LOCAL OPERATIONS

Unit: VI

Semester: II

Paper Code: GIS 07
Name of Paper: Geographic Information System

PG Diploma in RS & GIS

Dr. SHYAMA PRASAD MUKHERJEE UNIVERSITY, RANCHI

2.2 LOCAL OPERATIONS

A local operation involves the production of an output value as a function of the value(s) at the corresponding locations in the input layer(s). These operations can be considered point operations when performed on raster data, i.e. they operate on a pixel and its matching pixel position in other layers, as opposed to groups of neighbouring pixels. They can be grouped into those which derive statistics from multiple input layers (e.g. mean, median, minority), those which combine multiple input layers, those which identify values that satisfy specified criteria or the number of occurrences that satisfy specified criteria (e.g. greater than or less than), or those which identify the position in an input list that satisfies a specified criterion. All types of operator previously mentioned can be used in this context. Commonly they are subdivided according to the number of input layers involved at the start of the process. They include primary operations where nothing exists at the start, to n-ary operations where n layers may be involved; they are summarized in Table 1.3 and illustrated in Fig. 1.3.

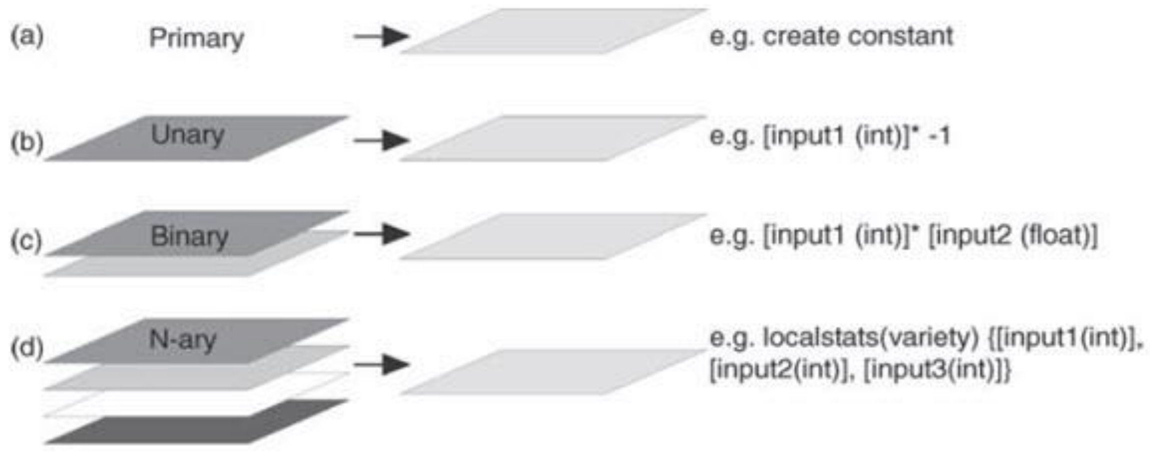

(a)
Primary
e.g. create constant
(b)
Unary
e.g. [input1 (int)]* -1
(c)
Binary
e.g. [input1 (int)]* [input2 (float)]
(d)
N-ary
e.g. localstats(variety) {[input1(int)], [input2(int)], [input3(int)]}

Fig. 2.3. Classifying map algebra operations in terms of the number of input layers and some examples**.**

2.2.1 Primary operations

This description refers primarily to operations used to generate a layer, conceptually from nothing, for example the creation of a raster of constant value, or containing randomly generated numbers, such as could be used to test for error propagation through some analysis. An output pixel size, extent, data type and output DN value (either constant or random between set limits) must be specified for the creation of such a new layer.

1.2.2 Unary operations

These operations act on one layer to produce a new output layer and they include tasks such as rescaling, negation, comparison with other numbers, application of functions and reclassification. Rescaling is especially useful in preparation for multi-criteria analysis where all the input layers should have consistent units and value range: for instance, in converting from byte data, with 0 to 255 value range, to a percentage scale (0-100) or a range of between 0 and 1, and vice versa. Negation is used in a similar context, in modifying the value range of a dataset from being entirely positive to entirely negative and vice versa. Comparisons create feature grids: the places where the comparison is true can be considered features on the earth's surface. They map the regions where a logical condition (the comparison) holds. These could be regions where, say, ozone concentrations exceed a threshold, ocean depths are below a certain target, or land use equals a given code. Mathematical functions are useful for changing the visualization of a grid. An equal interval classification using the square roots of the values will differ from an equal interval classification of the values themselves, for instance. Functions are also important as intermediate steps in many models. Reclassification is especially significant in data preparation for spatial analysis, and so deserves rather more in-depth description, but all these activities can be and are commonly carried out in image processing systems.

To illustrate different applications succinctly, suppose that three grids appear in the current view: "Integer" is an integer grid, "Float" is a floating-point grid, and "Indicator" is an integer grid containing only 0, 1, and NoData values. A value of 0 can be interpreted as a logical "false" and a value of 1 as a logical "true". In practice, of course, we will replace these names by the names of our themes.

- **Rescale** a grid: that is, Multiply all its values by a constant value.

[Float] * 3.1415927 Multiply all values by Pi

[Integer].Float * (39.37/12) Convert meters to feet

[Integer] * (-1) Negate all values

Not [Indicator] Negate all logical values: 0 becomes 1, 1 becomes 0

- **Compare** a grid to a constant value. The result of a comparison is true, 0 where the comparison is false, and NoData where the original value is NoData

[Float] < 1 Returns 1 where values are less than 1, otherwise returns 0

[Integer] = 0 Converts all zeros to ones and all other values to zeros

- Apply a mathematical (or logical) **function** to a grid, cell by cell.

[Float].Cos Computes the cosine of each value (interpreted as radians).

[Float].Int Rounds all values and converts the result to an integer grid.

[Float].Sqrt Computes the square root of each value. Negative values return NoData (because the square root is not defined for negative values).

[Float].IsNull Returns 1 at all cells with NoData values, otherwise returns 0.

1.2.2.1 Reclassification

This involves the process of reassigning a value, a range of values, or a list of values in a raster to new output values, in a new output raster. If one class (or group or range of classes) is more interesting to us than the other classes, its original values can be assigned a specific value and all the others can be changed into a different (background) value. This involves the creation of a discrete raster from either a continuous one or another discrete raster. Reclassification can be applied to both vector and raster objects.

In the case of discrete raster data, a reclassification may be required to produce consistent units among a set of input raster images, in which case a one-to-one value change may be applied. The output raster would look no different, spatially, from the input, having the same number of classes, but the values would have changed.

Different classes or types of feature may be reclassified according to some criteria that are important to the overall analysis. During the reclassification process, weighting can be applied to the output values to give additional emphasis to the significant classes, and at the same time reducing the significance of other classes.

The example in Fig. 1.4a shows a discrete raster representation of a geological map in which nine lithological units are coded with values 1 to 9 and labelled for the purposes of presentation, according to their name, rock type and ages. For the purposes of some analysis it may be necessary to simplify this lithological information, for example according to the broad ages of the units, PreCambrian, Palaeozoic and Mesozoic, for instance. The result of such a simplification is shown in Fig. 1.4c; now the map has only three classes and it can be seen that the older rocks (Precambrian and Palaeozoic) are clustered in the south-western part of the area, with the younger rocks (Mesozoic) forming the majority of the area as an envelope around the older rocks. So the simplification of the seemingly quite complex lithological information shown in Fig. 1.4a has revealed spatial patterns in that information which are of significance and which were not immediately apparent beforehand. Fig. 1.4d shows a second reclassification of the original lithological map, this time on the basis of relative permeability. The information is again simplified by reducing the number of classes to two, impermeable and permeable. Such a map might form a useful intermediary layer in an exercise to select land suitable for waste disposal but also illustrates that subjective judgements are involved at the early stage of data preparation. In the very act of simplifying information, we introduce bias and, strictly speaking, error into the analysis. We also have to accept the assumptions that the original classes are homogeneous and true representations everywhere on the map, which they may not be. In reality there is almost certainly heterogeneity within classes and the boundaries between the classes may not actually be as rigid as our classified map suggests.

Continuous raster data can also be reclassified in the same way. The image in Fig. 1.5a shows a DEM of the same area with values ranging between 37 and 277, representing elevation in meters above sea level. Reclassification of this dataset into three classes of equal interval to show areas of low, medium and high altitude produces the simplified image in Fig. 1.5b. Comparison with Fig. 1.5b shows that the areas of high elevation coincide with the areas where older rocks exist at the surface in the south-west of the area, again revealing spatial patterns not immediately evident in the original image. Reclassification of the DEM into three classes, this time with the classes defined according to the natural breaks in the image histogram (shown in Fig. 1.6), produces a slightly different result, Fig. 1.5c. The high-elevation areas are again in the south-west but the shape and distribution of those areas are different. This demonstrates several things. Firstly, that very different results can be produced when we simplify data so that (and secondly) we should be careful in doing so, and, thirdly, that the use of the image histogram is fundamental to the understanding of and sensible use of reclassification of continuous raster data.

Reclassification forms a very basic but important part of spatial analysis, in the preparation of data layers for combination, in the simplification of layer information and especially when the layers have dissimilar value ranges. Reclassification is one of several methods of producing a common range among input data layers that hold values on different measurement scales.

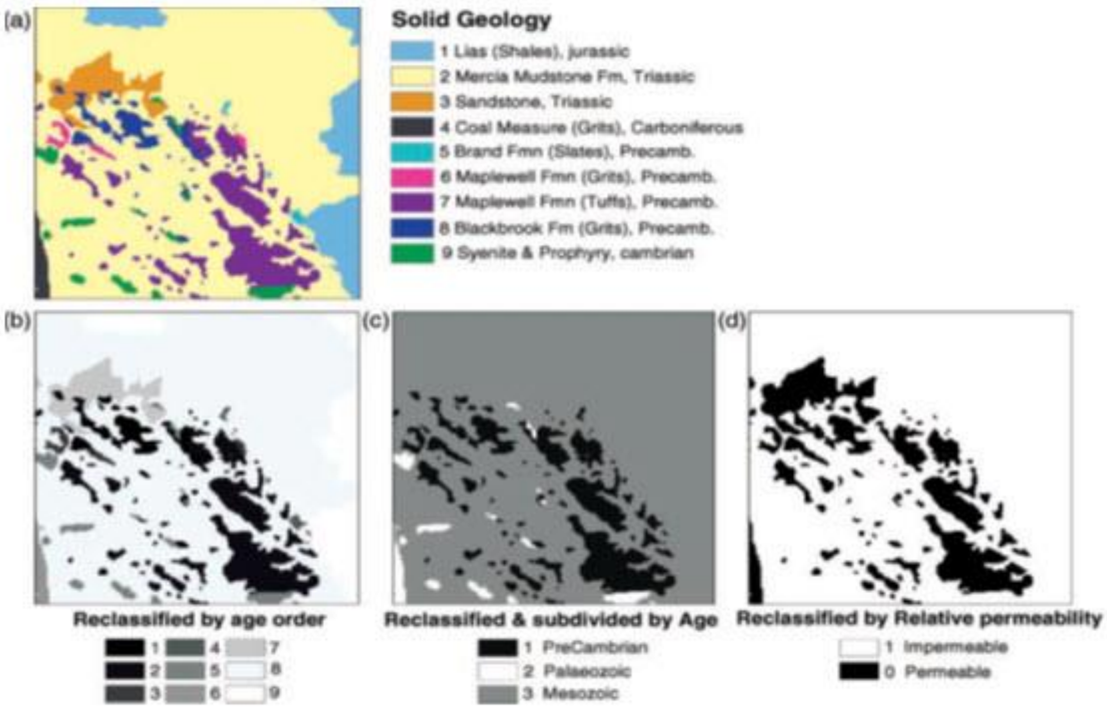

(a)
Solid Geology
1 Lias (Shales), jurassic
2 Mercia Mudstone Fm, Triassic
3 Sandstone, Triassic
4 Coal Measure (Grits), Carboniferous
5 Brand Fmn (Slates), Precamb.
6 Maplewell Fmn (Grits), Precamb.
7 Maplewell Fmn (Tuffs), Precamb.
8 Blackbrook Fm (Grits), Precamb.
9 Syenite & Prophyry, cambrian
(b)
(c)
(d)
Reclassified by age order
1
2
3
4
5
6
7
8
9
Reclassified & subdivided by Age
1 PreCambrian
2 Palaeozoic
3 Mesozoic
Reclassified by Relative permeability
1 Impermeable
0 Permeable

Fig. 1.4 (a) Discrete rastFig.er representation of a geological map, with nine classes representing different lithologies; (b) one-to-one reclassification by age order (1 representing the oldest, 9 the youngest); (c) a reclassified and simplified version where the lithological classes have been grouped and recoded into three broad age categories (Pre-Cambrian, Palaeozoic and Mesozoic); (d) a second reclassified version where the lithologies have been grouped according to their relative permeability, with 1 representing impermeable rocks and 0 permeable; such an image could be used as a mask. (Source: Liu and Mason, 2009)

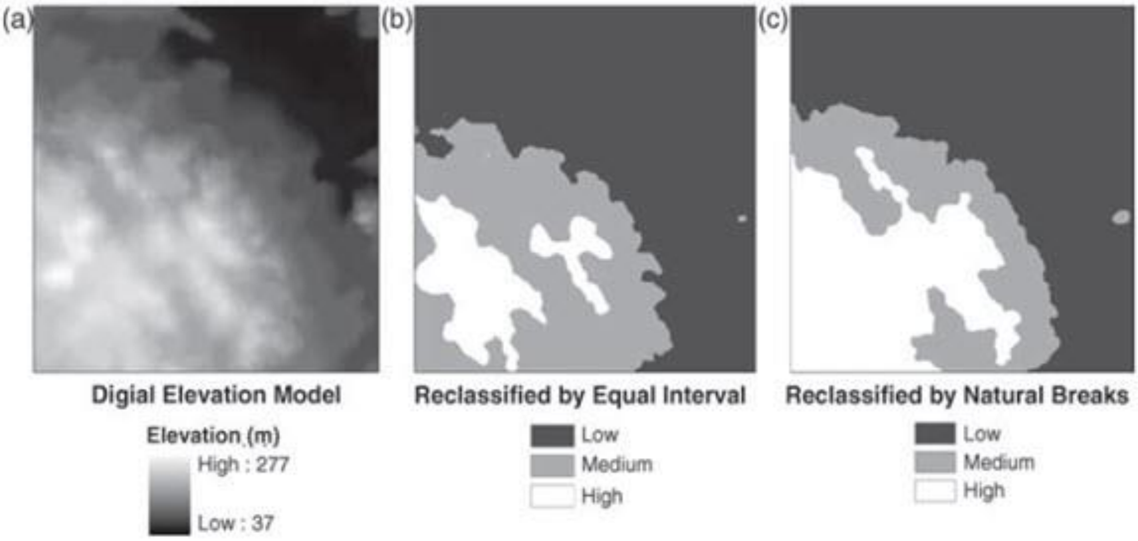


Fig. 1.5. (a) A DEM; (b) a DEM reclassified into three equal interval classes; and (c) a DEM reclassified into three classes by natural breaks in the histogram (shown in Fig. 1.6).

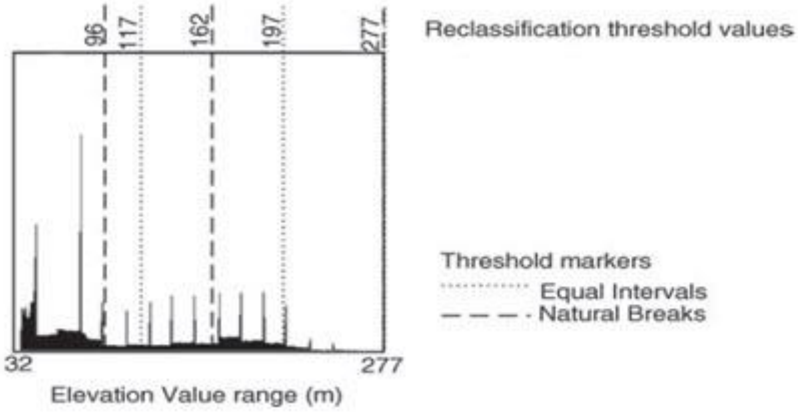


Fig. 1.6. Image histogram of the DEM shown in Fig. 1.5a and the positions of the reclassification thresholds set by equal interval and natural break methods (shown in Fig. 1.5b and c, respectively).

1.2.3 Binary operations

Binary numeric operations act on ordered pairs of numbers. Likewise, binary grid operations act on the pairs of numbers obtained in each set of matching cells. The resulting grid is defined only where the two input grids overlap.

Suppose there are several floating-point grids represented by themes named "Float", "Float1", "Float2", and so on; with a similar supposition for integer and logical grids.

- **Mathematical operators**

[Float] + [Integer] Converts the values in [Integer] to floats, then performs the additions.

- **Logical operators**

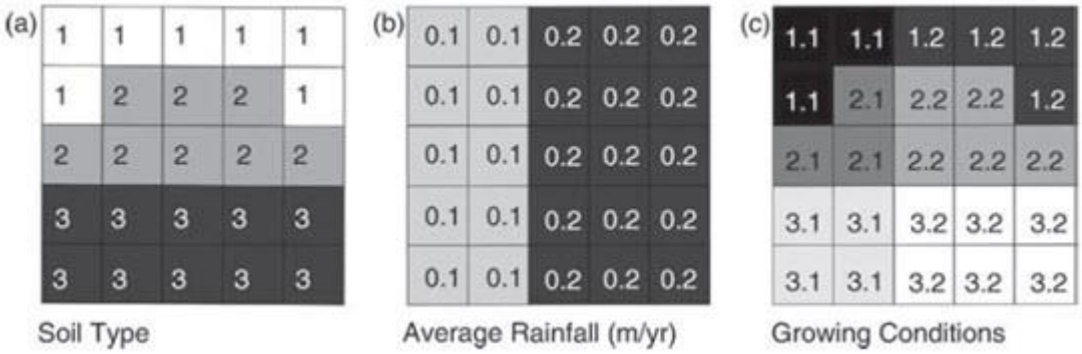
[Float1] < [Float2] Returns 1 in each cell where [Float1]'s value is less than [Float2]'s value; otherwise, returns 0.

[Indicator1] And [Indicator2] Returns 1 where both values are nonzero otherwise returns 0.

This description refers to operations in which there are two input layers, leading to the production of a single output layer. Overlay refers to the combination of more than one layer of data, to create one new layer. The example shown in Fig. 1.7 illustrates how a layer representing average rainfall, and another representing soil type, can be combined to produce a simple, qualitative map showing optimum growing conditions for a particular crop. Such operations are equivalent to the application of formulae to multiband images, to generate ratios, differences and other inter-band indices and as mentioned in relation to point operations on multi-spectral images, it is important to consider the value ranges of the input bands or layers, when combining their values arithmetically in some way. Just as image differencing requires some form of stretch applied to each input layer, to ensure that the real meaning of the differencing process is revealed in the output, here we should do the same. Either the inputs must be scaled to the same value range, or if the inputs represent values on an absolute measurement scale then those scales should have the same units.

The example shown in Fig. 1.7 represents two inputs with relative values on arbitrary nominal or ordinal (Fig. 1.7a) and interval (Fig. 1.7b) scales. The resultant values are also given on an interval scale and this is acceptable providing the range of potential output values is understood, having first understood the value ranges of the inputs, since they may mean nothing outside the scope of this simple exercise.

Another example could be the combination of two rasters as part of a cost-weighted analysis and possibly as part of a wider least cost pathway exercise. The two input rasters may represent measures of cost, as produced through reclassification of, for instance, slope angle and land value, cost here being a measure of friction or the real cost of moving or operating across the area in question. These two cost rasters are then aggregated or summed to produce an output representing total cost for a particular area (Fig.1.8).



(a)

1	1	1	1	1
1	2	2	2	1
2	2	2	2	2
3	3	3	3	3
3	3	3	3	3

Soil Type

(b)

0.1	0.1	0.2	0.2	0.2
0.1	0.1	0.2	0.2	0.2
0.1	0.1	0.2	0.2	0.2
0.1	0.1	0.2	0.2	0.2
0.1	0.1	0.2	0.2	0.2

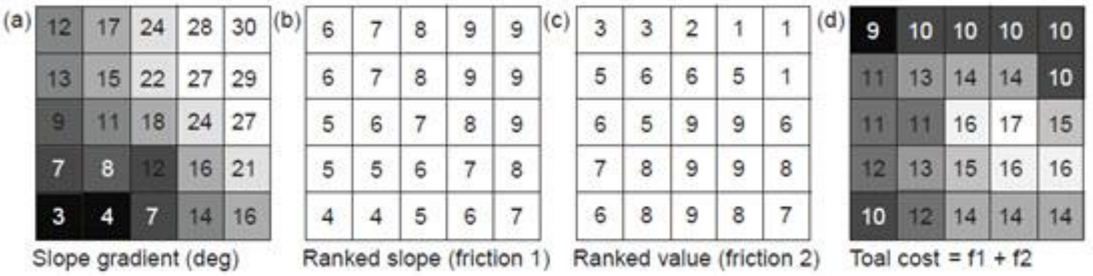
Average Rainfall (m/yr)

(c)

1.1	1.1	1.2	1.2	1.2
1.1	2.1	2.2	2.2	1.2
2.1	2.1	2.2	2.2	2.2
3.1	3.1	3.2	3.2	3.2
3.1	3.1	3.2	3.2	3.2

Growing Conditions

Fig. 1.7. An example of a simple overlay operation involving two input rasters: (a) an integer raster representing soil classes (class 2, representing sandy loam, is considered optimum); (b) a floating-point raster representing average rainfall, in metres per year (0.2 is considered optimum); and (c) the output raster derived by addition of a and b to produce a result representing conditions for a crop; a value of 2.2 (2 þ 0.2), on this rather arbitrary scale, represents optimum growing conditions and it can be seen that there are five pixel positions which satisfy this condition.



(a)

12	17	24	28	30
13	15	22	27	29
9	11	18	24	27
7	8	12	16	21
3	4	7	14	16

Slope gradient (deg)

(b)

6	7	8	9	9
6	7	8	9	9
5	6	7	8	9
5	5	6	7	8
4	4	5	6	7

Ranked slope (friction 1)

(c)

3	3	2	1	1
5	6	6	5	1
6	5	9	9	6
7	8	9	9	8
6	8	9	8	7

Ranked value (friction 2)

(d)

9	10	10	10	10
11	13	14	14	10
11	11	16	17	15
12	13	15	16	16
10	12	14	14	14

Toal cost = f1 + f2

Fig. 1.8 (a) Slope gradient in degrees; (b) ranked (reclassified) slope gradient constituting the first cost or friction input; (c) ranked land value (produced from a separate input land-use raster) representing the second cost or friction input; and (d) total cost raster produced by aggregation of the input friction rasters (f1 and f2). This total cost raster could then be used within a cost-weighted distance analysis exercise.

1.2.4 N-ary operations

Here we deal with a potentially unlimited number of input layers to derive any of a series of standard statistical parameters, such as the mean, standard deviation, majority and variety. Ideally there should be a minimum of three layers involved but, in many instances, it is possible for the processes to be performed on single layers; the result may, however, be rather meaningless in that case. The more commonly used statistical operations and their functionalities are summarized in Table 1.4. As with the other local operations, these statistical parameters are point operations derived for each individual pixel position, from the values at corresponding pixel positions in all the layers, rather than from the values within each layer.

Table 1.4. Summary of local pixel statistical operations, their functionality and input/output data format.

Statistic	Input format	Functionality	Data type
Variety	Only rasters. If a number is input, it will be converted to a raster constant for that value	Reports the number of different of different DN values occurring in the input rasters	Output in integer
Mean		Reports the average DN value among the input rasters	Output is floating point
Standard deviation	Rasters, numbers and constants	Reports the standard deviation of the DN values among the input rasters	Output is floating point
Medium	Only rasters. If a number is input, it will be converted to a raster constant for that value	Reports the middle DN value among the input raster pixel values. With an even number of inputs, the values are ranked and the middle two values are averaged. If inputs are all integer, output will be truncated to integer	If inputs are all integer, output will be integer, unless one is a float, there the input will be a float
Sum		Reports the total DN value among the input rasters	
Range		Reports the difference between maximum and minimum DN	
Maximum		Reports the highest DN value among the input rasters	
Minimum		Reports the lowest DN value among the input rasters	
		Reports the DN value which occurs most frequently among the input rasters. If no clear majority, output = null, for example if there are three inputs all with different values. If all inputs have equal value, output=input	
Minority		Reports the DN value which occurs least frequently among the input rasters. If no clear minority, as majority If only two inputs, where different, output= null. If all inputs equal, output = input. If only one input, output= input	

1.2.4.1 Local statistics

When we have many related grids defined in the same region, we often want to assess change: at each cell, how varied are the grid results? How large do they get? How small? What is the average? These questions make sense for numerical data.

For grids with ordinal data--that is, values that can be ordered, but which may not have any absolute meaning--you can still ask about order statistics. These are the relative rankings of values within the ordered collections of values observed at each cell.

For grids with categorical data, you might want to know at each cell whether one category predominates throughout the collection of grids and how many different categories actually appear at the cell's location.

In all these cases, imagine a stack of grids with common mesh.

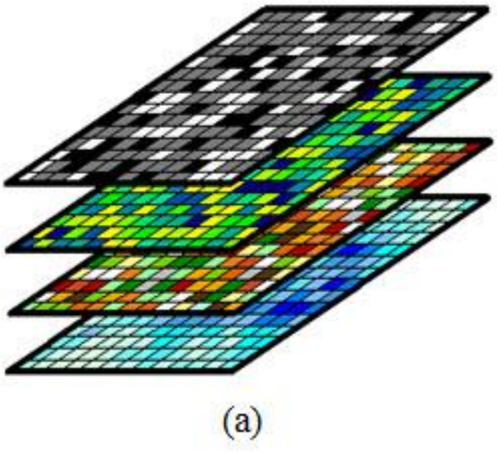


Fig. 1.9. (a) stack of grids with common mesh.

At each cell location there is a stack of values, one for each grid.

The N-ary operators create a new grid whose values depend on the stack of input data at each cell location.

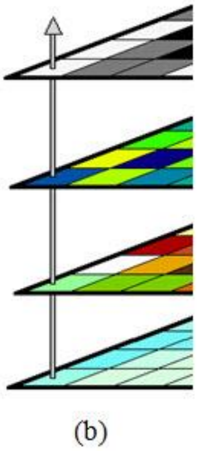


Fig. 1.9. (b) New grid by N-ary operator.

The Spatial Analyst syntax for some of these requests is strange, because it wants to force expressions into the form "aGrid. Request (list of other grids)". This is inherently asymmetric because it singles out one grid in the collection to play the role of the object ("aGrid") to which the calculation is applied and leaves the other grids in the role of a list of arguments ("list of other grids"). Despite this syntax, for some requests, such as the local statistics, there is no asymmetry in the calculation itself: all the grids are equivalent. For some other requests, there is an asymmetry in the calculation: one grid plays a special role.

Spatial Analyst constructs lists with curly braces {} and separates the elements by commas.

- **Compute local statistics**

[Float]. LocalStats (#GRID_STATYPE_MAX, {[Float1], [Float2], [Float3]}) Computes the largest value among four grids.

[Float]. LocalStats (#GRID_STATYPE_MEDIAN, {[Float1], [Float2]}) Computes the median of three values.

[Integer]. LocalStats (#GRID_STATYPE_MAJORITY, {[Integer1], [Integer2]}) Computes the value occurring the most times (out of the three input values at each cell). If two or more values occur an equal number of times, Spatial Analyst returns NoData.

The Majority statistic evidently is not very useful when many ties occur: that is, when there are many cells where two or more values occur equally often.

- **Compare one grid** (a “base’ grid) to many others simultaneously

[Float]. Grids Greater than ({[Float1], [Float2], [Float3], [Float4]}) For each base cell in [Float], computes the number of times corresponding cells from [Float1], ..., [Float4] exceed (and do not equal) the base cell’s value. There is a corresponding Grids Less Than operator.

- **Combine** the values of two grids based on values at a third grid

[Indicator]. Con ([Float1], [Float2]) Creates a grid with the values of [Float1] where [Indicator] is nonzero and with the values of [Float2] where [Indicator] is zero.

The **Con** request is especially useful. The result of **Con**, by default, is the second grid ([Float2] or [Mosaic] in the examples). However, at cells where [Indicator] is true, the values of the first grid ([Float1] or [Average]) are "painted" over the default values. Thus, the **Con** request is a natural vehicle for selectively editing grids.

References

- ✓ Jian Guo Liu, Philippa J. Mason, "Essential Image Processing and GIS for Remote Sensing," Imperial College London, UK, 261-280 (2009).
- ✓ 1997. User Interfaces for Map Algebra. Journal of the Urban and Regional Information Systems Association, Vol.9, No. 1, pp. 44-54.
- ✓ http://www.quantdec.com/SYSEN597/GTKAV/section9/map_algebra2.htm