

Codifica di dati territoriali:
il sistema DIME centrale e suo confronto
con altri tipi di codifica.

L. Maffei, P. Mogorovich, M. Venturini

Rapporto interno C85-10

Istituto CNUCE - Pisa

Copyright Settembre 1985



INDICE

1. INTRODUZIONE	pag. 4
2. IL DATO TERRITORIALE	pag. 6
3. SISTEMI DI GESTIONE DI DATI TERRITORIALI	pag. 9
3.1 - Introduzione	pag. 9
3.2 - Acquisizione e correzione	pag. 10
3.3 - Gestione	pag. 17
4. IL DIME CENTRALE	pag. 22
4.1 - Introduzione	pag. 22
4.2 - L'acquisizione	pag. 23
4.3 - La compensazione	pag. 23
4.4 - La chiusura	pag. 25
5. LO SFOLTIMENTO	pag. 28
5.1 - Alcuni metodi	pag. 28
5.2 - Il nostro metodo	pag. 29
5.3 - I tratti "fantasma"	pag. 34
6. LA COMPENSAZIONE	pag. 36
6.1 - Il metodo del baricentro	pag. 36
6.2 - Un problema nascosto	pag. 38
6.3 - Il nostro metodo	pag. 41
6.4 - I punti "pendenti"	pag. 43

7. LA CHIUSURA	pag. 45
7.1 - Suddivisione del problema e soluzioni ...	pag. 45
7.2 - Problema P1	pag. 46
7.3 - Problema P2	pag. 55
7.4 - La nostra soluzione	pag. 59
8. APPENDICE	pag. 65
8.1 - Le prove della fase di compensazione	pag. 65
8.2 - Le prove della fase di chiusura	pag. 66
8.3 - Le tabelle	pag. 68
Bibliografia	pag. 70
Indice delle figure	pag. 71

1. INTRODUZIONE

Il problema affrontato riguarda una serie di operazioni da applicare ai dati di tipo territoriale, per renderli coerenti e quindi gestibili con tecniche informatiche nell'ambito di sistemi informativi territoriali.

Nel capitolo 2. diamo la definizione di dato territoriale, come elemento composto da un'informazione geografica e da una descrittiva riferita ad una struttura territoriale.

Nel capitolo 3. presentiamo due sistemi di rappresentazione di dati territoriali areali: la struttura per aree e la struttura DIME originale, soffermandoci in particolare su quest'ultima.

Nel capitolo 4. esporremo la struttura DIME centrale

presentando le fasi per la sua gestione: l'acquisizione, la compensazione e la chiusura.

Nei capitoli 5. e 6. affronteremo il problema di togliere ridondanza ai tratti, eliminando alcuni punti considerati non portatori di informazione (operazione di sfoltimento), e quello di rendere numericamente uguali gli estremi dei tratti contigui (operazione di compensazione).

Oltre ad alcune soluzioni già note per tali problemi, proporremo una nostra soluzione, mostrando alcuni inconvenienti che ci siamo trovati a dover risolvere.

I problemi che affrontiamo nel capitolo 7. sono:

- 1 - determinare, per ogni punto isolato, i tratti di confine orientati dell'area individuata da tale punto;
- 2 - attribuire ad ogni tratto il codice destro/sinistro, in base a quanto ottenuto al punto precedente.

Esporremo alcune soluzioni già note di tali problemi e la soluzione da noi adottata.

2. IL DATO TERRITORIALE

Per dato territoriale (D.T.) si intende un'informazione che ha almeno un riferimento sul territorio.

Il D.T. puo' essere visto come suddiviso in due tipi di informazione: una di tipo descrittivo e una di tipo geografico. L'informazione descrittiva e' composta da una serie di valori numerici cui e' associato un significato; l'informazione geografica e' un insieme di valori numerici che localizza tramite punti, linee, aree una parte del territorio.

Osserviamo che quasi tutto e' un D.T., come per esempio la popolazione di una regione, la destinazione d'uso di un'area secondo un piano regolatore, il bilancio di un comune.

Le informazioni descrittive sono dedotte da archivi di vario tipo (statistico, amministrativo, censuario, fisico, ecc.).

Il legame tra l'informazione descrittiva e l'informazione geografica, esistente nella definizione di D.T., e' la base per ottenere informazioni relative a particolari aree di un territorio, calando nella struttura geografica (che e' solo un contenitore!) le informazioni tematiche via via necessarie.

Possiamo quindi dire che intendiamo per D.T. non un archivio, ma un insieme di archivi, ciascuno dei quali contiene al suo interno informazioni omogenee. Tali informazioni sono formate da una parte descrittiva e da una geografica. Quest'ultima, se correttamente codificata, e' compatibile verticalmente su tutti i singoli archivi, e permette un confronto della parte descrittiva di archivi diversi (fig.2.1).

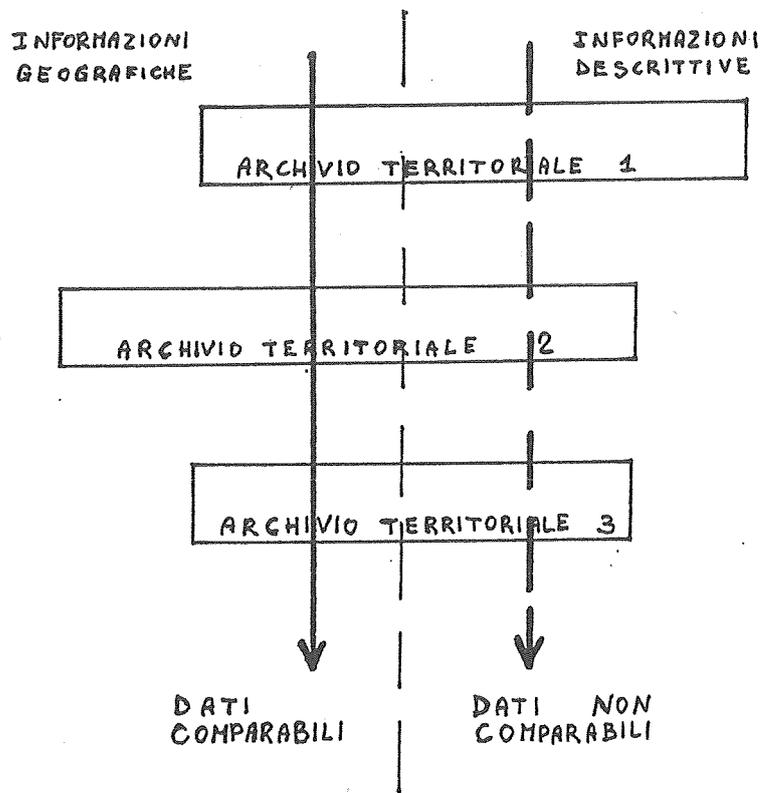


fig.2.1 - Il dato territoriale e' formato da informazioni descrittive e da informazioni geografiche. Solo grazie a quest'ultime e' possibile il confronto tra archivi territoriali eterogenei.

3. SISTEMI DI GESTIONE DI DATI TERRITORIALI .

3.1 - Introduzione.

Per la soluzione di problemi relativi al controllo del territorio, quali ad esempio la gestione di PRG, la realizzazione di inventario forestale, la progettazione di parchi regionali, ecc. sono necessarie molte informazioni ed elaborazioni; tali elaborazioni risultano piu' veloci, precise e sintetiche se si usano tecniche di elaborazione automatica: ad esempio e' stato possibile trasformare dati tabellari in immagini (grafici, mappe), offrendo cosi' un modo piu' efficace e sintetico di vedere per decidere.

Come esempio possiamo citare la sperimentazione di uso di computer graphics condotta dall'Ilres presso la Selesta Cisinet in occasione della progettazione di due sistemi di parco regionale (cfr. [2] pagg. 283-292), e la costruzione di un modello tridimensionale a partire da dati cartografici come ausilio alla progettazione urbanistica (cfr. [2] pagg. 303-311).

Un sistema di gestione di dati territoriali dovrà prevedere, sia per quanto riguarda l'informazione geografica, che quella descrittiva, le seguenti operazioni: acquisizione, correzione, aggiornamento, gestione e uscita.

Tali operazioni sono molto diverse nei due casi.

3.2 - Acquisizione e correzione.

L'acquisizione dei D.T. proviene per lo più da un numero molto grande di fonti, disomogenee e spesso poco attendibili. Perciò il D.T. una volta acquisito deve essere corretto prima di essere elaborato e restituito.

Concentreremo la nostra attenzione sulla parte geografica del D.T..

L'acquisizione dei dati geografici che prenderemo in considerazione, viene compiuta tramite digitalizzatori da carte geografiche in scala da 1:100.000 a 1:5.000 o maggiori. Naturalmente, le coordinate di un punto acquisito attraverso l'uso di un digitalizzatore, devono essere trasferite dal sistema di riferimento cartesiano ortogonale del tavolo di digitalizzazione (si parla di coordinate strumentali) in un sistema di riferimento geografico scelto dall'utente: geografico, UTM, Gauss-Boaga, catastale, ecc. (si parla di coordinate reali) (cfr. [3] pagg. 3-10).

I problemi che si incontrano durante l'acquisizione

possono dipendere dalla diversità di un D.T. all'altro e dagli obiettivi che vogliamo porci sulla struttura dati che stiamo definendo.

La parte geografica del D.T., come del resto quella descrittiva, può essere molto diversa da un D.T. all'altro, per esempio per i seguenti aspetti:

- 1 - natura del dato: può essere rappresentato con un punto, una linea, una superficie a maglia regolare, una superficie irregolare (un punto può rappresentare un comune, un pozzo, ecc.; una linea può rappresentare un fiume, una strada, ecc.; una superficie può rappresentare l'area di un comune, o di una regione, ecc.);
- 2 - il sistema di coordinate adottato, che varia per motivi tecnici e storici da un dato all'altro (geografico, UTM, Gauss-Boaga, catastale, ecc.);
- 3 - la scala con cui sono rappresentati i dati, che in una gestione digitale appare come una caratteristica di precisione del dato, piuttosto che come una "dimensione" del disegno.

Gli obiettivi da tenere presenti nel definire la struttura di dati geografici sono la flessibilità della struttura ai diversi tipi di operazioni (aritmetiche, topologiche, insiemistiche), la trattabilità informatica, la trasportabilità in altri tipi di struttura, la chiusura

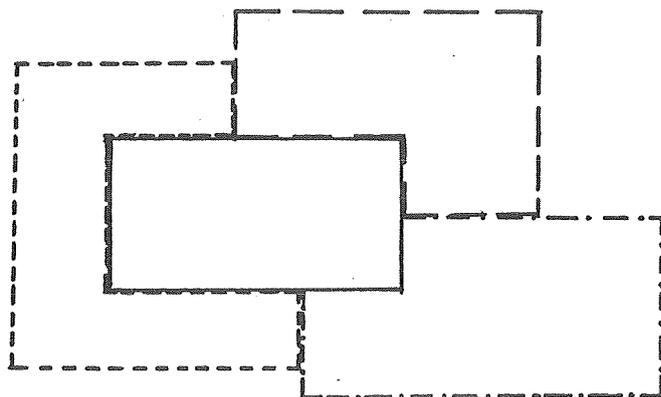
a possibili sviluppi, ecc..

Per chiarire i problemi che nascono al momento dell'acquisizione di un dato geografico, supponiamo di voler digitalizzare i confini comunali di una Regione. Possiamo procedere usando le due seguenti tecniche:

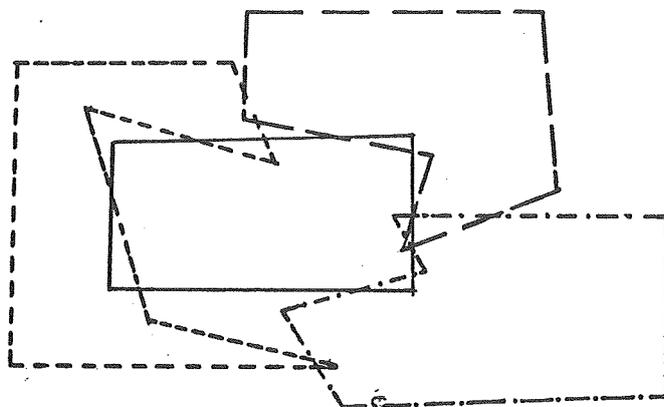
- 1 - per ogni Comune si acquisisce il perimetro completo (acquisizione per aree);
- 2 - si acquisisce un insieme di confini orientati che delimitano coppie di aree, associando ad ogni confine il codice dell'area destra e quello di sinistra (DIME originale).

- Aree.

L'acquisizione per aree e' quella concettualmente piu' semplice: dovendo gestire superfici di territorio acquisisco direttamente per aree; cosi' un'area e' memorizzata come siamo sempre abituati a vederla, ed e' semplice per chi acquisisce nel senso che gli inconvenienti, derivati dall'uso di questa tecnica, consistono, quasi esclusivamente, nella duplicazione (per giunta non esatta) di tratti acquisiti (fig.3.1).



a)



b)

fig.3.1 - a) La situazione reale.
b) La situazione dopo la digitalizzazione.

- DIME originale.

L'acquisizione in DIME originale e' piu' pesante per l'operatore che acquisisce: ogni volta che questi digitalizza un tratto deve preoccuparsi di assegnargli il codice destro ed il codice sinistro; cio' e' causa di una maggior fonte di errori, quali per esempio:

- i) sbaglio nell'assegnazione di un codice (destro o sinistro);
- ii) inversione del codice destro con il sinistro;
- iii) dimenticanza di un tratto;
- iv) duplicazione di un tratto.

Tuttavia quest'ultima tecnica, oltre a presentare vantaggi nelle fasi successive di elaborazione, permette di innescare procedure di correzione degli errori che eliminano oltre al problema principale della prima tecnica, ovvero la duplicazione di linee, gli altri problemi succitati.

Due procedure che permettono di rilevare, e in parte correggere automaticamente, errori dei tipi suddetti, sono:

- a) validazione di "ciclo sugli estremi",
- b) validazione per "ricostruzione poligono",

che sono descritte in dettaglio in [3] pagg. 18-19.

Osserviamo come un errore di tipo ii) possa essere

corretto automaticamente da queste procedure, mentre gli errori di tipo i), iii) e iv) sono soltanto riconosciuti.

Esempio di digitalizzazione DIME originale, con esemplificazione degli errori.

Si vuole digitalizzare la regione di fig.3.2, dove A, B, C e D sono Comuni. L'operatore digitalizza ciascun tratto (le frecce di fig.3.3 indicano il verso di digitalizzazione) e gli attribuisce il codice destra/sinistra. Possiamo riassumere quanto fatto dall'operatore con la tabella 1.

Appare qui evidente come una dimenticanza di un tratto venga scoperta dalla procedura "validazione di ciclo sugli estremi" (fig.3.4), o come un errore sul codice sia scoperto dalla procedura di "validazione per ricostruzione poligono".

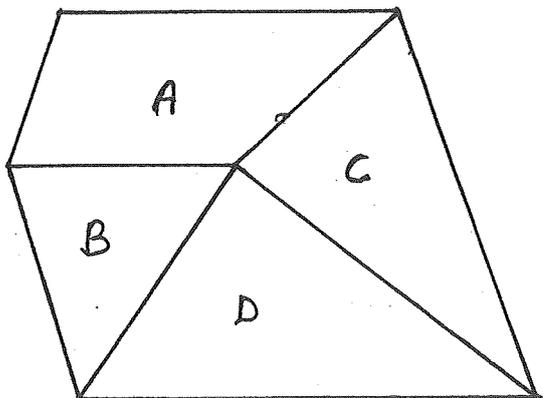


fig.3.2

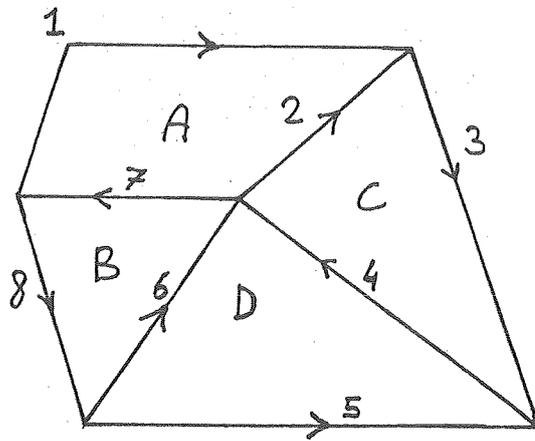


fig.3.3

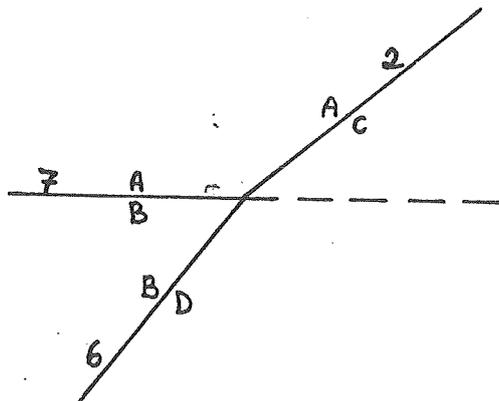


fig.3.4

3.3 - Gestione.

- Aree.

E' chiaro che se si acquisisce "per aree" la gestione sara' "per aree". Ci sembra importante far notare come la struttura per aree risulti troppo povera e poco maneggevole, a causa della rigidita' del dato geografico: il perimetro di un Comune non e' visto come un insieme di tratti che confinano con altri Comuni, ma un'informazione indivisibile associata solo al Comune che rappresenta.

Con questa tecnica di gestione del D.T. risultano difficili operazioni quali: aggregazioni di aree (dato l'archivio dei confini comunali, trovare il confine della Provincia P), restituzione (il plotter disegnera' due volte il confine tra due Comuni).

Persino il calcolo della superficie di un'area richiede maggiori operazioni con questa struttura rispetto a quella DIME. Infatti, riferendoci alla fig.3.3, possiamo osservare:

- se ho la struttura DIME, per ogni tratto, calcolo una volta per sempre il proprio contributo alla superficie $S(nt)$, dove nt e' il numero del tratto; allora il valore dell'area, ad esempio del Comune A, sara' data da $S(1)-S(2)+S(7)$;
- se ho la struttura per aree, per ogni Comune che considero, calcolo la superficie, integrando la spezzata

che delimita il Comune; e' chiaro che, in questo caso, per i tratti di confine ripetero' piu' volte lo stesso calcolo.

Questo vale a maggior ragione nel caso di aggregazione estemporanea di aree elementari in aree di livello gerarchico superiore.

- DIME originale.

Come gia' accennato precedentemente, la struttura DIME originale, nonostante sia nata sostanzialmente come strumento per l'acquisizione, si presenta interessante anche in termini gestionali per i seguenti motivi:

- 1 - e' una struttura per tratti, e tramite opportuna sostituzione dei codici destra/sinistra con altri codici, puo' divenire una codifica di reti (cfr. 4), mantenendo cosi' uniformita' di formati records;
- 2 - si presta ottimamente a gestire aree con diversi livelli di aggregazione;
- 3 - si presta ad elaborazioni di tipo matematico (per esempio calcolo di superfici, verifica punto interno;

ecc.).

Esempi con riferimento alla fig.3.3:

- i) il perimetro del comune A si ottiene dalla tabella 1 prendendo i tratti che contengono il codice A a sinistra o a destra;
- ii) se la provincia P e' formata dai comuni A e C, per avere il suo perimetro basta prendere dalla tabella 1 i tratti che contengono solo A o solo C.

Per dare al lettore una visione piu' completa di questa importante tecnica, dalla quale e' nata quella che illustreremo nel prossimo capitolo e sulla quale abbiamo operato, esponiamo brevemente la struttura del DIME originale, rimandando a [1] per una esposizione piu' dettagliata.

- La struttura del DIME originale.

La struttura base e' un tratto che collega due punti (nodi), dai quali si dipartono, a loro volta, due o piu' tratti. Il tratto e' formato dalle coordinate di una serie di punti P_1, P_2, \dots, P_N e da altre informazioni aggiuntive.

Si rivela opportuno codificare l'arco su due livelli:

livello superiore, livello inferiore (fig.3.5). Il livello inferiore contiene tutta la descrizione geometrica (i punti da P1 a PN), mentre quello superiore contiene tutte le informazioni che possono servire nella gestione ed in alcune operazioni matematiche piu' comuni. Questa struttura e' motivata dal fatto che gran parte delle operazioni, in cui il dato grafico e' coinvolto nelle procedure di acquisizione e gestione, richiedono l'utilizzo di informazioni comprese nel livello superiore, mentre sono abbastanza poche quelle in cui e' coinvolto il livello inferiore, se si esclude, naturalmente, la restituzione.

Inoltre il livello superiore puo' avere "formati records fissi", a differenza di quello inferiore, e si presta quindi bene a operazioni di accesso diretto tramite i comuni linguaggi di programmazione.

N. del TRATTO	P1	PN	N. di PUNTI	PUNTATORE	RETTANGOLO D'INTERVENTO	INFORMAZIONI AUSILIARIE
---------------	----	----	-------------	-----------	-------------------------	-------------------------

P2	P3	PN-2	PN-1
----	----	-----------	------	------

fig.3.5 - Descrizione dei campi:

N. di TRATTO: e' il numero d'ordine del tratto;
 P1 e PN : sono i punti iniziale e finale del tratto;
 N. di PUNTI : e' il numero di punti di cui e' formato il tratto;
 PUNTATORE : e' utilizzato per la ricerca dei dati del livello inferiore;
 RETTANGOLO D'INTERVENTO: indica, di un tratto, le coordinate massime e minime;
 INFORMAZIONI AUSILIARIE : sono informazioni caratterizzanti il tratto e utili per la gestione, come:
 - la scala di acquisizione;
 - la lunghezza;
 - il contributo alla superficie.

N. di TRATTO	N. di PUNTI	P1	P2	PN-1	PN
--------------	-------------	----	----	-----------	------	----

fig.4.1

4. IL DIME CENTRALE

4.1 - Introduzione.

Da quanto detto, il DIME originale e' molto potente per la gestione, ma risulta gravoso per l'operatore che acquisisce. E' nata quindi l'esigenza di trovare una struttura meno pesante per l'operatore. Questa tecnica, detta DIME centrale, ha la seguente struttura:

- 1 - si acquisiscono due archivi: un archivio di tratti (come nel DIME originale, ma senza codice destro/sinistro) e un archivio di punti (fase di acquisizione);
- 2 - si correggono gli errori "di estremo" sui tratti acquisiti (fase di compensazione);
- 3 - si attribuisce il codice destro/sinistro ad ogni tratto (fase di chiusura).

4.2 - L'acquisizione.

L'operatore deve digitalizzare tratti, intendendo per tratto una linea di confine tra due aree; ogni tratto deve essere digitalizzato una sola volta; si e' cosi' acquisito l'archivio dei tratti; un record di tale archivio e' descritto in fig.4.1.

L'operatore deve poi digitalizzare punti (chiamati punti isolati o centrali), cioe' per ogni area si digitalizza un solo punto interno a questa area; si acquisisce cosi' l' archivio dei punti.

L'acquisizione dei tratti e' piu' semplice del DIME originale, in quanto non bisogna attribuire il codice destro/sinistro ad ogni tratto. Rispetto al DIME originale abbiamo in piu' l'acquisizione dei punti centrali, ma cio' non e' pesante per l'operatore, essendo questa un'operazione semplice indipendente dalla prima, ed inoltre spesso il punto centrale e' un punto significativo dell'area che si sta digitalizzando.

4.3 - La compensazione.

Come gia' detto, i tratti digitalizzati sono soggetti ad errori: due tratti che sulla carta sono contigui possono

non risultare tali dopo l'acquisizione, perche' l'estremo comune ad entrambi puo' risultare numericamente diverso nei due tratti. Inoltre, quando l'operatore al digitalizzatore preleva le coordinate di un tratto, possono essere campionati piu' punti dello stretto necessario per la successiva restituzione grafica del tratto stesso. In certi casi le coordinate di due punti successivi possono essere addirittura uguali.

Il primo problema, che e' dovuto al modo con cui si acquisisce, causa degli inconvenienti al momento della fase di chiusura, dove ha molta importanza che due tratti, contigui nella realta', siano riconosciuti tali dalla macchina. La soluzione di questo problema e' pertanto vitale ai fini di una gestione efficiente e sicura.

Il secondo problema, che potrebbe sembrare solo un inconveniente di precisione al momento della restituzione, puo' causare (per gli arrotondamenti della macchina) situazioni di overflow durante operazioni piu' semplici e frequenti, quali, ad esempio, il calcolo dell'angolo formato da due semirette, o la distanza di un punto da un segmento; tali operazioni sono richieste, per esempio, per la soluzione del problema precedente.

Sono quindi necessarie delle procedure (procedure di sfoltimento e di compensazione) che cerchino di eliminare gli inconvenienti citati.

4.4 - La chiusura.

Ogni punto isolato P individua univocamente un'area: questa e' l'area individuata dal poligono Q , avente il perimetro piu' vicino a P tra tutti i poligoni che contengono il punto P , meno l'area di ogni altro poligono contenuto in Q (i poligoni contenuti in Q rappresentano delle isole); v. fig.4.2.

Possiamo imporre che ogni confine di un'area sia orientato in modo da lasciare l'area sempre sulla destra (fig.4.3), quindi ogni tratto che appartiene ai poligoni di un'area puo' essere considerato come tratto orientato: l'orientazione e' positiva se il tratto e' stato digitalizzato secondo l'orientazione del poligono, negativa altrimenti.

Se per ogni punto isolato conosciamo i tratti di confine orientati dell'area da esso individuata, possiamo attribuire ad ogni tratto il codice destro/sinistro. Infatti sara' sufficiente cercare quali sono i poligoni, individuati da un punto isolato, a cui il tratto appartiene: se il tratto appartiene al poligono individuato dal punto P ed il suo segno e' "+" (cioe' e' stato digitalizzato nel verso di percorrenza del poligono), allora il codice destro del tratto e' P ; se il suo segno e' "-" allora il codice sinistro del tratto e' P .

Quindi il problema di attribuire il codice

destro/sinistro di un'area e' ricondotto ad un problema di chiusura: per ogni punto isolato trovare i tratti di confine orientati dell'area individuata dal punto.

E' opportuno mettere in evidenza che il punto di arrivo del DIME centrale e' la creazione dell'archivio dei tratti cosi' come venivano acquisiti col DIME originale.

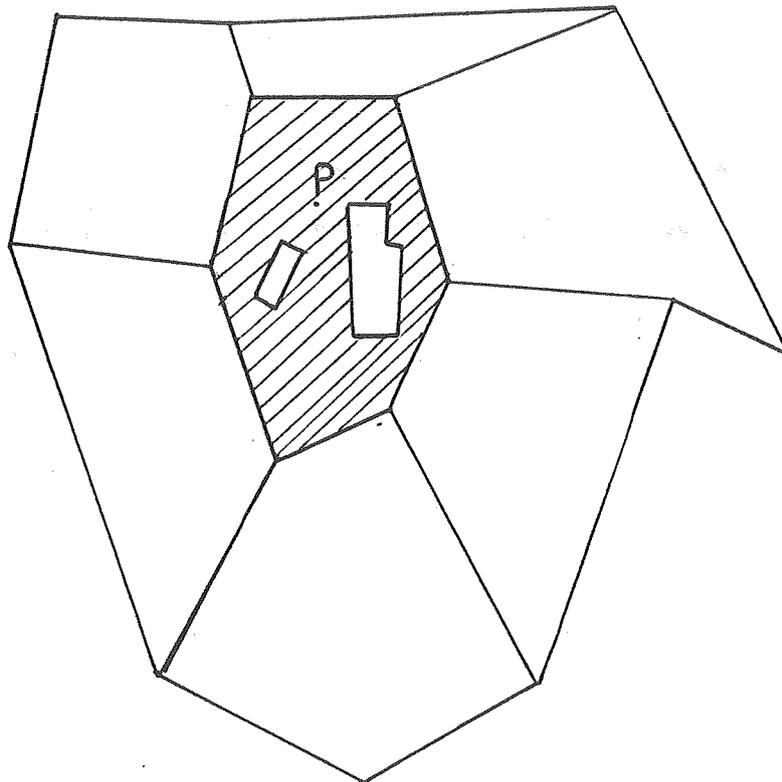


fig.4.2

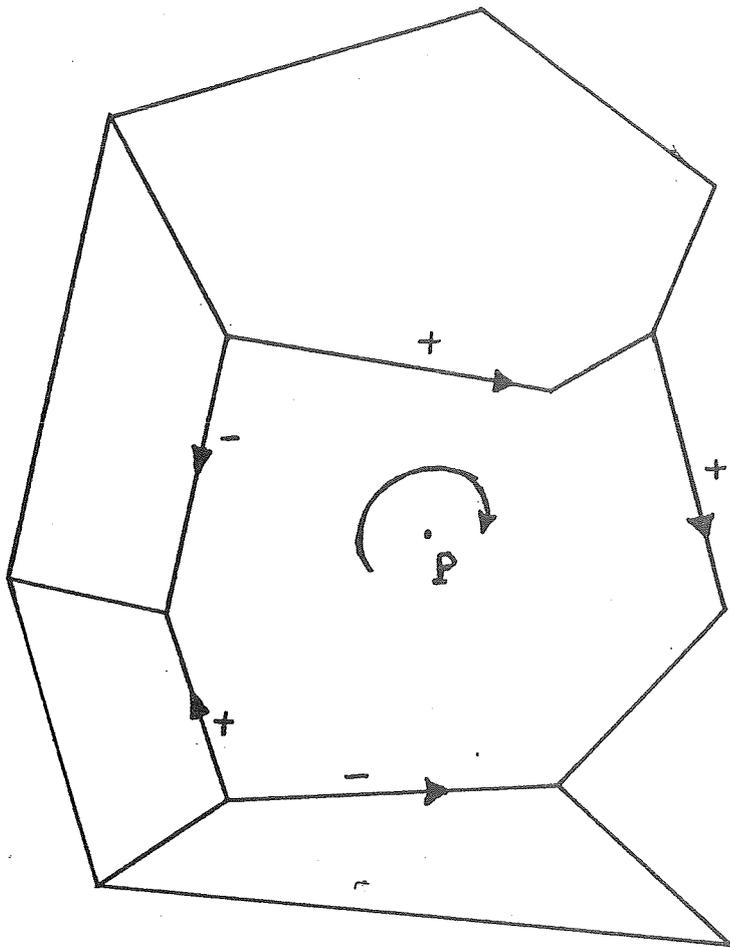


fig.4.3

5. LO SFOLTIMENTO

5.1 - Alcuni metodi.

Ci sembra importante far notare a questo punto, che la soluzione di questo problema, oltre a correggere gli inconvenienti a cui si accennava nel paragrafo 4.3 del capitolo precedente, riduce sensibilmente i records degli archivi dei tratti. Cio' non e' del tutto irrilevante se si pensa che la loro lunghezza puo' essere dell'ordine di decine di migliaia di records; cosi' una riduzione del 50% permette di ottenere una maggiore velocita' di esecuzione dei programmi, dato che le operazioni di I/O sono abbastanza costose in questo senso. Dati statistici mostrano che buone compensazioni possono ridurre fino all'80% gli archivi di tratti acquisiti con la tecnica "a tempo" o "a spazio/tempo", senza degradare il dato iniziale.

Osserviamo innanzitutto che l'operazione di riduzione delle coordinate necessarie a rappresentare un elemento

grafico puo' essere effettuata piu' facilmente su elementi di tipo lineare, come tratti o curve, piuttosto che su elementi di tipo areale.

Uno dei metodi di sfoltimento piu' usati, consiste nel seguente procedimento:

- esaminare una serie di punti rappresentanti una sequenza di tratti lineari e nell'eliminare tutti i punti della catena che causano una deflessione della poligonale, rispetto alla retta passante per i due punti precedenti, minore di un valore angolare prefissato; sono conservati tutti i punti che, pur causando una deflessione minore di quella considerata limite, distino dall'ultimo punto conservato piu' di un fissato valore, oppure che siano stati "marcati" con un codice particolare.

Per maggiori dettagli su questo metodo, si veda [3] pagg. 12-14.

5.2 - Il nostro metodo.

Supponiamo che l'errore numerico massimo, di cui sono affetti i tratti digitalizzati (che, come sappiamo, e' in

parte derivato dal supporto cartografico, ed in parte e' dovuto all'operatore), sia una quantita' ϵ .

Cio' vuol dire che, se due punti consecutivi di un tratto distano meno di ϵ , non riusciamo a distinguerli sulla carta; pertanto uno di essi puo' essere tolto. Inoltre, se, dati tre punti consecutivi A, B e C, B dista dal segmento AC meno di ϵ , significa che B puo' essere tolto, perche' non e' distinguibile sulla carta dal segmento AC.

Osserviamo che il nostro scopo e' compensare i tratti (cioe' rendere numericamente uguali gli estremi di due tratti contigui nella realta'); e' bene allora non modificare gli estremi dei tratti, poiche' si spera che l'operatore abbia cercato di avvicinarsi il piu' possibile a questi. Quindi, in questa fase, gli estremi di ogni tratto devono essere considerati punti "intoccabili". Dobbiamo innescare un algoritmo che calcoli gli altri punti del tratto a partire da questi.

La procedura di sfoltimento e' suddivisa in due passi. I dati sono un tratto di N punti P_1, \dots, P_N . Con F indicheremo un punto candidato "intoccabile"; con P e Q punti del tratto; una PILA conterra' alla fine di ogni passo i punti eliminabili; useremo un VETTORE ausiliario per memorizzazioni intermedie.

- Passo A.

Riduciamo i punti di ogni tratto in modo che: per ogni punto P , appartenente al tratto, il cerchio di centro P e raggio ϵ intersechi il tratto solo in P .

Diamo di seguito lo schema logico dell'algoritmo:

- A0 - inizializziamo PILA;
- A1 - poniamo $P = P_1$;
- A2 - poniamo $F = P$;
- A3 - poniamo $P =$ punto seguente P nel tratto;
- A4 - se P e' "fuori" del tratto, allora andiamo ad A7;
- A5 - calcoliamo la distanza D fra F e P ;
- A6 - se D e' minore di ϵ ,
 - allora se $P = P_N$,
 - allora inseriamo F in PILA ed usciamo;
 - altrimenti inseriamo P in PILA e ripartiamo da A3;
 - altrimenti ripartiamo da A2;
- A7 - togliamo i punti inseriti in PILA dal tratto e usciamo.

- Passo B.

Riduciamo i punti di ogni tratto in modo che: dati tre punti consecutivi A , B e C , B disti dal segmento AC piu' di ϵ (la fig.5.1 mostra un esempio di applicazione del passo

B).

Diamo di seguito lo schema logico dell' algoritmo:

B0 - inizializziamo PILA;

B1 - poniamo $P = P_1$;

B2 - poniamo $F = P$;

B3 - se F e' il penultimo o ultimo punto del tratto, allora andiamo a B9;

B4 - poniamo $P =$ punto seguente P nel tratto;

B5 - se P e' l'ultimo punto del tratto, allora andiamo a B9;

B6 - poniamo $Q =$ punto seguente P nel tratto;

B7 - calcoliamo le distanze di P e dei punti inseriti nel VETTORE ausiliario dal segmento FQ ;

B8 - se tutte le distanze calcolate in B7 sono minori di ϵ ,

allora inseriamo P nel VETTORE ausiliario e ripartiamo da B4;

altrimenti inseriamo i punti contenuti nel VETTORE ausiliario in PILA e ripartiamo da B2;

B9 - togliamo i punti inseriti in PILA dal tratto e usciamo.

Osserviamo per chiarezza, che, dati tre punti A , B e C , intendiamo per distanza del punto B dal segmento AC (fig.5.2), il segmento BH nel caso a), il segmento BA nel caso b) e il segmento BC nel caso c).

Se P_5 dista dal segmento P_1-P_6 piu' di ϵ , allora i punti P_2 , P_3 e P_4 vengono inseriti in PILA.

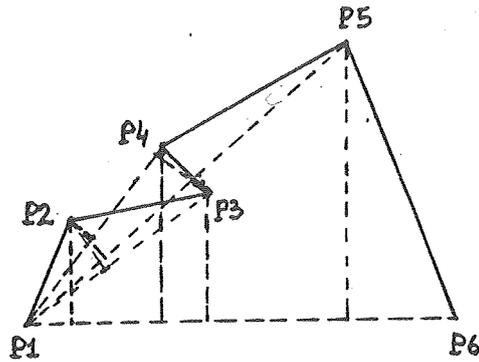


fig.5.1

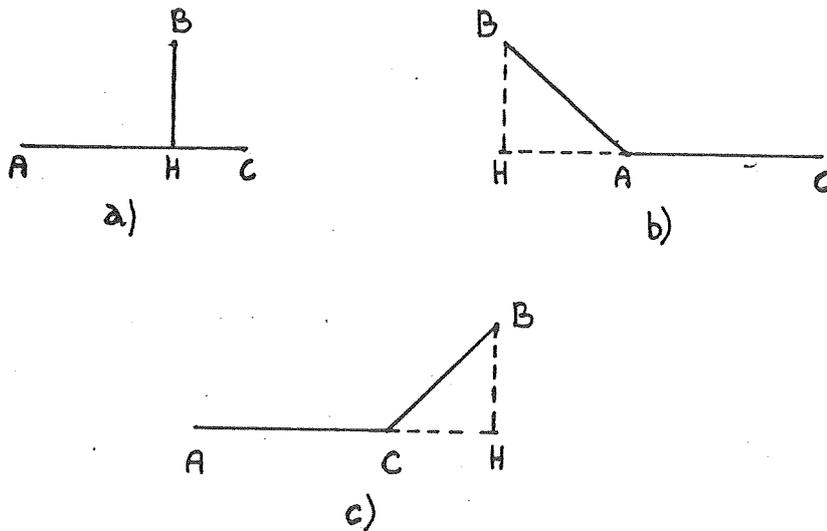


fig.5.2

Si sara' senz'altro notato, che il passo A dell'algoritmo sopra esposto, e' compreso nel passo B; d'altra parte osserviamo che quest'ultimo e' piu' costoso, dovendo calcolare la distanza di un punto da un segmento, rispetto alla distanza fra due punti calcolata col primo passo. Quindi col passo A riduciamo il numero dei punti del tratto e cosi' applicheremo il passo B su un numero minore di punti. L'adozione di questo algoritmo risponde tra l'altro ad una precisa esigenza di calcolo, come descritto nel paragrafo 6.3.

5.3 - I tratti "fantasma".

Puo' accadere, con l'algoritmo di sfoltimento descritto nel paragrafo precedente, che alcuni tratti siano ridotti ad un solo punto: si parla in tal caso di tratti "fantasma". Quando l'algoritmo di sfoltimento incontra un tratto "fantasma", il tratto viene memorizzato nel file di output cosi' come era stato digitalizzato, senza alcuna modifica.

Per decidere sulla sorte di un tratto "fantasma", abbiamo ritenuto opportuno l'uso di un video grafico interattivo, su cui visualizzare la situazione. Il tratto in questione e' rappresentato con un colore diverso da quello usato per gli altri tratti; inoltre ad ogni tratto, disegnato sul video, viene associato il proprio numero

d'ordine e una freccetta che indica il verso di digitalizzazione del tratto (la freccetta esce dal P1 e punta verso il PN). Un menu permette all'operatore di prendere le decisioni appunto in modo interattivo.

Osserviamo che i punti di un tratto "fantasma" sono interni al cerchio di centro P1 e raggio ξ ; quindi il tratto "fantasma" deve essere considerato un errore dell'operatore anche se dal disegno del video sembrerebbe ben digitalizzato. Pertanto il tratto "fantasma" in genere deve essere tolto.

Questa conclusione e' rafforzata, se si tiene conto del metodo di compensazione usato (v. paragrafo 6.3); infatti con tale metodo il tratto "fantasma" sarebbe ridotto ad avere gli estremi identici; cio', non solo modifica sostanzialmente il tratto (che diventerebbe un'isola), ma puo' essere causa di divisioni per zero in alcune operazioni piu' semplici. Una possibile soluzione e' quella di usare, per lo sfoltimento e per la compensazione, due valori di errore (ξ) diversi (v. paragrafo 6.4).

6. LA COMPENSAZIONE

6.1 - Il metodo del baricentro.

In questa fase dobbiamo rendere numericamente uguali gli estremi dei tratti che sono contigui nella realta'; estremi, che, per errori manuali dell'operatore, dopo la digitalizzazione possono essere numericamente diversi.

Un metodo usato per risolvere questo problema e' quello del baricentro (fig.6.1). Con questo metodo dobbiamo, dato un estremo P di un tratto:

- ricercare tutti i tratti che hanno un estremo distante meno di ϵ da P;
- calcolare il baricentro B dei punti estremi trovati;
- allungare i tratti in questione fino al punto B.

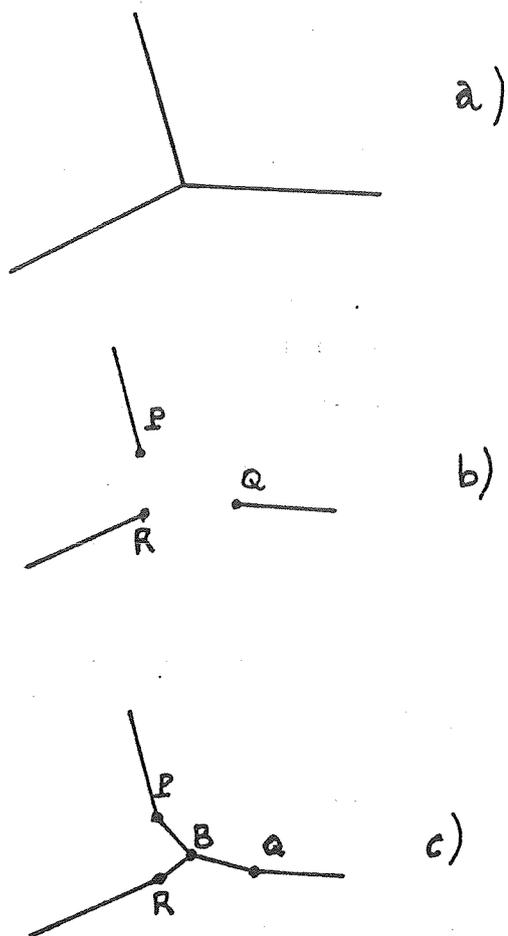


fig.6.1 - a) Situazione reale;
b) Situazione dopo la digitalizzazione;
c) Situazione dopo la compensazione.

6.2 - Un problema nascosto.

Sia la fig.6.2 b) la digitalizzazione della situazione reale rappresentata in fig.6.2 a). Facciamo notare che, se i punti P, Q ed R sono distanti tra loro meno di ϵ , questa situazione puo' presentarsi benissimo e rientra nella tolleranza dell'errore imputabile all'operatore. Nella fig.6.2 c) osserviamo la situazione dopo la compensazione: si e' creata un'area non esistente nella realta' e per di piu' cio' non e' causato da alcun errore.

Osserviamo, infine, che il crearsi di una tale area in fase di compensazione e' completamente trasparente all'operatore; per di piu' essa e' dannosa successivamente in fase di chiusura, a causa del modo con cui siamo costretti a chiudere i poligoni usando la struttura del DIME centrale, cfr. capitolo 7. Occorrono, allora, procedure apposite capaci di riconoscere questo inconveniente.

Ci sembra opportuno far notare come, facendo uso della struttura offerta dal DIME originale (in cui ogni tratto porta con se' l'informazione del codice destro/sinistro), un tale inconveniente puo' essere riconosciuto dalla procedura di validazione di "ciclo sugli estremi" esposta nel paragrafo 3.2 (fig.6.3).

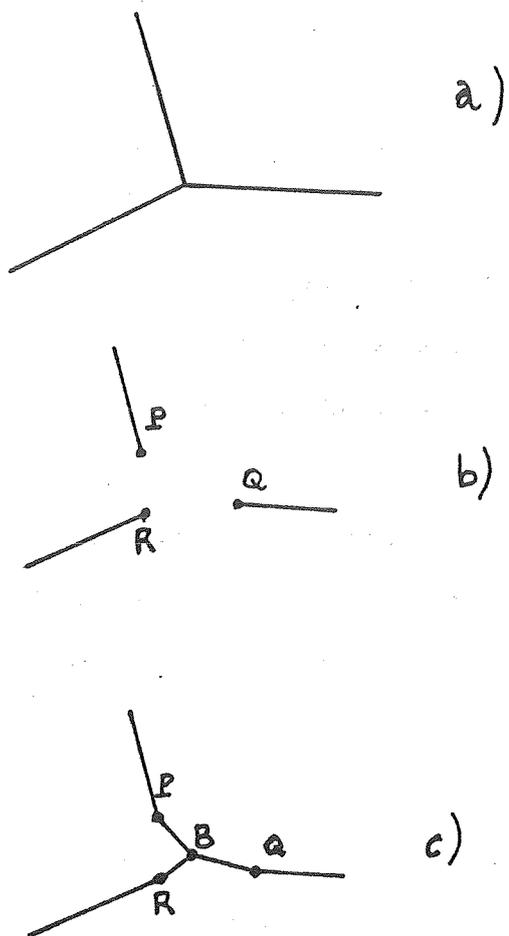


fig.6.1 - a) Situazione reale;
b) Situazione dopo la digitalizzazione;
c) Situazione dopo la compensazione.

6.2 - Un problema nascosto.

Sia la fig.6.2 b) la digitalizzazione della situazione reale rappresentata in fig.6.2 a). Facciamo notare che, se i punti P, Q ed R sono distanti tra loro meno di ϵ , questa situazione puo' presentarsi benissimo e rientra nella tolleranza dell'errore imputabile all'operatore. Nella fig.6.2 c) osserviamo la situazione dopo la compensazione: si e' creata un'area non esistente nella realta' e per di piu' cio' non e' causato da alcun errore.

Osserviamo, infine, che il crearsi di una tale area in fase di compensazione e' completamente trasparente all'operatore; per di piu' essa e' dannosa successivamente in fase di chiusura, a causa del modo con cui siamo costretti a chiudere i poligoni usando la struttura del DIME centrale, cfr. capitolo 7. Occorrono, allora, procedure apposite capaci di riconoscere questo inconveniente.

Ci sembra opportuno far notare come, facendo uso della struttura offerta dal DIME originale (in cui ogni tratto porta con se' l'informazione del codice destro/sinistro), un tale inconveniente puo' essere riconosciuto dalla procedura di validazione di "ciclo sugli estremi" esposta nel paragrafo 3.2 (fig.6.3).

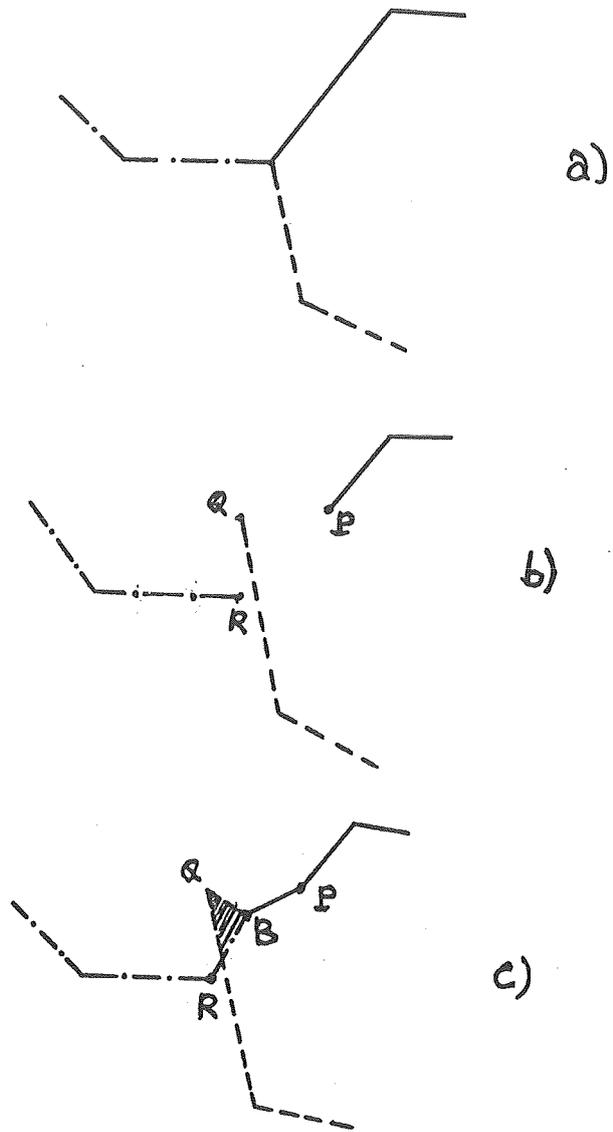


fig.6.2 - a) Situazione reale;
b) Situazione dopo la digitalizzazione;
c) Situazione dopo la compensazione.

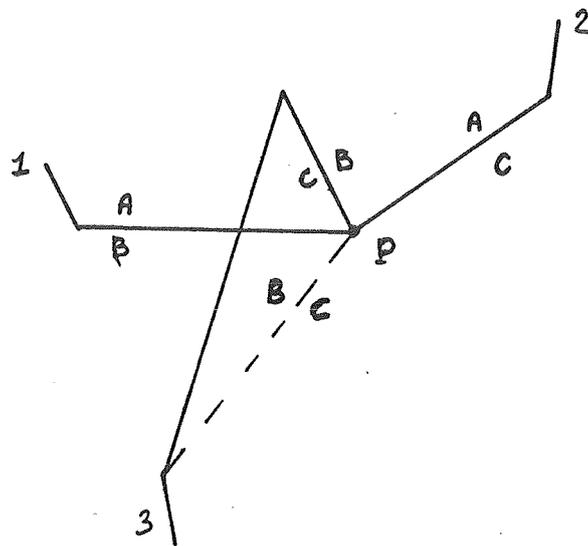


fig.6.3 - E' evidente che il nodo P non "cicli"; inoltre la presenza dei codici "giusti" B e C sul tratto n. 3, ma non al "posto giusto", determinano la presenza di un'area "nascosta".

6.3 - Il nostro metodo.

Il nostro metodo di compensazione si puo' cosi' schematizzare:

- 1 - dato il tratto di numero d'ordine I consideriamo gli estremi P1 e PN;
- 2 - se P1 dista da PN meno di ϵ , allora poniamo P1=PN (il tratto che stiamo considerando rappresenta un'isola);
- 3 - per ogni J maggiore di I consideriamo gli estremi Q1 e QN del tratto di numero d'ordine J;
- 4 - se P1 dista da Q1 meno di ϵ , allora poniamo Q1=P1 e andiamo a 6;
- 5 - se P1 dista da QN meno di ϵ , allora poniamo QN=P1 e andiamo a 7;
- 6 - se PN dista da QN meno di ϵ , allora poniamo QN=PN e ripartiamo da 3;
- 7 - se PN dista da Q1 meno di ϵ , allora poniamo Q1=P1 e ripartiamo da 3.

Osserviamo innanzitutto che questo metodo e' senz'altro piu' veloce di quello del baricentro; inoltre, per la tecnica di sfoltimento usata sui tratti, non puo' crearsi un'area del tipo descritto nel paragrafo precedente, come si puo' notare osservando la fig.6.4.

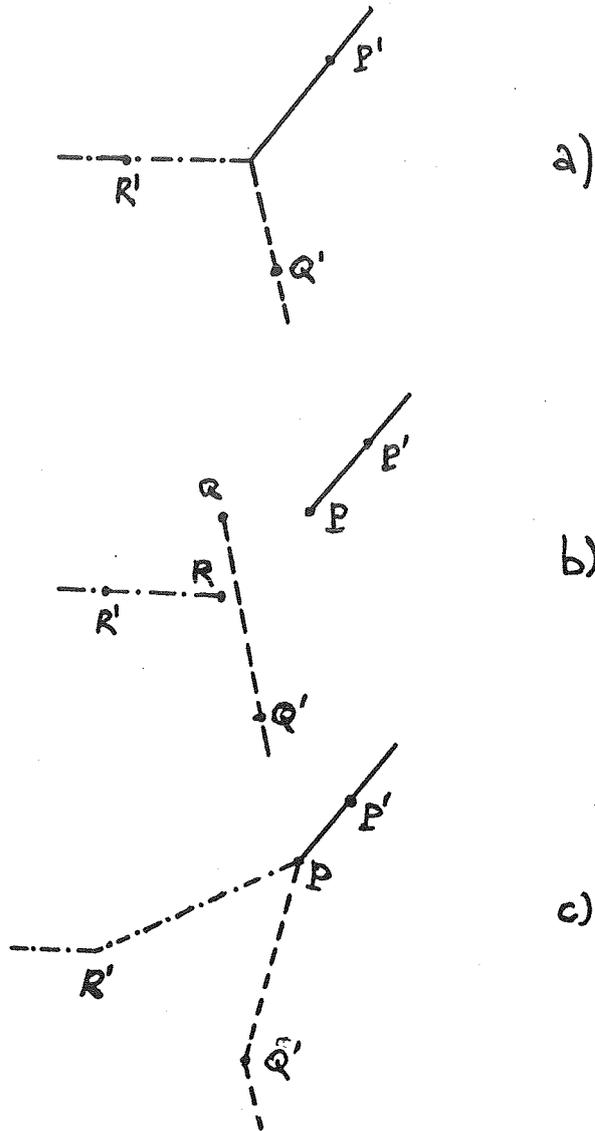


fig.6.4 - a) Situazione reale;
b) Situazione dopo la digitalizzazione;
c) Situazione dopo la compensazione.

Vogliamo infine far notare, come il metodo del baricentro, quando non si presenti l'inconveniente dell'area "nascosta", ripartisca piu' equamente la regione di piano compresa tra i punti P, Q e R, a differenza della nostra soluzione che tende a favorire piu' un poligono rispetto agli altri. Si tenga comunque presente che la regione di piano "in gioco" e' contenuta in un intorno dell'errore ammissibile.

6.4 - I punti "pendenti".

Succede frequentemente che gli errori manuali siano numericamente maggiori dell' ϵ assunto per le procedure di sfoltimento e di compensazione; la possibilita' di far rieseguire le procedure con un ϵ sempre piu' grande e' purtroppo un'illusione, poiche', ad un certo punto, tratti significativi cominciano a ridursi a punti, oppure vengono collegati estremi non identici nella realta'. Pertanto, al termine della procedura che compensa automaticamente i tratti, bisogna attenderci alcuni estremi non collegati; parliamo in questo caso di punti "pendenti".

Per decidere sulla sorte dei punti "pendenti", abbiamo utilizzato, anche in questa occasione, un video grafico interattivo (v. paragrafo 5.3). Un menu permette di

collegare, in modo interattivo, estremi di tratti disegnati sul video, mostrando i risultati della decisione presa e dando l'opportunita' di modificare la decisione stessa, se ritenuta non buona.

A questo punto puo' ripresentarsi l'area "nascosta" di cui si e' parlato nel paragrafo 6.2; ma in questo caso la vediamo (!) direttamente sul video: non possiamo evitare questo inconveniente, ma ne siamo informati, e possiamo correggere i dati in altro modo.

Durante alcune prove, ci siamo imbattuti nell'inconveniente di cui parlavamo all'inizio di questo paragrafo; cioe' dopo aver eseguito la fase di compensazione automatica dei tratti (abbiamo lavorato sui confini comunali della Puglia digitalizzati dall'IGM), abbiamo osservato che qualche tratto veniva ridotto ad un punto ed alcuni estremi venivano collegati, mentre non lo erano nella realta'. Abbiamo quindi pensato, che sia piu' giusto sfoltire con un certo errore e compensare con un errore minore; cio' non esclude ovviamente che si ripresenti l'inconveniente suddetto, ma ne riduce sensibilmente le probabilita'.

7. LA CHIUSURA

7.1 - Suddivisione del problema e soluzioni

Poiche', in generale, le aree che contengono al loro interno delle isole sono, in numero minore di quelle che non le contengono, abbiamo affrontato il problema della chiusura suddividendolo in due problemi distinti:

- P1 Per ogni punto isolato, P , trovare i tratti di confine orientati del poligono Q avente, tra tutti i poligoni che contengono il punto P , il perimetro piu' vicino a tale punto.

- P2 Trovare i tratti di confine orientati dei poligoni interni (poligoni che determinano isole) al poligono Q .

Prima di esporre possibili soluzioni dei problemi $P1$ e $P2$ osserviamo esplicitamente che, in tutto cio' che segue, supponiamo che la compensazione sul file tratti sia gia' stata fatta (v. capitolo 6), e chiameremo il file che

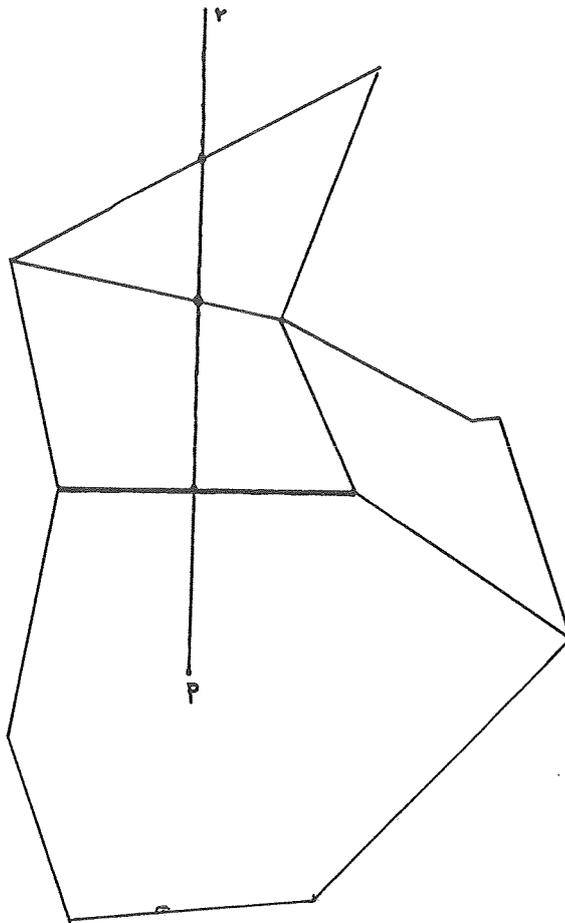


fig.7.2

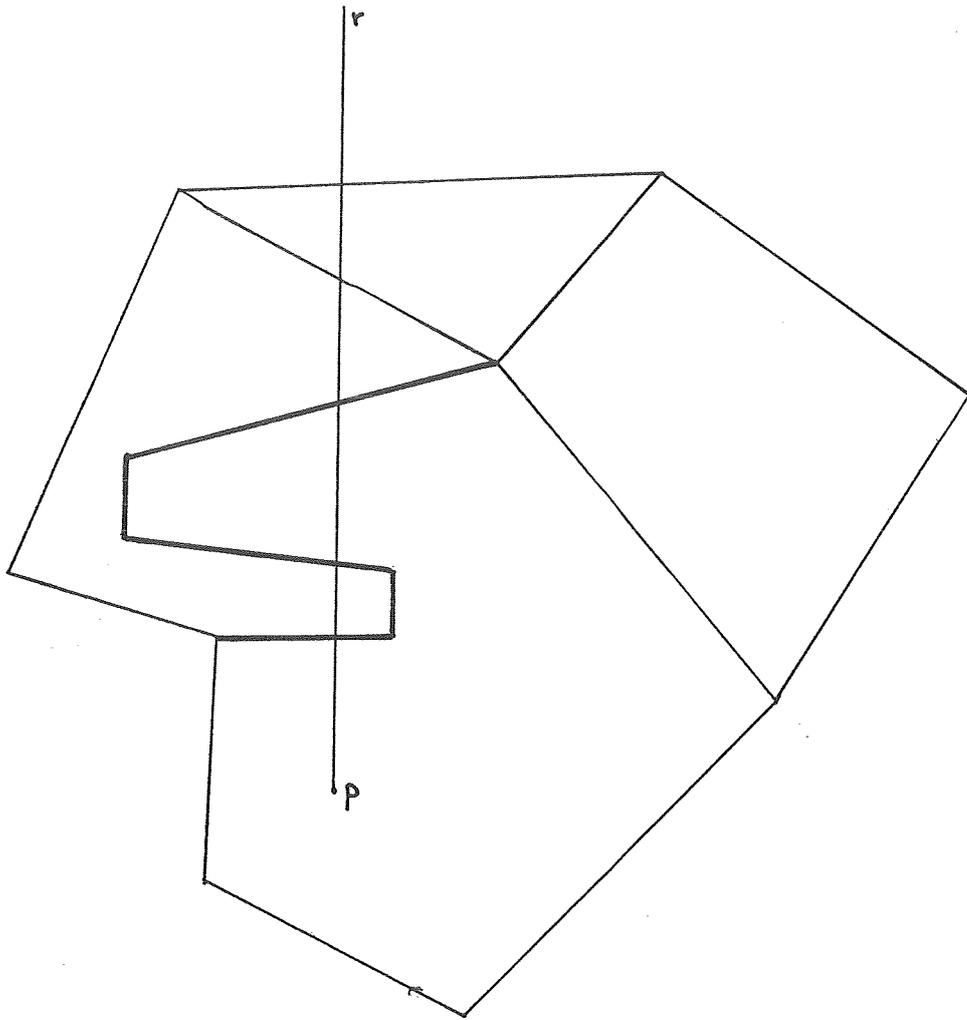


fig.7.3

E' evidente come tale tratto, TC, appartenga sicuramente al confine dell'area individuata dal punto P. Inoltre considereremo il confine di un'area orientato in modo da lasciare l'area sulla destra, cosicche' sia possibile associare un'orientazione positiva (segno "+") o negativa (segno "-") ad ogni tratto appartenente al confine; vediamo quindi di stabilire l'orientazione del tratto TC.

Consideriamo il segmento del tratto TC che ha il punto di intersezione con la semiretta r a minor distanza da P ; se il segmento e' digitalizzato in modo tale che, percorrendolo nel verso di digitalizzazione, il punto P rimane sulla destra si attribuisce al segmento il segno + (orientazione positiva) altrimenti si attribuisce al segmento segno - (orientazione negativa). Tale segmento determina l'orientazione del tratto TC : positiva se l'orientazione del segmento e' positiva, negativa se l'orientazione del segmento e' negativa.

Conoscendo il tratto TC possiamo determinare una poligonale di confine usando il seguente algoritmo :

Algoritmo 1

- 1- Si memorizza il tratto TC con il suo segno.
- 2- Se il punto iniziale, PI , di TC coincide col punto finale, PN , FINE (si ha un isola).
- 3- Se il tratto TC ha orientazione positiva si pone $PA=PN$ e $PP=PI$ altrimenti $PA=PI$ e $PP=PN$.
- 4- Fra tutti i tratti che hanno un estremo coincidente col

punto PA, sia T il tratto che forma con TC l'angolo minore misurato in senso antiorario.

- a) Se e' il punto iniziale di T a coincidere con PA si pone PS= punto finale di T, l'orientazione di T e' positiva quindi si attribuisce a T il segno "+".
- b) Se e' il punto finale di T a coincidere con PA si pone PS = punto iniziale di T, l'orientazione di T e' negativa quindi si attribuisce a T il segno "-".

5- Si memorizza T col suo segno.

6- Se il punto PS e' uguale a PP allora FINE altrimenti si pone PA=PS, TC=T e si va a 4.

Se l'area individuata dal punto P e' semplicemente connessa, cioe' non contiene isole al suo interno, allora la poligonale che si determina con tale algoritmo (fig.7.4) rappresenta il confine dell'area; se l'area non e' semplicemente connessa, ma le isole che essa contiene non intersecano la semiretta r, allora la poligonale e' il perimetro del poligono Q; altrimenti e' il perimetro di un poligono interno all' area (fig.7.5).

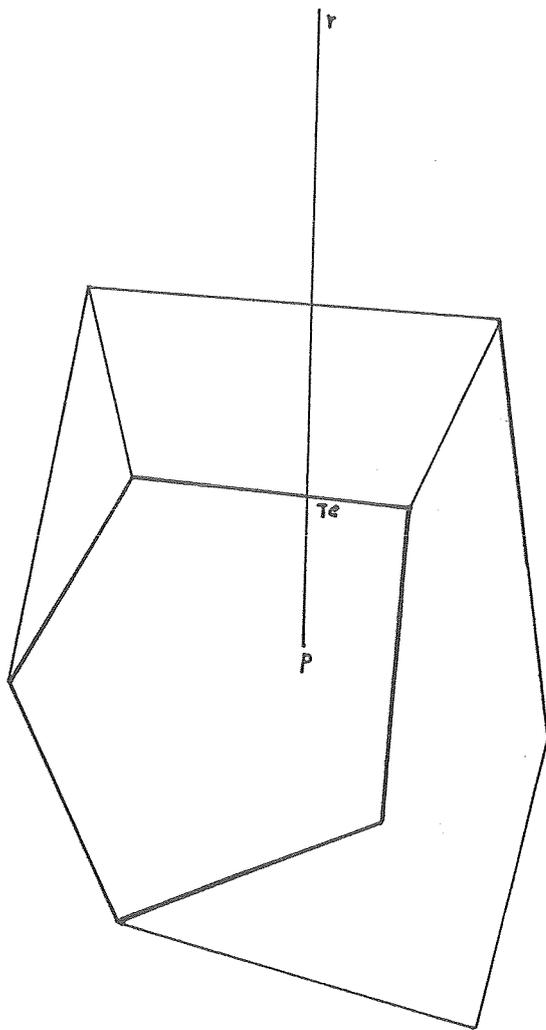


fig.7.4

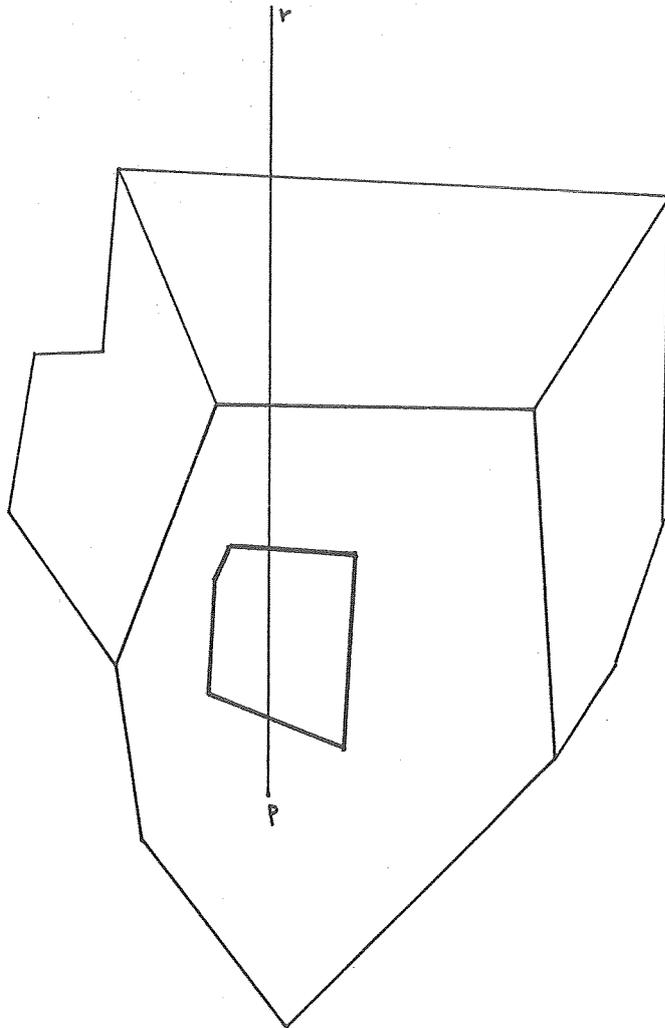


fig.7.5

Quindi l'algoritmo precedente non risolve completamente il problema P1; non solo, con tale algoritmo, non sappiamo se la poligonale trovata e' il confine del poligono Q o di un'isola interna. Esistono degli algoritmi capaci di determinare se un punto e' esterno od interno ad un poligono (cfr. [3] pagg. 22-23). Usando uno di tali algoritmi potremo risolvere completamente il problema P1 con il seguente metodo:

- 1- Si crea un file tratti ausiliario nel quale si copia il file tratti compensati.
- 2- Si determina, col metodo esposto precedentemente, un tratto, TC, che sicuramente appartiene al confine.
- 3- Si esegue l'algoritmo 1 dandogli in ingresso il file tratti ausiliario ed il tratto TC; si determina cosi' una poligonale.
- 4- Se il punto P e' interno al poligono individuato da tale poligonale l'algoritmo e' concluso; se il punto P e' esterno si tolgono dal file tratti ausiliario i tratti della poligonale e si va a 2.

Se le isole sono relativamente poche, come nei casi che abbiamo preso in considerazione, il metodo precedente e' costoso: anche se un punto isolato determina un'area semplicemente connessa si deve verificare se il punto e' interno a tale area.

Abbiamo preferito abbandonare tale metodo e risolvere il problema P1 in modo interattivo al momento

dell'assegnazione del codice destro/sinistro (cfr. paragrafo 7.4).

7.3 - Problema P2

Esporremo alcune possibili soluzioni del problema P2 supponendo di conoscere per ogni punto isolato P il poligono Q del problema P1.

Soluzione 1

- 1- Si costruisce un file contenente tutti i tratti contenuti nel poligono Q; sia tale file "fileQ".
- 2- Se il "fileQ" e' vuoto l' area e' semplicemente connessa.
- 3- Si considera il tratto piu' vicino al punto P, TC, si applica l'algoritmo 1 (cfr. pag. 50) dandogli in ingresso il tratto TC ed il "fileQ"; si determina cosi' una poligonale di confine.
- 4- Si tolgono dal "fileQ" i tratti della poligonale
- 5- Se il "fileQ" non e' vuoto si torna a 3, altrimenti fine.

Soluzione 2

Con questo metodo si ribalta il problema P2, nel senso che non si determinano le eventuali aree interne al poligono Q,

ma si cerca di vedere se il poligono Q e' un'isola di un altro poligono; per fare questo pero' abbiamo bisogno di attribuire un codice (provvisorio) destro/sinistro ad ogni tratto. Il codice e' provvisorio nel senso che, si attribuisce tale codice in base allo stato attuale della chiusura: per ogni punto isolato P noi conosciamo il poligono Q del problema P1.

1- Attribuzione provvisoria del codice destro sinistro ad ad ogni tratto:

per ogni tratto si va a vedere il poligono al quale il tratto appartiene; se il tratto appartiene al poligono A allora: se il segno del tratto nel poligono e' "+" si pone codice destro del tratto = A, se il segno e' "-" si pone codice sinistro del tratto = A.

(Se un tratto non ha il codice destro o il codice sinistro (supponiamo che il codice destro/sinistro sia 0) significa che il tratto appartiene ad un solo poligono noto e non conosciamo con chi confina il poligono lungo tale tratto, cioe' l'altro poligono al quale il tratto appartiene).

2- Si memorizzano i tratti con un codice zero in un file, detto "filec", insieme al codice diverso da zero.

3- Se il "filec" e' vuoto o EOF si termina l'esecuzione, altrimenti leggi dal "filec" un tratto, T, e il codice.

- 4- Il codice del tratto individua uno, dei due poligoni, al quale il tratto appartiene, sia P il punto che determina tale poligono.
- 5- Si crea un file tratti ausiliario, detto "fileta", nel quale si copia il file tratti compensati.
- 6- Si considera una semiretta con origine il punto P e che interseca tale tratto. Togliamo dal "fileta" il tratto T e i tratti che appartengono ai confini noti di P. Fra tutti i tratti, appartenenti al "fileta", che intersecano tale semiretta consideriamo quello, TP, che ha un punto d'intersezione a minor distanza da P.
- 7- Tale tratto appartiene ad uno o due poligoni; siano B e C. Se il punto e' interno a uno di tali poligoni, supponiamo il poligono B, allora il poligono determinato da P e' un'isola di B, quindi si puo' ampliare i confini di B inserendo i tratti del poligono determinato da P che hanno un codice 0 e togliere tali tratti dal "filec".
- 8- si va a 3.

La soluzione 1 e' costosa perche' per determinare i tratti contenuti in un poligono bisogna fare moltissimi controlli.

Per quanto riguarda la soluzione 2 osserviamo che, tutto cio' che e' esterno alle aree individuate da punti isolati, e' il "resto del mondo"; cioe' quando si definisce

il codice destro/sinistro di un tratto, che confina con il "resto del mondo", il suo codice destro o il suo codice sinistro sarà zero; i tratti che confinano con il "resto del mondo" sono numerosi, appartengono al "filec", e quindi per ognuno di questi tratti con la soluzione 2 si cerca di determinare il poligono B, poligono che non esiste. Si potrebbe pensare di ovviare a questo inconveniente attribuendo un codice diverso da zero a tali tratti ma, per far questo, bisogna introdurre un ulteriore punto isolato, punto che individua il "resto del mondo" e quindi chiudere anche tale punto; ma chiudere il "resto del mondo" è costoso per la grande quantità di tratti e di isole che lo costituiscono.

Le soluzioni esposte per i problemi P1 e P2, oltre ad essere costose, non tengono conto degli errori che l'operatore che digitalizza può fare:

- E1- Un punto isolato non è stato digitalizzato.
- E2- Un tratto non è stato digitalizzato.
- E3- In un poligono sono stati digitalizzati due o più punti isolati anziché uno.
- E4- Un tratto è stato digitalizzato due volte.

Quindi nelle varie soluzioni bisogna introdurre alcuni controlli che permettono di rilevare tali errori e segnalarli.

7.4 - La nostra soluzione.

Come nel caso della compensazione, abbiamo ritenuto opportuno usare un video grafico interattivo per risolvere completamente il problema della chiusura.

Prima di esporre il nostro metodo diamo la struttura del file tratti compensati e del file punti isolati. La struttura dei records del file tratti compensati e' riportata nella tabella 2; mentre quella del file punti isolati e' riportata nella tabella 3.

Possiamo attribuire ad ogni punto isolato un numero d'ordine: il punto i -esimo del file punti avra' numero d'ordine " i ".

Usando l'algoritmo 1 (cfr. pag. 50) si determina, per ogni punto isolato P , una poligonale chiusa che rappresenta una parte del confine (o tutto il confine) dell'area individuata dal punto (non sappiamo, come gia' detto, se la poligonale e' il confine del poligono Q del problema P_1 o di un'isola interna a tale poligono); si memorizza in un file poligoni il punto P (piu' precisamente il suo numero d'ordine) e il numero d'ordine dei tratti che compongono tale poligonale con il loro segno.

La struttura records del file poligoni e' riportata nella tabella 4. Il valore "-1", in tale tabella indica la fine dell'elenco dei tratti che formano il poligono.

Puo' capitare che un tratto, appartenente al file

tratti compensati, non appartenga a nessun poligono, o perche' e' stato commesso un errore di digitalizzazione, E1 o E4, (fig.7.6), o perche' il tratto appartiene ad un poligono, individuato da un punto P, che contiene al suo interno un'isola e la poligonale che determiniamo con l'algoritmo 1 e' il confine dell'isola (fig.7.7); quindi si controlla, facendo un confronto fra i tratti presenti nel file tratti compensati e quelli nel file poligoni, se esistono tratti che non appartengono a nessun poligono. Appena troviamo uno di tali tratti avvertiamo l'operatore alla console della sua esistenza, visualizziamo la situazione sul video grafico interattivo, rappresentando il tratto in questione di un colore diverso da quello usato per gli altri tratti. Inoltre su ogni tratto, disegnato sul video, abbiamo fatto scrivere il proprio numero d'ordine e una freccetta che indica il verso di digitalizzazione del tratto; i punti isolati sono rappresentati, sul video, con il loro numero d'ordine. A questo punto, un menu permette all'operatore di prendere le opportune decisioni: introdurre da terminale un nuovo punto, eliminare il tratto, assegnare il poligono di appartenenza per il tratto in questione.

Dopo aver fatto questa chiusura parziale si attribuisce il codice destro/sinistro ad ogni tratto usando il seguente algoritmo:

Algoritmo 2

Con riferimento alla struttura record descritta nella tabella 4.

- 1- Si pone il codice destro/sinistro di ogni tratto = 0.
- 2- Si apre il file poligoni.
- 3- Si legge dal file il punto, P.
- 4- se eof FINE.
- 5- Si legge dal file il numero d'ordine, NORD, di un tratto ed il segno; se NORD e' uguale a "-1" si va a 3.
- 6- Se il segno e' "+" e il codice destro del tratto e' zero si pone il codice destro del tratto, avente numero d'ordine NORD, uguale a P;
se il segno del tratto e' "+" ed il codice destro non e' zero si manda un messaggio di errore: errore di digitalizzazione (E2 o E3).
- 7- Se il segno e' "-" e il codice sinistro del tratto e' zero si pone il codice sinistro del tratto, avente numero d'ordine NORD, uguale a P;
se il segno del tratto e' "-" ed il codice sinistro non e' zero si manda un messaggio di errore: errore di digitalizzazione (E2 o E3).
- 8- Si va a 3.

Dopo aver applicato tale algoritmo puo' capitare che il codice destro/sinistro di un tratto sia zero perche', o il tratto confina con il "resto del mondo", o e' un'isola

interna od e' un tratto digitalizzato due volte. Se un tratto ha un codice zero lo comunichiamo all'operatore alla console e visualizziamo la situazione sul video grafico interattivo; ancora una volta un menu' permette all'operatore di prendere delle decisioni:

a- togliere il tratto

b- dire a quale poligono appartiene il tratto.

Per definire a quale poligono appartiene il tratto l'operatore deve dare il numero d'ordine del punto, che individua il poligono, e dire se tale punto si trova sulla destra o sulla sinistra del tratto, rispetto al verso di digitalizzazione; cio' non e' difficile per l'operatore in quanto sul video grafico sono riportati il numero d'ordine dei punti isolati e, su ogni tratto, una freccia che indica il verso di digitalizzazione. Se l'operatore prende la decisione b) allora, a partire da tale tratto, usando l'algoritmo 1, si determina il confine del poligono e si attribuisce il codice ad ogni tratto di tale poligono.

All'operatore conviene attribuire un codice destro/sinistro particolare, per esempio "-1", ai tratti che confinano con il "resto del mondo", infatti, in questo caso, iniziando da uno di questi tratti, si fa la chiusura del "resto del mondo" e cosi' tratti che compongono i confini del "resto del mondo" hanno codice diverso da zero.

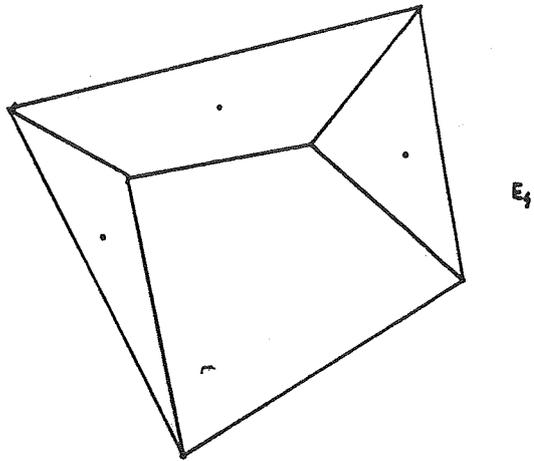
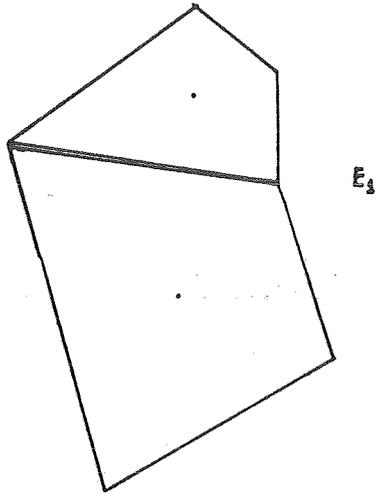


Fig.7.6

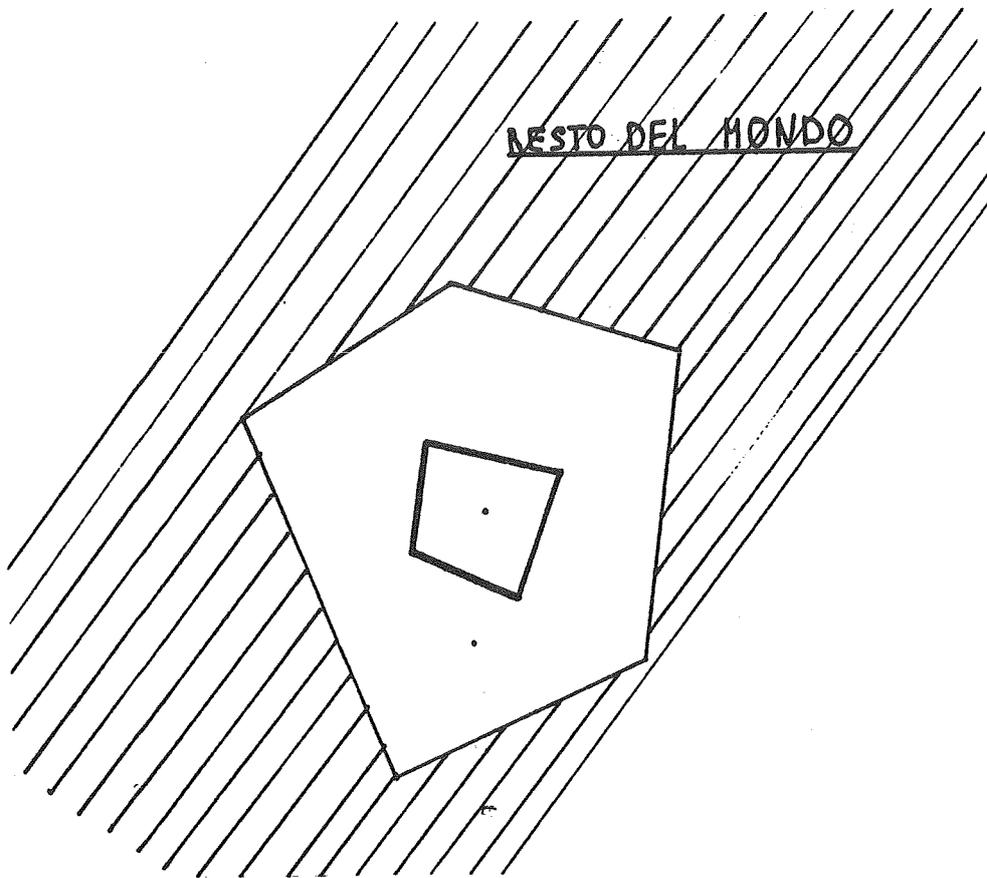


fig.7.7

8. APPENDICE

8.1 - Le prove della fase di compensazione.

Come accennato nel paragrafo 6.4, abbiamo provato i nostri programmi sui confini comunali della Puglia, digitalizzati dall'IGM da una carta 1:100.000. Il file originale e' costituito da 1339 tratti, per un totale di 59.779 records, la cui struttura e' riportata nella tabella 2.

Abbiamo fatto eseguire la procedura di sfoltimento con un ϵ di 10.000 (pari ad un millimetro sulla carta). Si sono scoperti 9 tratti "pendenti" ed il file di output si e' ridotto a 11.878 records.

La procedura di compensazione, eseguita con un ϵ di 8.000, ha riconosciuto 38 punti "pendenti", di cui solo due hanno creato un'area "nascosta", mentre per altri tre ci e' risultato impossibile tentare un collegamento, a causa della loro posizione irrazionale rispetto alla situazione globale mostrata sul video.

L'intera prova, eseguita sul sistema IBM 3081 del CNUCE di Pisa, ha richiesto circa 400 secondi di CPU, mostrando una certa snellezza nell'esecuzione e non impegnando pesantemente l'operatore alla console.

Il tempo totale della seduta a terminale per l'intera operazione e' stato di circa un'ora, in un periodo di carico normale di macchina.

8.2 - Le prove della fase di chiusura.

Non abbiamo potuto provare i programmi di chiusura sui confini comunali della Puglia perche' non avevamo a disposizione i punti centrali. Abbiamo provato detti programmi su dei dati messi a disposizione dalla Regione Toscana.

Il file tratti compensati, ottenuto a partire da questi dati, e' costituito da 729 tratti per un totale di 4426 records, la cui struttura e' quella riportata nella tabella 2; il file punti isolati e' costituito da 77 punti.

Questo tipo di dati ha costituito una verifica particolarmente severa per la procedura; erano infatti presenti numerosissime 'isole', (oltre l'ottanta per cento delle aree presenti) per cui il test della procedura e' avvenuto nelle condizioni piu' sfavorevoli, con numerosissimi ricorsi ad interventi di tipo interattivo da

parte dell'operatore.

Il tempo globale impiegato dall'operatore e' stato di meno di un'ora, nelle stesse condizioni operative del paragrafo precedente.

8.3 - Tabelle.

N. TRATTO	Cod. Destro	Cod. Sinistro
1	A	O
2	C	A
3	C	O
4	C	D
5	O	D
6	D	B
7	A	B
8	O	B

tabella 1

NUMERO del RECORD	CAMPO 1	CAMPO 2
i	N. d'ord. del tr.	N. di P.ti del tr.
i+1	ascissa di P1	ordinata di P1
.	.	.
.	.	.
.	.	.
i+N	ascissa di PN	ordinata di PN

tabella 2

NUMERO del RECORD	CAMPO 1	CAMPO 2
i	ascissa del punto	ordinata del punto

tabella 3

N. del REC.	CAMPO 1	CAMPO 2
1	n. d'ord.del 1° pol.	non gestito
2	n. ord.tratto 1 di pol. 1	segno
3	n.ord.tratto 2 di pol. 1	segno
.	.	.
.	.	.
.	.	.
N+1	n.ord.tratto N di pol.1	segno
N+2	"-1"	non gestito
N+3	n. d'ord.del 2° pol.	non gestito
N+4	n.ord.tratto 1 pol.2	segno
.	.	.
.	.	.
.	.	.
N+K+3	n.ord.tratto K pol.2	segno
N+K+4	"-1"	non gestito
.	.	.
.	.	.
.	.	.
i	n. d'ord. del j° pol.	non gestito
i+1	n.ord.tratto 1 pol.j	segno
.	.	.
.	.	.
i+M	n.ord tratto M pol.j	segno
i+M+1	"-1"	non gestito

tabella 4

BIBLIOGRAFIA

[1] REGIONE TOSCANA - Giunta regionale - SEDD: Sistema di gestione di dati territoriali. Rapporti/3.

[2] C. Ragazzi - P. Gilardi: L'uso di SC2: un sistema per la cartografia mediante elaboratore, applicato alla progettazione di sistemi di parco regionale.

M. Bernardi - V. Franchina - C. M. Serio:
Costruzione di un modello tridimensionale a partire da dati cartografici come ausilio alla progettazione urbanistica.

Aicographics 82. Milano - 26/29 ottobre 1982

[3] GEOSYSTEMS - Sistemi informativi territoriali: Studio e sperimentazione sulla struttura del dato grafico.

[4] REGIONE TOSCANA - SEDD: Inventario forestale della Toscana. Lavori di acquisizione digitale di dati relativi ai bacini idrografici e al vincolo idrogeologico della Toscana. Firenze 1983.

INDICE DELLE FIGURE

fig.2.1	pag. 8
fig.3.1	pag. 13
fig.3.2	pag. 15
fig.3.3	pag. 16
fig.3.4	pag. 16
fig.3.5	pag. 21
fig.4.1	pag. 21
fig.4.2	pag. 26
fig.4.3	pag. 27
fig.5.1	pag. 33
fig.5.2	pag. 33
fig.6.1	pag. 37
fig.6.2	pag. 39
fig.6.3	pag. 40
fig.6.4	pag. 42
fig.7.1	pag. 47
fig.7.2	pag. 48
fig.7.3	pag. 49
fig.7.4	pag. 52
fig.7.5	pag. 53
fig.7.6	pag. 63
fig.7.7	pag. 64

