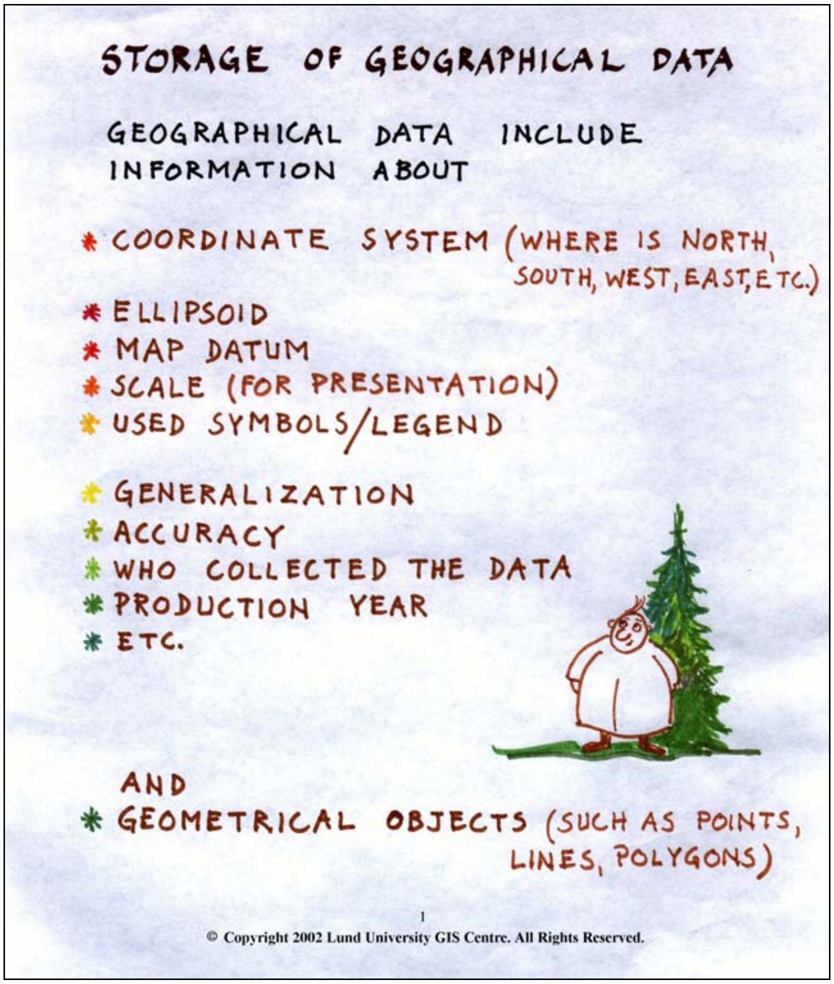
Dept of Physical Geography and Ecosystems Analysis, and
GIS Centre at Lund University

GIS401: Introduction to Geographical Information Systems (GIS)

An Internet-based university course
given within Lund University Master's Programme in GIS (LUMA-GIS)

VECTOR STRUCTURE

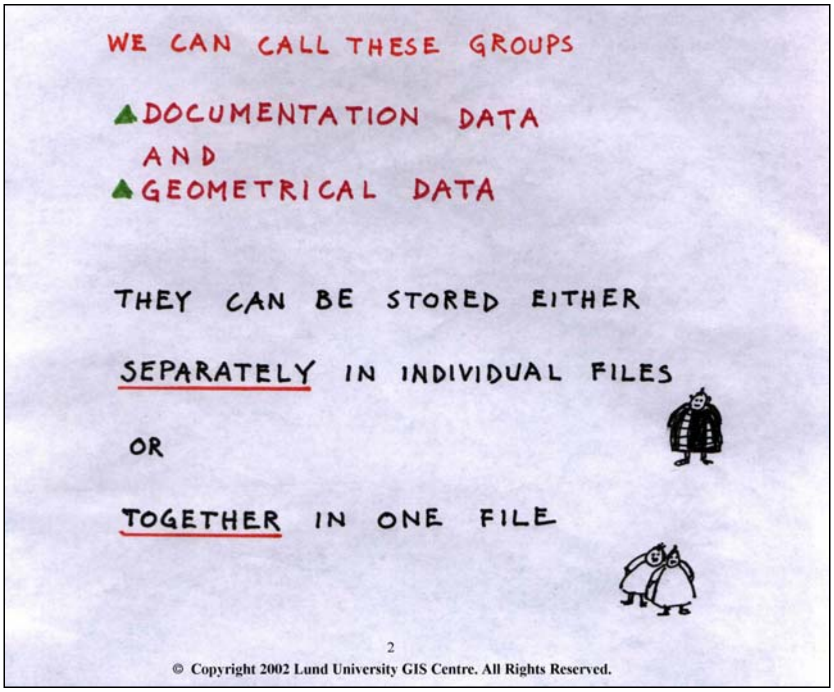
LECTURE



Slide 1

This chapter is about different data structures in GIS and how to store information (mainly the geographical data). In vector GIS, geographical data are points, lines and polygons. In general, the information for these features can be divided in three parts:

- Information about the geographical objects – X, Y and Z co-ordinates
- Additional information – Accuracy, Scale, Producer, Production year, etc
- Attribute data

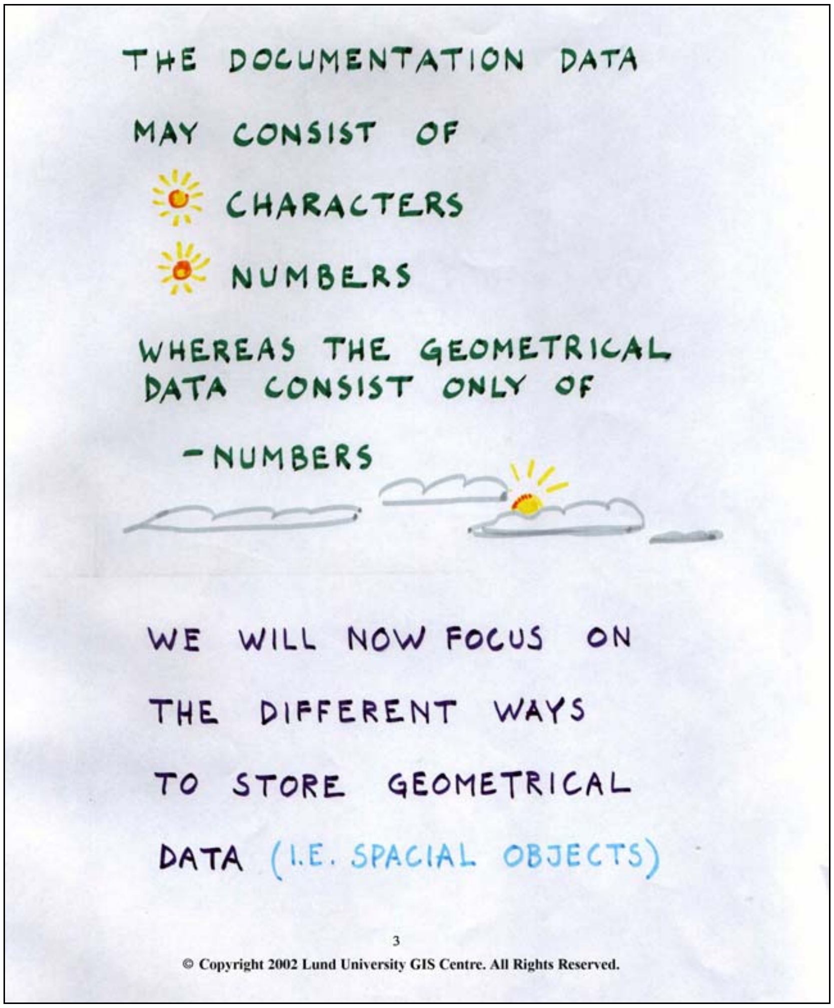


Slide 2

The geographical information can be further divided in two main groups:

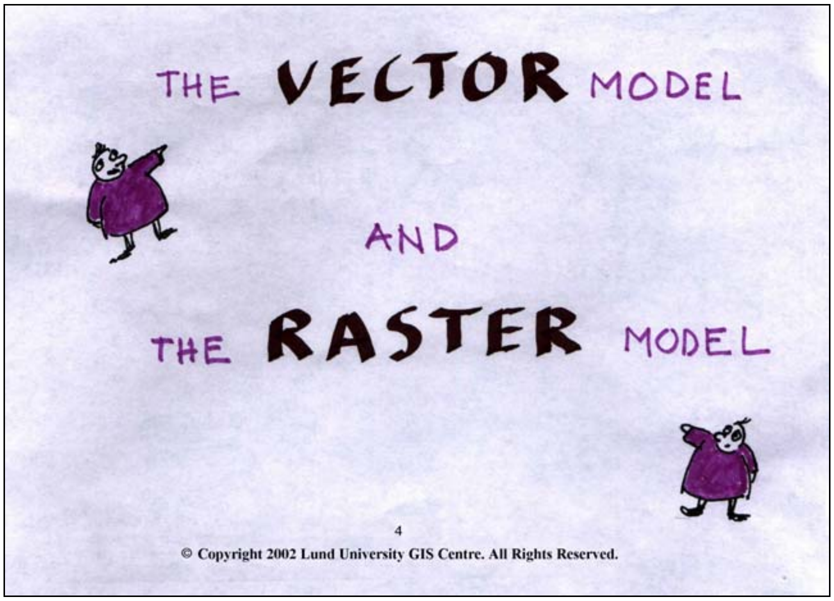
1. Documentary data that describes the data
2. Geometrical data that describes the location (co-ordinates).

This information can be stored in two different ways; in the same file or in separate files. If stored separately, one file will contain the geometric data and one file the documentary data. Both files must normally be imported to the GIS to make a complete geographical database. Data can also be stored together in a single file. The documentation data is then called a "header", telling something about the rest of the data in the file.

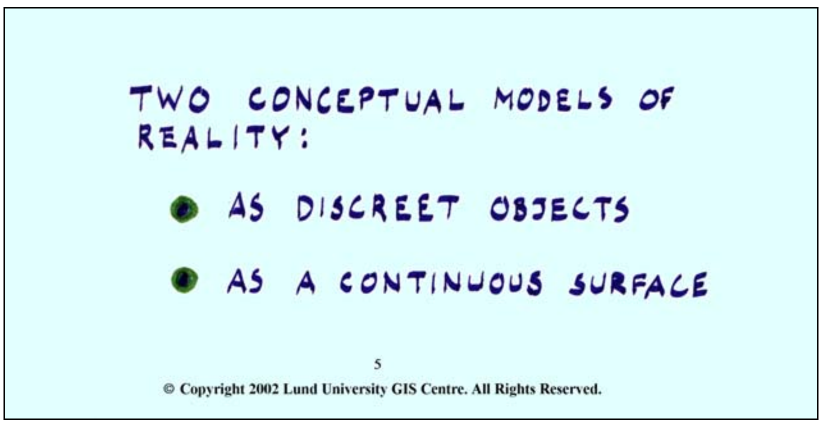


Slide 3

The documentation data can be alphanumerical (characters, i.e. text and numbers) but could also be a video clip, an image etc. The geometric data (the co-ordinates) is always numerical. The lecture will focus on how the geometric objects (points, lines and polygons) are stored in the computer.



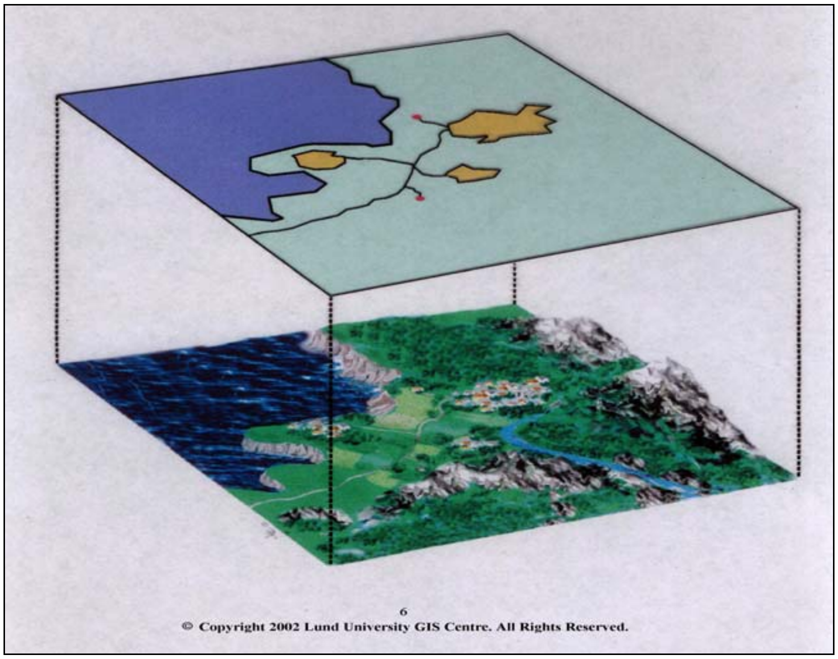
Slide 4
There are two conceptually different data models available for storing GIS-databases; the vector model and the raster model. Most advanced GIS software can handle both models and even convert data between the two.



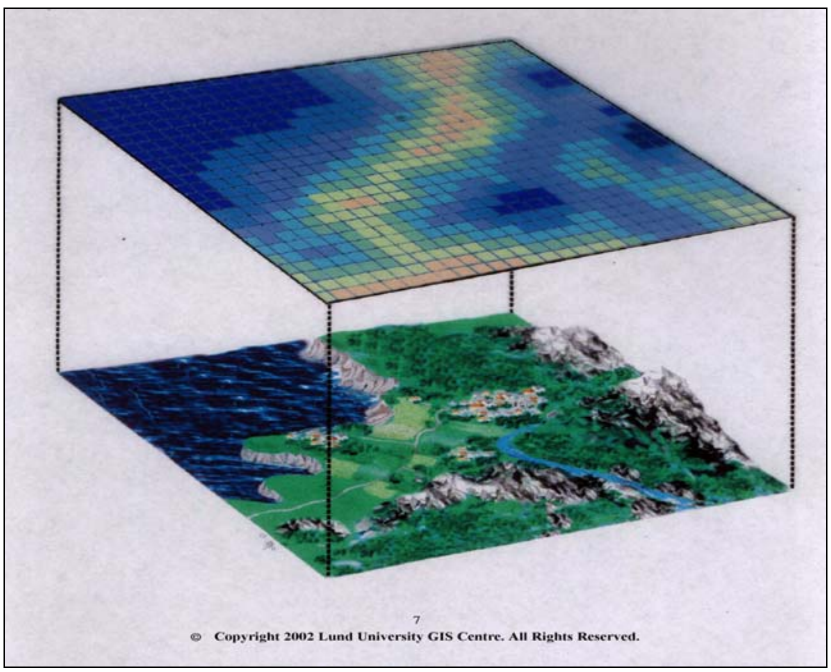
Slide 5
The real world can be described using two conceptually different models:

1. As discrete objects, possible to represent as points, lines or polygons.
2. As a continuous surface with no discrete or distinct borders, like temperature and precipitation.

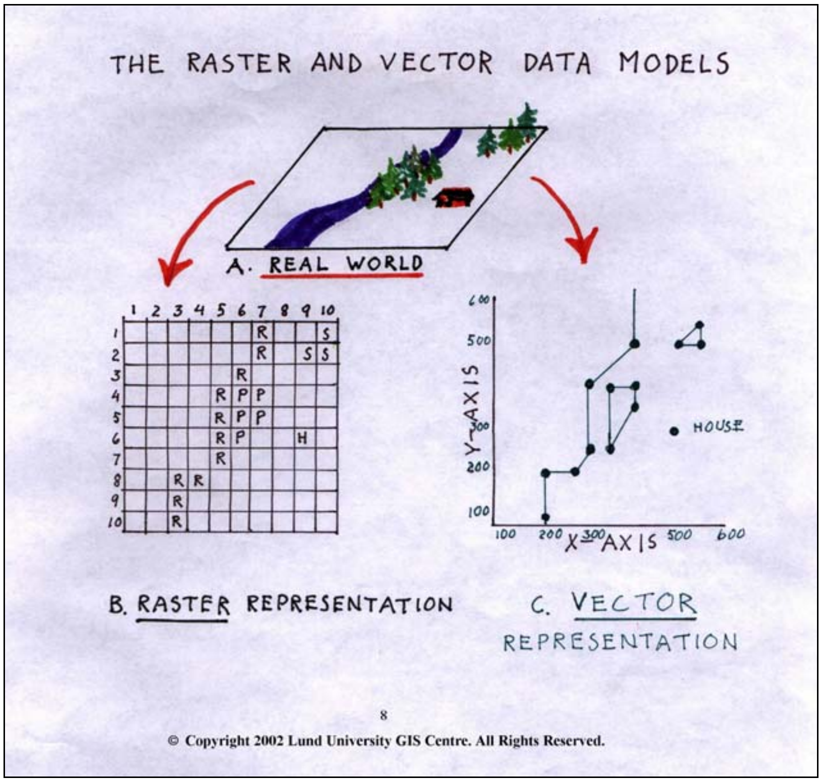
To map houses and roads in an area, discrete objects are more suitable to use since these have a defined spatial extent, but to make a topographic map, a continuous surface should be used since topography has a continuous spatial variation.



Slide 6
This is an example of a vector model representation of the real world, picturing a lake, some roads, villages and some smaller properties, all separated from each other by defined borderlines.

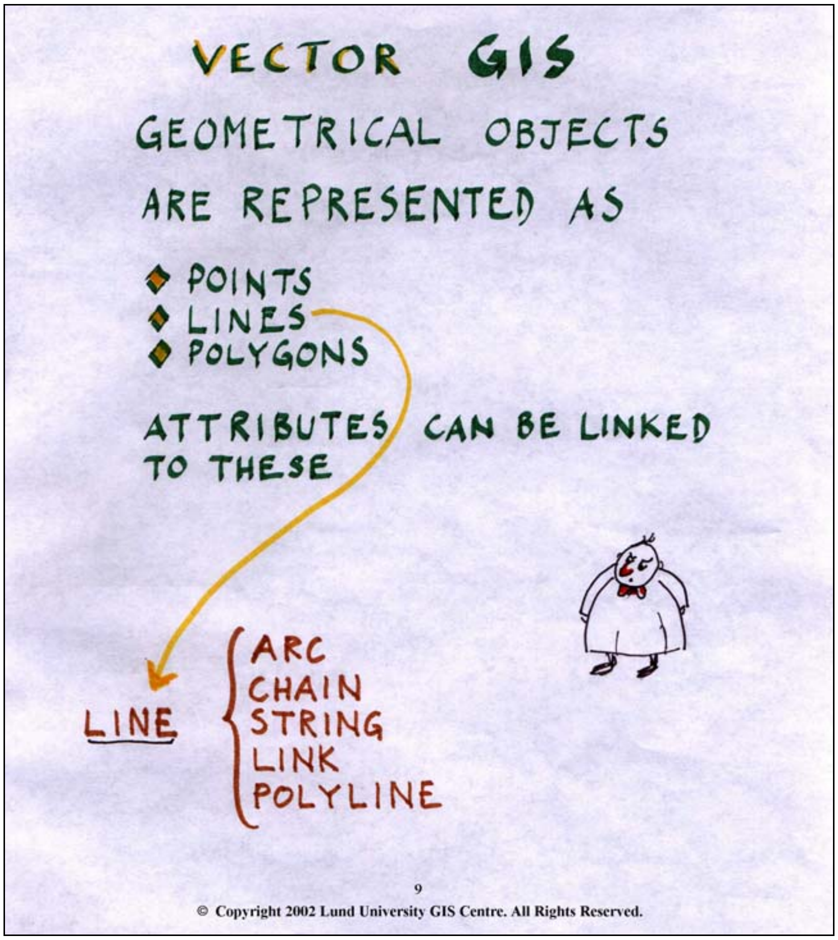


Slide 7
This is a raster model representation of the real world picturing gradual changes in topography (a continuous surface) with no distinct boarder lines.

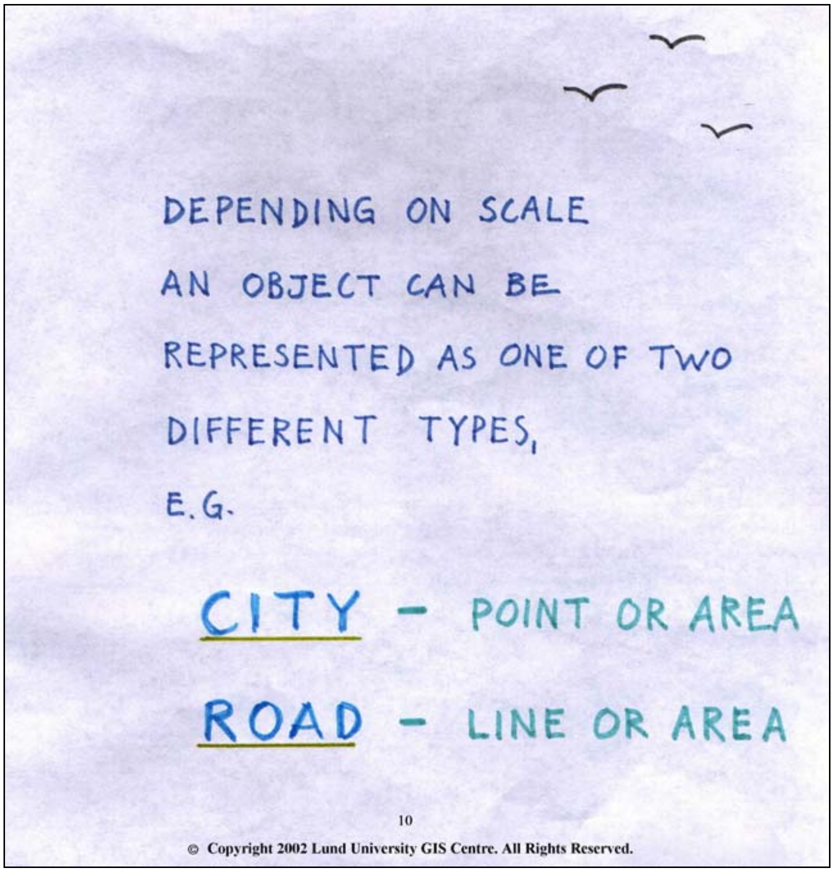


Slide 8

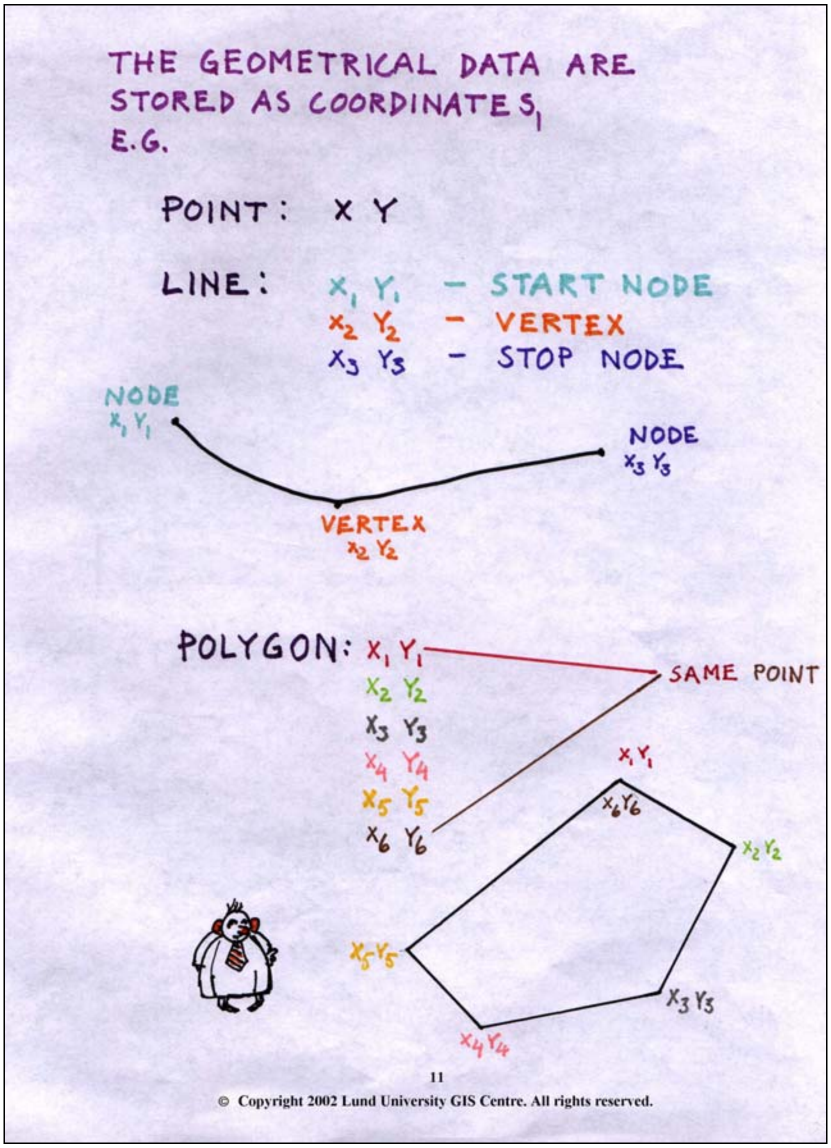
The real world (A) can be represented as Raster model (B) or Vector model (C). When representing as raster, a raster/matrix/grid with a fixed cell size is placed over the area, and each raster cell is coded with a code representing the feature in that particular area. In this case, cells covering areas of the river are coded as “R”, cells covering pine forest are coded “P”, cells covering Spruce forest are coded “S” and the cell covering the house is coded “H”. The vector model (C) uses a totally different way for representing the different objects in the real world. The river is represented as a series of lines, with breakpoints every time the line changes direction. The x and y co-ordinate for each breakpoint is stored and the river is represented as a line in the vector model database. An attribute defining the line as river “R” is connected to each line segment. The forests are represented in a similar way by storing the x and y co-ordinate for each breakpoint of the forest’s border line. Defining this line as boarder line to an area permits construction of a surface or polygon and an attribute describing the type of forest is connected to it. The house is stored as a point, described with a single pair of co-ordinates.



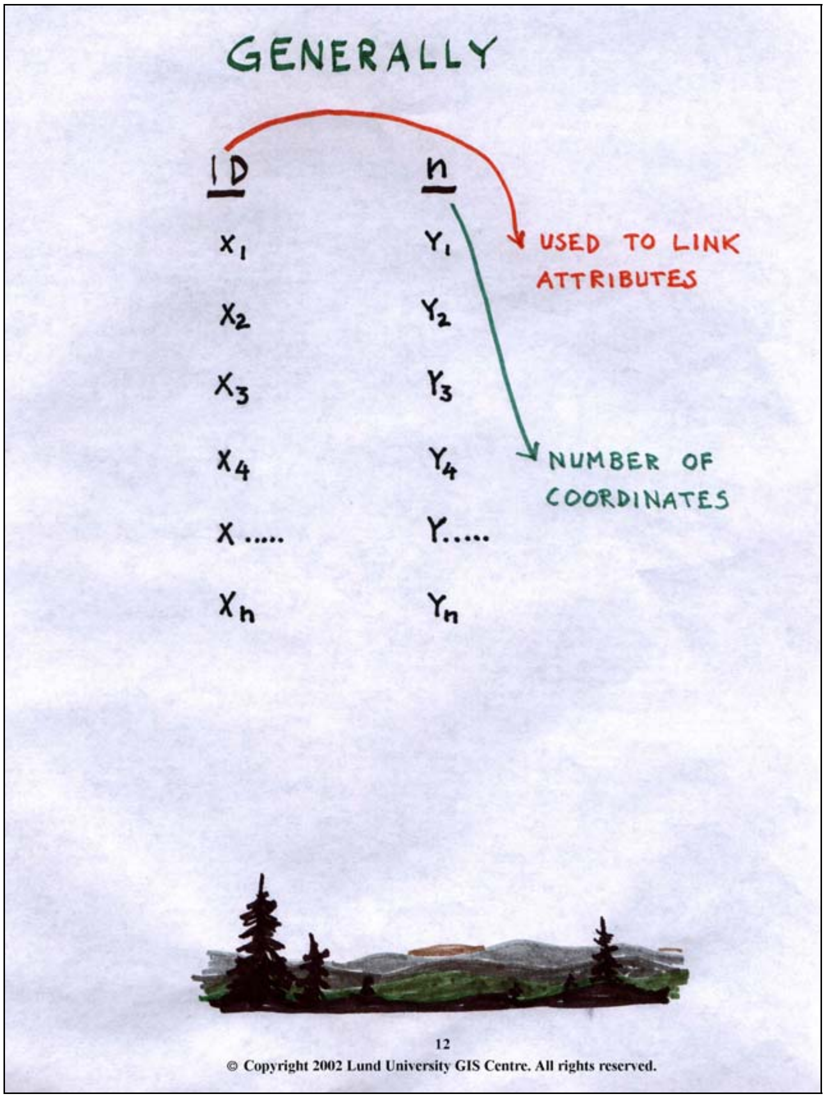
Slide 9
Each object type can have an indefinite number of attributes connected to it, and the attributes are used for describing different aspects of the object. Note that in the literature, and in different software manuals, the objects may have different names, all meaning the same type of object, e.g. a line can also be called arcs and/or polylines.



Slide 10
Depending on the map scale, an object in the real world may be represented by different geometric objects. A city on a world map will probably be represented as a point, while on a more detailed map, the city will be represented as a polygon. The same is valid for lines and points. The type of geometric object selected for representing different real world objects is a matter of generalisation and purpose of presentation, which is important to keep in mind when working with GIS-databases.



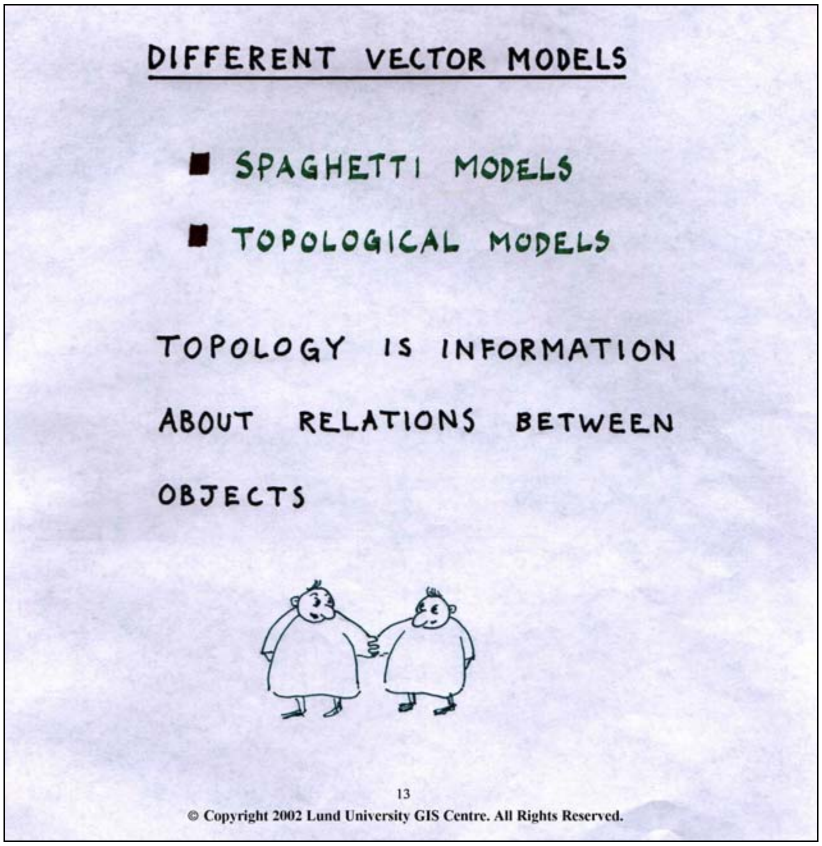
Slide 11
The different geometrical objects in the vector model are stored as one or several pairs of co-ordinates. The basic geometrical object, the point, is easiest to represent since the point has no spatial extent and can be represented by a single pair of co-ordinates. A linear object is more complicated since it has a one-dimensional extension and consequently must be represented with a series of co-ordinate pairs describing this extension. In the simplest case of a straight line between two points, the vector model stores the start point and the end point of the line. More complex lines need several pairs of co-ordinates, one at each breakpoint, where the line changes direction. Smooth real world objects, like for example rivers that change direction more or less constantly, will have to have an indefinitely large number of breakpoints (pairs of co-ordinates) in order to be represented correctly. In GIS terminology, the start point and end point of a line are called "node" and the breakpoints in between are called "vertex" (vertices in plural). Storing of polygon features is similar to line storing. The polygon is defined by its border line, which in turn is described by start node, stop node and a number of vertices. The only difference is that the co-ordinates for the start and stop nodes must be the same, otherwise the polygon will not be closed. The GIS software can thus distinguish between the three geometric objects simply by looking on the list of co-ordinate pairs used to describe it. If it is a point, there is only one pair of co-ordinates, if the object is composed of several pairs of co-ordinates it must be a line and if the co-ordinates for the start node and stop node are the same, the object is a polygon.



Slide 12

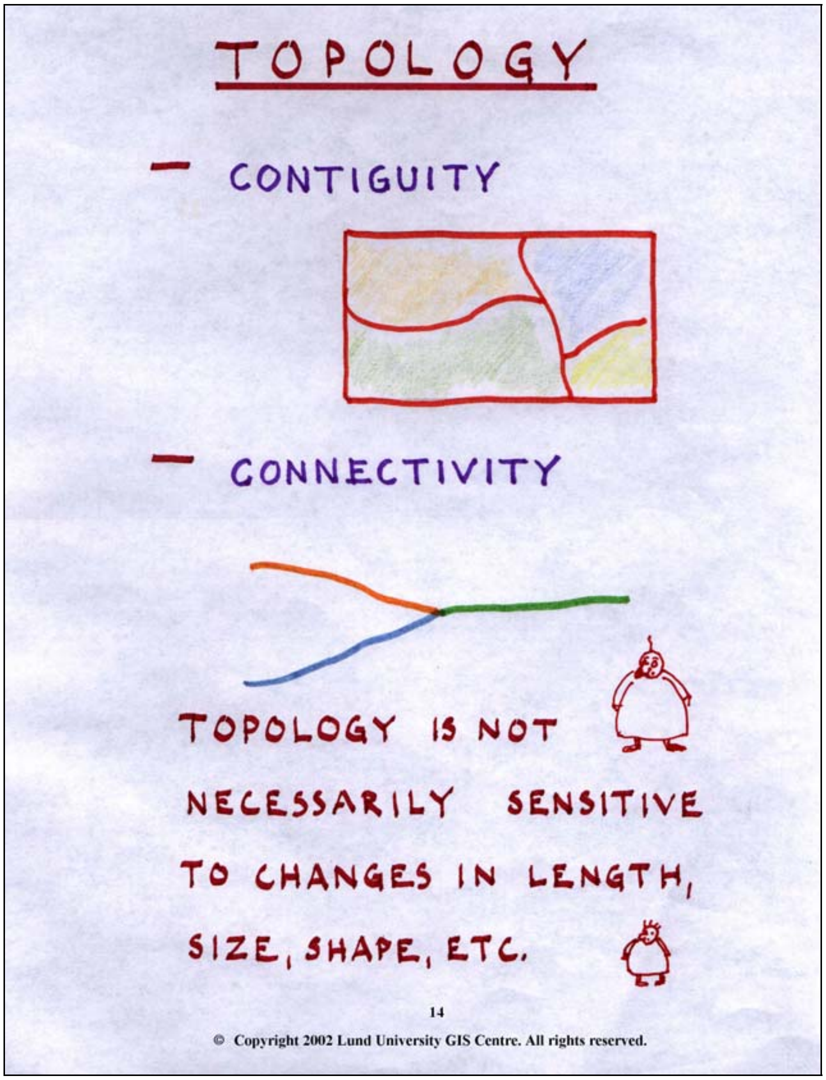
The general format for storage of vector data in a GIS database contains three components:

1. The object Id-number, which has to be unique (used to identify the object and to link attributes to it).
2. The n-term that states how many pairs of co-ordinates are used to build the object
3. The co-ordinate pairs as defined by the n-term.



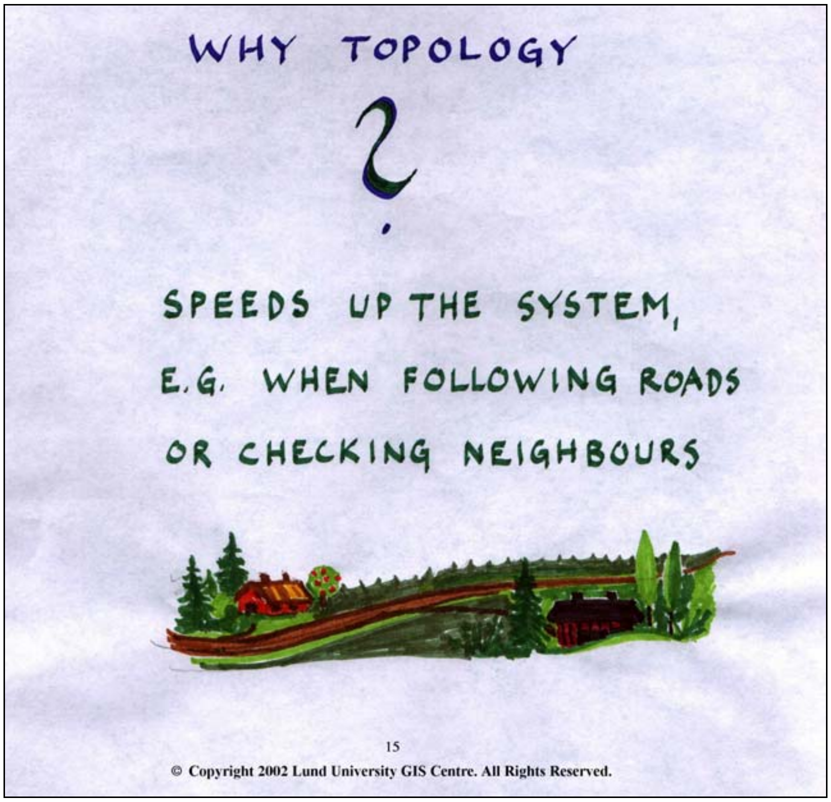
Slide 13

In vector GIS, there are several different ways of storing geometrical information ranging from the simpler, so called **Spaghetti** model, to the more complex Topological model. Both these models are common in GIS and the difference between them is in the level of structure and organisation of the data. In the simple spaghetti model, objects are stored more or less as described in the general vector data format (described on the last slide), while in the topological model, additional information about an objects relations to neighbouring objects is stored together with the co-ordinates. Both models can be used for all types of operations and analysis, but for complex operations the structured topological model will perform more efficiently than spaghetti models. **Topology** is a very common GIS term. It simply means that when topology is present, each object in the database has "knowledge" about its relation to adjacent objects.

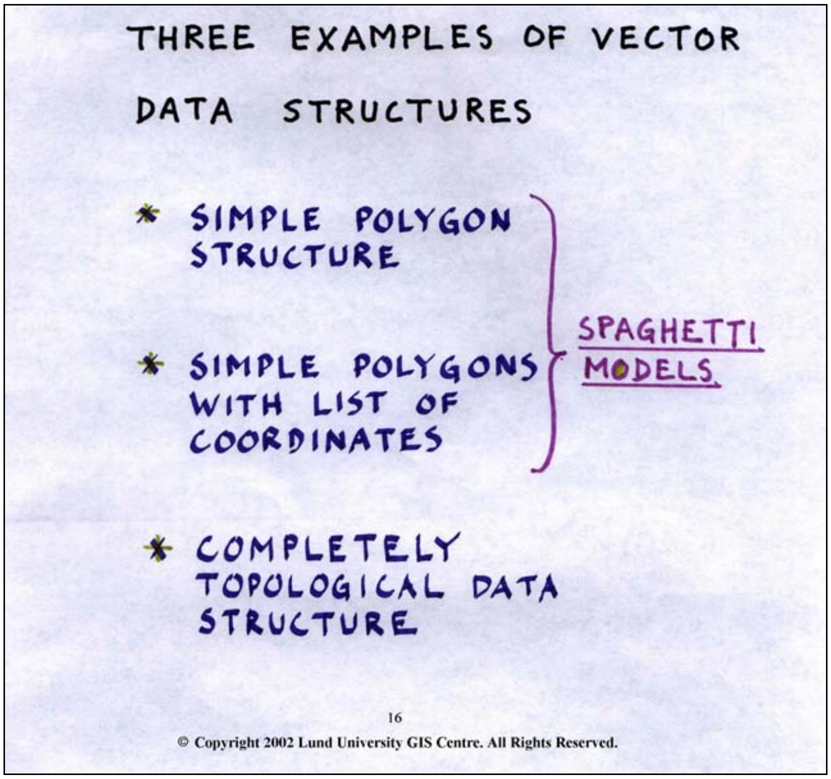


Slide 14

In complete topological models, polygons have **contiguity** (information about their neighbours) and line **connectivity** (information about which nodes are common to other lines). This information decreases processing time, e.g. in network analysis when searching the shortest route between two cities, since the computer doesn't have to check which lines are connected since this is already stored in a table in the database. It is important to note that topology is not necessarily sensitive to changes in length, size and shape of the objects. Changing the shape of a border between to polygon does not change the fact that they are still neighbours and changing the length of a line does not necessarily change the connections to other lines.



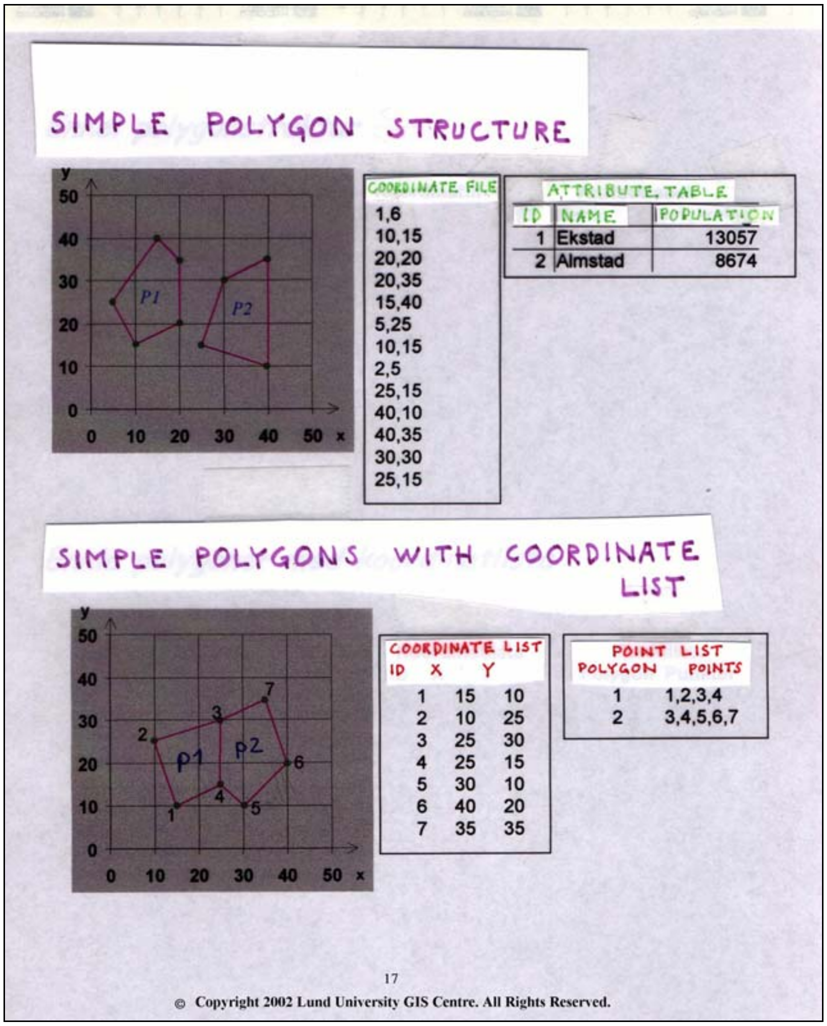
Slide 15
The main reasons for advocating the use of complete topology are that processing times are decreased and data storing is more efficient, conserving space on the hard disk.



Slide 16

The following slides will describe three examples of vector models, all of them in frequent use by different GIS software:

1.	Simple polygon structure	Spaghetti model
2.	Simple polygons with co-ordinate list	Spaghetti model
3.	Completely topological data structure	Topological model

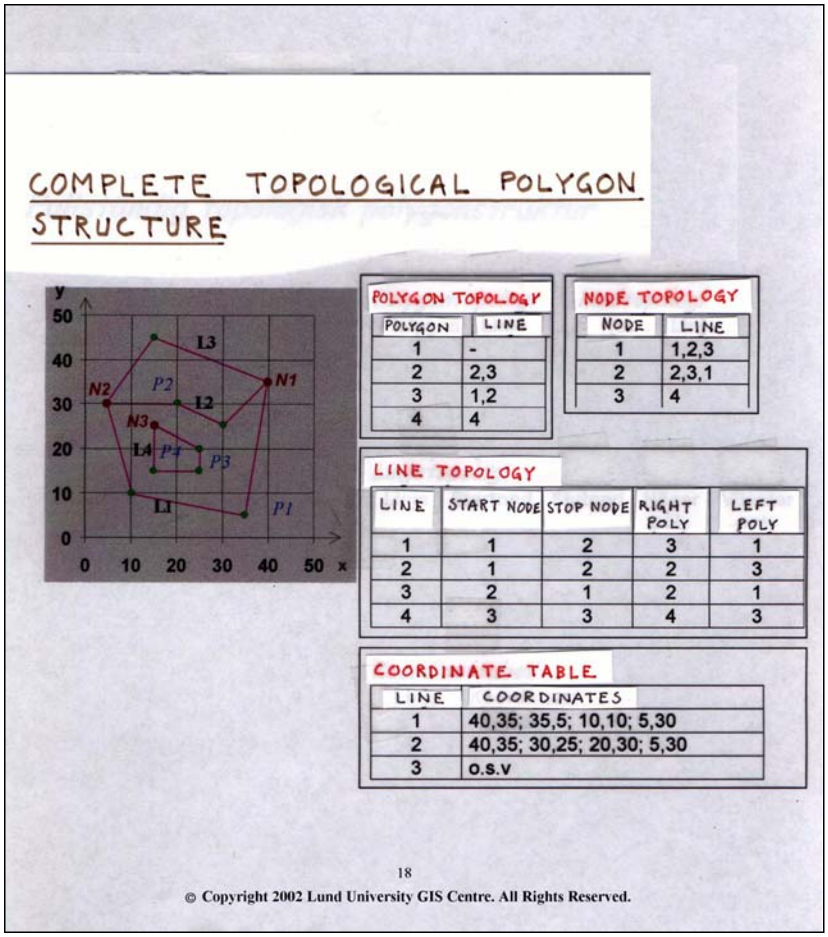


Slide 17

The **simple polygon structure** data storing format is more or less an exact copy of the general vector data storing model, i.e. Id-number, number of pairs of co-ordinates (*n*) and finally a list of the x and y co-ordinates. In the example, polygon Id = 1 has 6 pairs of co-ordinates (x=10, y=15; x=20, y=20; x=20, y=35…). The Id-number is used to link attributes to the geometrical objects, yielding an operational GIS database structure. Note that the number of breakpoints on the polygon boarder is n-1 (one less than the number of pairs of co-ordinates) due to the reason that both the start and stop node co-ordinates has to be stored. There are some disadvantages with this type of polygon structure:

1. No information about neighbouring polygons is stored.
2. All pairs of co-ordinates that are common to neighbouring polygons have to be stored in the co-ordinate list of all polygons.
3. When digitising maps, this means that common boarders have to be digitised once for every polygon, increasing the risk of miss-match along the common boarders.

In the **simple polygon structure with coordinate list structure**, each vertex has a unique Id-number a co-ordinate list, meaning that a polygon can then be defined by listing the vertex Id-numbers for the vertices on the boarder line of that particular polygon. In the example, polygon 1 is described by vertex 1,2,3,4 and polygon 2 by vertex 3,4,5,6,7. Storing the polygon structure in this way gives limited topological information since, by comparing the list of vertices, it is possible to identify polygons with vertices in common, which means that they must have a common boarder line, thus being neighbours. This way of storing the data structure is a bit more memory efficient than the simple polygon structure. As in the simple polygon structure, the polygon Id-number is used to link attributes to the geometry.



POLYGON	LINE
1	-
2	2,3
3	1,2
4	4

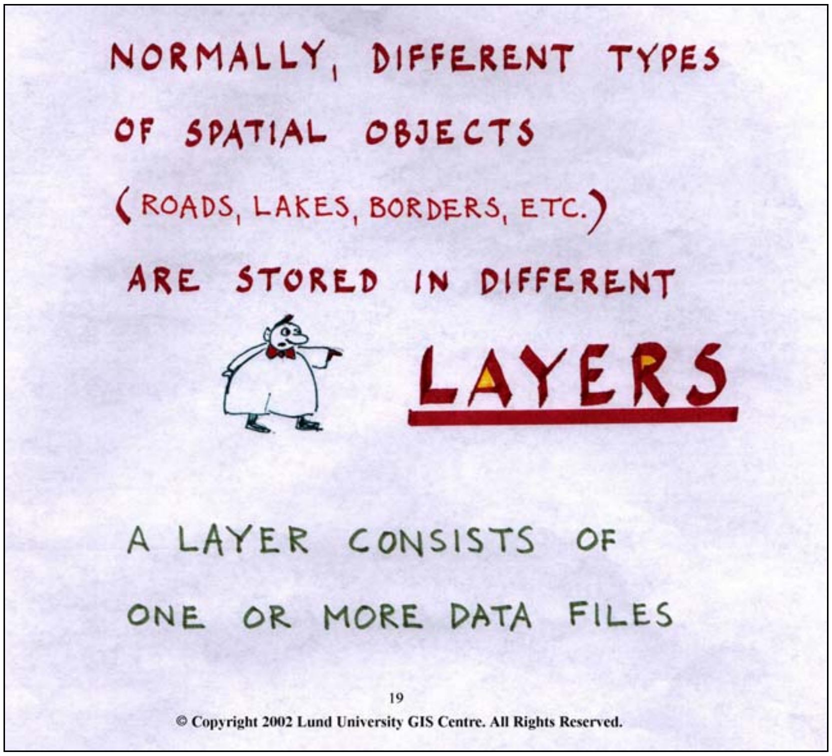
NODE	LINE
1	1,2,3
2	2,3,1
3	4

LINE	START NODE	STOP NODE	RIGHT POLY	LEFT POLY
1	1	2	3	1
2	1	2	2	3
3	2	1	2	1
4	3	3	4	3

LINE	COORDINATES
1	40,35; 35,5; 10,10; 5,30
2	40,35; 30,25; 20,30; 5,30
3	o.s.v

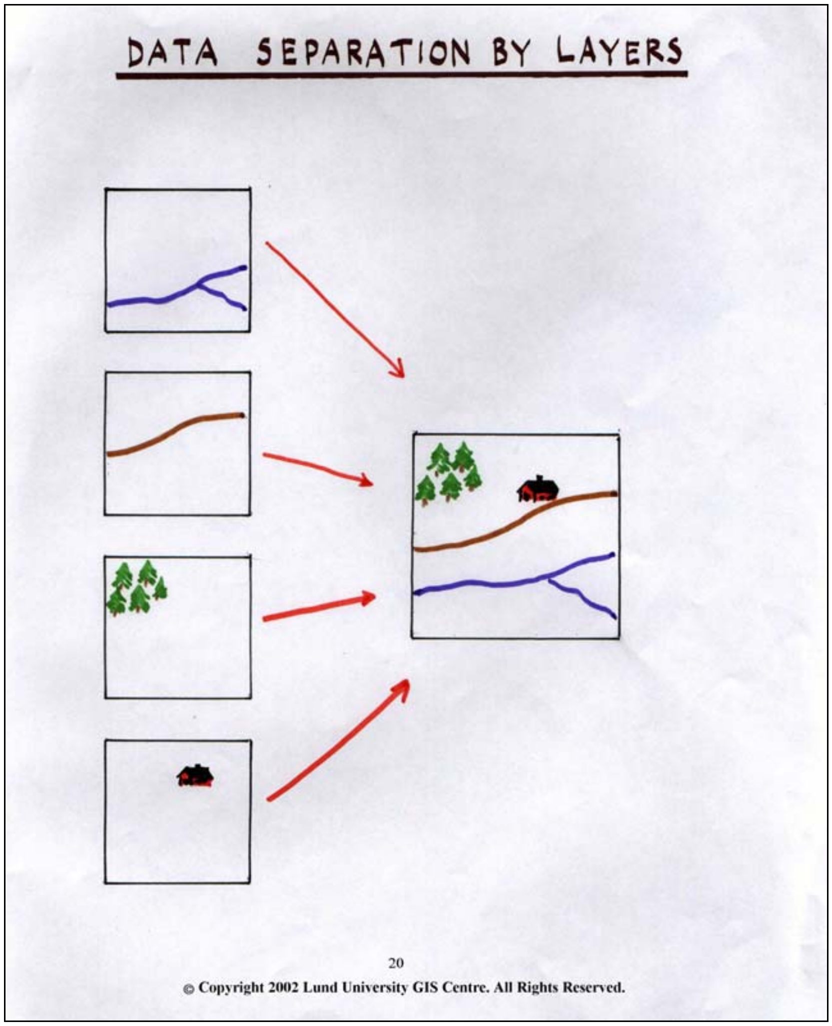
Slide 18

The **complete topological polygon structure** describes the interrelationship between all geometrical objects in the database. At first glance, this structure might look much more complicated and inefficient, but in fact the opposite is true. The complete topological structure is composed of four tables; the **polygon topology table, the node topology table, the line topology table and the co-ordinate table**. The co-ordinate table lists all the co-ordinate pairs for the lines in the database. It begins with the co-ordinate pair for the start node followed by a number of vertex co-ordinate pairs and ends with the stop node co-ordinates. In the example, line number 1 starts with a node at co-ordinates 40,35 followed by vertices at 35,5 and at 10,10 and ends at the coordinates 5,30. The line topology table lists the start and stop nodes for each line, which describes line connectivity and direction of each line. The table also contains information about the polygons at the right-hand and left-hand side of the line, based on the direction of the line (from start node to stop node). This is important since it gives information about what polygons that are neighbours. The node topology table also contains useful connectivity information since lines connect at nodes. In the example, lines 1, 2 and 3 are connected in node number 1. The polygon topology table store information about lines building the polygons. Instead of comparing all pairs of co-ordinates when searching for neighbouring polygons, the computer only has to search the polygon topology table to identify them. Note: Polygon number 1 is called the background polygon surrounding all other polygons and is used internally by the GIS software, with no application for a user. It is very important to be able to distinguish the different data structures from each other and to know the difference between spaghetti structure and topological structure since it affects how the software handles your data. In the example, a complete polygon topology has been described. If a database doesn't contain polygons, the polygon topology table is omitted from the structure.

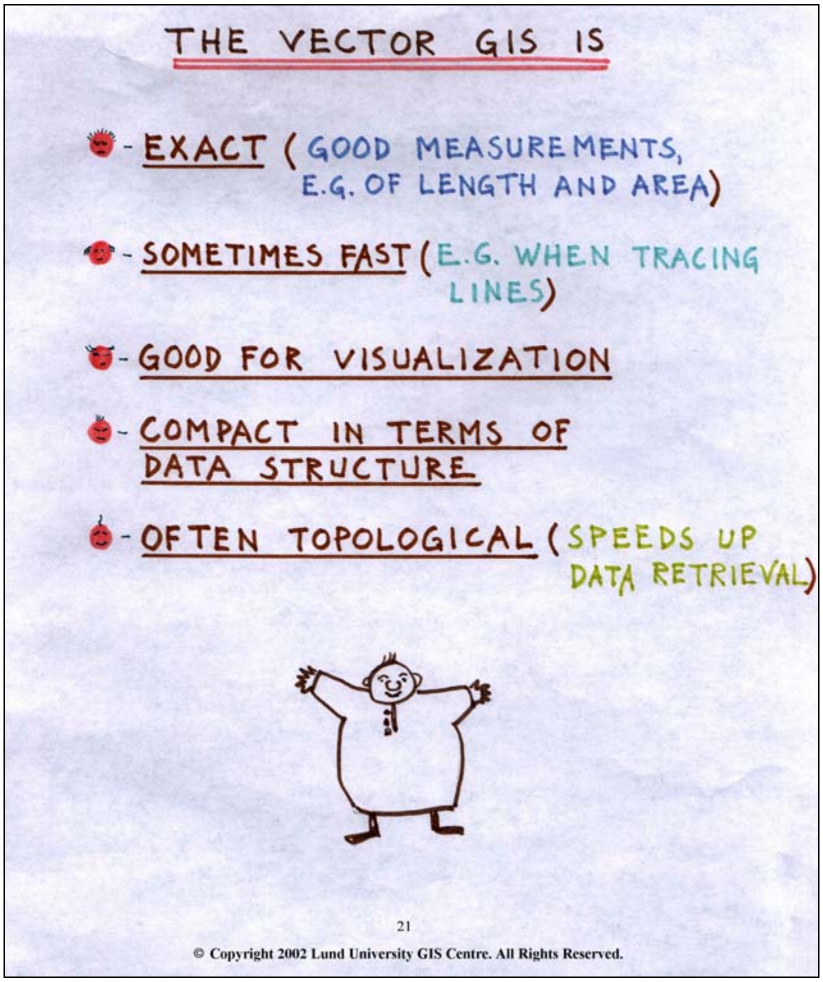


Slide 19

Different types of geometrical objects (points, lines and polygons) are normally stored in different map layers (data files) in order to handle them more efficiently. Different thematic information of the same character, e.g. roads, railroads and rivers (assuming they are all represented as lines) are frequently also stored in different databases, but may also be stored together in the same database and separated by attaching attributes describing their character.



Slide 20
Different thematic map layers, in this case consisting of a river layer, a road layer, a forest layer and a house layer are combined with the GIS software to present the desired information on a map. Provided that the cartographic settings are the same for all layers, it is simple and straightforward to combine any type of data on the same map.



Slide 21

To summarise, vector GIS is:

- Exact – The geometry of the vector model inherits the accuracy of the original data, as collected by field surveyors, GPS, photogrammetry, etc, since the structure of the model is based on storing the actual co-ordinates describing the location of different objects. This means that measurements done in vector database are as exact as the original data.

- Sometimes fast – Many operations are easy to perform on vector model data, e.g. network analysis (tracing lines and measuring distances along networks).

- Good for visualisation – For the same reason as it is exact, the vector model yields neat looking maps for all types of objects that are suitably represented by the model.

- Compact in terms of data structure – The only information that has to be stored are the nodes and vertices for lines and borderlines. Vector data structures demand much less computer storage space than raster data structure.

- Often topological – Most GIS software handles complete topological data structures, which speeds up the data retrieval and gives information about contiguity and connectivity.