suo sviluppo è stato diviso il progetto in quattro componenti indipendenti che dialogano tra di loro, in modo da rendere più isolate le varie componenti, rendendole meno soggette a errori non previsti. Queste si dividono in:

- Interfaccia grafica
- Lettura dei file in ingresso
- Gestione ed elaborazione dei dati
- Scrittura dei file in uscita

Questa strutturazione permette anche il riciclo del codice utilizzato per risolvere i sottoproblemi in altri progetti con la stessa necessità o la facile estensione del plugin in un secondo momento

2.1.1 Interfaccia grafica

Al momento della progettazione dell'applicazione è stato necessario pensare come realizzare l'inserimento dei dati da elaborare da parte dell'utente. Questo, solitamente, viene fatto in due modi:

- Attraverso l'uso di un **terminale** in cui vengono passati dei comandi che eseguono le operazioni richieste e mostra a schermo tutti i dati dei comandi eseguiti. Questo però obbliga l'utente a sapere i comandi necessari alla esecuzione dei task e ritorna a schermo dati di poco interesse per l'utente medio, molto utile invece se sei un programmatore e vuoi controllare come procedono i vari passi dei comandi eseguiti e permette di vedere subito ove avvengono gli errori e cosa li generano.

- Attraverso l'uso di una **interfaccia grafica o GUI** in cui viene selezionato, attraverso l'uso di bottoni, menu e interruttori le funzioni che si vogliono eseguire e si danno i file in input con un selettore grafico. Permette all'utente di non interessarsi a come è stata realizzata l'operazione richiesta, di non consultare guide e manuali con gli elenchi dei comandi disponibili e di ricevere a schermo solo le informazioni di cui è realmente interessato. Oltre a essere la scelta più pratica per un nuovo utente è anche quella che dà al programmatore che sviluppa questa applicazione la maggior libertà e indipendenza. Infatti la presenza di una grafica comune tra aggiornamenti diversi permette al programmatore di riscrivere interamente i comandi che vengono eseguiti al di sotto dell'interfaccia senza dover avvisare l'utente del cambiamento dei comandi all'interno del terminale.

Per come è strutturato QGIS è possibile avere entrambi implementati nel plugin ma è stato preferito dare maggiore attenzione alla componente grafica anche se è comunque possibile aprire il terminale di QGIS ed eseguire dei comandi.

In particolare in questa applicazione è stata implementata l'interfaccia grafica in quanto risulta molto più adatta all'uso effettivo dell'applicazione.

Infatti come si vede nell'immagine [\[link immagine\]](#) per il progetto in esame è necessario soltanto una gui di inserimento dati in cui, in base al tipo di file inserito e al tipo di output, li elabora correttamente. Questo, su terminale, risulta invece più difficile in quanto bisognerebbe avere una lista dei driver necessari per la corretta lettura e inserirli assieme al file da leggere¹.

Qt4

Le librerie grafiche per Python² sono molte e varie ma la scelta è stata abbastanza facile: QT.

Questa scelta è stata dettata dal desiderio di ridurre al minimo le dipendenze esterne dal plugin. Questo comporta l'utilizzo preferenziale delle librerie utilizzate da QGIS per semplificare l'installazione delle dipendenze. In particolare, con QGIS è stato necessario scegliere anche la versione di QT con cui sviluppare il plugin. La scelta è stata fatta su QT4 ovvero la funzione supportata dalla versione 2.0 in poi di QGIS. Questo permette un enorme supporto di versioni di QGIS da parte del plugin.

La versione QT5 che verrà impiegata per le future versioni di QGIS è già interamente supportata dall'applicazione. Questo è stato possibile attraverso un attento sviluppo attraverso funzioni non deprecate e controllo continuo delle nuove funzionalità di QT5 e delle modifiche che questa fa nel funzionamento complessivo in modo di essere già compatibile quando sarà necessario passare a QT5 in quanto QT4 non sarà più prerequisito di QGIS.

Ovviamente si poteva avere lo stesso risultato utilizzando altre librerie grafiche esterne o quelle di python ma è stato scelto QT anche per avere una coerenza grafica tra le varie finestre che vanno ad aprirsi. Infatti se venisse utilizzata un'altra libreria grafica sarebbe necessario una rielaborazione della grafica standard della libreria in modo che si comporti coerentemente rispetto a QT per ogni sistema operativo, il che porterebbe a una complicazione del codice non necessaria.

¹Questo viene fatto automaticamente dall'interfaccia grafica selezionando il file desiderato

²Il linguaggio in cui è stato scritto il plugin

2.1.2 Gestione inputoutput

Input

Per una questione di semplicità è stato deciso di passare i dati in input come file, in modo che possano essere facilmente condivisi. Per fare ciò è stato necessario fare una ricerca sui tipi di file utilizzati per i dati georeferenziati.

Per questo motivo è stata fatta una ricerca sui formati di file supportati nativamente da QGis che fossero di tipo raster³ e quali fossero supportati attraverso plugin esterni.

E' stata quindi usata una delle funzioni del core di QGis per il riconoscimento automatico dei formati riconosciuti da QGis in modo da eseguire l'elaborazione del singolo file col driver corrispondente installato migliore.

Questo permette di avere un plugin più elastico, in quanto non necessita di aggiornamento se viene aggiunto il supporto a un tipo aggiuntivo di file per QGis ma recupera direttamente dalle configurazioni di QGis i dati necessari per la lettura, decodifica e rielaborazione del file in questione.

In oltre, il codice di gestione per i file in input è integrato nell'interfaccia grafica in modo da rendere semplice l'inserimento di dati all'interno del programma stesso. Supporta comunque l'utilizzo attraverso il terminale di QGis anche se ne è sconsigliato l'uso ai meno esperti e a chi si approccia per la prima volta a questo plugin.

Come vedete dall'immagine il plugin apre una finestra di sistema differente in base al sistema operativo utilizzato, in quanto la gestione della scelta del file è delegata a esso. Una volta avvenuta la scelta questa finestra si chiuderà passando la stringa corrispondente al path del file aperto al campo adiacente presente nella gui principale. Se l'utente lo desidera può editare o direttamente inserire il path al file scelto direttamente dall'utente.

Output

2.1.3 Gestione dati

Grass

Osgeo

2.1.4 Gestione output

³Unico tipo di dati che il plugin accetta in input in quanto è l'unica struttura dati adatta a contenere i dati che ci servono per l'elaborazione

Appendice A

Appendice

A.1 Strutturare un plugin

Bibliografia

- [Lut11] Mark Lutz. *Programming Python*. A cura di O'Reilly. O'Reilly, 2011.
- [Pan09] Lorenzo Pantieri. *L'arte di gestire la bibliografia con L^AT_EX*. 2009. URL: http://www.lorenzopantieri.net/LaTeX_files/Bibliografia.pdf.
- [PG11] Lorenzo Pantieri e Tommaso Gordini. *L'arte di scrivere con L^AT_EX*. 2011. URL: http://www.lorenzopantieri.net/LaTeX_files/ArteLaTeX.pdf.
- [Sita] URL: <http://www.qgis.org/it/>.
- [Sitb] URL: <http://creativecommons.org/licenses/by-sa/3.0/>.
- [Zen13] Michele Zen. “Metodi e strumenti per la costruzione di territori virtuali per l'applicazione di modelli e di scenari ambientali”. Tesi di laurea mag. Università Ca Foscari Venezia, 2013/2014.