Mobile Systeme
Grundlagen und Anwendungen standortbezogener Dienste

Location Based Services in the Context of Web 2.0

Department of Informatics - MIN Faculty - University of Hamburg
Lecture Summer Term 2007

Dr. Thilo Horstmann

CLDC

NMEA

MIDP

Google Earth

OpenGIS

SQL

KML

Bluetooth

Mash-Ups

Web 2.0

J2ME

Loxodrome

Euclidean Spaces

RDMS

GPS

PostGIS

GPX

Maps

JSR 179

Polar Coordinates

Threads

API

Today: Introduction to GIS and Spatial Databases

- Introduction
- Terms and Definitions, representations of spatial data
- OpenGIS/PostGIS

Last week ...

- we saw that GPS coordinates represent a (quite accurate) position on earth (i.e., in most cases WGS84 reference ellipsoid)
- But 53,5992N, 9,9337E is not very meaningful to most people
  - you would rather say „Vogt-Kölln-Straße 30“ or „University of Hamburg, Fachbereich Informatik“
- Similarly, the *line-of-sight distance* between two positions may be interesting
  - but in many cases you will be interested in the *shortest route distance* in a road or public transport network
- Another important topic are questions around the relationships of geo objects.
  - Cf. a (topological) public transport map and a street map

Spatial databases and Geographic Information Systems (GIS)

- Key components for the mapping between spatial and descriptive location information as well as for maintaining and deriving relationships between locations in general
- Geographic Information System (GIS) is defined as an information system that is used to input, store, retrieve, manipulate, analyze and output geographically referenced data or geospatial data, in order to support decision making for planning and management of land use, natural resources, environment, transportation, urban facilities, and other administrative records.
- Spatial Database Management System (SDBMS): The storage of geographic data in a prescribed format, including the location, shape, and description of geographical features as well as the relationships between different features. A spatial database usually includes coordinates and topological information.

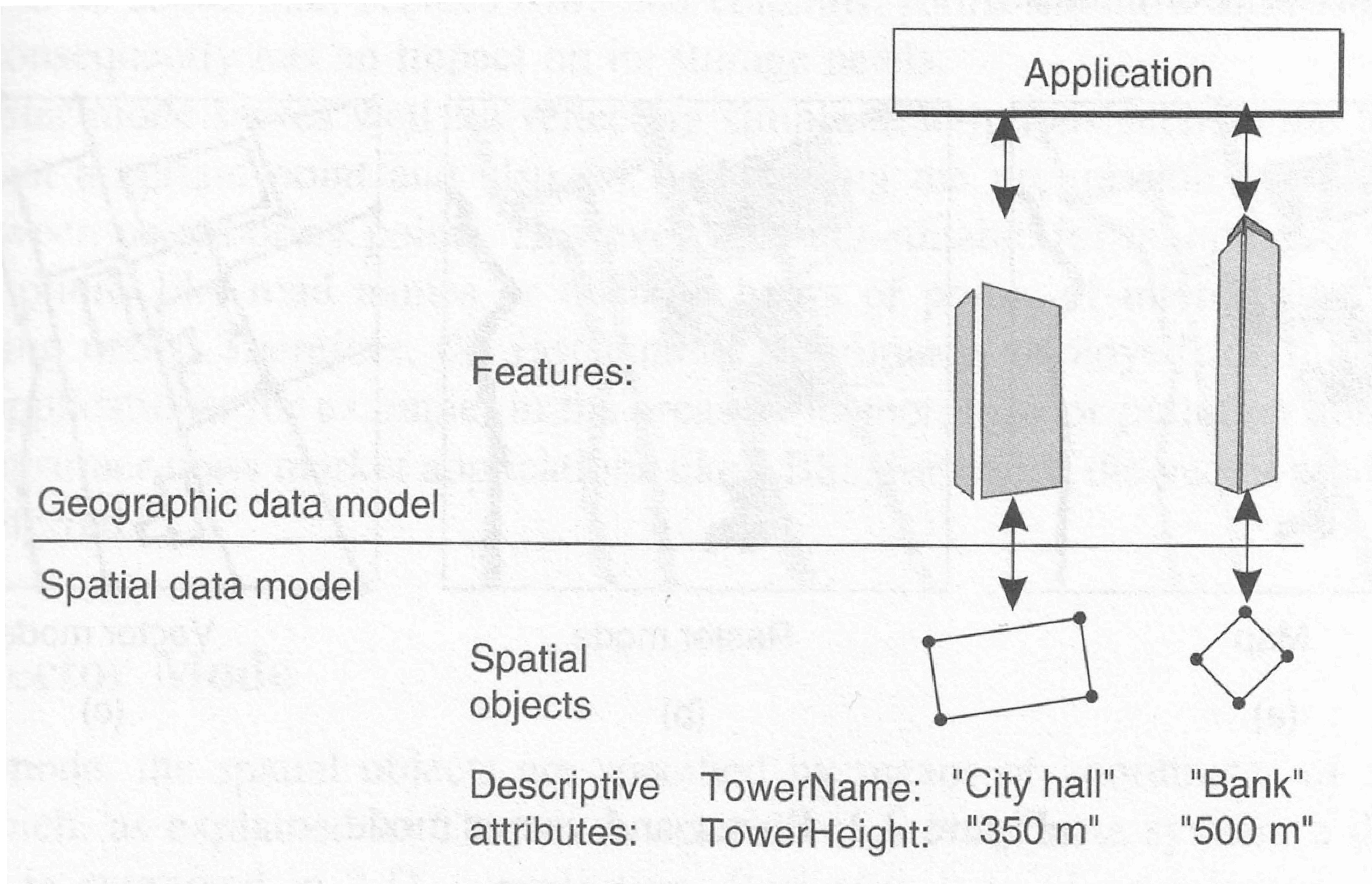
Some important terms: Geocoding

- *Geocoding* is the process of mapping (meaningful) descriptive location information onto spatial location (Latitude, Longitude)
  - The inverse process is sometimes called *reverse geocoding*
- Example geocoders (demo):
  - Google
  - Geonames (www.geonames.org)

Geographic vs. Spatial Data Models

- To understand GIS, it is necessary to distinguish 2 levels of abstractions:
  - ***Geographic data model***: a conceptual view of geographic content in terms of units called ***features***
  - A feature represents a real world entity: a building, road, river, country, city, etc.
  - A feature consists of
    - a spatial component that defines its location, shape, and topological relationship to other entities
    - and a description which provides non-spatial information about the geographic entity like the name of a city, population of a country, etc.

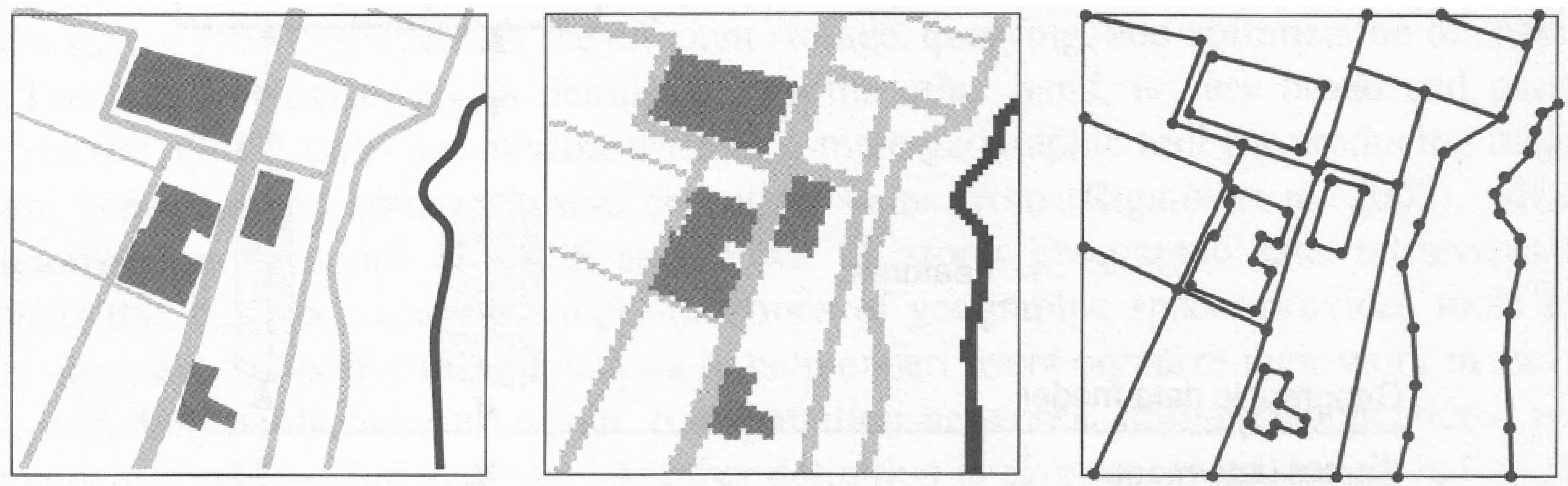
Layered GIS approach



Spatial data model

- ***Spatial data model***:
  - deals with all aspects of physical data management including storage, query processing, and optimization, concurrency and recovery
  - of particular interest: the representation of the spatial components of features called „***spatial objects***“
  - Spatial objects are generally classified into two representation categories:
    - raster mode
    - vector mode

Vector vs. raster map



Map (a) | Raster mode (b) | Vector mode (c)

Raster mode

- bitmap image based on pixels organized in a grid of rows and columns
- in most cased derived from satellite or plane photos
- Shape of a spatial object is reflected by the arrangement of adjacent pixels
  - the position is implicitly given by its pixel coordinates
  - resolution of the grid is important for the quality of the raster data
  - attributes of spatial objects (e.g. descriptions) can be linked implicitly using the value range of pixels: „All pixels with a 0x777777 grey value represent buildings.“
- drawbacks: quality vs. size (memory, bandwidth)
- simple data structures and simple logical and algebraic functions

Vector mode

- Vector model uses discrete points, lines and/or areas corresponding to discrete objects with name or code number of attributes.
- Spatial objects are specified using a reference coordinate system
  - i.e. a coordinate system, a datum, and a projection
- Spatial objects can be simple to complex ranging from points to complex polygons with several boundaries
- high geometric precision, small data size
- map creation process more complex
- more complex data structures (polygons, multi polygons)

Linear and Surfacic spatial objects

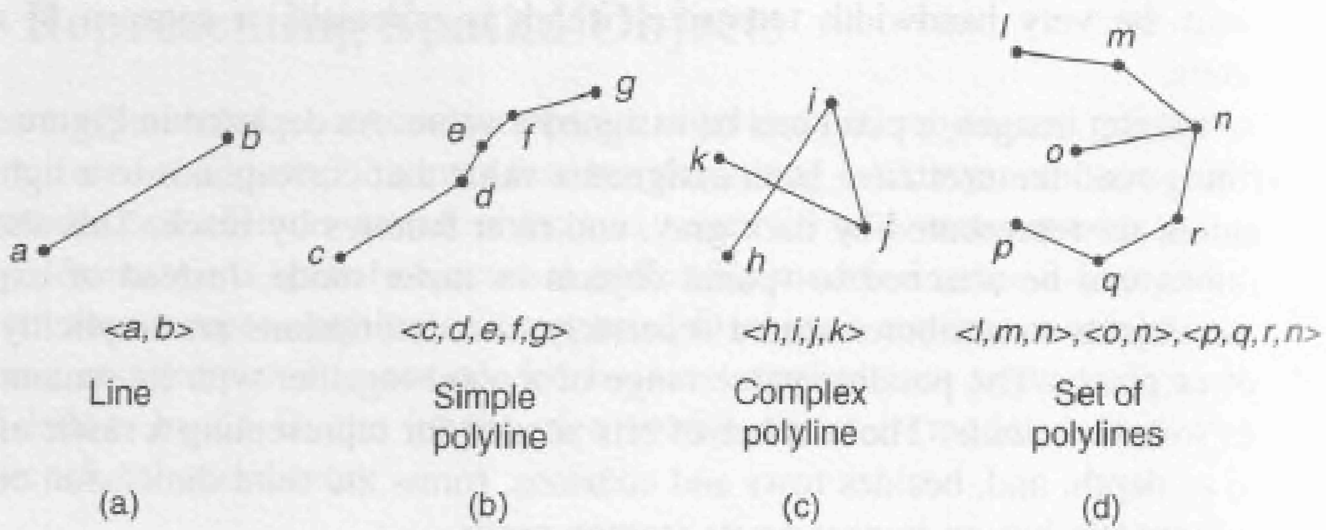
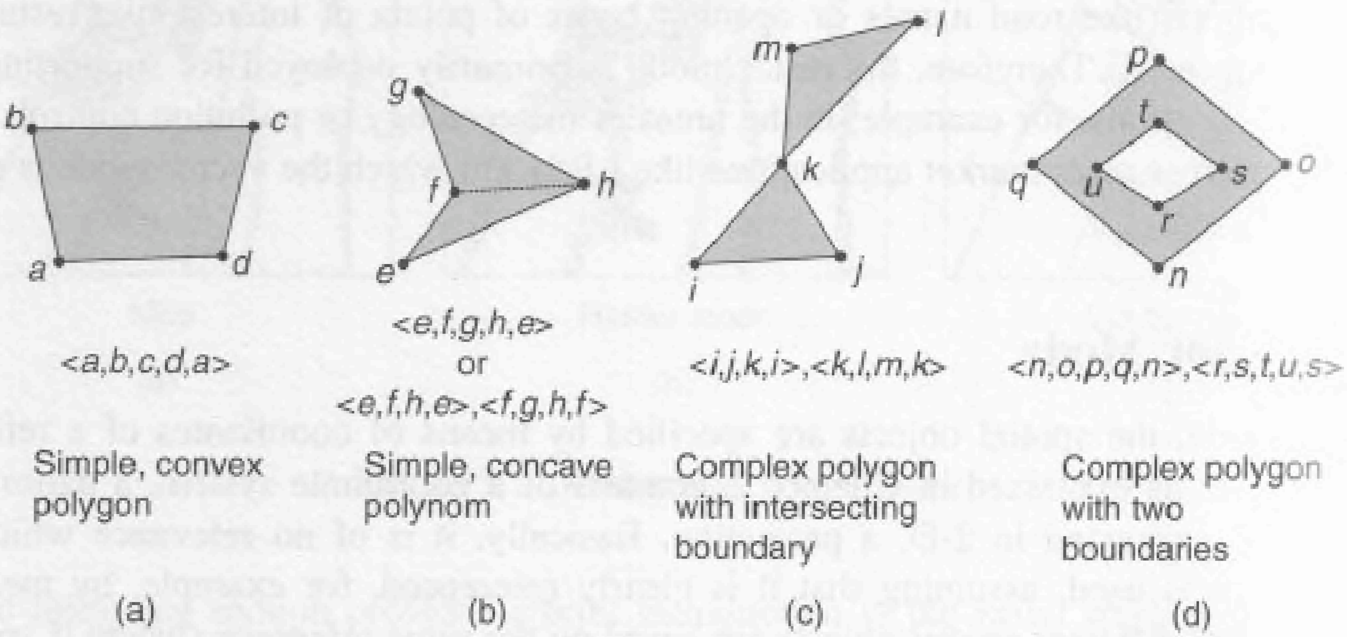


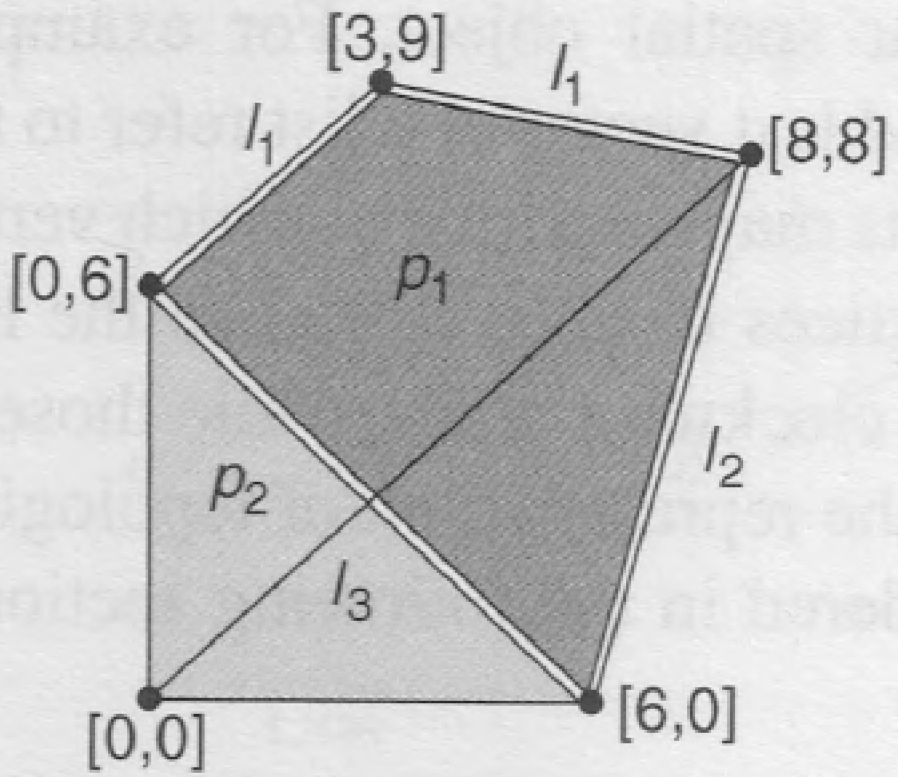
Figure 3.3 Linear spatial objects.



Topological Relationships

- Topological relationships describe relationships between objects in space.
  - proximity, connectivity, adjacency, membership, orientation, etc.
- 2 approaches to derive topological relationships:
  - at runtime using appropriate algorithms
  - explicit modeling of relationships when designing the data model
- We distinguish 3 explicit models:
  - spaghetti, network, topological models

Explicit Modeling of Topological Relationships



Spaghetti model

(a)

Lines:

l_1:<[0,6],[3,9],[8,8]>

l_2:<[6,0],[8,8]>

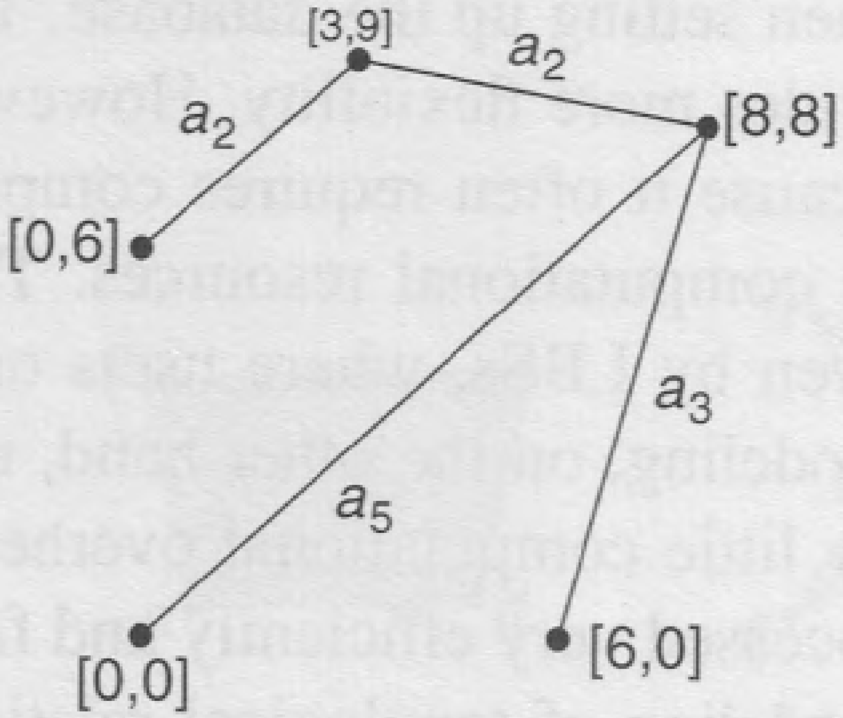
l_3:<[0,0],[8,8]>

Polygons:

p_1:<[0,6],[3,9],[8,8],[6,0],[0,6]>

p_2:<[0,0],[0,6],[6,0],[0,0]>

Explicit Modeling of Topological Relationships



Network model

(b)

Nodes:

$n_1:[[0,0],<a_5>]$
$n_2:[[0,6],<a_2>]$
$n_3:[[8,8],<a_2,a_3,a_5>]$
$n_4:[[6,0],<a_3>]$
$n_5:[[3,3],<a_5,a_6,a_7,a_8>]$

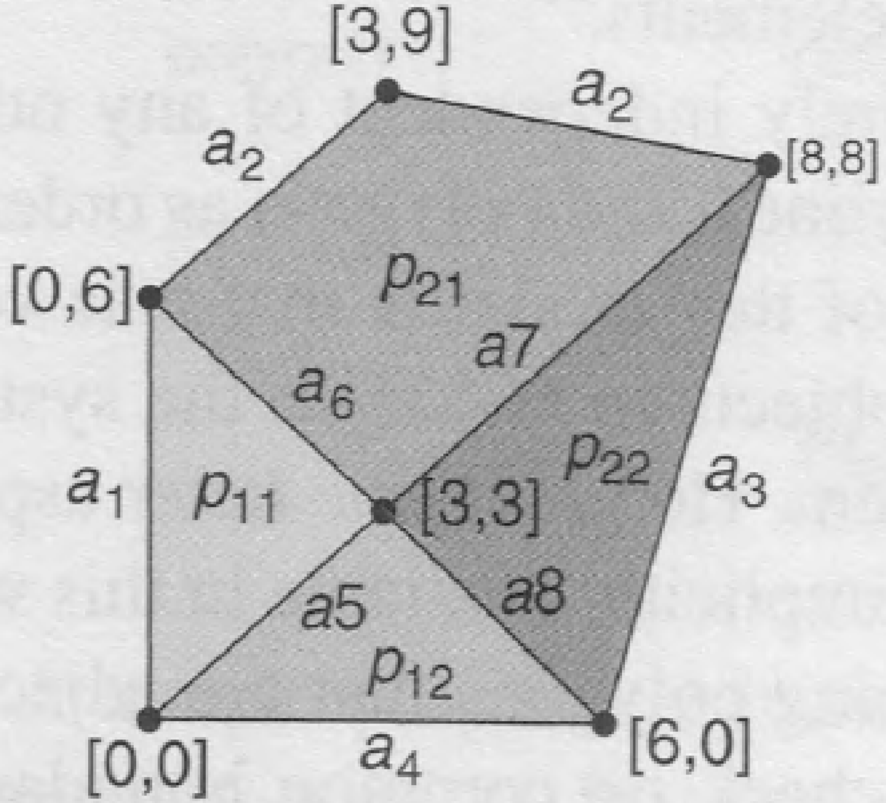
Arcs:

$a_2:[n_2,n_3,<3,9>]$
$a_3:[n_3,n_4,\_]$
$a_5:[n_1,n_3,\_]$

Polygons:

$p_1:<[0,6],[3,9],[8,8],[6,0],[0,6]$
$p_2:<[0,0],[0,6],[6,0],[0,0]>$

Explicit Modeling of Topological Relationships



Topological model

(c)

Nodes:

$n_1:[[0,0],<a_1,a_4,a_5>]$

$n_2:[[0,6],<a_1,a_2,a_6>]$

$n_3:[[8,8],<a_2,a_3,a_7>]$

$n_4:[[6,0],<a_3,a_4,a_8>]$

$n_5:[[3,3],<a_5,a_6,a_7,a_8>]$

Arcs:

$a_1:[n_1,n_2,\_,p_{11},\_]$

$a_2:[n_2,n_3,\_,p_{21},<3,9>]$

$a_3:[n_3,n_4,\_,p_{22},\_]$

$a_4:[n_4,n_1,\_,p_{12},\_]$

$a_5:[n_1,n_5,p_{11},p_{12},\_]$

$a_6:[n_2,n_5,p_{21},p_{11},\_]$

$a_7:[n_3,n_5,p_{22},p_{21},\_]$

$a_8:[n_4,n_5,p_{12},p_{22},\_]$

Polygons:

$p_{11}:<a_1,a_6,a_5>$

$p_{12}:<a_4,a_5,a_8>$

$p_{21}:<a_2,a_7,a_6>$

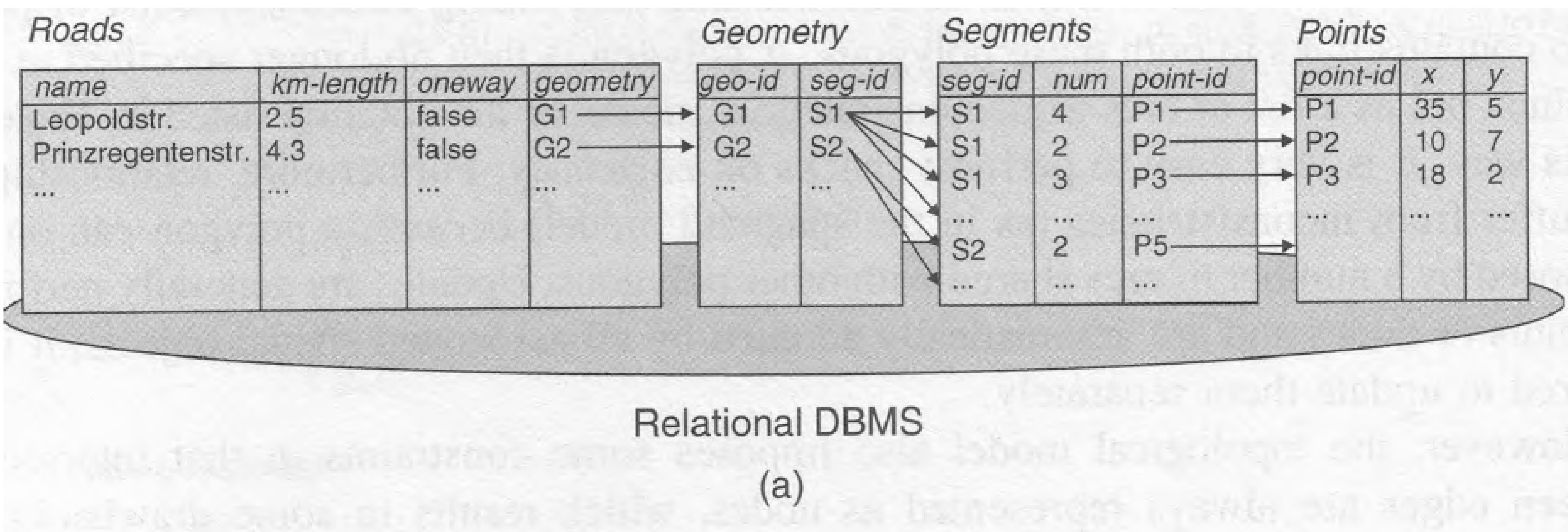
$p_{22}:<a_3,a_8,a_7>$

Data Approaches of Spatial Objects

- How to store geo data in a database?
  - there are many database approaches: relational DBMS, object oriented DBMS, object oriented relational DBMS, hierarchical DBMS, etc.
- Basic idea: use or extend conventional RDBMS despite known drawbacks
  - standardized query language
  - popularity
  - available tools and software components (OR mapper)
- Literature distinguish 3 approaches:
  - relational, loosely coupled, and integrated approach

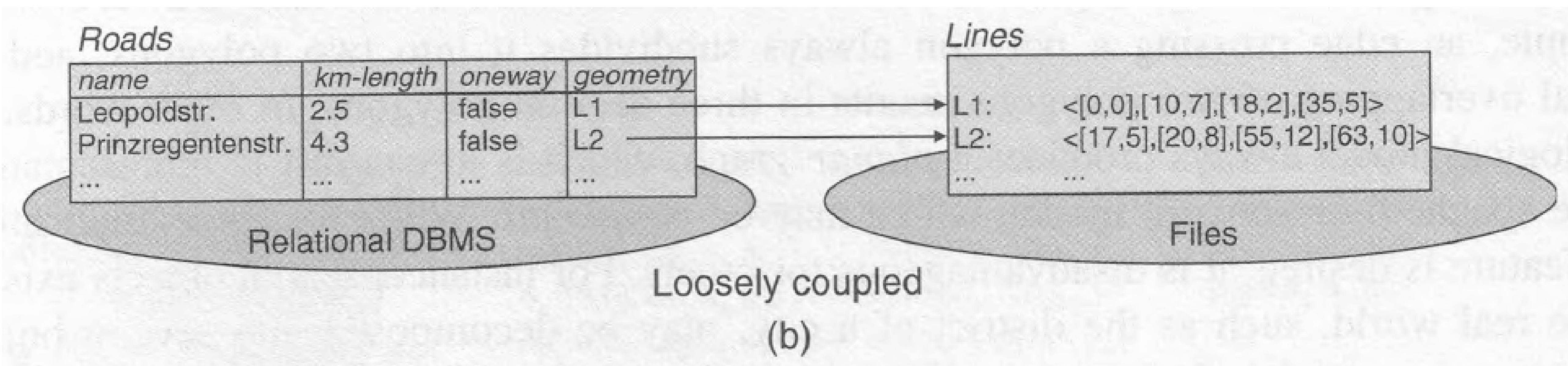
Database Approaches: RDBMS

- model geo data using standard data types like string and integer
- spatial objects are composed by relations between tables



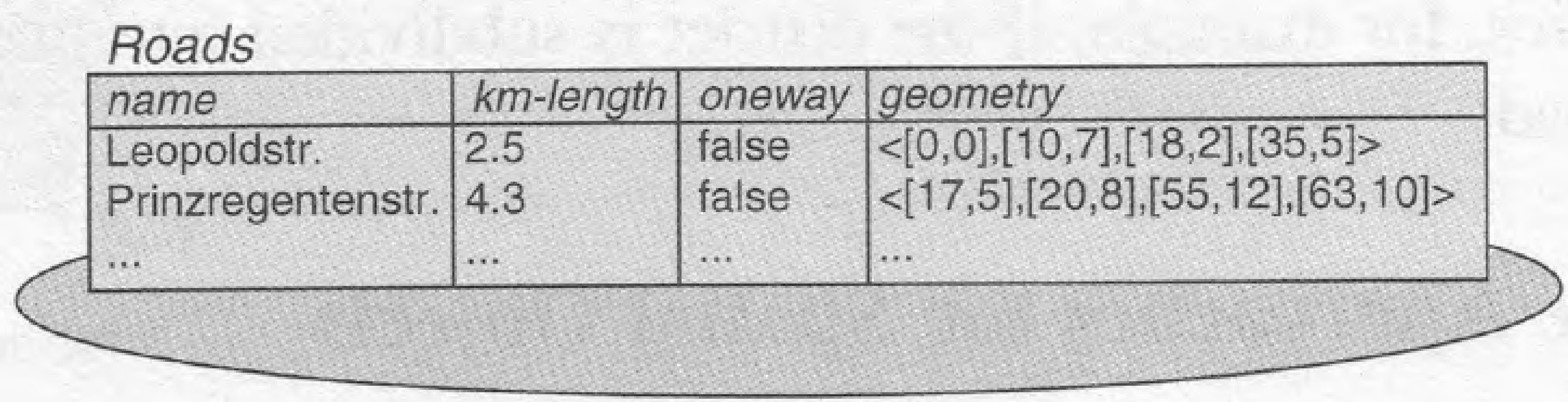
Loosely coupled approach

- idea: use 2 subsystems: one for storing descriptive (geo) attributes, another for maintaining spatial data, i.e. a RDBMS and a file system
- more suitable for storing geo data compared to the RDBMS approach
- data base features including querying, recovering, optimizing etc. not applicable



Integrated Approach

- Idea: Introduce new spatial data types like lines, polygons etc. and operations
  - this implies to extend the query language (SQL)
- results in efficient query processing
- used in most commercial GIS products. Supported by Oracle and Postgres
- Standardized in OpenGIS



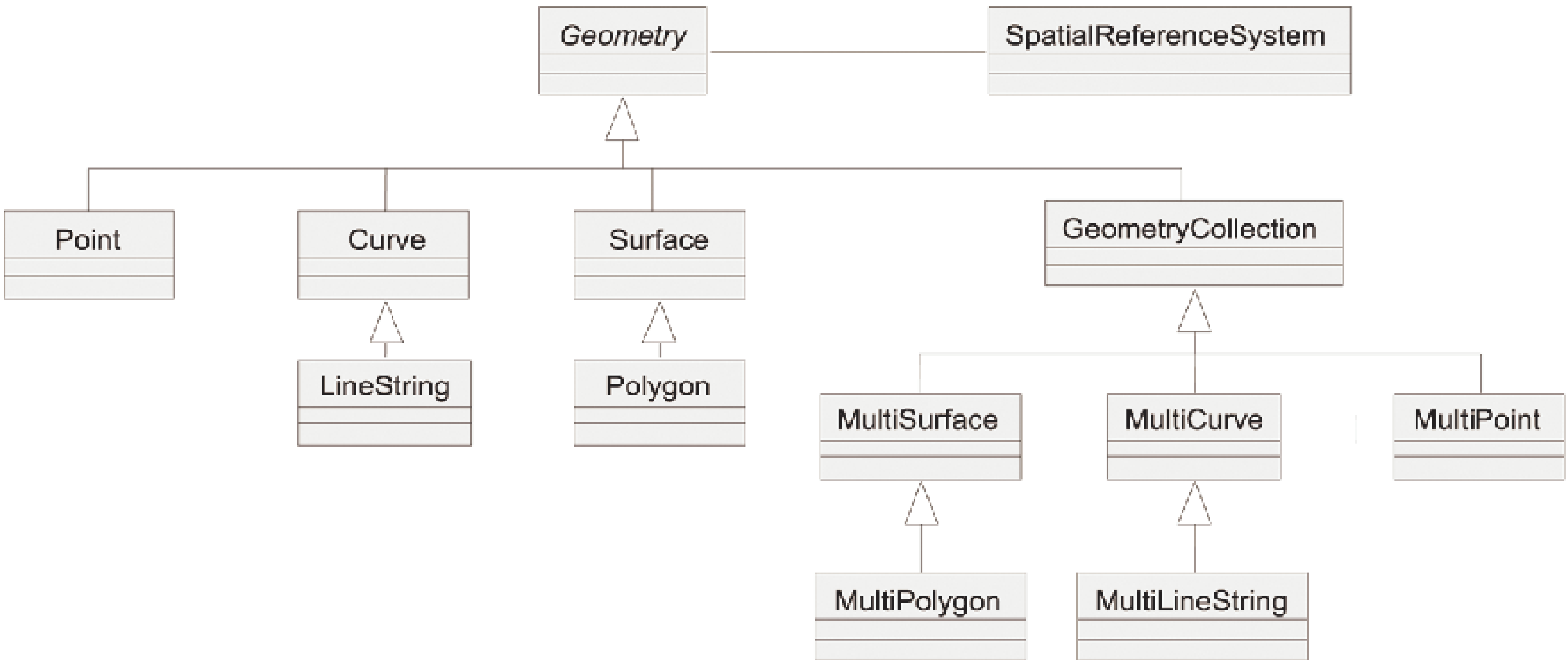
Roads

name	*km-length*	*oneway*	*geometry*
Leopoldstr.	2.5	false	<[0,0],[10,7],[18,2],[35,5]>
Prinzregentenstr.	4.3	false	<[17,5],[20,8],[55,12],[63,10]>
...	...	...	...

Integrated approach

(c)

Typical Spatial Data Types (OpenGIS)

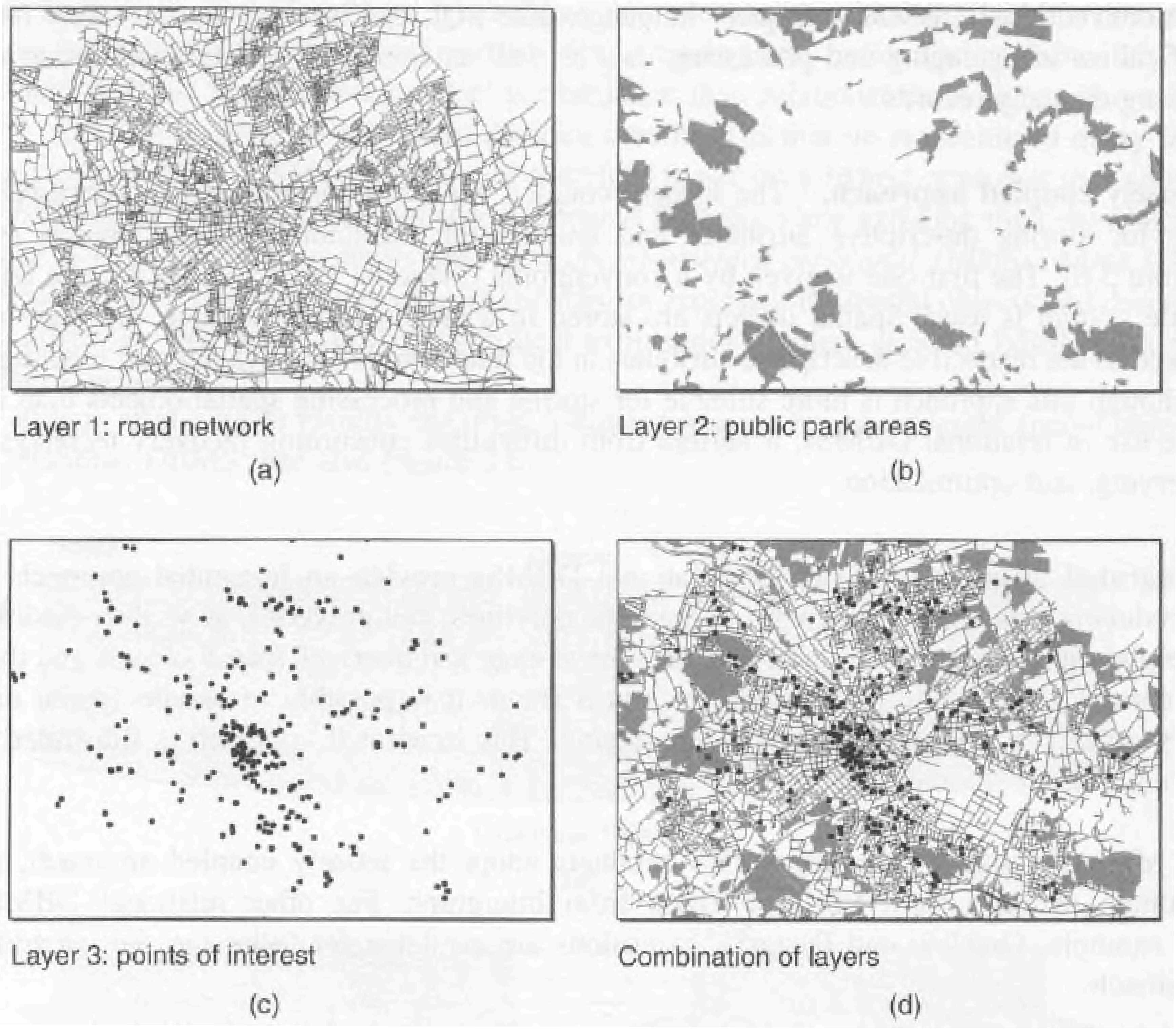


Themes

- The geographic data model provides another level of abstraction
- Goal: Compose spatial content by some sort of overlay concept
  - e.g. in a certain area you may want to have weather data in one layer and traffic information in another. Layers shall be displayed separately or together
- The appearance and structure of of features as well as the topological relationships among them are fixed by ***themes***.
- ***Theme*** (or data layers): spatially related sets of data in a GIS

Features and Themes

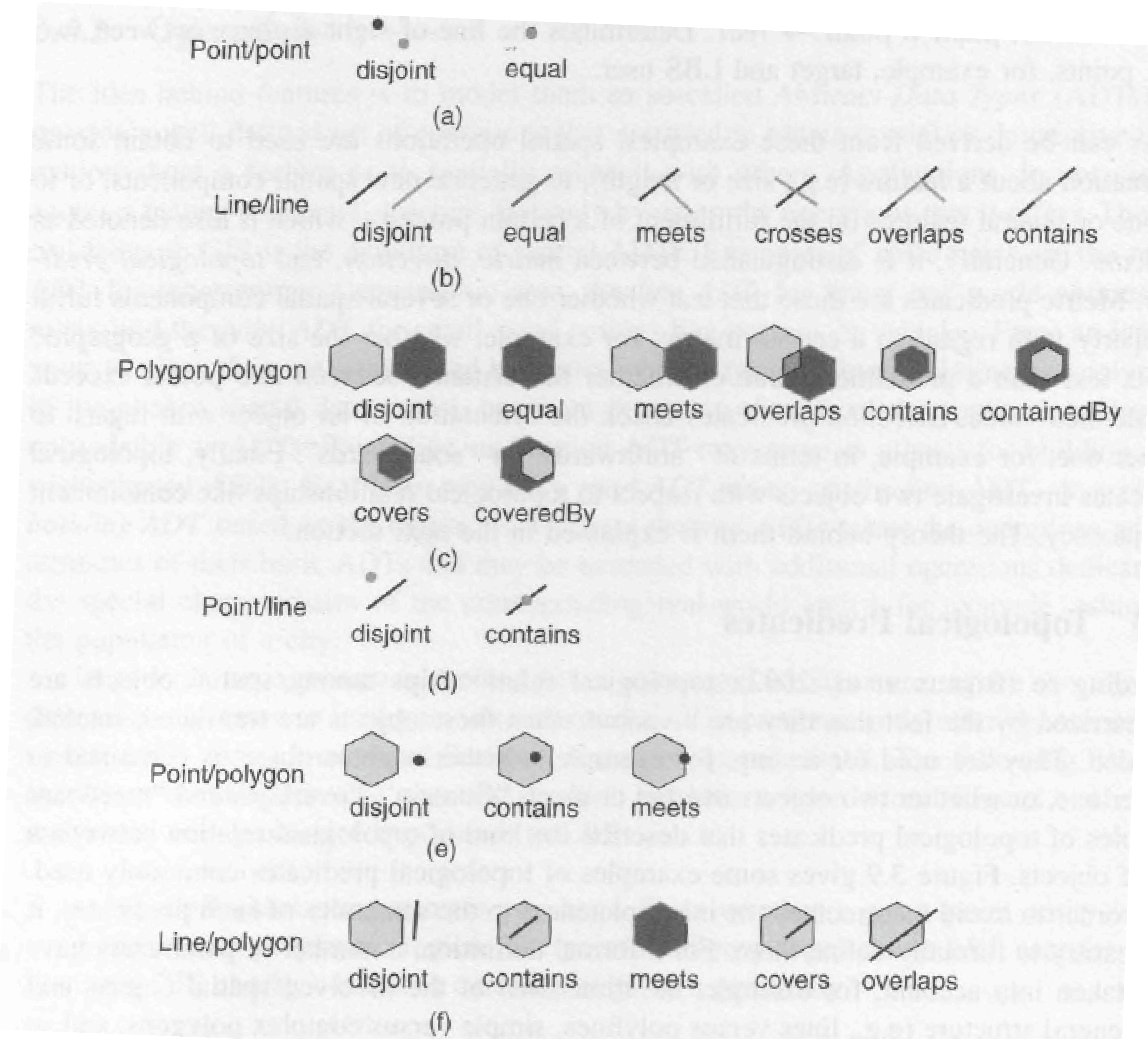
- Feature: represents a real word entity by a spatial component and a meta data (description)
- Theme: defines the appearance and structure as well as the topological relation ship between features



Operations using Abstract Data Types (ADT)

- PointInRegion: point × region ⇒ bool. Tests whether a given point entity is contained in a geography region. The point ADT may represent an LBS user for which the GIS checks whether it currently stays in a given city.
- Overlaps: region × region ⇒ bool. Tests whether two geographical areas overlap.
- Intersection: region × region ⇒ region. Computes and returns die intersection of two geographical areas.
- Meets: region × region ⇒ bool. Tests two given geographical areas on adjacency. Can be used for selecting points of interest in die user‘ s current and all neighboring districts of a city.
- AreaSize: region ⇒ real. Returns the area size of a geographical area, for example, the size of die country the user stays in.
- PointInLine: line × point ⇒ bool. Tests die intersection between a point and a line. May be used to derive the user‘s location in terms of a station or line of a public transportation network.
- ShortestRoute: point × point ⇒ line. Returns the shortest route between two points, for example, as needed for navigation services.
- Length: line ⇒ real. Calculates the length of a line, for example, a route in a navigation application.
- Distance: point × point ⇒ real. Determines the line-of-sight distance between two points, for example, target and LBS user.

Topological Predicates



Intersection Matrix

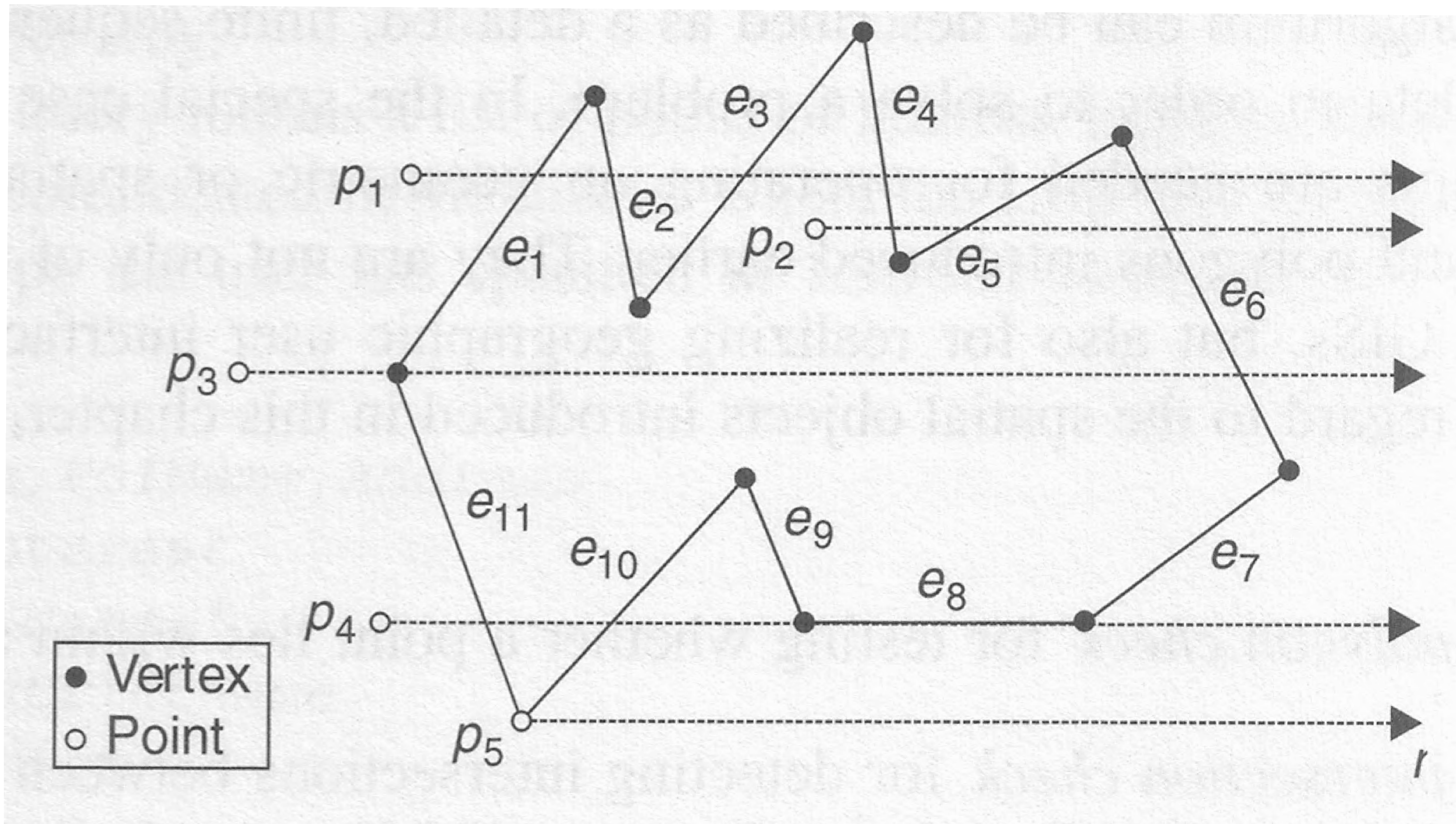
- The 9-Intersection Matrix formally defines spatial object relations using a 3x3 matrix for two given point sets A and B.
- For a given point set A
  - A^0 is the interior of A
  - δA the set of border points
  - A^{-1} the exterior (complement) of A

$$I_9(A, B) = \begin{bmatrix} A^0 \cap B^0 & A^0 \cap \delta B & A^0 \cap B^{-1} \\ \delta A \cap B^0 & \delta A \cap \delta B & \delta A \cap B^{-1} \\ A^{-1} \cap B^0 & A^{-1} \cap \delta B & A^{-1} \cap B^{-1} \end{bmatrix}$$

Typical Geo algorithms

- the point-in-polygon check for testing whether a point lies within a polygon
- the polyline-intersection check for detecting intersections between polylines,
- the polygon-intersection check for detecting intersections between polygons,
- the polygon-intersection computation for calculating a new spatial object from the intersection of two polygons,
- the area-size computation for calculating the area of a polygon,
- the length computation for calculating the length of a line or polyline,
- the shortest path computation for calculating the shortest path between two points in a network.

Example: Point in polygon check



Demo of OpenGIS/PostGIS

This Lecture

- Küpper, A.: Location-based Services - Fundamentals and Operation, John Wiley & Sons, 2005
  - Chapter 3
- de Lange, N.: Geoinformatik in Theorie und Praxis, Springer, 2006
  - Chapter 9.2, 3.5
- OpenGIS, PostGIS
  - www.opengis.org, www.postgis.org

Thank you!

Dr. Thilo Horstmann

e-mail: thilo.horstmann@gmail.com
blog: http://www.das-zentralorgan.de