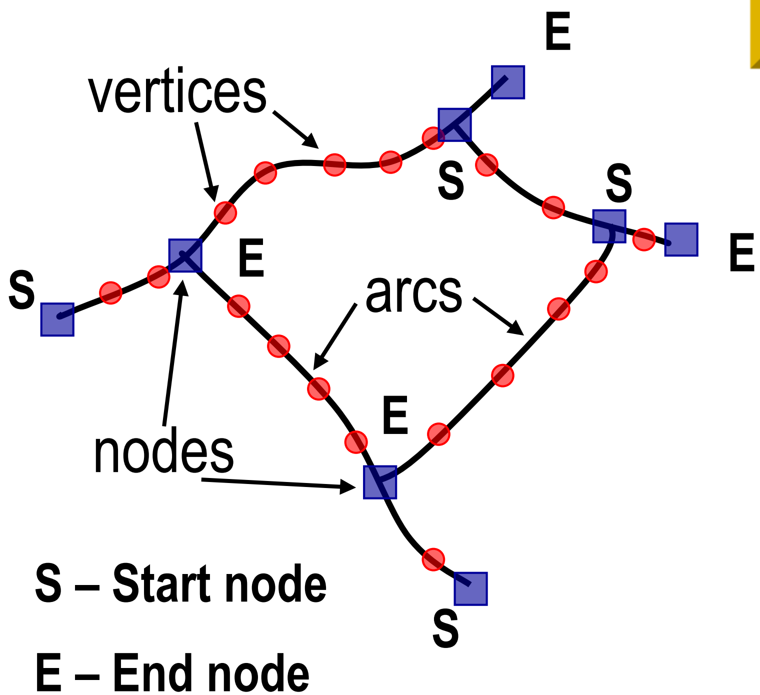
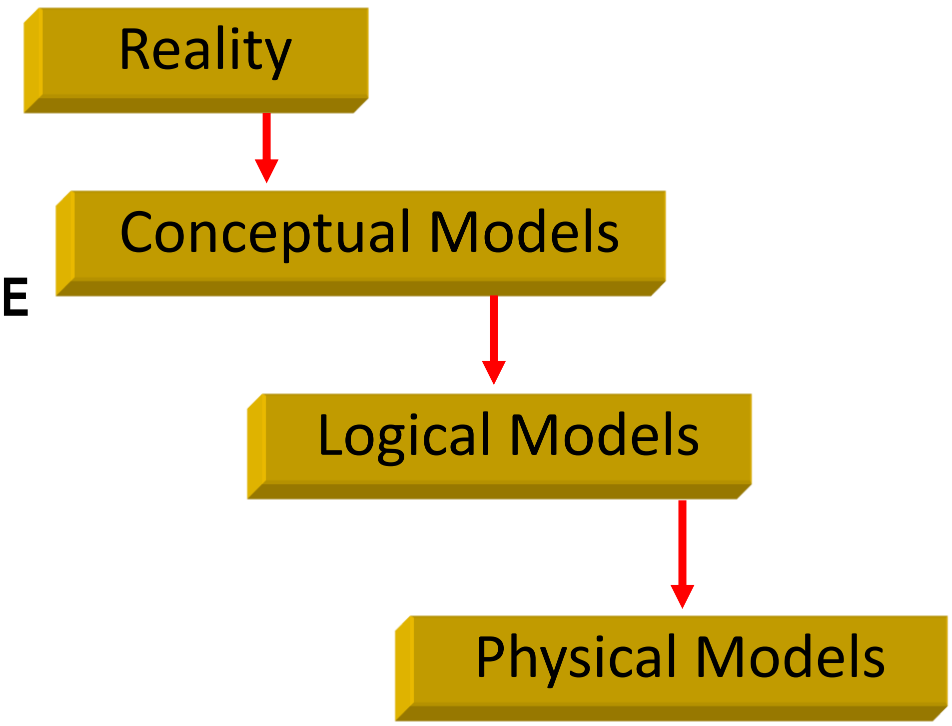
# Maps as Numbers: Data Models



# The Task

- An accurate, registered, digital map that can be queried and analyzed…

Translate:

Real World Locations, Paper Map ➡ Computer Files

Spatial Data Models, Topology

Entity Info. ➡ Queriable Database Files

Relational or Object-Oriented Databases

Relate Spatial Coordinates to Entity Info.

**"Spatial DBMS" software = GIS software!**

# Data Models

- How is reality abstracted and codified?

Abstraction

Codification

**Reality**

Wells produce from rocks that contain oil and gas

**Conceptual Models**

Wells are points, rock units are polygons (both are objects)

**Logical Models**

Well A penetrates Fm. 1; produces oil. Well B penetrates Fm. 3; produces gas. Fm 3 overlies Fm. 1.

**Physical Models**

Store well locations with a particular file structure, relationships in a dBase table. Associate table with location.

# Conceptual Models

Characterized all features or phenomenon as:

- *Discrete objects*; e.g. wells, roads, rock bodies, etc.
  - ***Object-based models***
- *Continuous phenomena*; e.g. gravity, magnetic intensity, topography, temperature, snowfall, soil pH, etc. = "fields" of values
  - ***Field-based models***
- Organize objects and fields by a *common theme; e.g. geology, hydrography, transportation*
  - ***Thematic layers***

# Logical Models

## ❑ VECTOR MODEL

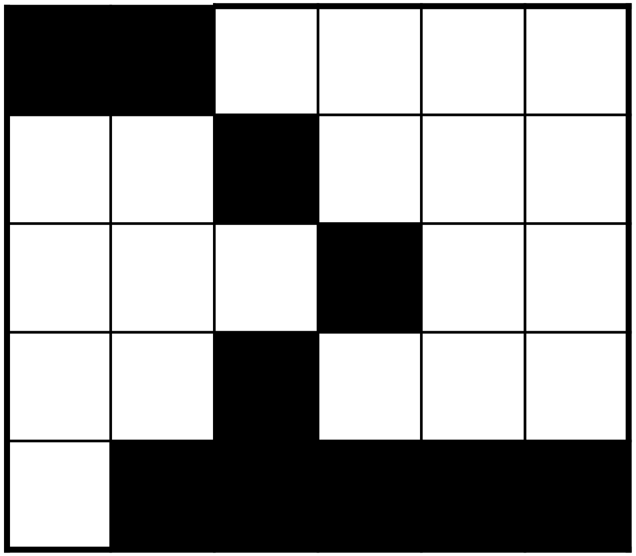
- ❑ Discrete objects are represented by points and vectors, continuous fields by irregular tessellation of triangles (A Triangulated Irregular Network: “TIN”)

## ❑ RASTER MODEL

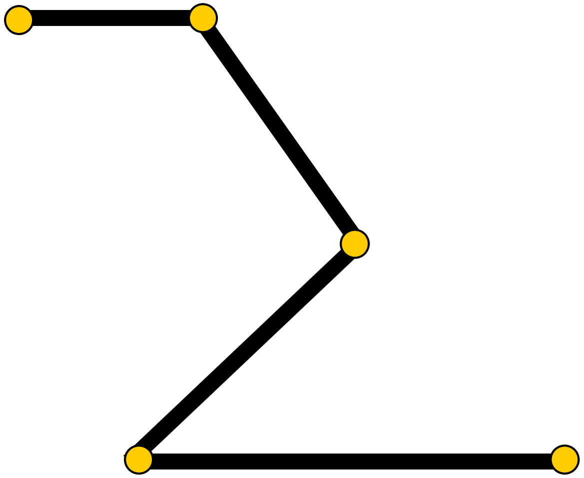
- ❑ Discrete objects and continuous fields are represented by an array of square cells (pixels)

# Logical Models

- How should discrete objects be coded?



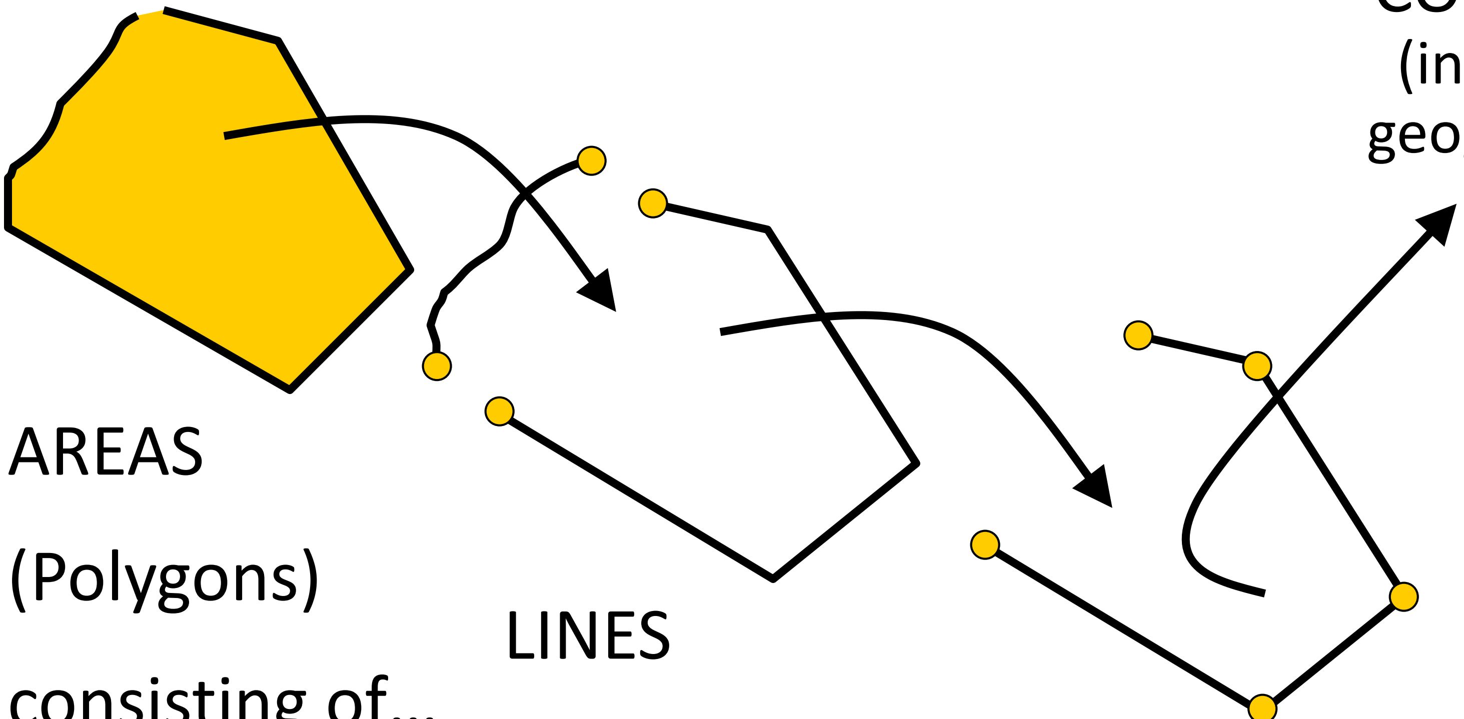
Raster Model



Vector Model

Logical Models

# Vector Model



AREAS

(Polygons)

consisting of…

LINES

(Arcs)

consisting of…

POINTS consisting of…

COORDINATES (in cartesian or geographic units)

(x, y)

(1, 5)

(5, 1)

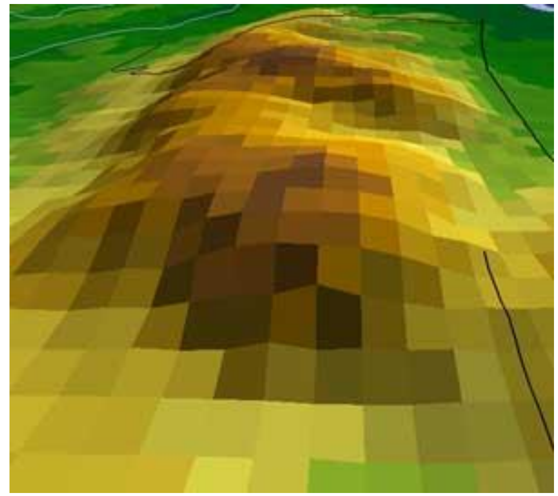
(7, 2)

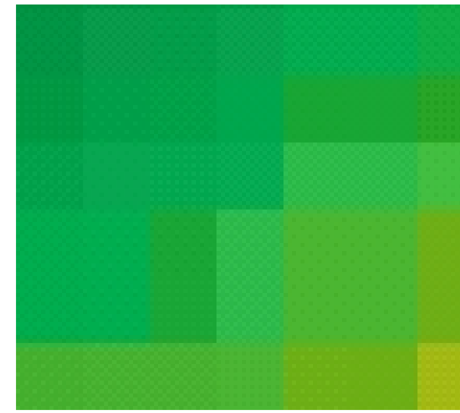
(5, 7)

(3, 8)

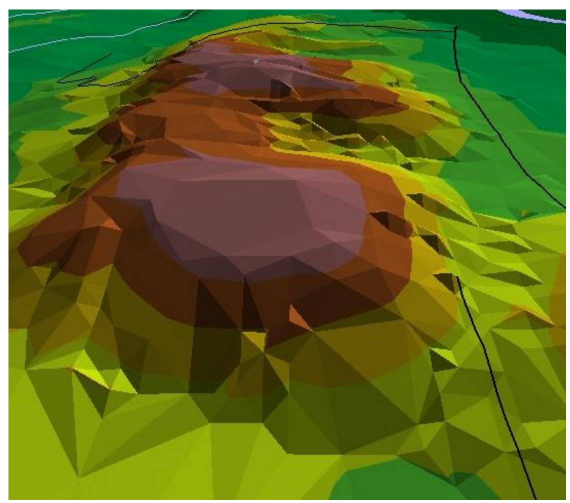
# Continuous Phenomena As Surfaces

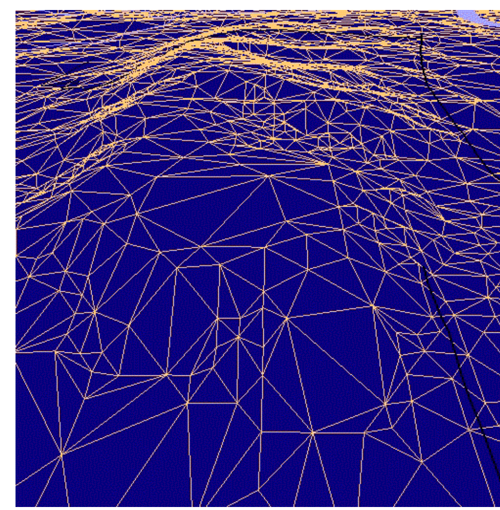
- Raster Topography
  - Regular tessellations, e.g. DEM, DTM





- Vector Topography
  - Irregular tessellations, e.g. T.I.N.





# Simple Vector Data Structure

Vector Line

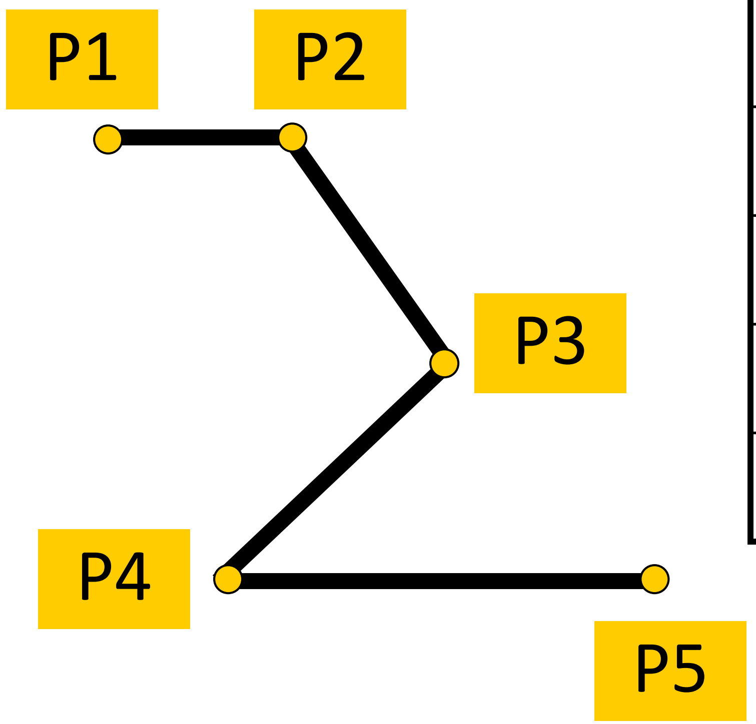


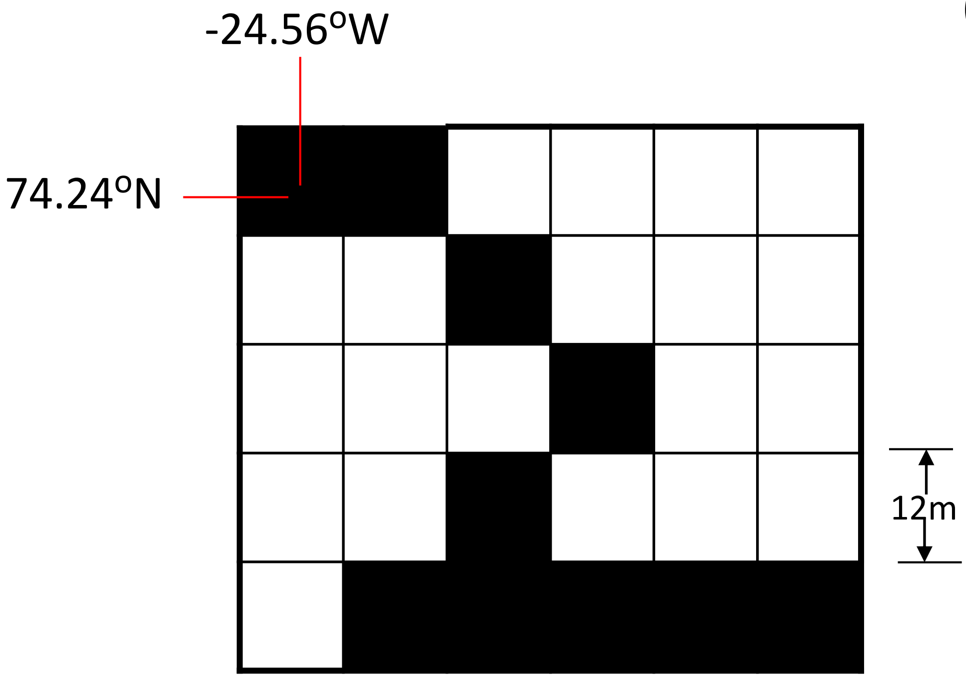

Table of Points

| ID | X | Y | Node |
|---|---|---|---|
| P1 | 503200 | 3200522 | From |
| P2 | 503250 | 3200522 | |
| P3 | 503300 | 3200460 | |
| P4 | 503245 | 3200410 | |
| P5 | 503350 | 3200410 | To |

(in UTM coordinates)

# Simple Raster Data Structure:

## Raster Line




"Dimension"= 5x6

"Resolution"= 12m

## Equivalent Binary "Flat File"

(Plus "Header" with Raster dimension, resolution and location)

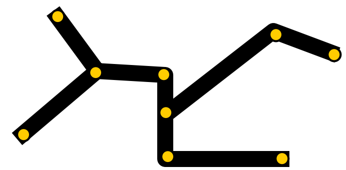
```
1 1 0 0 0 0
0 0 1 0 0 0
0 0 0 1 0 0
0 0 1 0 0 0
0 1 1 1 1 1
```

# Vector Models (Raster Next Time…)

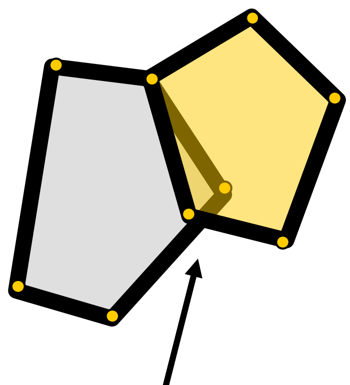
- “Graphical”
- Topologic/georelational
- T.I.N.
- Network

# “Graphical” Vector Model

- Lines have arbitrary beginning and end, like spaghetti on a plate



- Common lines between adjacent polygons duplicated



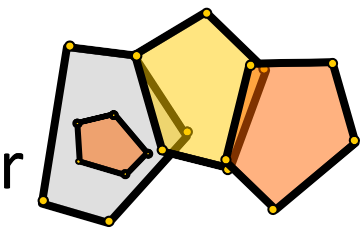
  - Can leads to “slivers” of unassigned area = “sliver polygons”

# “Graphical” Vector Model

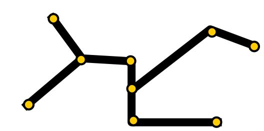
- Shortcomings for maps:
  - No real world coordinates required
  - No identification of individual objects; no way to attach attributes
  - Details of relationships among object (e.g. what’s adjacent, where a line starts and ends) not stored, but needed for spatial analysis

# Graphical Vector Structure

- Contains no explicit information about *adjacency*, *containment* or *contiguity* i.e.
  - Which polygons are adjacent?
  - Which polygons are contained within other polygons?



  - Which lines are connected? Where are they connected? Where do lines begin and end?



= “*Spaghetti Data Model*“

# Topological Vector Model

- Store pts. as x,y geographic coordinates
- Store lines as arcs of connected pts.
- Store polygons as closed paths

Also explicitly store ....

- Where lines start and end (connectivity)
- Which polygons are to the right and left side of a common line (adjacency)
- Containment: what's within what?

# Topology

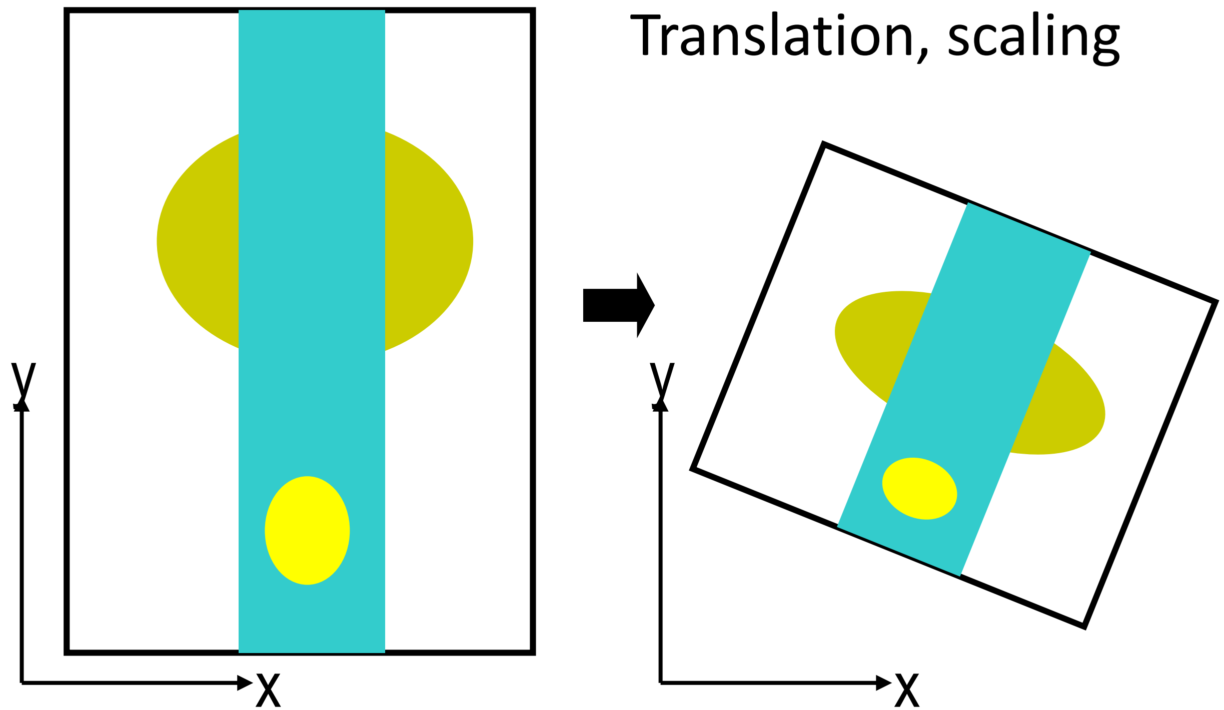
- The geometric relationship(s) between entities (e. g. points, lines, areas); where is one thing with respect to another?

# Topological Properties

- Spatial characteristics that are unchanged by transformations like *scaling*, *rotation* and *translation* are topologic
  - **Non-topological**: x, y coordinates, area, distance, orientation
  - **Topological:**
    - **Contiguity** – what's adjacent
    - **Connectivity** – what's connected
    - **Containment** – what's inside or outside of a region

# Topological Properties

- **Contiguity**: Adjacency
- **Connectivity**: What's connected
- **Containment**: What's inside or outside of a region



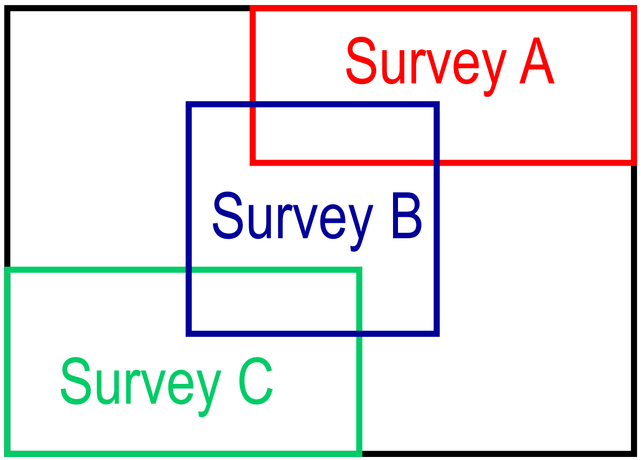

- Unchanged by translation, scaling, rotation

# Maintaining Topology: Planar Enforcement

- One and only one feature at every x, y location
  - Lines cross at nodes; polygons space-filling, exhaustive, mutually exclusive (no overlaps or gaps)
  - Sum of the area of all individual polygons equals the area of extent of all polygons
  - Common boundaries stored only once
- A PLANAR GRAPH meets these conditions
- Allows spatial queries for adjacency, containment and rapid what-is-where
- (All raster data is of this sort)

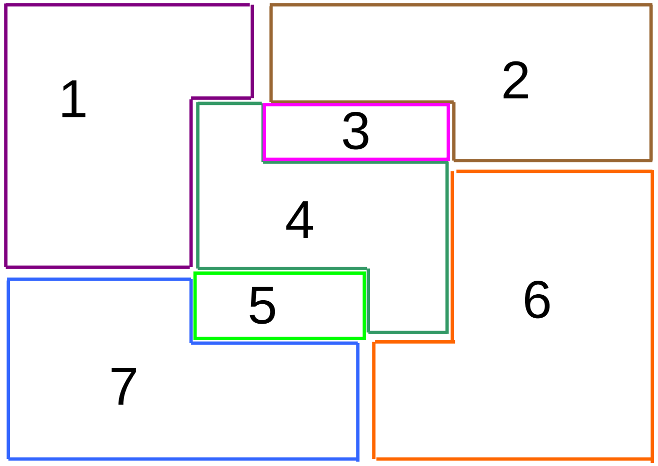
# Non-Planar vs. Planar Graphs

- Spaghetti




after Bonham-Carter, 1994

- Topologic




| Polygons | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
|---|---|---|---|---|---|---|---|
| Survey A | 0 | 1 | 1 | 0 | 0 | 0 | 0 |
| Survey B | 0 | 0 | 1 | 1 | 1 | 0 | 0 |
| Survey C | 0 | 0 | 0 | 0 | 1 | 0 | 1 |
| None | 1 | 0 | 0 | 0 | 0 | 1 | 0 |

# Lines: Graphic vs. Topologic

- Graphic (Spaghetti)

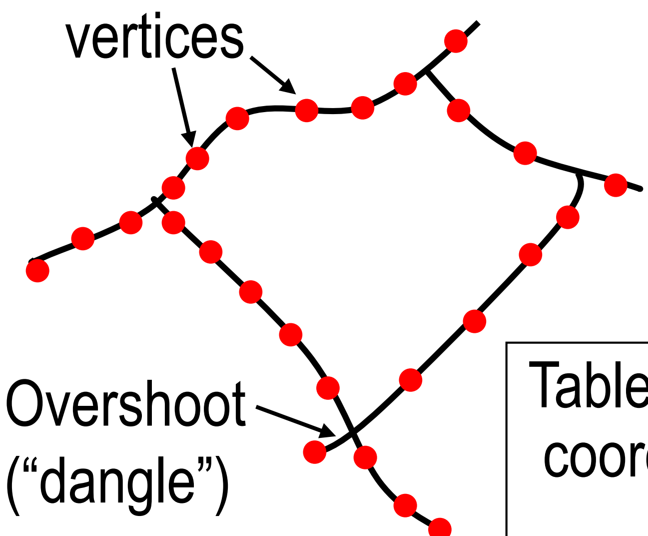
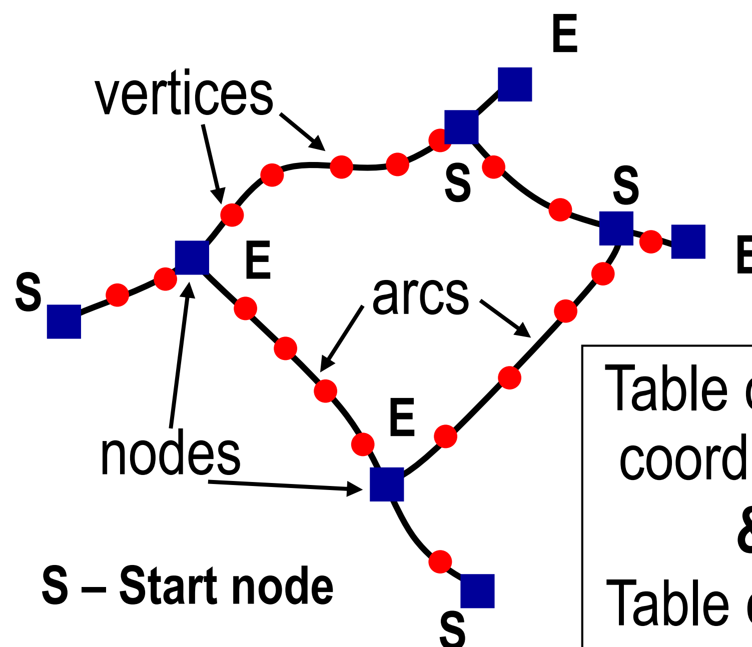



Table of (x,y) coordinates

- Topologic (with meatballs)




**S – Start node**

**E – End node**

Table of (x,y) coordinates
**&**
Table of arcs with IDs, starting and ending nodes

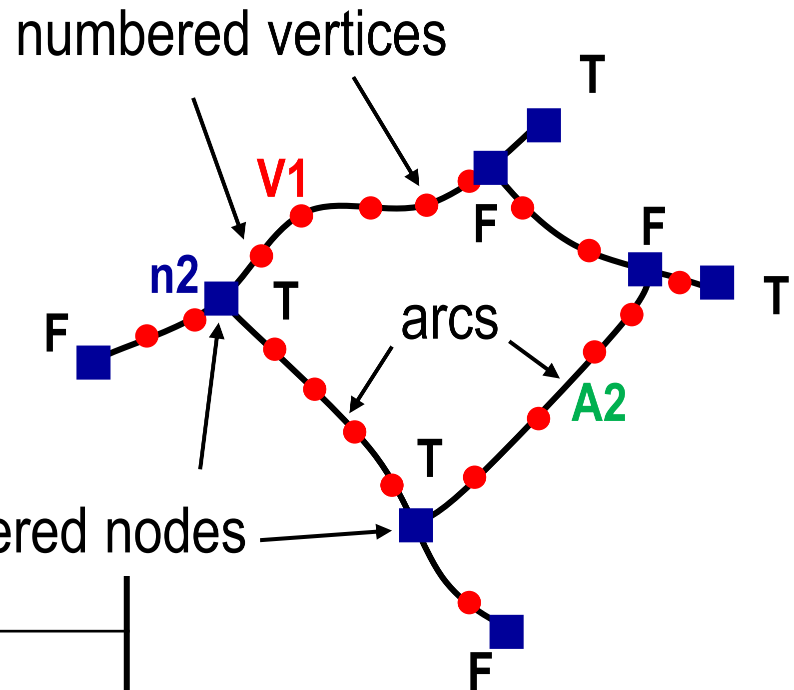
# Lines: Arc-Node Topology

**Vertex Table (V)**

| ID | x | y |
|---|---|---|
| 1 | 0 | 0 |
| . | . | . |
| . | . | . |
| 19 | 3 | 5 |

**Node Table (n)**

| ID | x | y |
|---|---|---|
| 1 | 0 | 0 |
| . | . | . |
| . | . | . |
| 8 | 3 | 5 |




**Arc Table (A)**

| ID | FID | F Node | T Node | Vertices |
|---|---|---|---|---|
| 1 | 100 | 1 | 2 | 1, 2 |
| 2 | 102 | 3 | 2 | 3, 4, 5, 6 |
| 3 | 103 | 3 | 4 | null |

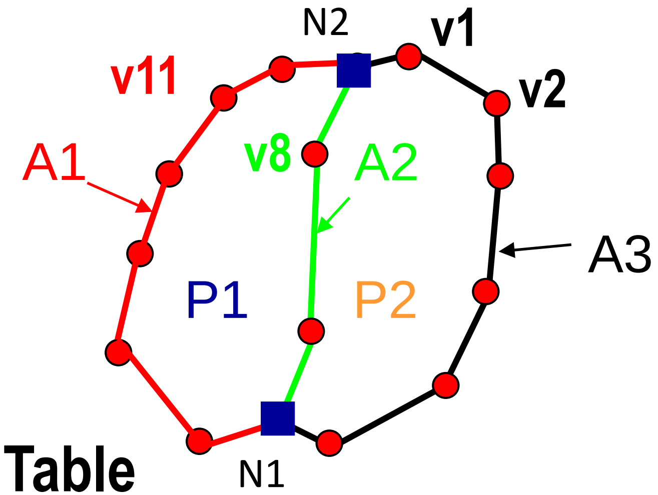
F =“Start” node (F: “From” node)

T = “End” node or (T: “To” node)

# Polygons: Polygon-Arc Topology

## Arc Table

| Arc ID | L. Poly | R. Poly | F Node | T Node |
|---|---|---|---|---|
| A1 | World | P1 | N1 | N2 |
| A2 | P1 | P2 | N2 | N1 |
| A3 | P2 | World | N2 | N1 |




## Polygon Table

| Poly ID | FID | Arcs. |
|---|---|---|
| P1 | 100 | A1, A2 |
| P2 | 102 | A2, A3 |

## Arc Coordinates Table

| Arc | Start | Vertices | End |
|---|---|---|---|
| A1 | N1 | v7, …,v11, … | N2 |
| A2 | N2 | …, v8 | N1 |
| A3 | N2 | v1, v2, …,v6 | N1 |

# Why Bother With Topology?

- Provides a way of error trapping and geometry validation after data entry
  - All lines must meet at nodes, all polygons must close, polygons can't overlap, all lines in a network must join
- *Permits spatial queries, precise measurements*

# What Kind of Queries Does Topology Permit?

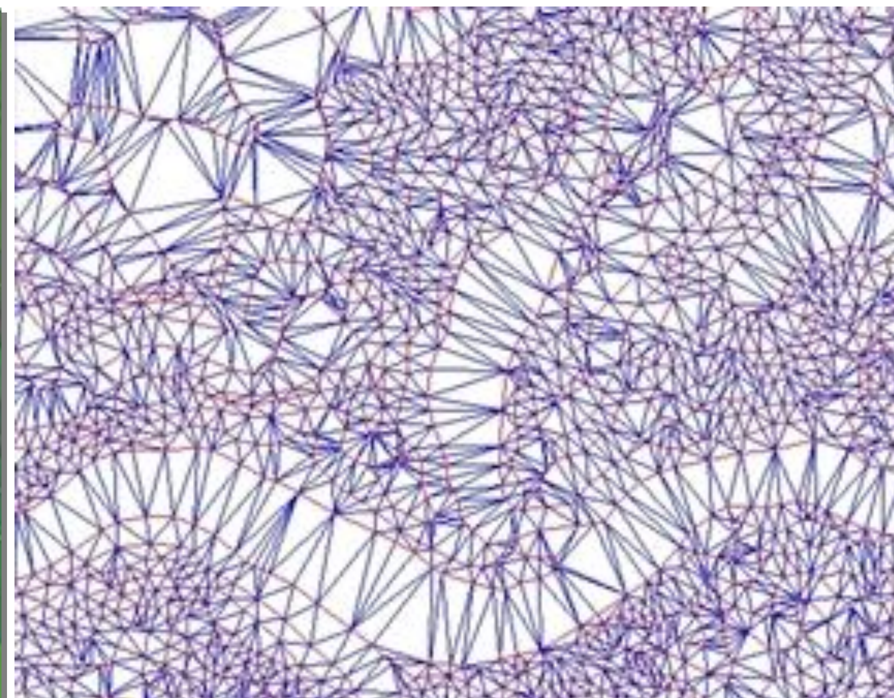
- Connectivity
  - What is shortest path between features or locations? (networks, flow)
  - Find all fault trace intersections
- Contiguity
  - What's adjacent: e.g. Show all granite/limestone contacts
  - Combine all contiguous units with a specific attribute (e.g. lithology) into a single unit
- Containment (= "Area Definition")
  - What proportion of an area is underlain by a specific rock type?
  - What is spatial density of specific feature(s)?

# Vector Models

- Graphical 
- Topologic/"georelational" 
- T.I.N. ←
- Network

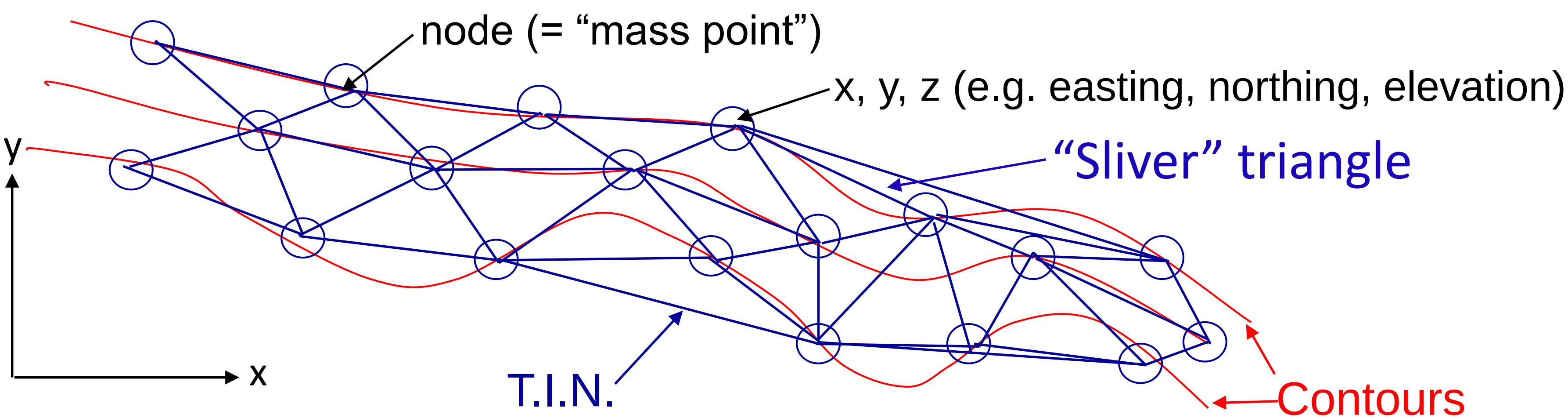
# Triangulated Irregular Network -TIN

- Topological 3-D model for representing continuous surfaces using a tessellation of triangles

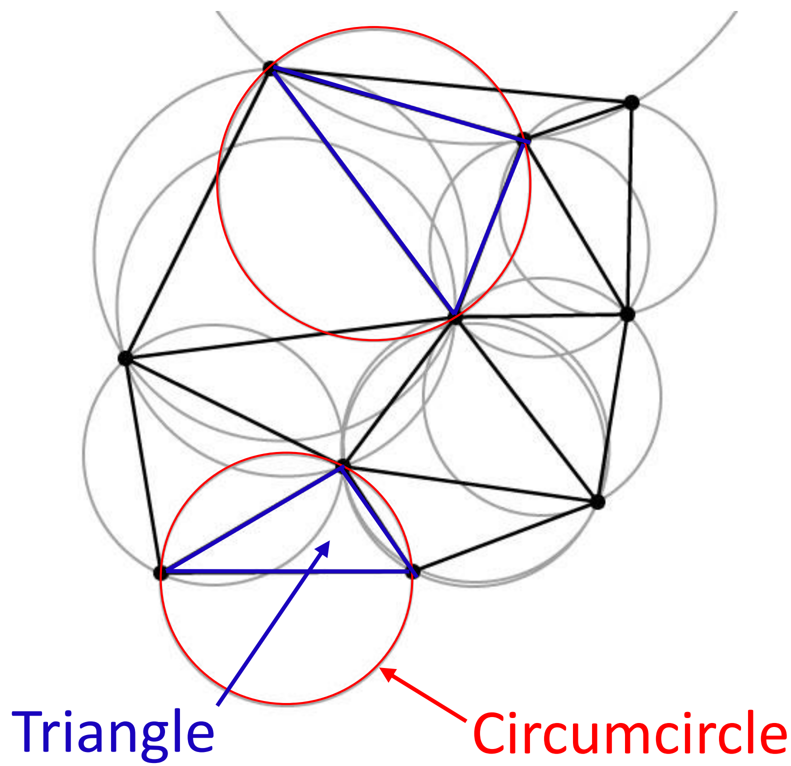
# Triangular Irregular Network

- Network ("tessellation") of edge-sharing triangles made from irregularly spaced points with x, y and z values
- Density of triangles varies with density of data points (e.g. spacing of contours) - c.f. raster with uniform data density – advantages for file size
- Triangle sides are constructed by connecting adjacent points so that the minimum angle of each triangle is maximized (see "Delaunay Triangulation" for details); i.e. a "fat" triangle, not a "sliver" triangle.
- Can render faces, calculate slope, aspect, surface shade, hidden-line removal, etc.
- Practical limit for computation on desktop is ~ 10-15 million nodes



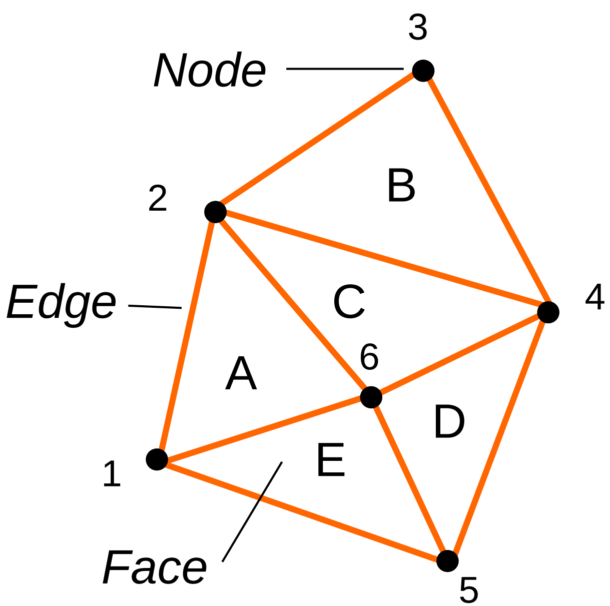
# How Are Triangle Created? Find the Delaunay Triangulation

- Find the set of circumcircles such that no point lies within a circumcircle
  - Circumcircle is the circle that passes through all 3 corners of a triangle
  - For 4 or more points on the same circumcircle (e.g. a rectangle) the D. Triangulation is not unique
  - For a set of points on a line, the D. Triangulation is degenerate (no triangle)




- D. Triangulation avoids sliver triangles – better represents average slopes and aspects

# TIN Topology




## Node Table

| Node | x | y | z |
|---|---|---|---|
| 1 | 3 | 5 | 5 |
| 2 | 5 | 9 | 12 |
| 3 | 11 | 12 | 16 |
| 4 | 15 | 5 | 3 |
| 5 | 13 | 3 | 44 |
| 6 | 10 | 7 | 50 |

Node Elevations

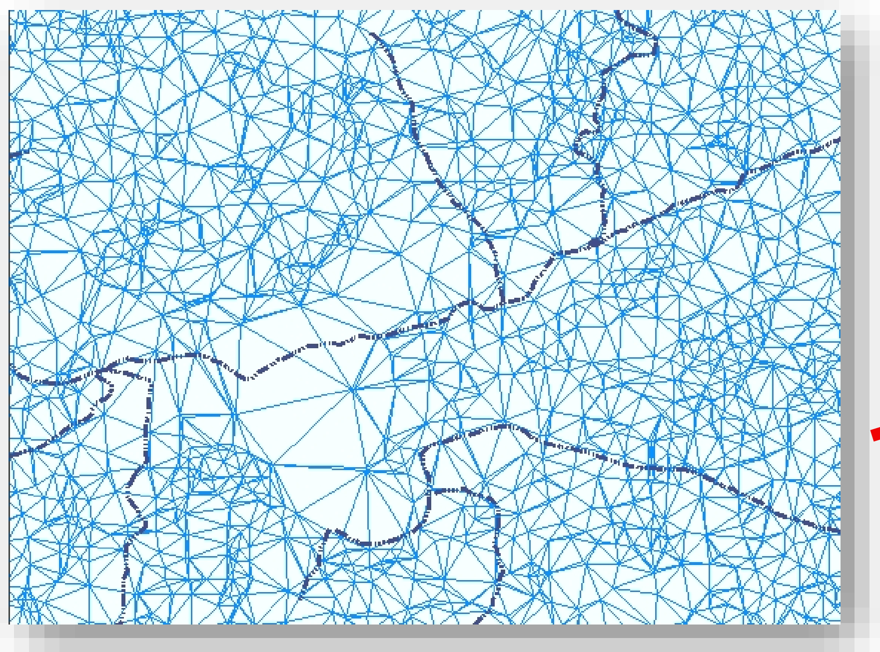
## Tin Topology Table

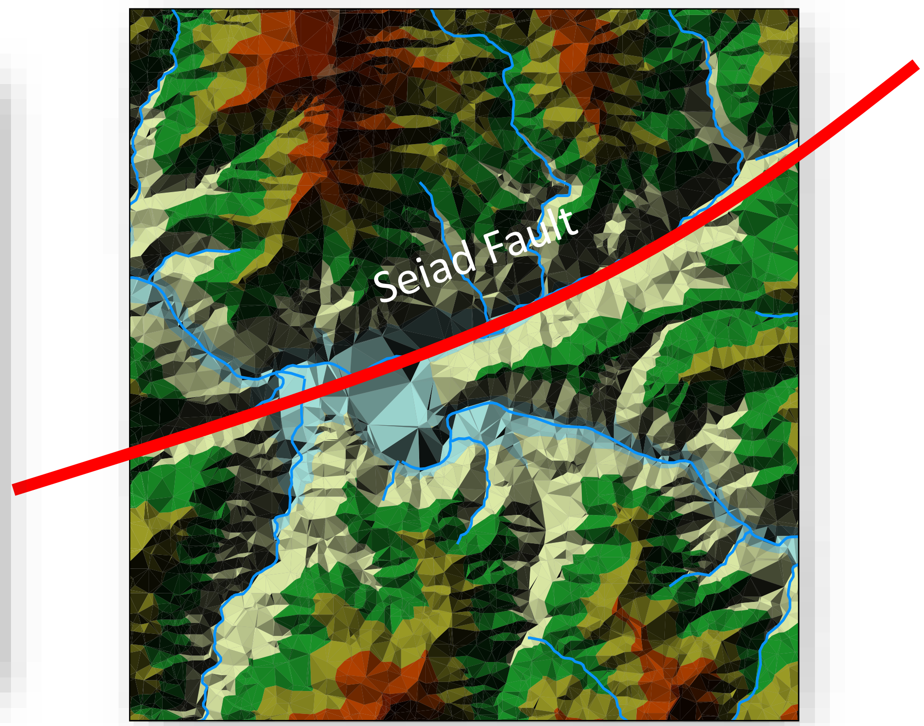
| Triangle | Node list | Neighbors |
|---|---|---|
| A | 1, 2, 6 | -, C, E |
| B | 2, 3, 4 | -, -, C |
| C | 2, 4, 6 | B, D, A |
| D | 4, 5, 6 | E, C, - |
| E | 5, 1, 6 | A, C, D |

After Zeiler, Modeling our World, p. 165

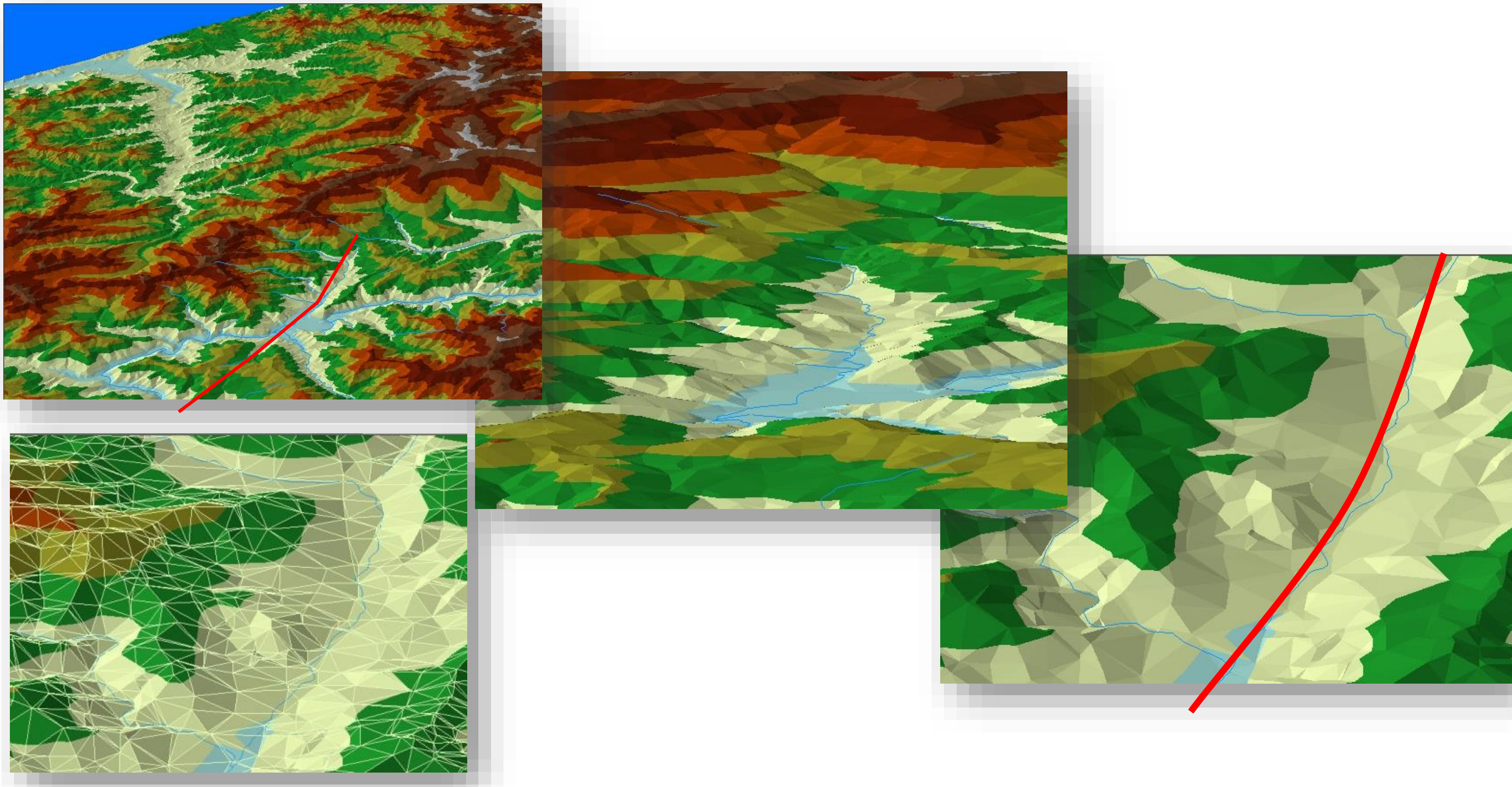
# TIN for Seiad Valley, CA

- Triangle edges symbolized
- Faces symbolized for elevation & aspect

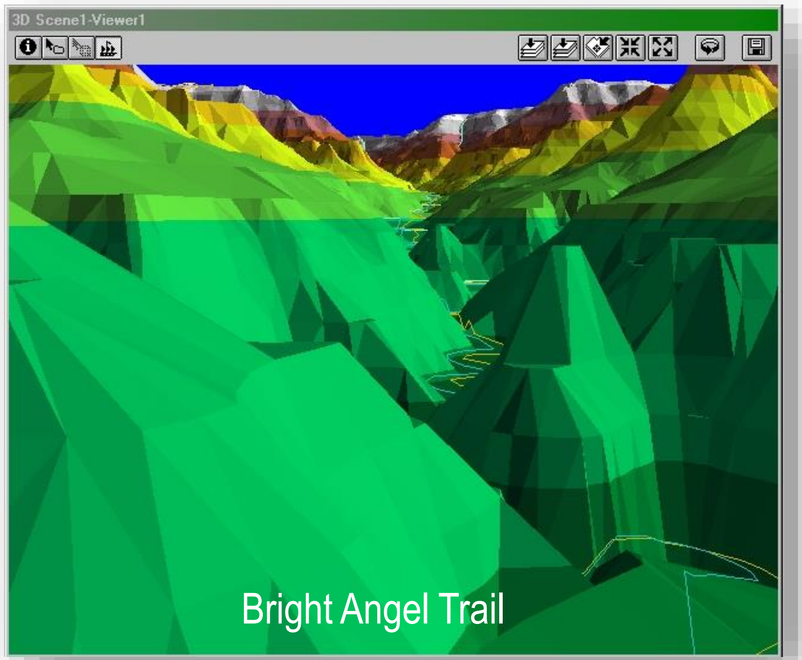


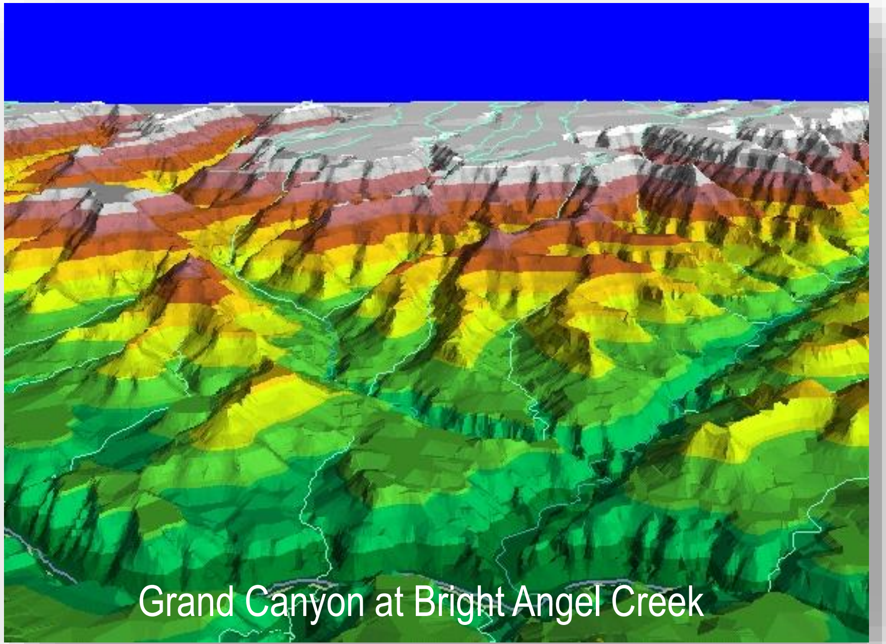


# 3-D TIN Scenes of Seiad Valley fault



# 3-D TINS, Grand Canyon



# Vector Models

- Graphical 
- Topologic/"georelational" 
- T.I.N. 
- Network - not discussed, see Help files