

# Gestione dell'informazione Geospaziale

Caccaro Sebastiano  
A.A.2019/2020

# Contents

<b>1</b>	<b>Introduzione</b>	<b>3</b>
1.1	Aspetti organizzativi . . . . .	3
1.1.1	Obiettivi . . . . .	3
1.1.2	Organizzazione del corso . . . . .	3
1.1.3	Materiale . . . . .	3
1.2	Concetti Base . . . . .	4
1.2.1	Esempio di OpenStreetMaps . . . . .	4
1.2.2	Spazio . . . . .	4
1.2.3	Acquisizione dei dati . . . . .	5
1.2.4	Trattamento dei dati . . . . .	5
<b>2</b>	<b>Coordinate Geospaziali</b>	<b>5</b>
2.1	Coordinate Geografiche . . . . .	6
2.2	Latitudine e longitudine . . . . .	7
2.3	Distanza . . . . .	7
2.4	Forma della Terra . . . . .	8
2.4.1	Trasformazione di datum . . . . .	9
2.5	Coordinate Proiettate . . . . .	9
2.5.1	LatLong projection . . . . .	10
2.5.2	Proiezione di Gauss . . . . .	10
2.6	Sistema di coordinate Geografiche (CRS) . . . . .	11
2.6.1	Trasformazione dei sistemi di coordinate . . . . .	12
2.7	Scala . . . . .	12
<b>3</b>	<b>Modelli dei dati e gestione dei dati spaziali</b>	<b>12</b>
3.1	Modelli per la rappresentazione dei dati spaziali . . . . .	13
3.1.1	Modello Vettoriale . . . . .	13
3.1.2	Modelli basato su tassellazione . . . . .	13
3.2	Raster . . . . .	13
3.2.1	Raster VS Vettoriale . . . . .	14
3.3	Modello TIN (Triangular Irregular Network) . . . . .	14
<b>4</b>	<b>Basi di dati spaziali</b>	<b>14</b>
4.1	Architetture dei dati . . . . .	15
4.1.1	Gestione dei dati tramite file system . . . . .	15
4.1.2	Modello relazionale esteso . . . . .	16
4.1.3	Modello relazionale ad oggetti . . . . .	16
4.1.4	Modello OGC Simple Features . . . . .	16
4.2	Modelli Mesh . . . . .	16

4.3	OCG Simple Features . . . . .	17
4.3.1	Point . . . . .	18
4.3.2	Curve . . . . .	18
4.3.3	Polygon . . . . .	18
4.4	Geometry Collection . . . . .	19
4.5	Formati per la rappresentazione geometrica . . . . .	20
4.5.1	Esempi di WKT . . . . .	20
4.6	Validità per la rappresentazione Geometrica . . . . .	20
<b>5</b>	<b>Relazioni e Predicati Spaziali</b>	<b>21</b>
5.1	Relazioni Metriche . . . . .	21
5.2	Relazioni Topografiche . . . . .	22

# 1 Introduzione

**Geospaziale** tratta di dati sulla superficie terrestre. Posso trattare vari tipi di spazi, anche su unità di misura diverse, come lo spazio-tempo.

## 1.1 Aspetti organizzativi

### 1.1.1 Obiettivi

Concetti base:

- Acquisizione dei dati
- Gestione dei dati: li mantengo in dei DBMS spaziali.
- Analisi dei dati, clustering (= raggruppare oggetti in base a criteri di omogeneità) ecc.

### 1.1.2 Organizzazione del corso

Il corso verrà così articolato:

- Concetti base
- DBMS Spaziali
- Rappresentazione di oggetti in movimento
- Analisi dei dati spaziali.

#### **Bisogna sapere PostgreSQL!**

L'esame consiste in un progetto più di un orale (potrebbe diventare una prova scritta all'ultima lezione del corso).

Il sito del corso è `homes.di.unimi.mdamiani/corsi/gig/`

**User:** gis7 **Pwd:** sql07sql

### 1.1.3 Materiale

È presente un libro in formato PDF sul sito della docente.

## 1.2 Concetti Base

### 1.2.1 Esempio di OpenStreetMaps

OpenStreetMaps è una **mappa** aperta **collaborativa**, sostanzialmente un disegno. Ma non ci interessano i colori delle strade ecc. Cosa vuol dire Mappa Collaborativa?

- **Mappa** è una banca dati che contiene informazioni geografiche coerenti, che poi vengono rappresentate tramite una mappa.
- **Collaborativa** la base di dati è modificabile da chiunque.

Costruire mappe è sempre stato altamente dispendioso, soprattutto per quanto riguarda l'acquisizione dei dati. OpenStreetMaps rende facile il reperimento e l'uso di dati spaziali.

### 1.2.2 Spazio

Parliamo di spazio geometrico con longitudine e latitudine, che vogliono una posizione e un sistema di riferimento. È però molto più facile usare lo spazio cartesiano. Lo spazio simbolico invece rappresenta dei luoghi anche in base alla loro funzione (esempio cartine indoor). A seconda del tipo di spazio che sto analizzando, cambia anche la nozione di distanza. Ad esempio, come misuro la distanza in ambiente indoor?

Un oggetto può avere vari tipi di proprietà:

- **Geometriche**: forma
- **Topografiche**: i collegamenti, a mo' di grafo
- **Tematiche**: caratteristiche, ad esempio il numero di abitanti di un edificio

Ci possono essere movimenti di tipo:

- **Continuo**: ad esempio, movimento di palla nello spazio.
- **Discreto**: ho un numero di posizioni finito, ad esempio posso essere in un dato momento solo sotto una cella telefonica.

È importantissimo poter visualizzare i dati. Altrimenti, non riesco a farmi un'idea di cosa ho in mano.

### 1.2.3 Acquisizione dei dati

Ci sono vari strumenti e metodi (arei, GPS, ecc.).

La posizione non è mai precisa al 100%. I dati geospaziali dovrebbero essere sempre accompagnati anche dalla misura della loro incertezza.

### 1.2.4 Trattamento dei dati

Sono presenti varie tecnologie che permettono di gestire e trattare i dati raccolti:

- **Sistemi GIS:** piattaforme software (= programmoni) che mi permettono di:
  - Acquisire dati: digitalizzarli, controllarne la correttezza, integrare dati eterogenei (con fonti e caratteristiche diverse).
  - Archiviare e accedere ai dati
  - Trattare i dati
  - Visualizzare dati
- **DBMS spaziali:** DBMS normali arricchiti con tipi e operazioni per dati spaziali. Praticamente sono SQL con estensioni spaziali.
- Package specializzati.

11 Ottobre 2019

## 2 Coordinate Geospaziali

### Geodesia

Disciplina che studia la forma della Terra, le sue dimensioni, i metodi per determinare la posizione dei punti sulla sua superficie.

### Cartografia

Disciplina che ha per oggetto la rappresentazione grafica, in proporzioni ridotte, della superficie terrestre o di una parte di essa, dei suoi aspetti caratteristici, fenomeni che si svolgono e la preparazione e la costruzione delle carte geografiche.

Le coordinate sono delle tuple di numerici che hanno senso in un sistema di riferimento (CRS - Coordinate Reference System). Bisogna distinguere due tipi di coordinate:

- **Coordinate Geografiche:** ex Latitudine e longitudine.
- **Coordinate Proiettate:** ovvero proiettate su un piano.

È possibile anche fornire delle coordinate indirette, come un indirizzo postale, IP, ecc. Questi metodi vengono poi mappati a coordinate.

## 2.1 Coordinate Geografiche

### Meridiano

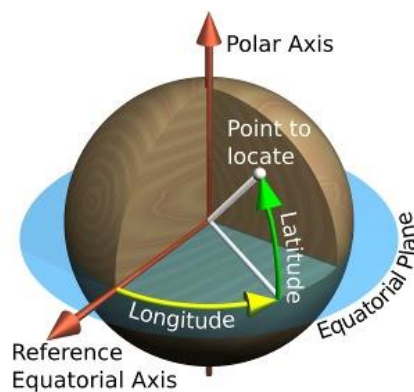
Semicirconferenza che unisce i due poli.

### Meridiano di Greenwich

Anche detto Meridiano Fondamentale”. Meridiano di riferimento per il calcolo della circonferenza.

### Piano Equatoriale

Piano che ha come proiezione l’equatore.



## 2.2 Latitudine e longitudine

### Longitudine

Angolo sul piano di meridiano fra l'equatore e la linea passante per P e perpendicolare alla superficie terrestre. Indicata con  $\phi$ .

### Latitudine

Angolo sul piano equatoriale formato dal piano del meridiano passante per P e il piano del meridiano fondamentale. Indicata con  $\lambda$

Sia latitudine e longitudine sono angoli, quindi si misurano in gradi. Ma ci sono 2 modi per esprimerli:

- **Sistemi sessasegimali:** Misuro in sessantesimi.
- **Sistemi digitali:** Misure in frazioni decimali di grado.

La longitudine è espressa fra **180 OVEST** e **180 EST** (-180 e 180). La latitudine è espresso fra **90 NORD** E **90 SUD** (0 -90 e 90). Per ovvi motivi, nei poli la longitudine non è definita.

## 2.3 Distanza

Chiaramente, per calcolare la distanza, non basta tirare una linea. La distanza più breve fra due punti, non è una linea retta sul piano.

### Cerchio Massimo

Circonferenza che risulta dall'intersezione di un piano passante per il centro e la superficie terrestre. Il cerchio massimo è un esempio di geodetica



### Distanza di Cerchio Massimo

Distanza più breve tra 2 punti A e B è data dall'arco di cerchio massimo passante per A e B (con estremi A e B). Ipotesi di Terra sferica (errore stimato 0.3%)



Figure 1: Distanza più breve (di cerchio massimo) fra due punti sulla terra

## 2.4 Forma della Terra

La terra non è una sfera, ci sono valli, montagne, ecc. Dobbiamo quindi stabilire una superficie di riferimento, un modello geometrico, **datum geodetico**.

Una prima approssimazione potrebbe essere il geoide, che "taglia" le montagne e pone il livello di riferimento alla superficie del mare. Sarebbe però troppo difficile da computare, quindi usiamo un modello regolare detto **Ellissoide**. Sono stati definiti vari tipi di ellissoide. Ciò è causato da vari fattori. Alcuni ellissoidi approssimano meglio alcune aree geografiche (es Europa, Italia ecc) (datum locale), mentre altri cercano di approssimare tutta la superficie terrestre senza fare troppe preferenze. La tendenza è quella di migrare verso dei modelli globali (datum globale), come WGS84, usato nel GPS.

Va da sè che latitudine e longitudine non sono sufficienti per identificare univocamente un punto, ma è necessario anche il codice dell'ellissoide.

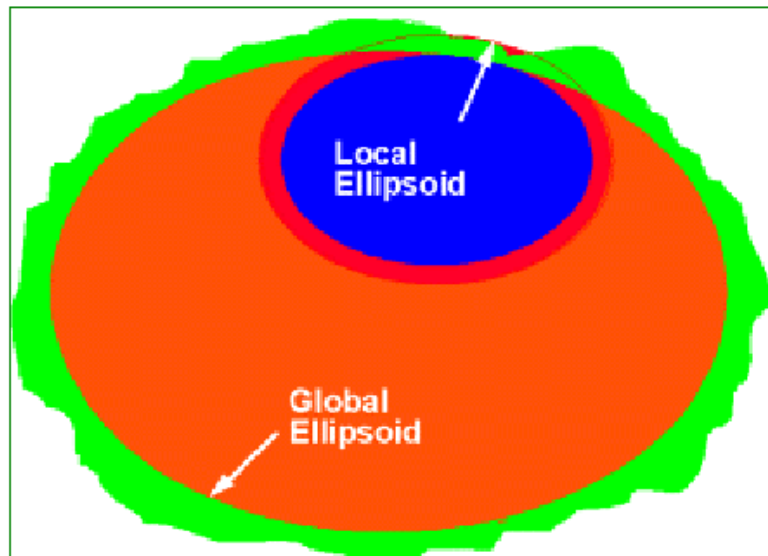


Figure 2: Ellissoidi locali e globali

#### 2.4.1 Trasformazione di datum

Può dover essere necessario eseguire delle trasformazioni di datum fra modelli geodetici differenti.

### 2.5 Coordinate Proiettate

Consistono sostanzialmente in un mapping da coordinate geografiche a coordinate cartesiane. È possibile passare da una proiezione all'altra senza perdita di dettagli tramite alcune formulette matematiche.

Ovviamente, questa mappatura causa delle deformazioni, e in base a queste posso classificare le proiezioni:

- **Equivalenti:** mantengono le aree
- **Conformi:** mantengono gli angoli
- **Equidistanti:** mantengono le distanze

Non esistono proiezioni migliori, ma solo più ideali rispetto al contesto. Generalmente ci viene **più facile** usare coordinate proiettate.

14 Ottobre 2019

### 2.5.1 LatLong projection

Latitudine e longitudine vengono usati come X e Y. Si utilizza perchè è molto semplice, senza troppi fronzoli. Si può usare quando non si hanno troppe esigenze di precisione.

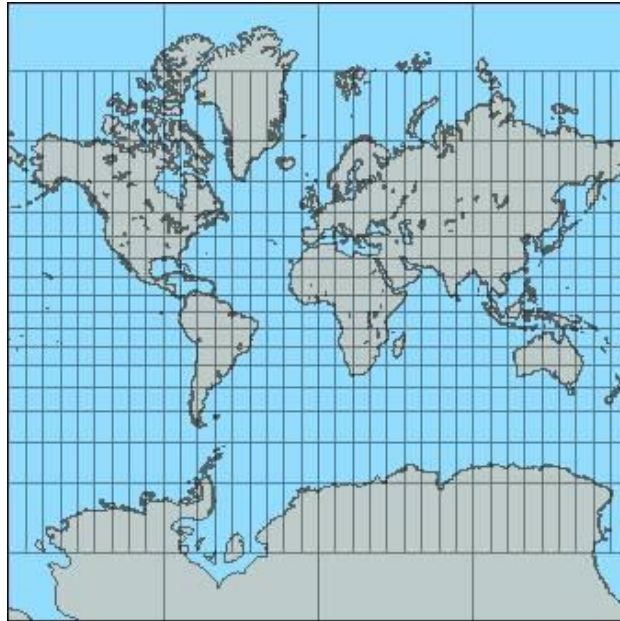


Figure 3: Proiezione LatLong

### 2.5.2 Proiezione di Gauss

Sta alla base di molte proiezioni, chiamata anche **Transverse Mercator**. È una **proiezione conforme**, quindi preserva gli angoli. Introduce grosse deformazioni, per cui usa per aree non più grandi di 6 gradi di longitudine dall'equatore (circa 600km).

La proiezione di Gauss è alla base di alcune proiezioni, fra cui:

- **UTM:** (Universal Transverse Mercator). La terra è divisa in zone delimitate da 60 meridiani e da spicchi di 6 gradi di longitudine. le coordinate sono espresse in metri (x dal meridiano centrale, y dall'equatore). Per non avere coordinate negative, lo zero corrisponde a 500km.
- **Proiezioni di Gauss-Boaga:** usata in Italia, e pensata da un italiano per il territorio italiano. Si divide la longitudine in due fusi (verticali). Anche in questo caso sono presenti delle false origini per avere sempre coordinate positive.

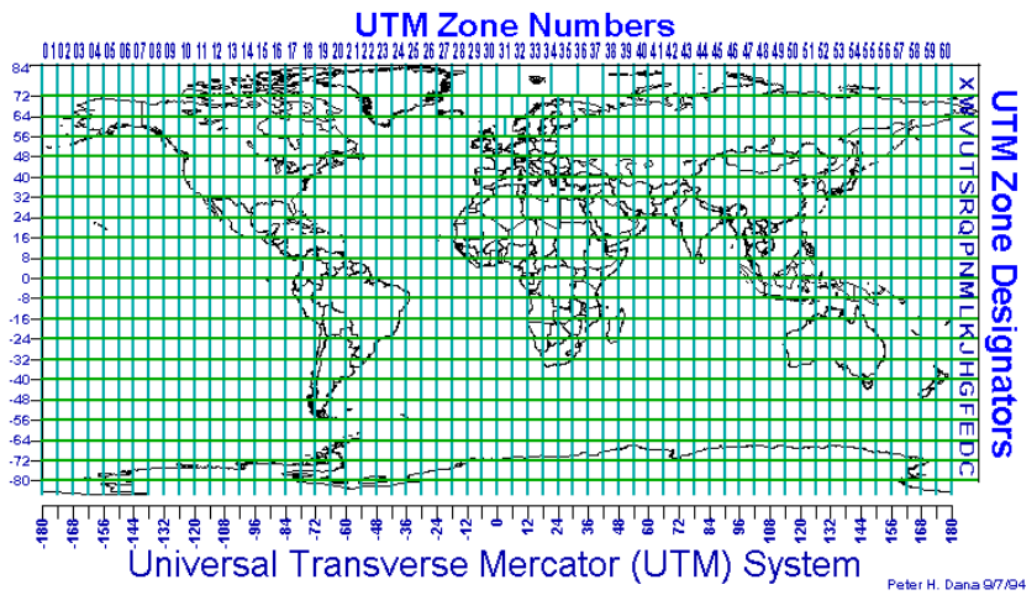


Figure 4: Divisione in zone UTM

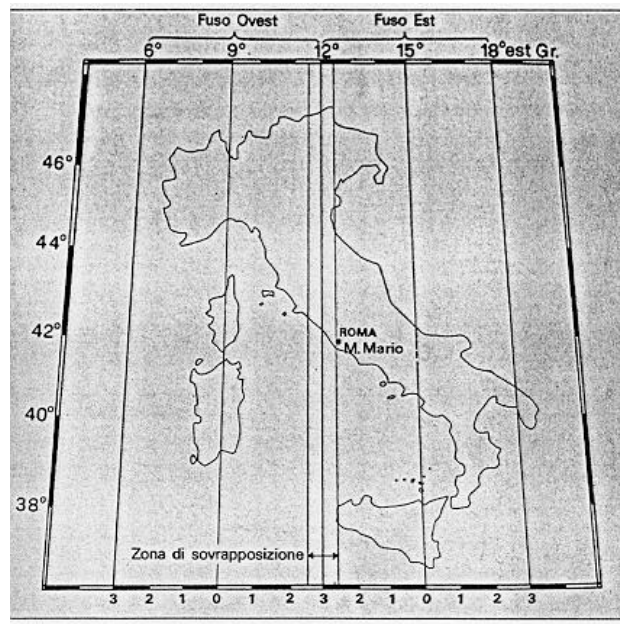


Figure 5: Proiezione di Gauss Boaga

## 2.6 Sistema di coordinate Geografiche (CRS)

Per stabilire un sistema di coordinate geografiche, ho bisogno di:

- Una proiezione cartografica
- Una Datum geodetico

Le varie combinazioni di CRS sono delle stringhe definite da un gruppo chiamato **EPSG**, che identificano le varie coppie, come **EPSG 4326** ecc.

### 2.6.1 Trasformazione dei sistemi di coordinate

La trasformazione fra coordinate è lossless, perchè posso ricondurre ogni coordinata alle coordinate geografiche e viceversa. È già più complicato passare fra due dati geoedici diversi.

## 2.7 Scala

Il concetto di scala è abbastanza banale. Nella cartografia tradizionale, il concetto di scala indica implicitamente anche la precisione di un dato, perchè non posso rappresentare oggetti più piccoli di un tot. Questo concetto è trasportato anche nella cartografia digitale, dove la scala può stare a indicare la cartografia di un dataset.

Più una scale è grande, più il rapporto è maggiore, e viceversa.

18 Ottobre

## 3 Modelli dei dati e gestione dei dati spaziali

Ho bisogno di formulare delle astrazione per poter visualizzare i dati in  $\mathbb{R}^2$ . Posso farlo in due modi:

- **Visione concettuale:** Realizzo delle astrazioni per visualizzare delle entità
- **Visione basata su entità:** ho uno spazio in cui riesco a definire degli oggetti con forme geometriche e.g. curve di livello in un campo scalare o un campo vettoriale.

In  $\mathbb{R}^2$  posso rappresentare:

- **Entità spaziali:** oggetti tra loro distinguibili e univocamente identificabili aventi una forma geometrica (e.g. vetta della montagna)
- **Campi:** funzione  $f$  che associa ad ogni punto di una regione  $W$  in  $\mathbb{R}^2$  un valore, che può essere:

- **Numerico:** campo scalare e.g. temperatura  $f: W \rightarrow \mathbb{R}$
- **Vettoriale:** campo vettoriale e.g. velocità  $f: W \rightarrow \mathbb{R}^n$

Ad esempio, potrei rappresentare una foresta come un campo (vettoriale o numerico) o come un poligono, cioè un'entità.

## 3.1 Modelli per la rappresentazione dei dati spaziali

Esistono vari modelli per rappresentare dati spaziali.

### 3.1.1 Modello Vettoriale

Utilizzato per la rappresentazione di entità spaziali. Rappresentata da un elemento geometrico (punto, linea, poligono). I tipi geometrici fondamentali sono in numero limitato, ad esempio le spline non sono importanti e non sempre definiti. Non mi serve avere poligoni molto complessi, dato che i sistemi GIS hanno una maggiore tolleranza (es un cerchio posso semplificarlo con un poligono a  $n$  lati).

### 3.1.2 Modelli basato su tassellazione

Utilizzato per la rappresentazione di campi. Il campo devo discretizzarlo, la funzione campo  $f$  è definita su uno spazio  $W$  discretizzato. La regione  $W$  è partizionata in forma regolare o non regolare e ad ogni regione associo un valore (è praticamente una mesh). Il modello più comune è quello raster.

## 3.2 Raster

La regione dello spazio è divisa con una matrice (regolare) e il sistema di coordinate definisce la posizione delle celle nello spazio. Ogni cella ha associato il valore (valore del campo in quella posizione). Può essere osservato subito o tramite interpolazione e può essere null. Posso avere più layers di raster sovrapposti per codificare più informazioni.

Ad esempio, ogni cella rappresenta una porzione di territorio. Posso avere layer sovrapposti: uno per la vegetazione, uno per la strada, una per l'altitudine, edifici, pressione etc.

Una differenza sostanziale delle immagini raster, oltre che ad avere  $n$  layer, è che ci sono coordinate geografiche, non "locali". Corrispondono quindi a pezzi di superficie terrestre.

### 3.2.1 Raster VS Vettoriale

Un **raster** semplifica la descrizione di oggetti dai confini non definiti o incerti, è una struttura dati semplice e ha una rappresentazione approssimata.

Una mappa **vettoriale** invece ha caratteristiche complementari. Descrizione precisa della forma geometrica, agevola operazioni geometriche. È però più complesso rappresentare i campi.

Non esiste un modello migliore, tutto dipende dall'uso che bisogna fare.

### 3.3 Modello TIN (Triangular Irregular Network)

Diverso modello tassellato. Dato un insieme di punti campione (x, y, z), la regione è suddivisa in triangoli di dimensioni variabili e disposti irregolarmente nello spazio. Il triangolo approssima l'area compresa tra i 3 punti. La suddivisione è realizzata da un algoritmo di triangolazione. Non è proprio tridimensionale dato che  $z=f(x,y)$ . Significa che io non posso avere due valori di z per lo stesso punto. Solamente 2.5D (non posso avere ad esempio un "precipizio", ad esempio come se fosse un campionamento con la \*structured light\*). Usiamo le seguenti definizioni:

- **Nodi:** punti campione
- **Bordi:** lati dei triangoli: un bordo ha due nodi e un nodo può essere collegato a più bordi
- **Triangoli o face:** un triangolo ha 3 bordi
- **Topologia:** descrive la struttura dei triangoli e la relazione di adiacenza tra triangoli (adiacenti se hanno un bordo in comune)

Non è possibile effettuare interrogazioni su campi ed entità in un'unica interrogazione. Utilizzeremo solo dati vettoriali. Le ultime versioni di QGIS possono utilizzare dati di tipo mesh.

## 4 Basi di dati spaziali

La gestione dei dati spaziali è complessa, dato che il dato vettoriale ha una struttura complessa. Ad esempio, un poligono può essere descritto da migliaia di punti organizzati in array o array di array (strutture più complesse). I dati sono numerosi e non è ovvio come interrogare i dati spaziali. Una base di dati geospaziali deve quindi garantire l'accesso ai dati in essa contenuti in modo efficiente.

### **Base di dati spaziale**

Insieme della base di dati, modello dei dati e del linguaggio di interrogazione, che insieme permettono di utilizzare tipi di dati spaziali. Consente l'accesso efficiente a questi dati tramite indici spaziali.

## **4.1 Architetture dei dati**

Storicamente tutti i dati spaziali venivano gestiti tramite file system, poi si è cominciato ad integrarli con un DMBS relazionale con campi BLOB e poi grazie al Object-Relational possiamo estendere i tipi di dato del DBMS.

### **4.1.1 Gestione dei dati tramite file system**

Organizzati in files, prima architettura utilizzata. L'organizzazione dei dati è data dal formato **ESRI Shapefile** (o shape) che ora è (forse?) open source. I dati sono organizzati per layer (insieme omogeneo di entità). Ogni layer viene descritto da un insieme di files:

- **.dbf**: dati alfanumerici (non spaziali)
- **.shp**: dati spaziali
- **.shx**: indice per accedere ai dati spaziali
- **.prj**: dati relativi al CRS

Questi modelli, però, presentano alcune limitazioni:

- Prolifera di files, gestione complessa (e.g. per errore elimino uno o più file)
- I dati spaziali e alfanumerici non sono integrati ma separati, su file diversi. Modifiche separate possono rendere i dati inconsistenti
- Non esiste un linguaggio di interrogazione standardizzato e ottimizzato: accedo ai dati tramite funzioni ad hoc e le operazioni sui dati alfanumerici e spaziali sono distinti



#### 4.1.2 Modello relazionale esteso

Usare un database tradizionale, per quanto possa sembrare una buona idea, può presentare alcuni di problemi. Si consideri la seguente situazione:

*Voglio rappresentare un parco come un poligono con degli attributi. Devo utilizzare il modello relazionale per rappresentare ciò. Creo due tabelle dove in una ho il nome dei parchi e delle città rispettive e in un'altra tabella la geometria del parco (vettori). Nella prima tabella ho una chiave esterna che referencia la chiave primaria della seconda tabella. Se io dovessi cambiare la geometria, allora devo propagare la modifica di tutte le sequenze di tutti i punti. Poi nel caso dovessimo sapere la geometria a quale parco riferisce, dobbiamo fare un JOIN. Ciò è molto inefficiente. Nel modello relazionale il dominio dell'attributo è definito su valori atomici (prima forma normale), mentre i punti sono coppie di valori; inoltre i punti sono ordinati ed i meccanismi di indicizzazione sono definiti su attributi alfanumerici. Non sono adatti per la indicizzazione di geometrie.*

Nel modello esteso, i dati spaziali sono memorizzati in BLOB (Binary Large Object) di ampie dimensioni. L'ampiezza del campo arriva a 2GB. Ho il vantaggio che ho tutto in un'unica tabella (sia i valori alfanumerici che spaziali), ma non posso fare interrogazioni su dati geometrici (come faccio a fare query su BLOB?) e non è possibile eseguire le query in modo ottimale (si utilizzano indici diversi)

#### 4.1.3 Modello relazionale ad oggetti

Tipi di dati strutturati: immagini, audio/video etc. e definizione di nuovi tipi di dato; ma ho comunque la gestione dei dati basata su modello relazionale (accessi avvengono con SQL e tutte le entità di interesse sono modellate tramite tabelle). Inoltre esistono indici spaziali.

#### 4.1.4 Modello OGC Simple Features

Consorzio che lavora per standardizzare la rappresentazione di dati spaziali di diversa natura. È il **modello di riferimento**. Funziona con PostGIS ed ESRI è OGC compliant.

21 Ottobre

### 4.2 Modelli Mesh

Per mesh si intende un reticolo. È una partizione dello spazio costruito a partire da un insieme di punti campione in tre dimensioni. I vari punti sono

collegati, e parliamo di vertici, spigoli, facce, e stanno spesso a formare dei triangoli. Spesso un punto ha uno o più attributi scalari o vettoriali.

### 4.3 OCG Simple Features

È un insieme di aziende che promuove lo sviluppo di interfacce standard per la gestione dei dati spaziali attraverso:

- Abstract Specifications
- Implementation Standard: documenti tecnici con le specifiche implementative (API), come le **Simple Features**

Per **feature** si intende un'entità nel mondo reale. Le feature possono essere raggruppati in insiemi omogenei chiamati tipi. Vengono chiamata **feature spaziale** un tipo di feature che contiene almeno un **attributo geometrico**. Ogni attributo geometrico ha associato un sistema di riferimento nelle specifiche. Tutte le classi in OCG sono specificate da un diagramma UML, la base del quale è la classe astratta **Geometry**.

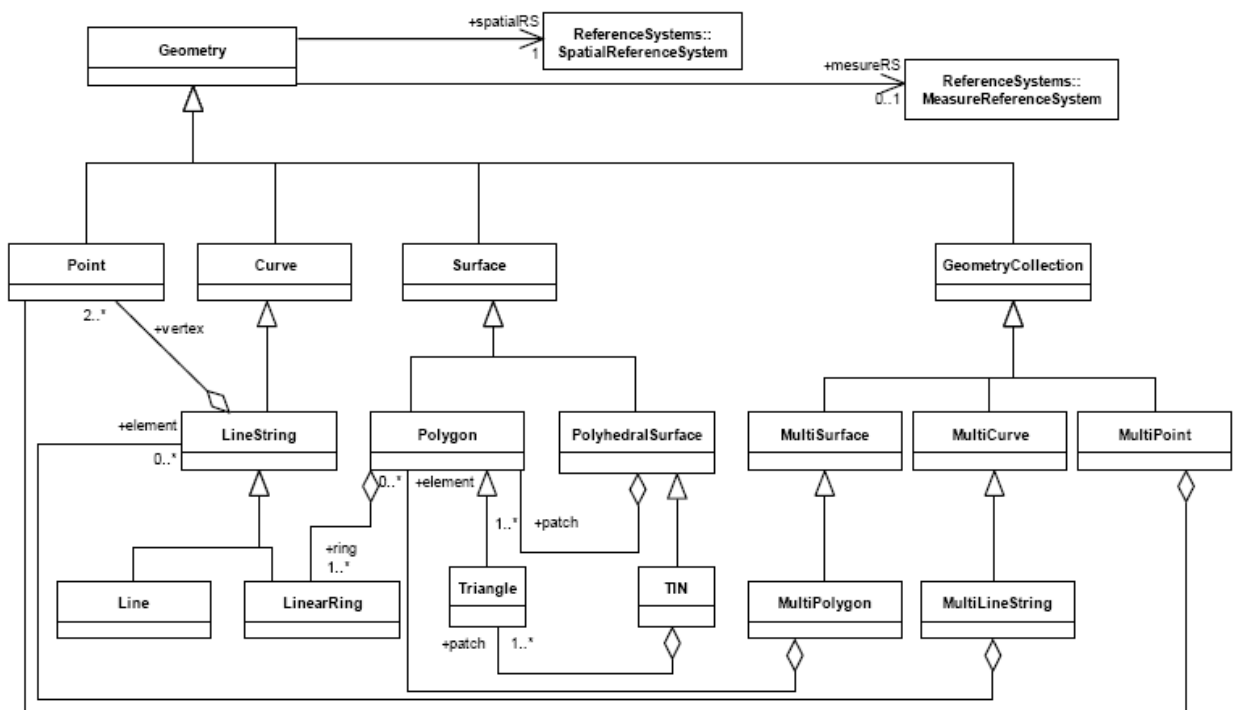


Figure 6: Diagramma UML di OCS SF

Analizziamo ora alcuni tipi degni di nota.

#### 4.3.1 Point

Tipo di dato che contiene delle coordinate e dei metodi per accedere a esse.

#### 4.3.2 Curve

Oggetto geometrico che estende **LineString**, è sostanzialmente una sequenza di punti. Gli estremi di questa curva sono chiamati **frontiera**. Esistono vari tipi di curva:

- **Curva Semplice**: Tipo di curva che non si interseca.
- **Curva Chiusa**: Tipo di curva in cui gli estremi coincidono, quindi la frontiera è nulla.
- **Anello**: curva semplice e chiusa.

#### 4.3.3 Polygon

Oggetto geometrico a due dimensioni. Definito da una frontiera esterna e una o più frontiere interne rappresentate da curve chiuse. Una frontiera interna delimita un **buco**. Le frontiere sono anelli e non si possono intersecare. Un anello può toccare un altro anello solo in un punto tangente. I poligoni devono essere **validi**, ovvero non devono presentare anomalie.

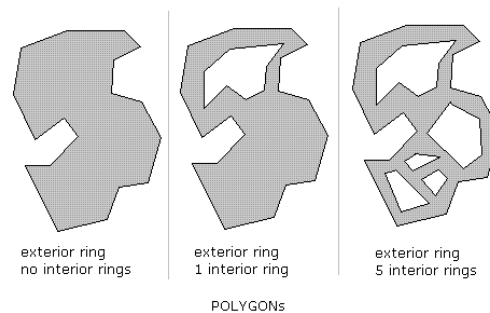


Figure 7: Esempio di Poligoni validi

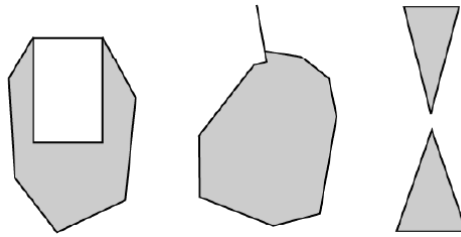


Figure 8: Esempio di Poligoni non validi

Internamente, rappresento un poligono come una sequenza di anelli, ad esempio  $([P1, P2, P3, P4, P1], [P5, P6, P7, P8, P5])$ , dove la sequenza più esterna rappresenta la frontiera esterna, mentre le altre sono i buchi. Esiste un tipo `surface` che estende `polygon`, che mi permette di calcolare il centro, l'area, capire se un punto è contenuto nel poligono ecc.

## 4.4 Geometry Collection

Collezione di geometrie. Viene esteso da varie classi concrete:

- **MultiPoint:** Insieme di `Point`. Viene detto semplice quando non ci sono due punti uguali.
- **MultiLineString:** Insieme di `LineString`. Fornisce un metodo `length()` che somma tutte le lunghezze, e un metodo `isClosed()` che ritorna `true` se tutte le linee sono chiuse.
- **MultiPolygon:** Rappresentazione di collezione poligonale. Gli elementi non possono avere punti in comuni se non punti di contatto su confini.

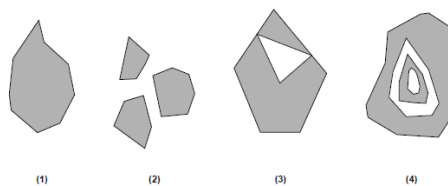


Figure 9: Esempio di MultiPoligono validi

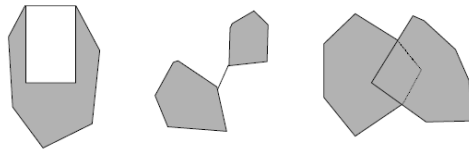


Figure 10: Esempio di MultiPoligono non validi

## 4.5 Formati per la rappresentazione geometrica

Esistono due tipi di formati per la rappresentazione geometrica nelle basi di dati: **Well-know Text (WKT)** che è un formato testuale, e **Well-Known Binary (WKB)** che è un formato binario. In una base dati, i dati sono conservati in WKB, ma vengono trasformati in WKT per essere interpretati.

### 4.5.1 Esempi di WKT

Multipunto con 3 punti:

```
MULTIPOINT(0 0, 20 20, 60 60)
```

Multilinea con 2 linee

```
MULTILINESTRING((10 10, 20 20), (15 15, 30 15))
```

Multipoligono con 2 poligoni

```
MULTIPOLYGON(((0 0,10 0,10 10,0 10,0 0)),((5 5,7 5,7 7,5 7, 5 5)))
```

Collezione di geometrie con 2 punti e una linea:

```
GEOMETRYCOLLECTION(POINT(10 10), POINT(30 30), LINESTRING(15 15,  
20 20))
```

## 4.6 Validità per la rappresentazione Geometrica

Le geometrie sono **valide** quando sono costruite correttamente in conformità con lo standard:

- Geometrie di tipo punto, singole o multiple, sono sempre valide
- Geometrie di tipo linea, singole o multiple, sono sempre valide purchè la geometria comprenda almeno 2 punti distinti
- Una geometria singola di tipo poligono è valida se:

- Ogni è una linea valida con almeno 3 vertici che formano un poligono.
- L'anello esterno contiene ogni anello interno
- Nessun anello può intersecare se stesso o un altro anello. Un anello tuttavia può avere un numero finito di punti in comune con un altro anello
- Una geometria multipla di tipo poligono è valida se le singole geometrie sono disgiunte o hanno un numero finito di punti di contatto

Posso vedere se sono valide con `ST_IsValid(Geometry)` appartenente alla classe `Geometry`. Alcuni sistemi hanno funzionalità aggiunte di validità ad esempio punti duplicati e linee non connesse.

4 Novembre

## 5 Relazioni e Predicati Spaziali

Esistono vari tipi di relazioni:

- **Relazioni Metriche:** tengono conto della distanza
- **Relazioni Topologiche:** tengono conto della forma e della posizione degli oggetti
- **Relazioni di Direzioni:** non me ne frega niente

### 5.1 Relazioni Metriche

Si basano sul concetto di **distanza**. Esistono vari concetti di distanza, come la **distanza di Manhattan**, o il cammino minimo in un grafo.

La distanza può essere 0 solo se due oggetti **x** e **y** sono uguali. Quindi, la distanza fra poligoni non è una distanza accettabile.

Nei sistemi di dati spaziali, usiamo due distanze:

- Distanza euclidea
- Distanza Geodetica

## 5.2 Relazioni Topografiche

Le proprietà topologiche sono caratteristiche che rimangono invarianti anche a seguito di deformazioni (senza strappi).

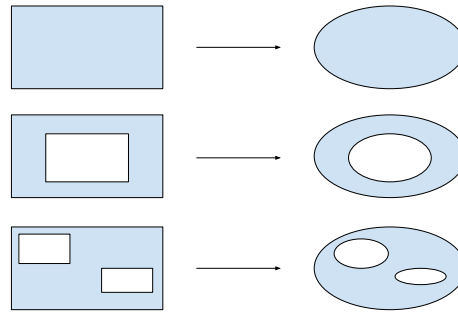


Figure 11: Esempi di poligoni topograficamente equivalenti

Ad esempio un quadrato e un'ellisse sono topologicamente equivalenti, perchè posso "stendere" il quadrato in un'ellisse. Ma se metto un buco nello stesso quadrato, non è più equivalente, perchè non posso più riempire il buco.

Considerando una regione topografica, abbiamo:

- **Interno** della regione
- **Frontiera** insieme di punti del perimetro di una regione

Dati due insiemi di punti , il fatto che un'intersezione sia vuota o non vuota è una proprietà topologica. Posso descrivere la relazione topologica fra due oggetti A e B andando a ragionare sui vari rapporti fra frontiere e interni.

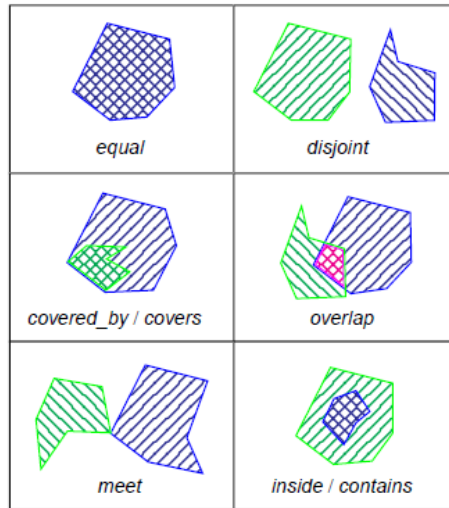


Figure 12: Relazioni in rapporto alle frontiere

Uso quindi la matrice ottenuta per definire le varie relazioni. Otteniamo quindi 6 relazioni:

- equal
- disjoint covered\_by/covers
- overlap
- meet
- inside/contains

Se però voglio considerare anche i poligoni con buchi, uso il modello 9 intersection, chiamato così perchè prodotto con una matrice 3x3.

Nella formulazione delle varie entità, il punto **non ha una frontiera**.

Con la matrice sopraindicata, ho 512 combinazioni possibili, delle quali me ne interessano solo **56 significative** nel modello DE-9I.

In questo modello, i valori delle celle sono:

- **T**: intersezione non vuota
- **F**: intersezione vuota
- **\***: valore non rilevante
- **0**: dimensione 0



- 1: dimensione 1
- 2: dimensione 2

	Boundary (B)	Interno (B)	Esterno(B)
Boundary(A)	$\emptyset/\neg\emptyset$	$\emptyset/\neg\emptyset$	$\emptyset/\neg\emptyset$
Interno (A)	$\emptyset/\neg\emptyset$	$\emptyset/\neg\emptyset$	$\emptyset/\neg\emptyset$
Esterno(A)	$\emptyset/\neg\emptyset$	$\emptyset/\neg\emptyset$	$\emptyset/\neg\emptyset$

Figure 13: Matrice dei bordi

Posso poi convertirle in stringa e usarle in sql. Esistono inoltre una serie di predicati utili a effettuare verifiche su relazioni. È possibile con l'operatore `Relate(A,B,string)`, nel quale è possibile inserire un pattern custom.

- `Disjoint(A,B)`: torna true se l'intersezione di A e B è vuota
- `Intersect(A,B)`: negazione del disjoint
- `Touches(A,B)`: Due oggetti si toccano in uno o più punti ma non all'interno
- `Crosses(A,B)`: Gli interni di due oggetti si intersecano, ma gli oggetti non sono contenuti l'un l'altro
- `Within(A,B)`: A è contenuto in B
- `Contains(A,B)`: A contiene B
- `Overlaps(A,B)`: Come intersect, ma solo fra oggetti con la stessa dimensione geometrica (aree con aree, linee con linee ecc). L'intersezione deve avere la stessa dimensione geometrica degli elementi comparati.
- `Equals(A,B)`: Quando due geometrie sono uguali, ovvero identificano lo stesso insieme di punti.
- `Relate(A,B,string)`: Lo uso per effettuare confronti con relazioni custom

Se effettuo dei join utilizzando come condizione dei dati spaziali, parlo di **join spaziale**.