



DEPARTMENT OF  
INFORMATION  
ENGINEERING  
UNIVERSITY OF PADOVA



**UNIVERSITÀ DEGLI STUDI DI PADOVA**  
**FACOLTÀ DI INGEGNERIA**

**UN APPROCCIO PER LA RIDUZIONE AUTOMATICA  
DELLE INCONGRUENZE TRA NUOVI DATI TOPOGRAFICI  
E DATI TEMATICI PREESISTENTI**

RELATORE: PROF. MASSIMO RUMOR

CORRELATORE : SANDRO SAVINO

LAUREANDO: ELISABETTA PANFILIO

CORSO DI LAUREA IN INGEGNERIA INFORMATICA  
ANNO ACCADEMICO 2012-2013

# INDICE

<b>INDICE .....</b>	<b>pag. 1</b>
<b>SOMMARIO .....</b>	<b>pag. 3</b>
<b>INTRODUZIONE AL PROBLEMA .....</b>	<b>pag. 4</b>
<b>ANALISI DEI DATI .....</b>	<b>pag. 8</b>
2.1 Analisi dei dati .....	pag. 8
2.2 Risultati dell'analisi .....	pag. 30
<b>SOLUZIONE PROPOSTA .....</b>	<b>pag. 31</b>
3.1 Descrizione della soluzione proposta .....	pag. 31
3.2 Descrizione dell'algoritmo .....	pag. 33
3.2.1 Prima parte: calcolo delle corrispondenze tra PRG e CTR e tra PRG e DBT .....	pag. 33
3.2.1.1 Analisi dei risultati ottenuti .....	pag. 35
3.2.2 Seconda parte: calcolo delle corrispondenze tra PRG e DBT senza corrispondenza nella CTR .....	pag. 38
3.2.2.1 Analisi dei risultati ottenuti .....	pag. 39
3.2.3 Terza parte: costruzione del nuovo PRG .....	pag. 40
3.2.3.1 Analisi dei risultati ottenuti .....	pag. 42
3.3 Considerazioni generali .....	pag. 47
<b>POSSIBILI EVOLUZIONI .....</b>	<b>pag. 48</b>
<b>CONSIDERAZIONIFINALI .....</b>	<b>pag. 49</b>
<b>BIBLIOGRAFIA E RIFERIMENTI UTILI .....</b>	<b>pag. 50</b>

<b>APPENDICE A: TECNOLOGIE UTILIZZATE .....</b>	<b>pag. 51</b>
A.1 POSTGRESQL .....	pag. 51
A.2 POSTGIS .....	pag. 53
A.3 OPENJUMP .....	pag. 54
A.4 ECLIPSE .....	pag. 55
A.5 Le classi del progetto CARGEN .....	pag. 56
<b>APPENDICE B: REALIZZAZIONE TECNICA .....</b>	<b>pag. 58</b>
<b>APPENDICE C: DESCRIZIONE DELLA BASE DI DATI .....</b>	<b>pag. 60</b>
<b>RINGRAZIAMENTI .....</b>	<b>pag. 65</b>

## SOMMARIO

Il presente lavoro affronta un problema rilevante per gli aspetti pratici che riveste, in particolare da parte degli enti locali e delle aziende di servizi a rete.

Queste organizzazioni possiedono molti dati “tematici” costruiti sulla base di dati topografici, spesso acquisiti dall'esterno. A seguito degli aggiornamenti di tali dati topografici risulta necessario ridisegnare manualmente i dati tematici con notevoli impegni di tempo e di risorse nonché con possibilità di generare errori .

Dopo aver analizzato uno dei casi più significativi, quello relativo al tema urbanistico, il lavoro formula una ipotesi di soluzione automatica. Viene poi descritto sinteticamente lo sviluppo di un algoritmo per implementare la soluzione ipotizzata e si dà conto dei risultati ottenuti.

I risultati sono soddisfacenti e si ritiene che quanto realizzato sia un passo significativo verso un meccanismo automatico che riduca significativamente l'attività manuale necessaria per l'adeguamento dei dati legacy a fronte di aggiornamenti della topografia.

## Capitolo Primo

# INTRODUZIONE AL PROBLEMA

---

La cartografia è uno dei campi in cui la ricerca si è concentrata negli ultimi anni per migliorare la qualità delle rappresentazioni geografiche sulla base dei dati rilevabili dal territorio.

I dati topografici permettono la rappresentazione della realtà fisica sul piano cartografico. Dato che, col passare del tempo, possono avvenire dei cambiamenti sugli elementi del territorio, i dati topografici subiscono aggiornamenti periodici per tenere conto dei cambiamenti apportati al territorio rappresentato.

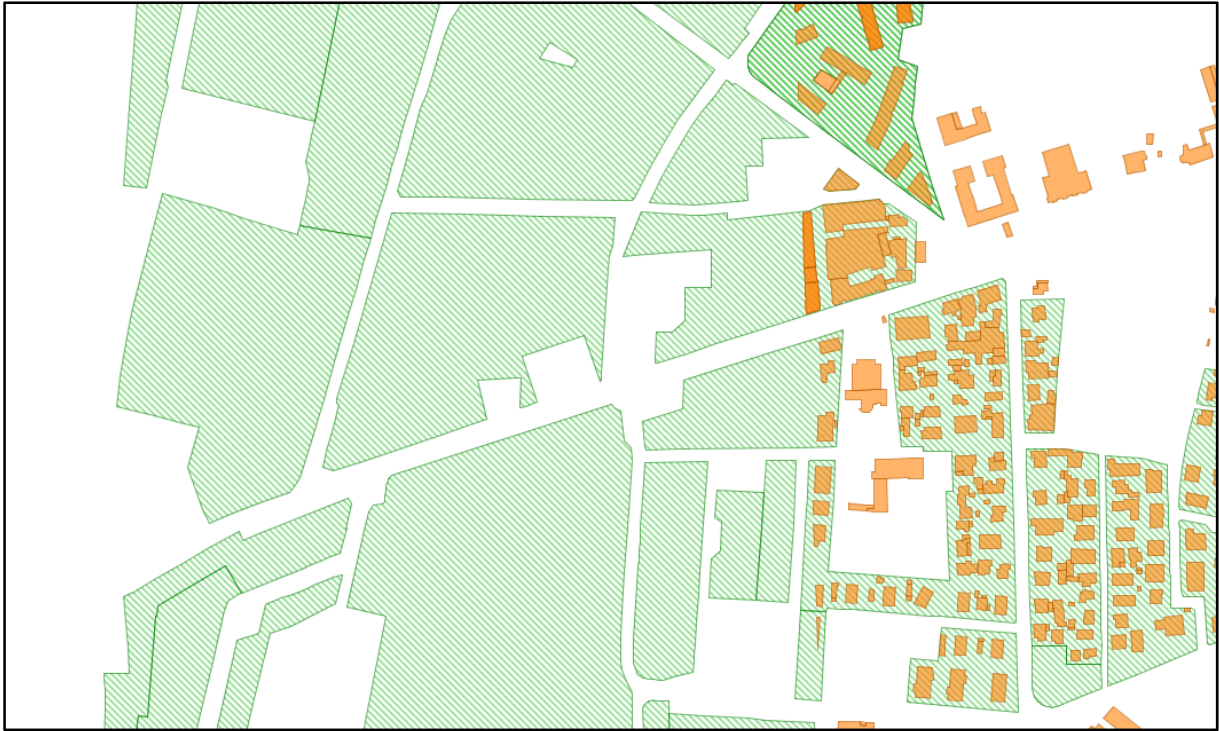
Oltre ai dati topografici esiste un'altra categoria di dati, i così detti dati tematici. Questi ultimi sono dati destinati ad usi specifici, come ad esempio:

- strumenti urbanisti, piano regolatore
- rappresentazione della proprietà: catasto
- carte pedologiche, climatiche
- ecc...

I dati tematici sono in genere rappresentati considerando le geometrie dei dati topografici, talvolta prendendoli come linea guida e in alcuni casi appoggiandovisi.

Dato che i dati topografici vengono aggiornati periodicamente, i dati tematici esistenti, cioè quelli rappresentati sulla base dei dati topografici precedenti l'aggiornamento, non sono più congruenti, e di conseguenza devono essere a loro volta modificati e aggiornati.

Un esempio tipico è rappresentato dal Piano Regolatore Generale (PRG).



**Figura 1.1.** La figura rappresenta una porzione di cartografia destinata a diventare area residenziale. Le aree verdi sono le zone residenziali che fanno parte del PRG. I blocchi arancioni rappresentano i fabbricati che fanno parte della vecchia cartografia.



**Figura 1.2.** La figura illustra come è stata modificata la zona rappresentata nella figura 1.1 nella nuova cartografia. Le aree verdi sono le zone residenziali che fanno parte del PRG, le aree rosa rappresentano le aree stradali, i blocchi gialli rappresentano gli edifici, elementi fanno parte della nuova cartografia.

Come si può notare anche dalle figure precedenti, una situazione tipica è quella degli strumenti urbanistici (esempio del PRG) caso in cui è previsto dalle norme un procedimento di adeguamento specifico. Attualmente questo tipo di operazioni di adeguamento viene svolto a mano. Visto che si tratta di un impegno di tempo rilevante e di attività soggetta ad errori, anche gravi, vi è la necessità di tentare una soluzione automatica che svolga queste operazioni nel modo più preciso possibile.

Per quanto riguarda i dati topografici la Regione del Veneto gestisce la progettazione, la redazione, la diffusione e l'aggiornamento della Carta Tecnica Regionale (CTR), nonché di carte tematiche e di altre elaborazioni.

La Carta Tecnica Regionale è una cartografia generale e metrica prodotta nelle scale di rappresentazione 1:5000 e 1:10000.

I particolari esistenti sul territorio, di origine naturale ed artificiale, riportati nella cartografia, sono rappresentati, nelle loro dimensioni, alla scala della carta, secondo le normative stabilite dalla Regione Veneto.

Le sue caratteristiche di precisione, qualità e rappresentazione dei particolari, la rendono un documento idoneo per ogni impiego di progettazione di massima di opere ed infrastrutture, strumento per la pianificazione urbanistica e territoriale ed efficace base per l'allestimento di qualsiasi rappresentazione di carte tematiche.

Oltre ad una cartografia generale vengono messi a disposizione dei dati tematici , rappresentati da un Piano Regolatore Generale (PRG) che rappresenta la divisione del territorio del comune considerato in zone sulla base della destinazione d'uso del territorio prevista.

Il Piano Regolatore Generale considera la totalità del territorio comunale e indica essenzialmente:

- la rete delle principali vie di comunicazione stradali, ferroviarie e navigabili e dei relativi impianti;
- la divisione in zone del territorio comunale con la precisazione delle zone destinate all'espansione dell'aggregato urbano e la determinazione dei vincoli e dei caratteri da osservare in ciascuna zona;
- le aree destinate a formare spazi di uso pubblico o sottoposte a speciali servizi;

- le aree da riservare ad edifici pubblici o di uso pubblico nonché ad opere ed impianti di interesse collettivo o sociale;
- i vincoli da osservare nelle zone a carattere storico, ambientale, paesistico;
- le norme per l'attuazione del piano. (art.7 L.1150 del 17 agosto 1942).

Il PRG ha generalmente una validità di dieci anni, ma le previsioni in esso contenute possono essere protratte nel tempo oltre ai termini indicati e/o modificati in tutte, o in parte, mediante variante che ne adegua le previsioni e le caratteristiche in relazione alle esigenze che nel tempo vengono manifestate.

Il PRG è stato progressivamente modificato con varianti generali che ne hanno modificato la struttura e varianti parziali volte a risolvere problemi specifici, fino alla formulazione attuale data dal PRG. Va tenuto presente che il PRG è stato costruito sulla base dei dati presenti nella CTR.

Di recente è stato adottato un nuovo tipo di cartografia costituito da un DataBase Territoriale (DBT) che contiene i dati aggiornati degli elementi presenti sul territorio suddivisi secondo la loro natura, e con una struttura dei dati modificata rispetto alla Carta Tecnica Regionale. Di conseguenza è sorta la necessità di adattare il Piano Regolatore Generale, costruito sulla Carta Tecnica Regionale, alla nuova cartografia per ottenere dei dati tematici il più possibile aggiornati.

Per fare questo è stata fatta un'analisi generale dei dati relativi alla Carta Tecnica Regionale, al Piano Regolatore Generale e al DataBase Territoriale.

In seguito si è passati alla costruzione del programma vero e proprio che ha lo scopo di confrontare i dati e di modificare le geometrie del Piano Regolatore Generale nel modo più opportuno.

Nei capitoli che seguono verranno analizzati in dettaglio le varie fasi di sviluppo del progetto.



## Capitolo Secondo

# ANALISI DEI DATI

---

Nel paragrafo che segue verrà fatta un'analisi generale dei dati geografici relativi alla città di Padova. Verrà indicato con CTR la Carta Tecnica Regionale, con PRG il Piano Regolatore Generale e con DBT il DataBase Territoriale.

L'analisi di questi dati permette la determinazione delle casistiche e problematiche più importanti da affrontare nella fase di risoluzione del problema.

## 2.1 ANALISI DEI DATI

I dati analizzati riguardano la città di Padova. Se si sovrappongono i dati del DBT con quelli della CTR ci si rende conto che pur rappresentando entrambi la stessa zona geografica, sono diversi e questo è dovuto al fatto che alcune geometrie sono cambiate nel frattempo perché la città si è trasformata oppure che alcune geometrie anche se non sono cambiate sono state rappresentate in maniera diversa in relazione alle tolleranze di acquisizione.

Se si sovrappongono i PRG con il DBT e la CTR, ci si accorge che i PRG si appoggiano bene con la CTR, mentre hanno delle discrepanze con il DBT. Questo è dovuto al fatto che quando sono stati fatti i piani regolatori il DBT non c'era sono stati costruiti sulla base della CTR.

Di seguito verranno analizzati ad uno ad uno alcuni degli elementi principali del DBT confrontandoli con il PRG e in alcuni casi con la CTR.

A\_PED (area pedonale):

Nella cartografia della città sono state rilevate delle aree pedonali sovrapposte ad aree residenziali dove in precedenza erano presenti dei fabbricati. Questo errore è da risolvere se

la variazione è minima. Se la sovrapposizione è evidente o c'è stato un errore o è stata fatta una modifica sul fabbricato in modo che l'area pedonale si appoggiasse sul nuovo edificio, questo errore va comunque risolto con una verifica dei dati o sul territorio. Nelle figure seguenti viene presentato un esempio di tale situazione.

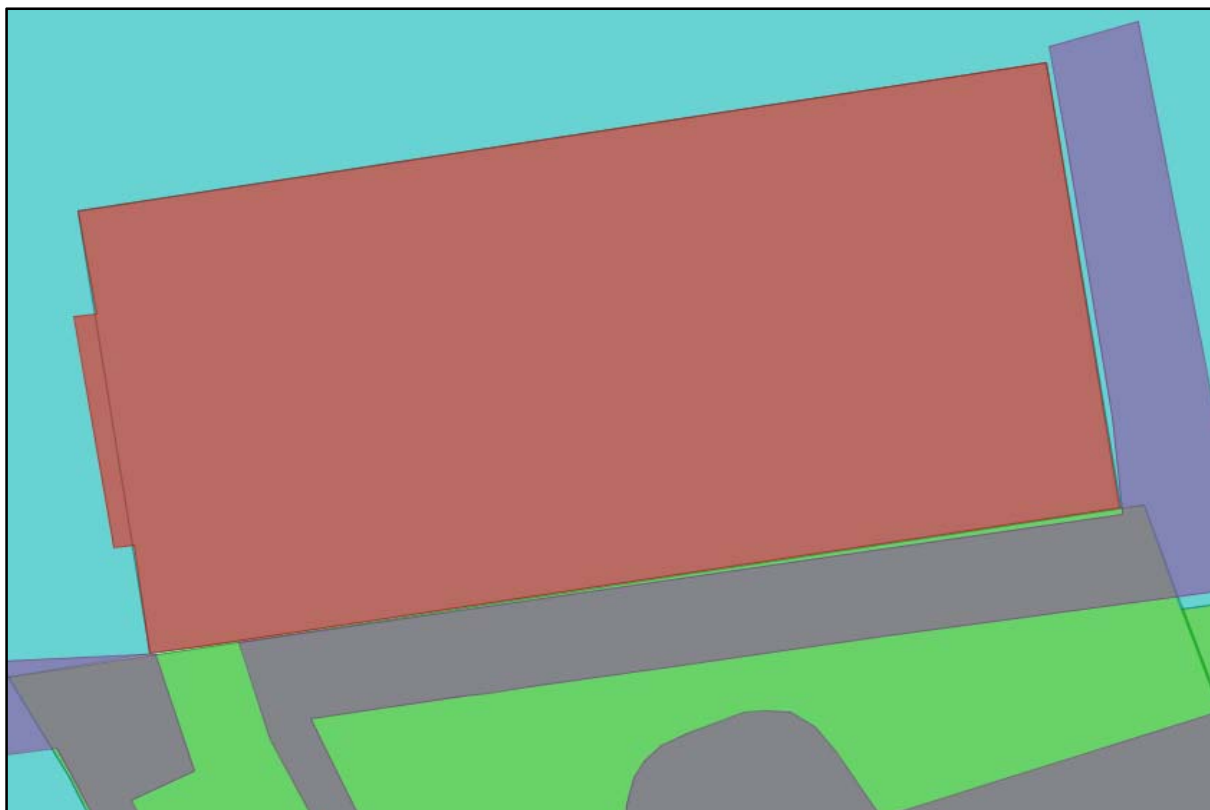


**Figura 2.1.** Le aree verdi sono le zone residenziali, quelle azzurre sono i servizi di quartiere e quelle viola sono le A\_PED



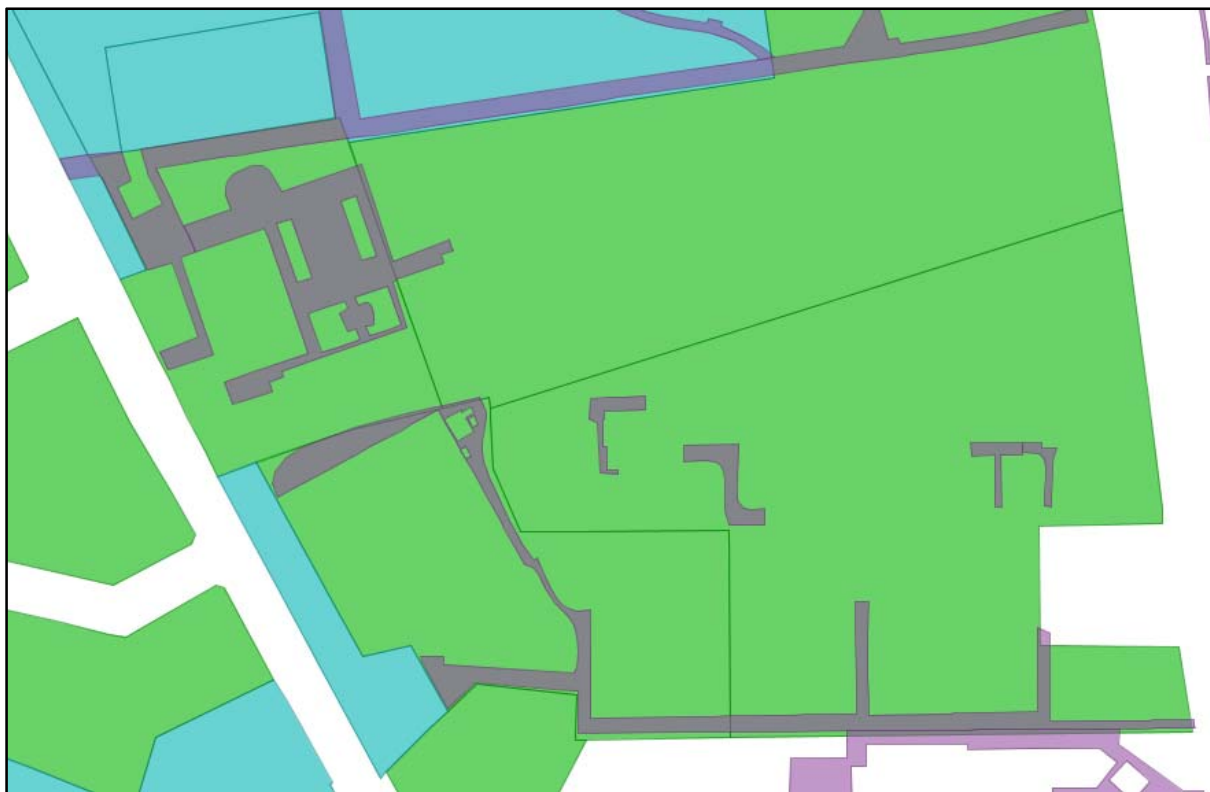
**Figura 2.2.** Le aree verdi sono le zone residenziali, quelle azzurre sono i servizi di quartiere, i blocchi rossi sono i fabbricati della vecchia cartografia e quelle viola sono le A\_PED

Sono stati inoltre rilevati dei casi in cui le aree pedonali non si appoggiano ai fabbricati, probabilmente i fabbricati o le aree pedonali sono stati costruiti dopo. Se la variazione è minima si potrebbe far combaciare il confine dell'area pedonale con quella del fabbricato. Nella figura seguente viene rappresentato un esempio di questa casistica.



**Figura 2.3.** Le aree verdi sono le zone residenziali, quelle azzurre sono i servizi di quartiere, i blocchi rossi sono i fabbricati della vecchia cartografia e quelle viola sono le A\_PED

Infine è possibile che in alcuni casi le linee delle aree pedonali non combacino con le linee del PRG. Si potrebbero far combaciare.



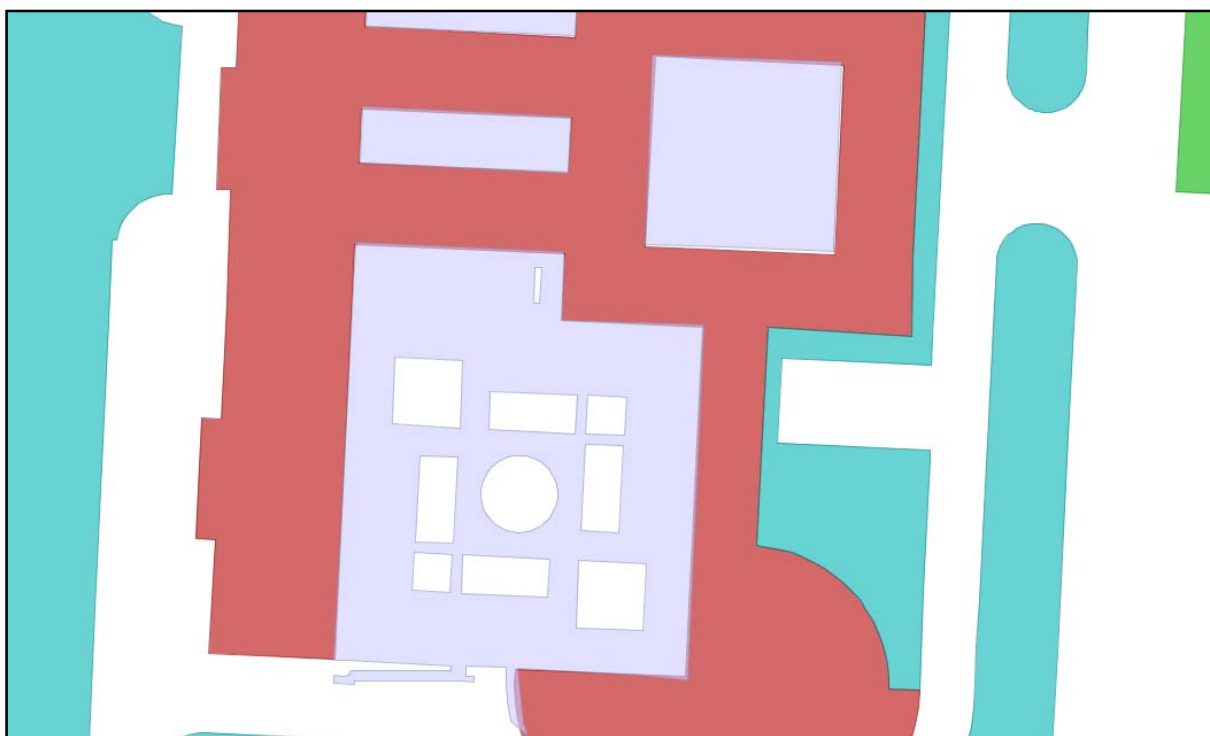
**Figura 2.4.** Le aree verdi sono le zone residenziali, quelle azzurre sono i servizi di quartiere e quelle viola sono le A\_PED

AC\_PED(area di circolazione pedonale):

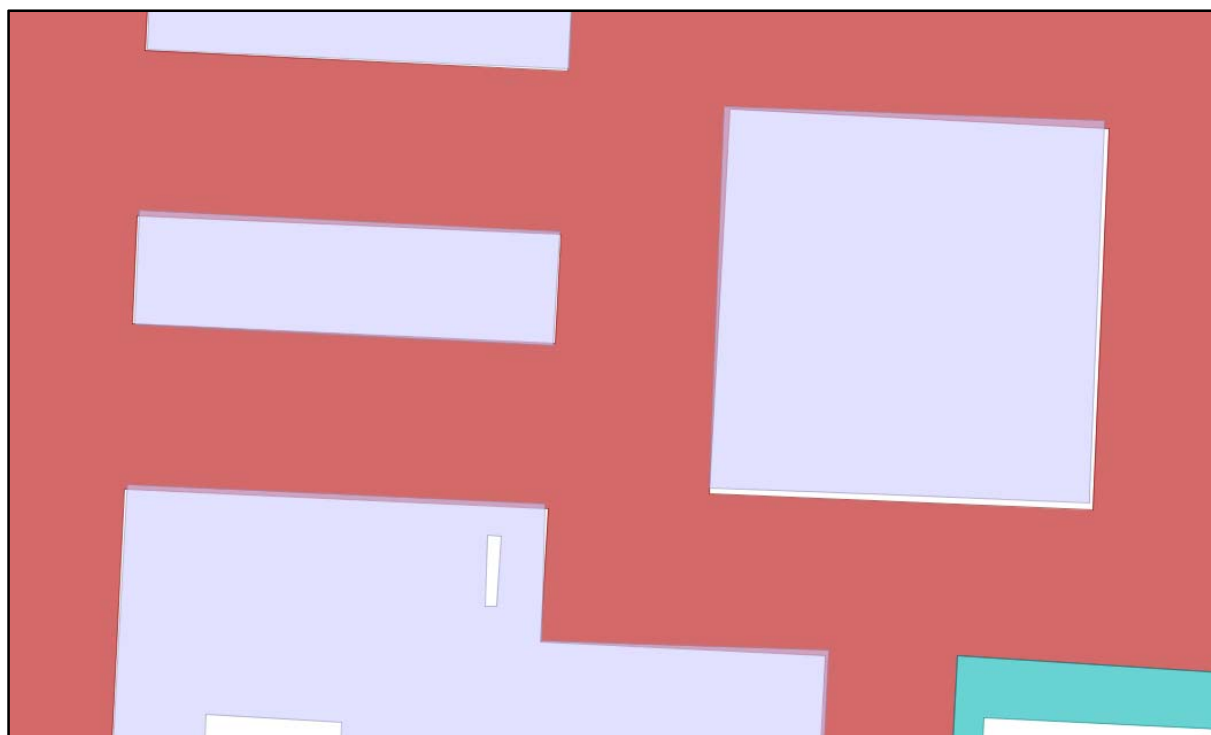
Per le aree di circolazione pedonale alcune linee di confine non coincidono di pochissimo con le linee delle aree del PRG. Si potrebbe risolvere il problema facendole combaciare.



**Figura 2.5.** Le aree verdi sono le zone residenziali, quelle azzurre sono i servizi di quartiere, le aree rosse sono le dir\_comm\_ind e le aree grigie sono le AC\_PED



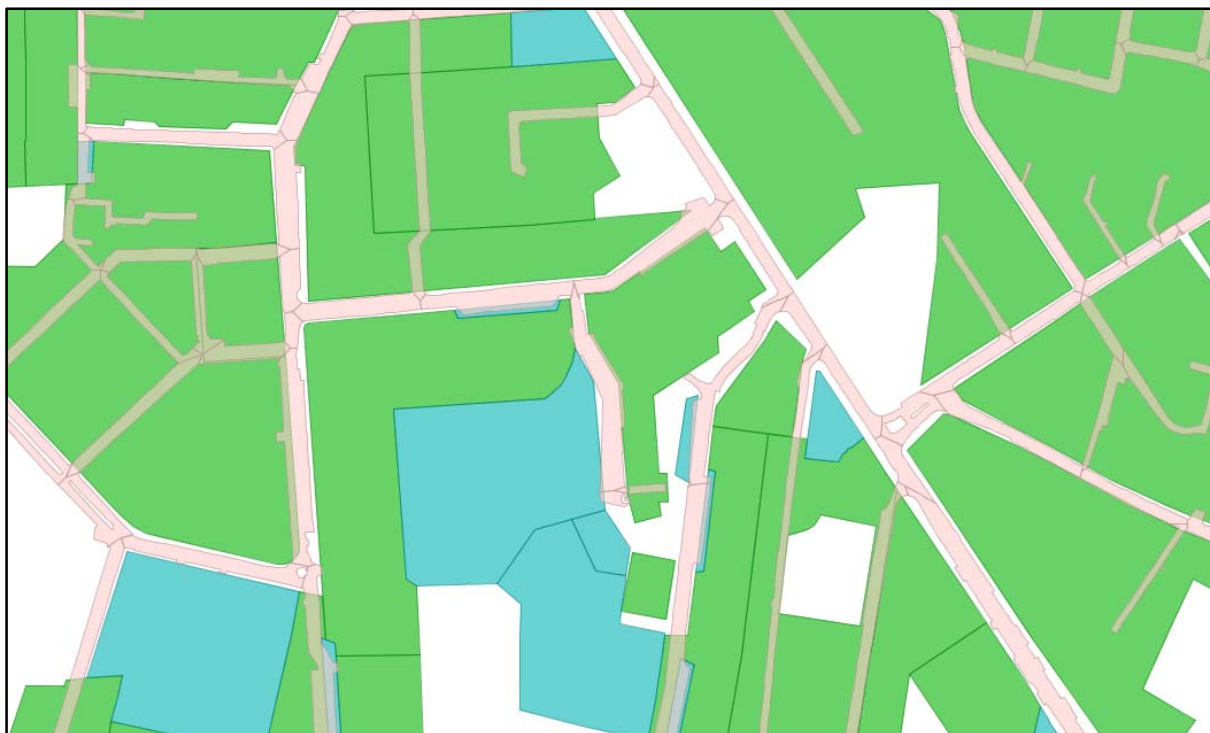
**Figura 2.6.** Le aree verdi sono le zone residenziali, quelle azzurre sono i servizi di quartiere, le aree rosse sono le dir\_comm\_ind e le aree grigie sono le AC\_PED



**Figura 2.7.** Le aree azzurre sono i servizi di quartiere, le aree rosse sono le dir\_comm\_ind e le aree grigie sono le AC\_PED

AC\_VEI (area di circolazione veicolare):

Per le aree di circolazione veicolare alcune linee di confine non combaciano con le linee delle aree del PRG. Se non ci sono marciapiedi o altri elementi si possono far combaciare.



**Figura 2.8.** Le aree verdi sono le zone residenziali, quelle azzurre sono i servizi di quartiere e le aree rosa sono le AC\_VEI

Altre linee sono staccate e vanno lasciate così.



**Figura 2.9.** Le aree verdi sono le zone residenziali, quelle azzurre sono i servizi di quartiere, le aree rosse sono le dir\_comm\_ind e le aree rosa sono le AC\_VEI

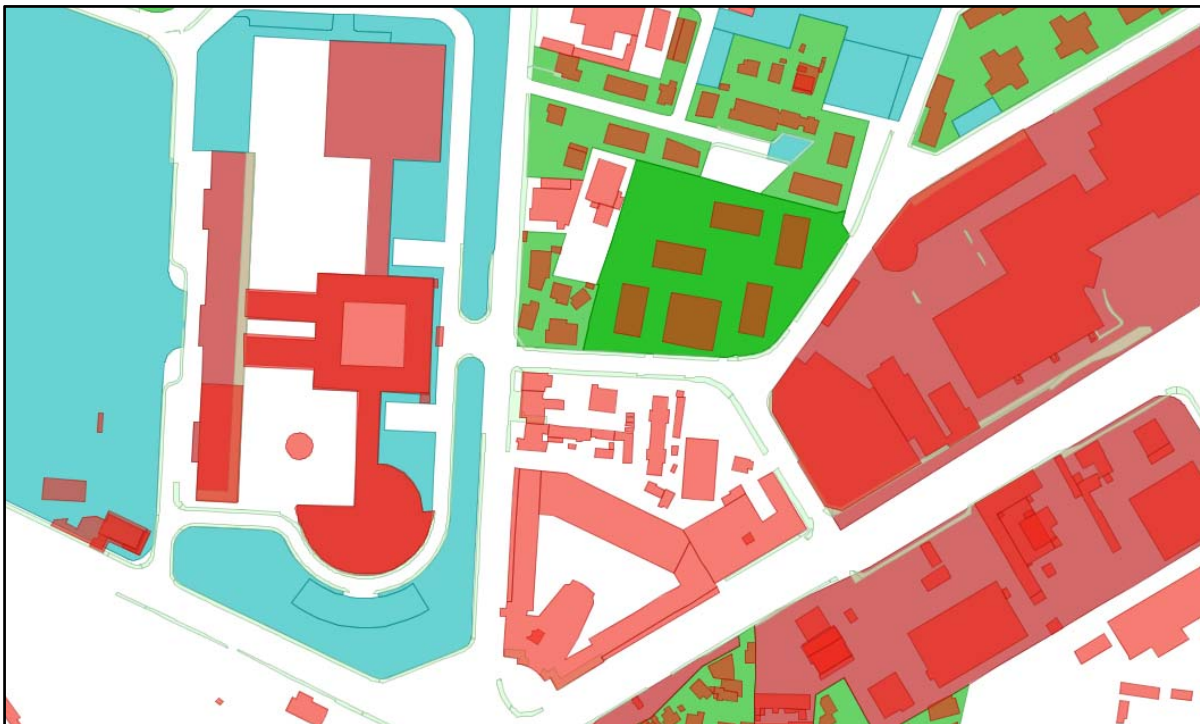


AR\_MARC (area marciapiedi):

Ci sono pezzi di marciapiede che sono sovrapposti a luoghi dove nella vecchia cartografia erano presenti dei fabbricati. Bisogna verificare la veridicità delle informazioni.

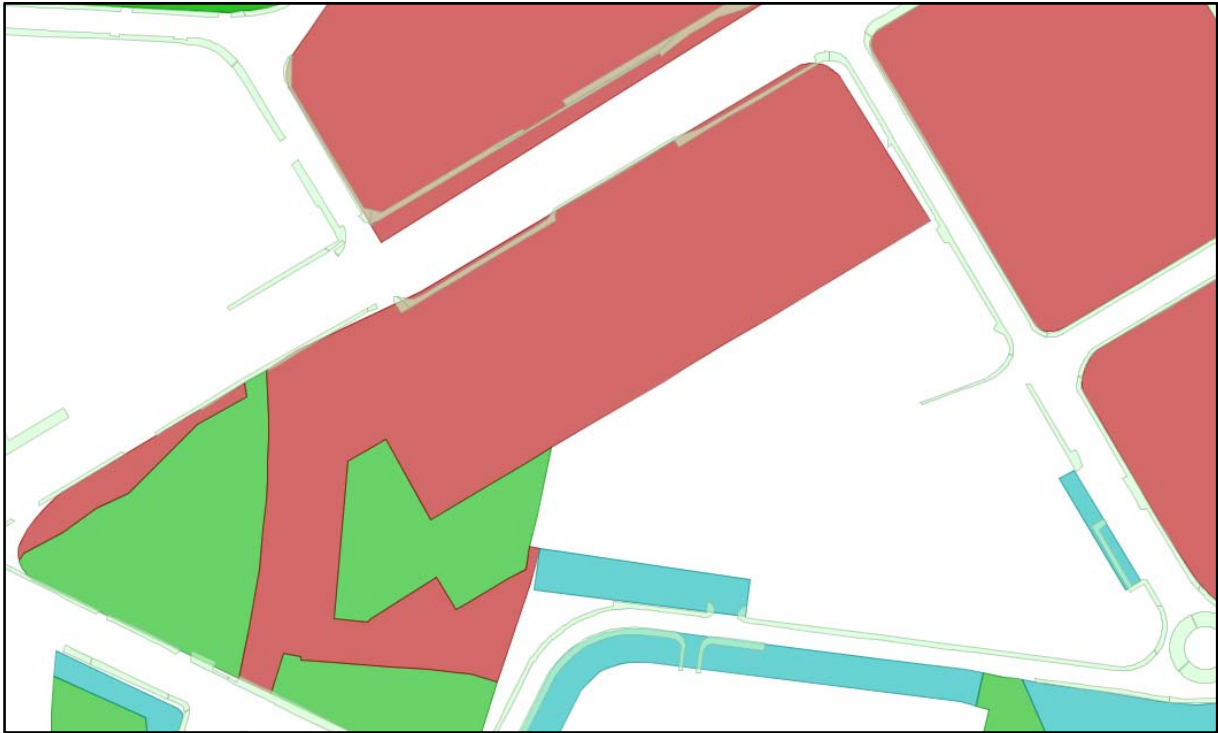


**Figura 2.10.** Le aree verdi sono le zone residenziali, quelle azzurre sono i servizi di quartiere, le aree rosse sono le dir\_comm\_ind e le aree azzurro chiaro sono le AR\_MARC



**Figura 2.11.** Le aree verdi sono le zone residenziali, quelle azzurre sono i servizi di quartiere, le aree rosse sono le dir\_comm\_ind, i blocchi rossi sono i fabbricati della vecchia cartografia e le aree azzurro chiaro sono le AR\_MARC

In alcuni casi ci sono leggere sovrapposizioni che si possono risolvere facendo combaciare le linee, sempre che non ci siano in mezzo altri elementi.



**Figura 2.12.** Le aree verdi sono le zone residenziali, quelle azzurre sono i servizi di quartiere, le aree rosse sono le dir\_comm\_ind e le aree azzurro chiaro sono le AR\_MARC

AR\_STR (area stradale):

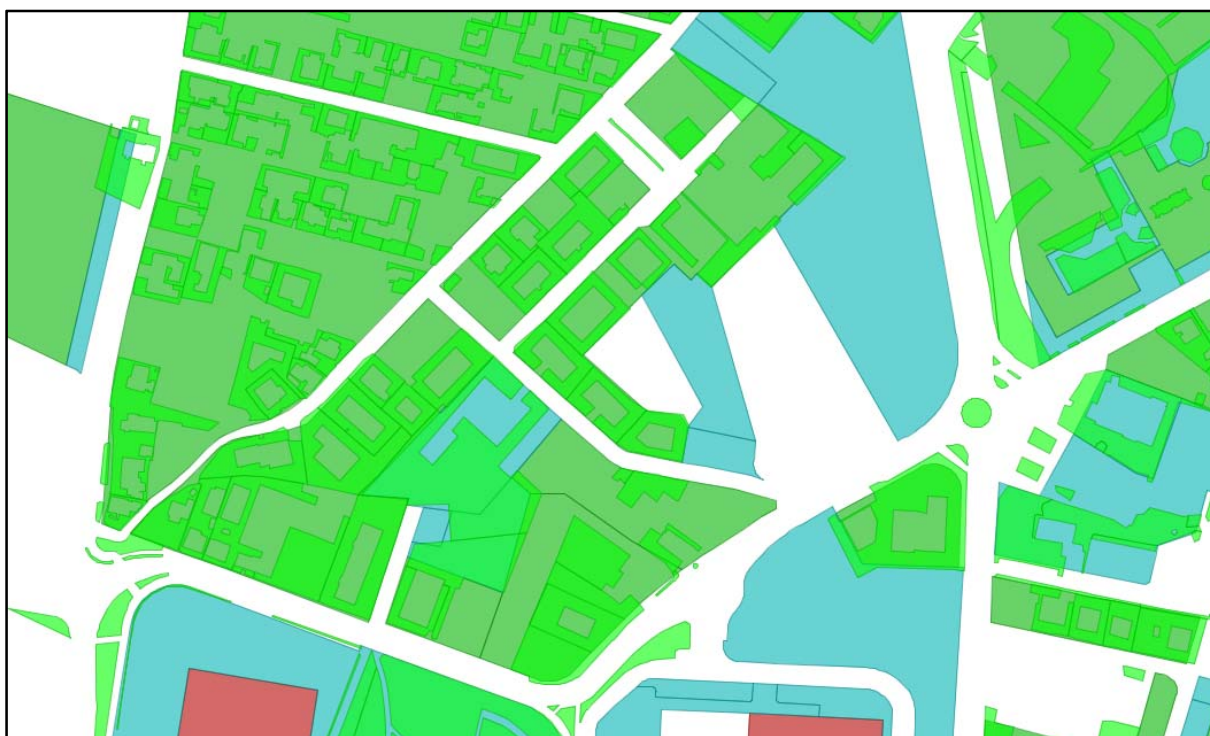
L'area stradale è costituita dall'area dei marciapiedi e dall'area di circolazione stradale. Vale quello detto per i marciapiedi. Dove non ci sono marciapiedi valgono comunque gli stessi criteri.

AR\_VRD (area verde):

Per le aree verdi alcuni blocchi non coincidono con le aree del PRG, ci sono variazioni rilevanti nell'area residenziale. Le variazioni minime si potrebbero far combaciare, ma quelle più rilevanti necessitano di una verifica più approfondita.



**Figura 2.13.** Le aree verdi sono le zone residenziali, quelle azzurre sono i servizi di quartiere, le aree rosse sono le dir\_comm\_ind e le aree verdi sopra le altre aree sono le AR\_VRD



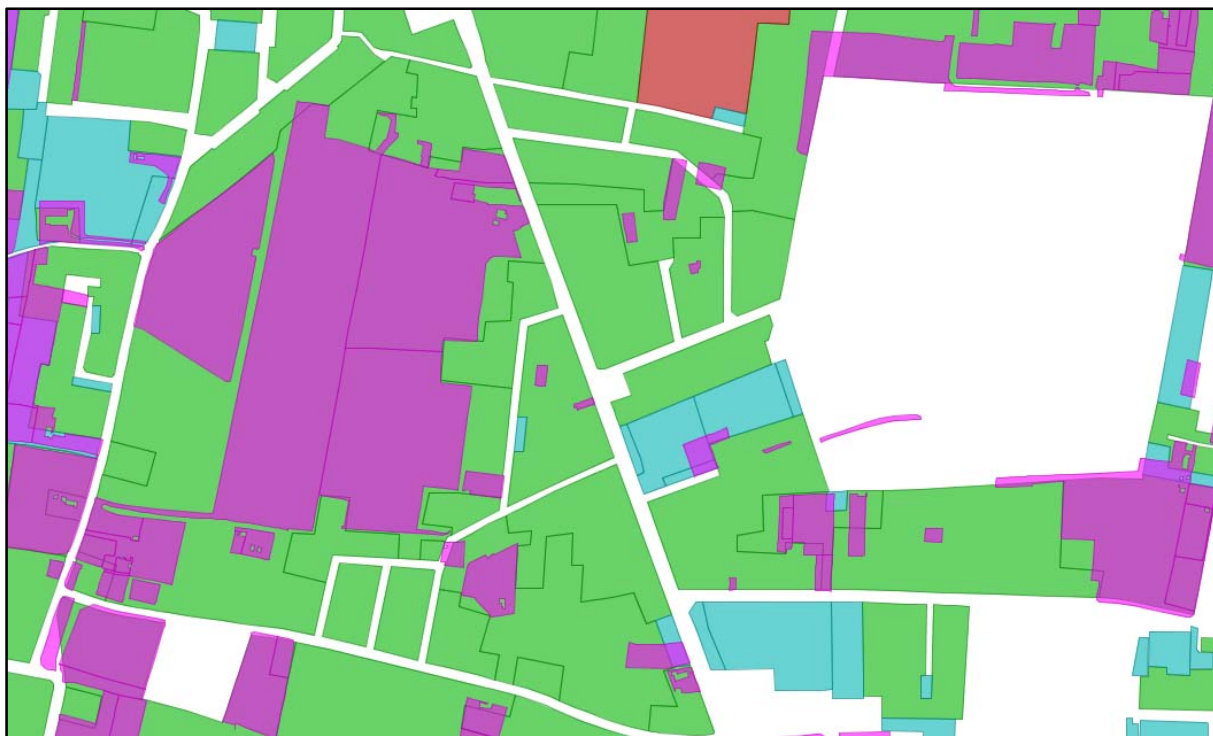
**Figura 2.14.** Le aree verdi sono le zone residenziali, quelle azzurre sono i servizi di quartiere, le aree rosse sono le dir\_comm\_ind e le aree verdi sopra le altre aree sono le AR\_VRD



**Figura 2.15.** Le aree verdi sono le zone residenziali, quelle azzurre sono i servizi di quartiere, le aree rosse sono le *dir\_comm\_ind* e le aree verdi sopra le altre aree sono le *AR\_VRD*

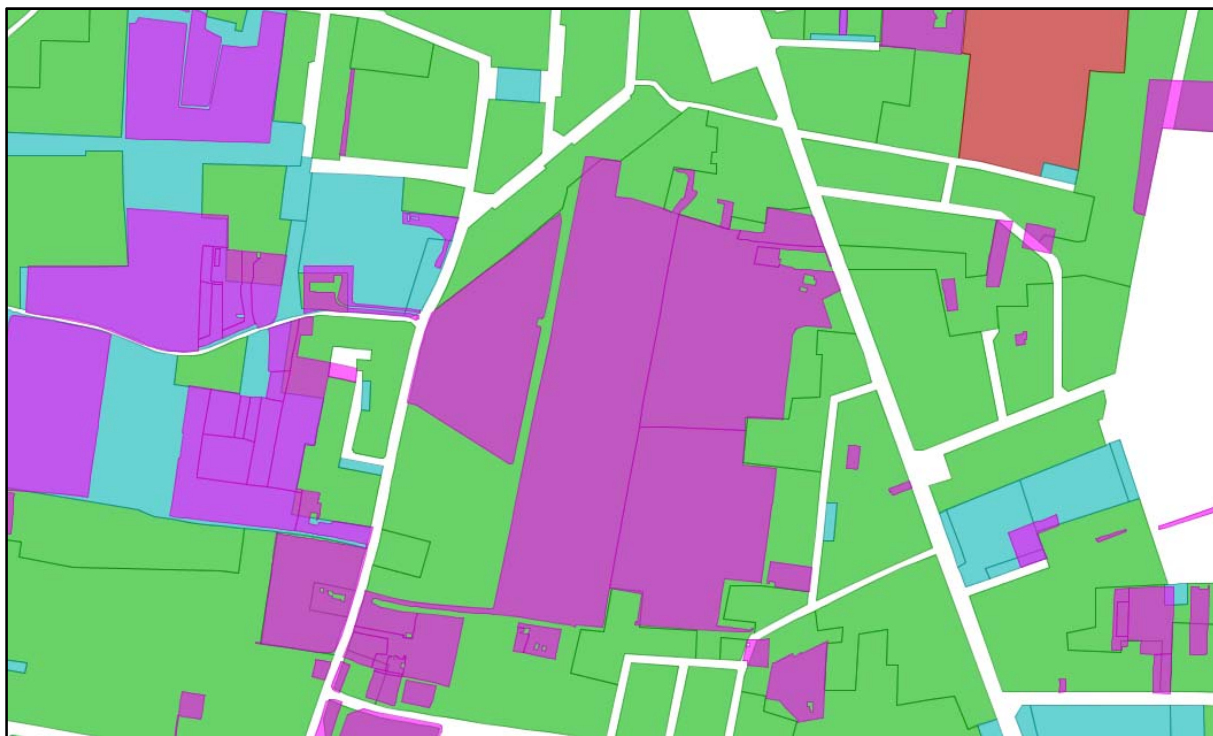
CL\_AGR (coltura agricola):

Per le colture agricole ci sono aree che non sono contenute all'interno delle aree del PRG e sfiorano di poco. Va fatta una verifica su quali siano i dati corretti.



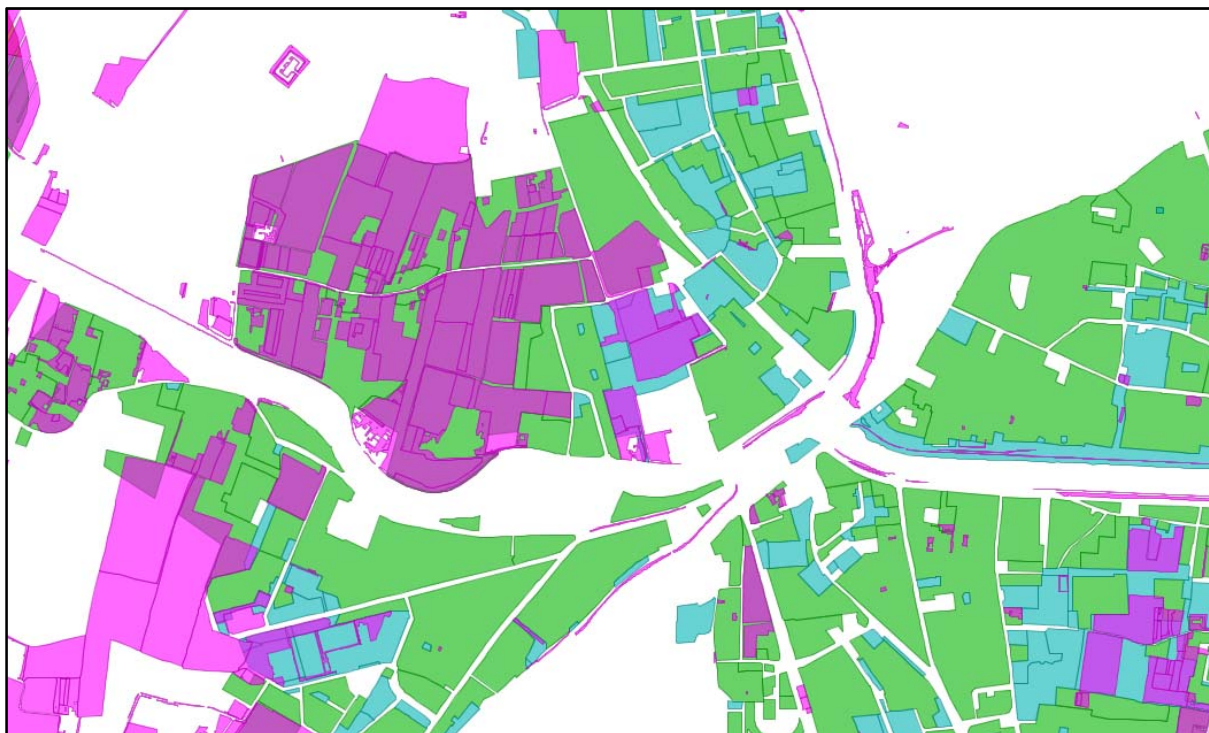
**Figura 2.16.** Le aree verdi sono le zone residenziali, quelle azzurre sono i servizi di quartiere, le aree rosse sono le dir\_comm\_ind e le aree fucsia sono le CL\_AGR

Alcune linee leggermente spostate si possono far combaciare con quelle del piano regolatore.



**Figura 2.17.** Le aree verdi sono le zone residenziali, quelle azzurre sono i servizi di quartiere, le aree rosse sono le dir\_comm\_ind e le aree fucsia sono le CL\_AGR

Alcune aree sono sia interne che esterne alle aree del PRG, va verificata la loro correttezza.



**Figura 2.18.** Le aree verdi sono le zone residenziali, quelle azzurre sono i servizi di quartiere, le aree rosse sono le `dir_comm_ind` e le aree fucsia sono le `CL_AGR`

EL\_DIV (elemento divisorio):

Ci sono elementi divisori che non combaciano con i confini delle aree del PRG. Se gli elementi sono ravvicinati si possono correggere facendo combaciare le linee.



**Figura 2.19.** Le aree verdi sono le zone residenziali, quelle azzurre sono i servizi di quartiere, le aree rosse sono le dir\_comm\_ind e le linee grigie sono le EL\_CIC



**Figura 2.20.** Le aree verdi sono le zone residenziali, quelle azzurre sono i servizi di quartiere, le aree rosse sono le dir\_comm\_ind e le linee grigie sono le EL\_CIC

Alcuni elementi divisori della vecchia cartografia sono stati spostati, forse sono state fatte delle modifiche strutturali.



**Figura 2.21.** Le aree verdi sono le zone residenziali, quelle azzurre sono i servizi di quartiere, le aree rosse sono le dir\_comm\_ind e le linee blu sono gli elementi divisori della vecchia cartografia



**Figura 2.22.** Le aree verdi sono le zone residenziali, quelle azzurre sono i servizi di quartiere, le aree rosse sono le dir\_comm\_ind e le linee viola sono le EL\_DIV





**Figura 2.23.** Le aree verdi sono le zone residenziali, quelle azzurre sono i servizi di quartiere, le aree rosse sono le dir\_comm\_ind, le linee blu sono gli elementi divisori della vecchia cartografia e le linee viola sono le EL\_DIV

Le differenze meno evidenti si potrebbero risolvere, ma le altre vanno verificate.

PE\_UINS (unità insediativa):

Per questi elementi alcune linee di confine non combaciano di poco e si potrebbero sistemare.



**Figura 2.24.** Le aree verdi sono le zone residenziali e le aree viola sono le PE\_UINS

Altre linee, invece, sono più scostate e andrebbero verificate.



**Figura 2.25.** Le aree verdi sono le zone residenziali, quelle azzurre sono i servizi di quartiere e le aree viola sono le PE\_UINS

UN\_VOL (unità volumetrica):

Per le unità volumetriche alcuni elementi sono fuori dalle zone del PRG.

Alcuni elementi, invece, non sono completamente contenuti nelle aree del PRG.



**Figura 2.26.** Le aree verdi sono le zone residenziali, quelle azzurre sono i servizi di quartiere e i blocchi marroni sono le UN\_VOL



**Figura 2.27.** Le aree verdi sono le zone residenziali, quelle azzurre sono i servizi di quartiere e i blocchi marroni sono le UN\_VOL

I bordi di alcuni elementi sono molto vicini rispetto ai confini delle aree del PRG, bisogna verificare se si possono spostare o meno.

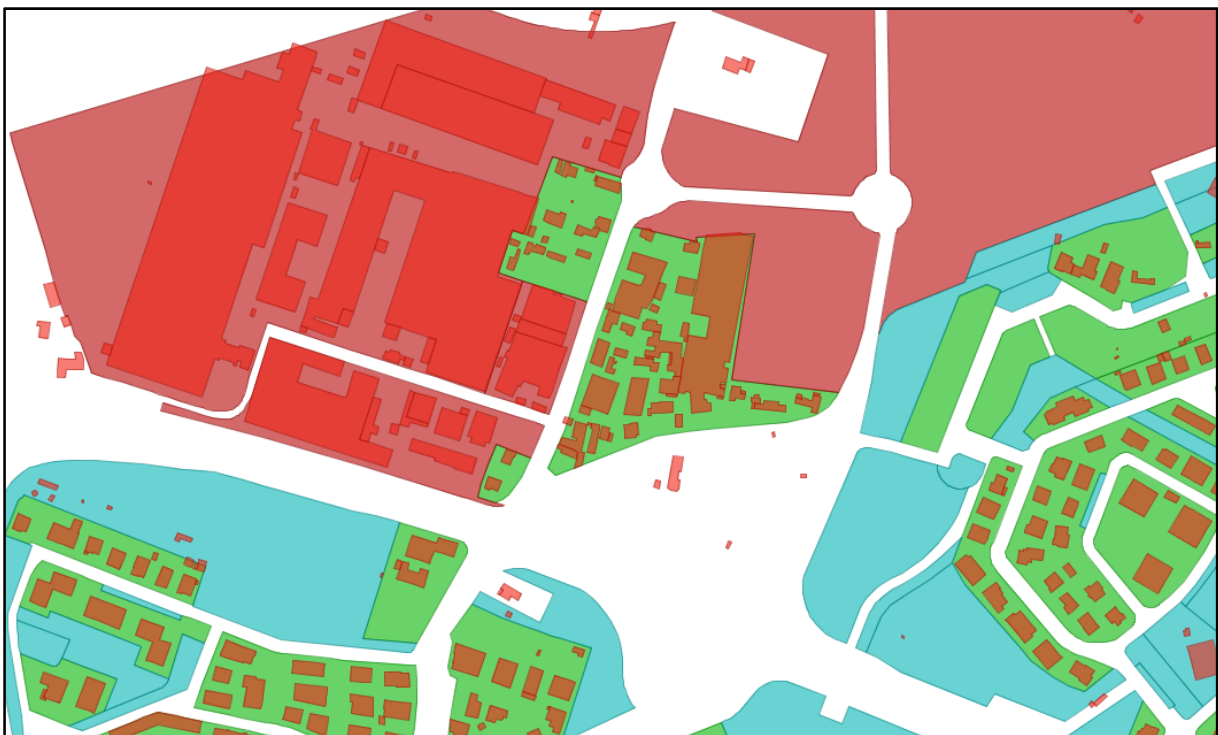


**Figura 2.28.** Le aree verdi sono le zone residenziali, quelle azzurre sono i servizi di quartiere e i blocchi marroni sono le UN\_VOL

Alcune unità volumetriche coincidono con i vecchi fabbricati, altre sono leggermente differenti, forse sono stati modificati dopo. Ce ne sono anche di nuovi.



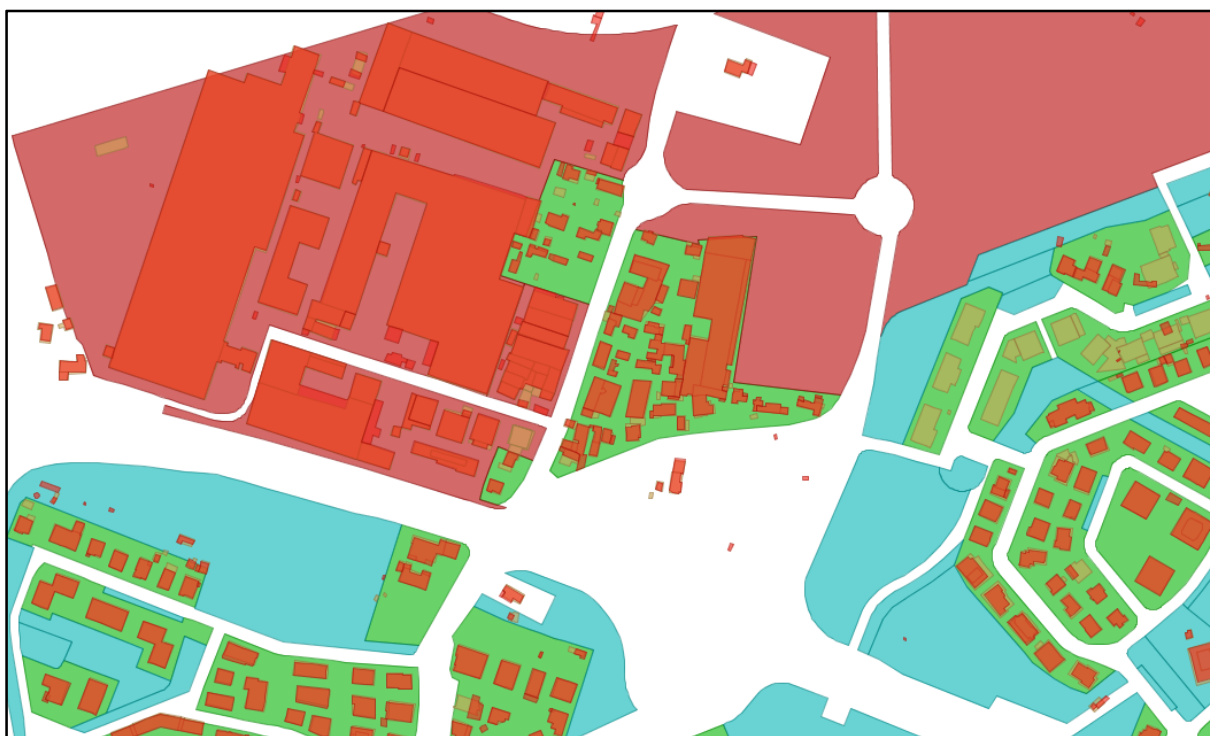
**Figura 2.29.** Le aree verdi sono le zone residenziali, quelle azzurre sono i servizi di quartiere, i blocchi rossi sono i fabbricati della vecchia cartografia e i blocchi marroni sono le UN\_VOL



**Figura 2.30.** Le aree verdi sono le zone residenziali, quelle azzurre sono i servizi di quartiere, le aree rosse sono le dir\_comm\_ind e i blocchi rossi sono i fabbricati della vecchia cartografia



**Figura 2.31.** Le aree verdi sono le zone residenziali, quelle azzurre sono i servizi di quartiere, le aree rosse sono le dir\_comm\_ind e i blocchi marroni sono le UN\_VOL



**Figura 2.32.** Le aree verdi sono le zone residenziali, quelle azzurre sono i servizi di quartiere, le aree rosse sono le dir\_comm\_ind, i blocchi rossi sono i fabbricati della vecchia cartografia e i blocchi marroni sono le UN\_VOL

## **2.2 RISULTATI DELL'ANALISI**

Dopo aver analizzato i dati topografici sono stati riscontrati diversi aspetti importanti da considerare.

In primo luogo c'è da notare la differenza tra i dati contenuti nella CTR e i dati presenti nel DBT. Come è già stato detto precedentemente la loro struttura è diversa, inoltre durante l'analisi dei dati sono state determinate differenze per quanto riguarda le geometrie. Questo è dovuto al fatto che nella nuova cartografia sono presenti modifiche fatte sul territorio successivamente alla redazione della cartografia CTR.

Oltre ad analizzare le differenze tra CTR e DBT, sono stati analizzati i dati topografici in relazione al PRG.

Confrontando gli elementi della CTR con il PRG è stata riscontrato che alcune parti di PRG si appoggiano perfettamente su elementi della CTR, mentre altre parti non hanno corrispondenze. Nel secondo caso è possibile che il PRG sia stato approssimato considerando elementi della CTR nelle vicinanze. Se non è così non esistono elementi della CTR su cui appoggiarsi e probabilmente quelle parti di PRG sono state costruite seguendo altri criteri.

Prendendo poi in considerazione anche il DBT, è stato riscontrato che alcuni elementi della CTR di appoggio per il PRG vengono mantenuti tali anche nella nuova cartografia, oppure subiscono piccole variazioni. In questi casi si può prendere come nuovi elementi di appoggio questi elementi del DBT. Nei casi in cui il PRG non ha elementi di appoggio nella CTR è possibile però che abbia delle corrispondenze nel DBT, e questi elementi vanno considerati come elementi di appoggio al pari dei precedenti.

Nell'analisi sono stati considerati tutti gli elementi presenti nel DBT, ma qui sono stati riportati solamente quelli più significativi per lo sviluppo del problema.

## Capitolo Terzo

# SOLUZIONE PROPOSTA

---

Nel capitolo seguente verranno analizzati in dettaglio i vari aspetti della soluzione proposta realizzata al fine di risolvere il problema presentato nei capitoli precedenti. Dopo aver analizzato l'algoritmo, ci si soffermerà sull'analisi dei problemi riscontrati nella fase di sviluppo dell'algoritmo.

### 3.1 DESCRIZIONE DELLA SOLUZIONE PROPOSTA

Prima di passare ad analizzare l'algoritmo vero e proprio è bene fare una breve analisi dei passi implementati per risolvere il problema in questione.

Si è scelto di realizzare una soluzione modulare, suddividendo l'algoritmo in varie parti, dove ognuna svolge una funzione specifica.

La soluzione proposta svolge le seguenti funzioni:

- Calcola le corrispondenze tra PRG e CTR determinando i punti di appoggio sulla base dei quali è stato costruito il PRG;
- Calcola i punti di appoggio sui quali dovranno appoggiarsi le nuove geometrie del piano regolatore;
- Aggiorna le geometrie del PRG in modo che i confini delle aree si appoggino sui nuovi punti di appoggio.

#### Calcolo delle corrispondenze tra PRG e CTR

Come prima cosa vanno calcolare le corrispondenze tra i confini delle aree della CTR e i confini delle aree del PRG. Questo permette di confrontare la cartografia vecchia e il PRG in modo da poter determinare su quali elementi della vecchia cartografia (CTR) ci si è



appoggiati per costruire il piano regolatore (PRG). Va tenuto presente che esistono parti di PRG che non hanno alcuna corrispondenza con la CTR. Trovate le corrispondenze, queste verranno proiettate sul segmento di PRG corrispondente.

### **Calcolo dei nuovi punti di appoggio**

Una volta determinate le corrispondenze tra PRG e CTR e le loro proiezioni sul PRG, è necessario determinare i punti d'appoggio sui quali dovrà appoggiarsi il nuovo piano regolatore. Per fare questo si confrontano le proiezioni delle corrispondenze tra CTR e PRG, che in pratica saranno delle linee, con gli elementi corrispondenti del DBT. Va tenuto presente che esistono delle parti della CTR che non hanno alcuna corrispondenza nel DBT, di conseguenza ci saranno parti di PRG che non avranno punti di appoggio nelle vicinanze. Per risolvere questo problema sono stati considerati i pezzi di PRG senza alcuna corrispondenza e si è cercata una loro corrispondenza direttamente nel DBT senza passare dalla CTR. In tal modo si trovano degli ulteriori punti di appoggio che permetteranno di ricostruire gli elementi del PRG.

### **Ricostruzione del piano regolatore**

Una volta determinati i nuovi punti di appoggio, che in pratica saranno delle linee, non rimane altro che modificare i confini delle aree del PRG basandosi su questi nuovi tratti. Questo comporta tutta una serie di problematiche che devono essere analizzate prima di andare a scrivere l'algoritmo. Va tenuto presente che possono esserci errori nella modifica delle geometrie, e casi complessi di modifica che vanno analizzati separatamente.

## 3.2 DESCRIZIONE DELL'ALGORITMO

Nei paragrafi che seguono verranno trattate nel dettaglio le varie fasi dell'algoritmo.

### 3.2.1 PRIMA PARTE: CALCOLO DELLE CORRISPONDENZE TRA PRG E CTR E TRA PRG E DBT

Prima di passare a costruire l'algoritmo è stato necessario analizzare i dati della CTR e del DBT e scegliere quali utilizzare nel programma. Si è scelto di prendere come riferimento gli elementi più importanti, in particolare i dati relativi alle strade, i dati relativi ai corsi d'acqua, gli elementi divisorii e gli edifici. Inoltre dato che in alcune zone della cartografia ci sono elementi diversi come possibili riferimenti, si è scelto di dare priorità diverse ai vari elementi. Hanno maggiore priorità gli elementi stradali, seguono i corsi d'acqua, gli elementi divisorii ed infine gli edifici. Questo vale sia per la CTR che per il DBT.

Inoltre dato che le linee di confine comuni a più elementi del PRG risultano doppie, per evitare problemi con le tolleranze sulle geometrie, è stato costruito tramite OpenJUMP il grafo planare degli elementi del PRG. Si utilizzano quindi le tabelle che costituiscono il grafo planare per tutte le operazioni sul PRG.

La prima parte dell'algoritmo si può dividere in due fasi:

- calcolo delle corrispondenze tra PRG e CTR
- calcolo delle corrispondenze tra PRG e DBT

La prima fase permette di determinare i punti di appoggio sulla base dei quali è stato costruito il PRG. La seconda fase permette invece di determinare su quali punti dovrà appoggiarsi il nuovo PRG.

#### **Calcolo delle corrispondenze tra PRG e CTR**

In questa fase si tratta di determinare quali porzioni del boundary del PRG e della CTR sono tra loro corrispondenti.

Prima di fare questo vengono estratti i boundary del PRG e della CTR e vengono suddivisi a loro volta in segmenti. Fatto questo per ogni segmento del PRG si va a determinare a quali segmenti della CTR si appoggia, tenendo presente che hanno maggiore priorità le strade, poi i fiumi, gli elementi divisorii e gli edifici. Si determinano, quindi, quali porzioni di PRG sono ad una certa distanza dai segmenti della CTR; la distanza varia in base al tipo di elemento della CTR che si va a considerare. Per fare questo si costruiscono dei buffer sui segmenti del PRG e si fa l'intersezione con i segmenti della CTR. La dimensione dei buffer dipende dal tipo di elemento della CTR che si considera; per strade, fiumi e elementi divisorii si è scelto di prendere una distanza di due metri, per gli edifici una distanza di un metro in quanto questi elementi hanno una precisione maggiore. Inoltre la lunghezza delle proiezioni ottenute in questa fase non deve essere superiore alla dimensione del buffer considerato, in quanto proiezioni troppo piccole potrebbero creare problemi nelle fasi successive.

Trovati gli elementi della CTR in corrispondenza del PRG è si proiettano queste corrispondenze sul relativo segmento di PRG a cui riferiscono. Questi nuovi segmenti vengono salvati in una tabella con le informazioni relative al segmento di PRG e di CTR ai quali fanno riferimento.

### **Calcolo delle corrispondenze tra PRG e DBT**

Questa seconda fase si occupa di controllare se le proiezioni determinate nella fase precedente hanno una corrispondenza nel DBT. Anche in questo caso conviene lavorare per segmenti.

Per ogni proiezione si controlla se esiste una porzione del DBT dello stesso tipo (cioè viabilità con aree stradali, idrografia con argini, ecc.) nelle vicinanze, per determinare come si è spostato l'elemento di appoggio trovato precedentemente. Per fare questo si utilizzano i buffer come nella fase precedente costruiti sulle proiezioni e si intersecano con gli elementi del DBT. Infine si proietta la proiezione determinata al punto precedente sul relativo segmento del DBT. Gli elementi trovati non devono essere troppo inclinati rispetto al segmento di PRG al quale riferiscono e la loro lunghezza non deve essere superiore alla dimensione del buffer considerato, altrimenti questo potrebbe causare problemi nelle fasi

successive dell'algoritmo. Queste proiezioni poi vanno memorizzate tenendo traccia delle informazioni relative al segmento di PRG, di CTR e DBT al quale fanno riferimento.

Viene creata infine una tabella con tutte le proiezioni trovate in questa fase e ordinate sulla base della loro distanza della prima coordinata del segmento di PRG al quale riferiscono.

### 3.2.1.1 ANALISI DEI RISULTATI OTTENUTI

Nel paragrafo seguente verranno esposti i risultati ottenuti nella prima fase dell'algoritmo.

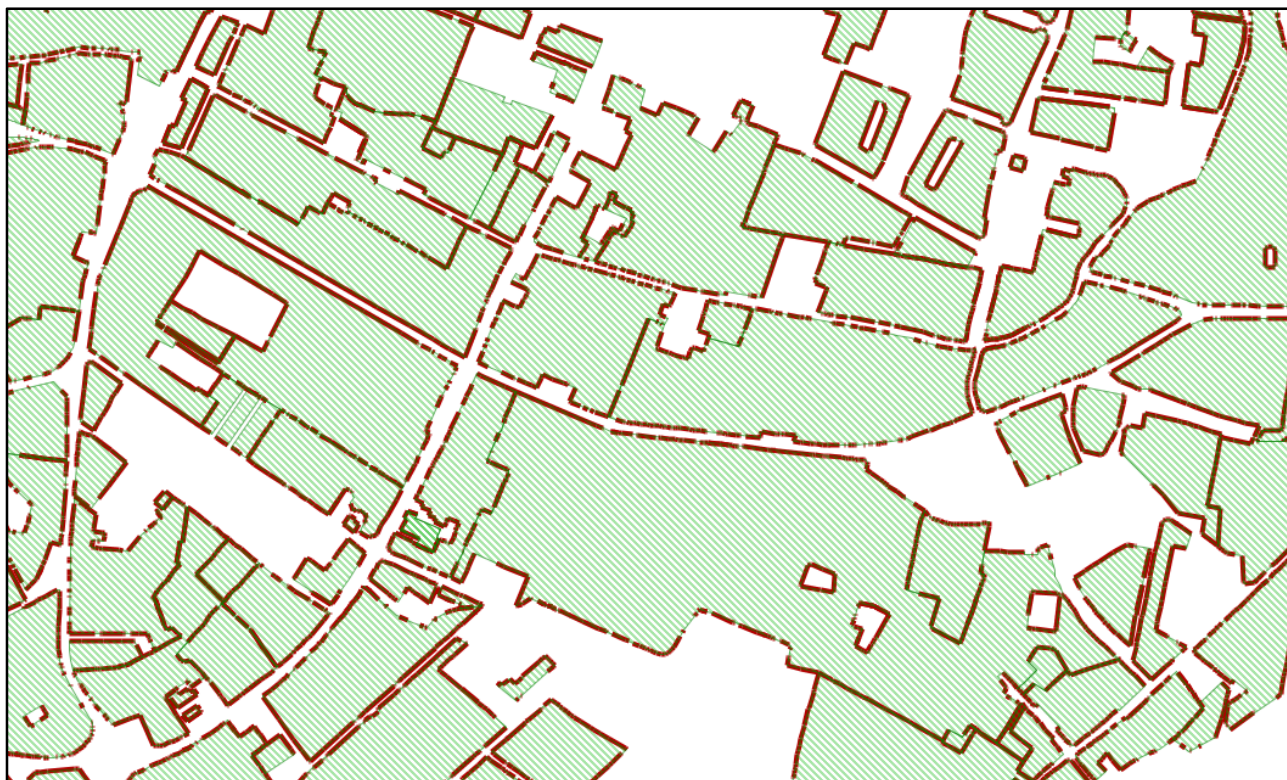
#### Calcolo delle corrispondenze tra PRG e CTR

Nella figura seguente è rappresentato un esempio dei risultati ottenuti in questa fase dell'algoritmo:



**Figura 3.1.** La figura rappresenta le proiezioni delle corrispondenze tra PRG e CTR sui segmenti del PRG. Le aree verdi sono le zone residenziali, le linee rosse sono le corrispondenze tra PRG e CTR

Come si può vedere dalla figura, alcune porzioni del perimetro del PRG non hanno una corrispondenza con elementi della CTR.



**Figura 3.2.** Particolare della figura precedente. Le aree verdi sono le zone residenziali, le linee rosse sono le corrispondenze tra PRG e CTR

### **Calcolo delle corrispondenze tra PRG e DBT**

Facendo riferimento alla porzione di cartografia rappresentata nella figura precedente, nella seconda fase dell'algoritmo si ottengono i seguenti risultati:

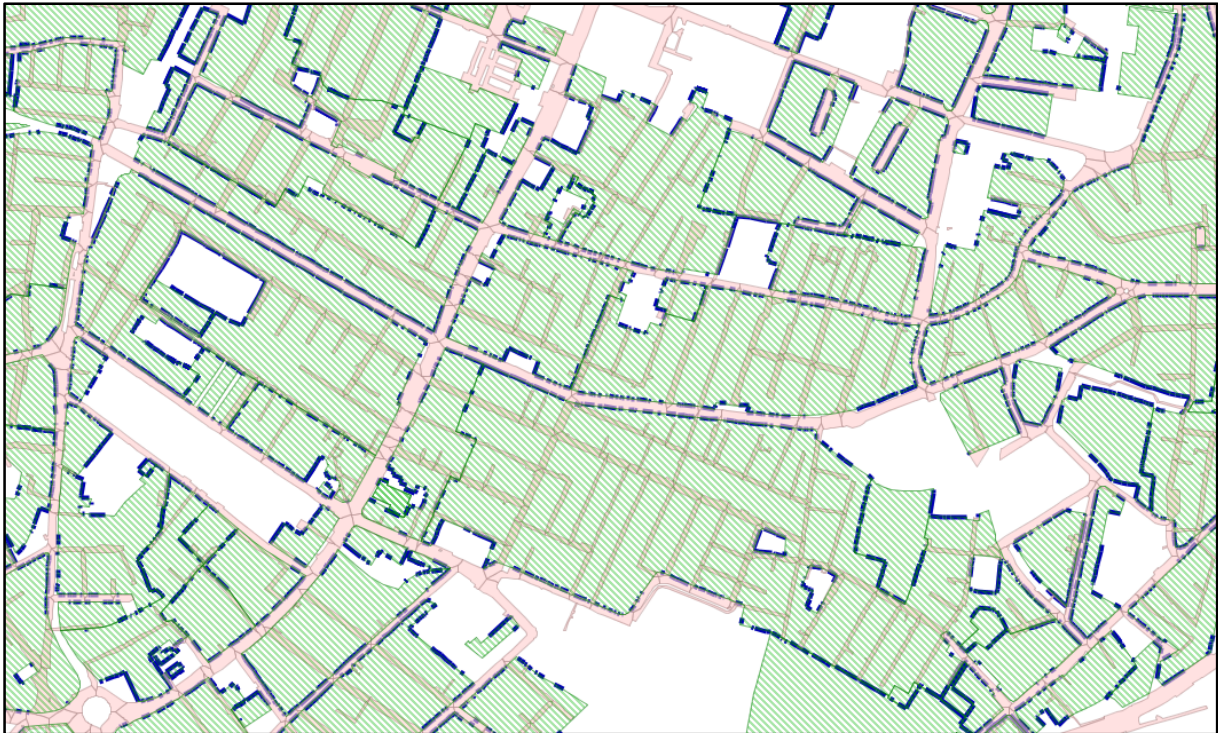


**Figura 3.3.** La figura rappresenta le proiezioni delle corrispondenze tra PRG e CTR sui segmenti del DBT. Le aree verdi sono le zone residenziali, le linee rosse sono le corrispondenze tra PRG e CTR, le linee blu sono le corrispondenze tra PRG e DBT

Come si può notare nella figura, ci sono elementi della CTR che non hanno una corrispondenza nel DBT, questi casi verranno considerate nella seconda fase dell'algorithm.



**Figura 3.4.** La figura rappresenta le proiezioni delle corrispondenze tra PRG e CTR sui segmenti del DBT. Le aree verdi sono le zone residenziali, le linee blu sono le corrispondenze tra PRG e DBT



**Figura 3.5.** La figura mostra il confronto delle corrispondenze tra PRG e DBT e le aree stradali. Le aree verdi sono le zone residenziali, le linee blu sono le corrispondenze tra PRG e DBT, le aree rosa rappresentano le aree stradali (AR\_STR)

Come si può notare dalle ultime figure, esistono parti di PRG che hanno corrispondenze nel DBT ma non sono state trovate in questa fase dell'algoritmo, questo è dovuto al fatto che non c'era corrispondenza nella CTR.

### 3.2.2 SECONDA PARTE: CALCOLO DELLE CORRISPONDENZE TRA PRG E DBT SENZA CORRISPONDENZA NELLA CTR

Dall'analisi dei risultati ottenuti dalla prima parte dell'algoritmo si è riscontrato che esistono molte porzioni di PRG che non hanno corrispondenza nella CTR ma hanno corrispondenza nel DBT e questi dati non vengono presi. Per evitare di perdere queste informazioni è necessario determinare una possibile corrispondenza degli elementi del DBT con le porzioni di PRG che non sono ancora state assegnate a nessun elemento del DBT. Per fare questo si determinano dapprima queste porzioni di PRG e si cerca nel DBT se ci sono degli elementi corrispondenti, tenendo sempre presente la loro priorità. Queste nuove proiezioni si ottengono costruendo dei buffer sulle porzioni di PRG senza corrispondenza e facendo l'intersezione con gli elementi del DBT. Infine si proiettano le porzioni di PRG sul relativo

segmento di DBT trovato. La dimensione dei buffer dipende da quali elementi del DBT si considerano, le dimensioni sono le stesse utilizzate nella prima parte dell'algoritmo. Anche per queste proiezioni, per quanto riguarda la loro dimensione e l'angolo con il PRG, vale quanto detto precedentemente.

Queste nuove proiezioni dati vengono memorizzate, aggiunte alle proiezioni ottenute nella fase precedente e riordinate.

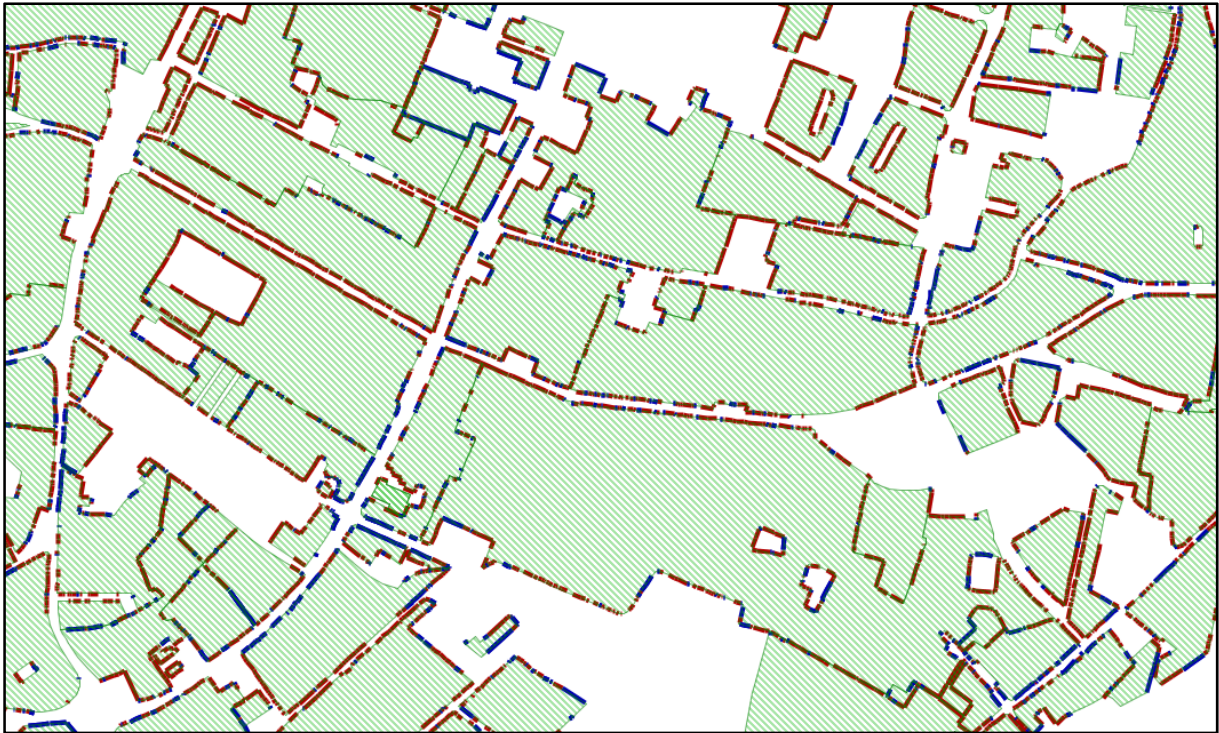
### 3.2.2.1 ANALISI DEI RISULTATI OTTENUTI

Sempre facendo riferimento alla porzione di cartografia precedente, nella seconda parte dell'algoritmo si ottengono i seguenti risultati:



**Figura 3.6.** La figura rappresenta le corrispondenze tra PRG e DBT senza corrispondenze nella CTR. Le aree verdi sono le zone residenziali, le linee blu sono le corrispondenze tra PRG e DBT senza corrispondenza nella CTR





**Figura 3.7.** La figura illustra quali sono le corrispondenze tra PRG e DBT ottenute nella prima e seconda parte dell'algoritmo. Le aree verdi sono le zone residenziali, le linee rosse sono le corrispondenze tra PRG e DBT determinate nella prima parte dell'algoritmo, le linee blu rappresentano le corrispondenze tra PRG e DBT senza corrispondenza nella CTR

Come si può notare, dopo questa fase molte porzioni di PRG sono state coperte da elementi del DBT, sulla base di questi dati si andrà in seguito a ricostruire il nuovo PRG.

### 3.2.3 TERZA PARTE: COSTRUZIONE DEL NUOVO PRG

Dalle fasi precedenti sono stati ottenuti i punti sui quali dovrà appoggiarsi il nuovo PRG. Dato che alcune parti di PRG non sono coperte da proiezioni, prima di andare a costruire il nuovo PRG, è necessario effettuare delle ulteriori operazioni.

Come prima cosa si congiungono tramite segmenti tutte le proiezioni consecutive che riferiscono ad uno stesso segmento di PRG. Questo permette di coprire alcune porzioni di PRG che non hanno corrispondenze nel DBT.

Come passo successivo si vanno ad analizzare gli incroci a T, cioè tutti quei casi dove si intersecano più di due segmenti di PRG. Se in queste situazioni si devono intersecare più di due proiezioni allora si determina un punto comune di intersezione e si congiungono le

proiezioni in quel punto. Per fare questo si determinano le due proiezioni che formano tra loro l'angolo più piccolo e si prolungano fino ad intersecarsi. Se il punto di intersezione è in prossimità del nodo di intersezione dei segmenti del PRG, allora si modificano tutte le proiezioni in modo che si intersechino in quel punto.

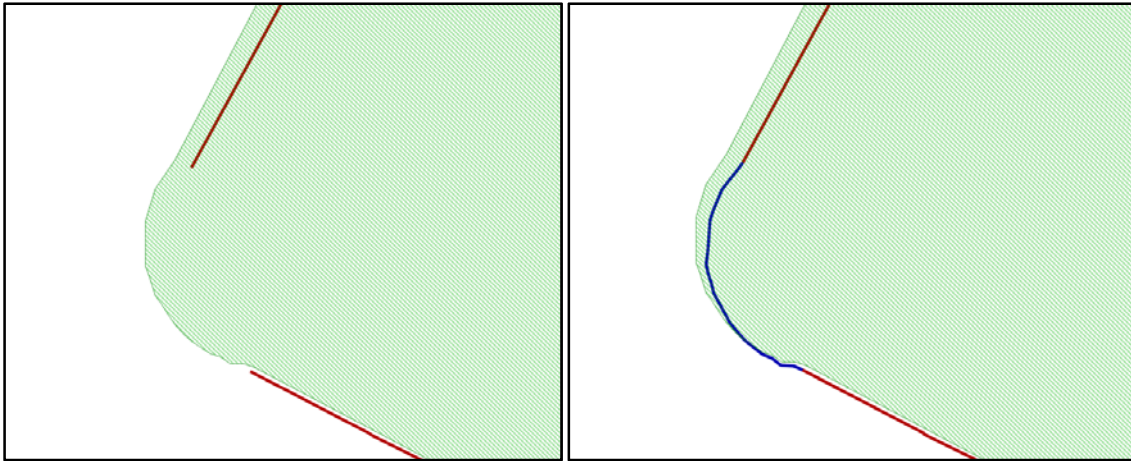
Dopo aver effettuato questa operazione molte zone di PRG sono già coperte, per le zone rimanenti occorre effettuare un'operazione più complessa.

Si calcola come prima cosa la percentuale di copertura di ogni poligono del PRG, cioè per ogni poligono si determina la lunghezza di perimetro coperta dalle proiezioni ottenute fin'ora. Se un elemento del PRG ha una percentuale di copertura superiore ad una certa soglia allora si può passare all'operazione successiva, altrimenti necessità di analisi più approfondite.

Una volta stabilito che i poligoni considerati possono essere analizzati, si estraggono tutte le proiezioni che riferiscono a quel poligono. Si passa poi a congiungere le proiezioni successive che riferiscono a segmenti di PRG diversi e che hanno una distanza inferiore ad una certa soglia. Per fare questo vanno considerate diverse casistiche:

- se l'angolo che formano è maggiore di  $175^\circ$ , le due proiezioni sono quasi collineari quindi basta congiungerle tramite un segmento;
- se l'angolo è inferiore, si prolungano le due proiezioni finché non si intersecano. La loro intersezione non deve essere troppo lontana dagli estremi che si vogliono congiungere. Se questo non si verifica, questi casi vengono analizzati nell'operazione successiva.

Rimane ora da analizzare i casi in cui la distanza tra le due proiezioni è superiore ad una certa soglia, e i casi in cui due proiezioni formano un angolo inferiore a  $175^\circ$  ma non si possono congiungere in un punto comune. In tutti questi casi si è deciso di utilizzare le trasformazioni affini. In particolare, tra due proiezioni consecutive che non si toccano, si determina la porzione di PRG non coperta e su questa geometria si fa una trasformazione affine agganciando la nuova geometria agli estremi delle due proiezioni considerate.

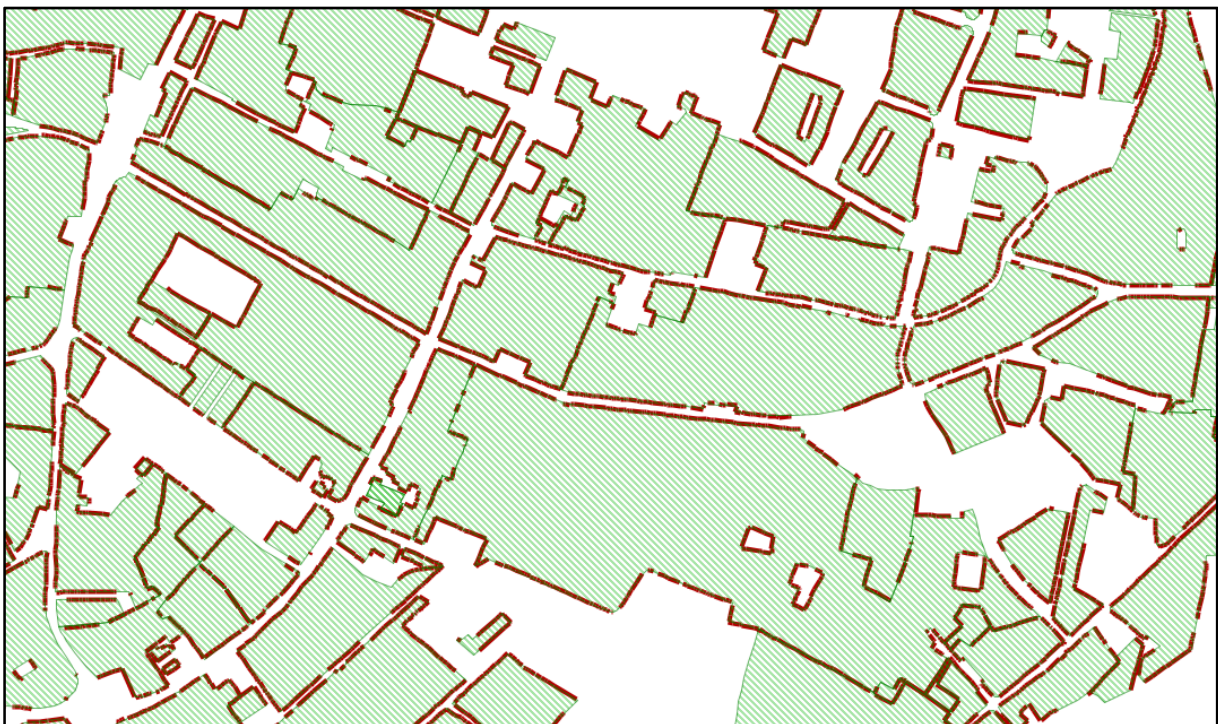


**Figura 3.8.** Esempio di trasformazione affine

Una volta fatta questa operazione per tutte le parti di PRG non coperte rimane da ricostruire i poligoni del PRG sulla base delle nuove geometrie. Questo si può fare unendo le proiezioni e le trasformazioni consecutive di uno stesso poligono di PRG ottenendo il risultato finale.

### 3.2.3.1 ANALISI DEI RISULTATI OTTENUTI

Dopo aver unito le proiezioni che riferiscono ad uno stesso segmento di PRG si ottengono i seguenti risultati:



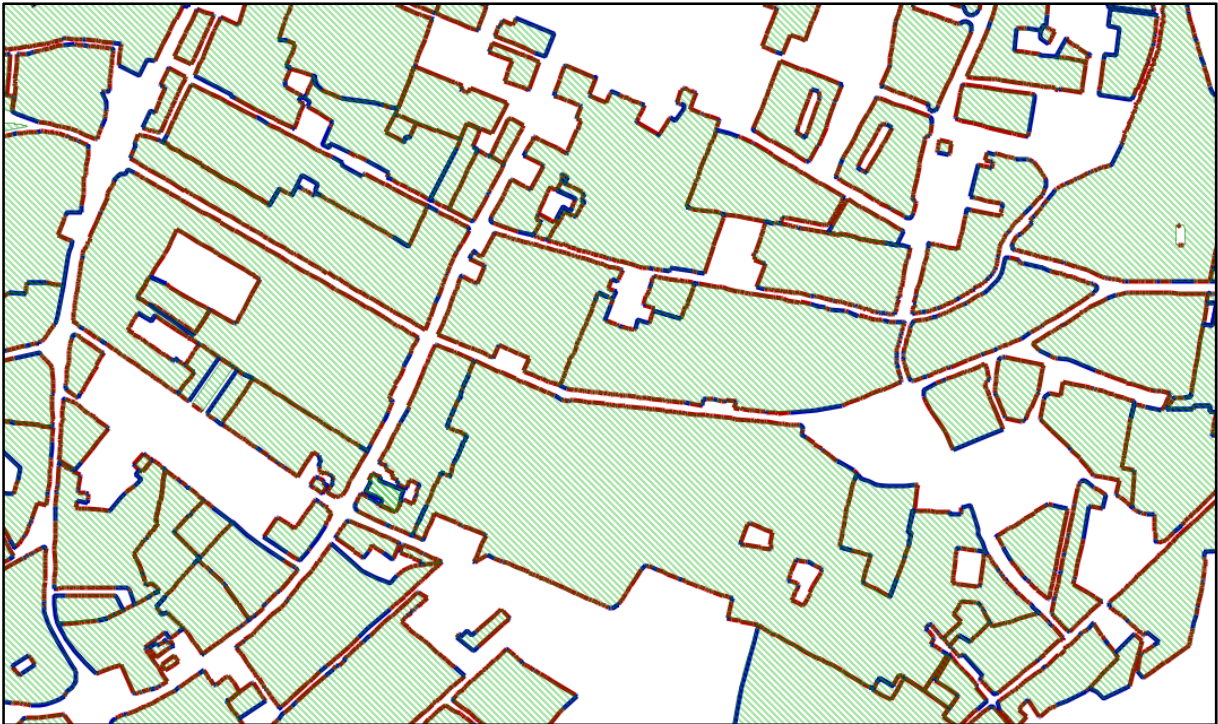
**Figura 3.9.** La figura descrive quali sono le corrispondenze tra PRG e DBT appartenenti ad uno stesso segmento di PRG unite tramite segmenti. Le aree verdi sono le zone residenziali, le linee rosse sono le corrispondenze tra PRG e DBT trovate nella prima e seconda parte dell'algoritmo

Dopo aver unito le proiezioni che riferiscono ad uno stesso segmento di PRG, vengono risolti i casi in cui si devono congiungere più proiezioni in un incrocio a T. Vengono calcolate poi le trasformazioni affini sulle porzioni di PRG per le quali non è stata trovata alcuna corrispondenza.



**Figura 3.10.** La figura rappresenta il calcolo delle trasformazioni affini. Le aree verdi sono le zone residenziali, le linee blu rappresentano le trasformazioni affini

Il risultato finale è il seguente:



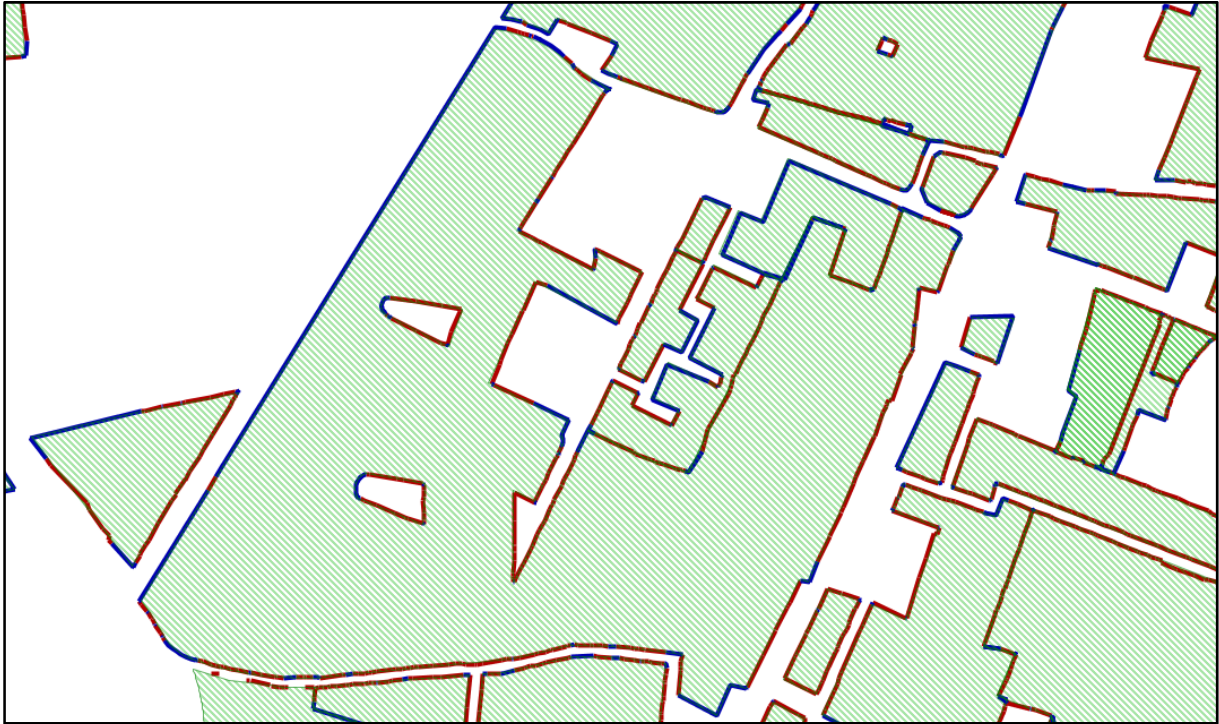
**Figura 3.11.** La figura illustra le corrispondenze tra PRG e DBT e il risultato delle trasformazioni affini. Le aree verdi sono le zone residenziali, le linee rosse sono le corrispondenze tra PRG e DBT trovate nella prima e seconda parte dell’algoritmo, le linee blu rappresentano le trasformazioni affini

Una volta calcolate le trasformazioni affini non rimane altro che costruire il nuovo PRG sulla base dei risultati ottenuti.

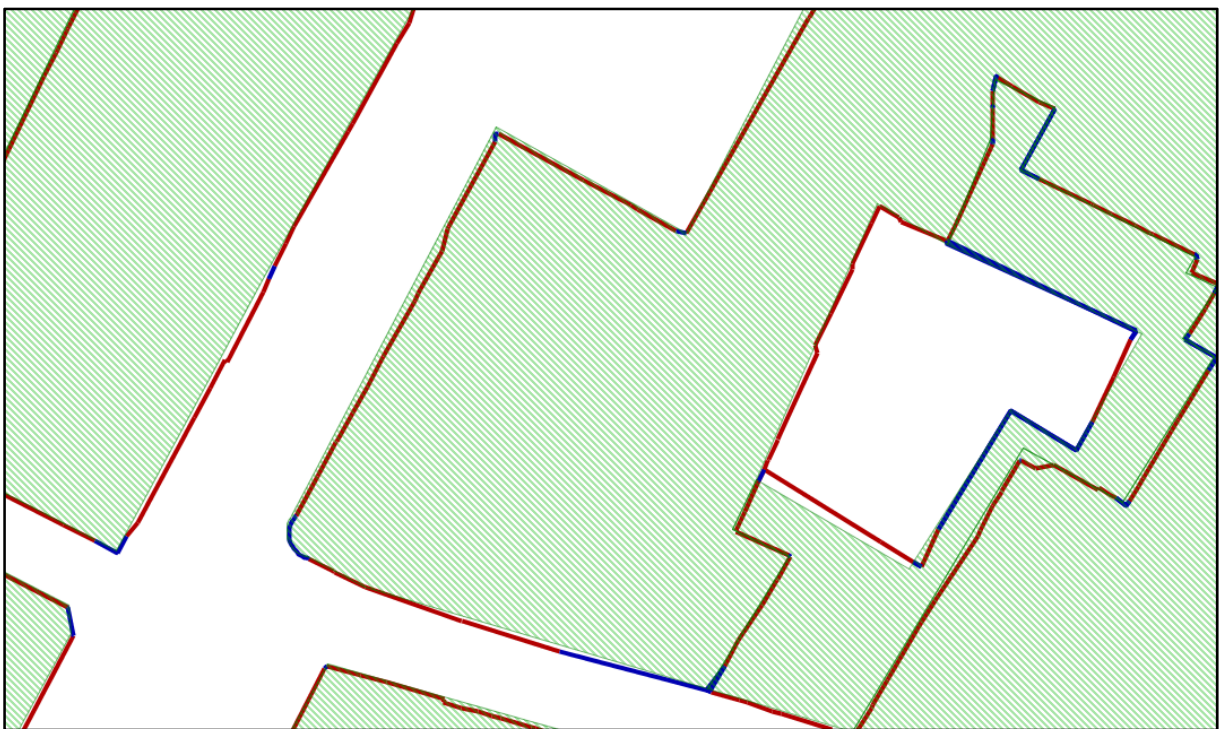


**Figura 3.12.** La figura mostra come risultano i confini del nuovo PRG dopo aver congiunto proiezioni e trasformazioni. Le aree verdi sono le zone residenziali, le linee verdi rappresentano i confini del nuovo PRG

Nelle figure seguenti vengono illustrate delle porzioni di cartografia ingrandite relative ai risultati ottenuti in questa parte dell'algoritmo.

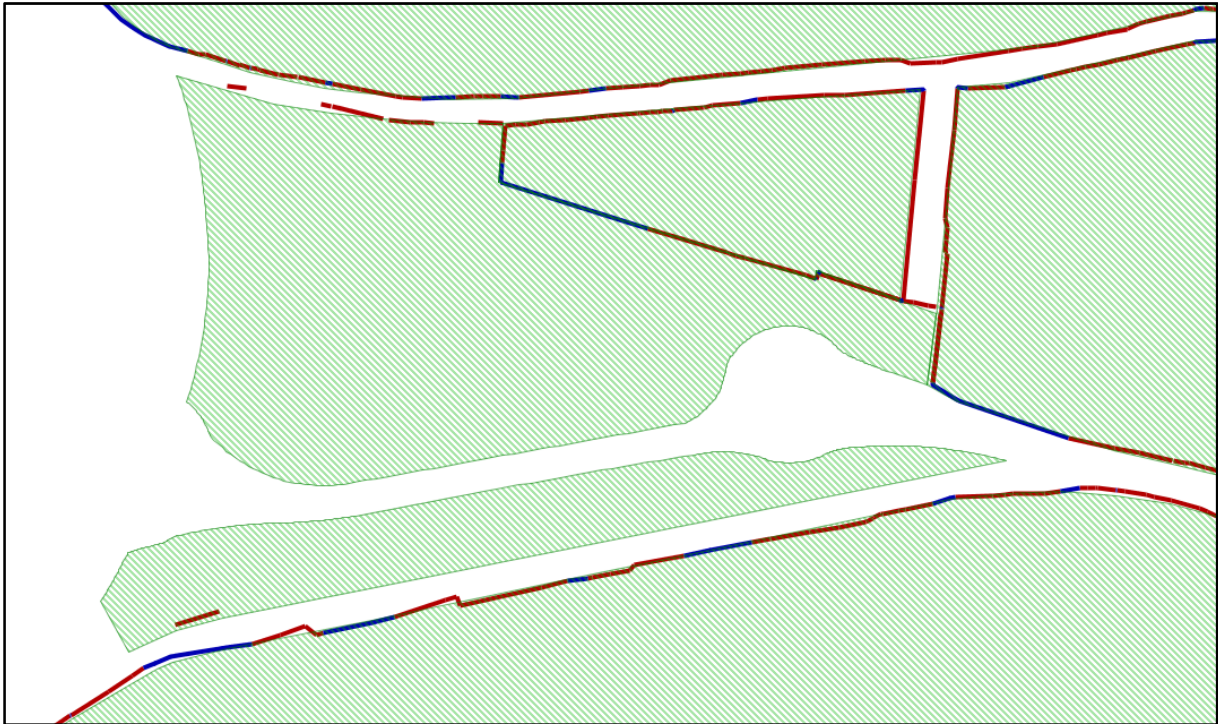


**Figura 3.13.** La figura illustra un particolare dei dati risultati dalle elaborazioni. Le aree verdi sono le zone residenziali, le linee rosse sono le corrispondenze tra PRG e DBT trovate nella prima e seconda parte dell'algoritmo, le linee blu rappresentano le trasformazioni affini



**Figura 3.14.** La figura illustra un particolare dei dati risultati dalle elaborazioni. Le aree verdi sono le zone residenziali, le linee rosse sono le corrispondenze tra PRG e DBT trovate nella prima e seconda parte dell'algoritmo, le linee blu rappresentano le trasformazioni affini

Come si può vedere il nuovo PRG risulterà scostato rispetto al vecchio PRG.



**Figura 3.15.** La figura rappresenta un caso particolare in cui la percentuale di occupazione è inferiore ad una certa soglia. Le aree verdi sono le zone residenziali, le linee rosse sono le corrispondenze tra PRG e DBT trovate nella prima e seconda parte dell'algoritmo, le linee blu rappresentano le trasformazioni affini

Come si può vedere esistono elementi del PRG per i quali non sono state trovate corrispondenze, di conseguenza queste zone non è possibile ricostruirle sulla base dei dati ottenuti nelle prime due parti dell'algoritmo.

Inoltre, nella figura 3.15 è possibile notare che, talvolta, le linee che compongono il nuovo PRG risultano seghettate. Questo è dovuto al fatto che vengono congiunti segmenti appartenenti ad elementi di DBT diversi.

### **3.3 CONSIDERAZIONI GENERALI**

Nel seguente paragrafo vengono analizzate alcune considerazioni di carattere generale determinate durante la fase di risoluzione del problema.

Innanzitutto vanno presi in esame i problemi riguardanti i dati utilizzati. Vi è infatti la possibilità di avere dei dati parzialmente corretti, cioè è possibile che i dati considerati presentino degli errori in alcune geometrie; errori relativi alla trascrizione delle informazioni e conseguente costruzione delle mappe, o relativi alla fase di misurazione e rilevamento degli elementi facenti parte del territorio. Questi errori sono difficili da individuare quindi si prendono per vere le informazioni di queste mappe tenendo presente la possibilità di avere questo tipo di errori.

Un altro errore per quanto concerne i dati riguarda la loro struttura, ed in particolare il fatto che in alcuni casi non c'è corrispondenza tra gli elementi della CTR e quelli del DBT. Per questo motivo è stato ritenuto necessario, nella seconda parte dell'algoritmo, determinare le corrispondenze tra il PRG e il DBT senza passare dalla CTR.

Un problema importante riguarda le tolleranze, può capitare che due linee apparentemente identiche risultino leggermente scostate. Questa problematica va tenuta presente ogni volta che si vanno ad effettuare delle operazioni sulle geometrie, per questo motivo si è scelto di utilizzare un grafo planare per evitare problemi nella prima e seconda parte dell'algoritmo.

Infine, dato che per alcune zone di PRG non sono state trovate corrispondenze rilevanti per la loro ricostruzione, per queste zone non è possibile fare nulla. Sarà necessaria un'ulteriore analisi più approfondita per cercare di risolvere il problema.



## Capitolo Quarto

# POSSIBILI EVOLUZIONI

---

Nel capitolo seguente verranno analizzate quali potrebbero essere le possibili migliorie da aggiungere all'algoritmo per renderlo più completo.

Come prima cosa c'è da notare la struttura dei dati utilizzati nella fase di test. Dato che, talvolta, le geometrie considerate presentano problemi di correttezza formale, potrebbe essere utile testare l'algoritmo con dati di migliore qualità al fine di ottenere risultati migliori.

Un altro aspetto riguarda le migliorie da implementare per migliorare il risultato finale. Potrebbe essere necessario, ad esempio, raddrizzare le linee di confine del nuovo PRG che risultano troppo seghettate; oppure migliorare i risultati ottenuti nel caso che parti di elementi del nuovo PRG si incrocino; e via di seguito.

Tutti questi miglioramenti permettono di ottenere un risultato finale più accurato rispetto ai riferimenti reali presenti sul territorio.

Infine, dato che non è possibile ricostruire gli elementi del vecchio PRG che hanno poche corrispondenze nella nuova cartografia, sarebbe necessario effettuare un'analisi più approfondita di questi casi e cercare di ricostruire anche quegli elementi del PRG che con la soluzione qui proposta non è possibile considerare. Una possibile soluzione potrebbe prevedere l'eventualità di utilizzare altri elementi della CTR e del DBT per ottenere maggiori corrispondenze, oppure stabilire delle regole specifiche per la loro ricostruzione.

Tutti gli aspetti discussi fin'ora permetterebbero all'algoritmo di gestire gran parte delle situazioni che in questa tesi non sono state affrontate. Va tenuto presente che in alcuni casi non è possibile risolvere il problema in modo automatico e non rimane altro che ricostruire questi elementi del PRG in modo manuale.

## Capitolo Quinto

# CONSIDERAZIONI FINALI

---

Questo lavoro di tesi ha permesso di affrontare la tematica del trattamento dell'informazione geografica che presenta elementi di elevato interesse..

Inoltre è stato possibile confrontarsi con problemi riguardanti lo sviluppo di software ed in particolare con le specifiche problematiche riguardanti i dati geografici.

Nonostante il fatto che la soluzione proposta possa essere migliorata, il risultato finale ottenuto dalle elaborazioni rappresenta un passo significativo verso una soluzione al problema posto.

## BIBLIOGRAFIA E RIFERIMENTI UTILI

- [1] <http://cargen.dei.unipd.it/index-2.html>
- [2] [http://it.wikipedia.org/wiki/Eclipse\\_\(informatica\)](http://it.wikipedia.org/wiki/Eclipse_(informatica))
- [3] <http://it.wikipedia.org/wiki/PostgreSQL>
- [4] <http://www.regione.veneto.it/Ambiente+e+Territorio/Territorio/Sistema+Informativo+Territoriale+e+Cartografia/Infrastruttura+Dati+Territoriali/Informazione+sui+prodotti/CartaTecnicaRegionale/>
- [5] [http://www.regione.veneto.it/NR/rdonlyres/EFE447B1-764F-49F3-AB3D-4F91FC6F5925/0/Carta\\_Tecnica\\_Regionale.pdf](http://www.regione.veneto.it/NR/rdonlyres/EFE447B1-764F-49F3-AB3D-4F91FC6F5925/0/Carta_Tecnica_Regionale.pdf)
- [6] <http://www.padovanet.it/dettaglio.jsp?id=2932>

## Appendice A

# TECNOLOGIE UTILIZZATE

---

Nel seguente paragrafo verranno citate quali sono le tecnologie utilizzate per realizzare l'algoritmo, mettendo in evidenza gli aspetti più importanti di ognuna di esse.

### A1. POSTGRESQL

PostgreSQL è un database relazionale ad oggetti rilasciato con licenza libera. PostgreSQL permette di memorizzare i dati come una serie di tabelle e conservare tutte le caratteristiche dei dati presenti nella base di dati (chiavi primarie, vincoli, chiavi esterne, ecc.). PostgreSQL utilizza il linguaggio SQL per eseguire delle query sui dati.

La programmabilità di PostgreSQL è il suo principale punto di forza, PostgreSQL rende più semplice costruire applicazioni per il mondo reale, utilizzando i dati prelevati dal database.

Lo svantaggio principale riguarda la conversione delle informazioni dal mondo SQL a quello di programmazione orientata agli oggetti; questa presenta difficoltà dovute principalmente al fatto che i due mondi utilizzano differenti modelli di organizzazione dei dati: cartografare i dati da un modello ad un altro può assorbire fino al 40% del tempo di sviluppo di un progetto.

PostgreSQL può risolvere molti di questi problemi direttamente nel database. PostgreSQL permette agli utenti di definire nuovi tipi basati sui normali tipi di dato SQL, permettendo al database stesso di comprendere dati complessi.

PostgreSQL, inoltre, permette l'ereditarietà dei tipi, uno dei principali concetti della programmazione orientata agli oggetti.

I principali vantaggi dell'utilizzo di PostgreSQL nella programmazione sono i seguenti:

- Incremento delle prestazioni, in quanto la logica viene applicata direttamente al server di database in una volta, riducendo il passaggio di informazioni tra client ed il server.
- Incremento dell'affidabilità, dovuto alla centralizzazione del codice di controllo sul server, non dovendo gestire la sincronizzazione della logica tra molteplici client e i dati memorizzati sul server.
- Inserendo livelli di astrazione dei dati direttamente sul server, il codice del client può essere più snello e semplice.

Questi vantaggi fanno di PostgreSQL, probabilmente, il più avanzato sistema database dal punto di vista della programmabilità. Utilizzare PostgreSQL può ridurre il tempo totale di programmazione di molti progetti, con i vantaggi suddetti che crescono con la complessità del progetto stesso, ovviamente se utilizzato da utenti esperti nel suo utilizzo.

Il problema principale di questo database è che PostgreSQL è relativamente poco utilizzato rispetto a concorrenti presenti sul mercato da maggiore tempo. Questo fa sì che i costi di sviluppo siano spesso superiori rispetto a database più conosciuti ed utilizzati. Con l'eventuale aumento della diffusione di PostgreSQL, questo problema potrebbe tendere ad affievolirsi.

PostgreSQL supporta alcuni tool grafici di gestione, tra cui:

- phpPgAdmin: è un'applicazione PHP libera che consente di amministrare in modo semplificato, con una interfaccia grafica basata sul web, database di PostgreSQL
- pgAdmin: è un'applicazione multiplatforma, scritta in C++, che consente di amministrare in modo semplificato, con una interfaccia grafica, database di PostgreSQL.

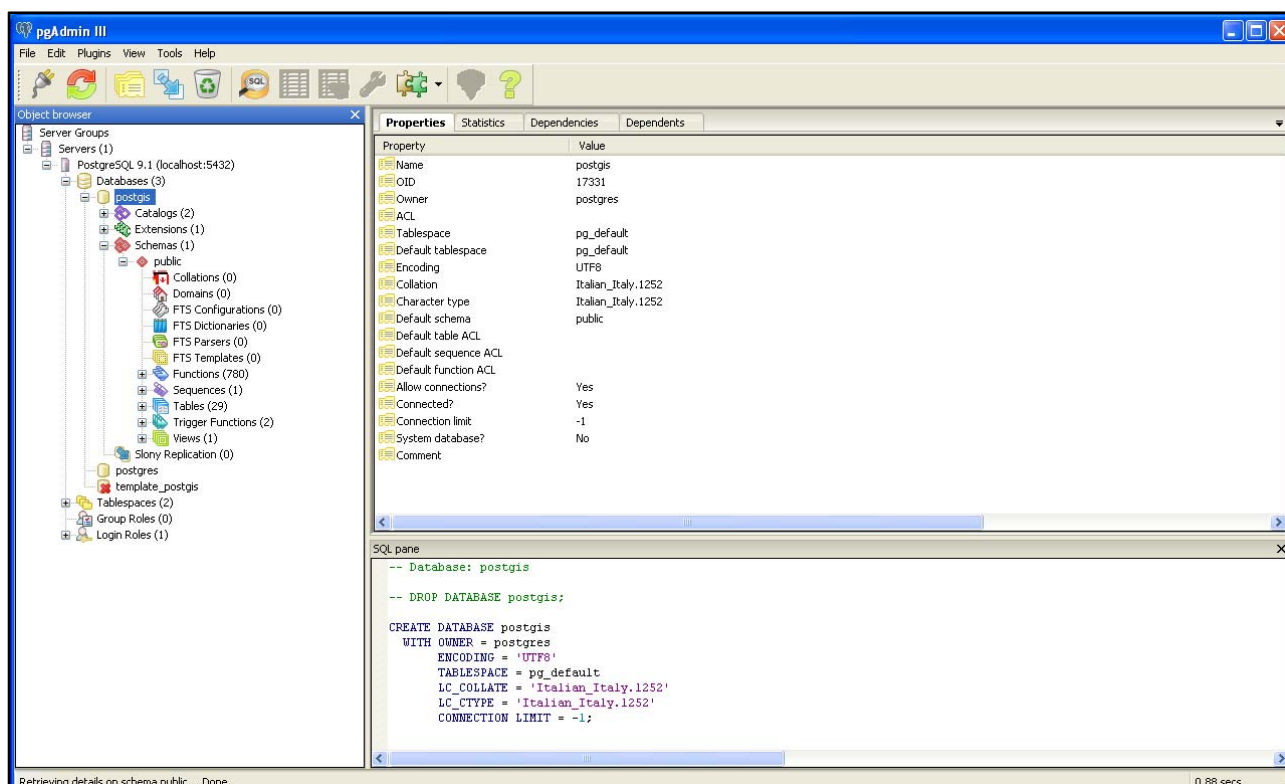


Figura 1. Schermata di PgAdmin

## A2. POSTGIS

PostGIS è l'estensione spaziale del server PostgreSQL che introduce i tipi di dato geometrico e le funzioni per lavorare con essi. Inoltre fornisce le definizioni dei sistemi di coordinate per eseguire trasformazioni tra geometrie presenti in sistemi di riferimento diversi.

PostGIS permette di gestire le seguenti tipologie di dato: POINT, MULTIPOINT, LINESTRING, MULTILINESTRING, POLYGON, MULTIPOLYGON, GEOMETRYCOLLECTIONS (più le estensioni SQL-MM CIRCULARSTRING, COMPOUNDCURVE, CURVEPOLYGON, MULTICURVE, MULTISURFACE) con estensione XYZ,XYM, XYZM.

I database gestiti da PostGIS si differenziano per la maggior robustezza e per la presenza degli indici spaziali. Infatti le elaborazioni GIS avvengono usando la sintassi SQL sui costrutti spaziali. Esistono comunque funzioni che costruiscono immediatamente tutta la struttura necessaria alla gestione dei dati territoriali. Analogamente a PostgreSQL, le

interrogazioni al database avvengono utilizzando “SQL query” utilizzando combinazioni di dati e funzioni e di test vero/falso. Per comprendere come funzioni una query spaziale occorre tenere presente due cose:

- esistono gli indici spaziali;
- le interrogazioni spaziali sono molto dispendiose in termini di calcolo se effettuate su un gran numero di entità geometriche.

Gli indici spaziali esistono per migliorare l'efficienza delle query. I dati geografici vengono "aggregati" e "amministrati" in gruppi spaziali distinti. In questo senso gli indici sono definiti per semplificare e nella semplificazione perdono informazioni. Ad esempio: spesso si vuole utilizzare l'operatore intersezione && che testa se il rettangolo che circonda le entità geometriche ne intersechi altre. Questa funzione è ottimizzata per l'utilizzo degli indici spaziali: serve a scremare i dati per eseguire una ricerca più raffinata usando le funzioni Distance, Intersects, Contains, Within, ecc. Le funzioni di PostGIS permettono quindi di ottenere solamente i dati necessari per lo sviluppo dei programmi, in modo da ridurre la complessità computazionale dei programmi.

### **A3. OPENJUMP**

OpenJUMP è un GIS (Geographic Information System) basato sulla piattaforma Java.

Il vantaggio di OpenJUMP è la sua capacità di edizione e di disegno: gli elementi possono essere scalati, ruotati, spostati, etc. La sezione strumenti contiene una lista completa di strumenti di edizione e di analisi spaziale. OpenJUMP può aprire file vettoriali, raster e database (PostGIS, Oracle) in modo da poter visualizzare qualunque tipo di dato.

OpenJUMP, essendo basato su piattaforma Java, è utile per visualizzare dati territoriali a seguito dell'elaborazione di un programma scritto in Java, che utilizza questo tipo di dato.

Nella figura seguente viene rappresentata la schermata principale di OpenJUMP e un esempio di visualizzazione dei dati tramite file shape.

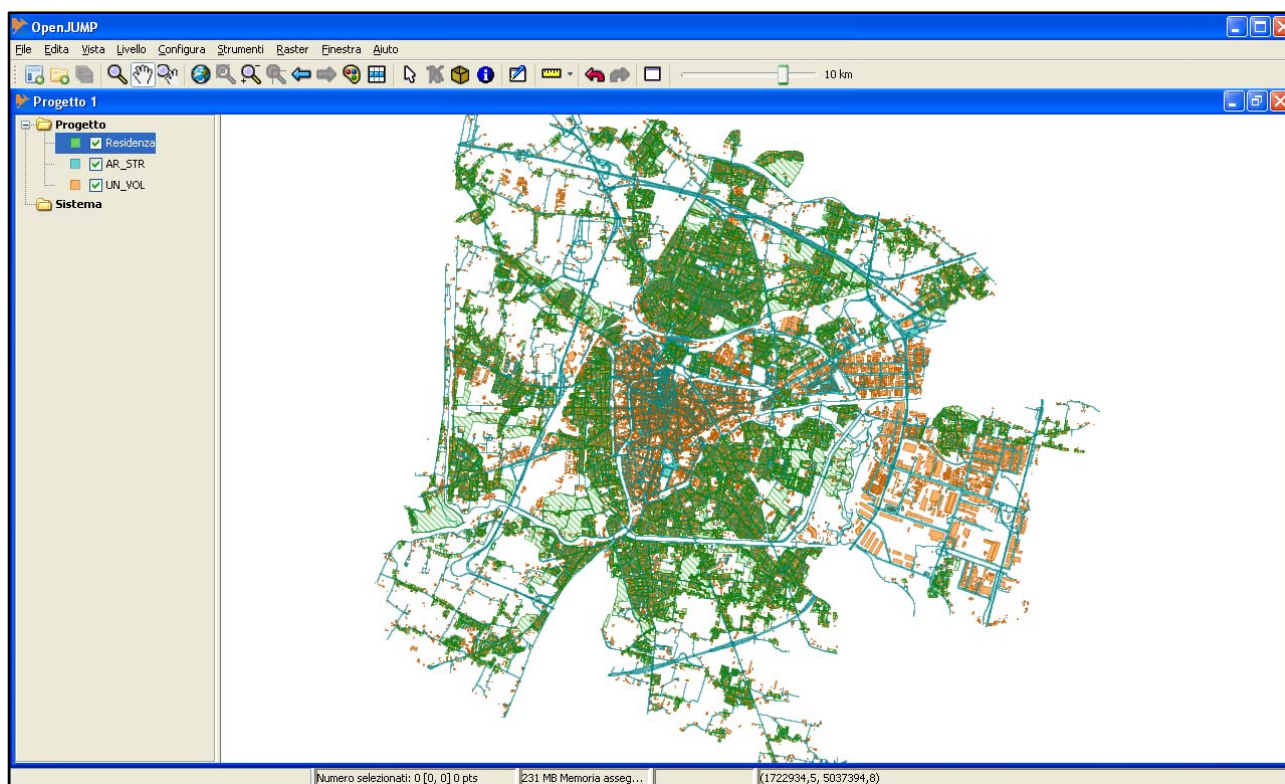


Figura 2. Schermata di OpenJump

## A4. ECLIPSE

Eclipse è un ambiente di sviluppo integrato multi-linguaggio e multiplatforma. Eclipse può essere utilizzato per la produzione di software di vario genere, tra cui: Java, C++, XML, JavaScript, PHP.

Il programma è scritto in linguaggio Java, ha una propria VM (Virtual Machine) e delle librerie che permettono la gestione di programmi Java.

La piattaforma di sviluppo è incentrata sull'uso di plug-in, delle componenti software ideate per uno specifico scopo. Nella versione base è possibile programmare in Java, usufruendo di comode funzioni di aiuto quali: completamento automatico, suggerimento dei tipi di parametri dei metodi, possibilità di accesso diretto a CVS e riscrittura automatica del codice in caso di cambiamenti nelle classi.

Eclipse permette di scrivere in modo semplice i propri programmi Java anche tramite l'utilizzo di queste funzionalità.

Nella figura che segue viene mostrata la schermata di Eclipse con un esempio di codice.



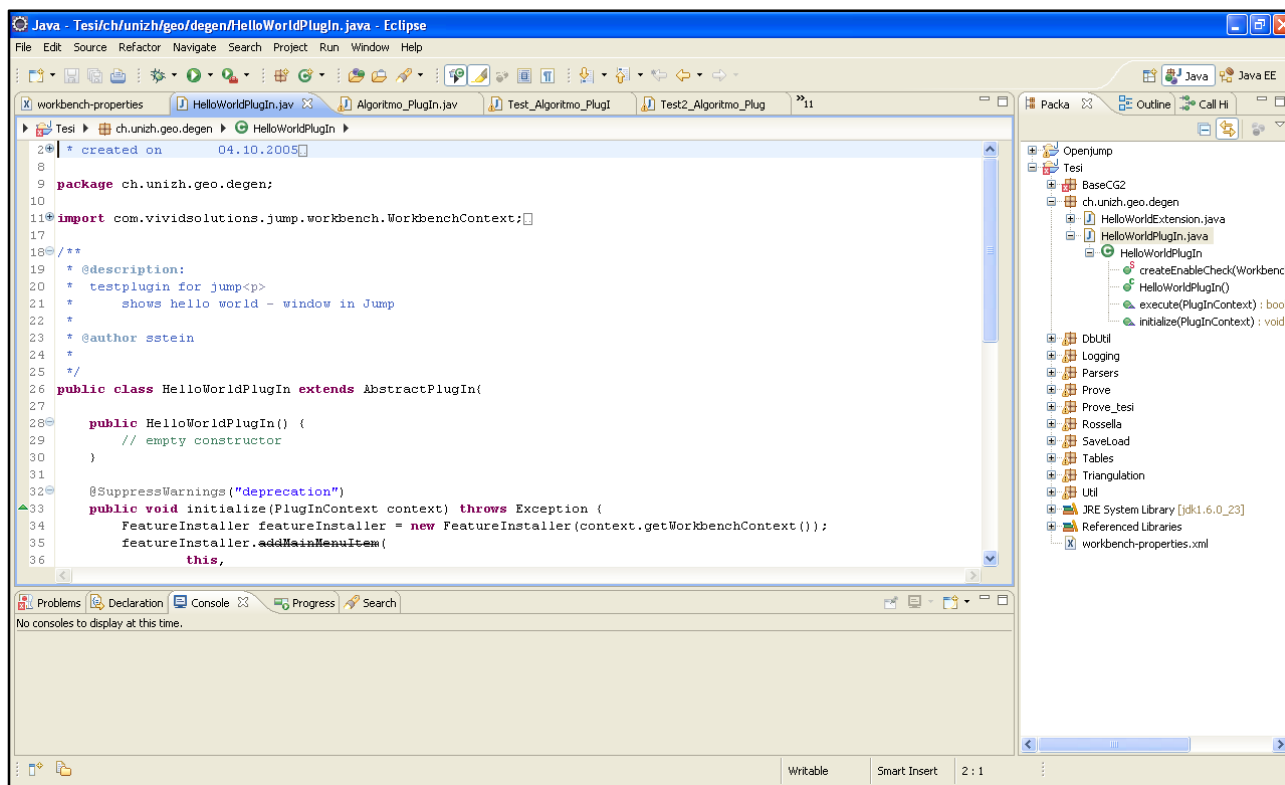


Figura 3. Esempio di schermata di Eclipse

## A5. LE CLASSI DEL PROGETTO CARGEN

Il progetto CARGEN (CARTografia GENERALizzata) è un progetto sviluppato dalla Regione del Veneto e l'Università degli Studi di Padova, d'intesa con l'Istituto Geografico Militare. Questo progetto ha come scopo la generalizzazione del Database alla scala 1:25000, derivato dal Database alla scala 1:5000 prodotto dalla Regione del Veneto; è stato, inoltre, articolato in diverse attività di ricerca e sperimentazione, finalizzate alla realizzazione dei processi di generalizzazione semantica e cartografica.

Lo sviluppo di questo progetto ha permesso di ottenere delle classi Java, utilizzabili in altri progetti al fine di migliorare l'efficienza degli algoritmi. Queste classi contengono dei metodi che permettono la gestione di oggetti complessi (come gli oggetti Table), la gestione delle connessioni al database indipendentemente dal tipo di software utilizzato, la gestione delle query spaziali, ecc. L'utilizzo di questi metodi e di alcuni oggetti gestiti da queste classi, permette all'algoritmo realizzato di essere più veloce rispetto ad un programma che utilizzi delle semplici query SQL, e di ridurre la complessità computazionale limitando il numero di dati sul quale il programma va ad agire.

Le classi del progetto CARGEN sono le seguenti:

- DbUtil: questa cartella contiene le classi per la gestione del db: ad esempio DbUtility, DbQuery, ecc.
- Table: questa cartella contiene le classi per la gestione degli oggetti Table: ad esempio TabLab, Table, TableConnection, ecc.
- Logging, Util, Parsers, Triangulation: questi package contengono altre classi utili per la realizzazione dei programmi.

## Appendice B

# REALIZZAZIONE TECNICA

---

Una volta analizzato accuratamente il problema e decise le tecnologie e gli strumenti da utilizzare si è passati alla realizzazione di un programma scritto in linguaggio Java che permetta di risolvere il problema considerato.

Il programma realizzato è stato costruito nel seguente modo:

- È stata creata una base di dati con le tabelle che contengono i dati principali da utilizzare all'interno del programma
- È stato scritto un PlugIn di esecuzione
- È stato scritto un package che implementa l'algoritmo

Per quanto riguarda la connessione alla base di dati, si dà la possibilità all'utente di scegliere se utilizzare PostGIS come DBMS oppure utilizzare una connessione virtuale a OpenJUMP.

Nel primo caso la base di dati contiene alcune tabelle predefinite che contengono i dati di input. L'algoritmo si collega direttamente alla base di dati per reperire le informazioni necessarie alle elaborazioni e memorizzare i risultati. Utilizzando PostGIS come DBMS le elaborazioni sono più lente, quindi conviene questa scelta se si utilizza una grande quantità di dati e se i dati di input rimangono invariati.

Se si utilizza una connessione virtuale a OpenJump si utilizza un PlugIn di esecuzione che ha lo scopo principale di caricare i dati di input nel database sotto forma di tabelle con i campi opportuni, richiamare i metodi principali dell'algoritmo, in particolare richiamerà i metodi per il calcolo delle corrispondenze tra PRG e CTR, i metodi per il calcolo delle corrispondenze tra PRG e DBT e i metodi per la costruzione del nuovo PRG. Infine questo PlugIn si occuperà di leggere dal database i risultati finali e stamparli su OpenJUMP.

Utilizzare questo tipo di connessione rende le elaborazioni più veloci ed è possibile caricare di volta in volta piccole porzioni di dati che permettono di verificare la correttezza dell'algoritmo anche in situazioni particolari. In PostGIS questo non è possibile in quanto caricare solo una porzione di dati implica la cancellazione dei dati precedenti e quindi si ha un dispendio di tempo a dover caricare di volta in volta i nuovi dati.

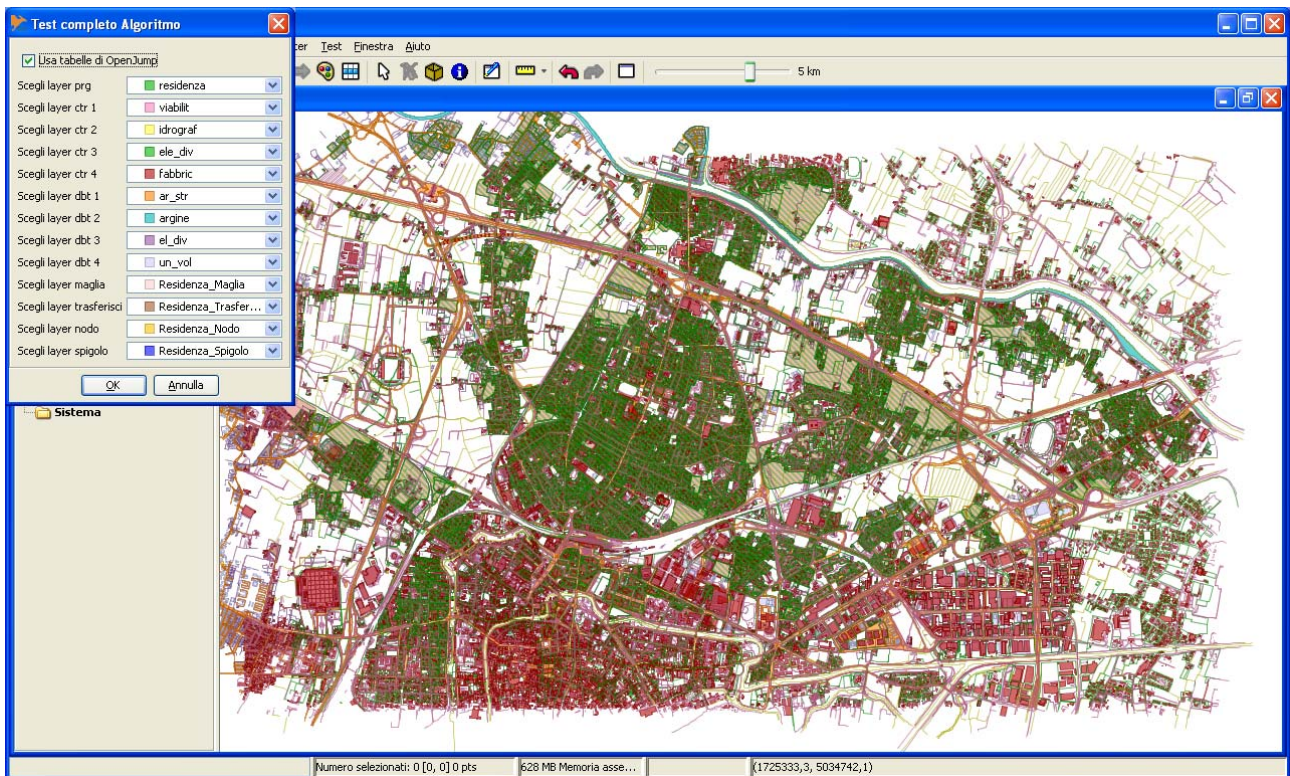


Figura 1. Esempio di schermata iniziale gestita dal PlugIn di esecuzione

## Appendice C

# DESCRIZIONE DELLA BASE DI DATI

---

Durante la realizzazione del programma è stato necessario tenere traccia delle informazioni ottenute durante le elaborazioni. A questo scopo è stata costruita una base di dati dove andare a memorizzare queste informazioni. La base di dati non possiede una struttura complessa, è composta da semplici tabelle, non collegate tra loro tramite chiavi esterne, dove vengono memorizzati i dati di input e le informazioni intermedie che servono ad ottenere il risultato finale.

Come primo passo è stato necessario memorizzare i dati di input. Per fare questo sono state create delle tabelle di input che contengono i dati relativi alla CTR, PRG e DBT. Inoltre dato che è possibile restringere il campo dei dati e applicare l'algoritmo su una porzione di essi, si è pensato di caricare nelle tre tabelle solo le informazioni relative a quel numero ristretto di dati che permettessero di sviluppare al meglio l'algoritmo e risolvere il problema nel suo complesso.

Le tabelle di input utilizzate per i test sono le seguenti:

- Residenza: contiene le zone di PRG corrispondenti alle aree residenziali;
- Viabilit: nella CTR rappresenta la viabilità;
- Idrograf: nella CTR rappresenta l'idrografia;
- Ele\_div: nella CTR rappresentano gli elementi divisorii;
- Fabbric: nella CTR rappresentano gli edifici;
- Ar\_str: nel DBT rappresentano le aree stradali (comprendono marciapiedi e aree a circolazione veicolare);
- Argine: nel DBT rappresentano gli argini dei fiumi e corsi d'acqua;
- El\_div: nel DBT rappresentano gli elementi divisorii;
- Un\_vol: nel DBT rappresentano gli edifici;

- **Residenza\_maglia**: rappresenta le maglie del grafo planare costruito sugli elementi del PRG considerato;
- **Residenza\_trasferisci**: rappresenta i poligoni del grafo planare costruito sugli elementi del PRG considerato;
- **Residenza\_nodo**: rappresenta i nodi degli elementi del grafo planare costruito sugli elementi del PRG considerato;
- **Residenza\_spigolo**: rappresenta gli spigoli (cioè i lati) del grafo planare costruito sugli elementi del PRG considerato.

Di seguito verranno prese in considerazione le tabelle intermedie utilizzate nelle fasi che compongono l'algoritmo.

### **Calcolo delle corrispondenze tra PRG e CTR e tra PRG e DBT**

La prima parte dell'algoritmo è caratterizzata dal calcolo delle corrispondenze tra gli elementi della CTR gli elementi del PRG, e tra gli elementi del PRG e gli elementi del DBT. In questa fase sono state create le seguenti tabelle:

- **segmenti\_prg(id, id\_prg, nome\_layer, geometry)**: questa tabella contiene i segmenti che costituiscono gli spigoli del grafo planare costruito sul PRG;
- **segmenti\_ctr(id, id\_ctr, nome\_layer, geometry)**: questa tabella contiene i segmenti che costituiscono i boundary degli elementi di una particolare tabella della CTR;
- **rimanenze(id, id\_prg, nome\_layer, geometry)**: contiene i pezzi dei segmenti di PRG da proiettare sui relativi segmenti di CTR;
- **proiezioni\_ctr\_prg\_viabilit(id, id\_segmento\_prg, id\_prg, layer\_prg, id\_segmento\_ctr, id\_ctr, layer\_ctr, geometry)**: rappresentano le proiezioni di alcune porzioni dei segmenti del PRG sui corrispondenti elementi della tabella viabilit del CTR;
- **proiezioni\_ctr\_prg\_idrograf(id, id\_segmento\_prg, id\_prg, layer\_prg, id\_segmento\_ctr, id\_ctr, layer\_ctr, geometry)**: rappresentano le proiezioni di alcune porzioni dei segmenti del PRG sui corrispondenti elementi della tabella idrograf del CTR;
- **proiezioni\_ctr\_prg\_ele\_div(id, id\_segmento\_prg, id\_prg, layer\_prg, id\_segmento\_ctr, id\_ctr, layer\_ctr, geometry)**: rappresentano le proiezioni di alcune porzioni dei segmenti del prg sui corrispondenti elementi della tabella ele\_div del CTR;

- proiezioni\_ctr\_prg\_fabbric(id, id\_segmento\_prg, id\_prg, layer\_prg, id\_segmento\_ctr, id\_ctr, layer\_ctr, geometry): rappresentano le proiezioni di alcune porzioni dei segmenti del prg sui corrispondenti elementi della tabella fabbric del CTR;
- segmenti\_dbt(id, id\_dbt, nome\_layer, geometry): questa tabella contiene i segmenti che costituiscono i boundary degli elementi di una particolare tabella del DBT;
- proiezioni\_prg\_dbt\_ar\_str(id, id\_segmento\_prg, id\_prg, layer\_prg, id\_segmento\_ctr, id\_ctr, layer\_ctr, id\_segmento\_dbt, id\_dbt, layer\_dbt, geometry): rappresentano le proiezioni degli elementi della tabella proiezioni\_ctr\_prg\_viabilit sui corrispondenti elementi della tabella ar\_str del DBT;
- proiezioni\_prg\_dbt\_argine(id, id\_segmento\_prg, id\_prg, layer\_prg, id\_segmento\_ctr, id\_ctr, layer\_ctr, id\_segmento\_dbt, id\_dbt, layer\_dbt, geometry): rappresentano le proiezioni degli elementi della tabella proiezioni\_ctr\_prg\_idrograf sui corrispondenti elementi della tabella argine del DBT;
- proiezioni\_prg\_dbt\_el\_div(id, id\_segmento\_prg, id\_prg, layer\_prg, id\_segmento\_ctr, id\_ctr, layer\_ctr, id\_segmento\_dbt, id\_dbt, layer\_dbt, geometry): rappresentano le proiezioni degli elementi della tabella proiezioni\_ctr\_prg\_ele\_div sui corrispondenti elementi della tabella el\_div del DBT;
- proiezioni\_prg\_dbt\_un\_vol(id, id\_segmento\_prg, id\_prg, layer\_prg, id\_segmento\_ctr, id\_ctr, layer\_ctr, id\_segmento\_dbt, id\_dbt, layer\_dbt, geometry): rappresentano le proiezioni degli elementi della tabella proiezioni\_ctr\_prg\_fabbric sui corrispondenti elementi della tabella un\_vol del DBT;
- proiezioni\_prg\_dbt\_completa(id, id\_segmento\_prg, id\_prg, layer\_prg, id\_segmento\_ctr, id\_ctr, layer\_ctr, id\_segmento\_dbt, id\_dbt, layer\_dbt, geometry): questa tabella contiene tutti gli elementi delle tabelle precedenti;
- proiezioni\_prg\_dbt\_ordinate(id, id\_segmento\_prg, id\_prg, layer\_prg, id\_segmento\_ctr, id\_ctr, layer\_ctr, id\_segmento\_dbt, id\_dbt, layer\_dbt, geometry): contiene gli elementi della tabella proiezioni\_prg\_dbt\_completa ma ordinati.

### **Calcolo delle corrispondenze tra PRG e DBT senza corrispondenza nella CTR**

La seconda parte riguarda il calcolo delle corrispondenze tra il PRG e il DBT senza corrispondenza nella CTR. Le tabelle utilizzate sono le seguenti:

- nuovi\_segmenti\_prg(id, id\_segmento\_prg, id\_prg, layer\_prg, geometry): i pezzi dei segmenti di PRG non ancora proiettati su elementi del DBT;
- nuove\_proiezioni\_prg\_dbt\_ar\_str(id, id\_segmento\_prg, id\_prg, layer\_prg, id\_segmento\_ctr, id\_ctr, layer\_ctr, id\_segmento\_dbt, id\_dbt, layer\_dbt, geometry): rappresentano le proiezioni degli elementi della tabella nuovi\_segmenti\_prg sugli elementi corrispondenti della tabella ar\_str;
- nuove\_proiezioni\_prg\_dbt\_argine(id, id\_segmento\_prg, id\_prg, layer\_prg, id\_segmento\_ctr, id\_ctr, layer\_ctr, id\_segmento\_dbt, id\_dbt, layer\_dbt, geometry): rappresentano le proiezioni degli elementi della tabella nuovi\_segmenti\_prg sugli elementi corrispondenti della tabella argine;
- nuove\_proiezioni\_prg\_dbt\_el\_div(id, id\_segmento\_prg, id\_prg, layer\_prg, id\_segmento\_ctr, id\_ctr, layer\_ctr, id\_segmento\_dbt, id\_dbt, layer\_dbt, geometry): rappresentano le proiezioni degli elementi della tabella nuovi\_segmenti\_prg sugli elementi corrispondenti della tabella el\_div;
- nuove\_proiezioni\_prg\_dbt\_un\_vol(id, id\_segmento\_prg, id\_prg, layer\_prg, id\_segmento\_ctr, id\_ctr, layer\_ctr, id\_segmento\_dbt, id\_dbt, layer\_dbt, geometry): rappresentano le proiezioni degli elementi della tabella nuovi\_segmenti\_prg sugli elementi corrispondenti della tabella un\_vol;
- nuove\_proiezioni\_prg\_dbt\_completa(id, id\_segmento\_prg, id\_prg, layer\_prg, id\_segmento\_ctr, id\_ctr, layer\_ctr, id\_segmento\_dbt, id\_dbt, layer\_dbt, geometry): questa tabella contiene tutti gli elementi delle tabelle precedenti;
- nuove\_proiezioni\_prg\_dbt\_ordinata(id, id\_segmento\_prg, id\_prg, layer\_prg, id\_segmento\_ctr, id\_ctr, layer\_ctr, id\_segmento\_dbt, id\_dbt, layer\_dbt, geometry): contiene gli elementi della tabella nuove\_proiezioni\_prg\_dbt\_completa ma ordinati;
- proiezioni\_prg\_dbt\_totali\_ordinate(id, id\_segmento\_prg, id\_prg, layer\_prg, id\_segmento\_ctr, id\_ctr, layer\_ctr, id\_segmento\_dbt, id\_dbt, layer\_dbt, geometry): contiene tutti gli elementi della tabella proiezioni\_prg\_dbt\_ordinate e della tabella nuove\_proiezioni\_prg\_dbt\_ordinata, queste proiezioni sono a loro volta ordinate.

## **Costruzione del nuovo piano regolatore**



La terza parte riguarda la costruzione del nuovo PRG. In quest'ultima fase le tabelle utilizzate sono:

- `segmenti_prg_poligoni(id, id_prg, nome_layer, geometry)`: questa tabella contiene i segmenti che costituiscono i poligoni contenuti nella tabella `residenza_trasferisci` del grafo planare costruito sul prg;
- `proiezioni_aggiornate(id, id_segmento_prg, id_prg, layer_prg, id_segmento_ctr, id_ctr, layer_ctr, id_segmento_dbt, id_dbt, layer_dbt, geometry)`: contiene gli elementi della tabella `proiezioni_prg_dbt_totali_ordinate` con i campi `id_segmento_prg` e `id_prg` aggiornati sulla base della loro corrispondenza agli elementi della tabella `segmenti_prg_poligoni`;
- `trasformazioni(id, id_segmento_prg, id_prg, layer_prg, id_segmento_ctr, id_ctr, layer_ctr, id_segmento_dbt, id_dbt, layer_dbt, geometry)`: questa tabella contiene gli elementi delle trasformazioni affini di alcune porzioni del boundary del PRG sulla base degli elementi contenuti nella tabella `proiezioni_aggiornate_ordinate`;
- `proiezioni_aggiornate_ordinate(id, id_segmento_prg, id_prg, layer_prg, id_segmento_ctr, id_ctr, layer_ctr, id_segmento_dbt, id_dbt, layer_dbt, geometry)`: contiene tutti gli elementi della tabella `proiezioni_aggiornate` ma ordinati, in seguito a questi elementi vengono aggiunti gli elementi della tabella `trasformazioni` senza però riordinare gli elementi;
- `nuovo_prg(id, nome_layer, geometry)`: contiene gli elementi del nuovo PRG, cioè il risultato finale di tutte le elaborazioni.

## RINGRAZIAMENTI

In quest'ultima sezione dell'elaborato vorrei ringraziare tutte le persone che mi hanno permesso arrivare fin qui.

Prima di tutto ringrazio la mia famiglia per avermi aiutato e sostenuto in questi anni, per avermi guidato in tutte le mie scelte con i loro preziosi consigli e per aver sempre avuto fiducia in me.

Ringrazio l'ing. Massimo Rumor e l'ing. Sandro Savino, rispettivamente relatore e correlatore, per la loro disponibilità e per il loro aiuto durante il lavoro di tesi.

Infine, ma non meno importanti, ringrazio gli amici che mi hanno sempre sostenuto anche nei momenti peggiori.

Grazie di cuore a tutti!

Elisabetta Panfilio