Raster Representations and Calculations

- The **raster concept**:  
  - A **2-D array** of attributes  
    - Each represented by **mathematical values**  
  - **Locations** on the cells on the ground are **implicitly encoded** based on their **row-column** positions
Spatially Straightforward, But What About Value Encoding?
We are building a **model of reality** here:
- We can make model **design decisions** based on what the **intended application** is

All models **selectively throw information away**:
- Whether presence/absence, or ordinal / interval / ratio categories or counts, any particular approach will be **selectively useful**

**Systematic coding strategies for determining cell values:**
1. Use the value from the cell center (centroid)
2. Use majority weighting within each cell
3. Calculate weighted values
   (+ **non-systematic** most important type)
Systematic Coding Strategies for Cell Values

1. Use the value from the cell center (centroid)

- The value at the centroid is assigned to the cell
- This is a simple approach, but it can overrepresent the values from small areas
Systematic Coding Strategies for Cell Values

2. Use majority weighting within each cell

• The value covering the **majority** of the area is assigned to a cell

• This is a “**fairer**” representation than cell centers
Systematic Coding Strategies for Cell Values

3. Calculate weighted values

- Priority weights are based upon the **importance** of different values
- The "**most important**" value present is assigned to a cell
- This ensures the representation of **crucial** geographic phenomena
Dealing With Crossing Linear Objects

• What happens when more than one linear object occurs in a single cell?
  – If each theme were separate, there is no problem
    • i.e. roads = one theme, rail = another
    • Use presence/absence coding
  – Otherwise…
    • Use most important type method
      – Requires you to decide which is most important
Crossing Linear Features

Most Important Type Method

Transportation Matrix

Data Matrix

= No Data
Separating The Objects into Individual Themes
Using Extended Raster Model with Crossing Linear Objects

Note how category 3 can be used to represent when roads cross.

<table>
<thead>
<tr>
<th>Value</th>
<th>Count</th>
<th>Description</th>
<th>Type</th>
<th>Construction</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>5</td>
<td>Highway</td>
<td>4 Lane</td>
<td>3/20/97</td>
</tr>
<tr>
<td>2</td>
<td>4</td>
<td>Road</td>
<td>2 Lane</td>
<td>9/17/99</td>
</tr>
<tr>
<td>3</td>
<td>2</td>
<td>Hwy/Road</td>
<td>N/A</td>
<td>N/A</td>
</tr>
</tbody>
</table>

Note also how additional attributes can be recorded with this method.
Cell Coincidence

- The **raster concept**:  
  - Each grid cell location for each theme **explicitly coincides** with its other thematic counterparts  
  - The **efficiency** of raster GIS modeling **depends** on this

\[
\text{TOTCOST}(4,2) = \text{COST1}(4,2) + \text{COST2}(4,2)
\]

\[26 = 20 + 6\]
Matrix Algebra

\[
\begin{pmatrix}
5 & 4 & 1 \\
2 & 1 & 2 \\
4 & 2 & 1 \\
\end{pmatrix}
+
\begin{pmatrix}
3 & 2 & 1 \\
1 & 4 & 5 \\
2 & 7 & 3 \\
\end{pmatrix}
=
\begin{pmatrix}
8 & 6 & 2 \\
3 & 5 & 7 \\
6 & 9 & 4 \\
\end{pmatrix}
\]

Map Algebra

\[
\begin{array}{ccc}
5 & 4 & 1 \\
2 & 1 & 2 \\
4 & 2 & 1 \\
\end{array}
+
\begin{array}{ccc}
3 & 2 & 1 \\
1 & 4 & 5 \\
2 & 7 & 3 \\
\end{array}
=
\begin{array}{ccc}
8 & 6 & 2 \\
3 & 5 & 7 \\
6 & 9 & 4 \\
\end{array}
\]

**Arithmetic operations:** the same for -, but not *, /, mod
Matrix Algebra

\[
\begin{pmatrix}
5 & 4 & 1 \\
2 & 1 & 2 \\
4 & 2 & 1 \\
\end{pmatrix} \times \begin{pmatrix}
3 & 2 & 1 \\
1 & 4 & 5 \\
2 & 7 & 3 \\
\end{pmatrix} = \begin{pmatrix}
21 & 33 & 28 \\
? & ? & ? \\
? & ? & ? \\
\end{pmatrix}
\]

Map Algebra

\[
\begin{array}{c|c|c|c}
5 & 4 & 1 \\
\hline
2 & 1 & 2 \\
\hline
4 & 2 & 1 \\
\end{array} \times \begin{array}{c|c|c}
3 & 2 & 1 \\
\hline
1 & 4 & 5 \\
\hline
2 & 7 & 3 \\
\end{array} = \begin{array}{c|c|c}
15 & 8 & 1 \\
\hline
2 & 4 & 10 \\
\hline
8 & 14 & 3 \\
\end{array}
\]

Arithmetic operations: the different for $\ast$, $/$, $\mod$
Introduction to Map Algebra

- Language components
- Syntax and rules
- Objects
- Operators
- Commands
Language Components

• A data manipulation language for raster
  – Math-like expressions
    • AgSuit = (SoilSuit * 0.75) + (SlpSuit * 0.25)

• Parts of the language
  – Objects: Raster, numbers, constants, and so on
  – Operators: “+”, “/”, “GT”, “LE”, “AND”, “OR”, and so on
  – Functions: Slope, FocalMean, Sin, and so on
  – Rules: For building expressions and using functions

• Most operators & functions implemented as tools
Map Algebra operators

**Arithmetic**
- Addition
- Subtraction
- Multiplication
- Division
- Modulus
- Unary minus

**Boolean**
- AND Logical And
- OR Logical Or
- XOR Logical Xor
- NOT Logical complement

**Combinatorial**
- CAND Combinatorial And
- COR Combinatorial Or
- CXOR Combinatorial Xor

**Relational**
- EQ Equal
- NE Not equal
- LT Less than
- LE Less than or equal
- GT Greater than
- GE Greater than or equal

**Logical**
- DIFF Logical difference
- IN {list} Contained in list
- OVER Replace

These work with two objects, like: **Slope GE 10**
Relational Operators in Map Algebra

- Relational Operators \((<,>,==,\geq,\leq)\)

\[
\begin{array}{cccc}
1 & 0 & 3 & 5 \\
6 & 9 & 3 & 1 \\
0 & 2 & 7 & 0 \\
2 & 8 & 5 & 1 \\
\end{array}
\]

\[
\begin{array}{cccc}
3 & 7 & 8 & 1 \\
5 & 9 & 4 & 0 \\
2 & 3 & 7 & 8 \\
7 & 2 & 7 & 0 \\
\end{array}
\]

\[
\begin{array}{cccc}
0 & 0 & 0 & 1 \\
1 & 1 & 0 & 1 \\
0 & 0 & 1 & 0 \\
0 & 1 & 0 & 1 \\
\end{array}
\]

\((A \geq B) = C\)
Boolean Operators in Map Algebra

- The AND operation requires that the value of cells in both input layers be equal to 1 for the output to have a value of 1:

\[
\begin{array}{ccc}
0 & 1 & 1 \\
0 & 0 & 1 \\
1 & 0 & 1 \\
\end{array}
\]

\[
\begin{array}{ccc}
0 & 0 & 0 \\
1 & 1 & 1 \\
0 & 0 & 1 \\
\end{array}
\]

= 

- The OR operation requires that the value of a cells in either input layer be equal to 1 for the output to have a value of 1:

\[
\begin{array}{ccc}
0 & 1 & 1 \\
0 & 0 & 1 \\
1 & 0 & 1 \\
\end{array}
\]

\[
\begin{array}{ccc}
0 & 0 & 0 \\
1 & 1 & 1 \\
0 & 0 & 1 \\
\end{array}
\]

= 

David Tenenbaum – EEOS 465 / 627 – UMass Boston
Arithmetic Operators in Map Algebra

• We can extend this concept from Boolean logic to algebra

• Map algebra:
  • Treats input layers as numeric inputs to mathematical operations (each layer is a separate numeric input)
  • The result of the operation on the inputs is calculated on a cell-by-cell basis

• This allows for complex overlay analyses that can use as many input layers and operations as necessary

• A common application of this approach is suitability analysis where multiple input layers determine suitable sites for a desired purpose by scoring cells in the input layers according to their effect on suitability and combining them, often weighting layers based on their importance
Simple Arithmetic Operations

**Summation**

\[
\begin{array}{ccc}
0 & 1 & 1 \\
0 & 0 & 1 \\
1 & 0 & 1 \\
\end{array}
\]  
\[+\]
\[
\begin{array}{ccc}
0 & 0 & 0 \\
1 & 1 & 1 \\
0 & 0 & 1 \\
\end{array}
\]  
\[=\]
\[
\begin{array}{ccc}
0 & 1 & 1 \\
1 & 1 & 2 \\
1 & 0 & 2 \\
\end{array}
\]

**Multiplication**

\[
\begin{array}{ccc}
0 & 1 & 1 \\
0 & 0 & 1 \\
1 & 0 & 1 \\
\end{array}
\]  
\[\times\]
\[
\begin{array}{ccc}
0 & 0 & 0 \\
1 & 1 & 1 \\
0 & 0 & 1 \\
\end{array}
\]  
\[=\]
\[
\begin{array}{ccc}
0 & 0 & 0 \\
0 & 0 & 1 \\
0 & 0 & 1 \\
\end{array}
\]

**Summation of more than two layers**

\[
\begin{array}{ccc}
0 & 1 & 1 \\
0 & 0 & 1 \\
1 & 0 & 1 \\
\end{array}
\]  
\[+\]
\[
\begin{array}{ccc}
0 & 0 & 0 \\
1 & 1 & 1 \\
0 & 0 & 1 \\
\end{array}
\]  
\[+\]
\[
\begin{array}{ccc}
0 & 0 & 0 \\
1 & 1 & 1 \\
0 & 0 & 1 \\
\end{array}
\]  
\[=\]
\[
\begin{array}{ccc}
0 & 1 & 1 \\
2 & 2 & 3 \\
1 & 0 & 3 \\
\end{array}
\]

David Tenenbaum – EEOS 465 / 627 – UMass Boston
Raster (Image) Difference

The difference between two layers

\[
\begin{array}{ccc}
5 & 4 & 3 \\
6 & 5 & 6 \\
7 & 1 & 5 \\
\end{array}
\quad - 
\quad
\begin{array}{ccc}
3 & 5 & 6 \\
1 & 4 & 5 \\
3 & 2 & 7 \\
\end{array}
\quad = 
\quad
\begin{array}{ccc}
2 & -1 & -3 \\
5 & 1 & 1 \\
4 & -1 & -2 \\
\end{array}
\]

- An application of taking the differences between layers is change detection:
  - Suppose we have two raster layers that each show a map of the same phenomenon at a particular location, and each was generated at a different point in time
  - By taking the difference between the layers, we can detect changes in that phenomenon over that interval of time

- Question: How can the locations where changes have occurred be identified using the difference layer?
Raster (Image) Division

Question: **Can we** perform the following operation? Are there any **circumstances** where we **cannot** perform this operation? Why or why not?

\[
\begin{array}{ccc}
\text{Raster Image} & \div & \text{Raster Image} \\
\hline
\text{Raster Image} & \text{Raster Image} & \text{Raster Image}
\end{array}
\]
More Complex Operations

Linear Transformation

\[
\begin{bmatrix}
1 & 2 & 4 \\
3 & 2 & 1 \\
5 & 3 & 2
\end{bmatrix}
+ \begin{bmatrix}
1 & 0 & 0 \\
5 & 1 & 1 \\
2 & 0 & 1
\end{bmatrix}
+ \begin{bmatrix}
0 & 0 & 0 \\
1 & 1 & 1 \\
0 & 0 & 1
\end{bmatrix}
= \begin{bmatrix}
\end{bmatrix}
\]

• We can multiply layers by constants (such as a, b, and c in the example above) before summation
• This could applied in the context of computing the results of a regression model (e.g. output \( y = a \times x_1 + b \times x_2 + c \times x_3 \)) using raster layers
• Another application is suitability analysis, where individual input layers might be various criteria, and the constants a, b, and c determine the weights associated with those criteria
Seven Interfaces for Spatial Analyst
The Raster Calculator

• Use to enter **map algebra expressions**:
  – Build with **buttons or type** into expression box
Expression Syntax Rules

• **Delimit** operators and objects with **blanks**:

  Wrong: Layer+Layer2+Layer3  
  Right: Layer1 Layer2 Layer3

• Operators evaluated by **precedence** level:

  Layer1 + Layer2 * Layer3

• **Override** operator precedence with **parentheses**:

  (Layer1 + Layer2) * Layer3

• **Nested** parenthetical expressions evaluate **first**:

  ((Layer1 + Layer2) / 4) - Layer3
Expression Results

• Expressions return grids, vector data, tables, etc.
  – Depends on functions used
  – Most return GRIDs

• Temporary or permanent?
  – For returned grids only
  – Temporary GRID if unnamed
  – Permanent GRID if named

• Layers added to ArcMap:
  – Table of Contents
Cell coincidence

\[
\text{TOTCOST}(4,2) = \text{COST1}(4,2) + \text{COST2}(4,2)
\]

\[26 = 20 + 6\]
Resampling

- Automatically applied when combining rasters

- Map Algebra operation

- Output cell center identifies the input value

- Different cell size
### Expression Evaluation (when Resampling)

- Expressions are processed as follows:

1. **Define** empty output GRID based on the analysis environment.
2. **Position** to the **next output cell** (start at row 0, column 0).
3. **Resample input raster(s)** to determine corresponding cell values.
4. **Evaluate** the expression and **write the result** to the output cell.
5. **Repeat** steps 2 - 4 for **all output cells**.

![Resampling Diagram](image)
User attributes in expressions

- You may use numeric VAT fields in expressions
- Reference with `[Layer].field` notation

<table>
<thead>
<tr>
<th>Vegetation.VAT</th>
<th>Soil.VAT</th>
</tr>
</thead>
<tbody>
<tr>
<td>Value</td>
<td>Count</td>
</tr>
<tr>
<td>101</td>
<td>2450</td>
</tr>
<tr>
<td>201</td>
<td>65780</td>
</tr>
<tr>
<td>301</td>
<td>32187</td>
</tr>
<tr>
<td>401</td>
<td>5433</td>
</tr>
</tbody>
</table>

- `[Layer]` alone is assumed to be `[Layer].Value`

- You may join tables to grids VAT file
  - Use joined fields for symbology, selection
  - Cannot use in Map Algebra expressions
Special cell values in Map Algebra

- Logical: Non-zero values are *True*, zero is *False*

- NoData: If any input is *NoData*, the output is *NoData*
Examples of Operators

- **Change detection**
  - Landuse90
  - Landuse95
  - LuDiff1
  - LuDiff2

- **Update**
  - NewMall
  - Landuse90
  - LuUpdated

- **Operators**
  - EQ
  - DIFF
  - OVER

David Tenenbaum – EEOS 465 / 627 – UMass Boston
Seven Interfaces for Spatial Analyst
Multi Output Map Algebra Tool

Constants, functions, variables OK

Cone = (40.6 / (4 * PI * GwTrans)) * Ln(Sqr(WellDist))
Head = GwHead + (Cone - 32.118427)

Multiple expressions
Must assign output Grid
Outputs not added to the Table of Contents

Grid, coverage, valid table, file, inputs (no layers)
Same spatial reference

Syntax for operators, functions, commands
Single Output Map Algebra Tool

One expression

No constants or variables

Any raster/layer input
Any spatial reference

Set output

Syntax for operators, functions

Set inputs for ModelBuilder parameters (optional)

Outputs are added to the Table of Contents

(40.6 / (4 * 3.14159 * GwTrans)) * Ln(Sqr(WellDist))

C:\RasterData\Cone

GwTrans
WellDist
Map Algebra vs. ModelBuilder

IdealFarm = SlopeSuit + SoilSuit + RainSuit
Exercise 3: Building a Raster Database

- EXERCISE 3A: BUILD A RASTER DATABASE
- EXERCISE 3B: GEOREFERENCE A RASTER
Building a Raster Database

• Designing a raster database
  – Poor design $\rightarrow$ consequences, costs in the future

• Need to evaluate needs and plan accordingly, before building a GIS database

• Decisions on the type of data to store, how to use the data, and on going maintenance
  – Considering these issues beforehand will help improve your design decisions
ESRI Data Models

• Data models = schema *templates*
• Templates for *implementing* GIS projects
• **Speed up** development time
• Available for *many industries*
• Provided as *templates* to create
• Personal or file *geodatabases*
Rasters in ArcCatalog

- Edit spatial reference
- Create metadata
- Build pyramids and statistics
Geoprocessing: Raster Management Tools

- General raster database management tools
  - Copy, paste, delete, calculate statistics, set spatial reference
- Data organization/preparation
  - Mosaicking
  - Raster catalogs
- Raster data storage
  - Pyramids
  - GDB technology
Geodatabase Raster Datasets

- **A single raster** in a geodatabase
  - May load many rasters into one raster dataset (mosaic)
  - Good for analysis and mapping
  - Seamless
  - Fast display at any scale

- **Personal geodatabase** format
  - Rasters converted to IMG format, stored in hidden .idb folder

- **File geodatabase** format
  - Rasters converted to FGDBR format, stored in the GDB folder

- **ArcSDE geodatabase** format
  - Rasters converted to ArcSDE raster format, stored in RDBMS

- **All formats** preserve GRID attribute fields
Geodatabase Raster Catalogs

- A **collection** of raster datasets
  - Behave as one, but are **stored separately** (rows in a table)
  - May overlap, have gaps, different cell sizes, bit depths
  - Must have **same spatial reference**
  - Good for archives, display, and mapping

- **Personal geodatabase** format
  - **Managed**: Converted, stored in .idb folder
  - **Unmanaged**: Referenced by path name

- **File geodatabase** format
  - Can be managed or unmanaged

- **ArcSDE geodatabase** format
  - Necessarily managed
Merging Rasters

• Combine **multiple rasters** into one
  – Six **methods** to handle **overlapping** areas:
General Raster Properties: Pyramids

- **Reduced resolution copies** of original raster
  - Pixel size **doubles** at each level
- **Improves query/display performance**
  - Returns **best resolution** for screen display
  - Returns about **same number of pixels as scale changes**
- **Personal GDB**
  - Stores pyramids in **RRD file**
- **File GDB**
  - Stores pyramids in **GDB folder**
- **ArcSDE GDB**
  - Stores pyramids **in tables**
Rasters in a Personal Geodatabase

• Stores a reference to external, file-based rasters
  – Microsoft Access MDB file is limited to 2 GB total size
  – Provides centralized access to rasters

• Stores raster datasets
  – A single raster
  – Best for data
  – Can use in analysis

• Stores raster catalogs
  – A collection of rasters
  – Best for imagery archives
  – Cannot use in analysis
Rasters in a File Geodatabase

- Stores **raster catalogs** and **raster datasets**
  - Up to one terabyte for each raster dataset or raster catalog
  - Provides **centralized access** to rasters
- **Useful for:**
  - A single user and small work groups
  - Some readers and one writer
Rasters in an ArcSDE Geodatabase

- ArcSDE **subdivides** a raster into **blocks** for storage
  - Size set by user
  - Automatic and required
  - Invisible to end users
- The **raster is a table**; a **block is a row** in the table
- Provides **faster access** to data
  - ArcSDE returns blocks for visible area
  - Improves display performance
Next Topic:

Raster Analysis and Functions