

## UNIT - 3

# DATA INPUT AND TOPOLOGY

### 3.1. INTRODUCTION TO SCANNERS FOR RASTER DATA INPUT

Scanners are used to convert from analog maps or photographs to digital image data in raster format. Digital image data are usually integer-based with one byte gray scale (256 gray tones from 0 to 255) for black and white image and a set of three gray scales of red (R), green (G) and blue(B) for color image. The following four types of scanner are commonly used in GIS and remote sensing.

#### (a) Mechanical Scanner

It is called drum scanner since a map or an image placed on a drum is digitized mechanically with rotation of the drum and shift of the sensor as shown in Figure 3.1 (a). It is accurate but slow.

#### (b) Video Camera

Video camera with CRT (cathode ray tube) is often used to digitize a small part of map of firm. This is not very accurate but cheap. (See Figure 3.1 (b)).

#### (c) CCD Camera

Area CCD camera (called digital still camera) instead of video camera will be also convenient to acquire digital image data (see Figure 3.1 (c)). It is more stable and accurate than video camera.

#### (d) CCD Scanner

Flat bed type or roll feed type scanner with linear CCD (charge coupled device) is now commonly used to digitize analog maps in raster format; either in mono-tone or color mode (see Figure 3.1 (d)). It is accurate but expensive.

Comparison of Scanners			
Type	Resolution	Accuracy	Cost
Mechanical Scanner	very high (25~100 $\mu$ m)	Very high	high
Vedeo Camera	low (500x500 pixels)	low	cheap
CCD Camera	medium (500 x 500 ~ 4,000x4,000)	medium	cheap (low resolution) high (high resolution)
CCD Scanner	very high ( 300~600 dpi)	high	high

Table.3.1. Shows Comparison of scanners

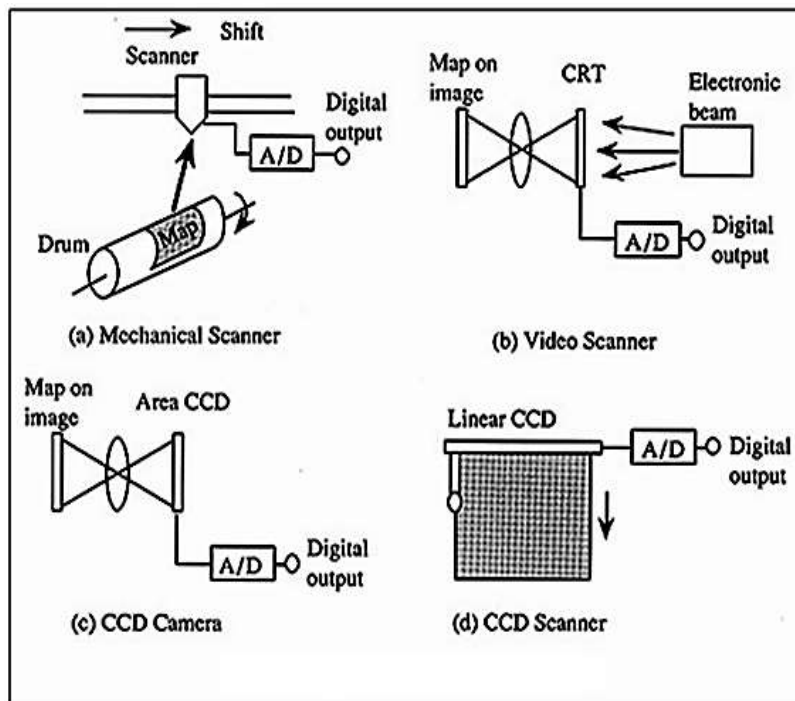


Fig.3.1. Scanners for Raster Data Input

### 3.1.1. Scanned data

- A scanner is used to convert analog source map or document into digital images by scanning successive lines across a map or document and recording the amount of light reflected from the data source.
- Documents such as building plans, CAD drawings, images and maps are scanned prior to vectorization.
- Scanning helps in reducing wear and tear; improves access and provides integrated storage.
- There are three different types of scanner that are widely used as shown in Figure 3.2.

### 3.1.2. Types of Scanner

- 1) Flat bed scanner
- 2) Rotating drum scanner
- 3) Large format feed scanner



*Fig.3.2. Types of Scanner*

Flat bed scanner is a PC peripheral which is small and comparatively inaccurate. The rotating drum scanners are accurate but they tend to be slow and expensive. Large format feed scanner are the most suitable type for inputting GIS data as they are cheap, quick and accurate.

## 3.2. RASTER DATA INPUT

### (A) Introduction

- Need to have tools to transform spatial data of various types into digital format
- data input is a major bottleneck in application of GIS technology
  - costs of input often consume 80% or more of project costs
  - data input is labor intensive, tedious, error-prone
  - there is a danger that construction of the database may become an end in itself and the project may not move on to analysis of the data collected
  - essential to find ways to reduce costs, maximize accuracy
- need to automate the input process as much as possible, but:
  - automated input often creates bigger editing problems later
  - source documents (maps) may often have to be redrafted to meet rigid quality requirements of automated input
- because of the costs involved, much research has gone into devising better input methods - however, few reductions in cost have been realized
- sharing of digital data is one way around the input bottleneck
  - more and more spatial data is becoming available in digital form
- data input to a GIS involves encoding both the locational and attribute data
- the locational data is encoded as coordinates on a particular Cartesian coordinate system

- source maps may have different projections, scales
- several stages of data transformation may be needed to bring all data to a common coordinate system
- attribute data is often obtained and stored in tables

### **3.2.1. Modes of data input**

- Keyboard entry for non-spatial attributes and occasionally locational data
- Manual locating devices
  - user directly manipulates a device whose location is recognized by the computer
  - e.g. digitizing
- Automated devices
  - automatically extract spatial data from maps and photography
  - e.g. scanning
- Conversion directly from other digital sources
- Voice input has been tried, particularly for controlling digitizer operations
  - Not very successful - machine needs to be recalibrated for each operator, after coffee breaks, etc.

### **3.2.2. Data Input Techniques**

Since the input of attribute data is usually quite simple, the discussion of data input techniques will be limited to spatial data only. There is no single method of entering the spatial data into a GIS. Rather, there are several, mutually compatible methods that can be used singly or in combination.

The choice of data input method is governed largely by the application, the available budget, and the type and the complexity of data being input.

There are at least four basic procedures for inputting spatial data into a GIS. These are:

- Manual digitizing;
- Automatic scanning;
- Entry of coordinates using coordinate geometry; and the
- Conversion of existing digital data.

#### ***Digitizing***

While considerable work has been done with newer technologies, the overwhelming majority of GIS spatial data entry is done by manual digitizing. A digitizer is an electronic device consisting of a table upon which the map or drawing is placed. The user traces the spatial features with a hand-held magnetic pen, often called a mouse or cursor. While tracing the features the coordinates of selected points, e.g. vertices, are sent to the computer and stored. All points that are recorded are registered against positional control points, usually the map corners that are keyed in

by the user at the beginning of the digitizing session. The coordinates are recorded in a user defined coordinate system or map projection. Latitude and longitude and UTM is most often used. The ability to adjust or transform data during digitizing from one projection to another is a desirable function of the GIS software. Numerous functional techniques exist to aid the operator in the digitizing process.

Digitizing can be done in a point mode, where single points are recorded one at a time, or in a stream mode, where a point is collected on regular intervals of time or distance, measured by an X and Y movement, e.g. every 3 metres. Digitizing can also be done blindly or with a graphics terminal. Blind digitizing infers that the graphic result is not immediately viewable to the person digitizing. Most systems display the digitized linework as it is being digitized on an accompanying graphics terminal.

Most GIS's use a spaghetti mode of digitizing. This allows the user to simply digitize lines by indicating a start point and an end point. Data can be captured in point or stream mode. However, some systems do allow the user to capture the data in an arc/node topological data structure. The arc/node data structure requires that the digitizer identify nodes.

Data capture in an arc/node approach helps to build a topologic data structure immediately. This lessens the amount of post processing required to clean and build the topological definitions. However, most often digitizing with an arc/node approach does not negate the requirement for editing and cleaning of the digitized line work before a complete topological structure can be obtained.

The building of topology is primarily a post-digitizing process that is commonly executed in batch mode after data has been cleaned. To date, only a few commercial vector GIS software offerings have successfully exhibited the capability to build topology interactively while the user digitizes.

Manual digitizing has many advantages. These include:

- Low capital cost, e.g. digitizing tables are cheap;
- Low cost of labour;
- Flexibility and adaptability to different data types and sources;
- Easily taught in a short amount of time - an easily mastered skill
- Generally the quality of data is high;
- Digitizing devices are very reliable and most often offer a greater precision than the data warrants; and
- Ability to easily register and update existing data.

For raster based GIS software data is still commonly digitized in a vector format and converted to a raster structure after the building of a clean topological structure. The procedure usually differs minimally from vector based software digitizing, other than some raster systems allow the user to define the resolution size of the grid-cell. Conversion to the raster structure may occur on-the-fly or afterwards as a separate conversion process.

### ***Automatic Scanning***

A variety of scanning devices exist for the automatic capture of spatial data. While several different technical approaches exist in scanning technology, all have the advantage of being able to capture spatial features from a map at a rapid rate of speed. However, as of yet, scanning has not proven to be a viable alternative for most GIS implementation. Scanners are generally expensive to acquire and operate. As well, most scanning devices have limitations with respect to the capture of selected features, e.g. text and symbol recognition. Experience has shown that most scanned data requires a substantial amount of manual editing to create a clean data layer. Given these basic constraints some other practical limitations of scanners should be identified. These include:

- Hard copy maps are often unable to be removed to where a scanning device is available, e.g. most companies or agencies cannot afford their own scanning device and therefore must send their maps to a private firm for scanning;
- Hard copy data may not be in a form that is viable for effective scanning, e.g. maps are of poor quality, or are in poor condition;
- Geographic features may be too few on a single map to make it practical, cost-justifiable, to scan;
- Often on busy maps a scanner may be unable to distinguish the features to be captured from the surrounding graphic information, e.g. dense contours with labels;
- With raster scanning there it is difficult to read unique labels (text) for a geographic feature effectively; and
- Scanning is much more expensive than manual digitizing, considering all the cost/performance issues.

Consensus within the GIS community indicates that scanners work best when the information on a map is kept very clean, very simple, and uncluttered with graphic symbology.

The sheer cost of scanning usually eliminates the possibility of using scanning methods for data capture in most GIS implementations. Large data capture shops and government agencies are those most likely to be using scanning technology.

Currently, general consensus is that the quality of data captured from scanning devices is not substantial enough to justify the cost of using scanning technology. However, major breakthroughs are being made in the field, with scanning techniques and with capabilities to automatically clean and prepare scanned data for topological encoding. These include a variety of line following and text recognition techniques. Users should be aware that this technology has great potential in the years to come, particularly for larger GIS installations.

### ***Coordinate Geometry***

A third technique for the input of spatial data involves the calculation and entry of coordinates using coordinate geometry (COGO) procedures. This involves entering, from survey data, the explicit measurement of features from some known monument. This input technique is obviously very costly and labour intensive. In fact, it is rarely used for natural resource applications in GIS. This method is useful for creating very precise cartographic definitions of

property, and accordingly is more appropriate for land records management at the cadastral or municipal scale.

### ***Conversion of Existing Digital Data***

A fourth technique that is becoming increasingly popular for data input is the conversion of existing digital data. A variety of spatial data, including digital maps, are openly available from a wide range of government and private sources. The most common digital data to be used in a GIS is data from CAD systems. A number of data conversion programs exist, mostly from GIS software vendors, to transform data from CAD formats to a raster or topological GIS data format. Several ad hoc standards for data exchange have been established in the market place. These are supplemented by a number of government distribution formats that have been developed. Given the wide variety of data formats that exist, most GIS vendors have developed and provide data exchange/conversion software to go from their format to those considered common in the market place.

Most GIS software vendors also provide an ASCII data exchange format specific to their product, and a programming subroutine library that will allow users to write their own data conversion routines to fulfill their own specific needs. As digital data becomes more readily available this capability becomes a necessity for any GIS. Data conversion from existing digital data is not a problem for most technical persons in the GIS field. However, for smaller GIS installations who have limited access to a GIS analyst this can be a major stumbling block in getting a GIS operational. Government agencies are usually a good source for technical information on data conversion requirements.

Some of the data formats common to the GIS marketplace are listed below. Please note that most formats are only utilized for graphic data. Attribute data is usually handled as ASCII text files. Vendor names are supplied where appropriate.

<b>IGDS - Interactive Graphics Design Software (Intergraph / Micro station)</b>	This binary format is a standard in the turnkey CAD market and has become a de facto standard in Canada's mapping industry. It is a proprietary format, however most GIS software vendors provide DGN translators.
<b>DLG - Digital Line Graph (US Geological Survey)</b>	This ASCII format is used by the USGS as a distribution standard and consequently is well utilized in the United States. It is not used very much in Canada even though most software vendors provide two way conversions to DLG.
<b>DXF - Drawing Exchange Format (Auto-cad)</b>	This ASCII format is used primarily to convert to/from the Auto-cad drawing format and is a standard in the engineering discipline. Most GIS software vendors provide a DXF translator.
<b>GENERATE - ARC/INFO Graphic Exchange Format</b>	A generic ASCII format for spatial data used by the ARC/INFO software to accommodate generic spatial data.
<b>EXPORT - ARC/INFO Export Format.</b>	An exchange format that includes both graphic and attribute data. This format is intended for transferring ARC/INFO data from one hardware platform, or site, to another. It is also often used for archiving.

	<p>ARC/INFO data. This is not a published data format, however some GIS and desktop mapping vendors provide translators. EXPORT format can come in either uncompressed, partially compressed, or fully compressed format</p>
--	--

A wide variety of other vendor specific data formats exist within the mapping and GIS industry. In particular, most GIS software vendors have their own proprietary formats. However, almost all provide data conversion to/from the above formats. As well, most GIS software vendors will develop data conversion programs dependant on specific requests by customers. Potential purchasers of commercial GIS packages should determine and clearly identify their data conversion needs, prior to purchase, to the software vendor.

### 3.3 RASTER DATA FILE FORMATS

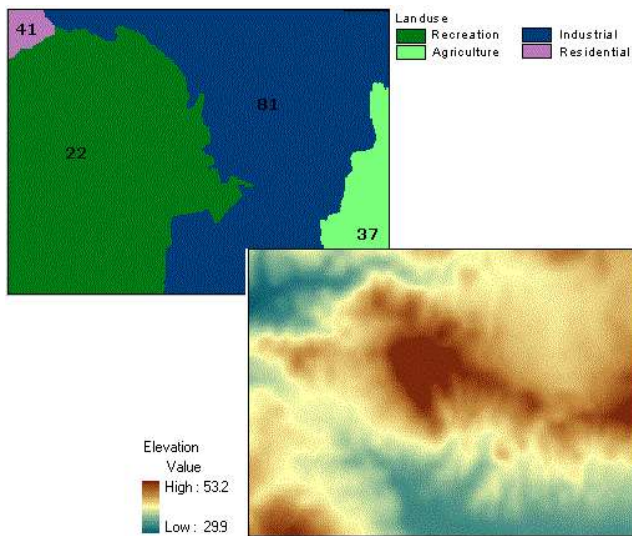
#### Raster data formats

Two common data formats based on the raster data model are grids and images.

#### 3.3.1. Grids

Grids are an ESRI file format used to store both discrete features such as buildings, roads, and parcels, and continuous phenomena such as elevation, temperature, and precipitation. Recall that the basic unit of the raster data model is the cell. Cells store information about what things are like at a particular location on the earth's surface. Depending on the type of data being stored, cell values can be either integers (whole numbers) or floating points (numbers with decimals). There are two types of grids: one store integers and the other stores floating points.

A discrete grid contains cells whose values are integers, often code numbers for a particular category. Cells can have the same value in a discrete grid. For example, in a discrete grid of land use, each land use type is coded by a different integer, but many cells may have the same code. Discrete grids have an attribute table that stores the cell values and their associated attributes.





Continuous grid is used to represent continuous phenomena; its cell values are floating points. Each cell in a continuous grid can have a different floating point value. For example, in a continuous grid representing elevation, one cell might store an elevation value of 564.3 meters, while the cell to the left might store an elevation value of 565.1 meters. Unlike discrete grids, continuous grids don't have an attribute table.

Discrete grids represent discrete features such as land use categories with integer values. Continuous grids represent continuous phenomena such as elevation with floating point values.

The attribute tables of discrete grids are INFO format, the same format in which coverage feature class attribute tables are stored. As with coverage attribute tables, the INFO table of a discrete grid is stored within an info folder, which is stored at the same level as the grid in a workspace folder. Again like coverages, there is one info folder for all the grids in a workspace folder. To avoid breaking or corrupting the connection between grid files and the info folder, always use ArcCatalog to move, copy, rename, and delete grids.

The Grids workspace folder contains two grids: soils and vegetation. The attribute tables for both grids are stored in the info folder. Auxiliary files called soils.aux and vegetation.aux link the grids and their attribute tables.

### 3.3.2. Images

The term "image" is a collective term for raster's whose cells, or pixels, store brightness values of reflected visible light or other types of electromagnetic radiation, such as emitted heat (infrared) or ultraviolet (UV). Aerial photos, satellite images, and scanned paper maps are examples of images commonly used in a GIS.

Images can be displayed as layers in a map or they can be used as attributes for vector features. For example, a real estate company might include photos of available houses as an attribute of a home's layer. To be displayed as a layer, however, images must be referenced to real-world locations.

For example, an aerial photo as it comes from the camera is just a static picture, like a picture of a house. There's no information about what part of the world the photo has captured, and the photo may contain distortion and scale variations caused by the angle of the camera. To display properly with other map layers, the aerial photo must be assigned a coordinate system and some of its pixels must be linked to known geographic coordinates.



Raster images, such as aerial photographs and scanned maps, can be referenced to real-world locations, then displayed as a layer in a GIS map.

There are many image file formats, which differ in the type of compression used to reduce the file size. Some of the image formats supported by ArcGIS software.

### **3.4. VECTOR DATA INPUT**

#### **Digitization:**

Digitizing is the process of interpreting and converting paper map or image data to vector digital data.

#### **3.4.1. Heads down digitization**

Digitizers are used to capture data from hardcopy maps shown in the (fig.3.3). Heads down digitization is done on a digitizing table using a magnetic pen known as Puck. The position of a cursor or puck is detected when passed over a table inlaid with a fine mesh of wires. The function of a digitizer is to input correctly the coordinates of the points and the lines. Digitization can be done in two modes:



*Fig.3.3. Heads down digitization*

- Point mode: In this mode, digitization is started by placing a point that marks the beginning of the feature to be digitized and after that more points are added to trace the particular feature (line or a polygon). The number of points to be added to trace the feature and the space interval between two consecutive points are decided by the operator.
- Stream mode: In stream digitizing, the cursor is placed at the beginning of the feature, a command is then sent to the computer to place the points at either equal or unequal intervals as per the position of the cursor moving over the image of the feature.

#### **3.4.2. Heads-up digitization**

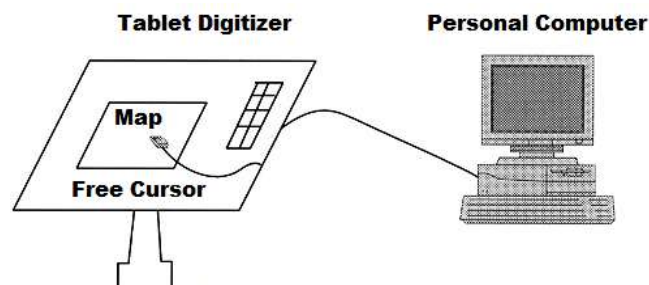
This method uses scanned copy of the map or image and digitization is done on the screen of the computer monitor (fig.3.4). The scanned map lays vertical which can be viewed without bending the head down and therefore is called as heads up digitization. Semi-automatic and automatic methods of digitizing requires post processing but saves lot of time and resources compared to manual method and is described in the following section.



*Fig.3.4. Screenshot of On-screen/Heads up digitization*

### 3.4.3. Digitizers for Vector Data Input

Tablet digitizers with a free cursor connected with a personal computer are the most common device for digitizing spatial features with the plan metric coordinates from analog maps.



*Fig.3.5. Digitizing Tablet*

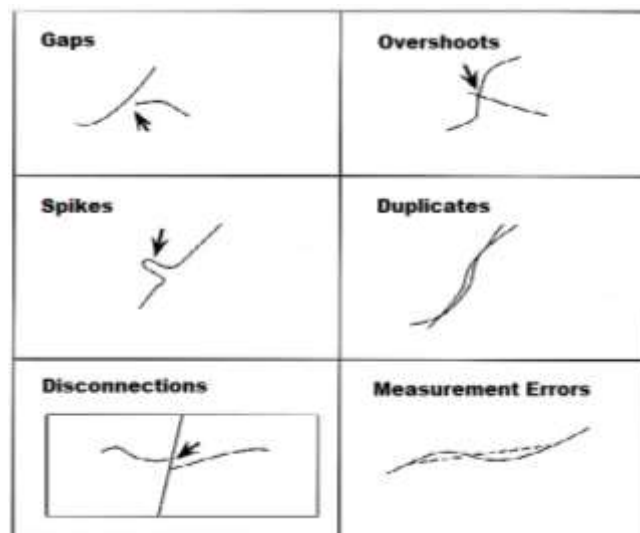
The analog map is placed on the surface of the digitizing tablet as shown in Figure 3.5. The size of digitizer usually ranges from A3 to A0 size.

*The digitizing operation is as follows.*

- Step 1:** a map is affixed to a digitizing table.
- Step 2:** control points or tics at four corners of this map sheet should be digitized by the digitizer and input to PC together with the map coordinates of the four corners.
- Step 3:** map contents are digitized according to the map layers and map code system in either point mode or stream mode at short time interval.
- Step 4:** editing errors such as small gaps at line junctions, overshoots, duplicates etc. should be made for a clean dataset without errors.
- Step 5:** conversion from digitizer coordinates to map coordinates to store in a spatial database.

### 3.4.4. Major problems of Map Digitization:

- The map will stretch or shrink day by day which makes the newly digitized points slightly off from the previous points.
- The map itself has errors discrepancies across neighboring map sheets will produce dis-connectivity.
- Operators will make a lot of errors and mistakes while digitizing as shown in Figure.3.6.



*Fig.3.6. Digitizing Typical Errors*

## 3.5. DIGITIZERS

- Digitizers are the most common device for extracting spatial information from maps and photographs
  - the map, photo, or other document is placed on the flat surface of the digitizing tablet

### 3.5.1. Hardware

- The position of an indicator as it is moved over the surface of the digitizing tablet is detected by the computer and interpreted as pairs of x,y coordinates
  - the indicator may be a pen-like stylus or a cursor (a small flat plate the size of a hockey puck with a cross-hair)
- frequently, there are control buttons on the cursor which permit control of the system without having to turn attention from the digitizing tablet to a computer terminal
- digitizing tablets can be purchased in sizes from 25x25 cm to 200x150 cm, at approximate costs from \$500 to \$5,000
- early digitizers (ca. 1965) were backlit glass tables

- a magnetic field generated by the cursor was tracked mechanically by an arm located behind the table
- the arm's motion was encoded, coordinates computed and sent to a host processor
- some early low-cost systems had mechanically linked cursors - the free-cursor digitizer was initially much more expensive
- the first solid-state systems used a spark generated by the cursor and detected by linear microphones
  - problems with errors generated by ambient noise
- contemporary tablets use a grid of wires embedded in the tablet to generate a magnetic field which is detected by the cursor
  - accuracies are typically better than 0.1 mm
  - this is better than the accuracy with which the average operator can position the cursor
  - functions for transforming coordinates are sometimes built into the tablet and used to process data before it is sent to the host

### ***The digitizing operation***

- The map is affixed to a digitizing table
- Three or more control points ("reference points", "tics", etc.) are digitized for each map sheet
  - these will be easily identified points (intersections of major streets, major peaks, points on coastline)
  - the coordinates of these points will be known in the coordinate system to be used in the final database, e.g. lat/long, State Plane Coordinates, military grid
  - the control points are used by the system to calculate the necessary mathematical transformations to convert all coordinates to the final system
  - the more control points, the better
- digitizing the map contents can be done in two different modes:
  - in point mode, the operator identifies the points to be captured explicitly by pressing a button
  - in stream mode points are captured at set time intervals (typically 10 per second) or on movement of the cursor by a fixed amount
- advantages and disadvantages:
  - in point mode the operator selects points subjectively
    - two point mode operators will not code a line in the same way
  - stream mode generates large numbers of points, many of which may be redundant
  - stream mode is more demanding on the user while point mode requires some judgement about how to represent the line
- most digitizing is currently done in point mode

### 3.5.2. Problems with digitizing maps

- arise since most maps were not drafted for the purpose of digitizing
  - paper maps are unstable: each time the map is removed from the digitizing table, the reference points must be re-entered when the map is affixed to the table again
  - if the map has stretched or shrunk in the interim, the newly digitized points will be slightly off in their location when compared to previously digitized points
  - errors occur on these maps, and these errors are entered into the GIS database as well
  - the level of error in the GIS database is directly related to the error level of the source maps
- maps are meant to display information, and do not always accurately record locational information
  - for example, when a railroad, stream and road all go through a narrow mountain pass, the pass may actually be depicted wider than its actual size to allow for the three symbols to be drafted in the pass
- discrepancies across map sheet boundaries can cause discrepancies in the total GIS database
  - e.g. roads or streams that do not meet exactly when two map sheets are placed next to each other
- user error causes overshoots, undershoots (gaps) and spikes at intersection of lines
- diagram
- user fatigue and boredom
- for a complete discussion on the manual digitizing process, see Marble et al, 1984

#### *Editing errors from digitizing*

- Some errors can be corrected automatically
  - small gaps at line junctions
  - overshoots and sudden spikes in lines
- Error rates depend on the complexity of the map, are high for small scale, complex maps
- These topics are explored in greater detail in later Units
  - Unit 13 looks at the process of editing digitized data
  - Units 45 and 46 discuss digitizing error

### 3.5.3. Digitizing costs

- A common rule of thumb in the industry is one digitized boundary per minute

- e.g. it would take  $99/60 = 1.65$  hours to digitize the boundaries of the 99 counties of Iowa

### 3.6 TOPOLOGY IN GIS

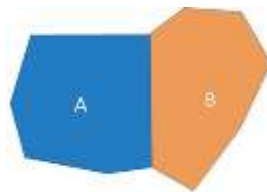
In geo-databases, a topology is a set of rules that defines how point, line, and polygon features share coincident geometry. Topology describes the means whereby lines, borders, and points meet up, intersect, and cross. This includes how street centrelines and census blocks share common geometry, and adjacent soil polygons share their common boundaries. Another example could be how two counties that have a common boundary between them will share an edge, creating a spatial relationship.

Common terms used when referring to topology include: dimensionality, adjacency, connectivity, and containment, with all but dimensional dealing directly with the spatial relationships of features.

**Dimensionality** - the distinction between point, line, area, and volume, which are said to have topological dimensions of 0, 1, 2, and 3 respectively.

#### 3.6.1. Adjacency

**Adjacency** including the touching of land parcels, counties, and nation-states (They share a common border).

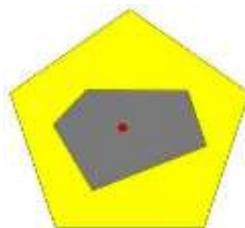


#### 3.6.2. Connectivity

**Connectivity** including junctions between streets, roads, railroads, and rivers (Very common topological error. See diagrams about "Overshoot" below).

#### 3.6.3. Containment

**Containment** when a point lies inside rather than outside an area.



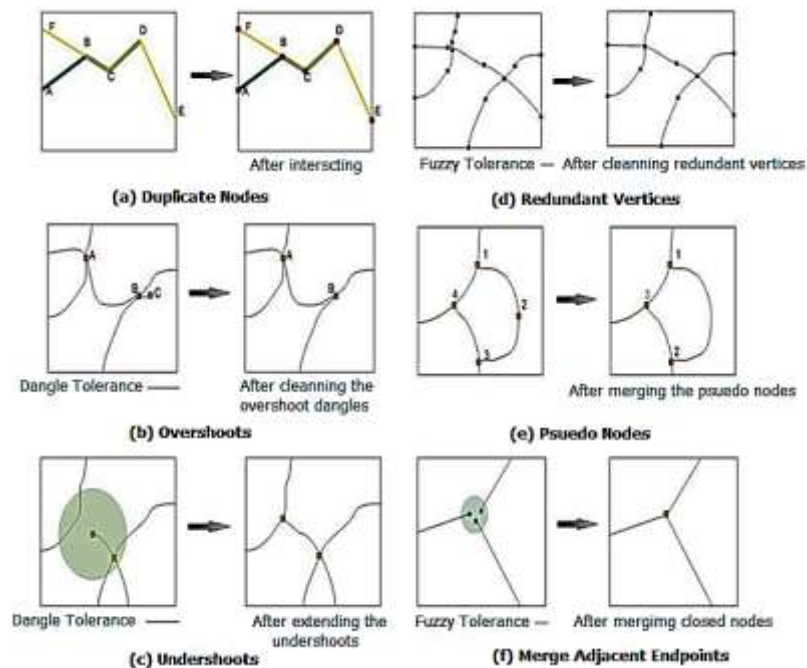
Topology defines and enforces data integrity rules (there should be no gaps between polygons). It supports topological relationship queries and navigation (navigating feature adjacency or connectivity), sophisticated editing tools, and allows feature construction from unstructured geometry (constructing polygons from lines).

Addressing topology is more than providing a data storage mechanism. In GIS, topology is maintained by using some of the following aspects:

- The geo-database includes a topological data model using an open storage format for simple features (i.e., feature classes of points, lines, and polygons), topology rules, and topologically integrated coordinates among features with shared geometry. The data model includes the ability to define the integrity rules and topological behaviour of the feature classes that participate in a topology.
- Most GIS programs include a set of tools for query, editing, validation, and error correction of topology.
- GIS software can navigate topological relationships, work with adjacency and connectivity, and assemble features from these elements. It can identify the polygons that share a specific common edge; list the edges that connect at a certain node; navigate along connected edges from the current location; add a new line and "burn" it into the topological graph; split lines at intersections; and create resulting edges, faces, and nodes.

### 3.6.4. Topological Consistency rules

Topological consistency describes the trustworthiness of the topological and logical relationships between the dataset segments (Joksic and Bajat, 2004). These relations typically involve spatial data inconsistencies such as incorrect line intersections, polygons not properly closed, duplicate lines or boundaries, or gaps in lines. It deals with the structural integrity of a given data set based on a formal framework for modelling of spatial data and relationships among objects. These types of errors must be corrected to avoid incomplete features and to ensure data integrity. Topological errors, which occur during digitizing and data exploration processes, are also known as semantic errors (Ubuda and Egenhofer, 1997). Topological errors exist due to violation of predefined topology rules. The most common topology errors in map data are shown in Figure including: - Duplicate Lines - Overshoots - Undershoots - Micro Segments - Pseudo Nodes - Merge Adjacent Endpoints - Self Intersection.



*Fig.3.7. Various Topological Errors*



### 3.6.5. Non Topological File Formats

#### *Raster formats*

- **ADRG** – National Geospatial-Intelligence Agency (NGA)'s ARC Digitized Raster Graphics
- **Binary file** – An unformatted file consisting of raster data written in one of several data types, where multiple band are stored in BSQ (band sequential), BIP (band interleaved by pixel) or BIL (band interleaved by line). Geo-referencing and other metadata are stored one or more sidecar files.
- **Digital raster graphic (DRG)** – digital scan of a paper USGS topographic map
- **ECRG** – National Geospatial-Intelligence Agency (NGA)'s Enhanced Compressed ARC Raster Graphics (Better resolution than CADRG and no color loss)
- **ECW** – Enhanced Compressed Wavelet (from ERDAS). A compressed wavelet format, often lossy.
- **Esri grid** – proprietary binary and meta-dataless ASCII raster formats used by Esri
- **GeoTIFF** – TIFF variant enriched with GIS relevant metadata
- **IMG** – ERDAS IMAGINE image file format
- **JPEG2000** – Open-source raster format. A compressed format, allows both lossy and lossless compression.
- **MrSID** – Multi-Resolution Seamless Image Database (by Lizardtech). A compressed wavelet format, allows both lossy and lossless compression.
- **netCDF-CF** – netCDF file format with CF metadata conventions for earth science data. Binary storage in open format with optional compression. Allows for direct web-access of subsets/aggregations of maps through OPeNDAP protocol.
- **RPF** – Raster Product Format, military file format specified in MIL-STD-2411
- **CADRG** – Compressed ADRG, developed by NGA, nominal compression of 55:1 over ADRG (type of Raster Product Format)
- **CIB** – Controlled Image Base, developed by NGA (type of Raster Product Format)

#### *Vector formats*

- **AutoCAD DXF** – contour elevation plots in AutoCAD DXF format (by Autodesk)
- **Cartesian coordinate system (XYZ)** – simple point cloud
- **Digital line graph (DLG)** – a USGS format for vector data
- **Esri TIN** - proprietary binary format for triangulated irregular network data used by Esri
- **Geography Markup Language (GML)** – XML based open standard (by OpenGIS) for GIS data exchange
- **GeoJSON** – a lightweight format based on JSON, used by many open source GIS packages

- **GeoMedia** – Intergraph's Microsoft Access based format for spatial vector storage
- **ISFC** – Intergraph's MicroStation based CAD solution attaching vector elements to a relational Microsoft Access database
- **Keyhole Markup Language (KML)** – XML based open standard (by OpenGIS) for GIS data exchange
- **MapInfo TAB format** – MapInfo's vector data format using TAB, DAT, ID and MAP files
- **National Transfer Format (NTF)** – National Transfer Format (mostly used by the UK Ordnance Survey)
- **Spatialite** – is a spatial extension to SQLite, providing vector geo-database functionality. It is similar to Post-GIS, Oracle Spatial, and SQL Server with spatial extensions
- **Shapefile** – a popular vector data GIS format, developed by Esri
- **Simple Features** – Open Geospatial Consortium specification for vector data
- **SOSI** – a spatial data format used for all public exchange of spatial data in Norway
- **Spatial Data File** – Autodesk's high-performance geo-database format, native to MapGuide
- **TIGER** – Topologically Integrated Geographic Encoding and Referencing
- **Vector Product Format (VPF)** – National Geospatial-Intelligence Agency (NGA)'s format of vectored data for large geographic databases

#### *Grid formats*

- **USGS DEM** – The USGS' Digital Elevation Model
- **GTOPO30** – Large complete Earth elevation model at 30 arc seconds, delivered in the USGS DEM format
- **DTED** – National Geospatial-Intelligence Agency (NGA)'s Digital Terrain Elevation Data, the military standard for elevation data
- **GeoTIFF** – TIFF variant enriched with GIS relevant metadata
- **SDTS** – The USGS' successor to DEM

### **3.7. LINKING THE ATTRIBUTE DATA TO THE SPATIAL DATA**

Before you can use your spatial data as a basis for exploring your attribute data, you must link the attribute data to the spatial data. One way to use the attribute data after you have linked it to the spatial data is by creating a theme to control the appearance of features in the spatial data. See Overview of SAS/GIS Software for more information.

In the layer bar, right-click the COUNTY layer name to open the pop-up menu for the COUNTY layer. Select Edit to open the GIS Layer window. In the definition for the COUNTY

layer, select Thematic. The GIS Attribute Data Sets window appears for you to define the link to the theme data set.

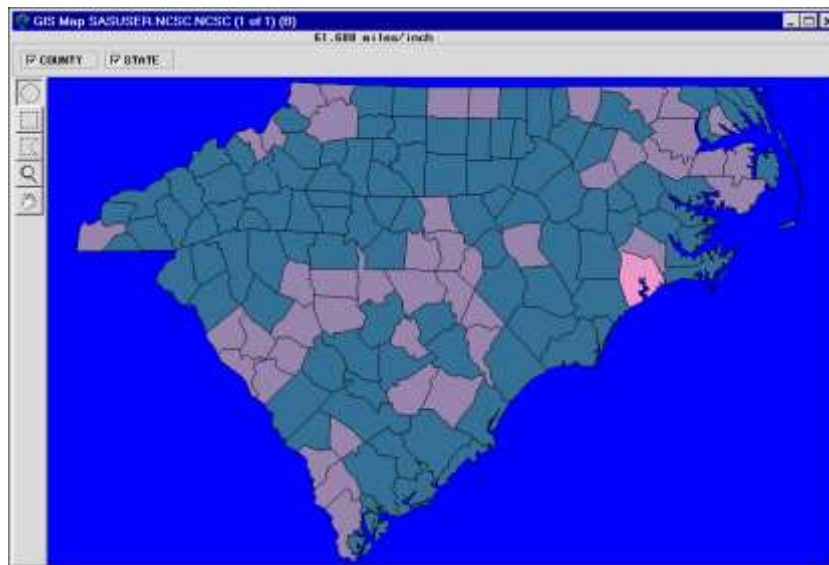
In the GIS Attribute Data Sets window, select New to define a new link. In the resulting select a Member window, select MAPS.USAAC. You must next specify the values that are common to both the attribute and spatial data, because the common values provide the connection between the spatial data and the attribute data.

The spatial database and the MAPS.USAAC data set share compatible state and county codes, so first select STATE in both the Data Set Vars and Compositeslists, and then select COUNTY in both lists. Select Save to save the link definition to the Links list. Finally, select Continue to close the GIS Attribute Data Setswindow.

After the GIS Attribute Data Sets window closes, the Var window automatically opens for you. Select which variable in the attribute data provides the theme data for your theme. Select the CHANGE variable to have the counties colored according to the level of change in the county population. Select OK to close the Var window.

The counties in the spatial data are colored according to the demographic values in the attribute data set, as shown in the following display.

### Linking the Attribute Data as a Theme



*Fig.3.8. Linking the Attribute Data as a Theme*

#### 3.7.1. Linking External Databases

The ArcGIS Maps Connect workflow supports external content from Microsoft SQL Server 2008 R2, 2012, 2012 R2, and 2014, including the SQL Server Express editions. The external content must contain data that can be geo-coded, such as an address, U.S. city, U.S. state, ZIP code, or world city. The external content must also contain a primary key column. Alternatively, the table can contain an existing SQL server spatial data type (geography or

geometry) column that is then converted by the Arc-GIS Maps Connect workflow for use in ArcGIS Maps for SharePoint.

If the external table has an existing spatial column that contains no data, the ArcGIS Maps Connect workflow populates the column based on other location information in the table (for example, address). If no spatial column exists, the ArcGIS Maps Connect workflow creates a geography spatial type column named EsriShape with a Spatial Reference Identifier (SRID) of 4326 (WGS 84). The EsriShape field supports all geometries including points, lines, and polygons. In all scenarios, the external content can be enriched with additional geographic data variables from ArcGIS.

### 3.7.2. Note

If the ArcGIS Maps Connect workflow fails, ensure the appropriate permissions for Microsoft SQL Server have been set. You can view the error messages in the SharePoint site workflow history to view exact details on the settings that need to be corrected.

When the ArcGIS Maps Connect workflow completes, the result is a regular SharePoint list, not an external list. That said, the fields created from the SQL Server database are of an external type, and edits made to these fields in SharePoint cannot be passed back to the database. SharePoint can only pass back the fields it has created, such as for the ArcGIS Maps Locate workflow and geoenrichment.

## 3.8. OPEN DATABASE CONNECTIVITY (ODBC)

In computing, Open Database Connectivity (ODBC) is a standard application programming interface (API) for accessing database management systems (DBMS). The designers of ODBC aimed to make it independent of database systems and operating systems. An application written using ODBC can be ported to other platforms, both on the client and server side, with few changes to the data access code.

ODBC accomplishes DBMS independence by using an ODBC driver as a translation layer between the application and the DBMS. The application uses ODBC functions through an ODBC driver manager with which it is linked, and the driver passes the query to the DBMS. An ODBC driver can be thought of as analogous to a printer driver or other driver, providing a standard set of functions for the application to use, and implementing DBMS-specific functionality. An application that can use ODBC is referred to as "ODBC-compliant". Any ODBC-compliant application can access any DBMS for which a driver is installed. Drivers exist for all major DBMSs, many other data sources like address book systems and Microsoft Excel, and even for text or comma-separated values (CSV) files.

ODBC was originally developed by Microsoft and Simba Technologies during the early 1990s, and became the basis for the Call Level Interface (CLI) standardized by SQL Access Group in the UNIX and mainframe field. ODBC retained several features that were removed as part of the CLI effort. Full ODBC was later ported back to those platforms, and became a de facto standard considerably better known than CLI. The CLI remains similar to ODBC, and applications can be ported from one platform to the other with few changes.

### 3.8.1. History of Before ODBC

The introduction of the mainframe-based relational database during the 1970s led to a proliferation of data access methods. Generally these systems operated together with a simple command processor that allowed users to type in English-like commands, and receive output. The best-known examples are SQL from IBM and QUEL from the Ingres project. These systems may or may not allow other applications to access the data directly, and those that did use a wide variety of methodologies. The introduction of SQL aimed to solve the problem of language standardization, although substantial differences in implementation remained.

Also, since the SQL language had only rudimentary programming features, users often wanted to use SQL within a program written in another language, say Fortran or C. This led to the concept of Embedded SQL, which allowed SQL code to be embedded within another language. For instance, a SQL statement like `SELECT * FROM city` could be inserted as text within C source code, and during compiling it would be converted into a custom format that directly called a function within a library that would pass the statement into the SQL system. Results returned from the statements would be interpreted back into C data formats like `char *` using similar library code.

There were several problems with the Embedded SQL approach. Like the different varieties of SQL, the Embedded SQLs that used them varied widely, not only from platform to platform, but even across languages on one platform – a system that allowed calls into IBM's DB2 would look very different from one that called into their own SQL/DS. Another key problem to the Embedded SQL concept was that the SQL code could only be changed in the program's source code, so that even small changes to the query required considerable programmer effort to modify. The SQL market referred to this as static SQL, versus dynamic SQL which could be changed at any time, like the command-line interfaces that shipped with almost all SQL systems, or a programming interface that left the SQL as plain text until it was called. Dynamic SQL systems became a major focus for SQL vendors during the 1980s.

Older mainframe databases, and the newer microcomputer based systems that were based on them, generally did not have a SQL-like command processor between the user and the database engine. Instead, the data was accessed directly by the program – a programming library in the case of large mainframe systems, or a command line interface or interactive forms system in the case of dBASE and similar applications. Data from dBASE could not generally be accessed directly by other programs running on the machine. Those programs may be given a way to access this data, often through libraries, but it would not work with any other database engine, or even different databases in the same engine. In effect, all such systems were static, which presented considerable problems.

### 3.8.2. Early efforts

By the mid-1980s the rapid improvement in micro-computers and especially the introduction of the graphical user interface and data-rich application programs like Lotus 1-2-3 led to an increasing interest in using personal computers as the client-side platform of choice in client-server computing. Under this model, large mainframes and minicomputers would be used primarily to serve up data over local area networks to microcomputers that would interpret, display and manipulate that data. For this model to work, a data access standard was a requirement – in the mainframe field it was highly likely that all of the computers in a shop were from one

vendor and clients were computer terminals talking directly to them, but in the micro field there was no such standardization and any client might access any server using any networking system.

By the late 1980s there were several efforts underway to provide an abstraction layer for this purpose. Some of these were mainframe related, designed to allow programs running on those machines to translate between the variety of SQL's and provide a single common interface which could then be called by other mainframe or microcomputer programs. These solutions included IBM's Distributed Relational Database Architecture (DRDA) and Apple Computer's Data Access Language. Much more common, however, were systems that ran entirely on microcomputers, including a complete protocol stack that included any required networking or file translation support.

One of the early examples of such a system was Lotus Development's DataLens, initially known as Blueprint. Blueprint, developed for 1-2-3, supported a variety of data sources, including SQL/DS, DB2, FOCUS and a variety of similar mainframe systems, as well as microcomputer systems like dBase and the early Microsoft/Ashton-Tate efforts that would eventually develop into Microsoft SQL Server. Unlike the later ODBC, Blueprint was a purely code-based system, lacking anything approximating a command language like SQL. Instead, programmers used data structures to store the query information, constructing a query by linking many of these structures together. Lotus referred to these compound structures as query trees.

Around the same time, an industry team including members from Sybase (Tom Haggin), Tandem Computers (Jim Gray & Rao Yendluri) and Microsoft (Kyle G) were working on a standardized dynamic SQL concept. Much of the system was based on Sybase's DB-Library system, with the Sybase-specific sections removed and several additions to support other platforms. DB-Library was aided by an industry-wide move from library systems that were tightly linked to a specific language, to library systems that were provided by the operating system and required the languages on that platform to conform to its standards. This meant that a single library could be used with (potentially) any programming language on a given platform.

The first draft of the Microsoft Data Access API was published in April 1989, about the same time as Lotus' announcement of Blueprint. In spite of Blueprint's great lead – it was running when MSDA was still a paper project – Lotus eventually joined the MSDA efforts as it became clear that SQL would become the de facto database standard. After considerable industry input, in the summer of 1989 the standard became SQL Connectivity (SQLC).

### 3.8.3. SAG and CLI

In 1988 several vendors, mostly from the UNIX and database communities, formed the SQL Access Group (SAG) in an effort to produce a single basic standard for the SQL language. At the first meeting there was considerable debate over whether or not the effort should work solely on the SQL language itself, or attempt a wider standardization which included a dynamic SQL language-embedding system as well, what they called a Call Level Interface (CLI). While attending the meeting with an early draft of what was then still known as MS Data Access, Kyle Geiger of Microsoft invited Jeff Balboni and Larry Barnes of Digital Equipment Corporation (DEC) to join the SQLC meetings as well. SQLC was a potential solution to the call for the CLI, which was being led by DEC.

The new SQLC "gang of four", MS, Tandem, DEC and Sybase, brought an updated version of SQLC to the next SAG meeting in June 1990. The SAG responded by opening the standard effort to any competing design, but of the many proposals, only Oracle Corp had a system that presented serious competition. In the end, SQLC won the votes and became the draft standard, but only after large portions of the API were removed – the standards document was trimmed from 120 pages to 50 during this time. It was also during this period that the name Call Level Interface was formally adopted. In 1995 SQL/CLI became part of the international SQL standard, ISO/IEC 9075-3.[8] The SAG itself was taken over by the X/Open group in 1996, and, over time, became part of The Open Group's Common Application Environment.

MS continued working with the original SQLC standard, retaining many of the advanced features that were removed from the CLI version. These included features like scrollable cursors, and metadata information queries. The commands in the API were split into groups; the Core group was identical to the CLI, the Level 1 extensions were commands that would be easy to implement in drivers, while Level 2 commands contained the more advanced features like cursors. A proposed standard was released in December 1991, and industry input was gathered and worked into the system through 1992, resulting in yet another name change to ODBC.

#### **3.8.4. JET and ODBC**

During this time, Microsoft was in the midst of developing their Jet database system. Jet combined three primary subsystems; an ISAM-based database engine (also named Jet, confusingly), a C-based interface allowing applications to access that data, and a selection of driver dynamic-link libraries (DLL) that allowed the same C interface to redirect input and output to other ISAM-based databases, like Paradox and xBase. Jet allowed using one set of calls to access common microcomputer databases in a fashion similar to Blueprint, by then renamed DataLens. However, Jet did not use SQL; like DataLens, the interface was in C and consisted of data structures and function calls.

The SAG standardization efforts presented an opportunity for Microsoft to adapt their Jet system to the new CLI standard. This would not only make Windows a premier platform for CLI development, but also allow users to use SQL to access both Jet and other databases as well. What was missing was the SQL parser that could convert those calls from their text form into the C-interface used in Jet. To solve this, MS partnered with PageAhead Software to use their existing query processor, SIMBA. SIMBA was used as a parser above Jet's C library, turning Jet into an SQL database. And because Jet could forward those C-based calls to other databases, this also allowed SIMBA to query other systems. Microsoft included drivers for Excel to turn its spreadsheet documents into SQL-accessible database tables.

#### **3.8.5. Release and continued development**

ODBC 1.0 was released in September 1992. At the time, there was little direct support for SQL databases (versus ISAM), and early drivers were noted for poor performance. Some of this was unavoidable due to the path that the calls took through the Jet-based stack; ODBC calls to SQL databases were first converted from Simba Technologies's SQL dialect to Jet's internal C-based format, then passed to a driver for conversion back into SQL calls for the database. Digital Equipment and Oracle both contracted Simba Technologies to develop drivers for their databases as well.

Circa 1993, OpenLink Software shipped one of the first independently developed third-party ODBC drivers, for the PROGRESS DBMS, and soon followed with their UDBC (a cross-platform API equivalent of ODBC and the SAG/CLI) SDK and associated drivers for PROGRESS, Sybase, Oracle, and other DBMS, for use on Unix-like OS (AIX, HP-UX, Solaris, Linux, etc.), VMS, Windows NT, OS/2, and other OS.

Meanwhile, the CLI standard effort dragged on, and it was not until March 1995 that the definitive version was finalized. By then, Microsoft had already granted Visigenic Software a source code license to develop ODBC on non-Windows platforms. Visigenic ported ODBC to a wide variety of Unix platforms, where ODBC quickly became the de facto standard. "Real" CLI is rare today. The two systems remain similar, and many applications can be ported from ODBC to CLI with few or no changes.

Over time, database vendors took over the driver interfaces and provided direct links to their products. Skipping the intermediate conversions to and from Jet or similar wrappers often resulted in higher performance. However, by then Microsoft had changed focus to their OLE DB concept (recently reinstated), which provided direct access to a wider variety of data sources from address books to text files. Several new systems followed which further turned their attention from ODBC, including ActiveX Data Objects (ADO) and ADO.net, which interacted more or less with ODBC over their lifetimes.

As Microsoft turned its attention away from working directly on ODBC, the UNIX field was increasingly embracing it. This was propelled by two changes within the market, the introduction of graphical user interfaces (GUIs) like GNOME that provided a need to access these sources in non-text form, and the emergence of open software database systems like PostgreSQL and MySQL, initially under Unix. The later adoption of ODBC by Apple for using the standard Unix-side iODBC package Mac OS X 10.2 (Jaguar) (which OpenLink Software had been independently providing for Mac OS X 10.0 and even Mac OS 9 since 2001) further cemented ODBC as the standard for cross-platform data access.

Sun Microsystems used the ODBC system as the basis for their own open standard, Java Database Connectivity (JDBC). In most ways, JDBC can be considered a version of ODBC for the programming language Java instead of C. JDBC-to-ODBC bridges allow Java-based programs to access data sources through ODBC drivers on platforms lacking a native JDBC driver, although these are now relatively rare. Inversely, ODBC-to-JDBC bridges allow C-based programs to access data sources through JDBC drivers on platforms or from databases lacking suitable ODBC drivers.

### 3.8.6. ODBC today

ODBC remains in wide use today, with drivers available for most platforms and most databases. It is not uncommon to find ODBC drivers for database engines that are meant to be embedded, like SQLite, as a way to allow existing tools to act as front-ends to these engines for testing and debugging.

However, the rise of thin client computing using HTML as an intermediate format has reduced the need for ODBC. Many web development platforms contain direct links to target databases – MySQL being very common. In these scenarios, there is no direct client-side access



nor multiple client software systems to support; everything goes through the programmer-supplied HTML application. The virtualization that ODBC offers is no longer a strong requirement, and development of ODBC is no longer as active as it once was.[citation needed]

### 3.9. GPS

Stands for "Global Positioning System." GPS is a satellite navigation system used to determine the ground position of an object. GPS technology was first used by the United States military in the 1960s and expanded into civilian use over the next few decades. Today, GPS receivers are included in many commercial products, such as automobiles, Smartphone, exercise watches, and GIS devices.

The GPS system includes 24 satellites deployed in space about 12,000 miles (19,300 kilometers) above the earth's surface. They orbit the earth once every 12 hours at an extremely fast pace of roughly 7,000 miles per hour (11,200 kilometers per hour). The satellites are evenly spread out so that four satellites are accessible via direct line-of-sight from anywhere on the globe.

Each GPS satellite broadcasts a message that includes the satellite's current position, orbit, and exact time. A GPS receiver combines the broadcasts from multiple satellites to calculate its exact position using a process called triangulation. Three satellites are required in order to determine a receiver's location, though a connection to four satellites is ideal since it provides greater accuracy.

In order for a GPS device to work correctly, it must first establish a connection to the required number of satellites. This process can take anywhere from a few seconds to a few minutes, depending on the strength of the receiver. For example, a car's GPS unit will typically establish a GPS connection faster than the receiver in a watch or Smartphone. Most GPS devices also use some type of location caching to speed up GPS detection. By memorizing its previous location, a GPS device can quickly determine what satellites will be available the next time it scans for a GPS signal.

#### 3.9.1. Uses of GPS

GPS has many uses, for example;

- **Clock synchronization:** The GPS time signals use highly accurate atomic clocks. This technology can be used for things like automatic updates of daylight saving times on cell phones
- Disaster relief and emergency services: Depend upon GPS for location
- Tracking a vehicle, person, pet or aircraft: Receivers provide continuous tracking and can provide an alert if the receiver leaves a set area. Pets can be chipped so they can be found if they become lost
- **Geotagging:** Applying location coordinates to digital objects such as photographs and other documents for purposes such as creating map overlays.
- **Bus tour commentary:** your location will determine what information is displayed about approaching points of interest
- **Bus stops:** to show how long the bus will take to arrive at a bus stop

- **Navigation:** eg Navman. The device uses voice activation to describe a preferred route based on the position of the receiver, the position of the destination and a street map
- **Personal Locator Beacons (PLB):** used to inform search and rescue authorities of your exact location in the event of an emergency
- **Recreation:** For example, geo-caching and way-marking
- **Surveying:** Surveyors use absolute locations to make maps and determine property boundaries
- **Tectonics:** enables fault motion measurement in earthquakes

Maps have come a long way since people first began drawings to show where they were. Modern maps are created using special software that combines lots of different sorts of information. This system of modern mapping is called GIS – Geographic Information Systems. GIS is used by organizations, such as city councils, that need access to data and need to be able to combine different data sets together. GIS gives people in these organisations graphical representations of data that allows them to:

- Analyze situations
- Write reports
- Track changes
- Make decisions
- Plan for the future, for example which parts of the high country have undergone tenure review

### ***GIS requires four things:***

**People:** people who use GIS are professionals who have been educated to use GIS and have made a career out of working with GIS

**Data:** geospatial information (where things are located) and the details of objects such as services, roads, buildings etc. are collected and entered into the GIS software

**Software:** GIS software analyses data and presents it in different combinations for the user

**Hardware:** includes hand held devices for collecting data and computers with GIS software

### **3.9.2. Basic structure of GPS**

Three-block configuration

GPS consists of the following three segments.

#### ***Space segment (GPS satellites)***

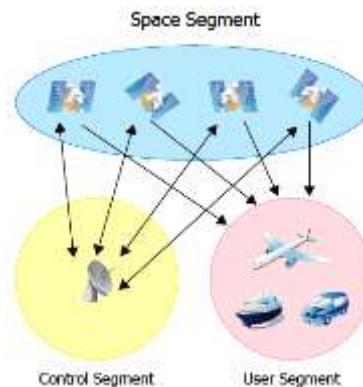
A number of GPS satellites are deployed on six orbits around the earth at the altitude of approximately 20,000 km (four GPS satellites per one orbit), and move around the earth at 12-hour-intervals.

←—————→  
**Control segment (Ground control stations)**

Ground control stations play roles of monitoring, controlling and maintaining satellite orbit to make sure that the deviation of the satellites from the orbit as well as GPS timing are within the tolerance level.

**User segment (GPS receivers)**

User segment (GPS receivers)

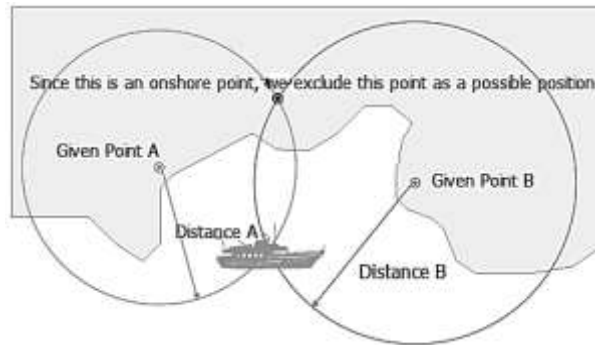


**Fig.3.9. Three elements of GPS**

### 3.9.3. GPS positioning

Firstly, the signal of time is sent from a GPS satellite at a given point. Subsequently, the time difference between GPS time and the point of time clock which GPS receiver receives the time signal will be calculated to generate the distance from the receiver to the satellite. The same process will be done with three other available satellites. It is possible to calculate the position of the GPS receiver from distance from the GPS receiver to three satellites. However, the position generated by means of this method is not accurate, for there is an error in calculated distance between satellites and a GPS receiver, which arises from a time error on the clock incorporated into a GPS receiver. For a satellite, an atomic clock is incorporated to generate on-the-spot time information, but the time generated by clocks incorporated into GPS receivers is not as precise as the time generated by atomic clocks on satellites. Here, the fourth satellite comes to play its role: the distance from the fourth satellite to the receiver can be used to compute the position in relations to the position data generated by distance between three satellites and the receiver, hence reducing the margin of error in position accuracy.

The Fig--- below illustrates an example of positioning by two dimensions (position acquisition by using two given points). We can compute where we are at by calculating distance from two given points, and the GPS is the system that can be illustrated by multiplying given points and replacing them with GPS satellites on this figure.



*Fig.3.10. position acquisition by using two given points*

### 3.9.4. GPS signals

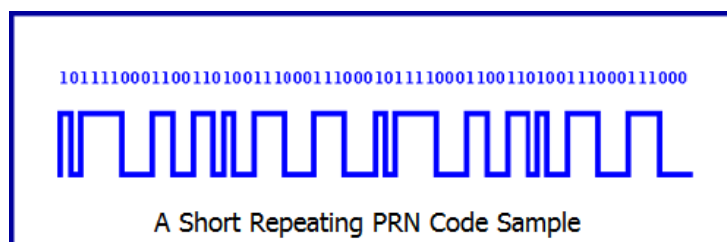
GPS satellites broadcast beams in two carrier frequencies; L1 (1,575.42 MHz) and L2 (1,227.60 MHz). Beams that can be accessible to the general public are encoded in C/A (Coarse/Acquisition) code, and the beams that can be used only by the US military force are encoded in P (Precise) code. C/A code consists of identification codes of each satellite and is broadcast together with navigation messages. The data of the orbit of each satellite is called the ephemeris\*, and the data of orbit of all satellite is called the almanac\*\*. The navigation messages are broadcast at a rate of 50 bits per second. Utilizing this collection of data, GPS receiver calculates distance between satellites and the receiver in order to generate position data. In the Fig 1-4, the details of C/A code is described, and in the Fig 1-5, navigation messages are described.

*\*The ephemeris provides the precise orbit for the satellite itself, which can be used to generate precise location of the satellite, necessary information for calculating position information. It is the indigenous data that is used only by each of the GPS satellites with specific identification number.*

*\*\*The almanac can be regarded as simplified ephemeris data and contains coarse orbit and status information for all satellites in the network. It is used to locate available satellites in order a GPS receiver to generate current position and time. It takes 12.5 minutes to receive all the almanac data.*

### 3.9.5. C/A code:

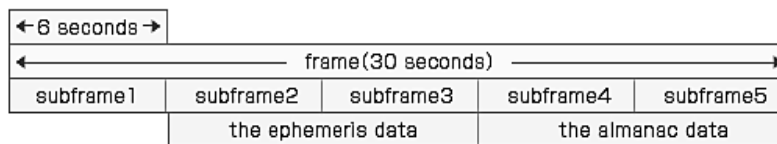
L1 signal from the GPS satellites is phase-modulated in C/A code, which is the pseudorandom code. The pseudorandom code is also called pseudorandom noise code, which is known as a Gold code. As the Fig. 1-4 illustrates, C/A code is a sequence of digital signals “1” and “0”. In GPS, 1,023 consecutive patterns comprise a sequence, and subsequently, this sequence will continually repeat one after another.



*Fig.3.11. C/A code used identify the GPS*

### 3.9.6. Navigation message

Navigation message consists of 25 frames, each of which includes 5 sub-frames of 300 bits each. The data length of 1 bit is 20 ms, and thus, the length of each sub-frame is 6 seconds, and each frame is a grouping of 1,500 bits of information with the frame length of 30 seconds. Since navigation message consists of 25 frames, this would add up to the message length of 12.5 minutes (30 seconds x 25=12.5 minutes). The GPS receiver requires 12.5 minutes to receive all the necessary set of data, necessary condition for positioning, when initial power activation takes place. The GPS receiver is capable of storing this set of data gained in the past internal backup battery, and it reads out the set of data when power reactivation takes place, hence instantaneously starting to receive GPS position.



**Fig.3.12. Navigation message**

#### Positioning accuracy

Factors that trigger GPS position errors

#### Ionosphere

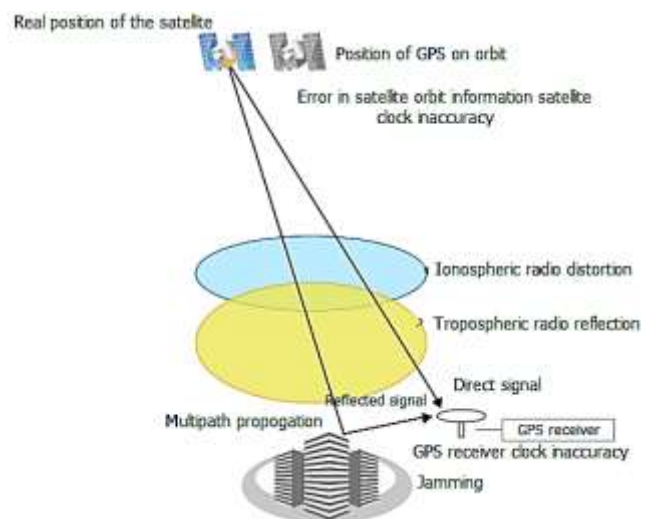
The ionosphere is a portion of the upper atmosphere, between the thermosphere and the exosphere. When GPS signals pass through this layer, the propagation velocity of the GPS signal goes slower, hence causing propagation error.

#### Troposphere

The troposphere is the lowest portion of Earth's atmosphere. Radio reflections caused by dry atmosphere and water vapor within provoke GPS position error.

#### Multipath propagation

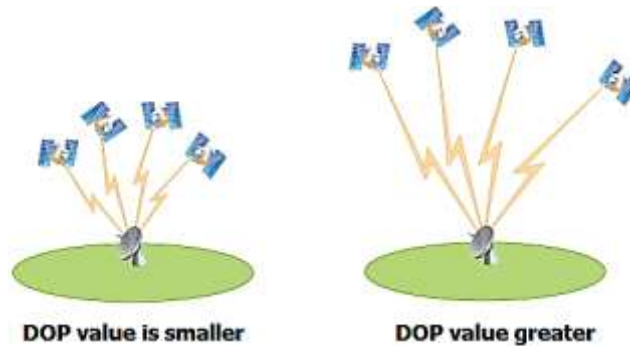
GPS signal is not immune to reflection when it hits on the ground, structures and many others. This phenomenon is called multipath propagation, one of the causes of GPS position errors.



**Fig.3.13. GPS position error**

### 3.9.7. DOP (Dilution of Precision)

DOP is a value that shows the degree of degradation of the GPS positioning accuracy. The smaller the value is, the higher the positioning accuracy is. This value depends upon the positions of the GPS satellites tracked for positioning. If the tracked satellites spread evenly over the earth, the positioning accuracy would become higher, and if the positions of tracked satellites are disproportionate, the positioning accuracy would become lower.

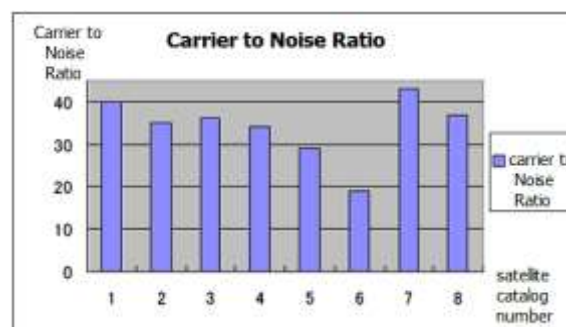


*Fig.3.14. DOP value is smaller*

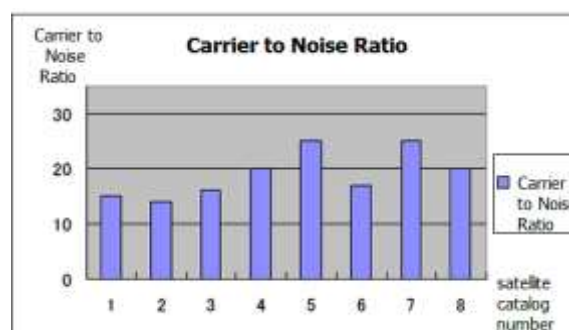
*Fig.3.15. DOP value greater*

### 3.9.8. Signal strength

State of reception of GPS depends upon the strength of GPS signals. The greater the signal strength is, the more stable the reception status is. Whereas the reception status would become unstable when the GPS signal became weaker, due to obstacles or noise sources in the vicinity of a GPS receiver.



*Fig.3.16. Example of stable GPS positioning*

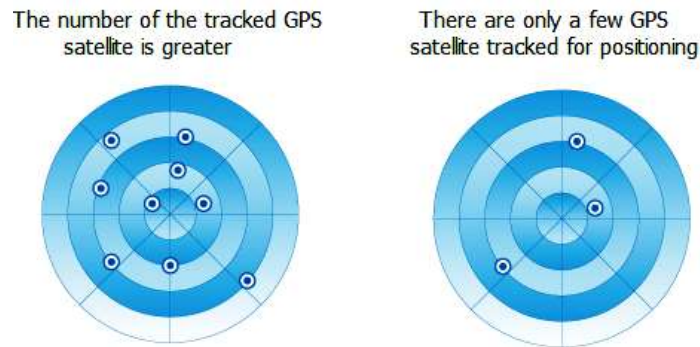


*Fig.3.17. Example of unstable GPS positioning*

### 3.9.9. Number of satellites tracked for positioning

State of reception of GPS depends upon the number of satellites tracked for positioning.

If the number of the tracked satellites is great, GPS positioning becomes greater, but if there were a fewer satellites tracked for positioning, it would be difficult to generate GPS position. The Fig. 1-11 illustrates the occasion where the GPS receiver tracks a greater number of satellites for positioning. The Fig. 1-12 illustrates the occasion where the GPS receiver tracks only a few number of satellites for positioning.



*Fig.3.18. GPS satellite are portrayed as blue circles above*

### 3.10. CONCEPT OF GPS BASED MAPPING

GPS consists of a constellation of radio navigation satellite and a ground control segment. It manages satellite operation and users with specialized receivers who use the satellite data to satisfy a broad range of positioning requirements.

*In brief, following are the key features of GPS:-*

- The basis of GPS is „triangulation“ more precisely trilateration from satellites
- A GPS receiver measures distance using the travel time of radio signals.
- To measure travel time GPS needs very accurate timing that is achieved with some techniques.
- Along with distance, one needs to know exactly where the satellites are in space.
- Finally one must correct for any delays, the signal experience as it travels through the atmosphere.

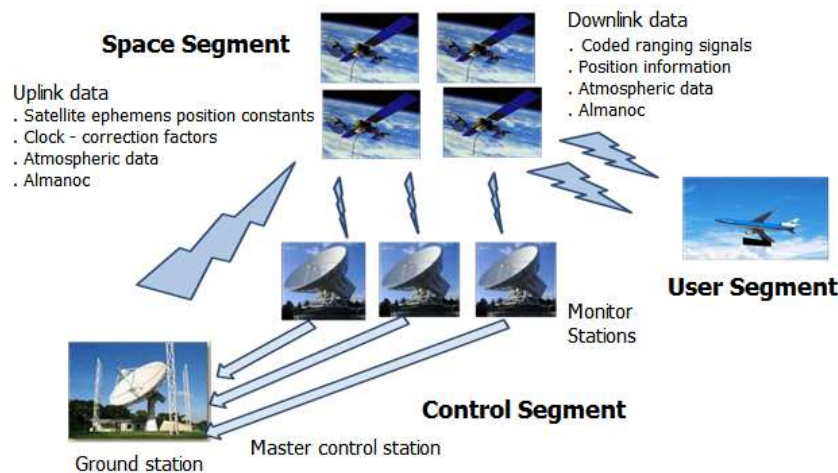
The whole idea behind GPS is to use satellites in space as reference points for location here on earth. By very accurately measuring the distances from at least three satellites, we can „triangulate“ our position anywhere on the earth by resection method.

#### 3.10.1. GPS Elements

GPS has 3 parts: the space segment, the user segment, and the control segment, Figure-1.2 illustrates the same. The space segment consists of a constellation of 24 satellites, each in its own orbit, which is 11,000 nautical miles above the Earth. The user segment consists of receivers,



which can be held in hand or mount in the vehicle. The control segment consists of ground stations (six of them, located around the world) that make sure the satellites are working properly. More details on each of these elements can be referred from any standard book or online literature on GPS.



*Fig.3.19. GPS segments*

### 3.10.2. GPS Satellite Navigation System

GPS is funded and controlled by the U. S. Department of Defense (DOD). While there are many thousands of civil users of GPS worldwide, the system was designed for and is operated by the U. S. military. It provides specially coded satellite signals that can be processed in a GPS receiver, enabling the receiver to compute position, velocity and time. Four GPS satellite signals are used to compute positions in three dimensions and the time offset in the receiver clock.

### 3.10.3. GPS Positioning Techniques

GPS positioning techniques may be categorized as being predominantly based on code or carrier measurements. Code techniques are generally simple and produce low accuracy, while carrier techniques are more complex and produce higher accuracy. There exist a variety of positioning methods for both code and carrier measurements. The suitability of each for a specific application is dependent on the desired accuracy, logistical constraints and costs. Many variables affect this accuracy, such as the baseline lengths, ionospheric conditions, magnitude of selective availability, receiver types used, and processing strategies adopted.

### 3.10.4. Differential GPS (DGPS)

The technique used to augment GPS is known as “differential”. The basic idea is to locate one or more reference GPS receivers at known locations in users’ vicinities and calibrate ranging errors as they occur. These errors are transmitted to the users in near real time. The errors are highly correlated across tens of kilometers and across many minutes. Use of such corrections can greatly improve the accuracy and integrity.

To increase the accuracy of positioning, Differential-GPS (D-GPS) was introduced. The idea is as follows: a reference station is located at a known and accurately surveyed point. The



GPS reference station determines its GPS position using four or more satellites. Given that the position of the GPS reference station is exactly known, the deviation of the measured position to the actual position and more importantly the measured pseudo range to each of the individual satellites can be calculated. The differences are either transmitted immediately by radio or used afterwards for correction after carrying out the measurements. The man made error like “Selective Availability” can be corrected using this.

### **3.10.5. GPS applications in Transportation**

Due to the high accuracy, usability, ease and economy of operations in all weather, offered by GPS, it has found numerous applications in many fields ranging from accuracy level of mm for the high precision geodesy to several meters for navigational positioning. Some of the applications in urban and transportation field are:

- Establishment of ground control points for imageries / map registration,
- Determination of a precise geo ID using GPS data,
- Survey control for topographical and cadastral surveys,
- Air, road, rail, and marine navigation,
- Intelligent traffic management system,
- Vehicle tracking system etc.