# Raster Data Model

- Why use a raster data model?
- Raster data representations
  - Cell value
  - Measurement, average, majority, significance
  - Cell-based vs. point-based
- Raster data components
  - Cells, Pixels, Grid
  - Cell size, spatial resolution
  - Bands
  - Spatial reference
  - Thumbnails
  - Pyramids
- Examples of raster data
  - Remotely sensed imagery (BV, DN)
  - DEM (elevation)
  - DRG (color)

# Raster Database

- ArcGIS Geodatabase Raster Catalog
- ArcGIS Geodatabase Mosaic Dataset
- ERDAS APOLLO Essentials – SDI (Spatial Data Infrastructure)

- Features
  - Spatial cataloging (raster footprints)
  - (Dynamic) mosaicking + histogram matching
  - (Dynamic) orthorectification
  - Generating derived layers (slope, hillshade, NDVI, etc)
  - Geoprocessing & image processing
  - Internet capability

# Raster Data Structure

- Header
  - Dimension and spatial resolution
  - Projection and coordinates
  - Thumbnails
- Data
  - Types: ascii, binary (1-bit), integer (8-bit), floating-point (4-byte)
  - Single- and multi-band structures (BSQ, BIL, BIP)
  - Quad tree
  - Data compression (lossless, lossy compression)
- Trailer
  - Color look-up table
  - Statistics
- (Supplemental) world file (.jpw, .tfw, .sdw, .xml)

# Raster Data Structure (cont.)

- Uncompressed/ lossless compression
  - Cell-by-cell encoding
  - Run-length encoding (RLE)
  - Quad Tree
- Lossy compression
  - JPEG
  - MrSID (Multi-resolution Seamless Image Database) – could be a lossless format

**LAN and GIS Files - Image Data**

LAN and GIS image files are stored in the same format. Each file contains a header record, followed by the image data. The image data are arranged in a Band Interleaved by Line (BIL) format. Each file is virtually unlimited in size - the file structure allows up to 274 billion bytes. The only size constraint is the capacity of the particular storage medium. The file consists of 512-byte records. The first 128 bytes of the first record contains the header information which consists of the following:

| Name | Byte(s) | Description |
|---|---|---|
| HDWORD | 1:6 | A 6 byte array containing 'HEAD74'. (Pre-7.4 files say 'HEADER'.) |
| IPACK | 7:8 | An integer value which indicates the pack type of the data: |
| | | 0 = 8 bit |
| | | 1 = 4 bit |
| | | 2 = 16 bit |
| NBANDS | 9:10 | An integer that indicates number of bands/channels per line. (Always 1 for GIS.) |
| | 11:16 | Unused. |
| ICOLS | 17:20 | An integer*4 number specifying the width of the file in pixels. |
| IROWS | 21:24 | An integer*4 number specifying the length of the file in lines of pixels. |
| XSTART | 25:28 | An integer*4 number specifying the database x-coordinate of the first pixel (upper left) in the file. |
| YSTART | 29:32 | An integer*4 number specifying the database y-coordinate of the first pixel (upper left) in the file. |
| | 33:88 | Unused. |
| MAPTYP | 89:90 | An integer which indicates the type of map projection associated with the file. See PRO files for a complete list. |
| NCLASS | 91:92 | An integer which indicates the number of classes in the data set. |
| | 93:106 | Unused. |
| IAUTYP | 107:108 | An integer which indicates the unit of area associated with each pixel: |
| | | 0 = NONE |

Raster Header Information

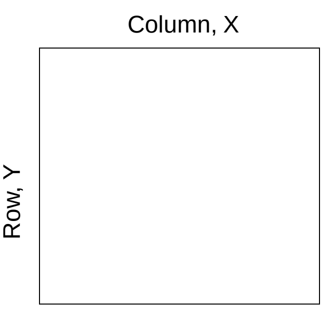
# Raster Coordinates

- Example
- Columns, Rows = 318, 463
- Cellsize X, Y = 30, 30
- LL Map Coor. = 499995, 5177175

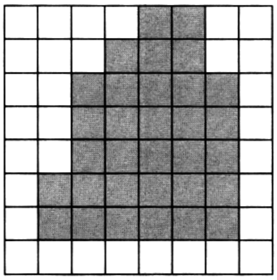
- Map Coor at Column, Row (= m, n)

  X = 499995 + m x 30

  Y = 5177175 – (463 – n) x 30

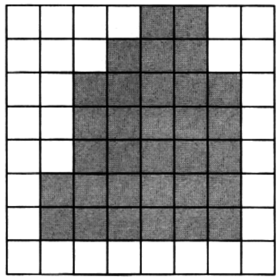


## Cell-by-Cell



Row 1: 0 0 0 0 1 1 0 0
Row 2: 0 0 0 1 1 1 0 0
Row 3: 0 0 1 1 1 1 1 0
Row 4: 0 0 1 1 1 1 1 0
Row 5: 0 0 1 1 1 1 1 0
Row 6: 0 1 1 1 1 1 1 0
Row 7: 0 1 1 1 1 1 1 0
Row 8: 0 0 0 0 0 0 0 0

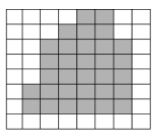
## RLE

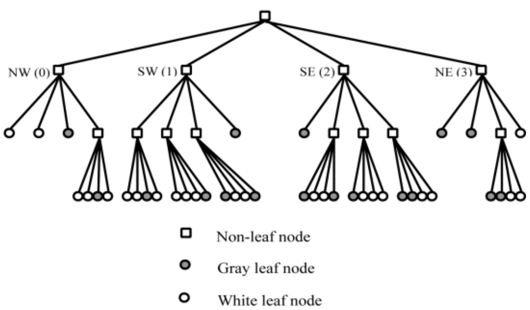


Row 1: 5 6
Row 2: 4 6
Row 3: 3 7
Row 4: 3 7
Row 5: 3 7
Row 6: 2 7
Row 7: 2 7

## Quad tree

| | |
|---|---|
| 0 | 3 |
| 1 | 2 |

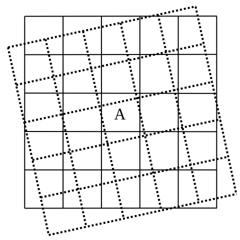




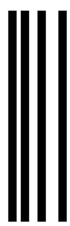

(02, 032), (102, 113, 120, 123, 13), (20, 210, 213, 220, 230, 231), (30, 31, 320, 321)

# Geometric Transformation

- Affine transformation (rotation, translation, scaling)
- Resampling
  - Nearest neighbor (the nearest cell)
  - Bilinear interpolation (4 nearest cells)
  - Cubic convolution (16 nearest cells)




# **Testing Interpolator Quality**

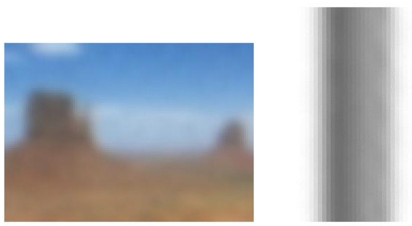


Helmut Dersch
(der@fh-furtwangen.de)
Technical University Furtwangen
Revised June 30, 1999

http://www.all-in-one.ee/~dersch/interpolator/interpolator.html

Original

Nearest Neighbor

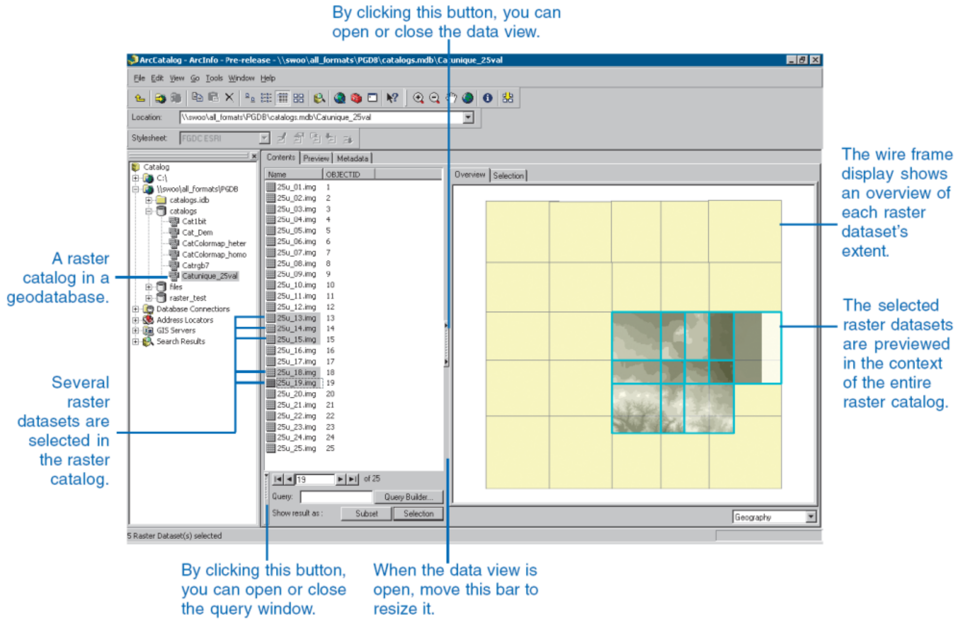


Bilinear Interpolation

Cubic Convolution

# Raster Data in ArcGIS

- GRID
- .img, .lan, .gis
- .tiff (.tif)
- MrSID
- Geodatabase
  - – Raster Band
  - – Raster Dataset
  - – Raster Catalog
    - Managed/unmanaged




# Header Information

- TIFF(**T**ag(ged) **I**mage **F**ile **F**ormat) – GeoTIFF, .tfw (TIFF World File)
- MrSID (Multi-resolution Seamless Image Database) - .sdw (MrSID World File)

## NetCDF (Network Common Data Form)

http://www.unidata.ucar.edu/software/netcdf/index.html

- NetCDF (network Common Data Form) is a set of software libraries and machine-independent data formats that support the creation, access, and sharing of **array-oriented** scientific data.

- ArcGIS can create a multidimensional layer from NetCDF. The dimension could be time, altitude, depth, etc. At any given time, only one slice of multidimensional data is visible.

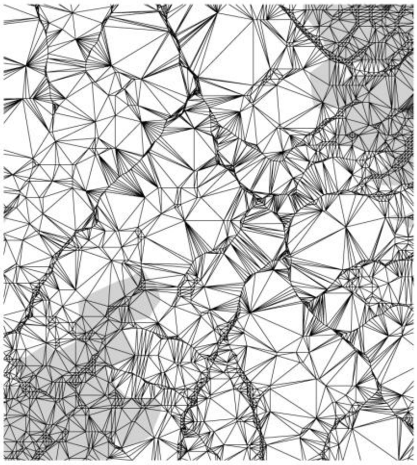
- Main ArcGIS application: animation

## What is a layer file (.lyr) in ArcGIS?

- Layer files reference geographic data stored on disk. You can think of them as a cartographic view of your geographic data. They are separate files on disk and have a .lyr extension.

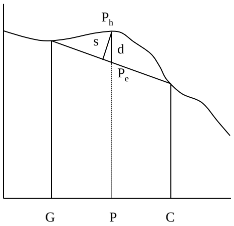
## Triangulated Irregular Network (TIN)

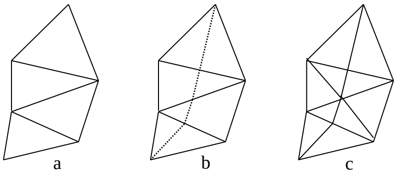
- Point (x, y, z)
- Edge
- Triangles
- Topology
  - –The triangle number
  - –The numbers of each adjacent triangle
  - –The three nodes defining the triangle
  - –The x, y coordinates of each node
  - –The surface z value of each node
  - –The edge type of each triangle edge (hard or soft)



## Generating TIN

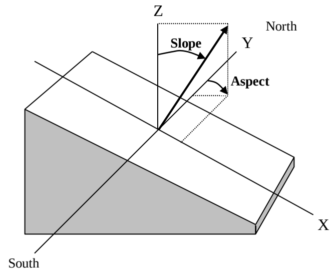
- Selection of significant points from DEM
  - – Very Important Points (VIP)
  - – Maximum z-tolerance
- Delaunay triangulation
  - – Breaklines (soft, hard)



# Terrain Analysis

- Contouring
- Slope
- Aspect
- Curvature
  - Convex vs. concave
  - Profile curvature
  - Planiform curvature
  - Surface curvature = profile – planiform

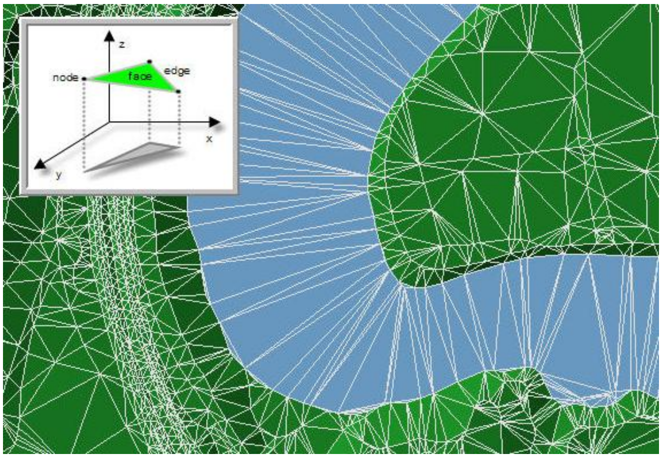



# ArcGIS 9.3 Terrain Dataset

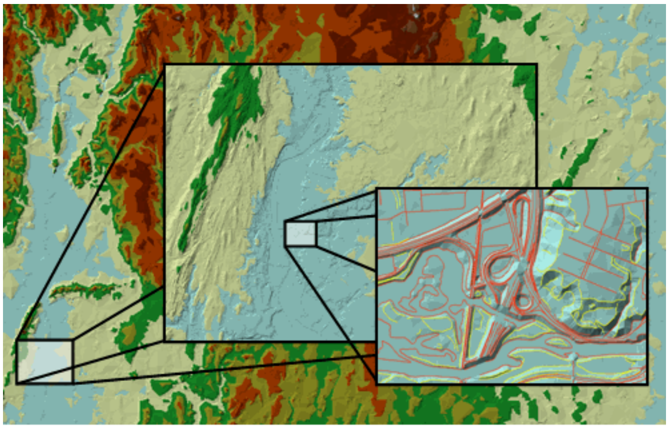
Terrain Dataset
- LiDAR/SONAR: mass points
- Contours/ridgelines/streamlines: breaklines
- GPS 3D elevation points
- Lake surfaces: polygons

# Terrain Dataset Concepts

- Terrains use a TIN data structure to represent surfaces




- Terrains use pyramids (multi-resolution TIN) to represent multiple levels of resolution



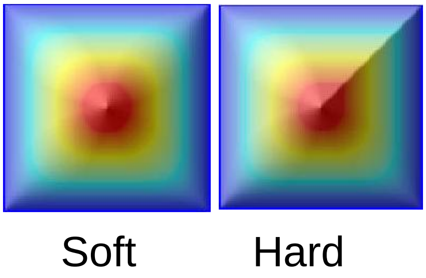
Types of feature class data sources in terrains

- **Mass points**
- **Breaklines**
- **Clipping polygons**
- **Erase polygons**
- **Replace polygons**
- **Hard and soft qualifiers**

# Mass Points

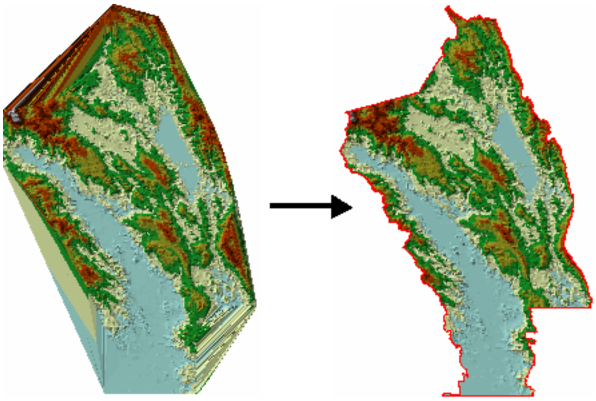
| Surface feature type | Feature class | Z value source in the feature class |
| --- | --- | --- |
| Mass points (x,y,z locations) | Point feature class | • Shape geometry, x,y,z per point<br>• Attribute column holding a z-value for each point |
| | Multipoint feature class | • Shape geometry, x,y,z per point<br>• Attribute column holding one z-value per shape |
| | Line feature class | • Shape geometry, x,y,z per vertex<br>• Attribute column holding one z-value per shape (such as a contour) |
| | Polygon feature class | • Shape geometry, x,y,z per vertex<br>• Attribute column holding one z-value per shape (such as a shoreline) |

# Breaklines




| Surface feature type | Feature class | Z value source in the feature class |
|---|---|---|
| Breaklines (hard or soft) | Line feature class | • Shape geometry, x,y,z per vertex<br>• Attribute column holding one z-value per shape (such as a contour)<br>• No height source; z-values are interpolated for each feature from the surface before being added |
| | Polygon feature class | • Shape geometry, x,y,z per vertex<br>• Attribute column holding one z-value per shape (such as a shoreline)<br>• No height source; z-values are interpolated for each feature from the surface before being added |

# Clipping Polygons



| Surface feature type | Feature class | Z value source in the feature class |
|---|---|---|
| Clipping polygons (hard or soft) | Polygon feature class | • Shape geometry, x,y,z per vertex<br>• Attribute column holding one z-value per shape (such as a shoreline)<br>• No height source; z-values are interpolated for each feature from the surface before being added |

# Erase Polygons

- Define holes in a terrain. These are used to represent areas for which you have no data or want no interpolation to occur. They will display as voids, and analysis will consider them to be areas of NoData.

| Surface feature type | Feature class | Z value source in the feature class |
|---|---|---|
| Erase polygons (hard or soft) | Polygon feature class | • Shape geometry, x,y,z per vertex<br>• Attribute column holding one z-value per shape (such as a shoreline)<br>• No height source; z-values are interpolated for each feature from the surface before being added |

# Replace Polygons

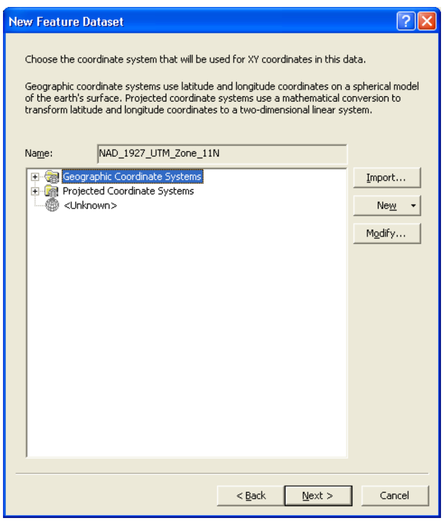
| Surface feature type | Feature class | Z value source in the feature class |
|---|---|---|
| Replace polygons (hard or soft) | Polygon feature class | • Shape geometry, x,y,z per vertex<br>• Attribute column holding one z-value for all vertices in each shape (such as a lake) |

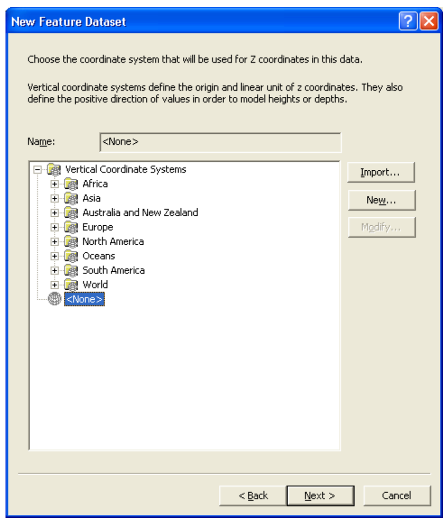
# Summary

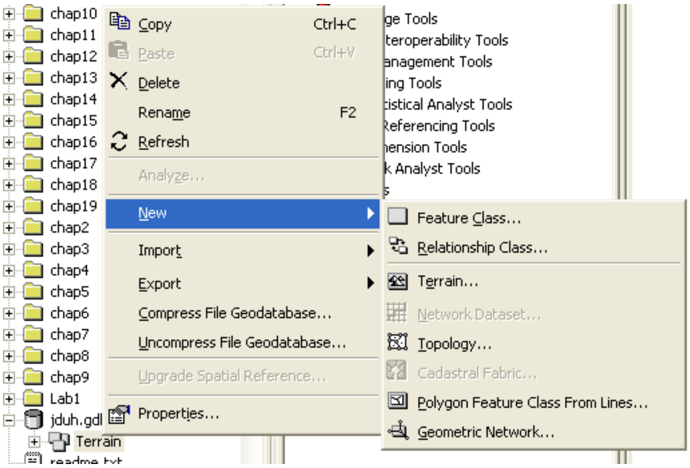
| Thematic type | Feature geometry type | SFType |
|---|---|---|
| spot heights<br>survey points<br>gps points | point | masspoint |
| mass points<br>LIDAR points<br>SONAR points | multipoint | masspoint |
| edge of pavement<br>canals<br>other linear features | polyline | hardline |
| contours | polyline | masspoint<br>softline |
| water bodies | polygon<br>polyline | hardline<br>hardreplace |
| obscured areas | polygon | softerase |
| study area boundary | polygon | softclip |

# Working with Terrain Dataset

- Terrain






## Special considerations: Rasters, all return LIDAR, TINs, and contours

**Rasters**

- Terrains should be made from vector-based source measurements rather than rasters. Without the inclusion of ancillary data to make improvements, the conversion of a raster to a terrain could not produce a better surface.

**All return LIDAR**

- Terrain pyramids are most effective with bare-earth LIDAR. The z-tolerance-based filtering employed by the pyramiding process does not work as well when tree canopy is included.

**TINs**

- A terrain should be made from the original features used to build a TIN rather than the TIN itself.

**Contours**

- Contours, as with rasters, are not the best source of data from which to build a terrain. They should be included in the terrain using an SFType of masspoint. Softline is also a possibility but less efficient.