Lesson 4: Vector Data Structure

## INTRODUCTION

This lesson is on the concept of vector data structures. Cartesian coordinates are covered in this lesson and examples of lines, points, polygons are explained. The lesson explains two vector data computer models, Spaghetti and Topological and further explains topology. The relationships between topological and non-topological features are exhibited and explained.

## LESSON OBJECTIVES

By the end of this lesson, you will be able to:

1. Define key terms relating to the structure of a vector data, vector data models, and topology.

2. Describe the nature of geometry and its relationship to both topological and non-topological features.

## LEARNING SEQUENCE

|  |  |
| --- | --- |
|  | |
| Required Reading | Read the following:  Vector Data Structure   * Vector Data * Vector Data Computer Models * Non-Spatial Attribute Relationships * Topology * Relationships * Topology Errors |
| Assignments | Complete the following assignments:   * Lab 4: Vector Data Structure * Quiz 4: Vector data structure |

## INSTRUCTION

## Vector Data

### Review of Vector Data

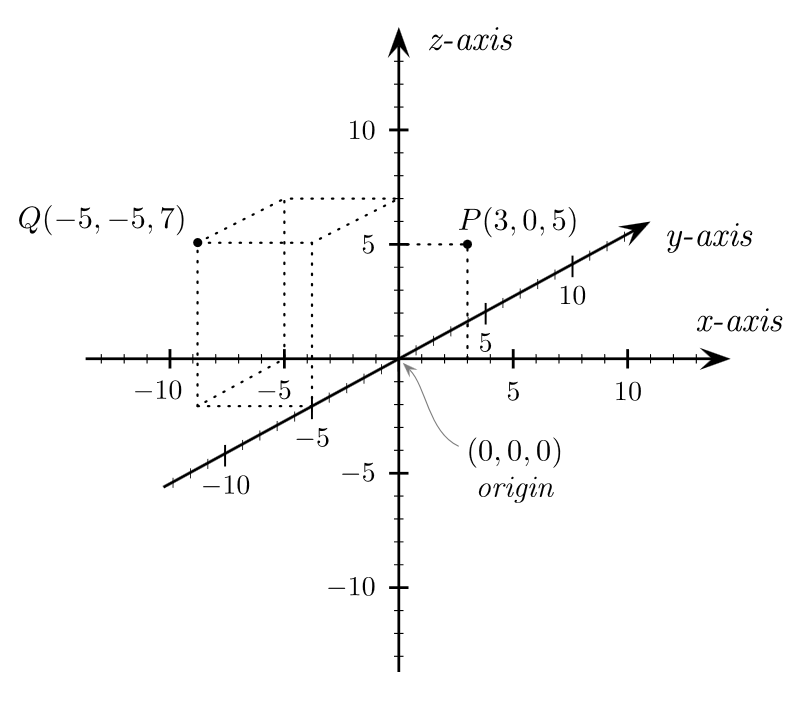
To understand Vector Data Structures it will help to do a quick review of vector data. Vector data is the representation of geographic objects with basic elements of point, line, and areas (also known as polygons). Each point is based on Cartesian coordinates, and the ordered accumulation of the coordinates result in a point, line, or area.

### Cartesian Coordinates

A Cartesian coordinate specifies a location uniquely on a plane by a pair of numerical coordinates. Cartesian coordinate systems can be both two-dimensional and three-dimensional.

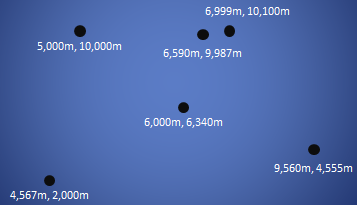
#### 3-D Cartesian coordinate system

This image is an example of a three-dimensional Cartesian coordinate system. The three axes are X axis, Y-axis, and Z-axis. The X and Y axes specify horizontal location, and the Z-axis specifies vertical location.



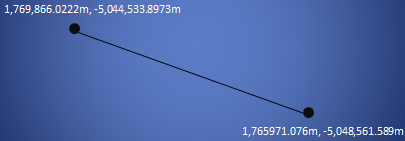
### Points

Points are specified as a pair of coordinates when considering horizontal location. As we are dealing with geographic locations, the coordinates will have a distance unit specified with it; in this case it is meters.



### Line

A line is based on at least two points. It will have a start and an end point which are referred to as nodes. With these two points the computer draws the lines between the two points when rendered on the screen. If a line has more than two points then the intermediate points are referred to as vertices and specify where the line turns or changes in attribute.



### Polygon

### A polygon is a closed accumulation of line elements. A polygon must be composed of at least two lines, consisting of at least three coordinate pairs. In our example here, the polygon is composed of four line elements, one for each side, and four nodes each consisting of a single coordinate pair.

### 

### Triangulated Irregular Network (TIN)

### A triangulated irregular network, commonly referred to as TIN, is a vector data structure that represents a surface as a set of contiguous, non-overlapping triangles. Within each triangle, its surface is represented as a flat plane. The triangles are made from sets of points, where each point is specified by a set of coordinates.

#### TIN example

### Here is an example of a triangulated irregular network. Note that the triangles do not overlap and the triangles vary in shape and size. All triangles have three edges meeting at three points and the surface inside the triangles will be represented as flat planes.

### [File:LakeMichiganMesh.png](//upload.wikimedia.org/wikipedia/commons/6/6c/LakeMichiganMesh.png)

### Networks

### One significant advantage of using vector data is that it can represent networks. A logical network can specify interconnectivity of objects, such as representing a network of streams and how they connect with each other. The lines can also be directly representing flow direction. If we are representing a street network, the lines could represent the direction of traffic flow. If we are representing streams as a network, the lines could represent the flow direction and volume of water.

### Network Data Sets

### Consider the network data set in the image. The black lines represent network paths. The yellow filled dots represent network links where multiple paths connect. We can add restrictions and cost of this network along each path. For instance, the red circle is a restriction that specifies that that network path is not passable.

### For the cost, we can assign each path the time it takes to traverse that path. Note that the non-passable path is assigned a value of infinity. Once we have assigned restrictions and costs to the network, we can then perform calculations such as finding the shortest path from a start point to an endpoint keeping in mind direction of flow and cost to traverse the network.

### [File:GISNetworkExample.svg](//upload.wikimedia.org/wikipedia/commons/0/04/GISNetworkExample.svg)

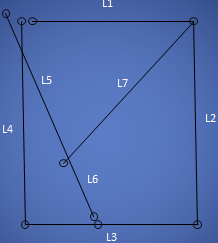
## Vector Data Computer Models

### What are Vector Data Computer Models?

Vector data is stored in different ways on the computer. There are two important vector data computer models that you should be aware of when dealing with vector data. They are the spaghetti model, and topological model.

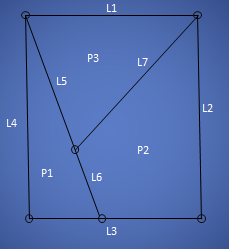
### Spaghetti Model

The spaghetti model is a simple model storing the data without a rigid structure. Lines are stored individually and the model does not record or recognize the connection of lines. Spaghetti model also does not record when lines cross. This also means that the shared boundary the polygon may have two separate lines, one for each polygon. The spaghetti model is still used regularly today even with the disadvantages because of its simplicity.

Here is an illustration of the spaghetti model showing seven lines. The seven lines are stored in no particular order in the spaghetti data model. Additionally, if the nodes of lines overlap exactly, those nodes are still stored as two separate nodes, one for each line. Lines that overlap in the spaghetti model are given no special preference nor are they even recognized, and nodes that should probably connect are also ignored because the spaghetti model does not keep track of connections or relationships. The spaghetti data model does provide the creator of the data ultimate say in what the data looks like, however, it does not provide for any advanced relationship tracking.

### Topological Model

The topological model is similar to the spaghetti model in that it records points, line, and polygons. It does differ significantly in the fact that it can enforce connectivity between vector elements. It also records connectivity and adjacency information and maintains other information on the relationships between the points, lines, and polygons. The topological model is more complex than the spaghetti model and it can record this advanced information.

Again, we have seven lines in our data set. The difference in this one is that rules were set up to state that lines must begin or end at the nodes of other lines and overlapping lines are not allowed. Additionally, since we have areas enclosed, these enclosed areas can also be considered polygons, and therefore we have assigned the three enclosed areas as the polygons P1, P2, and P3. Additionally, as we have these rules in place, if we move the node of a line, then all connected nodes will move with it. For instance, if we move the node connecting line two and line three together, then both line two and line three will move along with the node along with other connected lines such as line six, line five, and line seven.

### What is Topology?

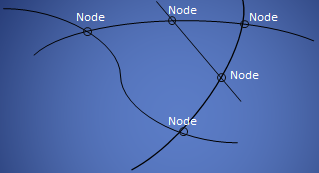
Topology studies the geometric properties that are preserved under continuous transformations. The topology records connectivity and adjacency relationships. These relationships are recorded separately from the coordinate data and do not change for certain operations that alter the shape of the features. This allows us to deform the features and have other features that have a relationship to the deformed feature change in concert with it thereby maintaining the original relationship. There are two types of the topology: planar, and non-planar.

### Planar

### Planar topology deals with two-dimensional surfaces only. In a planar topology, you may not have overlapping lines or polygons within the same data layer. At each line crossing, there must be an intersection in the intersection must be represented by a node. When polygons overlap, a new polygon must be created at that overlapping location.

#### Planar Lines

If we look at an example of planar topology with respect to lines we see that where lines intersect a node is created.



#### Planar Polygon

### Here is an example of planar topology which focuses on polygons. We see that where polygons 1 and 2 overlap, a new polygon, polygon 3, is created. When this happens, polygons 1 and 2 have the area representing polygon 3 removed from their geometries.

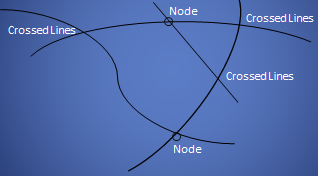
### 

### Non-Planar

Non-planar topology allows for overlap. For example, lines can overlap without an intersection being required and polygons can overlap without requiring another polygon to be created. Non-planar topology does not follow the same strict guidelines such as the planar.

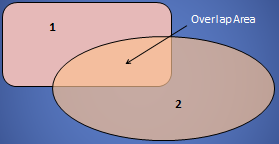
#### Non-Planar Lines

If we consider these lines in a non-planar topology, we can see that lines do cross without nodes being created at those locations, and lines have nodes created with a cross and others. It is up to the data creator to determine whether intersections should have nodes.



#### Non-Planar Polygon

In this example of non-planar topology, polygons 1 and 2 overlap, however, a 3rd polygon is not created where they overlap. Again, the data creator could choose to enforce overlapping rules here.



## Non-Spatial Attribute Relationships

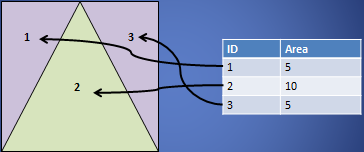
### What are Non-Spatial Attribute Relationships?

Non-spatial attributes can have relationships and these relationships can be treated much like tables that have relationships in databases. In a non-spatial attribute relationship a linkage is made between rows in a table in the spatial data in the topological table. These linkages can have one to one and many to one relationship similar to the relationships in a database.

### One-to-One Relationships

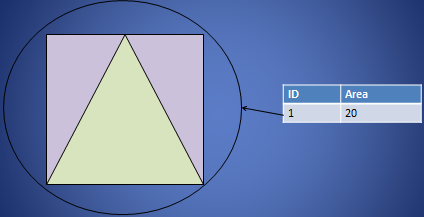
In a one-to-one relationship, for each feature in the spatial data layer, there is one, and only one, entry in the table containing the non-spatial data. This is similar to the relationships discussed in designing a database, as these rules also apply to this vector data model. This is the most common relationship when dealing with non-spatial attribute relationships.

In this illustration of a one-to-one relationship you can see that each polygon has a single entry in the table containing the non-spatial data.



### Many-to-One Relationships

In a many-to-one relationship, one table entry can have multiple features related to it in the spatial data. This illustration shows that a single entry the table relates to multiple objects in the spatial data.



## Topology

### What is Topology?

Topology is the mathematics of connectivity or adjacency of points, lines, or polygons. Topology determines exactly how and where points and lines connect by nodes. The order of connectivity defines the shape of an arc or polygon.

### Examples of Topology

To further your understanding, consider a few examples of topology. Perhaps the most common way in which topology is used is for network analysis. Network analysis can do things such as determine the shortest path between the start point and end point and calculating alternate paths. A GPS unit in a car or a mapping application on your mobile phone conducts network analysis on road data sets that have enforced topology.

Another example of topology is that during an emergency situation, emergency vehicles may use topology to gather multiple routes to decide which would be faster. For the third example, delivery vehicles used topology in deciding the fastest route to a location.

## Relationships

### What are Relationships?

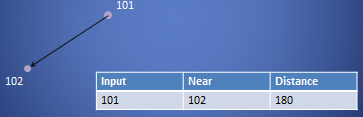
Within a topology, there are many different types of relationships that can be enforced. Each one of these relationship types provide a way for geographic objects to relate to each other. The relationship types which will be covered in this module are absolute and relative locations, distance between features, proximity of features, features in the “neighborhood” of other features, direction and movement from place to place, and Boolean relationships.

### Absolute and Relative Locations

In topology there are two types of locations: absolute and relative. An absolute location is considered to be mathematical location and is typically represented as a coordinate. For example, the absolute location of Corpus Christi Texas is 27.8003° North and 97.3961° West. Relative location is where we look at a place’s location relative to other landmarks. For example, the relative location of a local hamburger restaurant is right across the street from the grocery store.

### Distance between Features

The next relationship in a topology is distance between features. The distance between features simply shows the distance between input features.



### Proximity between Features

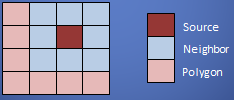
Proximity between features measures how close features are to one another. Outputs from proximity could be a buffer visually showing where features fall within the input feature, or a table listing all the features that were within a set distance of the input feature. This example shows a buffer around a single black line.



### Features in the “Neighborhood”

Features in the “neighborhood” of other features are what we consider polygon adjacency, overlaps, nodes, or coincident edges. Features are considered to be in the same neighborhood if they touch, overlap, share nodes, or share edges.

As an example, if we consider the 16 polygons in this image, with the red polygon being the source polygon, then the blue polygons are considered neighbors of the source polygon because they are either adjacent, share nodes, or coincident edges. The peach colored polygons are not considered neighbors because they do not meet any of the requirements to be considered a neighbor of the source polygon.



### Direction and Movement

Direction and movement is the storage of attributes relating to direction and movement. Examples of direction and movement attributes would be direction traffic travels along a street, and how fast the traffic travels along a street.

### Boolean Relationships

We can use Boolean relationships to combine attributes or other relationships to create more complex relationships inside of a topology. Consider these six Boolean relationships. First, is the ‘AND’ Boolean relationship which selects objects that have features in common. The ‘OR’ Boolean relationship selects features that are related in an “or” Boolean statement.

The inside relationship selects features that are inside of other objects and the outside relationship selects features that are outside of other objects. The intersecting relationship selects features that intersect other features and the non-intersecting relationship selects features that do not intersect other features.

## Topology Errors

### What are Topology Errors?

Topology errors can be classified by the type of vector geometry: point, line, and polygon.

#### Points

With points there are very few topology errors that can occur. One type of topology error that can occur of points is if the number of points is restricted somehow. For example, if we have a label point specified inside of a polygon, and there should only be one label point for each polygon. If a polygon has more than one label point then the labeling engine may not know which point to use as the anchor for the label. Additionally, if a polygon does not have a label point inside of it, then the label engine would not have anywhere to anchor its label.

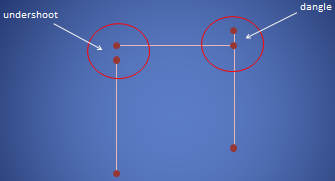
In this example, the red polygon does not have a label point, therefore the topology is invalid. The lilac colored polygon, does have a label point inside, therefore it has proper topology.

#### Lines

A line vector object has multiple ways in which the topology can have errors. Some common errors in line topology are when lines do not meet at nodes. Lines not meeting at nodes may overextend the lines that should have met which causes a dangle, or undershooting a line it should have met causing an undershoot.

Another type of error is if a continuous line has vertices inserted unnecessarily. Typically, a long continuous line should not have any vertices inserted unless the line changes direction. A third common topological error is if the direction of the line is incorrect thereby reversing the line’s flow.

This illustration has examples of both a dangle and an undershoot. To correct the undershoot the undershooting line should be extended so that it attaches at the other line’s node. To correct the dangle, the dangling line should be shortened so that it connects at the other line’s node.

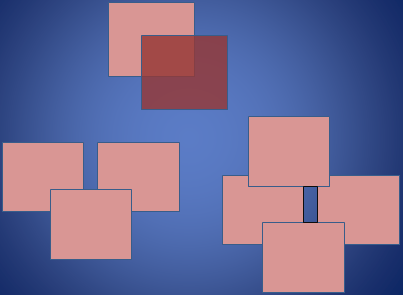


#### Polygons

There are three common topology errors with respect to polygons: overlapping polygons, unclosed gaps, and gaps between polygons. With overlapping polygons, if the two polygons represent a common boundary for instance, but the polygon boundaries do not match exactly, it will create small insignificant polygons commonly referred to as sliver polygons.

In unclosed gaps polygons that should be adjacent to each other are not thereby creating gaps were no gaps should exist. In the gaps between polygons topology error polygons that have proper relationships with each other may surround an empty area creating a gap between the polygons.

The top two polygons represent overlapping polygons that should have the same boundary but do not, thereby creating a sliver polygon or the overlap. The polygons on the lower left represent unclosed gaps for the top two polygons should touch of the middle but do not. The four polygons in the bottom right-hand corner demonstrate that while the polygons do interact properly with each other, it creates an empty gap inside that may need to be closed.



## SUMMARY

This lesson explained the concept of vector data structures and its elements. Cartesian Coordinates were covered and examples of lines, points, polygons were explained. The lesson identified two vector data computer models, Spaghetti and Topological and further explained topology. The relationships between topological and non-topological features were exhibited and explained.

## ASSIGNMENTS

1. Lab 4: Vector Data Structure
2. Quiz 4: Vector data structure