

UNIVERSITÀ DEGLI STUDI DI PADOVA
FACOLTÀ DI INGEGNERIA

Corso di laurea Magistrale in Ingegneria Informatica



PROGETTO CARGEN:
Generalizzazione di elementi lineari

Relatore: Prof. Massimo Rumor

Laureando: GIUSEPPE CASSANO
matricola N.602102/IF

A.A. 2010/2011

alla mia Famiglia

Sommario

Nella realizzazione di una carta geografica, la generalizzazione riveste il ruolo fondamentale di astrarre la realtà per rappresentarla in maniera chiara, comprensibile e leggibile, tenendo presente la funzione che la mappa deve svolgere e la scala di rappresentazione scelta.

Fino a pochi anni fa, quest'intensa opera di astrazione era interamente realizzata a mano dal cartografo, richiedendo talvolta anche tempi lunghissimi. Tuttavia, con l'avvento della moderna tecnologia, si è sviluppata la possibilità di automatizzare il processo di generalizzazione, riducendo tempi e costi.

In questo scenario si va a collocare il progetto di ricerca CARGEN, il cui obiettivo è quello di sviluppare un processo di generalizzazione automatico per la produzione di un database topografico in scala 1:25000 e 1:50000, partendo dal database territoriale in scala 1:5000, fornito dalla regione Veneto.

Il lavoro sviluppato in questa tesi si inserisce in tale progetto e si pone come obiettivo specifico la realizzazione di processi automatici per il trattamento delle adiacenze tra elementi lineari ed areali, il ripristino delle continuità degli elementi lineari ed infine la generalizzazione degli elementi divisorii.

Quindi nel seguito, dopo una breve introduzione alla cartografia e alla generalizzazione cartografica verranno presentate minuziosamente le idee sviluppate ed i risultati ottenuti inerenti allo studio dei tre problemi sopracitati.

Indice

Cenni di cartografia e di generalizzazione cartografica	5
1.1 Cenni storici	5
1.2 La carta Geografica.....	7
1.3 Il processo cartografico.....	10
1.4 I GIS.....	12
1.5 Definizione del concetto di generalizzazione cartografica	15
1.6 La ricerca sulla generalizzazione cartografica	18
1.7 Il processo di generalizzazione	19
1.7.1 La generalizzazione automatica	20
1.8 Modelli di generalizzazione	22
1.9 Gli operatori della generalizzazione	24
1.9.1 Gli operatori geometrici.....	25
1.9.2 Operatori per la trasformazione di attributi	26
1.9.3 Operatori e algoritmi	27
Il progetto di ricerca Cargen.....	29
2.1 La situazione cartografica in Italia.....	29
2.2 Il progetto CARGEN.....	31
2.2.1 La carta tecnica regionale numerica	31
2.2.2 Il GeoDBR	32
2.2.3 Il DB25.....	32
2.2.4 La derivazione del DB25 e del DB50	33
2.3 Strumenti e tecniche.....	35

2.3.1 JTS Topology Suite	36
2.3.2 OpenJump	36
2.3.3 Triangolazione di Delaunay	37
2.3.4 Cartografia utilizzata per i test	39
Tattamento adiacenze elementi areali ed elementi lineari.....	43
3.1 Semplificazione iniziale delle geometrie	43
3.2 Casistiche individuate	44
3.3 Caso A.....	45
3.4 Eliminazione dei broli e delle radure.....	48
3.5 Caso B.....	50
3.5.1 Triangolazione tra boschi e strade.....	51
3.5.2 Ricostruzione ed unione dei boschi.....	58
Ripristino della continuità degli elementi lineari	61
4.1 Pulizia dei dati in ingresso	61
4.1.1 Eliminazione delle sporgenze	62
4.1.2 Eliminazione delle micro interruzioni	64
4.2 Generazione di nuovi potenziali elementi lineari	67
4.2.1 Creazione di elementi lineari attraverso lati degli edifici	68
4.3 Chiusura degli elementi lineari	70
4.3.1 Selezione dei lati dei triangoli.....	71
4.3.2 Eliminazione di elementi non idonei.....	75
Generalizzazione di elementi divisorii	79
5.1 Selezione di elementi divisorii.....	79
Conclusioni.....	83

Bibliografia.....85

Capitolo 1

Cenni di cartografia e di generalizzazione cartografica

In questo capitolo vengono dapprima introdotte una serie di nozioni relative ai concetti di cartografia, carta geografica e scala. Successivamente verranno affrontati il tema della generalizzazione cartografica, i motivi che portano alla realizzazione della generalizzazione cartografica, la generalizzazione automatica ed infine saranno descritti alcuni modelli concettuali di generalizzazione presenti in letteratura.

1.1 Cenni storici

La cartografia è la scienza che si occupa di riprodurre l'aspetto della terra o di una sua parte con la minima distorsione possibile, in modo che le informazioni contenute sulla carta siano il più possibile accurate (Enc. Italiana Grolier, 1987).

Le prime testimonianze di carte che ci sono pervenute risalgono a circa 15000 anni fa: il graffito di Mezin (Ucraina) rappresenta un accampamento ed un fiume che scorre nelle vicinanze.

Un ulteriore reperto di interesse cartografico (10000 – 6000 anni fa) è il graffito di “Jebel Amud”, rinvenuto in Giordania. Si tratta di una pietra incisa con coppelle e canalini che rappresentano rispettivamente gli insediamenti abitativi ed i percorsi. Con lo sviluppo delle grandi civiltà mediorientali, si sentì la necessità di conservare traccia dei possedimenti conquistati e di rappresentare i limiti delle proprietà fondiarie. Risalgono a tale periodo (2400 – 2200 a.C.) numerosi abbozzi cartografici su tavolette di argilla che rappresentano la Terra come un cerchio circondato da acqua.

Anche gli Egizi produssero delle rappresentazioni della superficie terrestre, costituite da schizzi che schematizzavano i reciproci rapporti spaziali e gli

elementi raffigurati. Grandi matematici ed astronomi, tuttavia gli Egizi avevano anche compiuto studi geodetici per stimare il raggio terrestre, utilizzando la piramide di Cheope.

Successivamente, furono alcuni grandi pensatori Greci, come Pitagora ed Aristotele, i primi ad ipotizzare e sostenere la sfericità della terra. Quindi, si può affermare che la geodesia mosse i primi passi in questo periodo, con il primo calcolo del raggio terrestre da parte di Eratostene di Cirene (276 – 194 a.C.), al quale si deve anche la nascita del termine geografia. Risalgono anche a questo periodo, con Marino di Tiro, l'introduzione della misura delle latitudini e longitudini in gradi d'arco sessagesimale e la proiezione cartografica.

Le acquisizioni ellenistiche andarono perse nell'epoca Romana, quando furono abbandonati gli studi geodetici speculativi, limitandosi alla compilazione di diagrammi che riportano le sequenze di località e le distanze lungo gli itinerari. Non si tratta quindi di carte geografiche con una proiezione da cui si possono ricavare le distanze reciproche degli elementi rappresentati ma solo di semplici schemi, di cui è un esempio la Tavola Peutingeriana, riscoperta nel 1500. Dalla perdita delle conoscenze ellenistiche derivò il ritorno, per tutto il medioevo, alla navigazione sottocosta. La "Geographia", il testo di Tolomeo nel quale raccoglieva le cognizioni geografiche e geodetiche andò perduto e solo gli arabi ne conservarono delle copie, che ritornarono in Occidente solo nel XV secolo.

Per assistere ad un nuovo sviluppo della cartografia, come per molte altre scienze, bisogna aspettare il rinascimento. In questo periodo, la cartografia si dimostrò irrinunciabile supporto per il progresso delle conoscenze dell'umanità.

La creazione dei primi strumenti per l'osservazione celeste e terrestre, il perfezionamento della bussola e degli orologi, e l'invenzione della stampa, permettono la creazione e la diffusione di mappe sempre più accurate.

Le mappe diventano sempre più simili a quelle in uso oggi, con l'introduzione delle curve di livello per descrivere l'orografia (al posto della rappresentazione a mucchi di talpa) e l'utilizzo di proiezioni e rappresentazioni moderne come quella di Gauss.

Con la nascita della geodesia, la disciplina che studia la forma e le dimensioni della Terra e determina la posizione precisa dei punti sulla sua superficie (diz. DeMauro) e, con la fondazione dei primi Istituti Cartografici Nazionali (1791 Inghilterra, 1817 Francia, 1872 Italia), si assiste alla creazione delle cartografie nazionali: i primi a dotarsene furono i francesi, nel 1818, mentre in Italia solo nel 1921 venne completata la Nuova Carta Topografica d'Italia ad opera dell'Istituto Geografico Militare.

Negli ultimi anni dello scorso secolo la cartografia ha conosciuto un nuovo momento di grande sviluppo, che può essere considerato come una specie di rivoluzione, con l'introduzione dell'informatica in questo campo e la nascita dei primi Sistemi Informativi Geografici (GIS). I dati cartografici si svincolano dalla staticità del tradizionale supporto cartaceo e possono essere visualizzati in modo dinamico su un monitor, creando così la possibilità di nuovi utilizzi ed al contempo, introducendo nuove problematiche e necessità. I GIS diventano potenti strumenti di analisi dei dati geografici: le informazioni e le relazioni spaziali possono venire elaborate, manipolate e modificate. Per poter utilizzare le mappe con questi nuovi strumenti è necessario dapprima digitalizzarle e successivamente si dà avvio alla produzione in formato esclusivamente digitale.

I GIS rivoluzionano il modo di concepire e trattare il dato geografico: gli oggetti spaziali non sono più una collezione di segni grafici e simboli, ma diventano entità dotate di attributi, inserite in gerarchie relazionali, cui corrisponde non solo una rappresentazione grafica ma anche un contenuto logico e semantico.

1.2 La carta Geografica

La carta geografica è una rappresentazione ridotta, simbolica ed approssimata della superficie terrestre o di parte di essa su un piano. È ridotta in quanto le distanze e le aree vengono ridotte secondo un rapporto, detto *scala*, tra le misure reali e quelle riportate sulla carta; è simbolica in quanto si ricorre all'uso di simboli, astratti o figurativi, per rappresentare gli oggetti da riprodurre; è

approssimata per le deformazioni dovute all'impossibilità di sviluppare senza alterazioni su un piano una superficie sferica (Enc. Treccani).

La produzione di una carta geografica è un processo molto complesso durante il quale è necessario affrontare molteplici problemi.

Il primo deriva dal fatto che la superficie terrestre è ricurva, mentre il supporto cartaceo è piano; il secondo deriva dal fatto che gli elementi in questione non possono essere rappresentati nelle dimensioni originali, ma devono passare attraverso un processo di riduzione di scala; infine, l'ultimo problema deriva dall'eterogeneità degli elementi che compongono la realtà, rendendo possibile la sua rappresentazione solo ed esclusivamente dopo un'accurata semplificazione.

Da quanto appena detto consegue che la produzione di una carta geografica deriva da un'astrazione della realtà, in cui gli elementi subiscono sia trasformazioni spaziali, che trasformazioni semantiche. Il risultato di queste trasformazioni è un'approssimazione della realtà.

Un aspetto molto importante nella definizione di mappa cartografica è il concetto di scala. Essa è il rapporto di riduzione delle lunghezze sulla carta rispetto a quelle nella realtà. La misura della scala è indicata da un rapporto $1 : n$, dove n è il numero di volte che sulla carta è stata ridotta la distanza topografica fra due punti. Più grande è il rapporto (e perciò più piccolo è il denominatore n), maggiori sono i dettagli che si vedono e viceversa. Se si vuole rappresentare un piccolo territorio con molti dettagli si utilizzerà una grande scala, viceversa, per rappresentare un grande territorio con i soli elementi principali basta una piccola scala.

Da un punto di vista metrico ogni carta è caratterizzata da due parametri i quali sono strettamente connessi alla scala: il grado di risoluzione e l'errore di posizionamento.

Il grado di risoluzione, cioè la dimensione lineare del particolare più piccolo rappresentabile, è dato dal minimo spessore del tratto grafico con cui la carta viene disegnata e viene assunto, per convenzione, uguale a 0,2 mm. L'errore massimo di posizionamento di un punto rappresenta il diametro del cerchio al cui interno il punto è sicuramente contenuto e corrisponde all'incertezza con cui è

rappresentata la posizione di un generico punto sulla carta. Questo valore è assunto pari a 0,5 mm.

Il livello di dettaglio di una mappa ha un impatto sull'accuratezza dei dati presentati, sul suo contenuto e in generale anche sulla rappresentazione grafica, ma anche sul costo di produzione della stessa.

Le carte geografiche possono essere classificate in base alla loro scala di riduzione oppure in base al loro contenuto. In base alla scala di riduzione si possono distinguere:

1. *Piante o mappe* hanno una scala maggiore di 1:10000 (cioè possono avere valori compresi tra 1:1 e 1:10000). Le piante rappresentano la planimetria di città, mentre le mappe riguardano essenzialmente la raffigurazione di zone rurali. Si tratta di carte molto dettagliate perché riguardano porzioni relativamente piccole di territorio. Fanno parte di questa categoria le carte tecniche regionali (CTR) ;
2. *Carte topografiche*. La loro scala è compresa tra 1:10000 e 1:150000. Esse includono aree maggiori e sono meno dettagliate delle piante e delle mappe, ma riescono comunque ad evidenziare particolari molto piccoli come una casa isolata o un ponte;
3. *Carte corografiche*. La loro scala va da 1:150000 a 1:1000000. Raffigurano zone assai estese ed il numero dei particolari è assai ridotto ma comunque significativo;
4. *Carte geografiche*. Hanno scala inferiore ad 1:1000000 e servono per rappresentare aree molto estese della superficie terrestre, quali ad esempio uno stato o un continente.

In base al contenuto le carte geografiche si classificano in:

1. *Carte tematiche* sono rappresentazioni di fenomeni o di concezioni astratte qualitative o quantitative, limitate ad uno o più temi specifici, come ad esempio la frequenza di tornado o la quantità di piogge di un certo territorio;

2. *Carte topografiche* contengono una rappresentazione della superficie fisica di un territorio, costituita da un terreno con le sue forme, dalle acque (laghi, mari, fiumi), da oggetti concreti durevoli (edifici, strade, ponti, ect.). Nella carta sono contenuti anche altri elementi informativi quali i confini amministrativi e variazioni di colore indicanti l'altimetria e la batimetria.

1.3 Il processo cartografico

Con processo cartografico si definisce tutto l'insieme di procedimenti e operazioni necessari alla creazione di una carta geografica.

Il processo cartografico è costituito dalle seguenti fasi:

1. Definizione;
2. Analisi;
3. Raccolta dei dati;
4. Costruzione della mappa;
5. Collaudo.

Durante la fasi di analisi e definizione, vengono decise le caratteristiche che il prodotto finito dovrà possedere. Viene deciso cosa rappresentare sulla mappa e come rappresentarlo e le decisioni prese in questa fase, influenzeranno sia le caratteristiche tecniche della carta, che quelle semantiche, relative cioè ai contenuti.

Per quanto riguarda le caratteristiche tecniche di una carta geografica, si possono evidenziare la superficie di riferimento, la superficie di proiezione adottata, il tipo di rappresentazione (conforme, equivalente, equidistante), la tipologia della sua generazione (geometrica, semigeometrica, analitica) e la scala. Invece, per quanto concerne le caratteristiche semantiche, è possibile condurre l'attività di astrazione ed esemplificazione dei dati in ingresso, in modo tale da focalizzare l'attenzione solo su alcuni particolari aspetti della realtà da rappresentare (cartografia meristica), oppure cercare di avere una rappresentazione il più completa possibile (cartografia olistica).

La fase successiva è la raccolta dei dati. In questa sezione si possono distinguere due differenti possibilità: la prima è che la campagna di raccolta dati sia effettuata tramite campagne di acquisizione effettuate sul territorio (carte rilevate), invece, la seconda possibilità consiste nell'utilizzare dati presi da cartografia preesistente come dati di partenza (carte derivate).

L'acquisizione dei dati per la realizzazione di cartografia rilevata si è evoluta molto nel corso degli anni, partendo dalle ricognizioni a terra tramite strumenti quali la tavoletta pretoriana e il teodolite, sino ad arrivare alle più recenti tecnologie quali GPS e fotogrammetria, che consiste nella rilevazione dei punti mediante l'utilizzo di immagini stereoscopiche sul terreno.

La costruzione della mappa consiste nella produzione di una carta che deve necessariamente soddisfare oltre che le specifiche decise in fase di definizione, anche alcuni requisiti fondamentali per una mappa, quali la leggibilità e l'usabilità della mappa stessa. Per perseguire questo obiettivo, è necessario estrarre dai dati di partenza una loro rappresentazione astratta ma nel contempo efficace e rappresentativa. E' in questa fase che viene eseguita la cosiddetta generalizzazione cartografica, cioè la fase in cui il cartografo sceglie e posiziona uno ad uno gli oggetti sulla carta finale.

Il collaudo è l'ultima fase del processo e prevede che la carta venga sottoposta ad una serie di test per verificarne la correttezza e la consistenza . In questa circostanza si può assistere ad un affinamento estetico del prodotto, oltre che ad una verifica della validità della rappresentazione creata.

L'introduzione dell'informatica nel mondo della cartografia, ha introdotto dei cambiamenti nel modo di produrre una mappa. Si può distinguere tra il processo cartografico map-oriented, nel senso più tradizionale di mappa cartacea, e quello gis-oriented, definizione che si applica alle mappe destinate ad una fruizione digitale.

Nel processo cartografico tradizionale il cartografo, durante la fase di compilazione, prende i dati in ingresso e posiziona le informazioni sulla mappa tenendo in considerazione una grande moltitudine di parametri (specifiche sulla simbologia, regole di posizionamento, standard di rappresentazione, ect).

Le attività di disegno e di compilazione sono fortemente connesse l'una con l'altra e il cartografo effettua le sue scelte sulla base di standard tecnici e grafici per la comunicazione visiva.

Durante questo processo, il contenuto informativo della mappa è considerato della stessa importanza della presentazione grafica delle informazioni: il rispetto dei vincoli di rappresentazione grafica, ad esempio l'uniformità del rapporto bianco-nero, ha la stessa priorità del rispetto di vincoli sul contenuto informativo della mappa come, ad esempio, la rappresentazione di tutte le strade maggiori.

In tutto questo, la scala gioca un ruolo fondamentale dal momento che, stabilendo la dimensione minima dei simboli, il livello di dettaglio, la densità massima di informazione e via dicendo; agisce come una sorta di filtro sul contenuto informativo della mappa.

Nel processo cartografico *gis-oriented*, invece, c'è una separazione molto più marcata tra il contenuto informativo e la rappresentazione dello stesso.

1.4 I GIS

Un GIS (Geographical Information System) o SIT (Sistemi Informativi Territoriali), è un sistema per la gestione l'analisi e la visualizzazione di informazioni con contenuto geografico spaziale.

L'informazione è gestita tramite insiemi di dati (dataset geografici) che costituiscono modelli di fenomeni geografici, cioè riferibili al territorio.

Con l'avvento dei GIS, le mappe diventano degli oggetti dinamici, in cui il singolo dato geografico è costituito da un'entità dotata di diversi attributi ed inserita in una gerarchia relazionale. Oltre che una rappresentazione grafica, a questa entità corrisponde anche una interpretazione logica e semantica.

Le entità, elementi base del GIS, sono memorizzate in una base di dati (GeoDB), il cosiddetto database spaziale, che ha la funzione di memorizzare, interrogare e manipolare informazioni geografiche e dati spaziali.

Dal punto di vista strutturale, si può dire un GIS è un sistema costituito da risorse e procedure. Le risorse coinvolte sono risorse umane, infrastrutturali e di dati; le

procedure, invece, si suddividono in acquisizione, archiviazione, elaborazione, presentazione e trasmissione.

I dati sono in assoluto l'elemento più prezioso del sistema, perché hanno un costo di produzione elevato. Essi, inoltre, per mantenere il loro valore, devono costantemente essere aggiornati. Ogni dato, che viene chiamato *feature*, è costituito da due componenti: la componente spaziale, che contiene informazioni relative alla posizione (geografica), alla geometria (forma e dimensione) e alla topologia (relazioni spaziali con altri dati) del dato; e la componente non spaziale, formata da dati descrittivi, o attributi, e metadati associati alla componente spaziale che velocizzano le operazioni di query spaziale.

L'acquisizione dei dati avviene normalmente digitalizzando le mappe cartacee tradizionali. Un altro metodo, invece, si basa sull'utilizzo di file contenenti le coordinate degli elementi geografici, espresse secondo un determinato sistema di riferimento. In entrambi i casi, l'acquisizione è seguita da un processo di astrazione e generalizzazione.

Ovviamente, al momento dell'acquisizione è necessario memorizzare i dati. Il modo in cui le informazioni spaziali vengono memorizzate dipende da quale modello dello spazio viene adottato dal GIS. Ci sono due modelli utilizzabili: il modello raster e il modello vettoriale.

Nel primo caso, lo spazio viene visto come un mosaico, dove ogni tassello, o pixel, contiene uno o più valori (ad esempio, il colore o l'altitudine). Il pixel rappresenta il luogo dove una grandezza assume un certo valore. Un file raster risulta allora costituito da una matrice quadrata dove ogni cella è un pixel e le dimensioni di tale file tendono ad essere molto elevate, in quanto ogni unità di spazio è sempre descritta da un pixel.

Il secondo modello, quello vettoriale, è formato da una serie di primitive geometriche che vengono istanziate all'interno di uno spazio vuoto. In questo secondo caso, a differenza di quanto avviene nel modello raster, lo spazio non viene riempito completamente, ma funge da contenitore di oggetti. Le primitive vettoriali che costituiscono il modello sono:

1. il punto, definito da una coppia (x,y) o da una terna di coordinate (x, y, z) ;

2. la linea, costituita da una sequenza ordinata di punti;
3. il poligono, anch'esso costituito da una sequenza ordinata di punti, dove però l'ultimo punto della sequenza corrisponde al primo.

Il punto viene utilizzato quando l'elemento che si vuole rappresentare è semanticamente un punto, oppure quando, alla scala in cui si opera, esso è di fatto un punto. La medesima considerazione vale anche per la linea. Il poligono, invece, viene utilizzato per definire una particolare area della superficie geografica, ad esempio rappresentando laghi, edifici o confini amministrativi.

Comparando il modello raster e quello vettoriale, emerge che quest'ultimo si presta molto di più alla manipolazione, all'elaborazione e all'interrogazione dei dati, che si possono integrare facilmente in un database relazionale, dove vengono trattati come valori di una colonna di una tabella (solitamente chiamata "Geometry") e processati per mezzo di molteplici operatori. Nonostante ciò, il GIS viene progettato in modo che possa supportare entrambi i modelli; a seconda di quale utilizzo viene fatto della carta, poi, sarà adottato un modello piuttosto che l'altro. Ad esempio, le scansioni aeree o le immagini satellitari nascono e vengono utilizzate in formato raster; inoltre, in fase di stampa la mappa viene spesso convertita dal formato vettoriale a quello raster, per consentire stampe in grande quantità con procedure tipografiche.

Una volta che i dati sono stati acquisiti e opportunamente memorizzati secondo uno dei due modelli appena illustrati, è possibile effettuare su di essi operazioni di elaborazione e trasformazione degli elementi geografici tramite degli strumenti di analisi forniti dal GIS.

Alcune di queste operazioni sono:

1. *Overlay topologico*, in cui si effettua una sovrapposizione tra gli elementi di due temi per creare un nuovo tematismo, ad esempio per sovrapporre il tema dei confini di un parco con i confini dei comuni in modo da determinare le superfici di competenza di ogni amministrazione o la percentuale di area comunale protetta;

2. *Overlay topologico*, in cui si effettua una sovrapposizione tra gli elementi di due temi per creare un nuovo tematismo, ad esempio per sovrapporre il tema dei confini di un parco con i confini dei comuni in modo da determinare le superfici di competenza di ogni amministrazione o la percentuale di area comunale protetta;
3. *Query spaziali*, ovvero interrogazioni dei dati a partire da criteri spaziali (vicinanza, inclusione, sovrapposizione etc.) ;
4. *Buffering*, che permette di creare un poligono che circonda la geometria originaria;
5. *Segmentazione*, che applicato su un elemento lineare, determina il punto alla distanza specificata dall'inizio dell'elemento;
6. *Network analysis*, che applicata su una rete di elementi lineari, ad esempio la rete stradale, determina i percorsi minimi tra due punti;
7. *Spatial analysis* effettua un'analisi spaziale di varia tipologia utilizzando un modello raster, come ad esempio l'analisi di visibilità.

1.5 Definizione del concetto di generalizzazione cartografica

Osservando una carta geografica, si può evincere come gli elementi della realtà vengono ridefiniti, classificati e posizionati secondo un ordine logico e grafico ben definito. Se non fossero compiute queste azioni di classificazione e ridefinizione, l'informazione veicolata tramite la carta sarebbe sensibilmente inferiore. Ad esempio, volendo riportare in una mappa topologica di una città in scala 1:100000 l'intero edificato, si otterrebbe nient'altro che una nuvola di punti, creando esclusivamente disordine e confusione.

Ne consegue che non ha senso tentare di rappresentare nella mappa tutta la realtà, al contrario è necessario selezionare le informazioni rilevanti. Quindi, si evince che occorre effettuare una astrazione della realtà per rappresentarla in maniera chiara, comprensibile e leggibile, tenendo presente la funzione che la mappa deve svolgere, in accordo con la scala e la rappresentazione scelta. Quanto appena

detto, può essere considerato come una breve sintesi della definizione di processo di generalizzazione.



Figura 1.1: generalizzazione cartografica

La generalizzazione è quel processo che si occupa di creare un modello della realtà osservata, i cui elementi siano semplificati in base alle caratteristiche e agli scopi della mappa che si vuol creare.

Inoltre, è importante sottolineare che il processo di generalizzazione deve creare una rappresentazione grafica della realtà adatta ad essere visualizzata sul supporto scelto, sia esso digitale o cartaceo.

I fattori che devono essere tenuti in considerazione durante il processo di generalizzazione sono svariati. Gli studiosi McMaster e Shea (1992) ne hanno individuato sei:

1. *riduzione della complessità* nel passaggio ad una scala inferiore, attraverso la diminuzione del numero di oggetti, evitando che questi ultimi entrino in conflitto, dando risalto sulla mappa agli elementi di maggiore importanza;
2. *mantenimento dell'accuratezza spaziale*. Quanto più alta è la scala, tanto più l'accuratezza spaziale deve essere rispettata, limitando l'errore dovuto alla diversa posizione degli oggetti nella mappa rispetto alla realtà;
3. *mantenimento dell'accuratezza degli attributi*, con l'obiettivo di minimizzare le alterazioni non intenzionali degli attributi della *feature*;
4. *mantenimento della qualità estetica*. Vari fattori influenzano questo parametro (colori utilizzati, simbologia, bilanciamento, stile topografico,

ect.) e il cartografo deve opportunamente mixare questi fattori in modo tale da ottenere una mappa esteticamente bella;

5. *mantenimento di una logica gerarchica*, attraverso la differenziazione di elementi appartenenti alla stessa categoria ma con importanza differente. Ad esempio una città grande deve risultare molto più accentuata rispetto ad una più piccola;

6. *applicazione coerente delle regole di generalizzazione* da parte del cartografo, decidendo l'ordine degli algoritmi da eseguire e i parametri di input necessari per conseguire il risultato desiderato, con lo scopo di ottenere una generalizzazione imparziale e coerente.

Se si sta disegnando una mappa derivata, partendo cioè da dati cartografici preesistenti, il processo di generalizzazione viene detto “map generalization”, oppure “generalizzazione cartografica”. Se, invece, si sta realizzando una mappa rilevata, in cui i dati sono frutto di acquisizione diretta, il processo viene detto “map compilation”.

Tuttavia, esaminando questi due processi di generalizzazione è possibile notare che essi devono interagire con le stesse problematiche riguardanti il contenuto della mappa e la sua rappresentazione, infatti entrambi devono sottostare alle specifiche della carta e sono ambedue influenzati dagli stessi parametri. Quindi, da quanto appena evidenziato, segue che non è errato affermare che non esiste una reale distinzione tra il processo di costruzione di una mappa rilevata e di una derivata. Per esempio, il problema di scegliere quali edifici selezionare da una mappa a scala più grande è del tutto sovrapponibile al problema di decidere quali selezionare da uno scatto di una ripresa fotogrammetrica.

La differenza fondamentale tra questi due processi di generalizzazione risiede nel fatto che nella produzione di una mappa rilevata bisogna creare una rappresentazione astratta partendo dalla realtà, nella produzione di una mappa derivata è necessario operare questa astrazione partendo da un modello della realtà. In entrambi i casi, comunque, il risultato sarà un nuovo modello della realtà.

1.6 La ricerca sulla generalizzazione cartografica

Tradizionalmente il processo di generalizzazione cartografica è stato realizzato tramite il lavoro manuale dei cartografi, costituendo un compito lungo, complesso e fastidioso.

Il progresso tecnologico degli ultimi 30 anni ha tuttavia incoraggiato la ricerca di una procedura automatica che potesse prima affiancare il cartografo nel suo lavoro manuale e infine sostituire completamente il suo operato.

Secondo lo studioso Kilpelainen, questa ricerca può essere divisa in tre periodi (Kilpelainen, 2000): nel primo periodo, dal 1960 al 1975, gli studi si sono focalizzati sullo sviluppo di algoritmi singoli che potessero risolvere particolari problemi ben specifici, come ad esempio la generalizzazione di una linea o del perimetro di un'area; nel secondo periodo, intorno agli anni '80, quello che era un lavoro di ricerca puntuale e circoscritto a pochi casi, è diventato un campo di ricerca a più ampio respiro: i ricercatori hanno fatto una revisione critica di quanto sviluppato fino a quel momento e si sono concentrati su un'analisi ad ampio raggio dell'intero concetto di generalizzazione. Il processo di generalizzazione è stato modellato, sono state concepite diverse strategie realizzative e il processo è stato suddiviso e schematizzato nelle sue parti costituenti.

Dagli anni '90 in poi tutto il lavoro di ricerca è confluito in nuovi approcci, più globali al problema; vengono introdotte nuove tecnologie e si assiste allo sviluppo di tecniche di intelligenza artificiale, sistemi esperti, reti neurali, algoritmi genetici e sistemi multiagente.

Nonostante tutti gli sforzi e i traguardi raggiunti in questo campo, al momento non è ancora stata trovata una soluzione definitiva al problema, e il processo di generalizzazione, per quanto sia facilitato dagli strumenti sviluppati in tutti questi anni di ricerca, richiede tuttora l'intervento manuale del cartografo.

Nel paragrafo seguente verranno illustrati i più validi modelli concettuali della generalizzazione realizzati negli oltre quaranta anni di ricerca in questo campo.

1.7 Il processo di generalizzazione

Secondo lo studioso Meyen (Meyen, 1973), la generalizzazione può essere definita come “la selezione e la rappresentazione semplificata dei dettagli che meglio si adattano alla scala ed allo scopo di una mappa”.

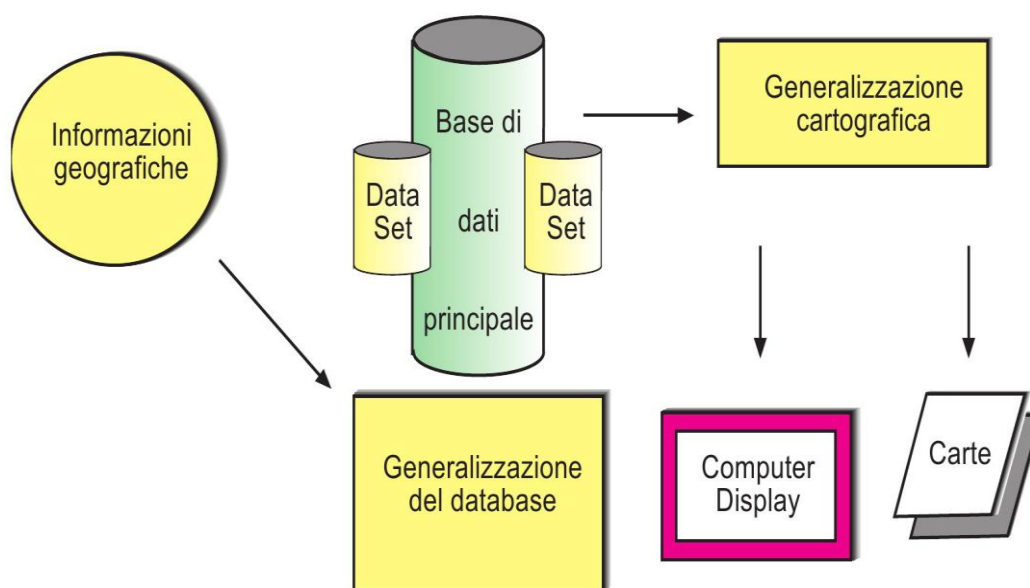


Figura 1.2: fasi della generalizzazione all'interno di un GIS, attraverso modellazione delle informazioni geografiche, che definisce il modello dati del GeoDBR, e generalizzazione delle feature.

Le componenti che caratterizzano questo processo (Figura 1.2) sono due: la generalizzazione del modello e la generalizzazione cartografica.

La generalizzazione del modello ha l'obiettivo di creare un modello astratto che sia in grado di rappresentare la realtà. La scelta di quali aspetti riportare sulla mappa e quali eliminare, è fortemente influenzata dallo scopo per il quale la mappa viene realizzata. Ad esempio in una carta politica, verrà data parecchia importanza ai confini amministrativi, al contrario gli elementi fisici quali rilievi e depressioni non verranno riportati.

Quindi, si sviluppa una specie di tassonomia degli elementi della realtà, cioè una loro suddivisione in classi. Anche i parametri che portano a distribuire gli oggetti

in più classi variano a seconda delle funzioni della mappa che si vuole realizzare. Ad esempio, in una carta civile le strade possono essere classificate in urbane ed extraurbane, mentre una carta militare può enfatizzare maggiormente una distinzione delle vie di comunicazione basata sulla larghezza e sul tipo di fondo.

Nell'ambito della cartografia digitale, la generalizzazione del modello corrisponde alla definizione di un GeoDatabase. Nella definizione di questa base di dati è fondamentale ricordare che una modellazione e strutturazione efficiente rappresenta un requisito indispensabile per la buona riuscita della generalizzazione cartografica, oltre che per un proficuo utilizzo della cartografia digitale in ambiente GIS.

Una volta definito il modello si passa alla generalizzazione cartografica. In questa fase viene effettuata una ridefinizione delle geometrie, allo scopo di fornire una rappresentazione grafica del modello dei dati. E' importante sottolineare che non esiste uno schema rigido nella ridefinizione delle geometrie da parte del cartografo.

I principali fattori che influenzano le scelte del cartografo sono la scala ed il contesto. Il contesto serve a stabilire il livello di importanza attribuito ad un oggetto con l'obiettivo di sancirne o meno la sua eliminazione. Ad esempio, un edificio posizionato in una zona densamente popolata potrebbe essere semplificato, o anche omesso del tutto, invece, una fattoria situata in aperta campagna, potrebbe essere riportata in maniera accurata.

1.7.1 La generalizzazione automatica

La generalizzazione manuale, che un tempo era l'unica opzione praticabile, è un'operazione lunga ed onerosa. Fortunatamente, le tecnologie informatiche hanno permesso di automatizzare questa procedura, riducendo notevolmente fatica e tempo che l'operatore umano deve dedicarvi.

Per capire l'importanza che riveste l'automazione del processo di produzione delle mappe geografiche, può esser utile ricorrere ad un esempio pratico. In Francia, per la produzione di un foglio 91 x 121 cm della serie Topo100 con i metodi tradizionali, l'IGN (Institute National Geographique) impiega mediamente 2000

ore di lavoro, di cui 1200 per la generalizzazione e 800 per il posizionamento della toponomastica. Se si suppone che i fogli vengano realizzati in sequenza, ci vogliono circa 20 anni per realizzare le mappe relative all'intero territorio francese.

Diminuendo il tempo di produzione, invece, si possono aggiornare le mappe con sempre maggior frequenza raggiungendo, naturalmente, anche un obiettivo di risparmio economico. Così, nel 2003, l'IGN ha utilizzato per la prima volta un software per la generalizzazione automatica (LAMPS2 della Laser-Scan) e ha ridotto i tempi medi per la produzione di un foglio a 150 ore per la generalizzazione e a 160 per il posizionamento dei nomi.

È evidente, dunque, il motivo per cui il campo della generalizzazione automatica desta sempre più interesse presso tutti gli enti cartografici del mondo, stimolando la produzione scientifica in questo innovativo settore.

La ricerca relativa all'automazione di questi processi si è focalizzata soprattutto sulla generalizzazione cartografica, in quanto sembra l'unica delle due fasi della generalizzazione per cui è possibile immaginare una forma di automazione; la generalizzazione del modello, infatti, presuppone una capacità di astrazione ed un livello di conoscenza difficilmente trasferibile ad un calcolatore.

La realizzazione di un processo di generalizzazione automatico si deve comunque scontrare con alcuni ostacoli (Mackness, 2007). Un primo e comprensibile ostacolo è sicuramente la complessità del processo di progettazione, che non può prescindere dalla valutazione di molteplici fattori e vincoli: la mappa è un complesso mix di *pattern* metrici e topologici che solitamente sono molto interdipendenti. Interpretare correttamente queste forme e individuare le caratteristiche notevoli della mappa richiede una conoscenza sia cartografica che geografica. Un secondo ostacolo, inoltre, ha a che fare con la trasformazione delle informazioni dovuta ad un cambiamento di scala: le mappe infatti, al variare della scala, mettono in luce caratteristiche geografiche diverse.

Il processo di generalizzazione deve, perciò, essere in grado di estrapolare queste differenti caratteristiche in funzione della scala, a partire dalla stessa base di dati.

1.8 Modelli di generalizzazione

La generalizzazione, dagli anni '70 ad oggi, ha subito un'evoluzione che, a partire da un approccio basato su semplici algoritmi, l'ha portata ad essere definita tramite modelli matematici sempre più complessi.

Le prime attività di studio sul tema della generalizzazione, infatti, si focalizzavano sullo sviluppo di singoli algoritmi che fossero in grado di risolvere specifici problemi, come ad esempio la generalizzazione di una linea o del perimetro di un'area. Dal 1980 in poi, i ricercatori hanno cominciato a dare un'impostazione più organica ai loro studi, investigando su una possibile modellazione concettuale della generalizzazione.

Alla fine degli anni '70, Robinson (1978) sviluppò uno dei primi modelli concettuali, allo scopo di capire meglio il processo di generalizzazione. Egli suddivise il processo in due sotto-processi principali: la selezione, o preprocessing, e l'effettivo processo di generalizzazione. La prima fase coinvolge l'attività di identificazione degli elementi geografici da memorizzare nel database; la seconda interessa la manipolazione geometrica e statistica degli oggetti. Quest'ultimo sotto-processo viene suddiviso ulteriormente in tre fasi: semplificazione, ovvero eliminazione dei dettagli geografici non necessari; classificazione, cioè la categorizzazione degli oggetti; e simbolizzazione, o associazione di una codifica grafica ad ogni elemento.

Pochi anni dopo, Bertin (1983) parlò di generalizzazione concettuale e strutturale. Nella generalizzazione concettuale è possibile cambiare la natura di un elemento, definendo un nuovo livello di concettualizzazione. Ad esempio, nella generalizzazione concettuale si può decidere di trasformare un cluster di punti in un'area. Nella generalizzazione strutturale la natura del fenomeno rimane inalterata, mentre si agisce per semplificare la sua distribuzione; ciò avviene basandosi solo sulle informazioni già presenti in fase di elaborazione. Ad esempio, il cluster di punti, invece di cambiare natura, viene generalizzato riducendo il suo numero di punti.

Grande importanza riveste anche il lavoro di Nickerson e Freeman (1986), che schematizzarono la generalizzazione usando un modello suddiviso in cinque compiti: modifica delle *feature* per mezzo di quattro operatori (selezione, semplificazione, combinazione e conversione di tipo); ridimensionamento dei simboli; displacement delle *feature*; riduzione della scala; posizionamento dei nomi. In questo modello, la generalizzazione della mappa alla scala finale passa per una mappa a scala intermedia, nella quale i simboli vengono prima scalati di un fattore k , per poi essere riposizionati tramite l'operazione di displacement.

Un grosso contributo deriva dal lavoro di McMaster e Shea (1992), che proposero un modello concettuale globale del processo di generalizzazione digitale. Essi suddivisero il processo in tre questioni perché generalizzare (*why*), quando (*when*) e come (*how*). Per rispondere alla domanda *why* vengono individuati degli obiettivi filosofici (elementi teorici, elementi computazionali ed elementi specifici); in merito al *when* si specifica una valutazione cartometrica delle condizioni che motivano la generalizzazione (condizioni geometriche, misure spaziali e olistiche, controllo delle trasformazioni); alla domanda *how* viene stilato un elenco degli operatori della generalizzazione, suddivisi in relazione all'aspetto dell'oggetto che trasformano (spaziale o degli attributi).

Lamy e Ruas (1999) hanno proposto recentemente un modello di generalizzazione innovativo, basato su un sistema ad agenti, implementato poi all'interno di un progetto europeo denominato AGENT (Automated GEneralization New Technology) (AGENT, 2003). Secondo gli autori, il vantaggio principale offerto da un sistema ad agenti è quello di consentire la manipolazione degli oggetti di una mappa a diversi livelli di dettaglio. Ad ogni agente viene affidato un oggetto della mappa (micro-agent), un gruppo di oggetti (meso-agent) o l'intera mappa (macro-agent) con il compito di soddisfare i vincoli di generalizzazione imposti, utilizzando le proprie funzionalità. Gli agenti interagiscono fino a quando non viene trovata una soluzione capace di soddisfare tutti i vincoli, o se questo non dovesse essere possibile, fino a quando non viene soddisfatta la maggior parte di essi.

La letteratura scientifica recente si è occupata anche di studiare i vincoli che si presentano nel passaggio da una scala grande ad una piccola. Di questi vincoli e degli operatori di generalizzazione, strumenti con i quali soddisfare i vincoli, si discuterà nel paragrafo seguente.

1.9 Gli operatori della generalizzazione

Quando si deve generalizzare una carta ci sono numerosi vincoli da soddisfare. Il progetto, infatti, ha delle specifiche che devono essere rispettate nella soluzione del problema di generalizzazione. Un vincolo è una specifica di progetto che deve essere rispettata nella soluzione di un problema di generalizzazione. Tra i più importanti vincoli si possono citare i seguenti:

1. *vincoli di tipo grafico*, legati a parametri come la dimensione, la larghezza e la distanza
2. *vincoli topologici*, legati a connessione, adiacenza e inclusione tra elementi;
3. *vincoli spaziali*, legati alla conservazione delle forme e degli allineamenti;
4. *vincoli semantici*, legati alle relazioni logiche che intercorrono tra gli oggetti e i loro insiemi;
5. *vincoli di disegno*, legati all'aspetto grafico della carta

Allo scopo di soddisfare questi vincoli, una grande varietà di soluzioni sono state studiate nel corso degli anni. Queste soluzioni si chiamano operatori di generalizzazione.

Un operatore di generalizzazione rappresenta un tipo di trasformazione spaziale che si vuole ottenere (Weibel and Dutton, 1999). Gli operatori sono stati sviluppati sia emulando le pratiche manuali dei cartografi, sia tramite studi puramente matematici. Ogni operatore, comunque, prende vita mediante un algoritmo di generalizzazione che lo implementa.

Diverse classificazioni sono state proposte per dare un ordine logico a questi operatori, in modo da facilitare i cartografi nella scelta dell'operatore che più si presta a soddisfare un particolare vincolo.

Alcuni ricercatori suddividono gli operatori in base al tipo di geometria a cui fanno riferimento (ad esempio l'operatore di semplificazione è progettato per elementi lineari, mentre l'operatore di fusione opera su oggetti areali).

Un'importante classificazione è quella proposta da McMaster e Shea (1992), che distinguono gli operatori in operatori che trasformano le geometrie e operatori che trasformano gli attributi.

1.9.1 Gli operatori geometrici

Gli operatori geometrici sono operatori che agiscono sull'aspetto grafico e topologico di uno o più oggetti geografici. Una possibile classificazione degli operatori geometrici è stata prodotta all'interno del progetto AGENT (2001); in cui questi sono stati classificati in base al numero di oggetti su cui agiscono. Nella prima classe troviamo quelli che operano su un singolo elemento che sono:

1. *simplification operator* riduce la granularità dei contorni di linee e aree, in pratica producendo una versione semplificata dell'oggetto grazie all'eliminazione della ridondanza di punti;
2. *collapse operator* trasforma oggetti areali in punti o linee risolvendo la progressiva mancanza di spazio.
3. *enhancement operator* valorizza un oggetto in vari modi, ovvero ingrandendo la sua forma completa (*enlargement*) o una sua parte (*exaggeration*), addolcendo il suo contorno per migliorare l'estetica (*smoothing*), squadrando la sua geometria se si avvicina ad una forma rettangolare (*squaring*);

Nella seconda classe vi sono gli operatori che possono essere applicati indistintamente ad uno o più oggetti:

1. *selection operator* seleziona gli elementi importanti considerando lo scopo della mappa;
2. *elimination operator* elimina dalla mappa gli oggetti ritenuti non importanti o ridondanti;

3. *displacement operator* sposta di posizione un oggetto o un gruppo mantenendone inalterata la forma.

Infine, nella terza categoria ricadono gli operatori che modificano un insieme di oggetti, e che vengono definiti operatori di aggregazione:

1. *combine operator* unisce in un unico oggetto punti che precedentemente erano separati e distinti; è identificato come un operatore 0-dimensionale, in quanto agisce su geometrie zero-dimensionali;
2. *merging operator* fonde 2 o più linee in un'unica linea, che normalmente viene posizionata a mezz'ora; è un operatore 1-dimensionale, in quanto agisce su geometrie lineari;
3. *amalgamation operator* fonde in un'unica geometria areale un gruppo di poligoni; è un operatore 2-dimensionale, in quanto agisce su geometrie areali;
4. *typification operator* riduce la complessità di un gruppo di oggetti attraverso la loro eliminazione, riposizionamento, allargamento o aggregazione mantenendo la disposizione tipica di quell'insieme di oggetti.

1.9.2 Operatori per la trasformazione di attributi

Gli operatori di trasformazione degli attributi si differenziano da quelli geometrici in quanto non vanno a manipolare l'informazione geometrica dell'oggetto, ma agiscono sulla sua componente statistica. Ad esempio, una piantagione di latifoglie e conifere potrebbe essere trasformata in una foresta, a causa di una riduzione di scala che mette in secondo piano le caratteristiche degli alberi.

Gli operatori di questo tipo, identificati sempre da McMaster e Shea (1992), sono:

1. *classification operator* riduce i dati grezzi in un insieme di classi, operazione spesso necessaria in quanto non è sempre praticabile associare ad ogni singolo fenomeno un simbolo diverso per successivamente mapparli;

2.symbolization operator codifica graficamente i dati e può essere applicato sia alla componente statistica sia a quella geografica.

1.9.3 Operatori e algoritmi

Gli operatori appena esaminati descrivono, di fatto, le singole operazioni che il cartografo compie per generalizzare la mappa. Ad ogni operatore, poi, corrisponde un algoritmo che implementa una trasformazione del dato geografico, anche se è facile imbattersi in più implementazioni dello stesso operatore, a seconda dell'oggetto su cui agisce. Ad esempio, l'operatore di semplificazione viene realizzato in modo diverso quando si applica ad una strada rispetto al caso in cui si applichi ad un edificio.

Ogni algoritmo è caratterizzato da una serie di parametri, che vanno impostati in funzione della scala, dell'oggetto a cui fanno riferimento e, in generale del contesto. Lo stesso algoritmo di semplificazione, così, potrebbe essere applicato con parametri differenti a seconda che l'oggetto si trovi all'interno di un centro abitato, dove troppi dettagli tendono a creare confusione, oppure in una zona scarsamente abitata, dove non ci sono questo tipo di complicazioni. Il primo problema da affrontare è, quindi, la scelta dei parametri degli algoritmi e l'opportunità di variarli in funzione del contesto.

Il secondo problema, non meno importante, è l'ordine con cui gli algoritmi vengono eseguiti: il cartografo, per decidere la sequenza delle azioni da compiere e la loro coerenza con lo scopo della mappa o con il contesto, utilizza la sua conoscenza cartografica e la sua capacità di vedere sotto diverse astrazioni la realtà rappresentata.

Gli algoritmi che implementano gli operatori di generalizzazione sono, infatti, potenti strumenti per risolvere i vincoli spaziali, semantici e grafici; essi, però, non vanno utilizzati indiscriminatamente sull'intera area da generalizzare o sempre secondo la stessa sequenza statica, ma vanno scelti, settati e applicati ogni volta diversamente, in funzione del contesto, della scala e dello scopo della mappa.

Capitolo 2

Il progetto di ricerca Cargen

Il progetto CARGEN, il cui acronimo significa CARtographic GENeralization, è un progetto di ricerca nato nel 2006, che vede coinvolti il Dipartimento di Ingegneria Informatica dell'Università di Padova e la Regione Veneto, con la collaborazione dell'Istituto Geografico Militare.

2.1 La situazione cartografica in Italia

In Italia, l'ente che si occupa di mantenere una cartografia aggiornata a livello nazionale è l'Istituto Geografico Militare (IGM). La prima produzione cartografica dell'istituto è stata la *Nuova Carta Topografica d'Italia*, realizzata alla scala 1:100 000.

L'IGM ha prodotto diverse carte che coprono il territorio a livello nazionale e solo una parte di queste vengono mantenute aggiornate.

Tra le varie serie sussiste un preciso rapporto matematico: ogni foglio di una mappa in scala 1:100 000 è diviso in quattro settori, rappresentati in altrettante mappe della serie 1:50 000. Queste mappe vengono dette “quadranti” e, ogni quadrante è a sua volta diviso in quattro parti, le “tavole”, che sono in scala 1:25 000. Particolarmente importante tra le serie in scala 1:25000, è la serie 25DB, che prende il posto della serie 25 e che sancisce il passaggio alla cartografia digitale. Su un totale di 2298 sezioni, questa serie ne copre solamente 68, ottenute tramite stereorestituzione numerica o come derivazione dalla cartografia tecnica regionale numerica.

Il lavoro dell'IGM è focalizzato sulla stesura della cartografia in piccola scala (1:25000 e superiore). Tra gli organi cartografici dello stato nominati con la legge n. 68 del 2 Febbraio 1960 "Norme sulla cartografia ufficiale dello Stato e sulla

disciplina della produzione e dei rilevamenti terrestri e idrografici", c'è il Catasto che si occupa della produzione delle carte alla grande e media scala (1:10000 e inferiore). Dal 1977 in poi, grazie al DPR n. 616 del 24 luglio 1977, è avvenuto il "Trasferimento funzioni alle regioni in materia di ambiente e territorio", per cui le regioni possono gestire in modo autonomo la creazione delle carte regionali, definite tecniche perché sono specifiche per i tecnici delle amministrazioni.

Le carte realizzate dalle Regioni sono carte ricche di particolari e vengono aggiornate frequentemente. L'operazione di aggiornamento è facilitata dalla ristretta porzione del territorio nazionale che viene rappresentata. La CTR costituisce la base di riferimento per la redazione degli strumenti urbanistici comunali, per i Piani di Coordinamento Provinciali, per i Piani d'Area e per i vari piani di settore della pianificazione e della programmazione regionale.

Recentemente anche gli uffici cartografici regionali hanno intrapreso la strada della digitalizzazione cartografica, attraverso la realizzazione dei database territoriali (DBT). Questo fatto, in concomitanza con il passaggio, da parte dell'IGM, dalla serie 25 alla 25DB, ha aperto un nuovo scenario nella cartografia italiana: la possibilità di usare la derivazione come mezzo per produrre la nuova serie 25DB.

In uno scenario del genere verrebbero limitate le rilevazioni dei dati sul territorio, e accelerati i tempi di produzione delle sezioni della serie 25DB. Inoltre, la realizzazione di una procedura informatica per la generalizzazione cartografica permetterebbe di ridurre ulteriormente i tempi, dando quindi la possibilità di automatizzare il processo di derivazione.

Lo scopo del progetto CARGEN è proprio questo, ovvero sviluppare un processo di generalizzazione delle carte tecniche regionali per produrre una base di dati coerente con il modello 25DB dell'IGM.

Quanto appena detto rappresenta la molla che ha fatto scattare il progetto Cargen.

2.2 Il progetto CARGEN

Il progetto CARGEN, nato da una convenzione tra la Regione Veneto ed il Dipartimento di Ingegneria dell'Informazione dell'Università di Padova, ha come obiettivo la progettazione e la sperimentazione di metodologie informatiche per la derivazione del database in scala 1:25000 dell'IGM a partire dal database in scala 1:5000 della Regione del Veneto.

Gli evidenti vantaggi (De Gennaro, 2009) che si possono ottenere dalla derivazione di cartografi a scala minore da quella a scala maggiore hanno comportato che il progetto CARGEN, pur essendo tra i primi in Italia, si inserisca tra i tanti che in ambito internazionale indagano il tema della generalizzazione automatica; si tratta di un'iniziativa che riporta l'Italia alla pari di altri stati europei, come Francia, Spagna, Inghilterra, dove già da tempo si stanno sperimentando soluzioni automatizzate per la produzione di carte a medio e piccola scala attraverso generalizzazione.

Il DB25 derivato dal DB5 della Carta Tecnica Regionale, nel rispetto delle indicazioni emerse dalla proficua attività di sperimentazione, si configura quindi non solo come un importante prototipo scientifico ma acquisisce la valenza di "dato territoriale istituzionalizzato" certificato da un'architettura di relazioni, in ambito pubblico e formalizzato secondo le nuove normative nazionali e le direttive europee.

Di seguito vengono descritti i tre modelli dati adoperati nel progetto.

2.2.1 La carta tecnica regionale numerica

La Carta Tecnica Regionale Numerica (CTRN) è una cartografia generale e metrica, in formato vettoriale, prodotta dalla Regione Veneto. La carta, la cui produzione trova la sua principale fonte di dati nel rilievo fotogrammetrico, gode di campagne d'aggiornamento piuttosto frequenti ed offre quindi un dato geografico piuttosto recente e di buona qualità.

Le scale di rappresentazione adottate sono la scala 1:5 000 per la quasi totalità del territorio regionale e la scala 1:10 000 per le zone montane scarsamente

urbanizzate. Gli oggetti e le informazioni territoriali contenuti nella Carta Tecnica Regionale, acquisiti in forma vettoriale, sono organizzati in Livelli e Codici: i Livelli costituiscono una primaria classe di aggregazione degli oggetti, che a loro volta sono suddivisi nei Codici, relativi alle caratteristiche particolari di ciascun oggetto. In totale sono presenti 16 Livelli principali, 12 Livelli di servizio e 6 Livelli funzionali per la gestione informatica dei grafi (assi e nodi di viabilità, idrografia e ferrovia); ciò consente la codifica di 480 oggetti ed informazioni.

I dati della CTRN, però, non si prestano bene all'analisi spaziale e ad un diretto utilizzo, in quanto sono realizzati prevalentemente tramite tecniche CAD, e perciò non offrono alcuna forma di controllo di coerenza topologica. Questo fatto si ripercuote nella necessità di attuare una lunga fase di controllo e pulizia dei dati.

2.2.2 Il GeoDBR

Il GeoDBR è un modello dati sviluppato dalla Regione Veneto nell'ambito di un progetto per l'aggiornamento del proprio sistema informativo territoriale.

Il GeoDBR ha una concezione molto moderna, ed è caratterizzato da una rappresentazione che prevede una divisione dell'informazione geografica in informazione di base e strati tematici, e la gestione di geometrie tridimensionali. Gli oggetti topografici sono rappresentati nel GeoDBR da *feature* inserite in un'organizzazione di livelli informativi a strati. Così, risulta presente una forte componente gerarchica nella strutturazione del modello dati: le *feature*, prima che negli strati informativi, sono raggruppate in *feature class* e distinte tra loro tramite attributi.

2.2.3 Il DB25

Il DB25 è un modello dati creato dall'IGM per la compilazione della cartografia della serie topografica 25DB. Esso contiene un modello di rappresentazione del mondo reale basato sulle *feature*, suddivise in oggetti semplici identificati dall'attributo LAB (Label). Ognuna di queste, è caratterizzata da una tipologia di

primitiva grafica, evidenziata dalla prima lettera del codice: C per la tipologia areale, L per quella lineare, P per quella puntuale e T per quelle testuali.

Le *feature* che popolano il DB25 si riferiscono a 291 particolari topografici (susceptibili di restituzione, di eventuale ricognizione e/o di acquisizione da Banche Dati di Enti Pubblici e Privati) e a 48 tipologie di testi, per la maggior parte legati a particolari topografici.

Le *feature* in cui gli oggetti sono inquadrati derivano la loro codifica e definizione dagli oggetti descritti nello standard DIGEST, Digital Geographic Information Exchange Standard, usato in applicazioni militari, in ambito NATO, e sviluppato dal gruppo di lavoro internazionale DGIWG (Digital Geographic Information Working Group). Ogni *feature* del DB25 è definita da un codice LAB, un nome, e un codice FACC (Feature and Attribute Coding Catalogue).

2.2.4 La derivazione del DB25 e del DB50

La parte del progetto CARGEN dedicata alla derivazione del DB25, condotta nei primi anni del progetto, è stata caratterizzata dalle seguenti tre fasi:

1. creazione del GeoDBR e suo popolamento con i dati provenienti dalla CTRN in scala 1:5.000;
2. analisi delle differenze tra i modelli di dati proprietari del GeoDBR e del DB25;
3. la creazione del prototipo DB25.

La prima fase, già realizzata, consiste nel completamento delle specifiche del GeoDBR, nella creazione delle tabelle all'interno del DBMS Oracle Spatial 10g e nel loro popolamento.

La migrazione dei dati dal modello della CTRN a quello del GeoDBR si è scontrata in primo luogo con la diversa rappresentazione delle geometrie usate nei due modelli e, in secondo luogo con gli errori topologici dei dati originali. All'interno del progetto sono stati perciò sviluppati algoritmi per la trasformazione delle geometrie e per il controllo topologico, in modo da migliorare la qualità dei dati facilitando la successiva fase di derivazione del DB25.

La seconda fase, che ha come obiettivo quello di trovare una corrispondenza tra le *feature* del GeoDBR e quelle del DB25, ha permesso di evidenziare alcune discrepanze e di conseguenza suggerire alcune migliorie ai due modelli.

La terza fase, consistente nella creazione del prototipo DB25, ha visto l'implementazione di un processo di generalizzazione cartografica, allo scopo di soddisfare i vincoli dettati dal passaggio di scala. Il processo di derivazione (Savino, 2007) è stato implementato in otto passi, utilizzando il DBMS Oracle per il mantenimento dei dati, Geomedia Professional 6 per l'elaborazione e l'export dei dati, e Dynamo/Dynagen per la fase di ricostruzione, acquisizione e generalizzazione dei dati.

Gli elementi geografici possono essere divisi in due classi:

- la prima, costituita da tutte le classi per cui la generalizzazione può essere ottenuta tramite le sole operazioni di selezione geometrica e/o spaziale descritte nel modello dati;
- la seconda, costituita da classi complesse che richiedono una generalizzazione più specifica e accurata, come la viabilità stradale, ferroviaria, l'idrografia e l'edificato.

Tramite la procedura di popolamento si possono quindi generalizzare gran parte delle classi del modello dati, mentre, per le restanti, è richiesto lo sviluppo di algoritmi appositi per rendere queste classi complesse adatte alla rappresentazione in scala 1: 25 000 e compatibili con l'applicazione delle specifiche del modello dati. Questo fatto ha reso necessario lo sviluppo di un certo numero di algoritmi parametrici, scritti in Java, per risolvere alcuni importanti problemi di generalizzazione.

Gli algoritmi sono stati inseriti in un processo di tipo batch, ovvero un processo che li esegue in ordine sequenziale e predefinito. Il risultato finale del processo dipende perciò solo esclusivamente dai parametri scelti per i singoli algoritmi.

I risultati della generalizzazione del prototipo DB25 sono stati presentati in un convegno nazionale tenutosi presso l'Università di Padova nel 2009 (De Gennaro,

2009). A partire da questa data, gli obiettivi del progetto sono stati estesi alla derivazione del DB50.

La seconda fase del progetto CARGEN, quella relativa alla derivazione del DB50, ha avuto come primo obiettivo quello di realizzare un prototipo di modello dati per il DB50. Per quest'ultimo non è mai stato definito un modello dall'IGM. Nel febbraio 2010 è stato redatto un documento contenente l'analisi della derivabilità della Carta d'Italia IGM in scala 1:50 000, a partire dal database topografico IGM DB25 in scala 1:25 000, fornendo una proposta di modello dati per un database topografico alla scala 1:50 000.

2.3 Strumenti e tecniche

Inizialmente, il modello adottato all'interno del progetto CARGEN per la gestione dei dati è stato quello client/server.

Il server è costituito da una macchina con installato un DBMS Oracle Spatial 10g, la cui funzione è quella di memorizzare e mantenere i dati spaziali, accessibili tramite query. Nel lato client, invece, sono stati installati i software Geomedia Professional 6 e Dynamo/Dynagen, entrambi di proprietà della Intergraph. Geomedia è utilizzato principalmente come strumento d'accesso ai dati spaziali; Dynamo/Dynagen sono, invece, gli strumenti usati durante il processo di generalizzazione cartografica.

Questa architettura, tuttavia, è stata in parte abbandonata recentemente. Infatti, sia allo scopo di semplificare lo sviluppo di nuovi algoritmi, sia allo scopo di migliorare le prestazioni temporali, è stato cambiato metodo d'accesso ai dati ed il software per visualizzarli. I dati vengono caricati in RAM e gestiti proprio come se fossero delle tabelle, grazie ad una libreria sviluppata all'interno del Progetto. Inoltre, si è fatto ricorso ad una libreria potentissima sviluppata in Java, la JTS (JTS Topology Suite), che fornisce una vasta serie di operatori spaziali, evitando così di ricorrere al DBMS di Oracle per effettuare le interrogazioni spaziali.

Nei successivi sottoparagrafi saranno brevemente descritti i software e le tecniche più importanti usati durante l'attività di tesi e i dati geografici su cui sono stati effettuati i relativi test.

2.3.1 JTS Topology Suite

La JTS Topology Suite è una libreria open source, scritta interamente in Java, che fornisce una modellazione ad oggetti per le geometrie lineari in uno spazio euclideo.

In questa libreria sono definite tre geometrie fondamentali, *Point*, *LineString* e *Polygon*, che rappresentano rispettivamente la geometria puntuale, lineare e areale.

La JTS mette a disposizione numerose funzioni geometriche, tra le quali è importante citare:

1. operatori topologici che realizzano le funzioni di intersezione, differenza, unione;
2. creazione di buffer (sia positivo, che negativo) intorno alle geometrie;
3. costruzione dell'involuppo convesso di geometrie;
4. semplificazione delle geometrie, ad esempio mediante l'algoritmo di Douglas-Peucker;
5. costruzione del Minimum Bounding Box,
6. triangolazione.

Oltre a queste funzioni, la JTS fornisce l'implementazione di indici spaziali, come il quadTree ed il KdTree, che offrono un modo veloce per la risoluzione di query spaziali.

La versione utilizzata nell'ambito di questa tesi è la JTS 1.11.

2.3.2 OpenJump

OpenJump10 è un Desktop GIS open source che permette di visualizzare, modificare e interrogare dati spaziali.

Essenzialmente esso è scritto in Java, si basa sulla JTS ed è in grado di gestire file raster, vettoriali e database (PostGis, Oracle, ArcSDE). Una caratteristica peculiare è la sua architettura modulare, che permette di estendere di molto le funzionalità di base, potendo integrare, per esempio, il proprio codice mediante la realizzazione di uno o più plugin.

In OpenJump, la creazione di un plugin, relativo al proprio codice, offre al programmatore il grosso vantaggio di poter visionare tramite l'interfaccia grafica gli effetti della propria applicazione. Il plugin diventa così, uno strumento essenziale nello sviluppo di nuovi algoritmi che manipolano geometrie e permette di avere una risposta grafica e istantanea, garantendo un grande aiuto nella fase di testing e debug.

La modalità con cui OpenJump gestisce le *feature* si basa sull'utilizzo dei *layer* (livelli) che svolgono il ruolo di contenitori. Ogni *layer* è in grado di contenere le *feature* relative ad uno specifico schema dati, chiamato *FeatureSchema*. Quest'ultimo, specifica il nome e la tipologia degli attributi che costituiscono la *feature*, similmente a quanto accade nelle tabelle dei database. Un *layer* rappresenta, quindi, una vera e propria tabella, il cui schema dati è specificato dal *FeatureSchema*. Il layer è interrogabile per mezzo di query, che possono essere sia spaziali che non spaziali. Per migliorare le query spaziali, è possibile associare al *layer* uno degli indici spaziali forniti dalla JTS.

La versione di OpenJump utilizzata nello svolgimento di questa tesi è la 1.3.1, rilasciata nell'aprile 2009.

2.3.3 Triangolazione di Delaunay

Una tecnica frequentemente utilizzata durante lo svolgimento di questa tesi è la triangolazione di Delaunay, che è una particolare suddivisione poligonale del piano.

A tale proposito, occorre ricordare che una suddivisione poligonale del piano è un insieme P (infinito) di poligoni convessi che soddisfano le seguenti proprietà:

1. Dati due poligoni qualsiasi di P , se la loro intersezione è non nulla allora è uguale ad un lato per entrambi, ovvero i poligoni si toccano al più, lungo un lato in comune;
2. L'unione di tutti i poligoni coincide con il piano, quindi non ci sono buchi.

Nella poligonazione si usano spesso i triangoli come poligoni (triangolazione), perché il triangolo ha il vantaggio di essere il poligono convesso più semplice. Inoltre, qualsiasi suddivisione poligonale è sempre triangolabile aggiungendo opportunamente dei lati (ciò deriva dal fatto di poter sempre decomporre un poligono convesso in un numero finito di triangoli).

Esistono diversi modi per effettuare la triangolazione. Quello tradizionalmente più usato è quello di Delaunay, definito così come segue:

- dato un insieme di N punti sul piano e T una triangolazione, $t \in T$ è un triangolo di Delaunay se soddisfa la condizione che il cerchio circoscritto al triangolo T non contiene altri punti di N .

Se la condizione è soddisfatta da tutti i triangoli, la triangolazione è detta di Delaunay, ed è un tipo di triangolazione che genera triangoli il più equiangoli possibile, il che è favorevole per l'interpolazione e genera ogni volta, in dipendenza dagli algoritmi scelti, lo stesso tipo di triangolazione.

Pur essendo implementata anche all'interno della Jts, si è scelto di usare l'algoritmo di triangolazione realizzato da Michel Michaud e messo a disposizione dal Dottor Sandro Savino, responsabile del progetto Cargen presso il laboratorio GIRTS (Geographic Information Systems and Real-Time Systems) del Dipartimento di Ingegneria dell'informazione dell'Università di Padova.

Tale scelta è scaturita dal fatto che tale algoritmo risulta ottimale per le esigenze del progetto Cargen.

Nelle classi java in cui è implementata la triangolazione di Delaunay, è disponibile anche una vasta gamma di metodi che permettono di trattare e gestire i triangoli ottenuti in fase di triangolazione.

2.3.4 Cartografia utilizzata per i test

Il progetto CARGEN ha come scopo dichiarato quello di realizzare il prototipo del DB25 IGM del Parco delle Dolomiti Bellunesi.

Tuttavia, oltre alla cartografia di questa zona sono state utilizzate anche ulteriori carte per il test, quali ad esempio la cartografia di Monselice o anche quella di Venezia. La cartografia di Monselice è una carta altamente densa di boschi e di strade che si trovano sia in prossimità dei boschi, che all'interno dei boschi stessi. Quindi essendo il trattamento delle adiacenze tra elementi areali ed elementi lineari uno dei temi cardine di questa tesi, ed essendo i boschi e le strade due classi di elementi che ricadono a pennello in questo contesto, si è deciso di selezionare questa zona. Oltre a ciò, è importante sottolineare che diverse parti del progetto Cargen sono state sviluppate utilizzando questo zona come dataset su cui effettuare i test.

L'area di Monselice consiste in un territorio di circa 150 chilometri quadrati, contenente 16253 segmenti stradali e 4000 elementi boschivi.

Oltre alla cartografia di Monselice, come detto precedentemente, è stata utilizzata anche la cartografia di Venezia, in quanto particolarmente densa di elementi lineari quali ad esempio gli elementi divisorii, che rappresentano l'oggetto di studio della seconda parte (elementi lineari) e della terza parte (elementi divisorii) di questa tesi.

I boschi in queste cartografie, sono classificati secondo il modello dati definito per il DBT (database territoriale), dove l'attributo LIVCOD specifica il tipo di strada:

CODICI DELLA VEGETAZIONE	
LIVCOD=0503 Limite di bosco (area)	LIVCOD=0505 Limite di bosco (punto)
LIVCOD=0506 Albero isolato	LIVCOD=0507 Filare di alberi
LIVCOD=0508 Limite di coltura arborea	LIVCOD=0509 Limite di vigneto
LIVCOD=0510 Filare di vite	LIVCOD=0511 Filare di vite con alberi
LIVCOD=0512 Delimitazione di vivaio	LIVCOD=0515 Serre stabili
LIVCOD=0516 Simbolo essenza di cedri	LIVCOD=0517 Simbolo essenza di abeti
LIVCOD=0518 Simbolo frutteto	LIVCOD=0519 Simbolo essenza di pini
LIVCOD=0520 Simbolo essenza di cipressi	LIVCOD=0521 Simbolo essenza di querce e/o olmi
LIVCOD=0522 Simbolo essenza di faggi	LIVCOD=0523 Simbolo essenza di castagni
LIVCOD=0524 Simbolo essenza di larici	LIVCOD=0525 Simbolo essenza di pioppi
LIVCOD=0526 Simbolo essenza di olivi	LIVCOD=0527 Simbolo essenza di lecci e querce da sughero
LIVCOD=0528 Seminativi e orti	LIVCOD=0529 Canneto
LIVCOD=0530 Prato	LIVCOD=0533 Incolto - macchia

Tabella 2.1: codifica della vegetazione

Gli elementi divisorii, invece, sono classificati secondo il seguente modello dati definito per la CTRN:

CODICI DEGLI ELEMENTI DIVISORI
LIVCOD=0301 Muri
LIVCOD=0302 Recinzioni – palizzate – filo spinato – staccionate
LIVCOD=0303 Siepi
LIVCOD=0304 Muri a secco
LIVCOD=0305 Bastioni – mura di città.

Tabella 2.3: codifica degli elementi divisorii

Le strade sono classificate secondo il seguente modello dati:

CODICI DELLA VIABILITÀ STRADALE	
LIVCOD=A201 Asse autostrada o assimilate	LIVCOD=A202 Asse autostrada o assimilate in costruzione
LIVCOD=A206 Asse strada secondaria	LIVCOD=A206 Asse strada secondaria
LIVCOD=A207 Asse strada carrozzabile	LIVCOD=A208 Asse strada carreggiabile
LIVCOD=A209 Asse strada in costruzione	LIVCOD=A210 Asse strada di campagna
LIVCOD=A213 Asse ponte	LIVCOD=A216 Asse guado
LIVCOD=A217 Asse sentiero facile	LIVCOD=A218 Asse sentiero difficile
LIVCOD=A219 Mulattiera	LIVCOD=A220 Passerella pedonale
LIVCOD=A221 Asse cavalcavia	LIVCOD=A222 Sottopassaggio stradale
LIVCOD=A223 Viadotto autostradale	LIVCOD=A224 Viadotto stradale
LIVCOD=A226 Tracciato galleria stradale	LIVCOD=A226A Tracciato galleria stradale
LIVCOD=A227 Asse ferrata	LIVCOD=A235 Asse pista ciclabile

Tabella 2.2: codifica della viabilità stradale

Capitolo 3

Trattamento adiacenze elementi areali ed elementi lineari

In questo capitolo viene affrontato il tema del trattamento delle adiacenze tra elementi lineari ed elementi areali. L'obiettivo finale che si vuole raggiungere è quello di estendere gli elementi areali fino a quelli lineari situati in prossimità di questi ultimi.

Le classi alle quali si applica questo problema all'interno di questa tesi, come già detto precedentemente, sono i boschi e le strade. Inoltre, occorre specificare che la nascita di questo problema deriva direttamente dalle specifiche IGM, le quali impongono tra i criteri per la derivazione del DB_25 e della relativa cartografia in scala 1:25000, partendo dal database territoriale, l'estensione di elementi areali quali ad esempio boschi, colture agricole e quant'altro, fino ad elementi lineari quali strade, recinzioni e scarpate.

Prima di addentrarsi nel procedimento di generalizzazione occorre fare una precisazione: i boschi pur essendo degli elementi areali veri e propri, vengono trattati alla stregua di elementi lineari, in quanto di essi viene considerato solo ed esclusivamente il perimetro.

3.1 Semplificazione iniziale delle geometrie

Il primo passo della generalizzazione ha l'obiettivo di ottenere degli elementi areali (boschi) la cui forma risulti più semplice possibile e che siano facili da trattare e gestire. Per perseguire questo obiettivo, occorre ridurre il numero di punti di ciascuna geometria, in modo tale da avere degli elementi più semplici da processare nelle operazioni a seguire, ottenendo tempi di computazione accettabili anche in presenza di dataset grandi e complessi.

Per effettuare questa semplificazione, si è usato l'algoritmo di Douglas-Peucker, che prende il nome dagli omonimi autori che lo misero a punto nel 1973 e che risulta essere ancora oggi, uno dei migliori algoritmi di semplificazione conosciuti. Questo algoritmo effettua una semplificazione riducendo il numero dei vertici in eccesso.

Tuttavia, in fase di implementazione del programma, si è deciso di rendere opzionale questo passaggio, mediante l'ausilio di una checkbox.

Quindi ricapitolando, la semplificazione dei boschi viene eseguita a discrezione dell'utente, cioè a seconda del fatto che quest'ultimo selezioni o meno la checkbox associata all'esecuzione di quel determinato algoritmo.

Anche per quanto riguarda gli elementi lineari (strade) è stata effettuata una specie di scrematura iniziale: in pratica non è stato fatto altro che analizzare ciascun record all'interno della tabella contenente gli elementi lineari (layer strade), andando ad eliminare ciascun record che nel campo GEOMETRY conteneva un elemento vuoto.

3.2 Casistiche individuate

Una volta effettuata la semplificazione iniziale dei dati in ingresso, ci si è concentrati su un'analisi morfologica dei dati, che ha messo in evidenza due principali e differenti casistiche da risolvere:

1. **caso B** in cui si ha un elemento lineare che è fiancheggiato da un singolo lato da un elemento areale;
2. **caso A** in cui si ha un elemento lineare fiancheggiato da ambo i lati da elementi areali della stessa classe e che, come mostrato in seguito, sarà molto più semplice ed immediato da risolvere rispetto all'altro caso.

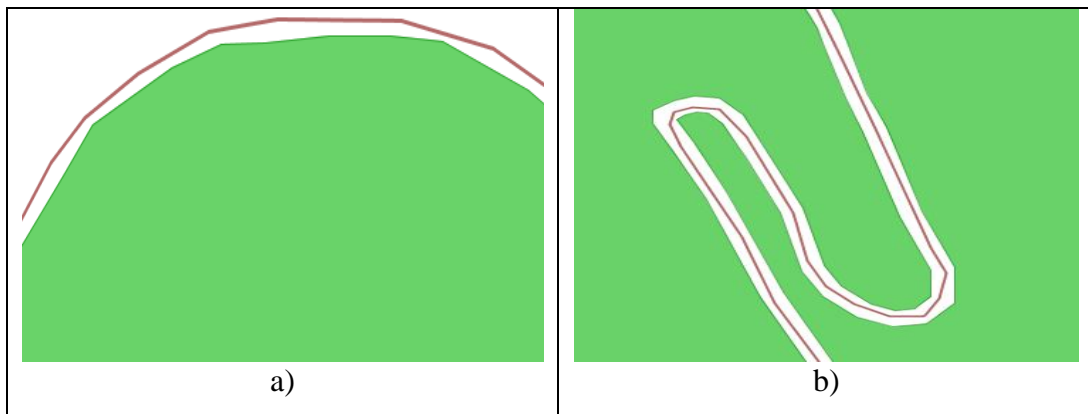


Figura 3.1: a) caso A, elemento lineare fiancheggiato su singolo lato b) caso B, elemento lineare fiancheggiato su ambo i lati

Per comodità di linguaggio e perché in fase di test si sono utilizzate queste due particolari categorie di classi, da adesso in avanti si userà il termine boschi per riferirsi agli elementi areali, mentre, si userà il termine strade per riferirsi agli elementi lineari.

3.3 Caso A

La strategia che si è adottata per la risoluzione del primo caso, vale a dire della strada immersa nel bosco e quindi fiancheggiata da quest'ultimo su ambo i lati, consiste nell'effettuare l'unione di tutti i boschi che hanno distanza tra di loro inferiore ad una certa soglia (la soglia viene inserita dall'utente come parametro di ingresso).

In questo modo, lo “spazio” esistente tra due boschi differenti, all'interno del quale come è emerso dall'analisi morfologica preliminare dei dataset di ingresso, spesso si trovano delle strade, quindi fiancheggiate dai due boschi e che in sostanza dividono un bosco in due frazioni, viene del tutto colmato.

Per perseguire questo obiettivo, la prima operazione effettuata consiste nella creazione di un buffer su ciascun bosco. E' importante sottolineare che la dimensione del buffer è un parametro inserito dall'utente.

Per avere delle geometrie più semplici e quindi composte anche dal minor numero possibile di punti, si è scelto di limitare il numero di quadrant segment nella

costruzione del buffer, cioè di ridurre il numero di punti presenti in una porzione circolare del buffer.

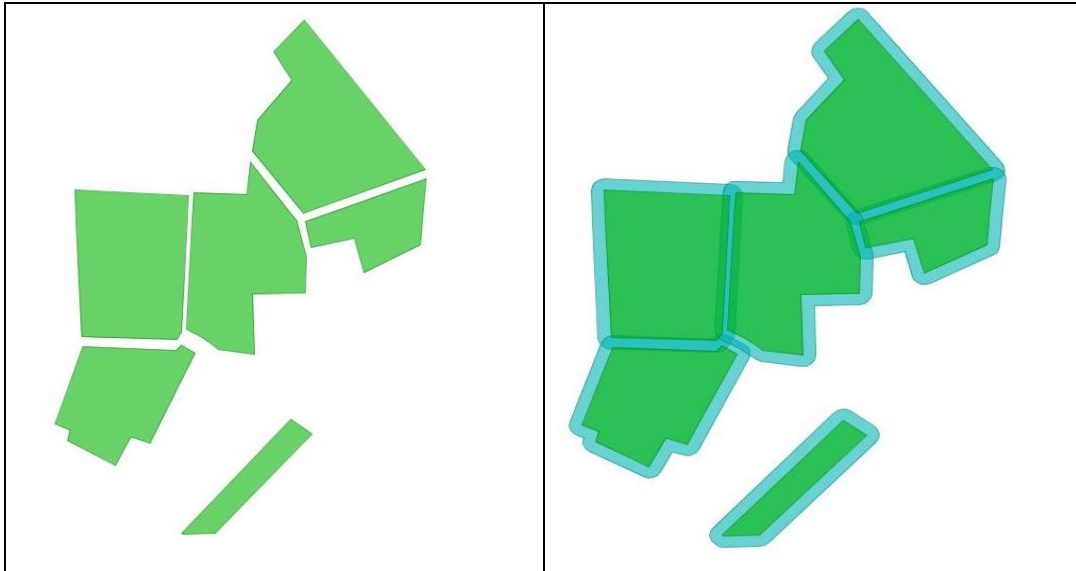


Figura 3.2: creazione del buffer positivo su ciascun bosco

Una volta espanso ciascun bosco, viene effettuato il join spaziale tra i buffer costruiti sui boschi e qui, tramite l'ausilio dei metodi `aggregateSpatialJoinTable` e `expandSpatialJoinTable`, situati all'interno del package `Tables`, si provvede ad effettuare ordinamento e aggregazione dei risultati ottenuti.

Alla fine si ottiene una tabella i cui record sono composti da due campi, in cui il primo campo contiene l'identificatore di un singolo bosco, mentre il secondo campo contiene una lista di identificatori che rappresentano tutti gli elementi che sono in relazione (si intersecano) con il bosco contenuto nel primo campo. Ovviamente, non sono ammesse ripetizioni tra primo e secondo campo e neanche tra i primi campi di ciascun record, in maniera tale da non avere record duplicati. In questo modo, è stato sufficiente scorrere i record di questa tabella e costruire un nuovo livello per ogni record, in cui ciascun record della nuova tabella è composto dall'unione tra il bosco indicato nel primo campo e quelli facenti parte della lista contenuta nel secondo campo.

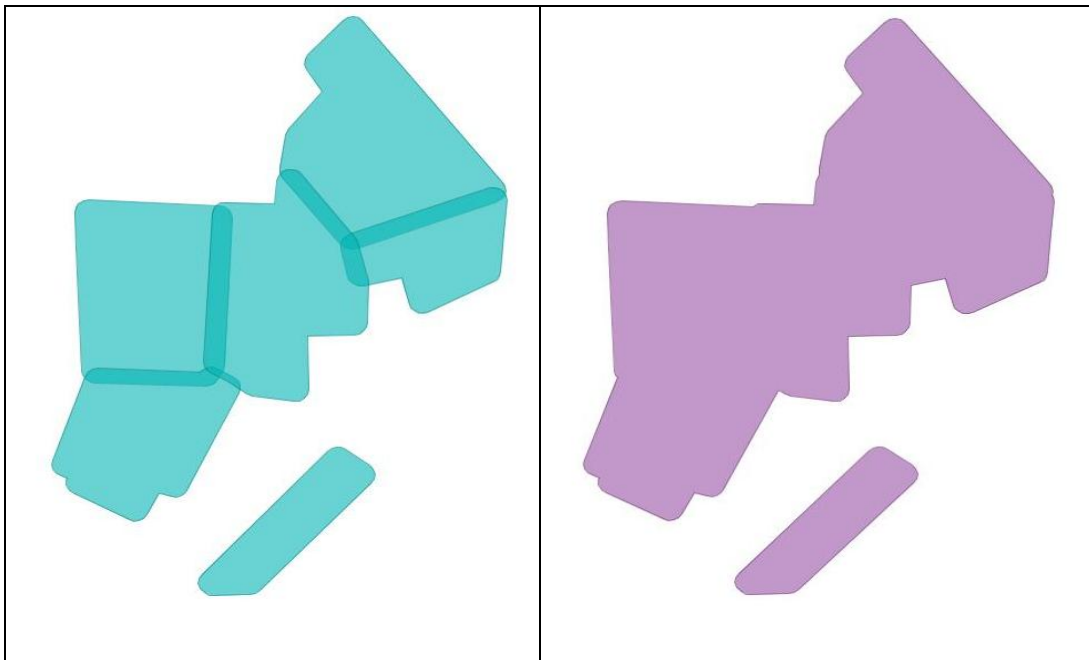


Figura 3.3: Unione dei buffer che si intersecano

Una volta fatta l'unione dei boschi, è stato sufficiente realizzare un buffer negativo sugli elementi della tabella appena realizzata, in modo tale da riportare i boschi alla dimensione originale. Anche in questo caso, nella creazione del buffer, si è deciso di ridurre al minimo il numero di punti per quadrante per rendere più snelli i dati.

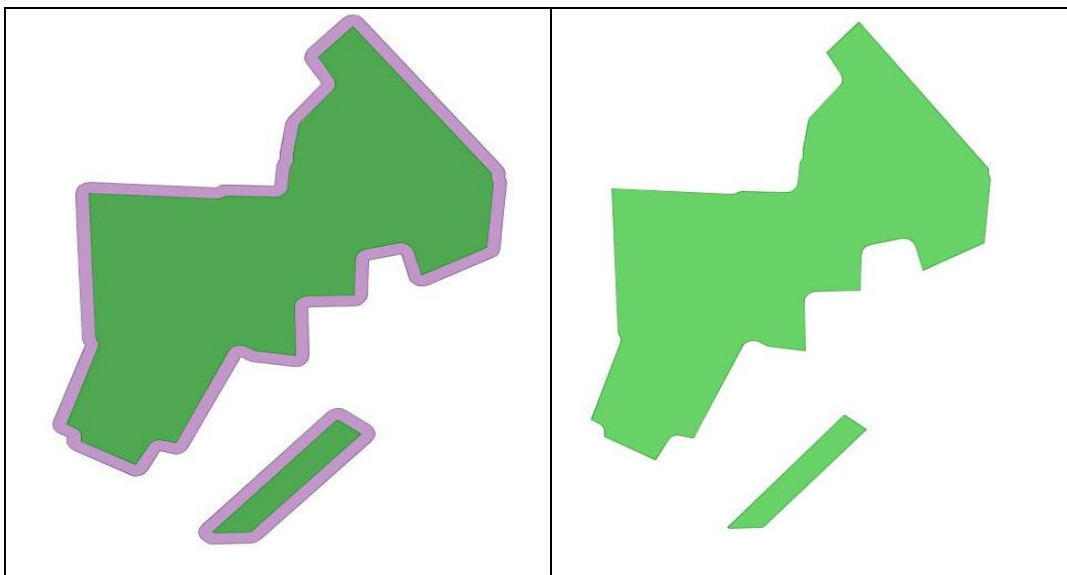


Figura 3.4: Creazione del buffer negativo

3.4 Eliminazione dei broli e delle radure

Il passo seguente consiste nell'effettuare una ulteriore operazione di pulitura e semplificazione dei boschi.

Da uno sguardo alle carte tecniche regionali (scala 1:5000) si evince che all'interno delle grandi aree boschive, sono spesso presenti numerose e piccole aree che non fanno parte dei boschi, costituite da *broli e radure*. Questi elementi costituiscono dei veri e propri “buchi” sulle mappe. Invece, analizzando attentamente le stesse aree geografiche, utilizzando però le carte dell'Istituto Geografico Militare (1:25000), si può chiaramente riscontrare che molti dei cosiddetti buchi vengono trascurati e perciò omessi a causa della loro dimensione esigua.

Oltre a quanto appena detto, bisogna evidenziare che la generazione di ulteriori buchi potrebbe essere un effetto collaterale verificatosi a seguito dell'unione dei boschi vicini e perciò, a maggior ragione, occorre eliminare questi buchi.

Detto questo, quello che si vuole realizzare in questa sezione è proprio l'eliminazione dei buchi presenti nei boschi che hanno dimensione inferiore ad una certa soglia, la quale viene decisa dall'utente e viene passata come parametro di input al programma.

Per fare ciò, è stata analizzata ogni geometria presente in ciascun record all'interno della tabella boschi, distinguendo due casistiche principali:

1. il bosco in questione era un poligono semplice (`bosco instanceof Polygon`);
2. il bosco in questione era un multipoligono (`bosco instanceof MultiPolygon`).

Nel primo caso è stato preso in considerazione il perimetro di ciascun poligono, attraverso l'ausilio del metodo `getExteriorRing()` e successivamente è stata analizzata ciascuna delle geometrie presenti al suo interno. Per accedere a queste geometrie interne si è usato il metodo `getInteriorRingN(j)`, facente parte della classe `Geometry` all'interno della `JTS`.

Fatto ciò, come detto in precedenza, si è effettuato il calcolo del perimetro di ciascun buco. Se quest'ultimo soddisfa i requisiti imposti dall'utente viene memorizzato e ad ogni iterazione si effettua l'unione di questo elemento con gli altri elementi che vengono progressivamente analizzati e selezionati, in caso contrario viene scartato. Alla fine dell'algoritmo, si provvede ad effettuare l'unione tra questa maxi geometria appena costruita, ed il perimetro del bosco.

Nel secondo caso, quello cioè dei MultiPoligoni è stato in sostanza effettuato l'algoritmo appena descritto per ciascuno dei poligoni che componevano la geometria, andando però a separare ciascuno dei poligoni analizzati nella tabella finale.



Figura 3.5: Bosco con presenza di broli

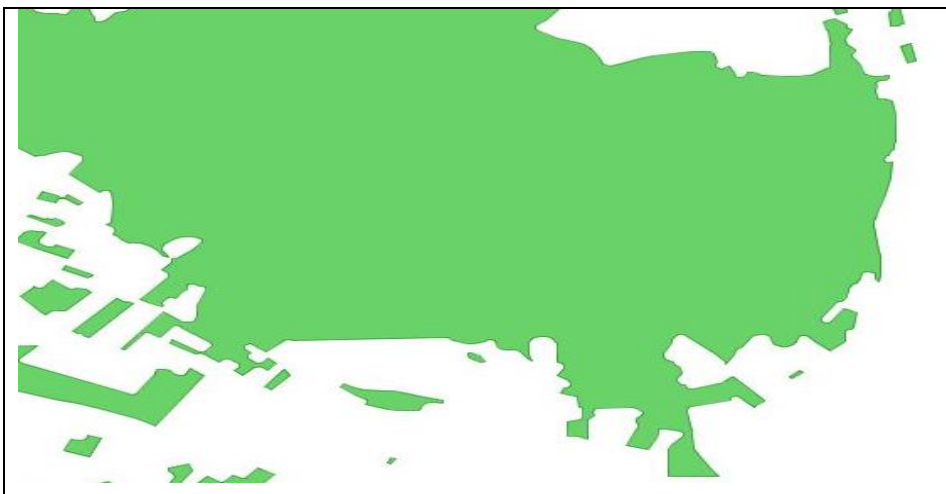


Figura 3.6: Rappresentazione dopo l'eliminazione dei broli

3.5 Caso B

Una volta effettuati l'analisi del caso A e l'eliminazione dei broli e delle radure, si passa all'analisi del caso generale, che come detto all'inizio del capitolo, è rappresentato da una strada che è fiancheggiata da un bosco su un singolo lato.

In questo caso occorre effettuare l'estensione dei boschi sino alle strade in un modo molto più artificioso rispetto a quello precedentemente descritto, basato anche su strumenti totalmente differenti quali ad esempio la triangolazione, ma non per questo meno efficienti.

Per prima cosa occorre identificare quali sono le aree da estendere, ed è a tale scopo che è stato realizzato un buffer su ciascuna strada, di dimensione uguale al parametro stabilito dall'utente.

Dopo aver realizzato il buffer, si è effettuato il join spaziale tra i buffer sulle strade ed i perimetri dei boschi, in modo tale da ottenere le zone da estendere. Al solito, come fatto in precedenza, si è ordinato i dati ottenuti dopo aver effettuato il join spaziale e successivamente, è stata effettuata l'aggregazione di questi ultimi.

Successivamente sono state memorizzate le geometrie ottenute all'interno di una nuova tabella chiamata intersezione, che oltre al campo `GEOMETRY`, è composta anche da ulteriori 3 campi:

1. `ID` costituisce un identificatore univoco per ciascun record;
2. `IDstrada` identifica la strada fino alla quale deve essere esteso il bosco;
3. `IDbosco` identifica a quale bosco appartiene il segmento in analisi.

Fatta questa operazione, viene effettuata una separazione dei dati contenuti nella tabella intersezioni. Quindi per ogni record contenente una singola linestring viene creato un nuovo record all'interno di una nuova tabella denominata `veget`, invece, se il record contiene più oggetti (è una multilinestring), viene creato un record per ogni singola linestring. Ovviamente poi tale tabella conterrà un ulteriore campo che indica qual è la strada che è situata in prossimità di quella porzione di perimetro di bosco. Questo disaccoppiamento dei risultati precedentemente ottenuti è dovuto al fatto che una stessa strada può essere vicina

ad uno stesso bosco in sezioni differenti e perciò è necessario disaccoppiare questi elementi.

Dopo avere realizzato questa separazione si crea un buffer su ciascun pezzo di bosco e si ricorre nuovamente all'operazione di intersezione spaziale, effettuando il join con ciascuna strada vicina, per trovare tra le strade precedentemente selezionate, la porzione che è realmente vicina a quella porzione di perimetro di bosco su cui è stato generato il buffer. Le geometrie ottenute a seguito di questa operazione saranno salvate in una nuova tabella denominata vie.

A questo punto, volendo fare un resoconto generale di quello che è stato fatto fino a questo momento si può dire che si sono costruite due tabelle parallele in cui:

1. il campo GEOMETRY all'interno del record i-esimo della prima tabella (veget) contiene un pezzo del perimetro di un bosco che è situato in prossimità di una strada;
2. il campo GEOMETRY dell'i-esimo record della seconda tabella (via) contiene il pezzo di strada che si trova in prossimità del pezzo di perimetro di bosco selezionato nella tabella veget, nell'i-esimo record.

L'area che si trova tra l'i-esimo oggetto geometry della prima tabella e l'i-esimo oggetto geometry della seconda tabella corrisponde all'area da estendere.

3.5.1 Triangolazione tra boschi e strade

Dopo aver spiegato lo scopo e il modo in cui sono state create le due tabelle parallele, è lecito domandarsi come si fa a selezionare e creare le geometrie finali che costituiranno l'area aggiuntiva da aggregare ai boschi.

Lo strumento principe usato per perseguire questo scopo è la triangolazione di Delaunay vincolata tra ciascuna coppia di oggetti corrispondenti strada/bosco. La triangolazione vincolata è un tipo particolare di triangolazione in cui viene imposto che ciascun triangolo costruito mediante tale algoritmo, possa avere più vertici ed anche lati che si sovrappongono alla geometria fornita in input al triangolatore, ma non è consentito in alcun modo che nessuno dei triangoli

costruiti tagli uno dei segmenti facenti parte della geometria fornita in ingresso al triangolatore.

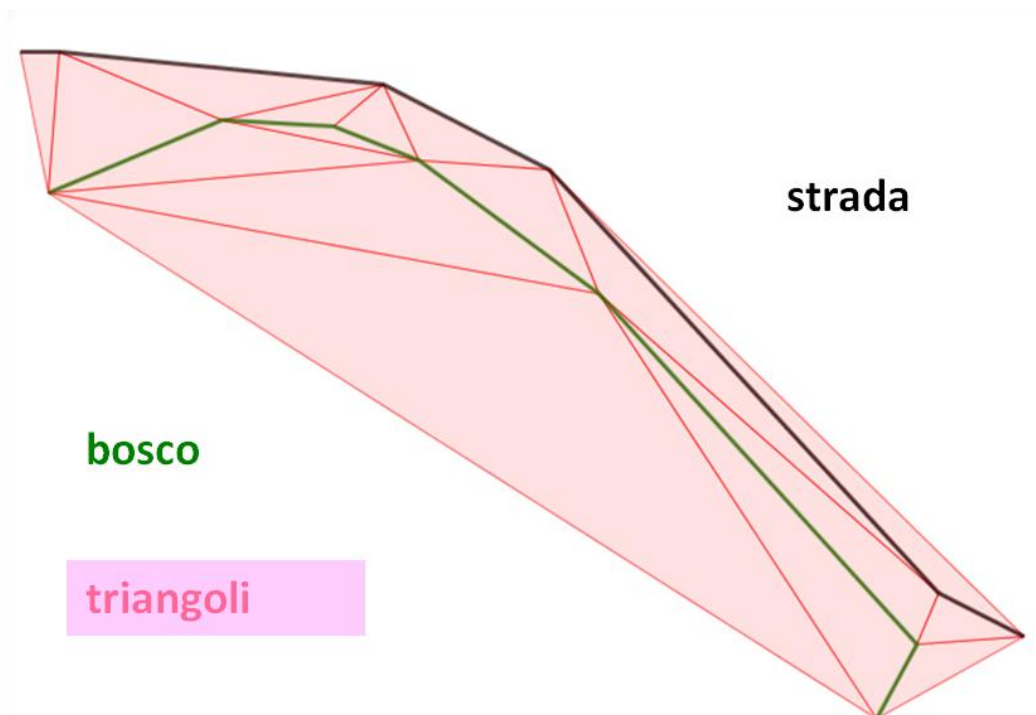


Figura 3.7: triangolazione tra un frammento di perimetro di bosco ed un pezzo di strada

Pur essendo possibile applicare altri artifici di geometria costruttiva per la costruzione delle aree, anche concettualmente più semplici ed apparentemente più veloci rispetto alla triangolazione, si è deciso di utilizzare quest'ultima perché permette di ottenere prestazioni migliori e risultati comunque robusti.

Realizzata la triangolazione e sfruttando gli opportuni metodi all'interno del package Traingulation, si memorizzano i triangoli in una nuova tabella denominata ou. Ciascun record di questa tabella è formato da diversi campi contenenti informazioni che ritorneranno utili in seguito. Si citano qui di seguito i campi, presenti in un record, usati frequentemente durante lo svolgimento di questa tesi:

1. ID rappresenta l'identificatore univoco di ciascun record;
2. VERTICES è da una terna di interi, in cui ognuno di questi numeri identifica un vertice del triangolo;

3. TRIANGLES è una terna di interi in cui ciascun numero rappresenta l'identificatore del triangolo con il quale si condivide quel lato, ossia con il quale quel lato è in comune;
4. FACES è una terna di interi in cui se uno di questi valori k è diverso da -1 significa che quel lato del triangolo è un segmento facente parte della k -esima geometria fornita in input al triangolatore, invece, in caso contrario ($k=-1$) significa che non è in alcun modo in relazione con le geometrie date in pasto al triangolatore;

Descritta la struttura dei record della tabella ou, si può passare alla descrizione dell'algoritmo di selezione dei triangoli.

```

...
if( !tr.isTriangleReal( tri )) continue;
for (int j=0; j<3; j++ )
{
    //Se ciascuna faccia non tocca elementi di interesse ==> non
    fare niente
    if ( tri.iffj[j] < 0 ) continue;
    //Controllo se tocca un bosco (un lato è sul bosco)
    if ( tri.iffj[j]<boschi.size() )
        toccaBosc = true;
    //Controllo se tocca una strada (un lato è sulla strada)
    if ( tri.iffj[j]>=boschi.size() )
        toccaStrad = true;
}
...

```

Tabella 3.1: Algoritmo per l'analisi iniziale delle facce di un triangolo

Il primo controllo effettuato consiste nel verificare se il triangolo esiste realmente, quindi in caso affermativo si eseguono le restanti operazioni all'interno del ciclo, invece, in caso negativo, si passa all'iterazione successiva (`continue`) e quindi alla selezione ed analisi di un nuovo triangolo.

Sel il triangolo esiste realmente, si controlla se esso ha dei lati che fanno parte delle geometrie di partenza, cioè quelle date in input al triangolatore. In caso affermativo si provvede ad accendere un'apposita variabile semaforo (`toccaStrad` o `toccaBosc`), a seconda del fatto che l'elemento in comune (elemento toccato) sia una strada oppure un bosco. Si discrimina tra bosco e strada

grazie al fatto che alle strade viene assegnato un identificatore superiore rispetto ai boschi, perchè inserite successivamente all'interno del triangolatore. Come già detto precedentemente, il campo che fornisce questa informazione è il campo `faces`, presente all'interno di ciascun record contenente i dati relativi a ciascun triangolo generato attraverso la Triangolazione di Delaunay vincolata. In particolare, per fare ciò, si usa il metodo `tri.iff[j]`, situato all'interno del package `Triangulation` nella classe `Triangulator.java`, che permette di analizzare la `j`-esima faccia del triangolo `tri` e di vedere con che cosa coincide.

Un altro metodo utile, è il metodo `getFacesAroundVertex(Triangle[] arTri, int startTri, int vertex)`, situato nella classe `Triangulation.java` che serve a controllare se anche i triangoli costruiti attorno a ciascun vertice del triangolo in analisi, hanno dei lati che fanno parte degli elementi forniti in input al triangolatore.

Fatta questa serie di controlli preliminari, si effettua un ulteriore controllo che ha l'obiettivo di verificare se esistono dei triangoli che sono situati all'interno della zona da estendere – se così la si può definire – che non hanno almeno un vertice su entrambe le geometrie. Questa tipologia di elementi, ovviamente deve far parte dell'area estensiva, ma l'algoritmo fin qui descritto non è in grado di individuare questi elementi causando la presenza di una serie di inaspettati “buchi” all'interno dell'area da estendere. Per questo particolare tipo di problematica è stato messo a punto un particolare algoritmo che attraverso un'analisi progressiva dei triangoli vicini al triangolo in analisi, riesce a discriminare se il triangolo in analisi appartiene o meno all'area da estendere, pur toccando una sola delle geometrie di input.

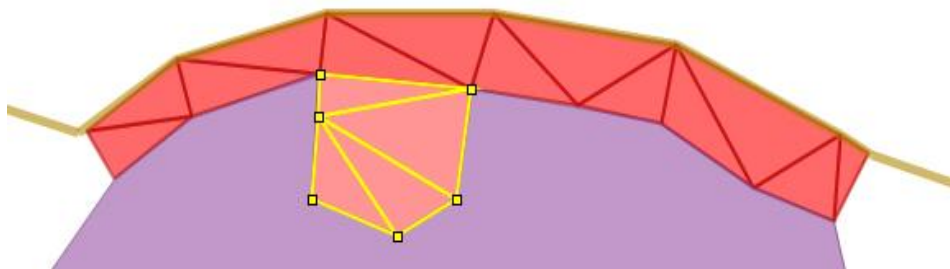


Figura 3.8: Esempio di triangoli che toccano solo il bosco

Se i controlli effettuati producono esito positivo, si memorizza il triangolo nella tabella estens, destinata a contenere l'area finale da aggiungere al bosco e si inseriscono in una hashMap l'identificatore e le coordinate di quel triangolo.

Si ricorda che una hashMap può essere definita come un algoritmo che restituisce una stringa di numeri e lettere a partire da un qualsiasi flusso di bit di qualsiasi dimensione. La stringa di output è univoca per ogni documento e ne è un identificatore (Goodrich, 2004).

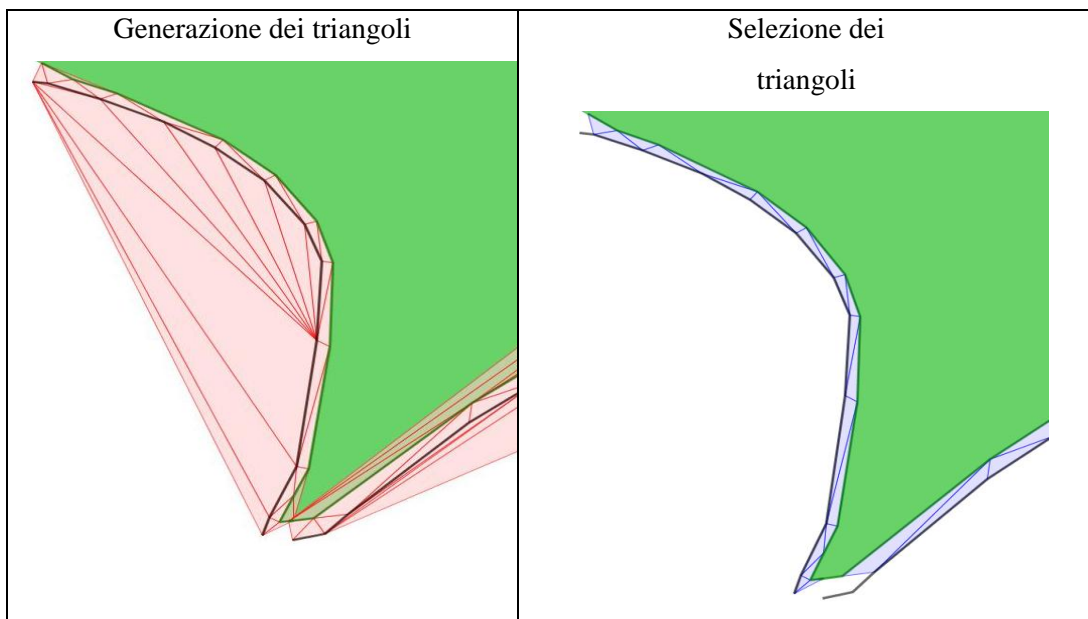


Figura 3.9: Esempio applicazione algoritmo di selezione triangoli

Prima di effettuare il salvataggio definitivo della tabella estens, occorre trovare ed eliminare i triangoli marginali.

Con il termine marginali, ci si riferisce a tutti quei triangoli che hanno un lato scoperto, cioè un lato che non è in comune a nessuno dei segmenti di strada o di bosco, forniti in input al triangolatore, ne tanto meno a nessuno degli altri triangoli presenti all'interno della tabella estens e che sono stati selezionati per essere aggregati ai boschi e che quindi, costituiscono l'area aggiuntiva finale.

Una volta scovati questi triangoli, occorre verificare che l'angolo opposto al lato giacente sulla strada sia superiore ad una certa soglia (si è scelto di optare per un valore di circa 80°), in maniera da ridurre un possibile e indesiderato effetto a

zampa d'elefante, cioè che il bosco si allarghi nella parte finale (in prossimità della strada), assumendo la forma di una zampa di elefante.

Per effettuare in maniera rapida la ricerca dei triangoli marginali si ricorre, come già detto poco fa, all'uso di una HashMap usata congiuntamente al controllo dei valori presenti all'interno del campo Faces.

Inizialmente si inserisce all'interno di questa struttura dati l'identificatore e le coordinate di tutti i triangoli che vengono man mano inseriti nella tabella estens.

Successivamente quando si effettua la ricerca dei triangoli marginali si procede nel modo seguente:

1. si controllano i valori presenti nel campo Faces[j] (per $i=0..2$) e ogni volta che uno dei tre valori presenti in tale campo è maggiore di -1 si incrementa una variabile denominata tot ed utilizzata come contatore, invece se il valore di Faces[j] vale -1 si passa al punto seguente;
2. per verificare se un triangolo ha un lato in comune con un triangolo presente nella tabella estens, occorre vedere se la chiave ricercata è presente o meno all'interno della struttura dati (HashMap). Se la chiave è presente si incrementa la variabile tot. Tale stratagemma permette di ottenere dei costi molto bassi nella ricerca dei triangoli marginali;
3. alla fine del ciclo che analizza ogni triangolo, si controlla il valore della variabile tot e, se questo è minore di 3, si effettua il calcolo dell'angolo ed in caso di valore non idoneo, si deve rimuovere il triangolo sotto analisi tabella estens (rimozione si effettua direttamente accedendo tramite ID) e si deve aggiungere questo triangolo alla tabella app all'interno della quale ad ogni iterazione vengono inseriti gli elementi da cancellare.

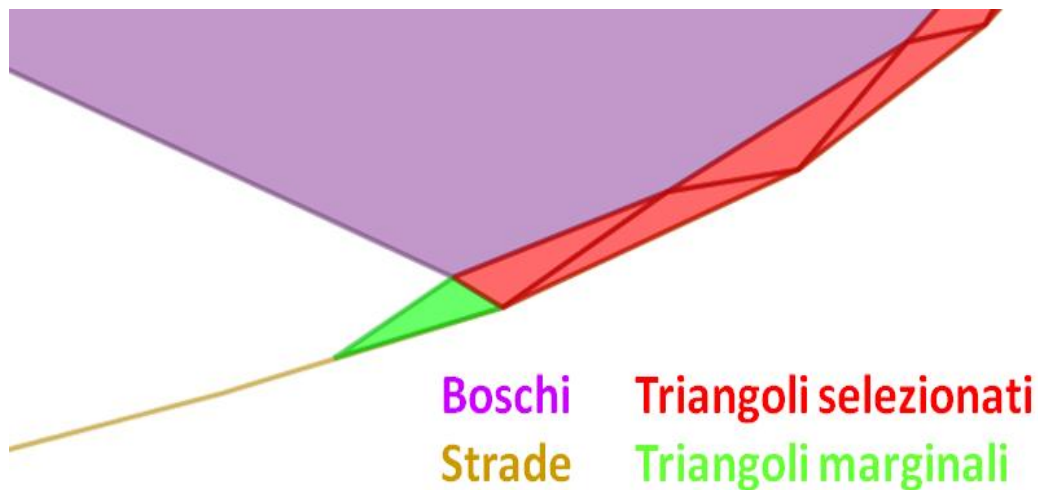


Figura 3.10: Esempio di triangolo marginale

Dopo l'analisi di ciascuna delle triangolazioni realizzate, la tabella app viene ripulita ed il suo contenuto (triangoli marginali individuati) viene fatto migrare nella tabella toCanc che viene usata come contenitore globale di triangoli marginali.

Lo stesso genere di operazione viene effettuato per la tabella estens: il suo contenuto viene trasferito ogni volta nella tabella estens2, le cui dimensioni crescono quindi progressivamente, mentre la tabella estens viene ripulita del tutto ogni volta, pronta per analizzare una nuova triangolazione.

Oltre a quanto appena detto, viene prodotta una geometria ottenuta attraverso l'unione di tutti i triangoli facenti parte dell'area da estendere. Tale dato viene inserito all'interno della tabella contenente l'effettiva area da estendere. I record di questa tabella, oltre che contenere l'oggetto Geometry contengono due campi:

1. ID costituisce un identificatore univoco per ciascun record. Gli ID partono da 1 e sono inseriti in maniera crescente;
2. IDbosco rappresenta l'identificatore univoco di un singolo bosco, cioè di un singolo record all'interno della tabella boschi, contenente la geometria alla quale deve essere aggiunta quella in questione.

3.5.2 Ricostruzione ed unione dei boschi

L'ultima parte del lavoro di generalizzazione dei boschi consiste nell'aggiungere ai boschi già esistenti le aree create come descritto nei paragrafi precedenti, sfruttando in maniera semplice e lineare le informazioni presenti nei vari campi delle tabelle via via costruite.

Prima di effettuare questa operazione, occorre però riportare i boschi nella loro forma originale, cioè occorre trasformare i perimetri, vale a dire gli elementi lineari con i quali si è lavorato fino ad ora, in aree.

Per far ciò si è preso in considerazione ciascun perimetro e lo si è trasformato in un'area mediante l'ausilio di una `GeometryFactory`. La `GeometryFactory` è un insieme di utility presenti all'interno della JTS che permettono di costruire oggetti di tipo `geometry`, a partire da un insieme di coordinate.

Fondamentalmente, nella ricostruzione dei boschi si distinguono due casi:

1. L'elemento è un oggetto `Geometry` di tipo `LineaRing`, cioè è un semplice anello chiuso;
2. L'elemento è una collezione di geometrie (`GeometryCollection`).

Nel primo caso basta costruire un poligono ricorrendo all'uso di una variabile di tipo `GeometryFactory`, alla quale vengono passati come parametri: l'anello esterno (`shell`) e gli eventuali anelli interni (`holes`) che, erano inesistenti in questo caso, con la conseguenza che il secondo parametro prende il valore `null`.

Nel secondo caso è necessario analizzare le singole geometrie presenti all'interno della collezione nel modo seguente: si costruisce il poligono relativo alla geometria contenente il perimetro esterno e via via, si procede con la sottrazione delle altre geometrie a quella costruita inizialmente.

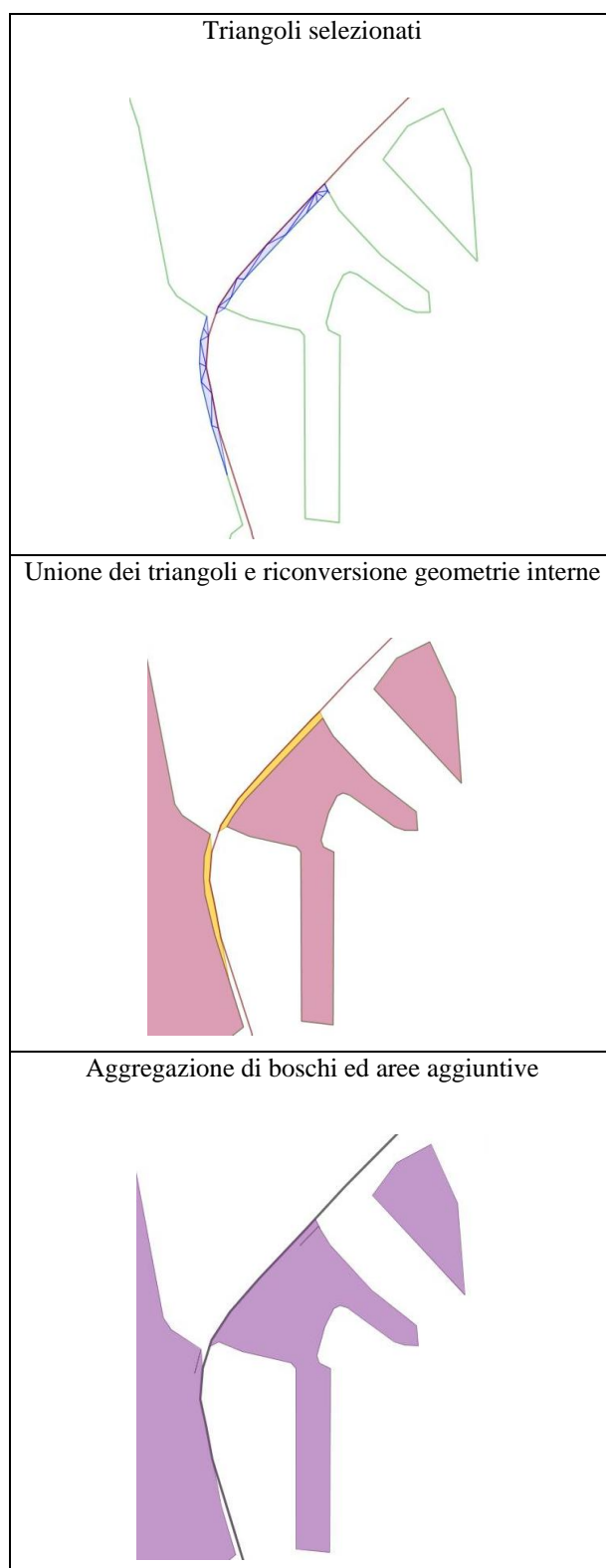


Figura 3.11: Processo di ricostruzione dei boschi

Capitolo 4

Ripristino della continuità degli elementi lineari

Il tema affrontato in questo quarto capitolo è il ripristino della continuità degli elementi lineari.

Con il termine elementi lineari si intende una vasta gamma di oggetti tra i quali si citano alcuni esempi: muri di sostegno, muri divisori, recinzioni, palizzate, recinzioni con filo spinato, staccionate, siepi, ect.

L'obiettivo principale del lavoro svolto è, come imposto dalle specifiche IGM, quello di ripristinare la continuità di tali elementi. Alcune tra le cause che alterano la continuità di questa classe di elementi sono:

1. Presenza di altre classi di elementi quali strade, edifici, ect.
2. Presenza di elementi non rilevati quali potrebbero essere ingressi ad edifici

4.1 Pulizia dei dati in ingresso

Come nei casi precedenti, anche in questa situazione, prima di procedere è stato necessario effettuare una accurata opera di pulizia e sistemazione dei dati di input.

Tra le principali operazioni di pulizia realizzare si citano:

1. Eliminazione di punti duplicati all'interno di uno stesso oggetto geometry;
2. Eliminazione dei record che al proprio interno contengono geometrie vuote;
3. Eliminazione delle geometrie duplicate.

Tutti gli elementi che hanno superato questo test sono stati memorizzati in una nuova tabella, denominata Nuovo_El.

4.1.1 Eliminazione delle sporgenze

La seconda parte dell'operazione di pulizia dei dati in ingresso consiste nell'eliminazione delle sporgenze presenti sui dati in ingresso.

Con il termine sporgenza si intende quella situazione che si viene a creare quando ci sono due elementi lineari che si intersecano in un punto e, questo punto di intersezione è molto vicino all'estremo finale, oppure a quello iniziale di uno dei due elementi, dando origine ad una vera e propria "sporgenza", che potrebbe generare problemi nelle operazioni descritte di seguito.

Quindi, tramite l'ausilio di una serie di metodi realizzati ad hoc, si vuole risolvere questo tipo di problematica. Per fare ciò è ovvio che occorre analizzare a coppie tutti gli oggetti facenti parte del dataset di partenza che si intersecano fra di loro.

Inizialmente, dopo aver selezionato le coppie di elementi che si intersecano, si ricorre al metodo *calcola*, al quale vengono passati come parametri le coordinate del punto di intersezione tra due elementi lineari e le coordinate della geometria sotto analisi. Qui, all'interno di questo metodo si cerca l'indice del punto più vicino all'intersezione e si calcola la distanza di quest'ultimo dall'intersezione. Naturalmente occorre eseguire il metodo, cioè effettuare il calcolo, per entrambe le geometrie che si intersecano e successivamente effettuare l'operazione di accorciamento anche in entrambi i casi se necessario, cioè anche nel caso in cui sono presenti due sporgenze su una stessa intersezione.

Detto questo, si provvede ad eseguire l'accorciamento della geometria più piccola modificando le coordinate del primo o dell'ultimo punto, a seconda della situazione in cui ci si trova. Però, prima di effettuare questa operazione occorre anche eseguire un'ulteriore metodo, denominato *Noding* che si prepone per l'appunto, l'obiettivo di effettuare il noding, cioè non si vuole far altro che aggiungere tra i punti dell'elemento che non viene accorciato, il punto di intersezione con l'altro elemento, in maniera tale da non incorrere in possibili errori topologici nell'esecuzione delle operazioni future.

Al metodo *noding* vengono passati come parametri:

1. *dato1* è l'elemento da accorciare;

2. dato2 è l'elemento sul quale bisogna effettuare il noding e di conseguenza al quale occorre aggiungere un punto;
3. vet2 è un array di coordinate contenente le coordinate della geometria dato2

All'interno di questo metodo per prima cosa si cerca quale tra i segmenti che compongono dato2 si interseca con dato1. Una volta trovato il segmento che interseca dato1, vengono copiati tutti i punti che precedono quel segmento in un nuovo array di coordinate, più grande di una unità rispetto a quello passato come parametro (dato1), in maniera tale da poter aggiungerci anche il nuovo punto. Successivamente, si aggiunge il nuovo punto ed infine si copiano in coda a questi i restanti punti dell'elemento di partenza. La geometria appena costruita sarà l'oggetto restituito dal metodo noding.

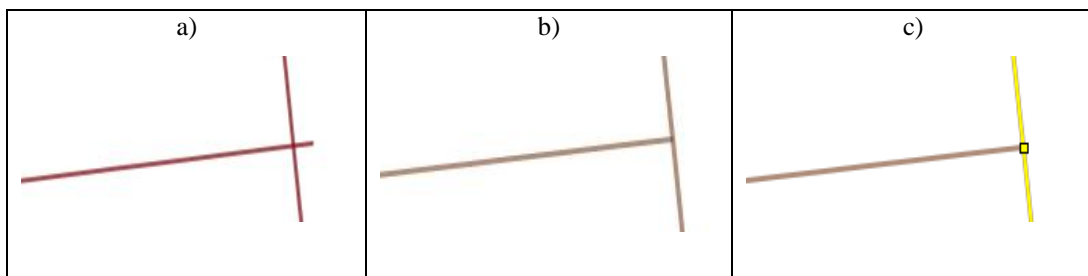


Figura 4.1: Rappresentazione del processo di eliminazione delle sporgenze. a) situazione di partenza, b) eliminazione sporgenza; c) noding sull'elemento non accorciato

Ora, dopo aver eseguito con successo il noding sul secondo elemento, occorre modificare il primo elemento accorciandolo, in maniera da eliminare la sporgenza. Anche per questa specifica operazione è stato realizzato un piccolo metodo, `updateCoord` che si preoccupa di accorciare l'elemento semplicemente modificandone le coordinate, al quale vengono forniti i seguenti parametri;

1. `vect` è un array di coordinate contenente l'elemento da modificare;
2. `index` è l'indice dell'elemento che occorre modificare all'interno dell'array `vect`;
3. `punto` è una singola coordinata e rappresenta la nuova coordinata dell'`index`-esimo elemento dell'array `vect`.

Alla fine del procedimento si può sostituire l'oggetto `geometry` presente

all'interno del record sotto analisi nella tabella elementi, con la geometria appena realizzata.

E' importante sottolineare che le operazioni descritte devono essere eseguite per tutti gli elementi lineari che si intersecano fra di loro.

4.1.2 Eliminazione delle micro interruzioni

La terza parte della fase di pulizia dei dati di ingresso consiste nell'eliminazione delle micro interruzioni presenti tra due differenti elementi. Con il termine micro interruzione, si intende quello spazio che è presente tra due elementi lineari che apparentemente potrebbero anche sembrare congiunti, ma che in realtà non lo sono. Tale tipo di errore è rilevabile dall'occhio umano solo se si osservano le mappe con scale molto alte, cosa facilmente realizzabile con openjump.

Quindi, al fine di ottenere il ripristino più completo possibile della continuità degli elementi lineari, come stabilito all'inizio del capitolo, è indispensabile eliminare queste micro interruzioni, allungando gli elementi lineari che hanno estremi finali o iniziali in prossimità di altri elementi lineari fino a congiungerli con questi ultimi.

Inizialmente è stato realizzato un metodo denominato generaAllung che costruisce una tabella all'interno della quale ogni record è composto da due campi (Geometry e ID) all'interno dei quali vengono rispettivamente memorizzati:

1. un oggetto di tipo geometry corrispondente all'allungamento della parte iniziale oppure di quella finale di un determinato elemento divisorio;
2. un identificatore che serve a riconoscere univocamente ciascun elemento.

La tabella è stata strutturata in modo che l'allungamento della parte finale dell' i -esimo elemento divisorio sia contenuto nell' $2i$ -esimo record della tabella prolungamenti, mentre l'allungamento corrispondente alla parte iniziale dell' i -esimo elemento divisorio, sia contenuto nel $(2i+1)$ -esimo record della tabella prolungamenti.

Al metodo generaAllung vengono passati i seguenti parametri:

1. `puntiFin` è una tabella in cui ciascun record contiene un campo in cui è memorizzato l'oggetto `geometry` che corrisponde al punto finale dell'elemento i -esimo ed un campo chiamato `ID`, che viene usato come identificatore di quel record;
2. `puntiIn` è una tabella in cui ciascun record contiene un campo in cui è memorizzato l'oggetto `geometry` che corrisponde al punto iniziale dell'elemento lineare i -esimo ed un campo chiamato `ID`, che viene usato come identificatore.

Le tabelle sono state costruite in modo che gli elementi contenenti dati aggiuntivi relativi all' i -esimo elemento divisorio (estremo iniziale, estremo finale) siano situati nell' i -esimo record di ciascuna tabella ed abbiano tutti lo stesso identificatore, vale a dire i .

Una volta conclusa questa tediosa ma, importante fase di costruzione di oggetti che saranno indispensabili all'eliminazione vera e propria delle micro interruzioni, si può passare all'effettiva rimozione delle microinterruzioni.

Si parte effettuando il join spaziale tra la tabella contenente gli elementi lineari e la tabella contenente gli allungamenti di ciascuno degli elementi lineari, in modo da sapere quale di questi si intersecano. Dopodichè, si esegue il metodo `cancelBuchi` che è quello che effettua la reale eliminazione delle interruzioni. A questo metodo occorre passare come parametro la tabella contenete i risultati derivanti dal join spaziale del quale si parlava pocanzi.

All'interno del metodo, per prima cosa, si verifica che il prolungamento dell'elemento divisorio che si sta analizzando, non sia relativo all'elemento stesso che lo ha generato, perchè se si ricade in questa situazione, si può ovviamente fare a meno di analizzare il caso.

In caso contrario, si verifica se l'indice dell'elemento in questione è dispari oppure pari:

- se è dispari vuol dire che si riferisce ad un segmento iniziale;
- se è pari si riferisce ad un elemento finale.

Fatti questi controlli, a seconda delle casistiche in cui si è ricaduti, si modifica il

primo o l'ultimo punto dell'elemento lineare, sostituendolo con il punto d'intersezione tra l'elemento lineare e l'allungamento in questione.

Dopo aver realizzato l'allungamento dell'elemento lineare, con la conseguente cancellazione della relativa micro interruzione, bisogna effettuare anche un'operazione di noding, proprio come nel caso delle sporgenze. Per questo motivo, occorre aggiungere il punto di intersezione tra l'elemento divisorio appena allungato e l'elemento che si intersecava con il prolungamento alle coordinate di quest'ultimo, in modo tale da evitare di incorrere in possibili errori topologici nelle operazioni seguenti.

La logica con la quale si effettua il noding è uguale a quella usata e descritta per la cancellazione delle sporgenze, ed è per questo motivo che si fa a meno di descrivere nuovamente il suo funzionamento.

Per ogni modifica effettuata si procede poi alla modifica della corrispondente geometria nella tabella elementi, che è la tabella principale contenente gli elementi di riferimento e che deve essere progressivamente aggiornata.

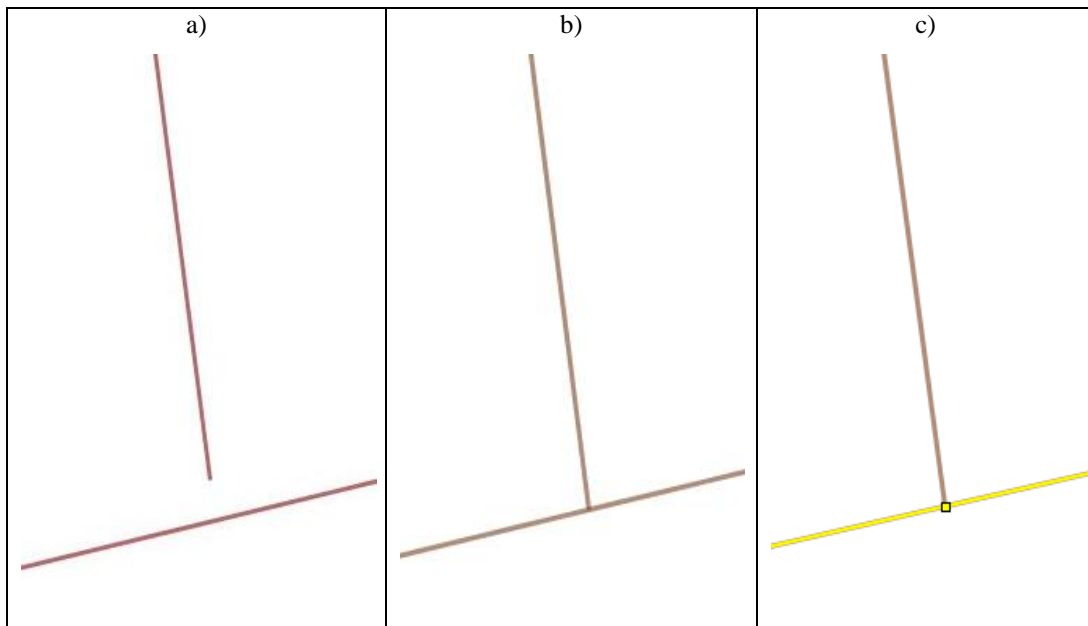


Figura 4.2: Rappresentazione grafica del processo di eliminazione delle micro interruzioni. a) situazione di partenza; b) eliminazione del buco; c) noding sull'elemento non accorciato.

4.2 Generazione di nuovi potenziali elementi lineari

Da un'attenta analisi delle mappe di partenza, usate come cartografia sulla quale effettuare i test, ci si accorge chiaramente che ci sono una quantità non trascurabile di vertici di elementi divisori che sono situati nei pressi di vertici di edifici. In alcuni casi, la vicinanza tra queste due categorie di elementi, è così marcata tanto che osservando la mappa con scale abbastanza piccole, alcune coppie di vertici appartenenti a questi elementi potrebbero sembrare tranquillamente sovrapposti.

Da quanto appena detto, segue che potrebbe essere utile e comodo, oltretutto efficientemente estetico, far coincidere realmente queste coppie di punti che sono molto vicini. Tuttavia, si rimanda alle pagine successive la spiegazione della reale motivazione che ha portato alla realizzazione di questa operazione.

Per spostare i vertici degli elementi lineari è stato utilizzato un kd-tree. Questa struttura dati è un albero binario nel quale ogni nodo rappresenta un punto k-dimensionale (C.L.R.S, 2005). Ogni nodo non foglia dell'albero può essere pensato come un iperpiano che divide lo spazio in due sottospazi. I punti a sinistra rappresentano il sottoalbero di sinistra di quel nodo, mentre i punti a destra sono rappresentati dal sottoalbero destro.

I kd-tree sono particolarmente utili per diverse applicazioni, quali ad esempio l'elaborazione efficiente di query di tipo geometrico, come nel caso del lavoro svolto nell'ambito di questa tesi.

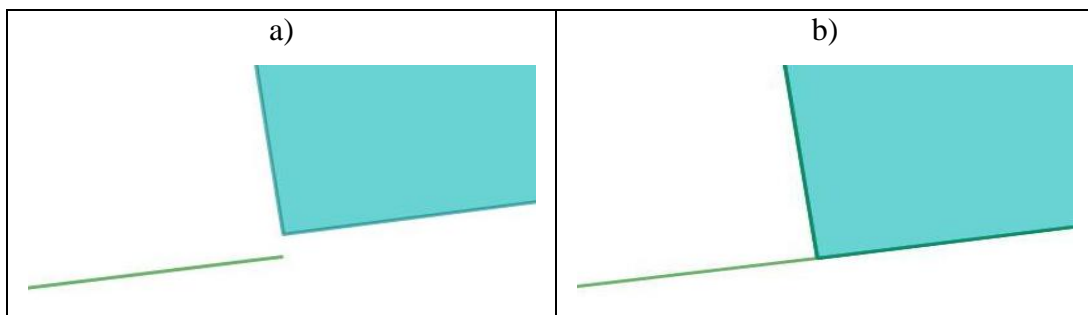


Figura 4.3: Rappresentazione grafica dello spostamento del vertice di un elemento lineare (segmento verde) situato in prossimità di un edificio (elemento di colore azzuro).

Perciò, si è deciso di realizzare un apposito metodo, chiamato nearEdif, all'interno del quale vengono inizialmente inseriti tutti i vertici degli edifici in un kd-tree.

Successivamente, si prendono in considerazione tutti i vertici iniziali e finali di ciascun elemento divisorio e si costruisce una envelope attorno a ciascuno di essi, sfruttando come al solito le potenzialità ed i metodi presenti nella JTS. A tale proposito, si ricorda che una envelope è una regione rettangolare bidimensionale, spesso usata per delimitare un'area attorno ad una determinata geometria.

Dopo la costruzione di ciascuna envelope, si verifica per ognuna di queste se contiene uno dei vertici degli edifici ed in caso affermativo si modifica il vertice dell'elemento lineare (iniziale o finale, a seconda della casistica che si sta analizzando), sostituendolo con quello dell'edificio che è coperto dall'envelope.

4.2.1 Creazione di elementi lineari attraverso lati degli edifici

Un'altra caratteristica venuta a galla attraverso un'accurata osservazione dei dataset di partenza, è che molti elementi lineari sono interrotti o separati da altre classi di elementi quali potrebbero essere alcuni lati di edifici. Da ciò nasce l'idea di costruire alcuni ulteriori ed ipotetici elementi lineari proprio in corrispondenza di tali interruzioni.

A questo punto, diventa chiara anche l'utilità del passo precedentemente effettuato, cioè dello spostamento dei vertici degli elementi lineari sui vertici degli edifici, che dunque, serve ad aumentare la possibilità di incontrare lati di edifici che possono essere considerati alla stregua di potenziali elementi lineari

Per generare nuovi potenziali elementi lineari è stato realizzato il metodo `latiEdif`, all'interno del quale viene inizialmente eseguita l'operazione di `join` spaziale fra la tabella contenente gli elementi lineari e quella contenente gli edifici (`fabbric`).

Fatta questa operazione si provvede all'inserimento in una `hashmap` di tutti i vertici degli elementi lineari che sono in contatto con edifici, che dunque sono presenti nella tabella che contiene i risultati ottenuti dopo l'esecuzione del `join` spaziale.

Tornando a parlare della `hashmap`, occorre sottolineare che bisogna inserire all'interno di essa per ciascun vertice di interesse dell'elemento lineare, una coppia di elementi (`<Coordinate,Integer>`) composta dall'identificatore

relativo all'elemento lineare del quale fa parte il punto e le coordinate del punto stesso.

Successivamente si considerano tutti gli identificatori relativi agli edifici selezionati dopo il join spaziale, i quali sono stati memorizzati in un array di interi. Per ciascuno di questi elementi si carica in memoria la corrispondente geometria andando a considerare tutti i punti di ciascuna di essa, in sostanza tutti i punti degli edifici selezionati.

Ora, scorrendo ciascun vertice di ciascuna geometria corrispondente ad un edificio, basta controllare se per una stessa geometria vi sono coppie di vertici consecutivi che si trovano all'interno della hashmap e, in caso affermativo, si provvede ad effettuare la costruzione del segmento avente come estremi questi due punti e successivamente, al salvataggio di tale dato in una nuova tabella chiamata *toccaFab*, all'interno della quale ciascun record è composto da due campi: uno contenente l'oggetto *geometry* e l'altro l'identificatore univoco di quell'oggetto.

Il metodo usato per verificare se un elemento è presente all'interno della hashmap è il metodo *containsKey (key)*, dove il parametro *key* rappresenta la chiave (coordinata del punto in questo caso) che si sta cercando. Tale metodo è localizzato all'interno della classe *HashMap.java* che è contenuta nel package *java.util*.

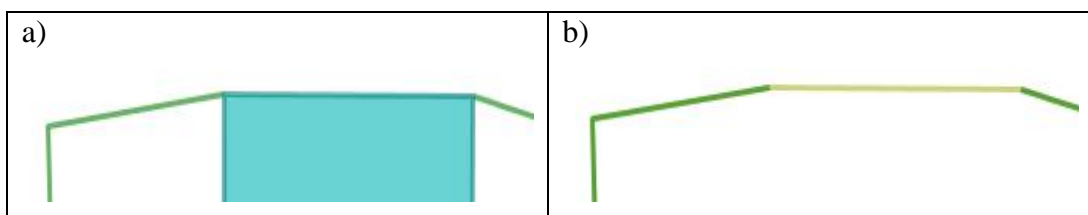


Figura 4.4: Rappresentazione grafica del processo di generazione di nuovi potenziali elementi lineari (segmento giallo nella parte b della figura)

E' possibile che alcuni tra gli elementi creati nel modo appena descritto siano già esistenti, oppure che essi siano parte di alcuni elementi già esistenti. Per questo motivo, occorre effettuare il join spaziale tra la tabella contenente gli elementi appena creati e quella contenente gli elementi di partenza, eliminando le categorie appena citate. Per far ciò, quando si effettua il join spaziale, occorre passare come

parametri al metodo join i seguenti dati:

1. tab è il nome della seconda tabella con la quale si vuole effettuare l'operazione spaziale;
2. mask è una stringa contenente la maschera spaziale che si vuole usare, "coveredby+within" in questo caso, che serve per selezionare gli elementi di una tabella che sono interamente coperti da elementi della seconda;
3. type è un valore intero che indica la sintassi utilizzata dalla maschera, 0 per la sintassi di Oracle, 1 per quella di PostGis.

Una volta eliminati gli elementi sovrapposti si può procedere alla fusione delle due tabelle, aggiungendo gli elementi superstiti in coda a quelli di partenza.

4.3 Chiusura degli elementi lineari

Dopo aver effettuato tutte le varie operazioni di ripulitura e adattamento delle geometrie, narrate nei paragrafi precedenti, si passa alla effettiva chiusura degli elementi lineari.

Osservando gli elementi divisorii situati attorno agli edifici, che sono una particolare categoria di elementi lineari, si può notare che è molto difficile identificare delle figure chiuse. In genere, queste cosiddette interruzioni sono dovute alla presenza di normali accessi agli edifici, che ovviamente nelle mappe in scala 1:25000, non vengono prese in considerazione e perciò non sono rappresentate sulle carte. Oltre a questo fattore però, come detto in precedenza, è possibile avere interruzioni anche a causa della presenza di elementi realmente rappresentati sulla mappa quali edifici, strade, ect.

Per effettuare la chiusura degli elementi lineari si usa anche in questo caso l'algoritmo di triangolazione di Delaunay vincolata tra gli elementi lineari, della cui semplicità, robustezza ed efficienza si è già parlato abbondantemente nelle sezioni precedenti.

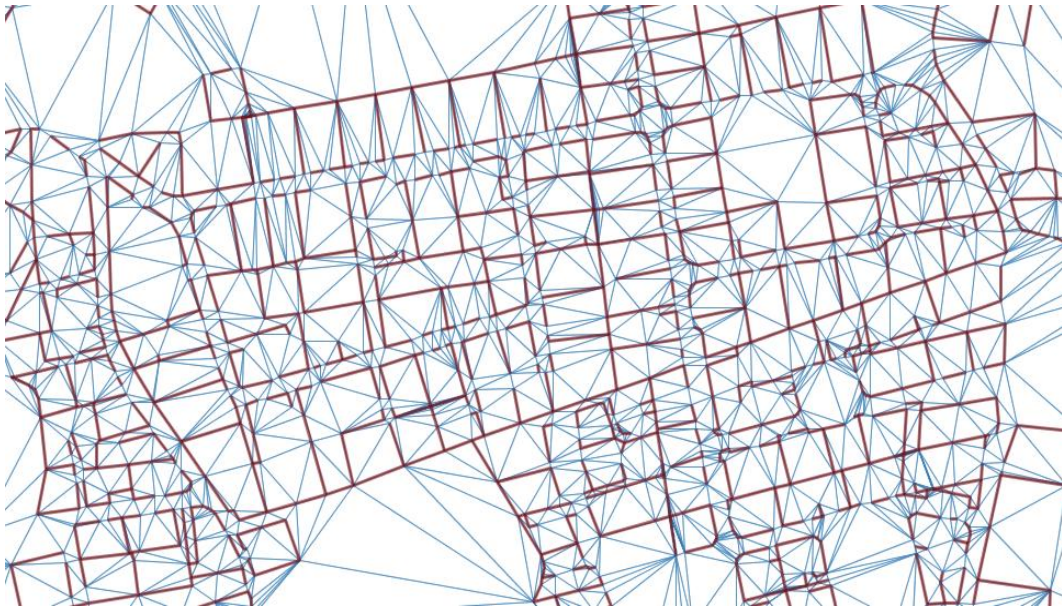


Figura 4.5: esempio triangolazione sugli elementi lineari

Si ricorda ancora una volta, che nessuno dei triangoli costruiti mediante l'algoritmo di triangolazione vincolata taglierà un'altra qualsiasi geometria tra quelle facenti parte degli elementi forniti in input al triangolatore.

Detto ciò, come al solito, si parte costruendo un oggetto arraylist di tipo geometry contenente tutte le geometrie corrispondenti agli elementi lineari da triangolare. Successivamente, viene passato questo oggetto come parametro al triangolatore e si effettua la triangolazione.

Infine viene usato il metodo `getTrianglesFeatures` che aggrega in una tabella tutti i dati di output relativi alla triangolazione.

4.3.1 Selezione dei lati dei triangoli

Dopo aver effettuato la triangolazione, occorre selezionare tra i lati dei triangoli generati, quelli che veramente provvedono a ripristinare la continuità degli elementi lineari in maniera idonea.

Per effettuare questa operazione è stato realizzato uno specifico metodo chiamato `selezionaLati`, il quale prende in input come parametri l'oggetto `Triangulator` precedentemente usato ed un array di tipo `Triangles`, contenente per l'appunto tutti i triangoli che sono stati generati dalla triangolazione.

Inizialmente, per ciascun triangolo occorre analizzare tutti i lati e prendere in considerazione tutti quelli che hanno i vertici coincidenti con vertici di elementi divisori. Per fare questo in maniera veloce, si utilizza il metodo `getFacesAroundVertex`, che prende come parametri di input un oggetto di tipo triangolo e uno dei suoi vertici e restituisce gli identificatori degli elementi toccati dai lati dei triangoli generati attorno a questo vertice.

Ottenuti gli identificatori di questi elementi, occorre verificare se i vertici del triangolo in questione sono vertici anche della geometria stessa, alla quale ci si sta riferendo. Tale controllo viene effettuato attraverso l'ausilio di un piccolo metodo realizzato anch'esso ad hoc, chiamato `isOnVertex`, che restituisce un valore booleano vero se è un vertice della geometria, falso in caso contrario. A questo metodo vengono passati come parametri: la coordinata di uno dei vertici del triangolo e l'identificatore di uno degli elementi toccati dai triangoli generati attorno a quel vertice.

Se un lato di un triangolo tocca due vertici distinti di geometrie facenti parte degli elementi lineari e, se la distanza tra questi due punti è minore di circa 15 metri, il lato viene preso in considerazione. Tuttavia, questo non è sufficiente per decidere se selezionare o scartare un lato.

L'obiettivo che ci si pone ora, è quello di selezionare tra i lati ritenuti fin qui buoni, quelli che formano con gli elementi lineari con i quali si congiungono degli angoli che sono prossimi a 90° , oppure suoi multipli, in modo tale da non avere dei congiungimenti esteticamente anomali tra i vari elementi lineari, che potrebbero sembrare inopportuni e forzati.

Inoltre, nell'accettazione dei lati si vuole dare meno importanza alle eventuali anomalie, man mano che diminuisce la lunghezza del segmento in analisi. Ciò succede perché quanto più piccolo è il lato, tanto meno evidente sarà l'anomalia, anche se gli angoli da esso formati risultano comunque non idonei.

Per fare tutto ciò sono stati realizzati due specifici metodi: `calcolaAngolo` e `accettaAngolo`.

Il primo di questi due metodi prende in input i seguenti parametri:

1. `ls` è un oggetto di tipo `LineString` corrispondente ad uno specifico lato di un

triangolo;

2. `elemDiv` è l'identificatore dell'elemento divisorio toccato dal segmento passato come parametro.

Questo metodo non fa altro che cercare il punto nel quale il segmento `ls` interseca l'elemento divisorio `elemDiv` e calcolare l'angolo formato da questi due, restituendo un valore numerico di tipo `double`. Per il calcolo dell'angolo viene usato il metodo `interiorAngle`, che si trova nella classe `Angle.java`, all'interno del package `com.vividsolutions.jts.algorithms`, facente parte della `JTS`.

Il metodo `interiorAngle` prende in input come parametri tre coordinate (`c0,c1` e `c2`), calcola l'angolo tra i due segmenti aventi come estremi rispettivamente le coppie di vertici (`c0,c1`) e (`c1,c2`).

Nel calcolo dell'angolo, nella situazione in cui il segmento selezionato interseca l'elemento lineare in un unico punto, si possono distinguere tre differenti possibilità:

1. l'intersezione tra l'elemento divisorio ed il lato del triangolo avviene nel punto di origine dell'elemento lineare e quindi, bisogna calcolare solamente un angolo;
2. l'intersezione tra l'elemento lineare ed il lato del triangolo avviene nel punto finale dell'elemento divisorio e, anche in questo caso, è sufficiente calcolare un solo angolo;
3. l'intersezione avviene in uno degli estremi centrali dell'elemento lineare e bisogna necessariamente misurare due angoli. Ciò è dovuto al fatto che in questo caso il segmento, pur essendo connesso ad un solo elemento lineare è effettivamente agganciato a due differenti segmenti dell'elemento divisorio ed è quindi necessario calcolare due volte l'angolo.

Se il segmento interseca l'elemento lineare in due differenti punti, corrispondenti agli estremi iniziali e finali dell'elemento lineare, significa che l'elemento viene chiuso da quel segmento e non esiste alcuna necessità di calcolare l'angolo, perciò si considera direttamente idoneo quel lato.

Il metodo `CalcolaAngolo` restituisce un array composto da due numeri di tipo

double, corrispondenti a due differenti angoli. Se l'angolo calcolato è solamente uno (a causa delle motivazioni appena spiegate), l'altro elemento dell'array di ritorno viene settato con un valore specifico, usato come semaforo (si è scelto 2000 come valore per questo flag e lo si rappresenta mediante la costante ANGLE_NOT_VALID).

Il metodo accettaAngolo è un metodo di tipo booleano che prende in input i seguenti parametri:

1. vect è un array composto da due double che contiene gli angoli che il segmento in analisi forma con l'elemento lineare;
2. dist è la lunghezza del segmento, cioè del lato del triangolo che si sta analizzando. Tale valore viene usato per calcolare la tolleranza da prendere in considerazione nella fase di accettazione dell'angolo.

All'interno del metodo accettaAngolo, per prima cosa viene controllato se nell'array vect, passato come parametro, ci sono valori uguali a quello della costante ANGLE_NOT_VALID.

Invece, se si è in presenza di un qualsiasi valore, viene effettuato il calcolo della tolleranza che sarà usata nel decidere se accettare o meno il lato in analisi.

Per il calcolo della tolleranza si è deciso di assegnare a questa un valore inversamente proporzionale alla lunghezza del lato, in modo che la probabilità di non scartare gli elementi più piccoli sia minore.

La formula usata per il calcolo della tolleranza è quella illustrata nel riquadro seguente:

$\text{tolleranza} = \text{vect}[i] * \text{LUNG_MAX} / \text{dist};$
dove:
<ul style="list-style-type: none">• vect[i] rappresenta il valore dell'angolo i-esimo;• LUNG MAX è la lunghezza massima consentita del segmento;• dist è la lunghezza del segmento.

Tabella 4.1: Calcolo della tolleranza

Una volta calcolata la tolleranza si controlla se l'angolo appartiene ad uno dei

seguenti intervalli:

$$k \frac{\pi}{2} - \text{tolleranza}, \quad k \frac{\pi}{2} + \text{tolleranza} \quad k[0..4]$$

In caso affermativo si ritorna il valore true, altrimenti si ritorna false e si scarnerà il lato.

Naturalmente, per ogni elemento divisorio occorre eseguire due volte le procedure per il calcolo dell'angolo e per l'accettazione dello stesso: una per il primo estremo del segmento, l'altra per il secondo estremo. Solo se entrambi gli angoli calcolati saranno accettabili, il lato del triangolo in questione sarà memorizzato in una nuova tabella destinata a contenere gli ulteriori elementi lineari aggiuntivi, in cui ciascun record sarà composto da due campi:

1. geometry contenente l'oggetto geometry;
2. id, un intero usato come identificatore dell'elemento.

Si ricorda, che la tabella della quale si sta parlando, all'interno della quale saranno memorizzate queste geometrie, è l'oggetto di ritorno del metodo selezionaLati, che si occupa di chiudere gli elementi lineari, generandone degli altri.

4.3.2 Eliminazione di elementi non idonei

Tra tutti i segmenti selezionati attraverso la procedura selezionaLati, ve ne sono una vasta gamma che possono essere classificati come non idonei e che non devono entrare a far parte del dataset finale contenente tutti gli elementi lineari.

Della categoria degli elementi non idonei fanno parte:

1. elementi che intersecano le strade;
2. elementi duplicati;
3. elementi che sono totalmente coperti da altri elementi lineari già esistenti;
4. elementi che intersecano gli edifici.

Per rimuovere questa gamma di elementi dalle dimensioni non trascurabili, occorre effettuare una serie di join spaziali.

Il primo join spaziale va effettuato tra la tabella contenente le strade e gli elementi lineari che sono stati costruiti attraverso il metodo `selezionaLati`, con l'obiettivo di rimuovere tutti i record che contengono oggetti di tipo `geometry` che risultano intersecare le strade. Queste operazioni vengono implementate all'interno del metodo `toccaStrada`, che prende in input la tabella con i segmenti generati attraverso il metodo `selezionaLati`, ne elimina da questa i record contenenti geometrie che non soddisfano le condizioni richieste e restituisce tale tabella come output.

La presenza degli elementi duplicati, è dovuta al fatto che quando si analizzano i triangoli uno stesso lato può essere analizzato più volte riferendosi a triangoli diversi.

Per rimuovere questa categoria di elementi dalle dimensioni non trascurabili, occorre effettuare ancora una serie di join spaziali.

Per l'eliminazione degli elementi duplicati viene effettuato un join spaziale tra la tabella contenente tutti gli elementi generati attraverso il metodo `selezionaLati` e se stessa, settando i parametri da passare al metodo che esegue l'operazione spaziale nel modo seguente:

1. "equals" è una stringa che indica il tipo di maschera spaziale che si deve utilizzare, è facile capire che usando questa maschera si selezionano gli elementi esattamente uguali;
2. il numero 1, indica che si sta utilizzando la sintassi di PostGis.

Successivamente viene eseguito il metodo `cancSovrapposti`. Questo metodo ha una logica molto simile a quella del metodo precedente. Le uniche differenze fra i due metodi utilizzati stanno nella maschera a cui si ricorre e nella tabella passata come parametro. Infatti, in questo caso viene utilizzata la maschera: "coveredby+within" che seleziona gli elementi che sono interamente coperti da quelli della prima tabella. Inoltre, l'operazione spaziale in questo caso viene eseguita tra due tabelle differenti: quella contenente gli elementi divisori presenti nel dataset di partenza e quella contenente gli elementi generati attraverso il metodo `selezionaLati`.

Dopo questa ulteriore scrematura, occorre eliminare tra gli elementi costruiti quelli che intersecano gli edifici. Per non essere ripetitivi, senza addentrarsi nei particolari si segnala soltanto che la logica di quest'operazione è simile a quelle descritte nelle righe precedenti

Alla fine di tutto, viene effettuata l'unione delle due tabelle, (quella contenente gli elementi lineari facenti parte del dataset di partenza e quelli generati attraverso il metodo `selezionaLati`) mettendo in coda alla prima tabella, gli elementi superstiti della seconda e assegnando a ciascuno di essi identificatori differenti e progressivamente crescenti.

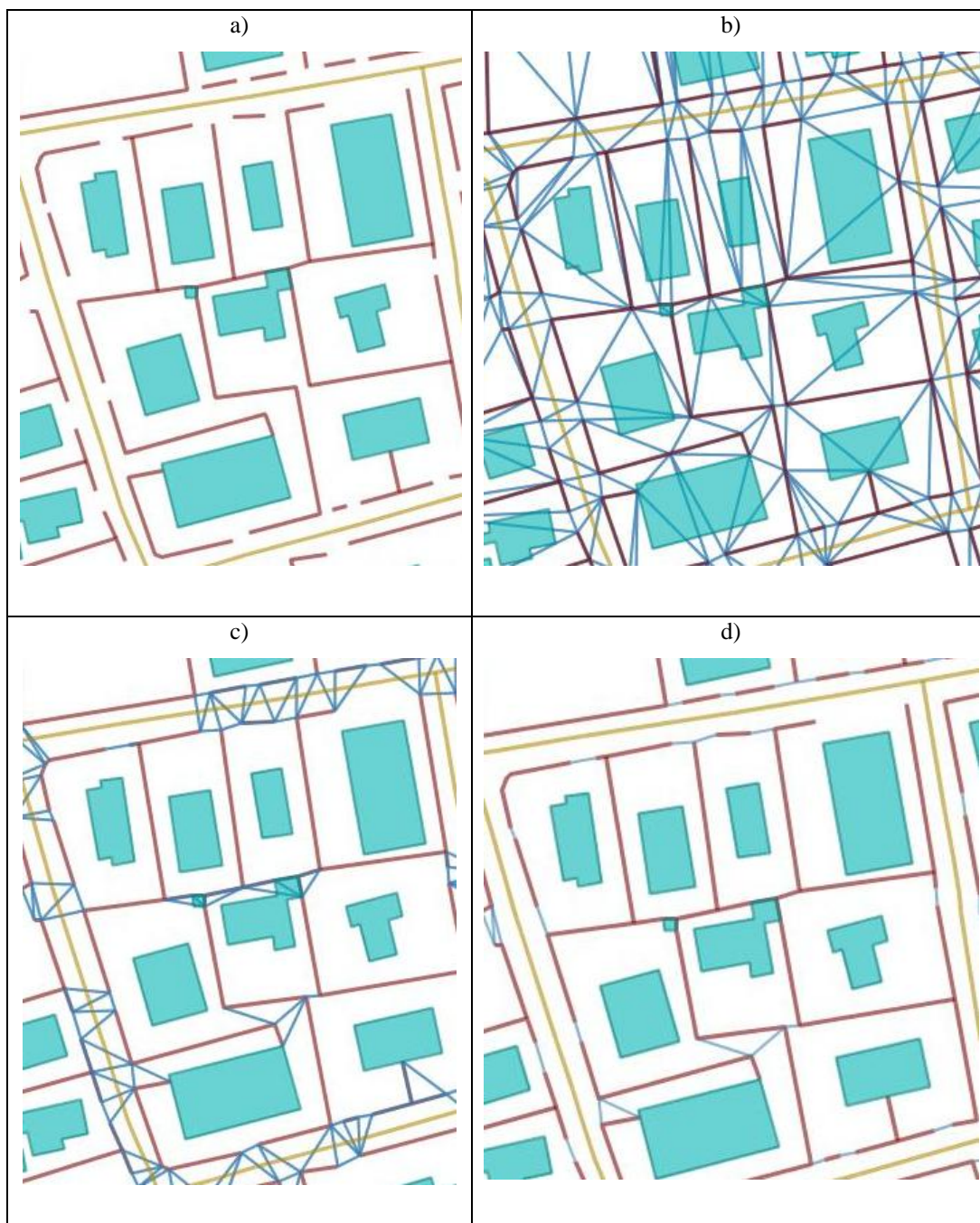


Figura 4.6: Ripristino della continuità - a) situazione di partenza; b) generazione nuovi lati mediante l'algorithmo di triangolazione di Delaunay vincolata tra gli elementi lineari; c) eliminazione degli elementi che intersecano gli edifici; d) eliminazione degli elementi che intersecano le strade

Capitolo 5

Generalizzazione di elementi divisorii

L'obiettivo del lavoro trattato in questo capitolo è quello di realizzare degli algoritmi per la generalizzazione degli elementi divisorii nella transizione dalla scala 1:5000 a quella 1:25000, ovviamente in accordo con le specifiche imposte da IGM.

Secondo le specifiche imposte da IGM fanno parte degli elementi divisorii le seguenti categorie di elementi: argini, muri di sostegno, muri divisorii, siepi, staccionate, palizzate, reti e filo spinato

5.1 Selezione di elementi divisorii

L'idea principale che si vuole sviluppare consiste nell'andare a studiare cosa c'è all'interno di ciascuna area chiusa delimitata attraverso elementi divisorii.

Tuttavia, prima di analizzare ciascuna area, è necessario effettuare la chiusura degli elementi divisorii, in modo da poter analizzare il maggior numero possibile di aree chiuse. E' a tale proposito che si applica l'algoritmo per il ripristino della continuità degli elementi divisorii, del quale si è abbondantemente discusso nel capitolo precedente e che calza a pennello in questa situazione.

Una volta effettuata la chiusura degli elementi divisorii, si ricorre all'utilizzo di un plugin già esistente, facente parte delle funzionalità offerte da openjump per individuare e selezionare ciascuna area delimitata dagli elementi divisorii.

Si tratta del plugin che effettua il grafico planare. L'esecuzione di tale plugin attiva una finestra denominata "Analisi Topologiche" che consente di analizzare alcune proprietà topografiche dei layer poligonali.

La funzione grafico planare crea quattro nuovi layer posti in un'apposita categoria denominata Grafico. I layer sono denominati con il nome del layer analizzato seguito da un suffisso:

1. “Nome_Nodo” è un layer di tipo puntuale che contiene il primo vertice di ciascun poligono;
2. “Nome_Spigolo” è un layer contenente geometrie di tipo polilinea, rappresentanti il confine dei poligoni; se è stata attivata l’opzione “Calcola la relazione arcs-nodes and/or arcs-faces” negli attributi vengono riportati il nodo di inizio, il nodo di fine, la “face” posta a destra e quella a sinistra della linea;
3. “Nome_Maglia” è un layer poligonale che riporta negli attributi un campo ID;
4. “Nome_Trasferisci” è un layer poligonale che riporta tutti gli attributi del layer analizzato se è stata attivata l’opzione “Mantieni attributi”.

Delle funzioni appena descritte, quella che è utile ed attinente all’obiettivo finale di questo lavoro è il layer Maglia. Questo livello contiene tutte le maglie che sono state formate a seguito della chiusura degli elementi divisori.

Dunque, effettuando un join spaziale tra questa tabella e quella contenente gli edifici si può facilmente vedere cosa è contenuto all’interno di ciascuna maglia e conseguentemente decidere se conservare o scartare l’elemento divisorio che delimita quella maglia.



Figura 5.1: Rappresentazione grafica delle maglie individuate attraverso l’ausilio del plugin di openjump PlanarGraph. Fi

L'algoritmo sviluppato conserva le maglie delimitate dagli elementi divisori che rispettano almeno una delle seguenti caratteristiche:

1. contengono edifici di rilievo quali possono essere ospedali, centrali elettriche, impianti di depurazione;
2. contengono edifici di grossa dimensione;
3. contengono un numero di edifici superiore alla soglia stabilita.

E' importante sottolineare che il procedimento appena descritto è applicabile per la selezione di elementi divisori che formano maglie chiuse. Se un elemento, pur importante, non viene chiuso attraverso gli elementi divisori, con la conseguenza che non esiste una maglia che lo contiene, allora gli elementi divisori attorno a questo elemento vengono irrimediabilmente scartati.

Si rimanda a sviluppi futuri la realizzazione di metodi per l'analisi e la selezione di elementi divisori che non formano maglie.

Capitolo 6

Conclusioni

Gli algoritmi sviluppati durante questo lavoro, sono stati realizzati tenendo forte considerazione per l'aspetto prestazionale, visto e considerato l'alto numero di elementi da trattare ed anche la grande dimensione di molti di essi.

Uno strumento che viene usato frequentemente (nel trattamento delle adiacenze tra elementi lineari ed areali e nel ripristino della continuità degli elementi lineari) è la triangolazione di Delaunay. Tale strumento è stato utilizzato in modo estensivo perché produce risultati robusti e facilmente analizzabili con buona velocità di esecuzione, a differenza di altre tecniche di geometria costruttiva che apparentemente si sono rilevate concettualmente più semplici rispetto alla triangolazione, ma in realtà hanno prodotto risultati meno efficienti sotto l'aspetto prestazionale e talvolta, anche meno buoni dal punto di vista della generalizzazione.

Continuando a parlare della triangolazione di Delaunay, occorre evidenziare che dei piccoli bug non del tutto risolti, sono presenti all'interno della versione vincolata dell'algoritmo. In particolare, l'esecuzione di tale algoritmo per alcuni dataset genera un errore di overflow che blocca l'esecuzione del programma.

Complessivamente, si può affermare che i risultati raggiunti alla fine del lavoro svolto possono considerarsi buoni.

Volendo dare uno sguardo ai possibili sviluppi futuri, si può affermare che una parte del lavoro realizzato che può essere senza dubbio ampliata ed estesa, è quella che riguarda la generalizzazione di elementi divisorii, attraverso la progettazione di ulteriori tecniche per la selezione delle maglie calcolate

BIBLIOGRAFIA

AGENT (2001) *Strategic Algorithms Using Organisations*, Public report, 2001, pp 6.

AGENT (2003) *Generalisation Modelling using an agent paradigm* Public report, 2003.

Bertin J. (1983) *Semiology of graphics: Diagrams, Networks, Maps* Madison, WI:University of Wisconsin Press, pp 300.

Cormen T.H., Leiserson C.E., Rivest R.L., Stein C: (2005) *Introduction to Algorithms*, MIT Press

De Gennaro M., Rumor M., Savino S. (2009) *Le procedure per la derivazione del DB25 dal DBT della Regione del Veneto: risultati del progetto CARGEN* Bollettino della Associazione Italiana di Cartografia, 135, Aprile 2009.

Douglas D. e Peucker T. (1978) *Algorithms for the Reduction of the Number of Points Required to Represent a Digitized Line or its Caricature*, The Canadian Cartographer.

Enciclopedia Italiana Grolier (1987) Vol. 4, *Cartografia*, pp 293 – 296.

Goodrich M.T., Tamassia R. (2004) *Data Structures and Algorithms in Java*, Fourth edition, Wiley

IGM (2004) *Norme e Segni convenzionali per la realizzazione dei fogli della Carta d'Italia alla scala 1:50 000* Istituto Geografico Militare, Firenze.

Kilpeläinen, T. (2000), *Maintenance of Multiple Representation Databases for Topographic Data* The Cartographic Journal Vol. 37, No 2, 101-107p.

Lamy S., Ruas A., Demazeau Y., Jackson M., Mackaness W. e Weibel, R. (1999) *The application of Agents in Automated Map Generalisation* Proceedings of 19th ICA meeting. Ottawa pp 160-169.

Mackaness W. A. (2007) *Understanding Geographic Space*, in *Generalisation of Geographic Information: Cartographic Modelling and Applications*, Mackaness, W. A., Ruas, A. and Sarjakoski, L. T. (eds.), pp. 1–10, Elsevier, Oxford.

McMaster R. B. e Shea K. S. (1992) *Generalization in Digital Cartography*, Association of American Geographers, Washington D.C.

Meyen E. (1973) *Multilingual dictionary of technical terms in cartography*, International Cartographic Association, Steiner, Wiesbaden.

Nickerson B. G. e Freeman H. R. (1986) *Development of a rule-based system for automatic map generalization* Proceedings of the Second International Symposium on spatial data Handling, Seattle, Washington.

Robinson A. H., Sale, R. e Morrison J. L. (1978) *Elements of Cartography*. New York: Wiley & Sons.

Savino S. (2007) *Il processo di generalizzazione cartografica: dalla Carta Tecnica Regionale al DB25 IGM*, Tesi di laurea specialistica, Università degli Studi di Padova, 2007.

Vivid Solutions (2003), *JTS Topology Developer'S Guide*, Version 1.4 URL: <http://www.vividsolutions.com/jts/jtshome.htm>

Weibel R. e Dutton G. (1999) *Generalising spatial data and dealing with multiple representations*, in *Geographic information systems: Principles, techniques, applications and management*, (eds), Second Edition, Cambridge, GeoInformation International, pp. 125-155.