### Spatial Analysis of Raster Data

<table>
<thead>
<tr>
<th></th>
<th>0</th>
<th>0</th>
<th>1</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td></td>
</tr>
</tbody>
</table>

+ |   | 2 | 4 | 4 | 4 |
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>4</td>
<td>2</td>
<td>4</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>4</td>
<td>4</td>
<td>2</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

= |   | 2 | 4 | 5 | 5 |
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>2</td>
<td>5</td>
<td>5</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>4</td>
<td>3</td>
<td>5</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>5</td>
<td>5</td>
<td>3</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

0 = shale  
1 = limestone  
2 = fault  
3 = Fault in shale  
4 = no Fault, shale  
5 = no Fault, limestone

### What is Spatial Analysis?

- **Spatial Query:**  
  "Where is...?"

- **Spatial Analysis (e.g. suitability analysis):**  
  "Where is the best place for...?"  
  "What is the least costly path between...?"

- "a set of methods with results that change when the location of objects being analysed changes"  
  –the spatial aspect of this form of analysis sets it apart

### Why Raster?

- Conceptually simple, easy to implement
- Well-suited for surface- and field-related phenomena (e.g. elevation, gravity, rainfall, etc.) and for discrete features
- Wide availability of data-sets; all remotely sensed data of this sort
- *De facto* standard approach – oldest, most widely implemented, mature, widest suite of tools and software
- Best suited for “Where” rather than “What” questions

### Where do rasters come from?

- Converted vector files, e.g. shapefiles, coverages – Tools available

- Created from interpolations of point values – Tools available

- Directly from raster sources; remotely sensed data, DEMs, meteorological measurements, etc.
What gets stored?

- Cell values may be:
  - Nominal – integers are attribute codes (tags)
  - Ordinal – integers are ranks
  - Ratio – Ratio of values makes sense, e.g. 300 m elevation is twice as high as 150 m; magnitude of ratio has some physical meaning

Nominal Raster – e.g. Geologic Map

- Each raster cells contains a value of 1 to 18:
  - 1 = water
  - 4 = Huckleberry R. Tuffs
  - 12 = Plio.-Pleist. Rhyolite etc.

What gets stored?

- Cell values may be:
  - Nominal – integers are attribute codes (tags). Though numbers, they are dimensionless and without scale.
  - Numbers as qualitative descriptors.
  - Mathematical operations on cell values are not meaningful as a measure of scalar magnitudes.

- Ordinal – integers are ranks
  - e.g. 1=excellent, 2=good, 3=poor; 1=low, 2=medium, 3=high
  - Ranking scheme used for hazard rating, density measures, etc.
  - Mathematical & most statistical operations meaningless; Median might be useful measure of central value
Ordinal Raster – e.g. Erosion Ranking

- Each raster cell contains a value of 1-12
- Yellow = 12 = Most Erosive
- Blue = 1 = Least Erosive

Ratio Raster – e.g. Elevation above MSL

- Each raster cell contains a value of 1544-3578 (meters)
- White = 3578 m
- Pale Blue = 1544 m

What gets stored?

- Each raster cell contains a value of 1-12
- Yellow = 12 = Most Erosive
- Blue = 1 = Least Erosive

What gets stored?

- Cell values may be:
  - Ratio – data are organized along a continuum and numbers do have an absolute meaning
  - e.g. lengths, volumes, heights, concentrations, etc.
  - Multiplication/Division, Subtraction/Addition make sense for arriving at meaningful new cell values.
What Gets Stored?

- Raster data model:
  - Simple Raster – Non-binary, one nominal value per cell
    - integer is a code for categorical attribute e.g. limestone = 1, sandstone = 2, mudstone = 3
    - requires one raster per theme

<table>
<thead>
<tr>
<th></th>
<th>Value</th>
<th>Count</th>
<th>Type</th>
<th>Porosity</th>
<th>Cement</th>
</tr>
</thead>
<tbody>
<tr>
<td>VAT for Geology raster</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>Value</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Count</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Type</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Type</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

What Are the Tools & Techniques?

- Map Algebra
  - Map Algebra employing:
    - Raster Operators
    - Raster Functions
  - Map Algebra takes raster(s) as input and return a result as a new raster.

Map Algebra Operators

1. Arithmetic
   - +, -, *, /; for pairs of rasters
   - Trigonometric, Log, Exponential, Powers for single rasters
Example: Raster Overlay

- E.g. “Find wells within limestone”

- Point-in-Polygon Query

0 = shale
1 = limestone
2 = well
3 = well in limestone
4 = no well
5 = no well, limestone

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

Map Algebra Operators

2. Relational

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

- Compare two rasters. Create a new raster such that if condition is false, return 0, if true 1.

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

Boolean Selections; Or, And, Not

From Bolstad, 4th edition

Map Algebra Operators

3. Boolean

- And, Or, Not

- And – both true
- Or – either true
- Not – switches true for false

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

From Bolstad, 4th edition
Map Algebra Operators

4) Combinatorial
   - Assign value in new raster on basis of the combination of values in compared rasters.

<table>
<thead>
<tr>
<th>R1</th>
<th>R2</th>
<th>Out</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>2</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>2</td>
<td>4</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>2</td>
</tr>
</tbody>
</table>

- Combinatorial Operator (COR):

```
COR = R1 \times R2
```

```
1 1 1
1 4 4
1 3 3
1 2 3
```

Output:
```
1 0 0
0 1 3
0 2 4
```

5) Accumulative
   - \(+=, \times=, -=\)
   - Add, subtract, multiply, raster values in specific order.

```
  1 1 1
  1 0 3
  1 3 3
+= Value = 13
```

6) Assignment
   - \(=\)
   - Assign all cells in a new raster a value by performing operation on old raster.

```
  1 1 1
  1 0 3
  1 3 3
* 5 =
```

```
  5 5 5
  5 0 15
  5 15 15
```
Raster Functions

- Higher-order operations built up of operators just listed; relationship of input to output cells:
  - Local – cell-to-cell functions: 1 input cell per output cell
  - Focal – by-neighborhood functions
  - Global – entire raster gives
  - Special Types

Local Functions: Reclassification

- Make new raster by performing function on old.
  - Nominal values reclassed as 0 or 1 (=binary masking) e.g. Boolean operators
  - Reduce range or number of values
    - floating point to integer values
  - Change measurement scale to weight values; convert nominal values to rank (ordinal or ratio values).

Reclassification – Binary Masking

- Beginning Raster:

```
S S S S S S S S S S S
S S S S S S S S S S S
S S S S S S S S S S S
S S S S S S S S S S S
2 2 2 2 2 2 2 2 2 2
S S S S S S S S S S S
S S S S S S S S S S S
S S S S S S S S S S S
```

- Simplify to raster with granite and non-granite cells to produce binary raster.

0 = limestone
5 = sandstone
8 = granite
Local Functions: Reclassification

- Binary Raster – composed of 0 and 1
  
  | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
  | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
  | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
  | 0 | 0 | 0 | 0 | 0 | 1 | 1 |
  | 0 | 0 | 0 | 0 | 0 | 1 | 1 |
  | 0 | 0 | 0 | 0 | 0 | 1 | 1 |
  | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
  | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

- Replaced nominal 2 and 5 by 0; 8 by 1.
- Simplified raster can be saved and used for further analysis

Reclassification - Weighting

- Reclassify to assign weighting factor for further analysis; nominal values become ordinal values for later calculation

<table>
<thead>
<tr>
<th>Lithology</th>
<th>Old Value</th>
<th>Weight</th>
<th>New Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Limestone</td>
<td>2</td>
<td>10</td>
<td>20</td>
</tr>
<tr>
<td>Sandstone</td>
<td>5</td>
<td>2</td>
<td>10</td>
</tr>
<tr>
<td>Granite</td>
<td>8</td>
<td>5</td>
<td>40</td>
</tr>
</tbody>
</table>

Granite is weighted 4x sandstone and 2x limestone

Local Functions: Raster Overlay

- All entities represented by cells:
  - point = single cell
  - line = chain of cells
  - polygon = group of cells
- Nominal values identify a related group of cells as an entity
- Rasters of continuous variables (e.g. rainfall, temp., elevation) have cells with ratio values

Local Functions: Raster Overlay

- Compare cell value among layers by Map Algebra
  - generate new raster as sum, difference, product, etc. of cells within two layers

| 0 | 0 | 1 | 1 | 2 | 0 | 0 | 0 |
| 0 | 0 | 1 | 1 | 0 | 2 | 0 | 0 |
| 1 | 0 | 1 | 1 | 0 | 0 | 2 | 0 |
| 1 | 1 | 1 | 1 | 0 | 0 | 0 | 2 |

+ =

| 2 | 0 | 1 | 1 |
| 0 | 2 | 1 | 1 |
| 1 | 0 | 3 | 1 |
| 1 | 1 | 1 | 3 |
Local Functions: Raster Overlay

- E.g. “Find faults cutting limestone OR shale”

<table>
<thead>
<tr>
<th>0 = shale</th>
<th>2 = fault</th>
<th>4 = no fault</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 0 1 1</td>
<td>4 4 4 4</td>
<td>2 4 5 5</td>
</tr>
<tr>
<td>0 0 1 1</td>
<td>4 2 4 4</td>
<td>4 2 5 5</td>
</tr>
<tr>
<td>1 0 1 1</td>
<td>4 4 2 4</td>
<td>5 4 3 5</td>
</tr>
<tr>
<td>1 1 1 1</td>
<td>4 4 4 2</td>
<td>5 5 5 3</td>
</tr>
</tbody>
</table>

Line-in-Polygon Query

- Operators include nearly all previously listed:
  - Arithmetic
  - Relational
  - Logical
  - etc.

Focal Functions

- Neighborhood functions: uses values in adjacent cells to return values for new raster.
- Used for:
  - Aggregation
  - Filtering
  - Computing slope and aspect

Aggregation

- “Down-sampling” – combining cells (average, central cell, median) to produce raster with fewer cells.
Focal Functions: Computing Slope

- Use 8 neighboring cells to compute slope of cell #5.
  \[ \text{Rise/run} = \tan (\text{slope}) \]

- Find slope in x direction
  \[ b = \tan (\text{slope}_x) = \frac{z_3 + z_6 + z_9 - z_1 - z_4 - z_7}{8D} \]

- Find slope in y direction
  \[ c = \tan (\text{slope}_y) = \frac{z_1 + z_2 + z_3 - z_7 - z_8 - z_9}{8D} \]

- Find slope in steepest direction
  \[ \tan (\text{slope}) = \left( b^2 + c^2 \right)^{1/2} \]

Focal Functions: Filtering

- Filtering – assign new value to cell on basis of neighboring cells. Save as new raster.
  - Define filter window as a group of cells ("kernal") around a target cell; size and shape can be specified.
  - Step window across entire raster, calculating new value for center of filter on basis of neighboring values within the filter and filter rule.

Neighborhood Functions: Filtering

- Rule – replace target cell (in center) with mean value encountered in filter
  - Define square filter of 3x3 cells
    - Target 1: mean = 18/9 = 2
      - replace target with 2
    - Target 2: mean = 15/9 = ~1.7
      - replace target with 2

Filtering effective for:
- removing noise
- revealing linear trends

Neighborhood (Proximity) Functions

- Buffering – calculate buffer zone based on proximity. Save as new raster.
  - Cell value of new raster is a measure of distance via proximity.

Original raster

<table>
<thead>
<tr>
<th>2</th>
<th>4</th>
<th>4</th>
<th>4</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>2</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>4</td>
<td>4</td>
<td>2</td>
<td>4</td>
</tr>
<tr>
<td>4</td>
<td>4</td>
<td>4</td>
<td>2</td>
</tr>
</tbody>
</table>

3 m Buffer raster

<table>
<thead>
<tr>
<th>4</th>
<th>4</th>
<th>4</th>
<th>4</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>4.5</td>
<td>5.0</td>
<td>5.7</td>
</tr>
<tr>
<td>3</td>
<td>3.6</td>
<td>4.2</td>
<td>5.0</td>
</tr>
<tr>
<td>2</td>
<td>2.8</td>
<td>3.6</td>
<td>4.5</td>
</tr>
<tr>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
</tr>
</tbody>
</table>