CARTOGRAPHIC MODELING

CARTOGRAPHIC MODELING (Tomlin)

- *Geographic Information Systems and Cartographic Modeling* by C. Dana Tomlin, Prentice-Hall, Englewood Cliffs, NJ, 1990 [Toml90].
- Overlay mapping
- Library of maps

  1. all in registration (i.e., a common origin)
  2. inherently two-dimensional
- Goal: perform a sequence of operations on them
- Cartographic model consists of a hierarchy with a map layer at its highest level
- Each map layer consists of zones
- Each zone consists of locations
- Each location consists of a pair of coordinate values specifying its address

MAP LAYER

1. Captures variation of just one variable
2. Description:
   - title
   - resolution
   - orientation
   - constituent zones
3. Different from a conventional map
   - in a conventional map each location has many attributes
   - each map layer has only one attribute

ZONES

1. Collection of locations
   - all have a specific attribute value, or
   - all lie within a given range of values
   - often called a *class*
2. Area in each zone need not be contiguous
   - mutually exclusive (zones are disjoint)
   - all inclusive (zones span entire map layer)
3. Description
   - label—written name of the zone
   - value—must be integer (exception is null value which is unknown)
   - location(s)

SPATIAL DATA

- Distinguished by being continuous
- Spatial variation
  1. isotropic
     - does not vary with direction
     - e.g., travel costs that increase uniformly with distance
  2. anisotropic
     - varies with direction
     - e.g., travel costs that increase at a non-linear rate with distance such as exponential
     - useful for accessibility computations such as whether a point is on or off a particular network of roads

TEMPORAL DATA

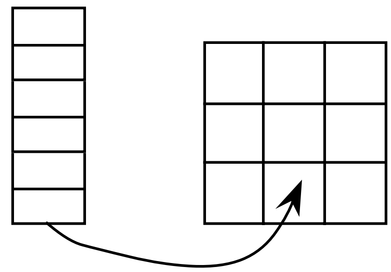
- Time can be a third dimension for 2-d data or a fourth dimension for 3-d data
- Could store as time slices—one layer per slice
- Applications
  1. remote sensing
  2. migration data
  3. snow coverage
  4. robotics—e.g., interference detection
  5. conventional databases:
     - two concepts:
       a. transaction time — when did information become known?
       b. valid time — when does the information hold?
     - time is continuous rather than discrete
     - how do we keep track of rate information?

LAYERS VS: OBJECTS

- Layers
  1. capture the variation of one variable over the surface of the Earth
  2. tendency to view a layer as an artificial way of describing variation
  3. a continuous view of the world
- Objects
  1. objects can be points, lines, areas, etc.
  2. a location can be occupied by any number of objects
  3. a discrete view of the world
  4. combines a description of the object with the specification of the applicable operations
  5. new objects can inherit properties of their parents
  6. layer view seems inefficient when variables are defined only over a limited geographic area or classes of objects
- No right answer!
  1. layers are good for managing continuous resources (e.g., forests, crops, etc.)
  2. objects are useful for facilities management tasks (e.g., keeping track of poles, cables, etc. for utilities)

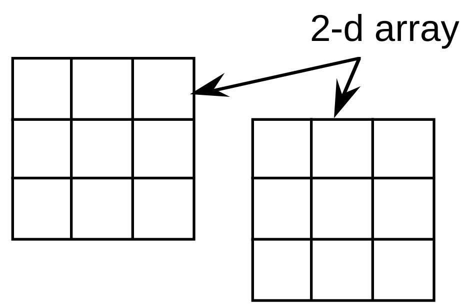
OVERLAY MAPPING DATA ORGANIZATION

1. Vertical array



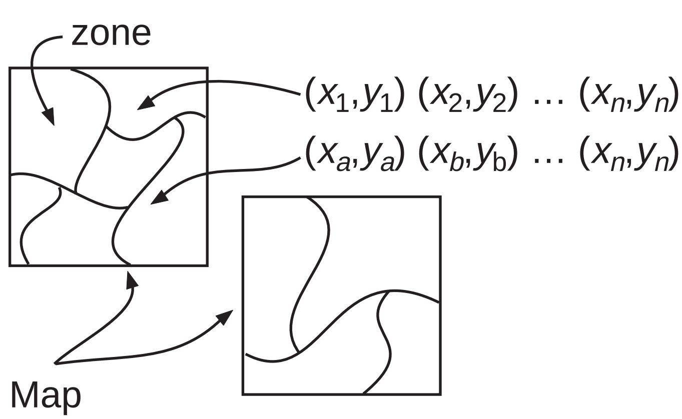
   - a 1-d array for each location
   - one entry for each overlay
   - entry contains attribute value for the overlay
   - good for random access

2. Overlay of locations



   - one 2-d array for each overlay
   - one entry for each location
   - contains the attribute value for the overlay
   - good for random access

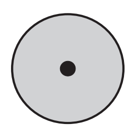
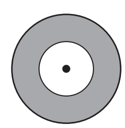
3. Overlay of zones

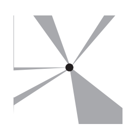


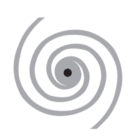
   - one record for each overlay
   - record is decomposed into zones of equal attribute values
   - each zone is a record consisting of
     a. attribute value
     b. list of locations comprising the zone
   - good for sequential access

LOCATIONS

1. Primitive elements of cartographic space
2. Each location in a layer is associated with a geographic characteristic
3. Each location is a grid square
   - uniform size, shape, and orientation
   - also known as a *pixel*
4. Aggregations of locations
   - rows
   - columns
   - neighborhoods

     a. distance
       - maximum distance 
       - range of distances 

     b. direction or range of directions 

     c. combination of distance and direction 
       - arc of a spiral ($r = a\theta$)

     d. different from a zone since can overlap

FRAME OF REFERENCE

- Locational
  1. Polar
     - based on distance and compass direction to a particular point (also known as *azimuthal*)
  2. Cartesian
     - based on values representing distances in perpendicular directions
     - origin may be arbitrary or fixed (e.g., Greenwich Mean Time) as well as free or even varying with time
- Forms of locational reference
  1. specific coordinate values
  2. identity of neighbors or adjacent entities
  3. physical objects such as lakes, rivers, signposts (e.g., exit 37 or 41 kilometers from Madrid)
  4. enclosing boxes
  5. place names
  6. reference to tiles or grid squares on a map

MEASUREMENTS

- Distinguish between a value and a measurement
- Value is characterized by
  1. scale — level of generalization (real world)
  2. resolution — smallest recording unit (# of pixels)
  3. precision — fineness of measurement (# of digits)
  4. accuracy — how accurate is the data (i.e., close to reality)
- Values correspond to measurements (in increasing order of complexity)
  1. nominals (e.g., wheat, corn, rice, ...)
  2. ordinals (e.g., first, second, third, …)
  3. ranges (i.e., intervals)
     - good for differences but not for proportions
     - arbitrary zero
  4. ratios (includes numbers)
     - real zero
     - permit multiplication and/or division
- Measurement classes
  1. categorical: nominals, ordinals
  2. scalar: ranges, ratios
- Not all operations are applicable to all measurement types
  1. can’t take maximum of a nominal
  2. can’t apply arithmetic functions to nominals
- If an operation is applicable to a measurement type, then it is applicable to each of its values
- Measurements can be derived
  1. reclassification via substitutions, combinations, etc.
  2. via statistical methods such as averages, maximums, minimums, etc.

SPECIFICATION OF LOCATIONS

1. Use coordinates

2. Explicit encoding

   - *x* and *y* coordinate values

   - value

3. Use the implicit association between locations and coordinates

   - takes advantage of regularity of cartographic grid

   - saves space

   - makes use of ordering of space

     a. simple ordering

        - by pixels à la Morton ordering

     b. couple with aggregation of similarly-valued locations

        - e.g., Morton ordering by quadtree blocks

   - basis of association

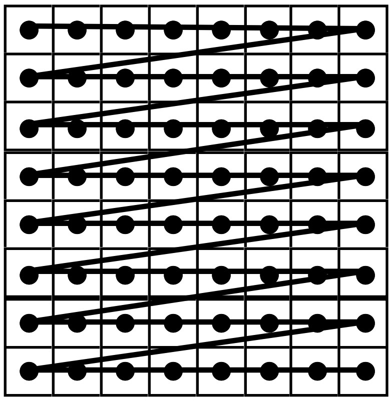
     a. interior of space occupied by similarly-valued locations

     b. boundary of space occupied by similarly-valued locations

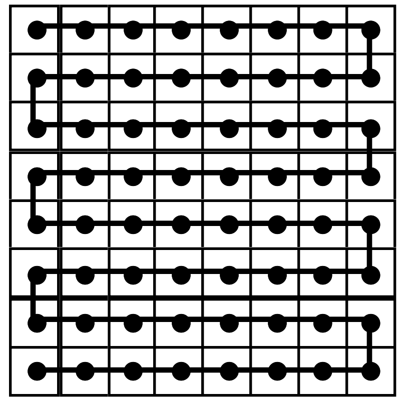
INTERIOR-BASED SPACE-ORDERING METHODS

1. Store and process locations based on their position in the cartographic grid
   - a linearization of the cartographic grid
   - only need 1 number per location (i.e., its value) instead of 3
   - form a space-filling curve
     a. raster-scan (order by rows and columns)
     b. Peano (order by blocks)
2. Augment ordering with aggregation based on similarity of values (e.g., runlength codes)
   - one-dimensional (e.g., by rows or columns)
   - two-dimensional
     a. medial axis transforms (MAT)
     b. quadtrees

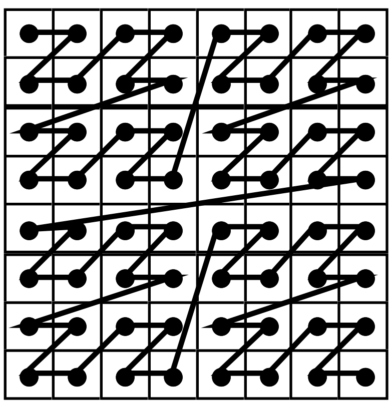
SAMPLE SPACE ORDERING METHODS



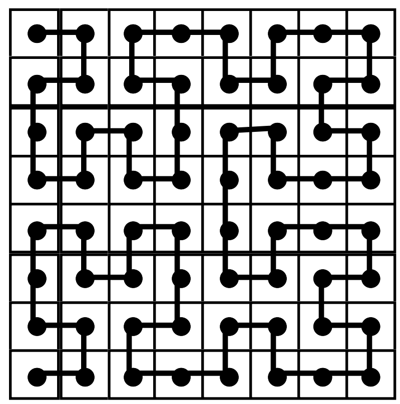
Row order



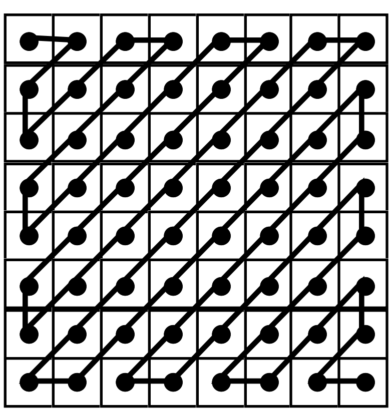
Row-prime order



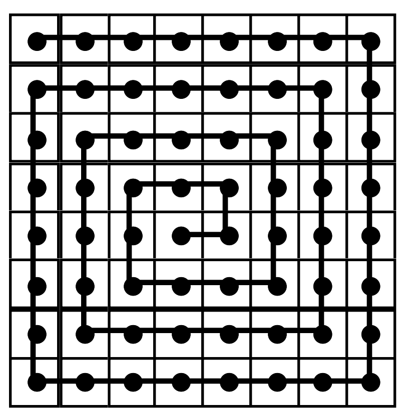
Morton order



Peano-Hilbert order



Cantor order



Spiral order

CONVERTING BETWEEN POINTS AND CURVES

- Need to know size of image for all but the Morton order

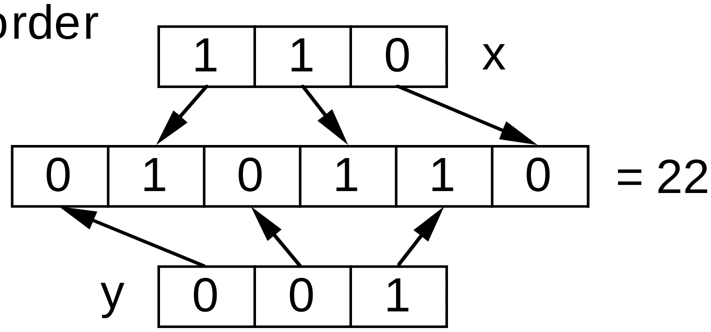
- Relatively easy for all but the Peano-Hilbert order which is difficult (although possible) to decode and encode to obtain the corresponding *x* and *y* coordinate values

- Morton order

  1. use bit interleaving of binary representation of the *x* and *y* coordinates of the point

  2. also known as Z-order

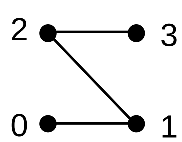
  3. Ex: `Atlanta` (6,1)



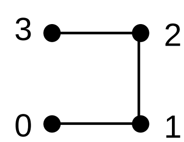
STABILITY OF SPACE ORDERING METHODS

- An order is *stable* if the relative order of the individual pixels is maintained when the resolution (i.e., the size of the space in which the cells are embedded) is doubled or halved
- Morton order is stable while the Peano-Hilbert order is not
- Ex:

Morton:



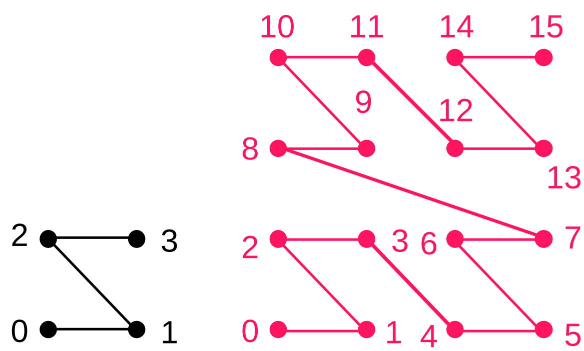
Peano-Hilbert:



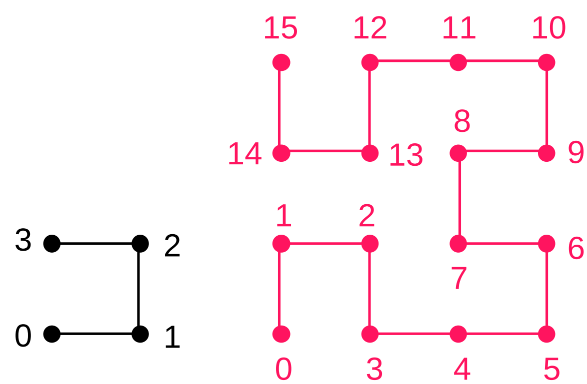
STABILITY OF SPACE ORDERING METHODS

- An order is *stable* if the relative order of the individual pixels is maintained when the resolution (i.e., the size of the space in which the cells are embedded) is doubled or halved
- Morton order is stable while the Peano-Hilbert order is not
- Ex:

Morton:



Peano-Hilbert:



- Result of doubling the resolution (i.e., the coverage)

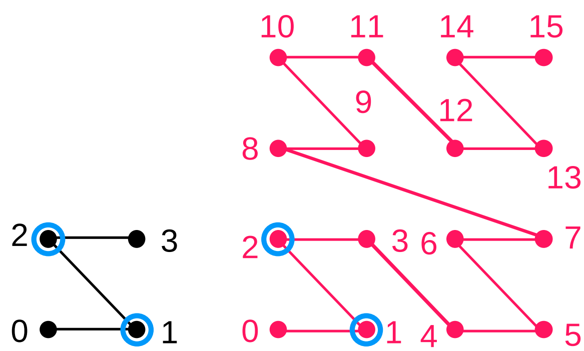
STABILITY OF SPACE ORDERING METHODS

- An order is *stable* if the relative order of the individual pixels is maintained when the resolution (i.e., the size of the space in which the cells are embedded) is doubled or halved

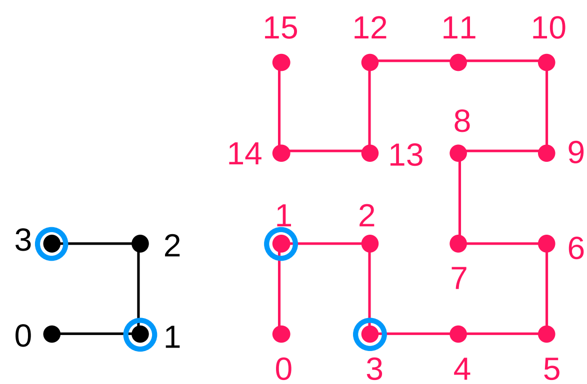
- Morton order is stable while the Peano-Hilbert order is not

- Ex:

Morton:



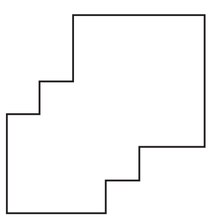
Peano-Hilbert:



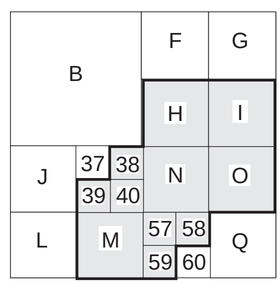
- Result of doubling the resolution (i.e., the coverage) in which case the circled points do not maintain the same relative order in the Peano-Hilbert order while they do in the Morton order

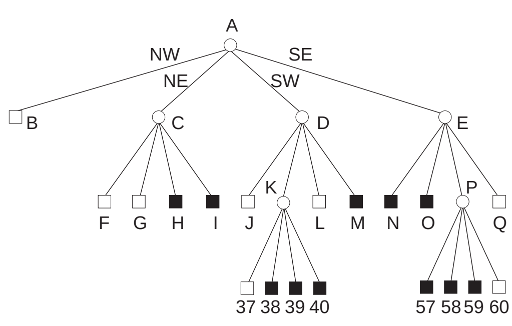
REGION QUADTREE

- Repeatedly subdivide until obtain homogeneous region
- For a binary image (BLACK ≡ 1 and WHITE ≡ 0)
- Can also use for multicolored data (e.g., a landuse class map associating colors with crops)
- Can also define the data structure for grayscale images
- A collection of maximal blocks of size power of two and placed at predetermined positions
  1. could implement as a list of blocks each of which has a unique pair of numbers:
     - concatenate a sequence of 2 bit codes corresponding to the path from the root to the block's node
     - the level of the block's node
  2. does not have to be implemented as a tree
     - tree good for logarithmic access
- A variable resolution data structure in contrast to a pyramid (i.e., a complete quadtree) which is a multiresolution data structure



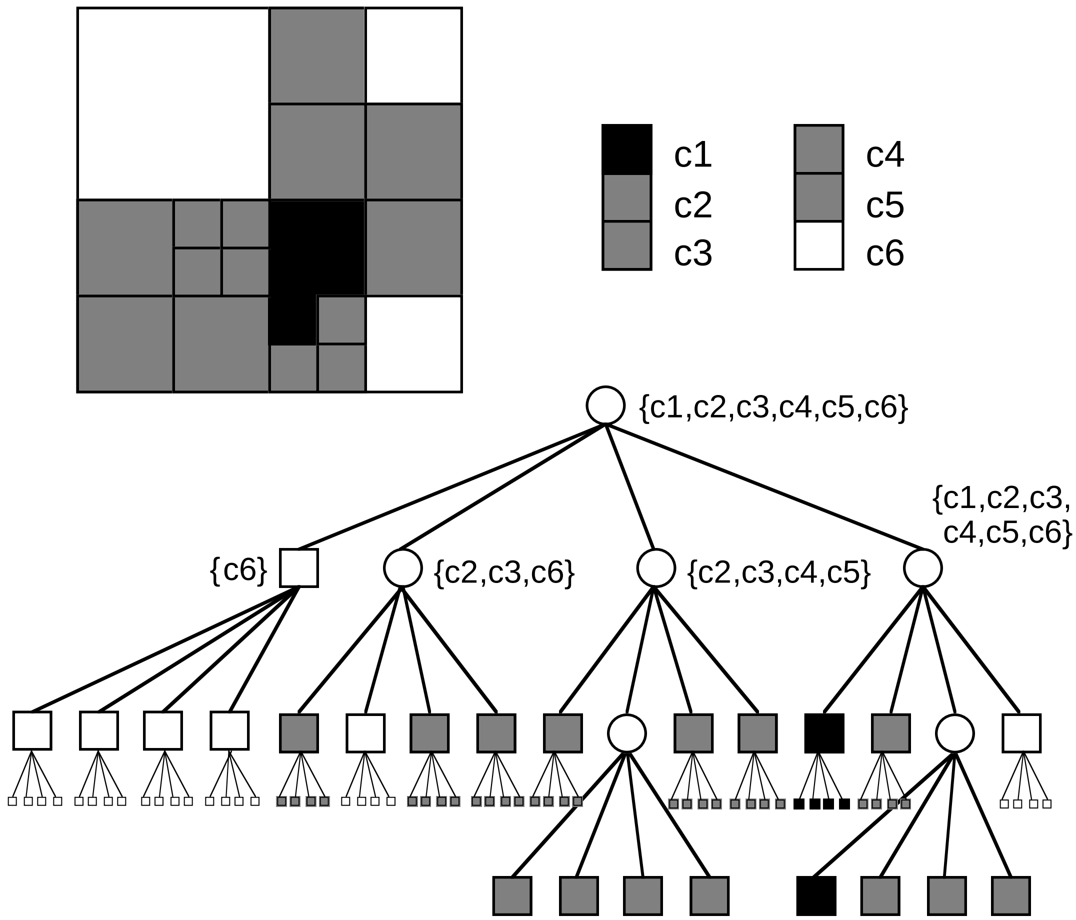
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	1	1	1	1
0	0	0	0	1	1	1	1
0	0	0	1	1	1	1	1
0	0	1	1	1	1	1	1
0	0	1	1	1	1	0	0
0	0	1	1	1	0	0	0





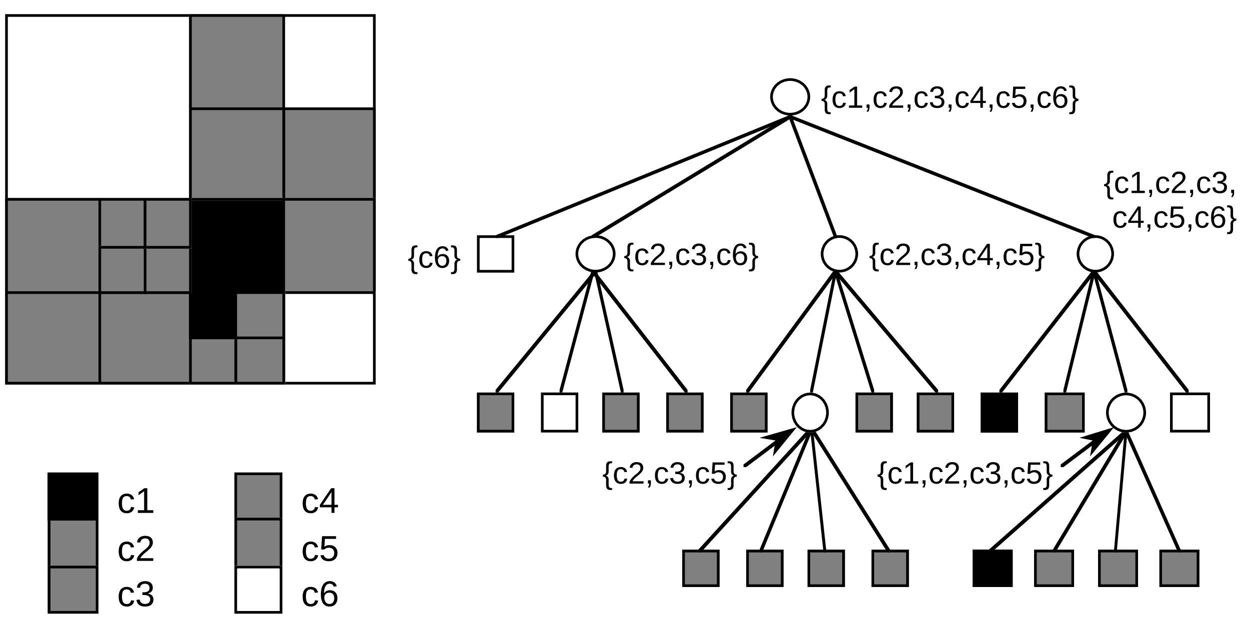
PYRAMID

- Internal nodes contain summary of information in nodes below them
- Useful for avoiding inspecting nodes where there could be no relevant information



QUADTREES VS. PYRAMIDS

- Quadtrees are good for location-based queries
  1. e.g., what is at location *x*?
  2. not good if looking for a particular feature as have to examine every block or location asking “are you the one I am looking for?”
- Pyramid is good for feature-based queries — e.g.,
  1. does wheat exist in region *x*?
     - if wheat does not appear at the root node, then impossible to find it in the rest of the structure and the search can cease
  2. report all crops in region *x* — just look at the root
  3. select all locations where wheat is grown
     - only descend a node if there is a possibility that wheat is in one of its four sons — implies little wasted work
- Ex: truncated pyramid where 4 identically-colored sons are merged



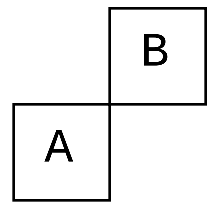
- Can represent as a list of leaf and nonleaf blocks (e.g., as a linear quadtree)

BOUNDARY-BASED SPACE-ORDERING METHODS

- Not very useful if no similarity of values as it degenerates to a collection of boundaries of individual locations
- Most useful with zones
- Ordering is relative
- Each location in the sequence is described by its relative position with respect to an adjacent location in the sequence

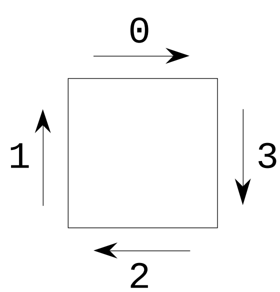
CHAIN CODE

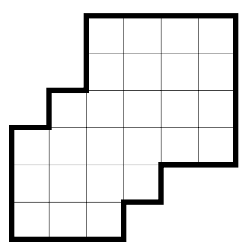
- Records relative position of boundaries of similarly-valued adjacent grid squares
- Four directions
- Usually assume the image is to the right
- Assume four-connected, meaning that A and B are not in same region unless eight-connected



- Aggregation similar to runlength representation
  1. based on direction of boundary rather than value of location
  2. no 2 consecutive elements on the aggregated boundary have the same direction
  3. implement with a number to indicate length, OR

code = direction /90





starting at lower left corner yields

$1^3\ 0^1\ 1^1\ 0^1\ 1^2\ 0^4$

$3^4\ 2^2\ 3^1\ 2^1\ 3^1\ 2^3$

  - can use +1 and –1 to indicate a change of direction
  - more compact than absolute directions 0,1,2,3

  4. just store the locations where the directions change
- Comparison
  1. 3 is relative while 4 is absolute
  2. 3 is more compact than 4
  3. 4 is more robust as an error means that only one element is corrupted

DATA FORMATS

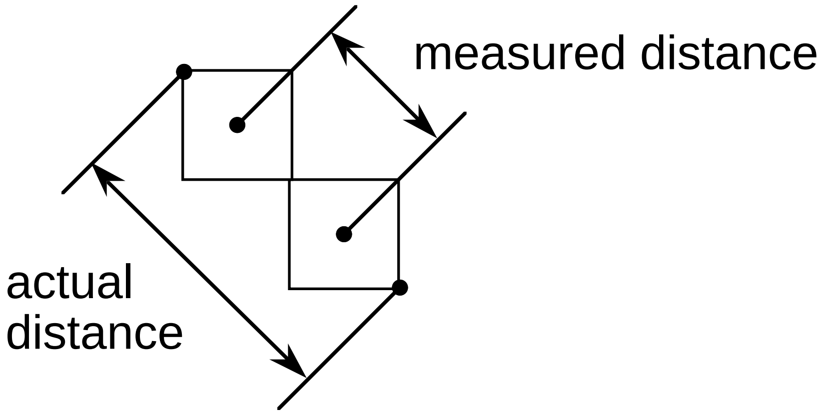
1. Data sampling
   - raster corresponds to regular sampling
     a. does not permit spatial variation
     b. somewhat wasteful in area of no variation
   - vector permits spatial variability as only sample at locations of interest
2. Vector format
   - involves boundaries (usually points and lines)
   - magnitude and direction
3. Raster format
   - interiors of locations (or sequences of locations)
   - objects do not exist as distinct entities—i.e., the locations are the entities
   - spatial features are represented by sequences of locations

PRECISION AND ACCURACY IN DATA FORMATS

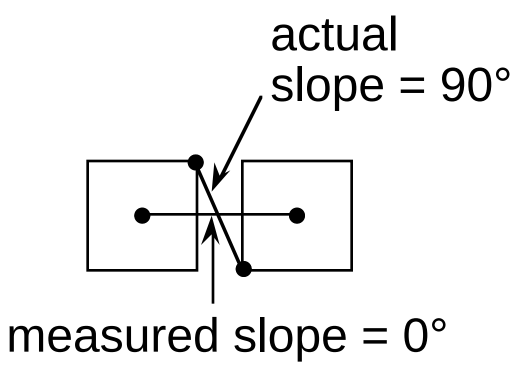
1. Precision is the degree of refinement with which a measurement can be represented

   - number of significant digits
   - 4.51118 is precise but not necessarily accurate
   - 5.5 is accurate but not precise
   - related to resolution if the number of pixels in the image corresponds to precision of the measurement
   - repeatability of the measurement
   - raster: precision increases as size of grid squares decreases but data volume increases dramatically
   - vector: only need to increase precision with which the coordinates are specified
   - result: percentage of locations with meaningful data on a raster map layer is several orders of magnitude smaller than on a vector map layer since so many locations on the raster layer have no data

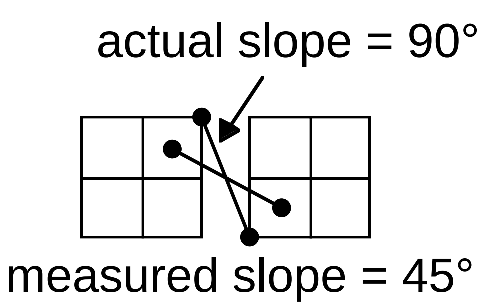
2. Accuracy is how close the measurement is to the actual data
   - raster approximates a point with the center of the grid square in which the point lies
   - error in measuring distance between 2 points may be as large as the length of the diagonal of grid square

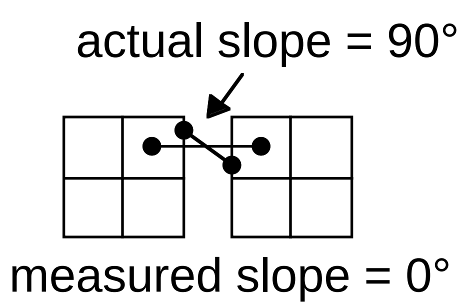


   - error in measuring slope of the line connecting two points may be as large as 90 degrees



   - increasing resolution reduces magnitude of error in distance measurements but does not always reduce the slope error



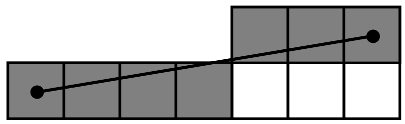


DISCRETIZATION OF CARTOGRAPHIC SPACE

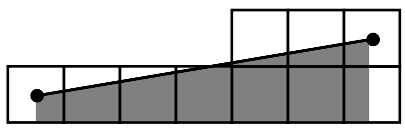
1. Yes for raster format

2. Partially so for vector format

   - yes for point data and endpoints of lineal data



   - no for other data since partition of space need not conform to underlying grid—i.e., some cells are partially within and outside the region bounded by the line



   - result: aggregates of locations can't represent all spatial characteristics—need vectors, polygons, and polyhedra

3. Discrete locations of raster format are more general

4. Vectors, polygons, and polyhedra of vector format are more specialized

5. Definition of cartographic space in terms of discrete locations can be supported by both raster and vector formats

   - raster is easy as it is just the set of locations defining a region

   - vector is a sequence of consecutive or connected locations

RASTER VS. VECTOR DATA FORMATS

1. Data acquisition
   - raster is good for automated methods
   - vector is basis of traditional paper map
2. Output
   - raster is more like a photographic image
   - raster is superior where the feature of interest is the spatial variability of a phenomenon
   - vector is more like a line drawing
3. Data interpretation
   - raster is useful for location-based queries
     a. position-oriented
        - function of location

     b. answers *what* is at a location

     c. records features associated with locations
   - vector is useful for feature-based queries
     a. theme-oriented
        - function of theme or attribute

     b. answers *where* a feature is

     c. records locations associated with features

TYPES OF OPERATIONS (renamed cm34.1)

1. Combinations of maps (Boolean or set-theoretic)
2. Measurements
   - average
   - diversity (e.g., standard deviation, moments)
   - majority
   - minimum and maximum
   - variety (e.g., cardinality)
3. Reclassification or recoding
4. Search

SPATIAL CLASSIFICATION OF OPERATIONS (renamed cm34.1)

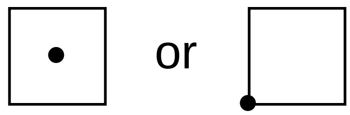
- Region of interest (also termed a *window*)
  1. individual locations (local)
  2. zones (zonal)
  3. neighborhoods of locations (focal)
     - can restrict extent of neighborhood by distance
     - can restrict extent of neighborhood by direction
  4. special neighborhoods when the data does not fit into the domain of grid squares (incremental)
     - only for immediate neighbors
     - no restriction on direction is permitted
- Relationship between the locations being operated on
  1. coincident in position (local)
  2. related thematically (zonal)
  3. related spatially (focal and incremental)

GEOGRAPHIC FEATURES

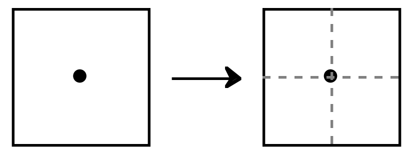
1. Points, lines, areas, and surfaces
2. Grid squares bear little resemblance to geographic space
3. Solution: add an additional map layer
   - value of location characterizes data in original layer
   - incremental operations operate on additional map layer

SAMPLE PROBLEMS WITH A RASTER GIS

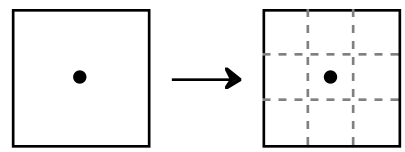
1. Is a point associated with center or corner of a grid square?



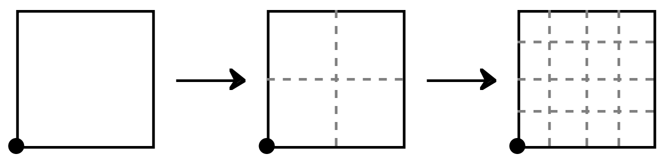
   - usually with center but cumbersome when resolution is increased



   - can avoid by tripling the resolution

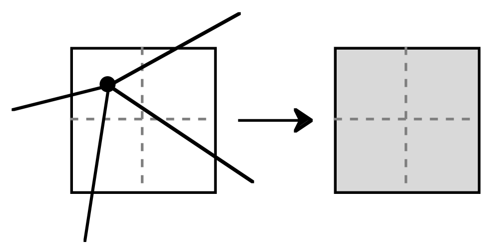


   - no such problems with corner



2. Do we associate a grid square with each part of a line?

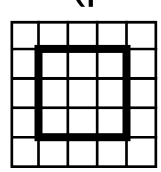
   - what if four lines meet at a point?



   - what if five lines meet at a point?

3. How is size of areal data measured (perimeter or area)?

4. What does perimeter mean?



   - number of grid squares on inner boundary (8)
   - number of grid squares on outer boundary (16)
   - number of boundary segments or cracks (12)

5. Surfaces

- distinguish between 3-d (volume) and 2.5-d (surface)

    a. surface is single-valued

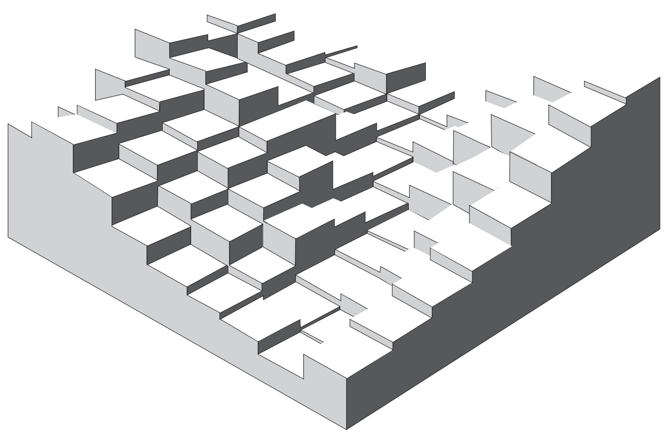
    b. multi-valued includes polyhedra, spheres, boxes, …

- representing a surface

    a. associate a single value with center of each grid square

    b. leads to discontinuities at boundaries

    c. need interpolation

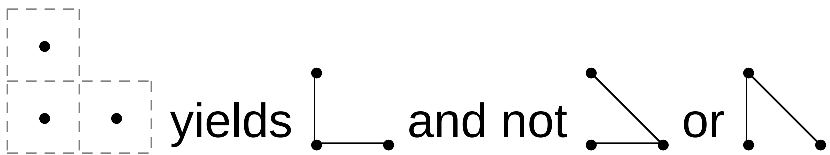


GRID SQUARE INTERPRETATION OF LINEAL DATA

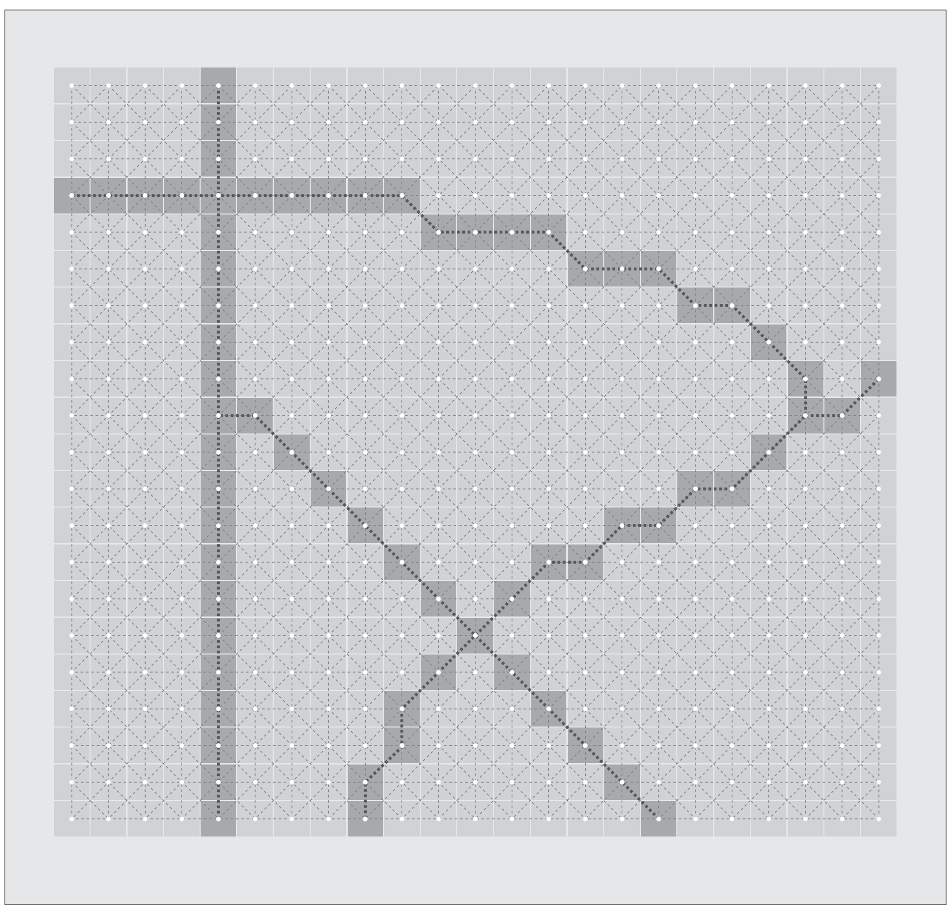
1. If 2 similarly-valued grid squares are adjacent along an edge or a corner, then a line exists between their centers

2. If 3 similarly-valued grid squares are adjacent along both an edge and a corner, then lateral adjacency takes precedence over diagonal adjacency

   - result is ‘L’-shaped



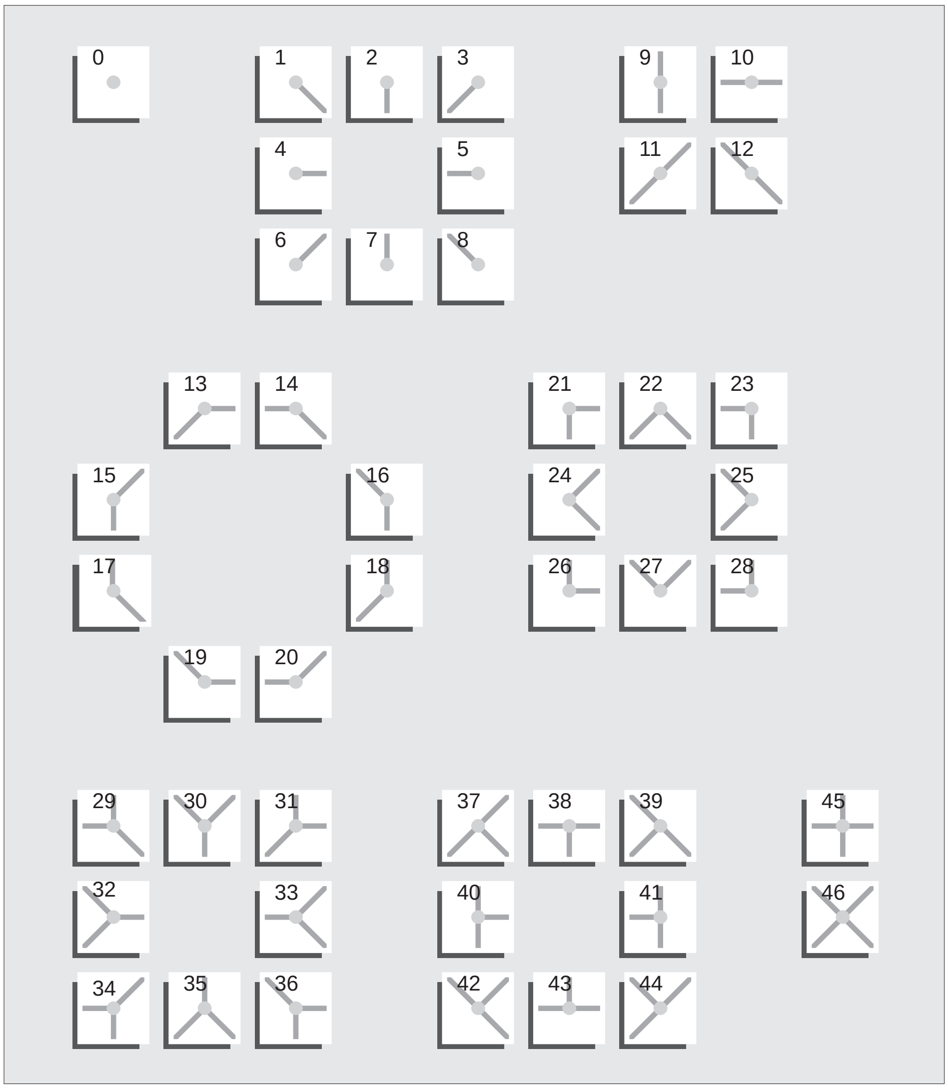
   - Ex: Figure 1-16 from [Toml90]



INCORPORATION OF LINEAL DATA

1. 47 legal configuration patterns
2. Additional map layer stores the identity of the appropriate configuration with each location

   Ex: Figure 1-17 from [Toml90]

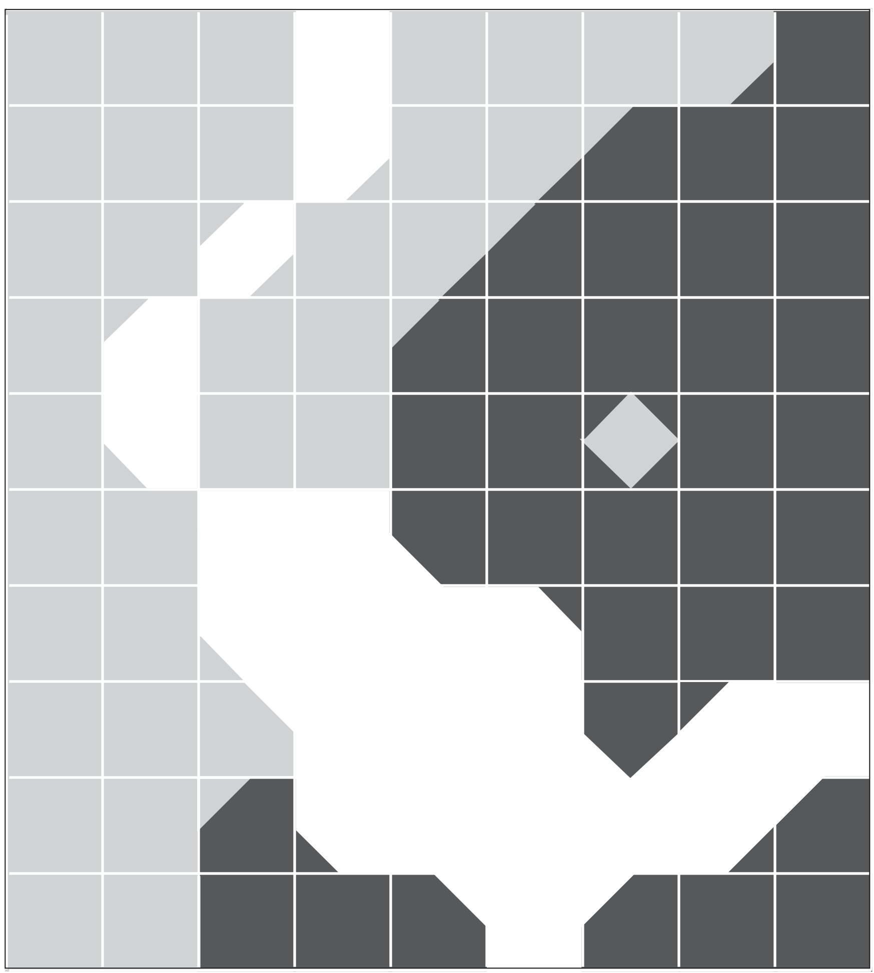


GRID SQUARE INTERPRETATION OF AREAL DATA

1. Interpret the corners of grid squares as being beveled

2. Nature of beveling depends on the configuration of values of the four grid squares that meet at each corner of a grid square

   Ex: Figure 1-18 from [Toml90]



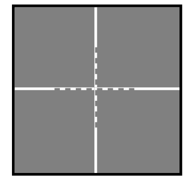
INCORPORATION OF AREAL DATA

1. Examine the values of the 4 locations that share each corner of a grid square
2. Each of these locations has at most 4 different values {black, black-gray, gray-white, white}
3. 4^4 = 256 possible configuration patterns
4. Reduced to 7 sets of configuration patterns using symmetry, rotations, and reflections
5. Additional map layer indicates which of each location’s corners are beveled by specifying the configuration patterns
6. Beveling information plus colors of neighboring locations suffice to describe any location
7. More refined inferences can be drawn by examining extended neighborhoods (i.e., more than 4 locations that share a corner)

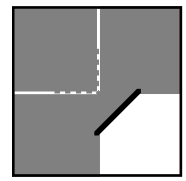
SETS OF AREAL CONFIGURATION PATTERNS

- broken lines: boundaries between similar values
- solid lines: boundaries between different values

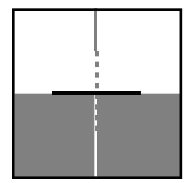
1. single color for all locations (4)



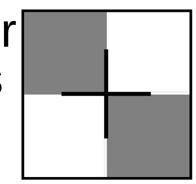
2. 3 locations with one color and another color for the remaining location ($4 \times 4 \times 3 = 48$)



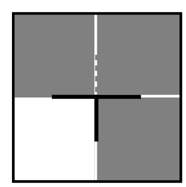
3. 2 laterally adjacent locations with one color and another color for the remaining locations ($4 \times 4 \times 3/2 = 24$)



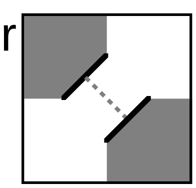
4. 2 diagonally adjacent locations with one color and another color for the remaining locations ($2 \times 4 \times 3/2 = 12$)



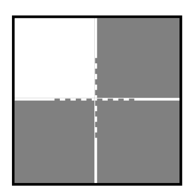
5. 2 laterally adjacent locations with one color and different colors for the remaining 2 locations ($4 \times 4 \times 3 \times 2 = 96$)



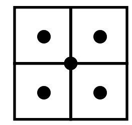
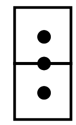
6. 2 diagonally adjacent locations with one color and different colors for the remaining 2 locations ($2 \times 4 \times 3 \times 2 = 48$)



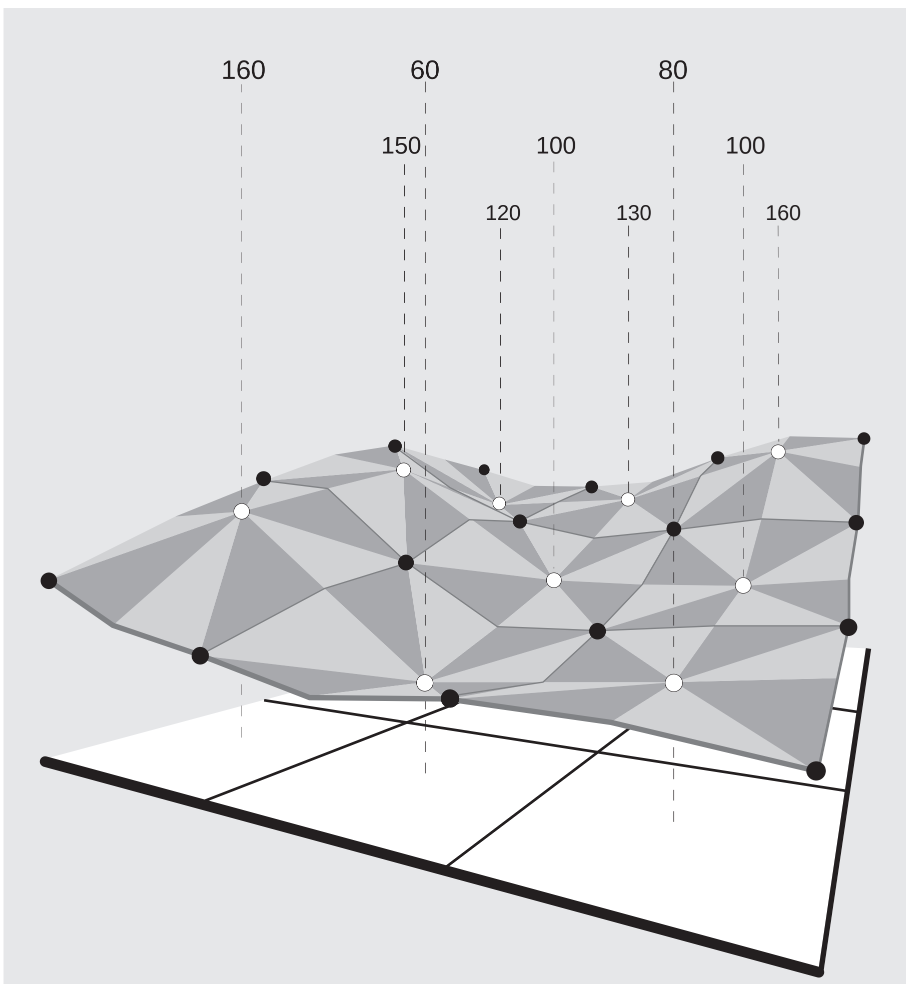
7. a different color for each location ($4 \times 3 \times 2 \times 1 = 24$)



GRID SQUARE INTERPRETATION OF SURFACE DATA

1. Rectangular volume elements
2. Surface consists of 8 triangular facets
   - associate an elevation with center of grid square
   - elevation of corner of grid square = average of four diagonally adjacent neighbors 
   - elevation of midpoints of sides = average of centers of two laterally adjacent neighbors 
3. Additional map layer stores the elevation of the center of the grid square

   Ex: Figure 1-21 from [Toml90]



TYPES OF OPERATIONS

1. Combinations of maps (Boolean or set-theoretic)
2. Measurements
   - average
   - diversity (e.g., standard deviation, moments)
   - majority
   - minimum and maximum
   - variety (e.g., cardinality)
3. Reclassification or recoding
4. Search

SPATIAL CLASSIFICATION OF OPERATIONS

- Region of interest (also termed a *window*)
  1. individual locations (local)
  2. zones (zonal)
  3. neighborhoods of locations (focal)
     - can restrict extent of neighborhood by distance
     - can restrict extent of neighborhood by direction
  4. special neighborhoods when the data does not fit into the domain of grid squares (incremental)
     - only for immediate neighbors
     - no restriction on direction is permitted
- Relationship between the locations being operated on
  1. coincident in position (local)
  2. related thematically (zonal)
  3. related spatially (focal and incremental)

LOCAL OPERATIONS

Ex: `DESTINATIONLAYER = LocalFunction of SOURCELAYER [and NEXTSOURCELAYER] etc.`

1. Each location in the `DESTINATION` map layer is set to a value that is a function of the location's existing values in one or more `SOURCE` map layers
2. At least one of the `SOURCE` layers is an actual map layer
3. Other `SOURCE` layers can be numbers in which case every location has this value
4. Types of operations
   - recoding or reclassification
   - measurement
   - search
   - polygon overlay (i.e., composition, superposition)
5. Example operations
   - `Sum, Mean, Difference, Maximum, Minimum, Product, Ratio, Majority, Minority, Variety` (i.e., cardinality of different values), etc.

RECODING OR RECLASSIFICATION

- One operand is a map layer
- Second operand, if present, is usually a number (e.g., a `ratio` or `product` operation where the result is a rescaling)
- `LocalRating` operation specifies an explicit correspondence from existing values or ranges of values to new values
  1. all values not explicitly reclassified are left alone
  2. can specify a correspondence with a map layer instead of a value. In this case, the location's value in the map layer serves as the new value reclassifying the value in the primary source operand
     - Boolean union of map1 with map2:

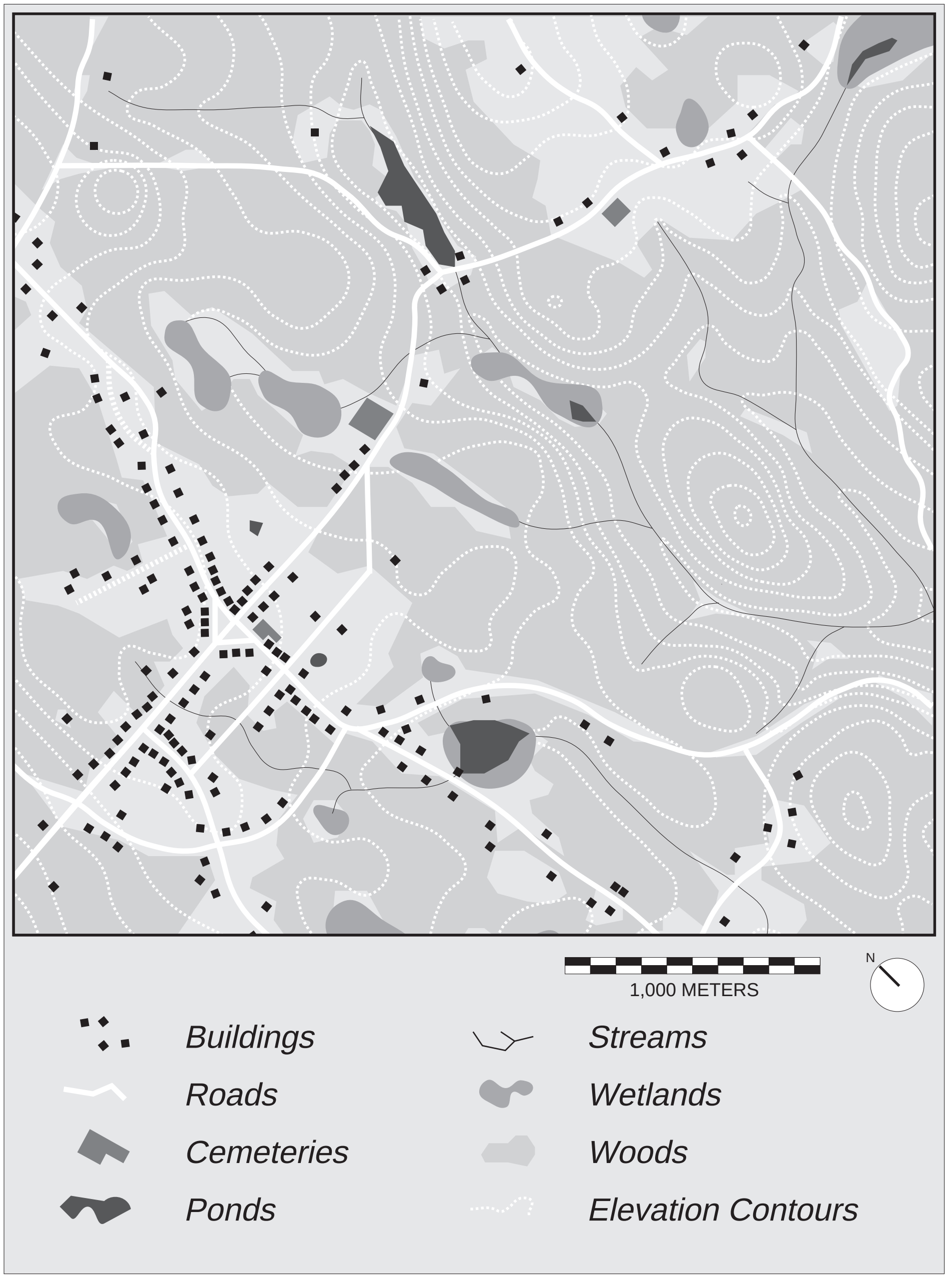
       ```
       LocalRating of map1 with 1 for 1 and
       map2 for 0
       ```

     - Boolean intersection of map1 with map2:

       ```
       LocalRating of map1 with 0 for 0 and
       map2 for 1
       ```

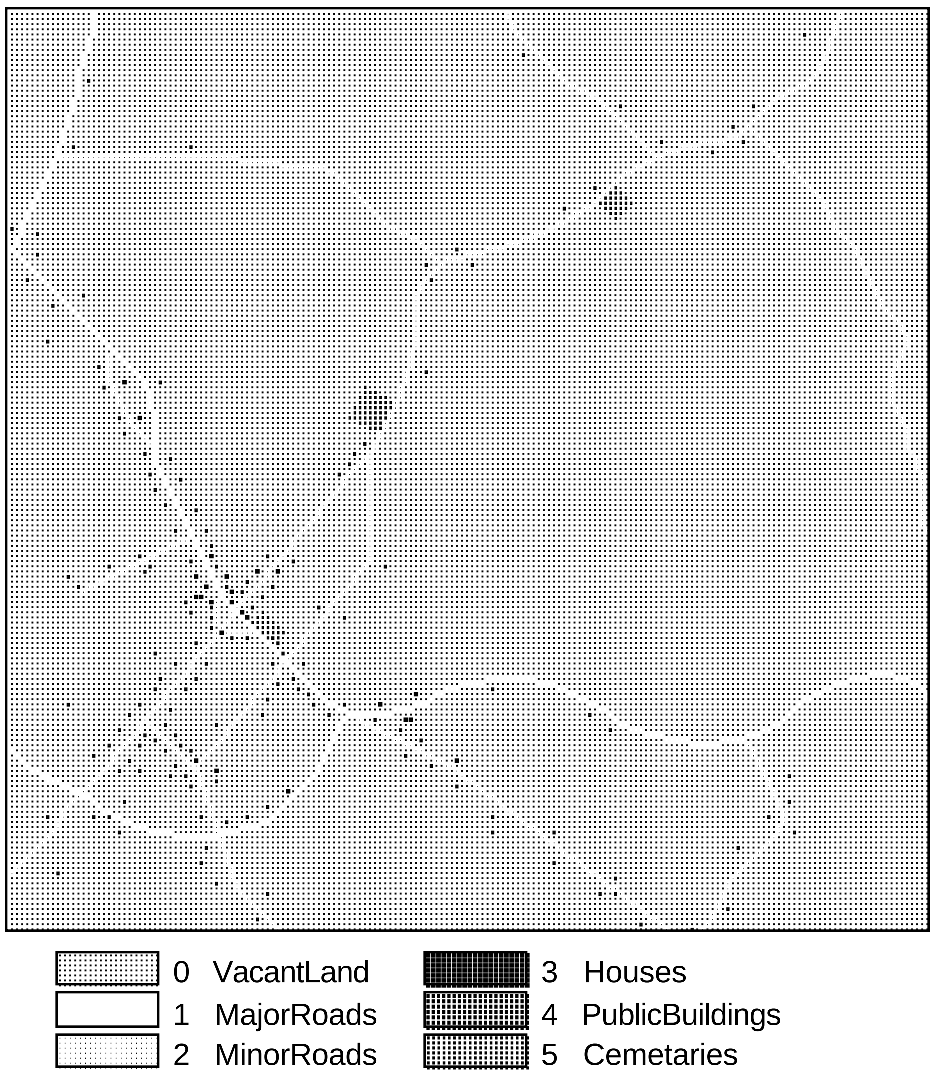
EXAMPLE OF STUDY AREA

Figure 1-3 from [Toml90]



DEVELOPMENT MAP

Figure 1-12 from [Toml90]

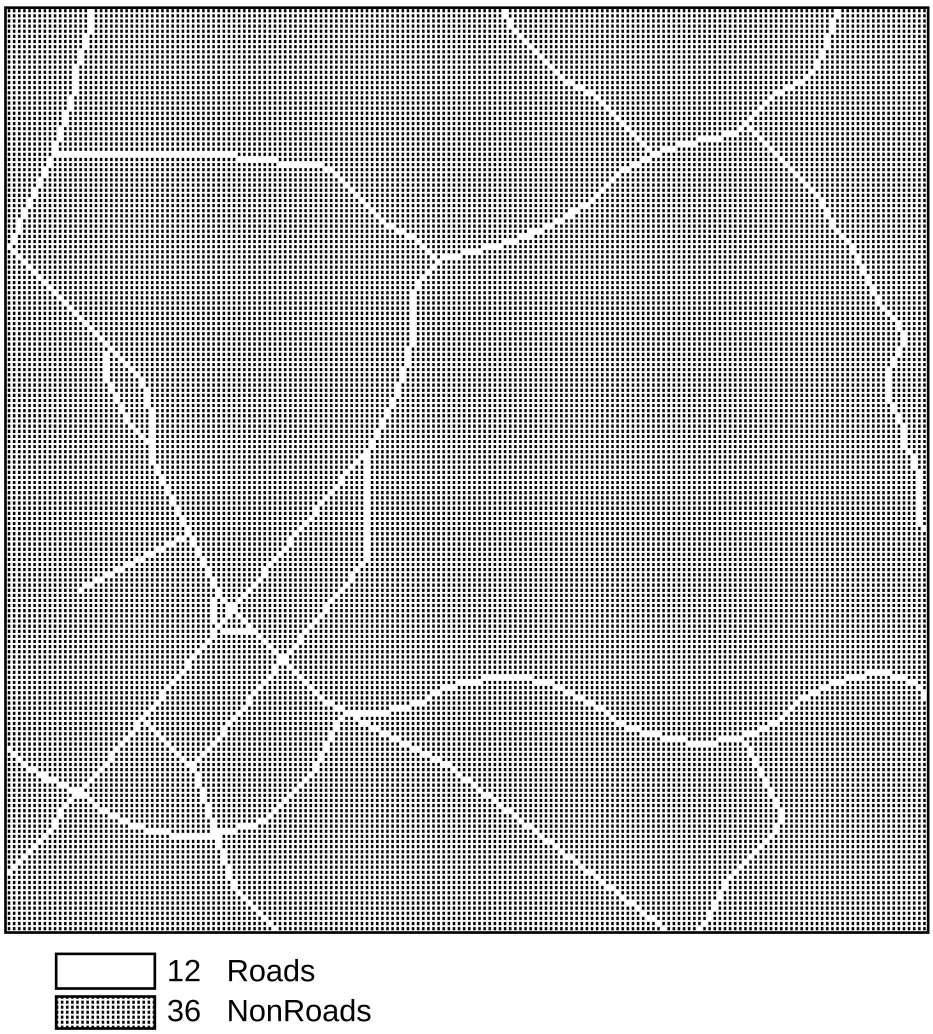


EXAMPLE OF RECODING/RECLASSIFICATION

Figure 4-2 from [Toml90]

```
Mobility = LocalRating of Development with
              12 for 1 2 with 36 for 0 3 4 5
```

Mobility Map



- Indicates locations lying along major or minor roads

POLYGON OVERLAY

- Apply the specified function to obtain a `DESTINATION` layer value based on the values associated with corresponding locations in two or more `SOURCE` layers
- Apply predefined function if possible
- Use `LocalRating` to specify an explicit correspondence between a `DESTINATION` layer value and each possible combination of `SOURCE` layer values (or ranges of values)

  Ex:
```
WindExposure = LocalRating of Altitude on
               Vegetation
               with 0 for 290… on 0
               with 1 for 290… on 1…3
               with 2 for …289 on 0 1 3
               with 3 for …289 on 2
```

- Result is a subset (because not all the possible pairs are specified) of the Cartesian product of the `SOURCE` layers
- Similar to a *spatial join* in relational database terminology where the join condition is coverage of the same group of grid squares

POLYGON OVERLAY WITH `LocalCombination`

- If range of `SOURCE` layer values is large

  1. `LocalRating` is cumbersome to use as all possible combinations must be specified

  2. many combinations never arise

     - 2 maps with 4 and 5 zones apiece

     - 20 possible destination values

  3. use `LocalCombination`

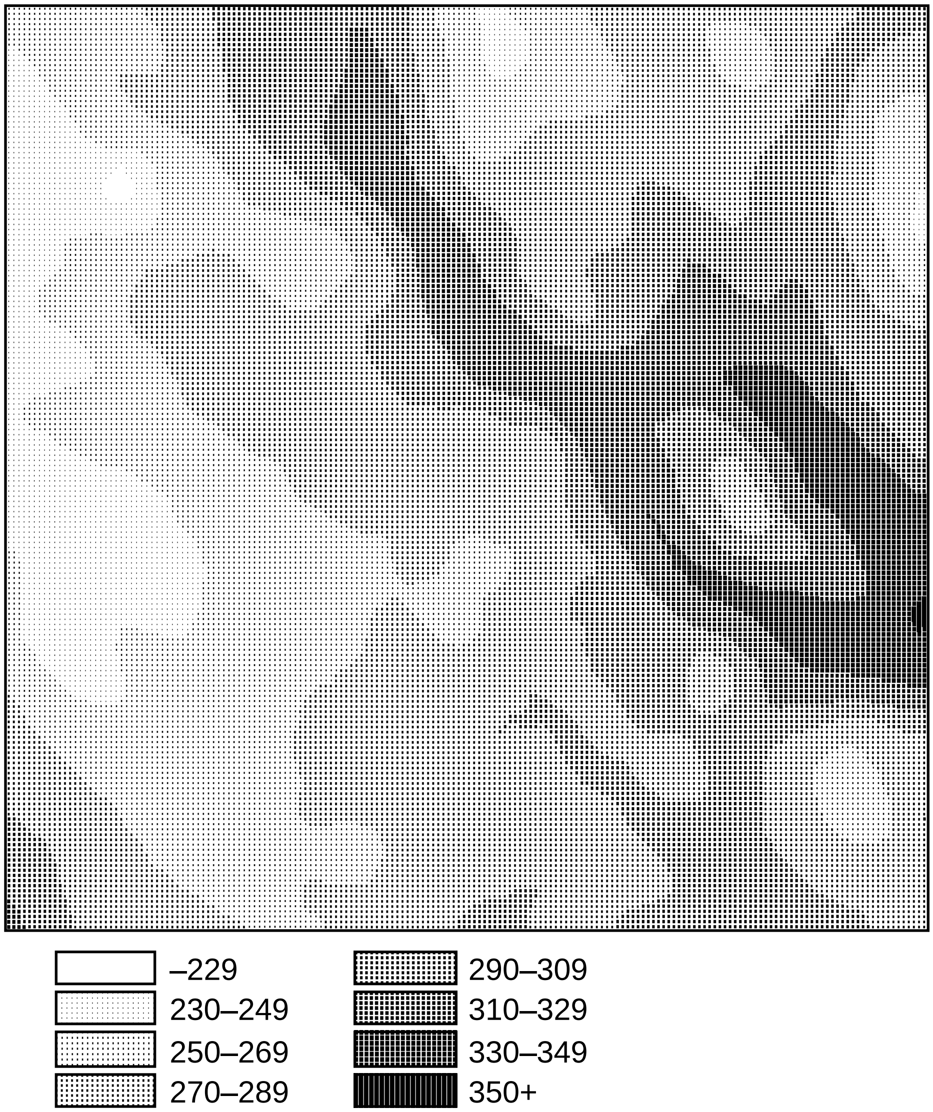
     - assigns a unique new value to each combination of values that actually occurs

     Ex: `LandCover = LocalCombination of Water and Vegetation and Development`

- True Cartesian product as all possible pairs or triples are permitted

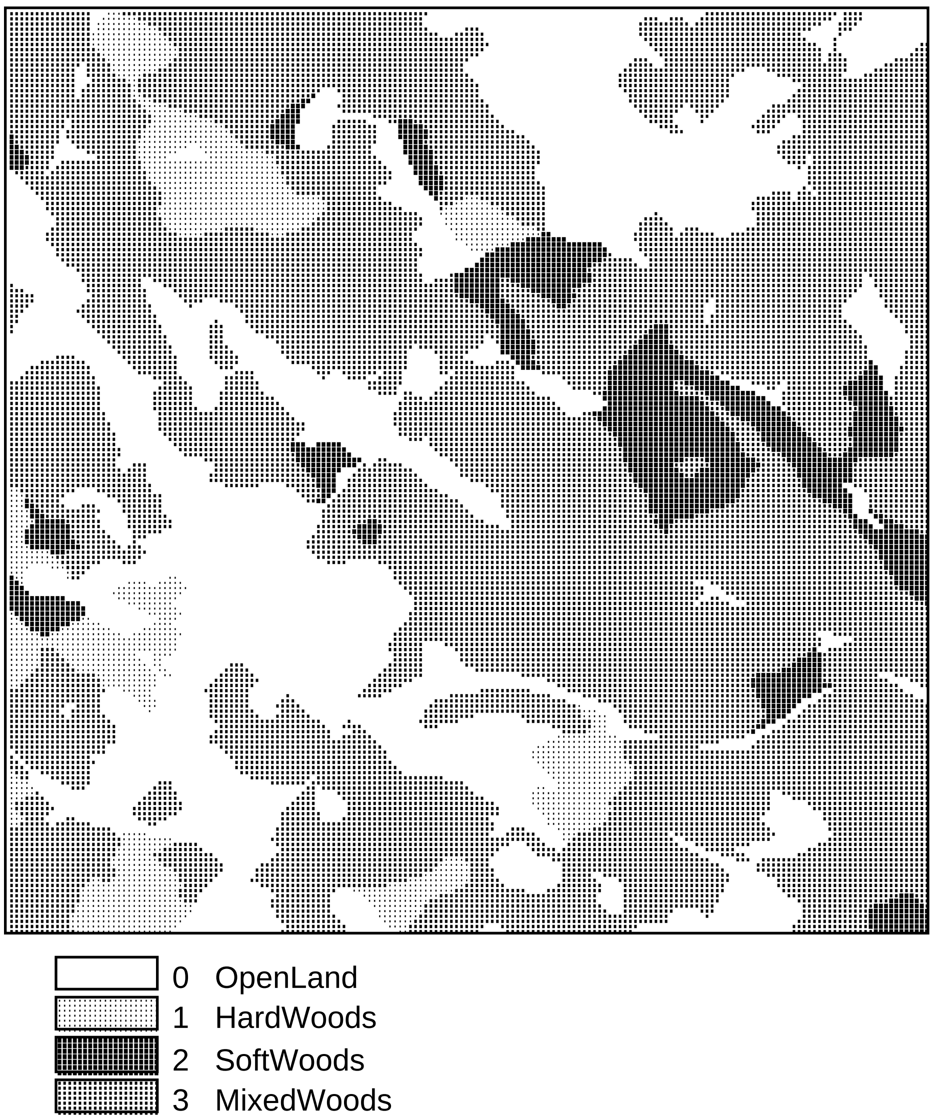
ALTITUDE MAP

Figure 1-9 from [Toml90]



VEGETATION MAP

Figure 1-11 from [Toml90]

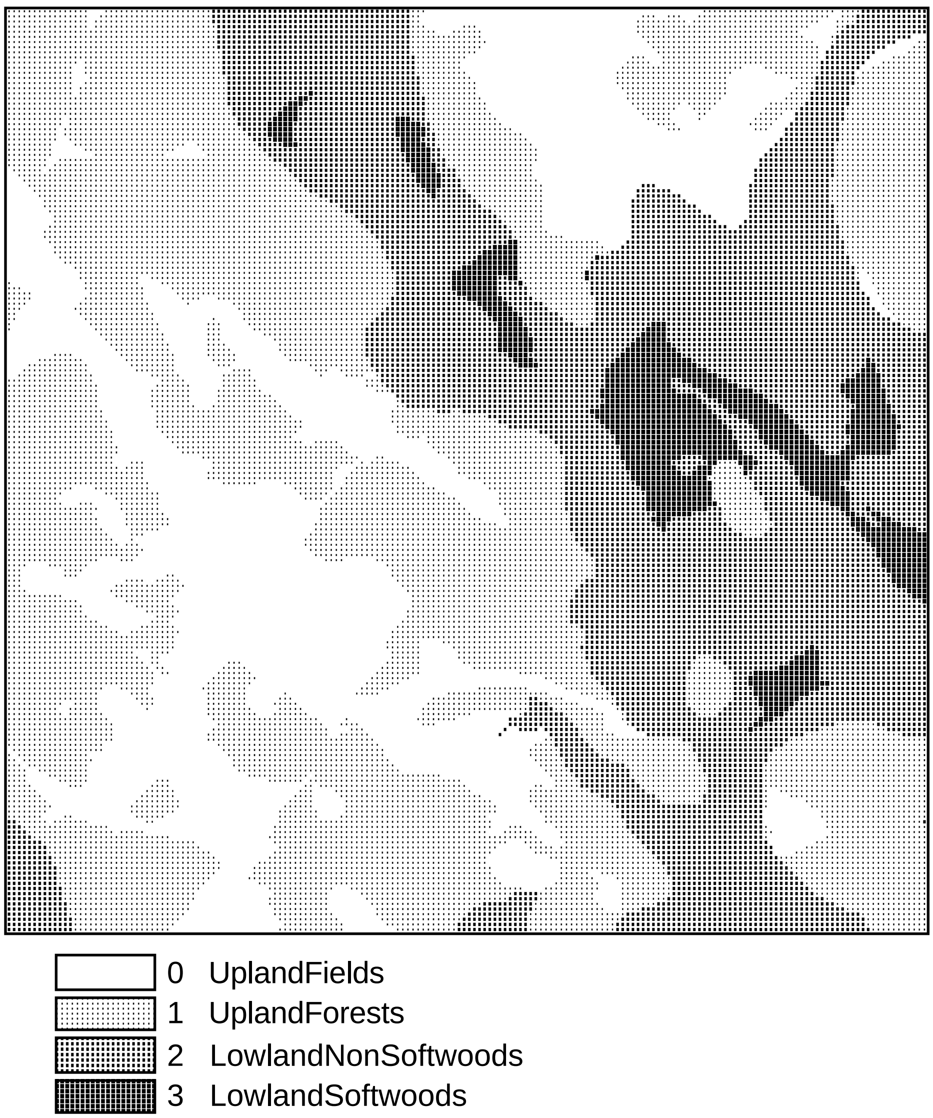


`LocalRating` EXAMPLE

Figure 4-7 from [Toml90]

```
WindExposure = LocalRating of Altitude and Vegetation
               with 0 for 290… on 0 with 1 for 290… on 1…3
               with 2 for …289 on 0 1 3 with 3 for …289 on
               2
```

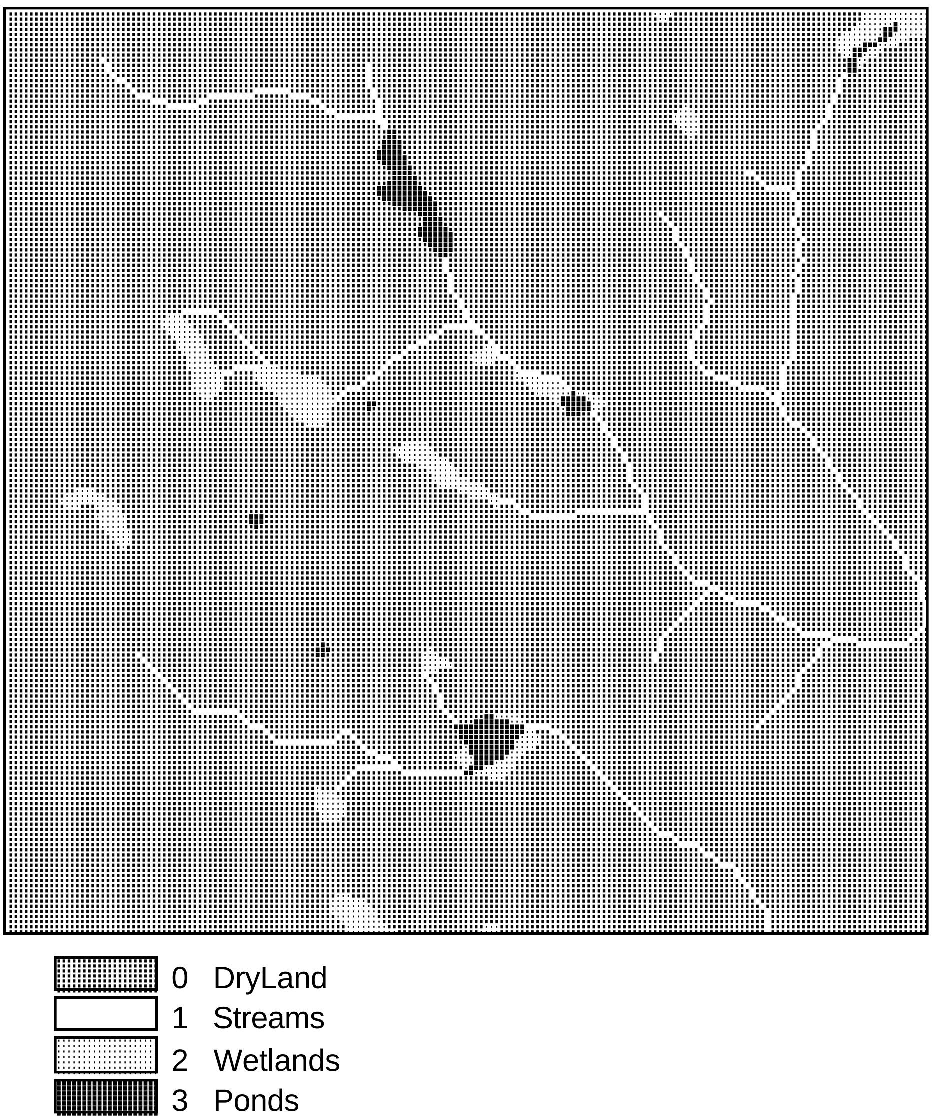
WindExposure Map



- Distinguishes uplands with and without tree cover from lowlands with and without softwood tree cover

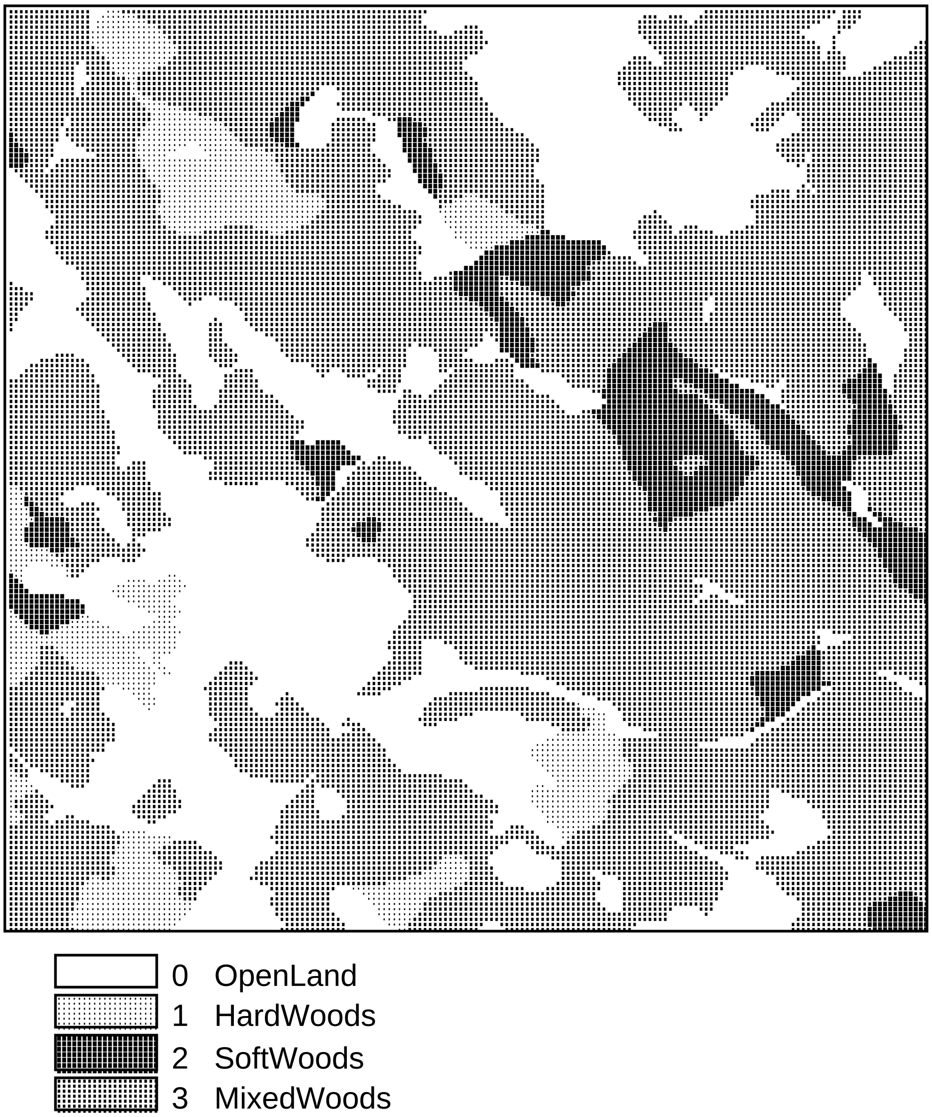
WATER MAP

Figure 1-10 from [Toml90]



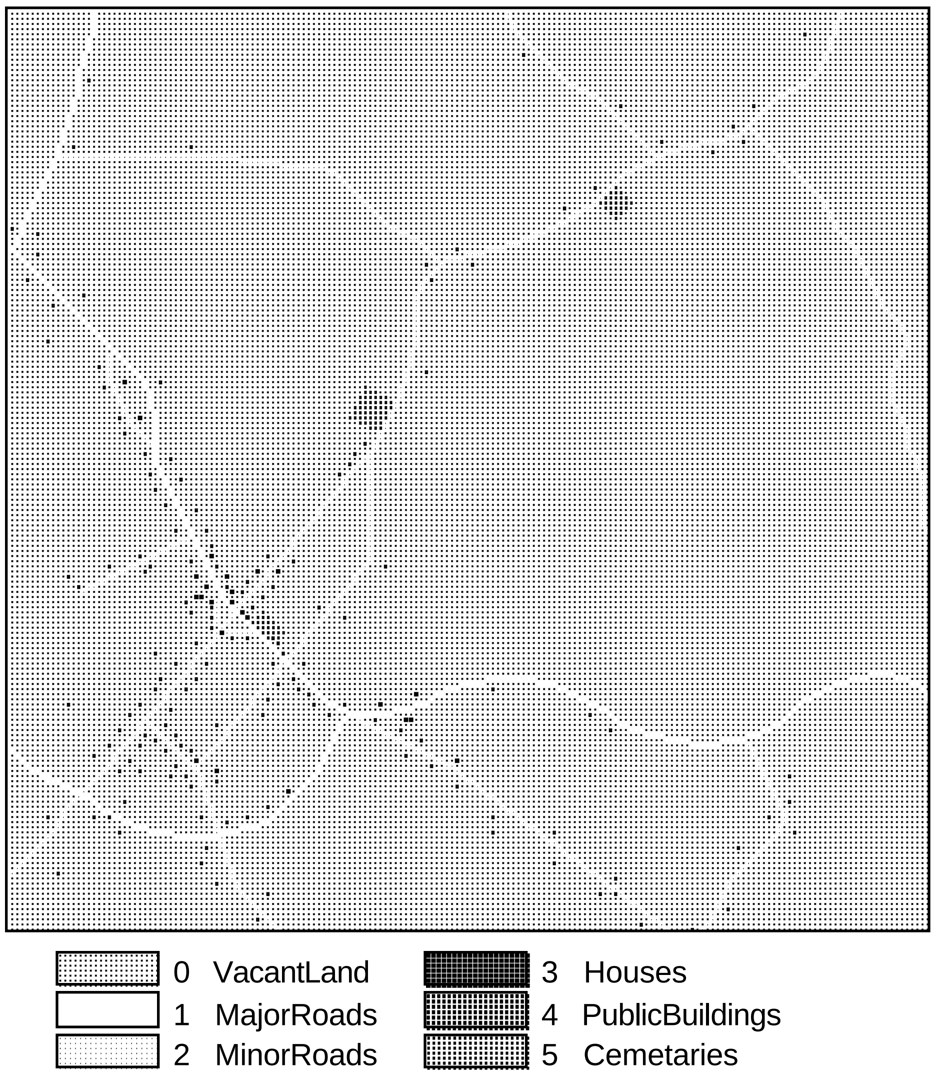
VEGETATION MAP

Figure 1-11 from [Toml90]



DEVELOPMENT MAP

Figure 1-12 from [Toml90]

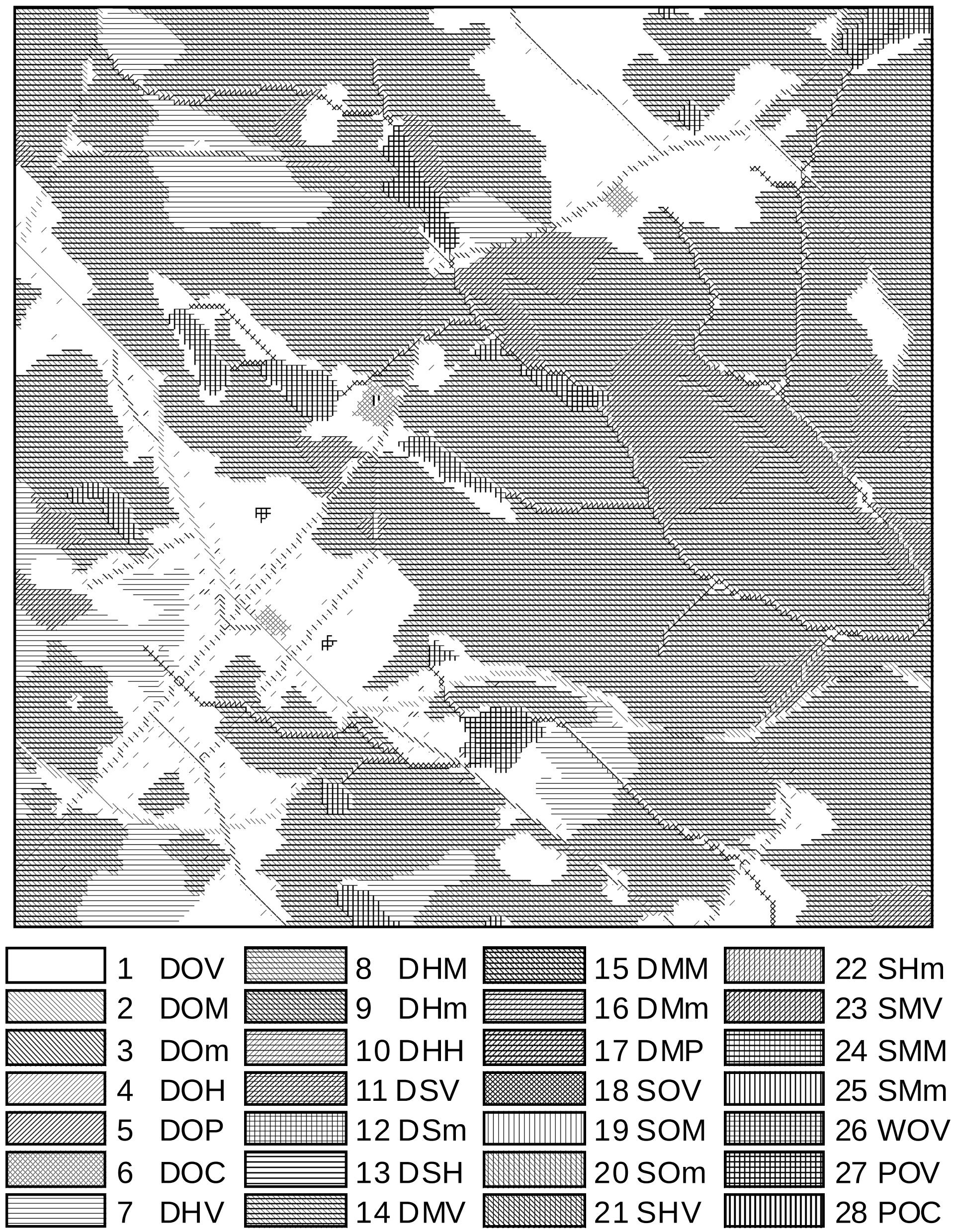


`LocalCombination` EXAMPLE

Figure 4-8 from [Toml90]

`LandCover` = `LocalCombination of Water and Vegetation and Development`

LandCover Map



- Indicates combinations present in each location

ZONAL OPERATIONS

Ex: `DESTINATIONLAYER = ZonalFunction of SOURCELAYER [within MASKLAYER]`

- one map is a crop map (zones-like attributes)
- other map yields average rainfall per location
- result is average rainfall per crop

1. Yields a partition of the SOURCE map layer into zones corresponding to the MASK layer where the value of each location in a zone of the DESTINATION map layer is the result of the application of the operation to all the locations in the zone in the SOURCE map layer
   - one value for each location in the SOURCE layer but they need not be distinct
   - if MASK layer is omitted, then assume it to be 1 zone
2. Operation is zonal with respect to partition induced by MASK layer while involving values in the SOURCE layer
   - ignores the partition induced by the SOURCE layer
   - ignores the values in the MASK layer
   - contrast with general polygon overlay operation where DESTINATION layer consists of a set of new zones which correspond to a subset of the Cartesian product of the zones in the two SOURCE layers and hence does not ignore the values in either layer
3. Results can be such that all locations in a zone
   - will have the same value
   - need not have the same value

EXAMPLES OF ZONAL OPERATIONS

1. Functions yielding a constant value in each zone

   - examples are `maximum`, `minimum`, `mean`, `product`, `sum`, etc.
   - unlike local operations, the value of a zone in the `DESTINATION` layer is a function of all of the differing values in the zone's corresponding locations in the `SOURCE` layer

     Ex: `ZonalSum` is the sum of the values associated with the zone's locations in the `SOURCE` layer
   - `ZonalVariety` gives the number of different values
   - `ZonalCombination`

     a. related to polygon overlay

     b. indicates for each zone in the `MASK` layer the collection of different values associated with the corresponding locations in the `SOURCE` layer

Ex:

```
EveryBlock    = LocalRating of Development
                with –1 for …5 with –0 for 1 2
EachBlock     = FocalInsularity of EveryBlock
Housing       = LocalRating for Development
                with 0 for … with 1 for 3
HomesPerBlock = ZonalSum for Housing within
                EachBlock
HomesPerBlock = LocalRating of HomesPerBlock
                with 0 for –0 with 10 for 10…
```

ZONAL EXAMPLE

Figure 1-12 from [Toml90]

Development Map

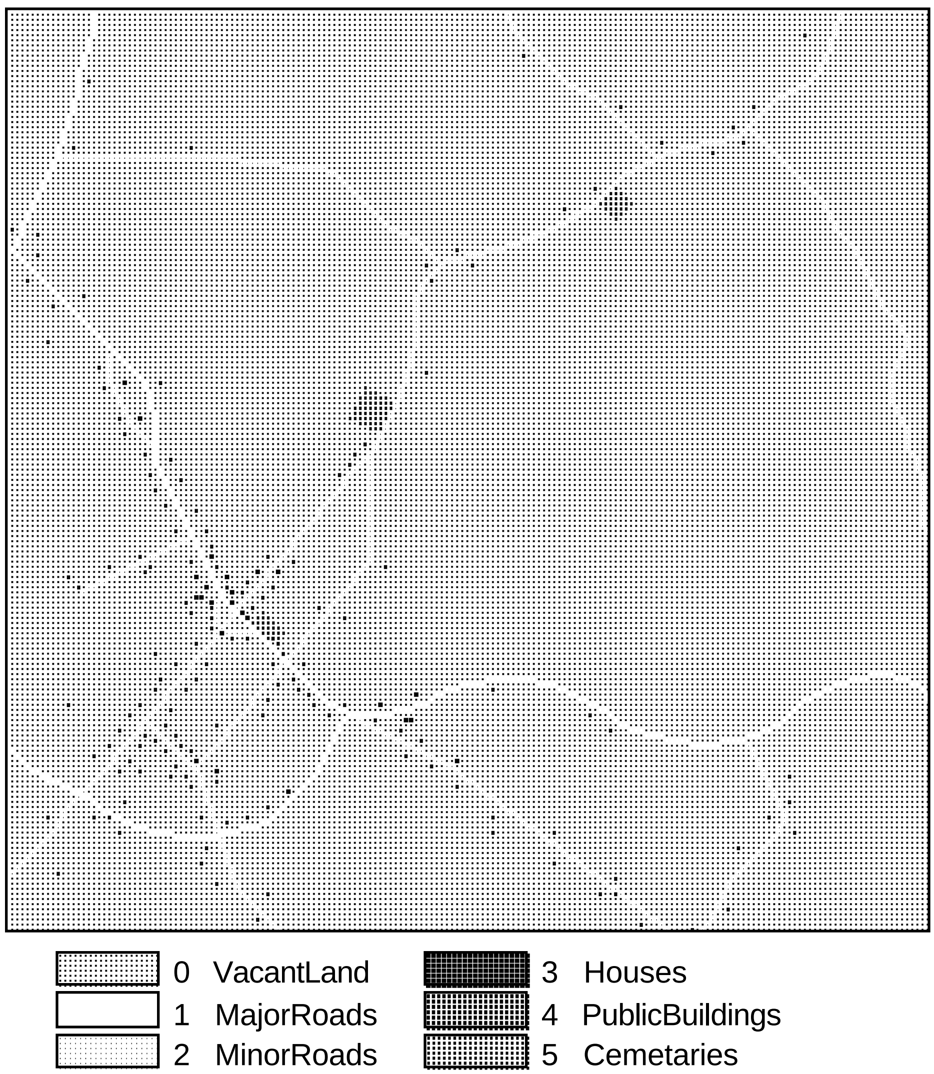
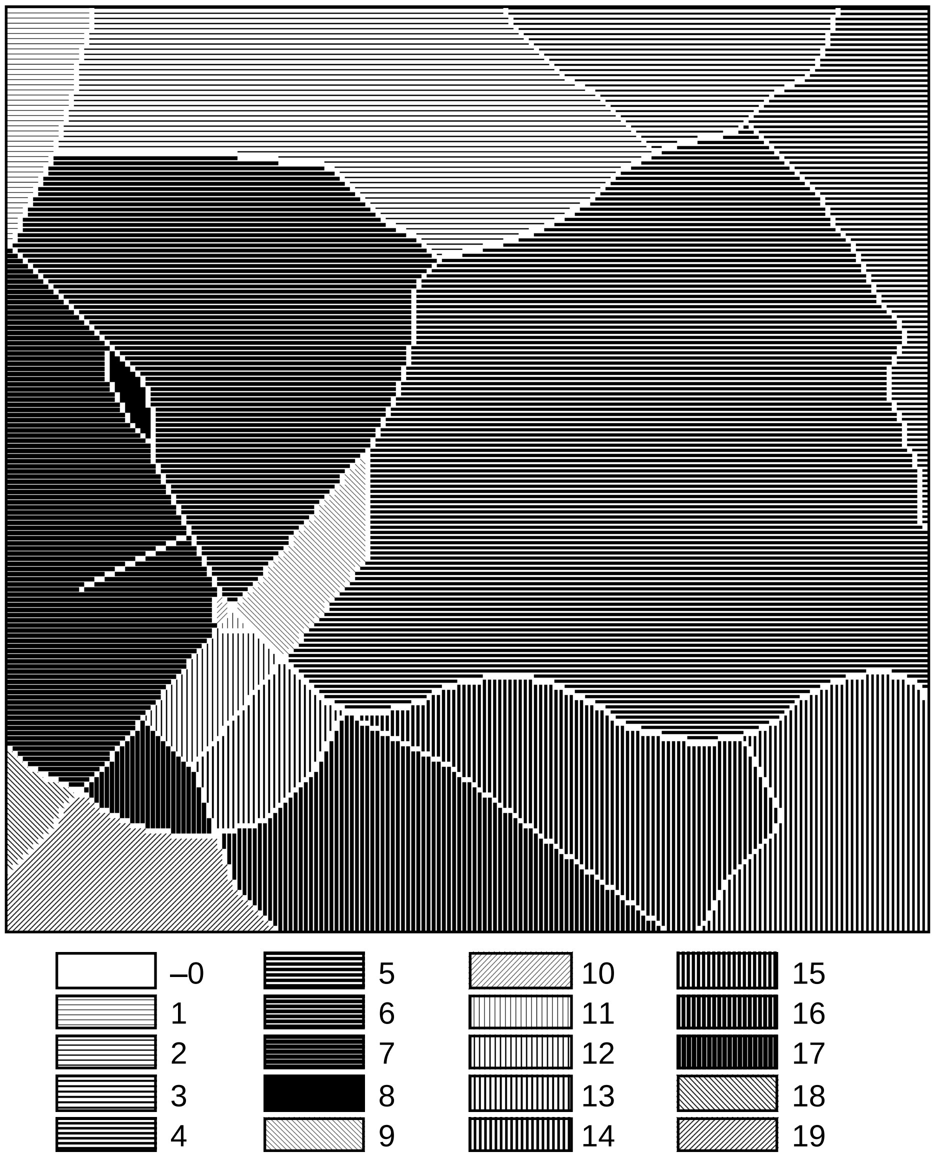


Figure 5-5 from [Toml90]

```
EveryBlock = LocalRating of Development with –1 for …5
             with –0 for 1 2
EachBlock  = FocalInsularity of EveryBlock
```

EachBlock Map

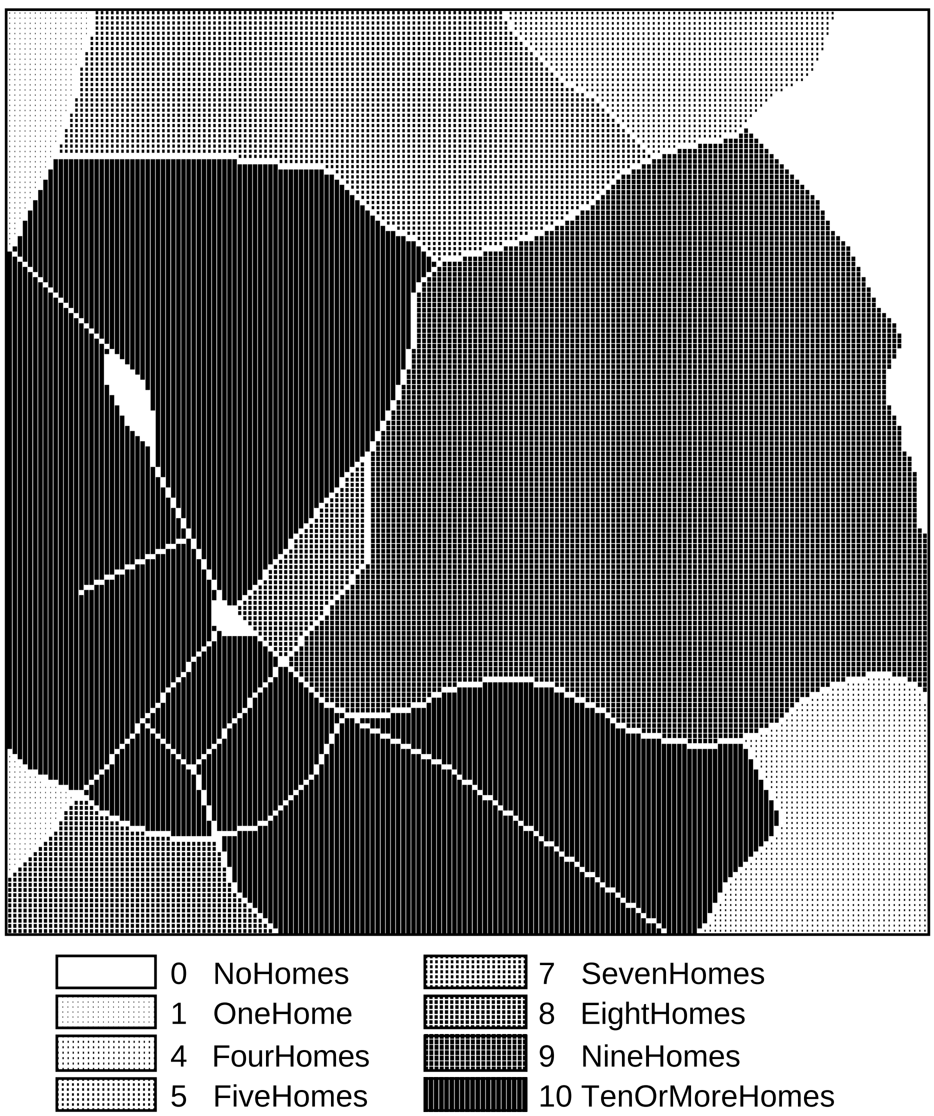


- FocalInsularity identifies all locations with the same value (i.e., connected component labeling)
- Indicates all continuous expanses of land between roads

Ex: Figure 6-2 from [Toml90]

```
Housing       = LocalRating of Development with
                0 for … with 1 for 3
HomesPerBlock = ZonalSum of Housing within
                EachBlock
HomesPerBlock = LocalRating of HomesPerBlock
                with 0 for –0 with 10 for 10…
```

HomesPerBlock Map



2. Functions whose result is not necessarily constant over a zone (termed *partial* zones)

   - contrast the value of each location *P* in the `SOURCE` layer to a statistic summarizing the values of all locations in the zone containing *P*

   - examples are percentage, percentile, ranking

     general example:

     given location *L* in zone *Z* with value *V*, how many values less than *V* are associated with locations in *Z*? (this is the new value associated with each location *L*)

     specific example:

     given a value of the rainfall at each location, rank the locations in a zone (e.g., where wheat is grown) on the basis of rainfall in the zone

INADEQUACY OF ZONES

- Some zonal operations don’t yield the desired results when zones are not contiguous

  Ex: input layer contains elevations and we want the maximum difference in elevations between two locations in the same zone. Clearly, all locations in the zone should be contiguous

- May need operations to deal with the interrelationship between a zone and its non-contiguous components
- Problem: concept of a zone indicates

  1. spatial contiguity, AND
  2. non-spatial contiguity

- Solution:

  1. operational:

     apply procedure that uniquely labels each similarly-valued location in a contiguous area (i.e., connected component labeling or a `FocalInsularity` operation)

  2. conceptual:

     define an additional grouping hierarchy that lies between a location and a zone which consists of contiguous locations

CLUMP

Def: additional grouping hierarchy lying between a location and a zone consisting of contiguous locations

- Result: a two-level aggregation hierarchy
- Zonal operation becomes:

Ex:
```
DESTINATIONLAYER = ZonalFunction of SOURCELAYER
                   [in CLUMPMASKLAYER]
                   [within ZONALMASKLAYER]
```

1. some operations only require a `CLUMPMASK` layer
   - Ex: maximum elevation difference in each crop region
2. some operations only require a `ZONALMASK` layer
   - Ex: average rainfall for each soil type
3. some operations require both a `CLUMPMASK` and a `ZONALMASK` layer
   - having both a `CLUMPMASK` and a `ZONALMASK` layer makes it easy to determine if two distinct locations that are in the same zone are in the same clump (i.e., component) of the zone
   - useful to have both `CLUMPMASK` and `ZONALMASK` layers if want to know how many non-contiguous regions make up each zone
4. given a `ZONALMASK` layer, there is no need for a `CLUMPMASK` layer
   - `CLUMPMASK` layer information can be obtained by applying connected component labeling (`FocalInsularity`)

NEIGHBORHOOD (FOCAL) OPERATIONS

- Concerned with relationships between locations on same map layer instead of between instances of a location on different map layers
- Neighborhood is a set of one or more locations
  1. within a specified distance of a given location
     - physical separation
     - reachability
     - trafficability
     - time
     - visibility (line of sight, field of view)
  2. within a specified direction of a given location (bearing)
     - all directions if no direction or range is specified
  3. within a distance containing a specified number of occurences of items with a particular attribute value
  4. location is termed *neighborhood focus*
- Extent of neighborhoods
  1. immediate (i.e., adjacent locations) and no distance or direction is specified
  2. extended
  3. distance and direction can be specified using maps in which case the neighborhood varies from location to location
  4. could be defined by a zone but need an additional zonal map layer which may be cumbersome if zones are not contiguous (e.g., field of view) since a search could be involved

IMMEDIATE NEIGHBORHOOD OPERATIONS

1. Similar to image processing counterparts (e.g., convolution, filtering, smoothing)

2. Similar to local and zonal operations

   - e.g., minimum, maximum, mean, product, sum, combination, variety

   - difference is that they are applied to values of multiple locations in the same layer rather than to the same location on different map layers as in local operations

   - `FocalCombination` indicates the combination of values that occur within each location's neighborhood

     Ex: `ShoreType` = `FocalCombination of Water`

   - `FocalRating` reclassifies each location's value according to the combination of values in its neighborhood

     a. a null value is associated with all unspecified combinations

     b. equivalent to a `FocalCombination` followed by a `LocalRating` (i.e., composition)

     Ex: `WoodsEdge` = `FocalRating of Vegetation with 1 for 0 1 with 2 for 0 2 with 3 for 0 3`

3. Summarize the neighborhood's values

   - accentuate the differences (e.g., `maximum` or `minimum`), OR

   - smooth the differences (e.g., `mean`, `sum`, `product`, `rating`)

   - can also be achieved by filtering—i.e., moving a window across the entire map

4. Identify all adjacent locations with the same value

   - `FocalInsularity`

   - assigns a unique number to each group of spatially contiguous locations that have the same value

   - known as connected component labeling in image processing and implies a flood-filling approach to its implementation

FILTERING

- Move a window across entire map
- Value for a location in middle of window is a weighted average of remaining locations in the window
- Sum of weights should add to 1
- Two types

1. low pass filter smooths the values (i.e., removes or reduces local detail)

   Ex: average with neighbors—severe smoothing

.11	.11	.11
.11	.11	.11
.11	.11	.11

   Ex: slight smoothing

.06	.06	.06
.06	.52	.06
.06	.06	.06

2. high pass filter enhances edges (i.e., exaggerates the differences or local detail)

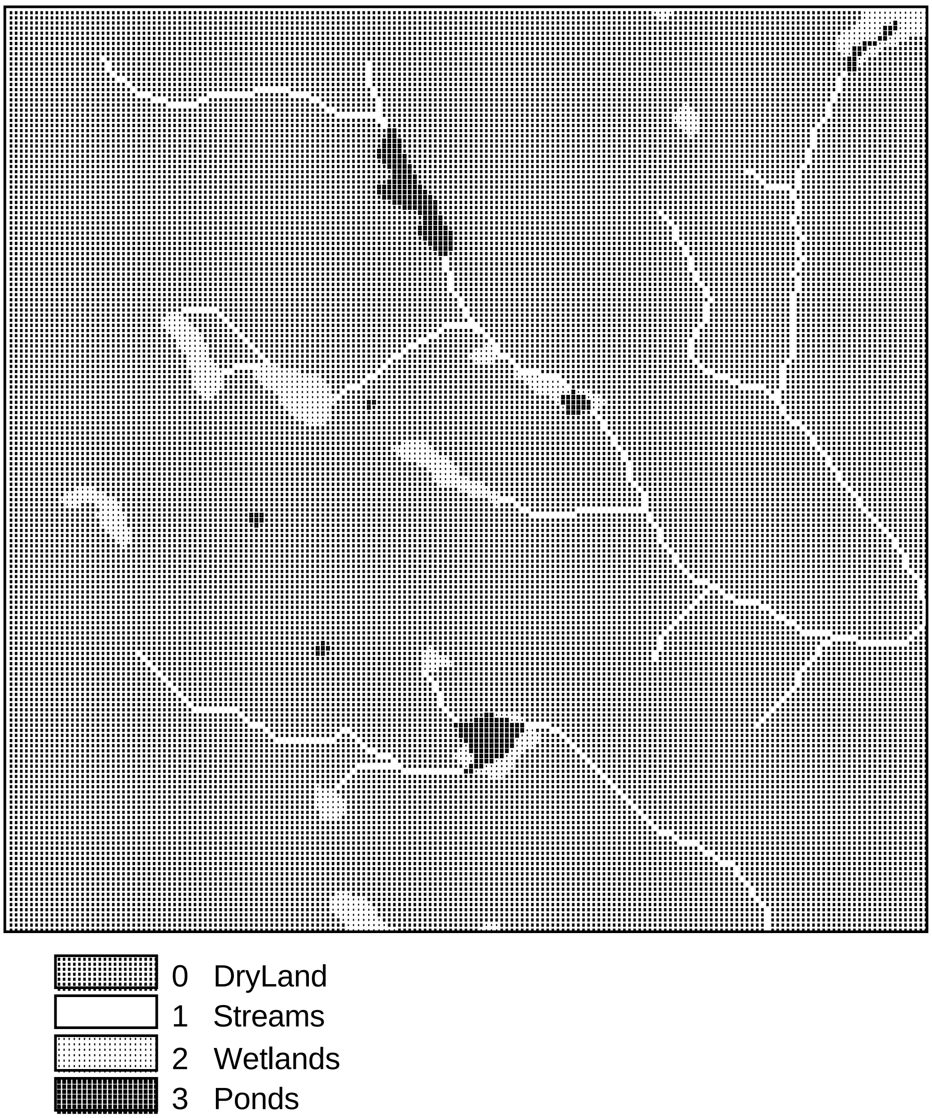
   Ex: some enhancement by removing effect of neighbors

–.06	–.06	–.06
–.06	1.48	–.06
–.06	–.06	–.06

`FocalCombination` EXAMPLE

Figure 1-10 from [Toml90]

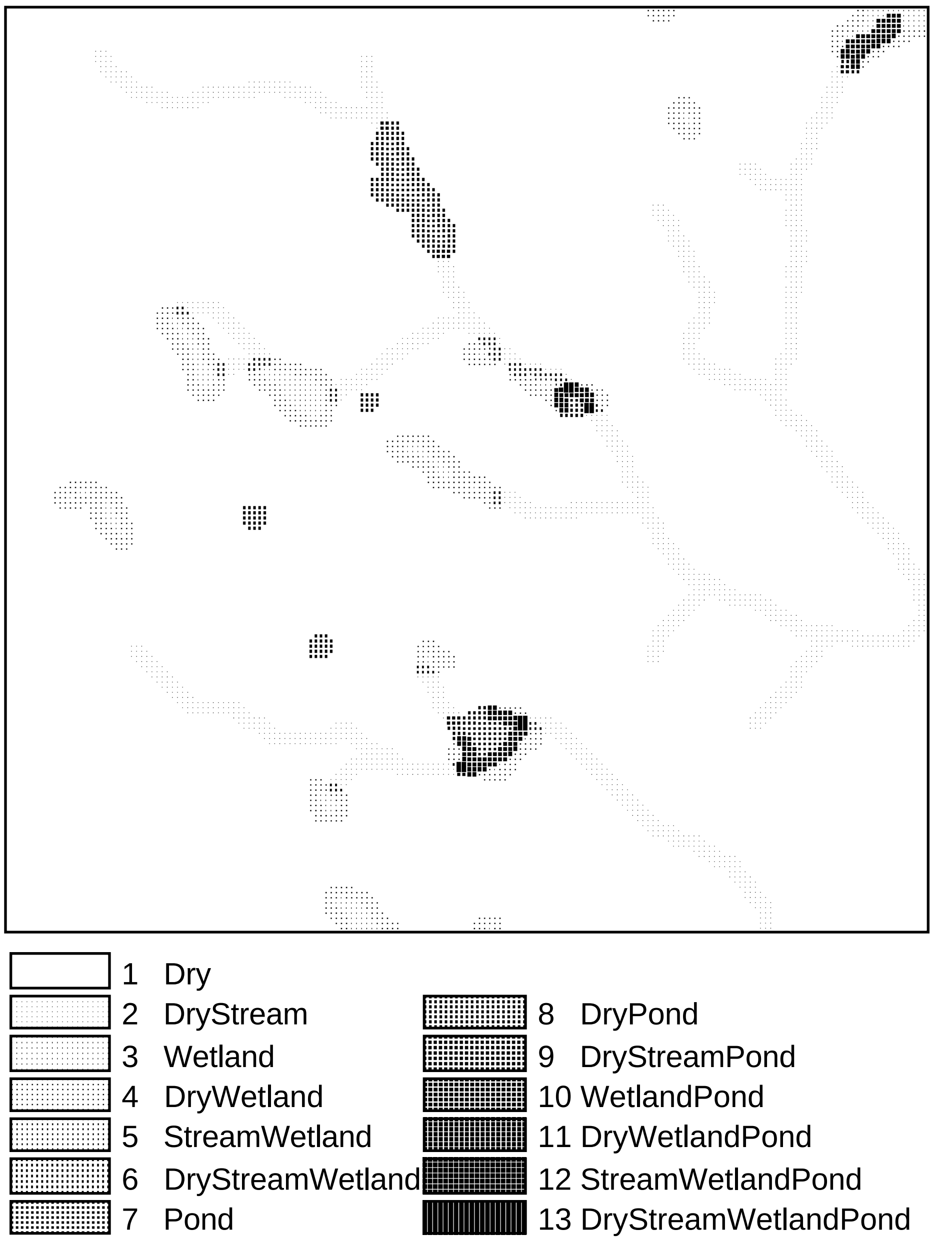
Water Map



Ex: Figure 5-2 from [Toml90]

`ShoreType = FocalCombination of Water`

ShoreType Map

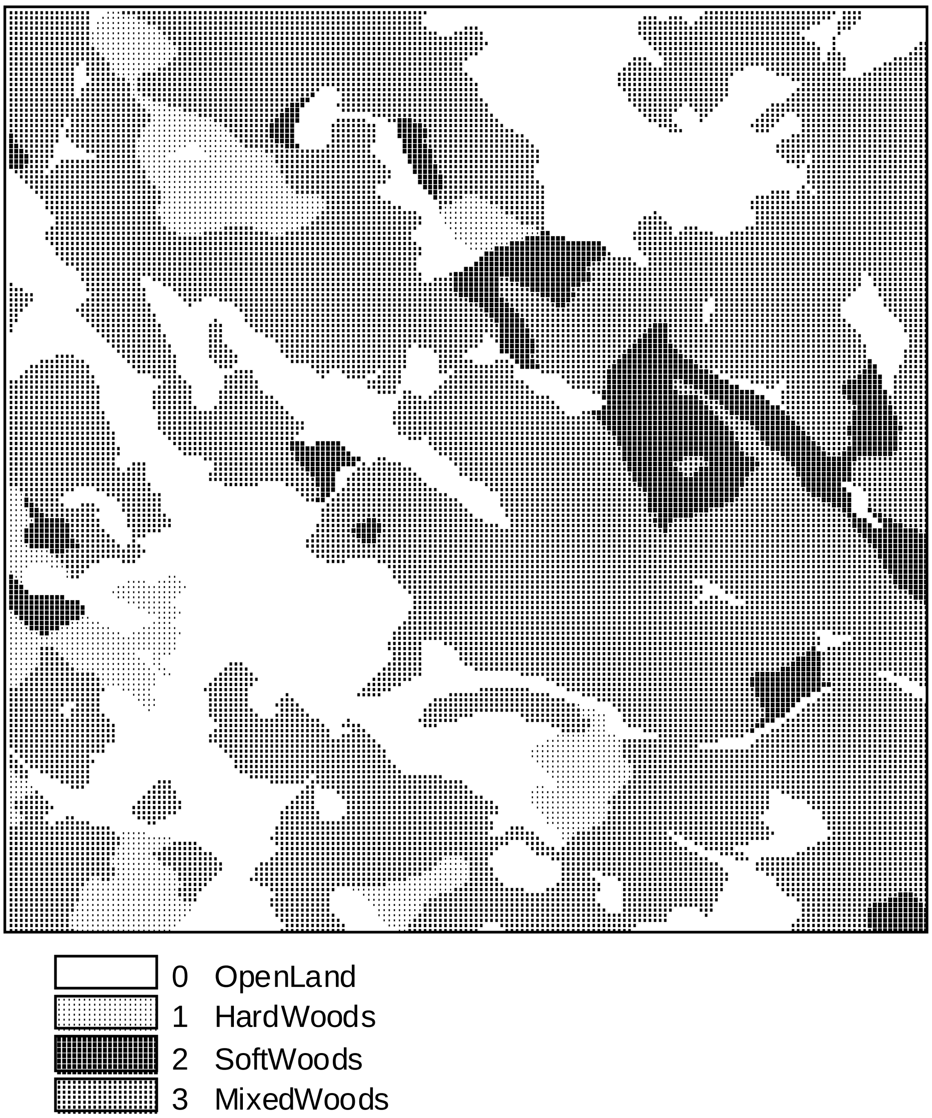


- Indicates the combination of water zones that occur in the immediate vicinity of each location

`FocalRating` EXAMPLE

Figure 1-11 from [Toml90]

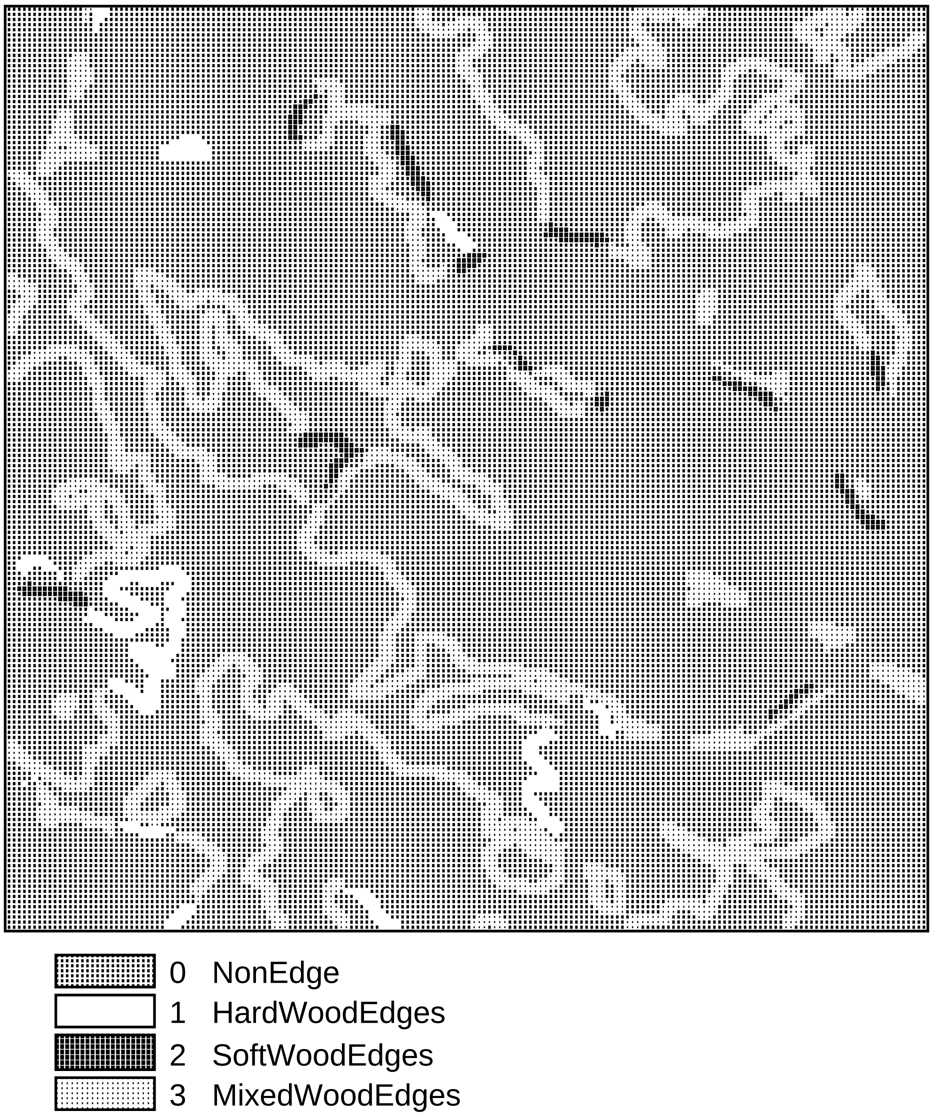
Vegetation Map



Ex: Figure 5-3 from [Toml90]

```
WoodsEdge = FocalRating of Vegetation with 1
            for 0 1 with 2 for 0 2 with 3
            for 0 3
```

WoodsEdge Map

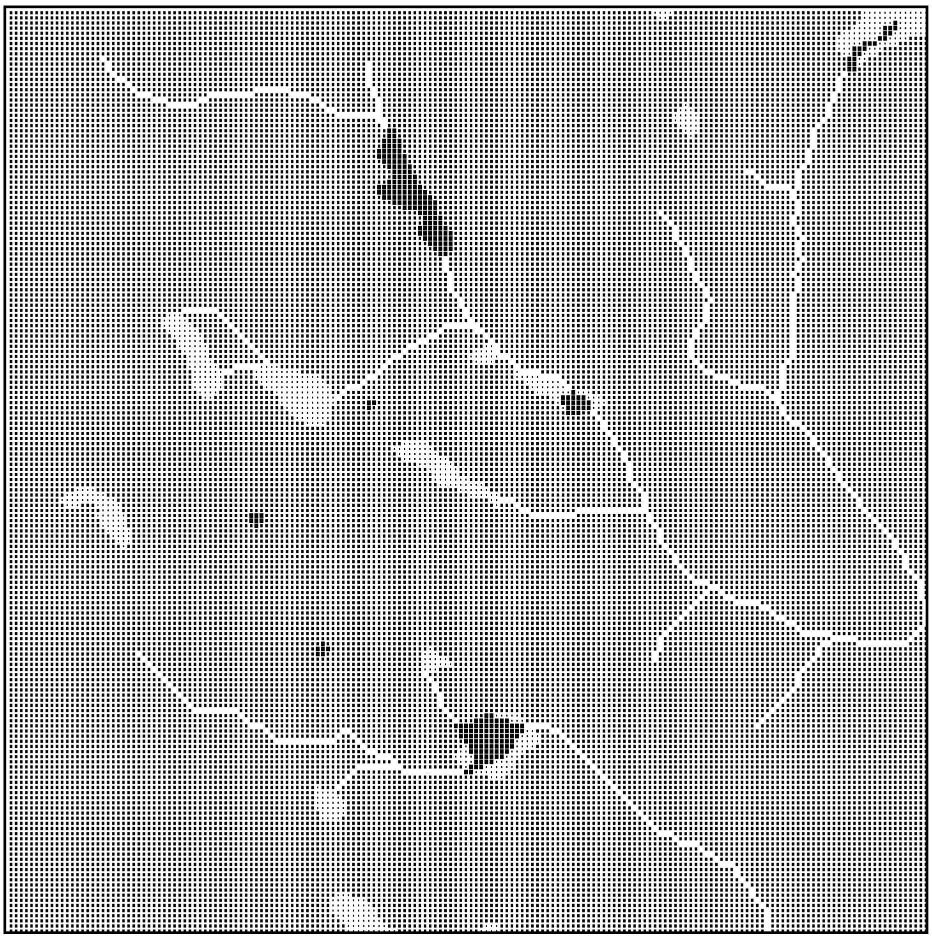


- Indicates locations where open land is immediately adjacent to forests

`FocalInsularity` EXAMPLE

Figure 1-10 from [Toml90]

Water Map

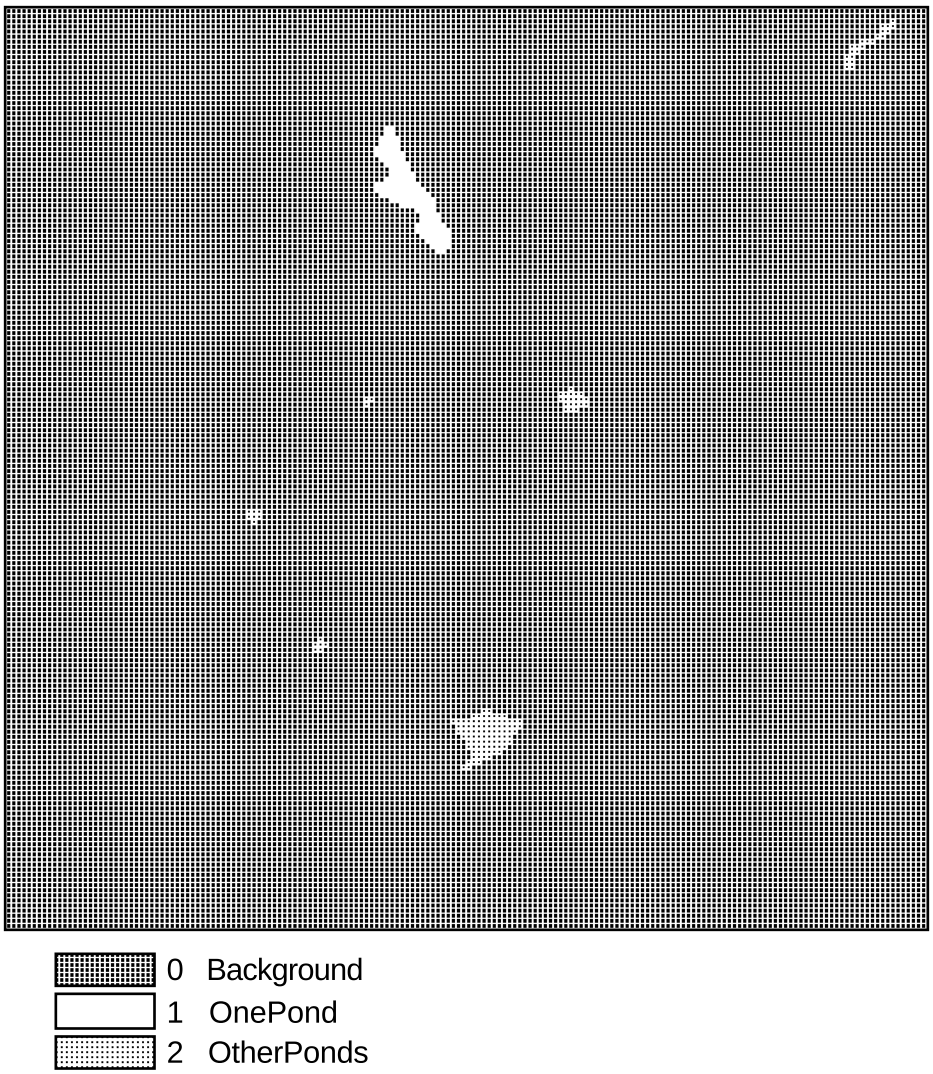


0 DryLand
1 Streams
2 Wetlands
3 Ponds

Figure I-1 from [Toml90]

```
EveryPond = LocalRating of Water with –0 for 0…2
EachPond  = FocalInsularity of EveryPond
WhichPond = LocalRating of EachPond with 0 for
            –0 with 2 for 1…6 with 1 for 2
```

WhichPond Map

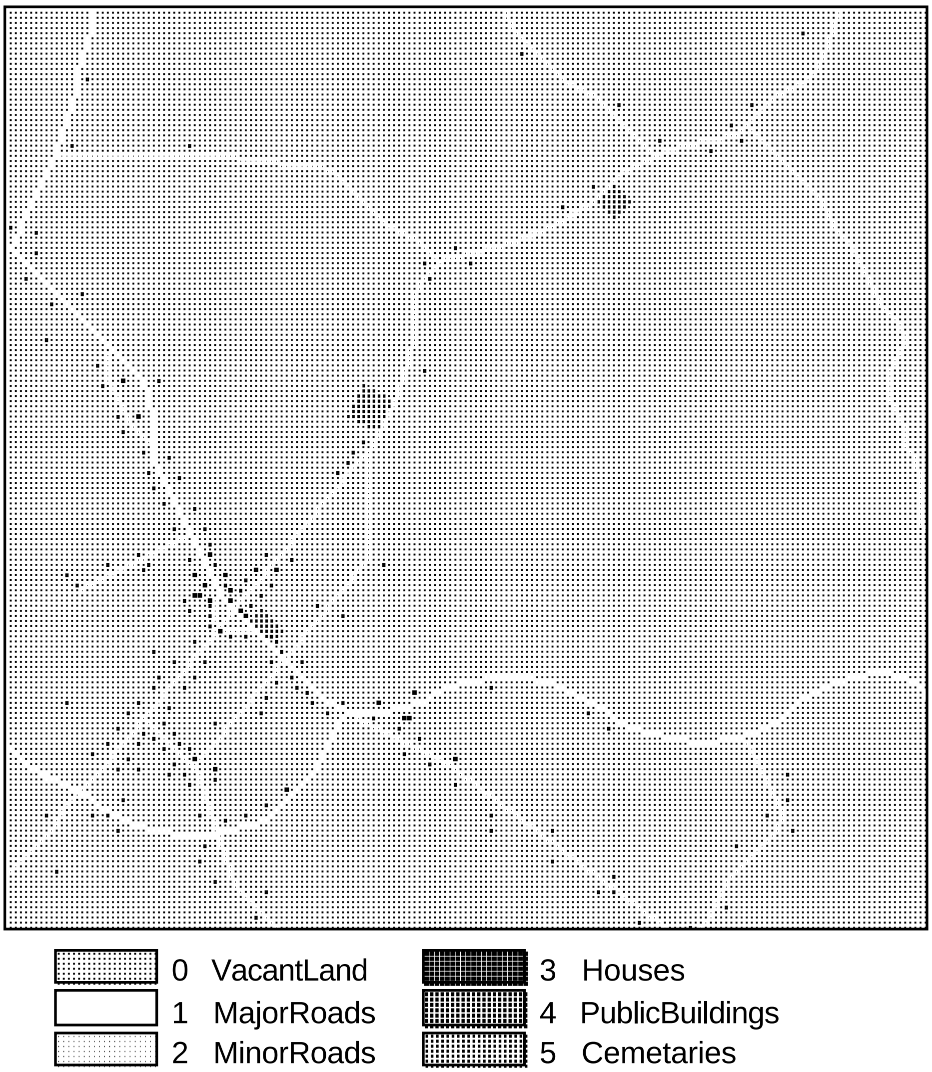


- Distinguishes one of the ponds from the rest of the ponds and the non-pond areas

ADDITIONAL `FocalInsularity` EXAMPLE

Figure 1-12 from [Toml90]

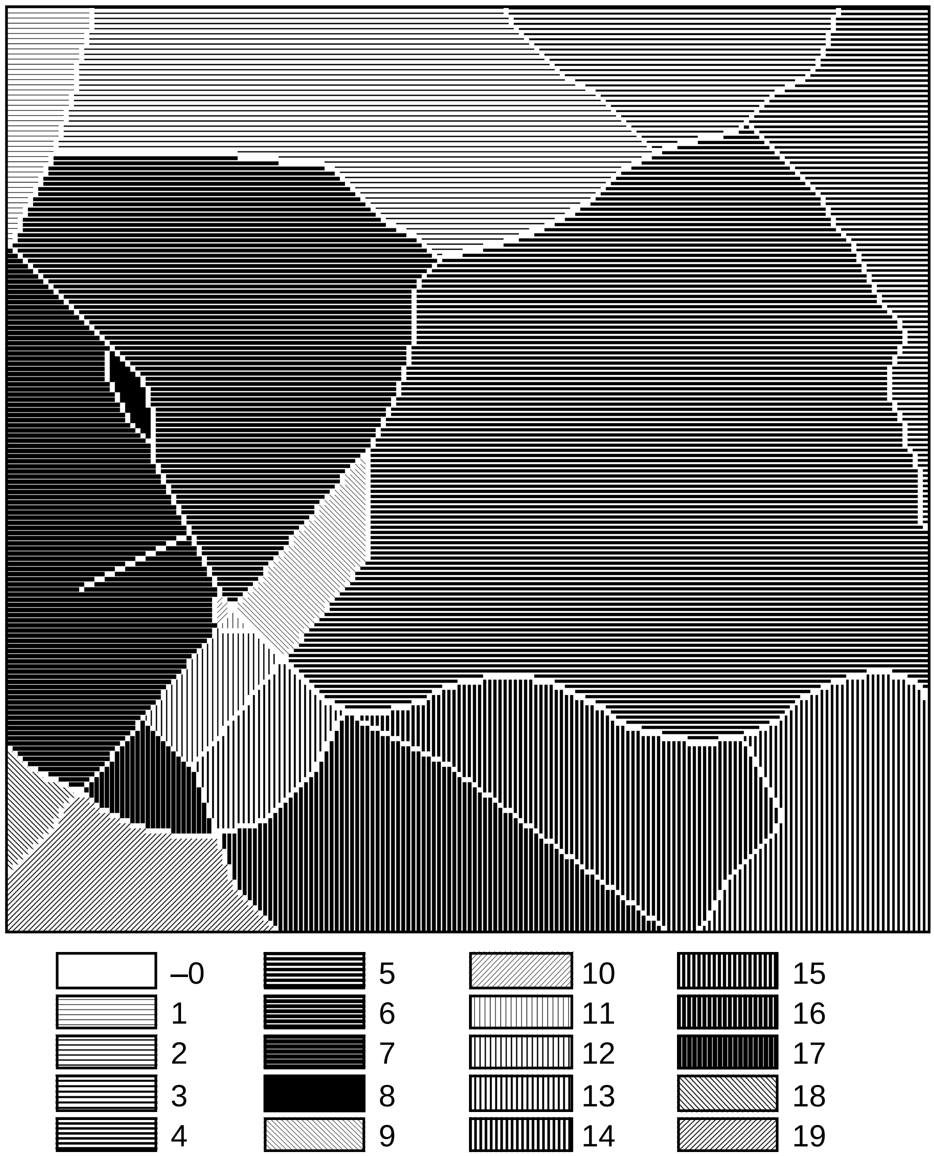
Development Map



Ex: Figure 5-5 from [Toml90]

```
EveryBlock = LocalRating of Development with –1 for …5
             with –0 for 1 2
EachBlock  = FocalInsularity of EveryBlock
```

EachBlock Map



- FocalInsularity identifies all locations with the same value (i.e., connected component labeling)
- Indicates all continuous expanses of land between roads

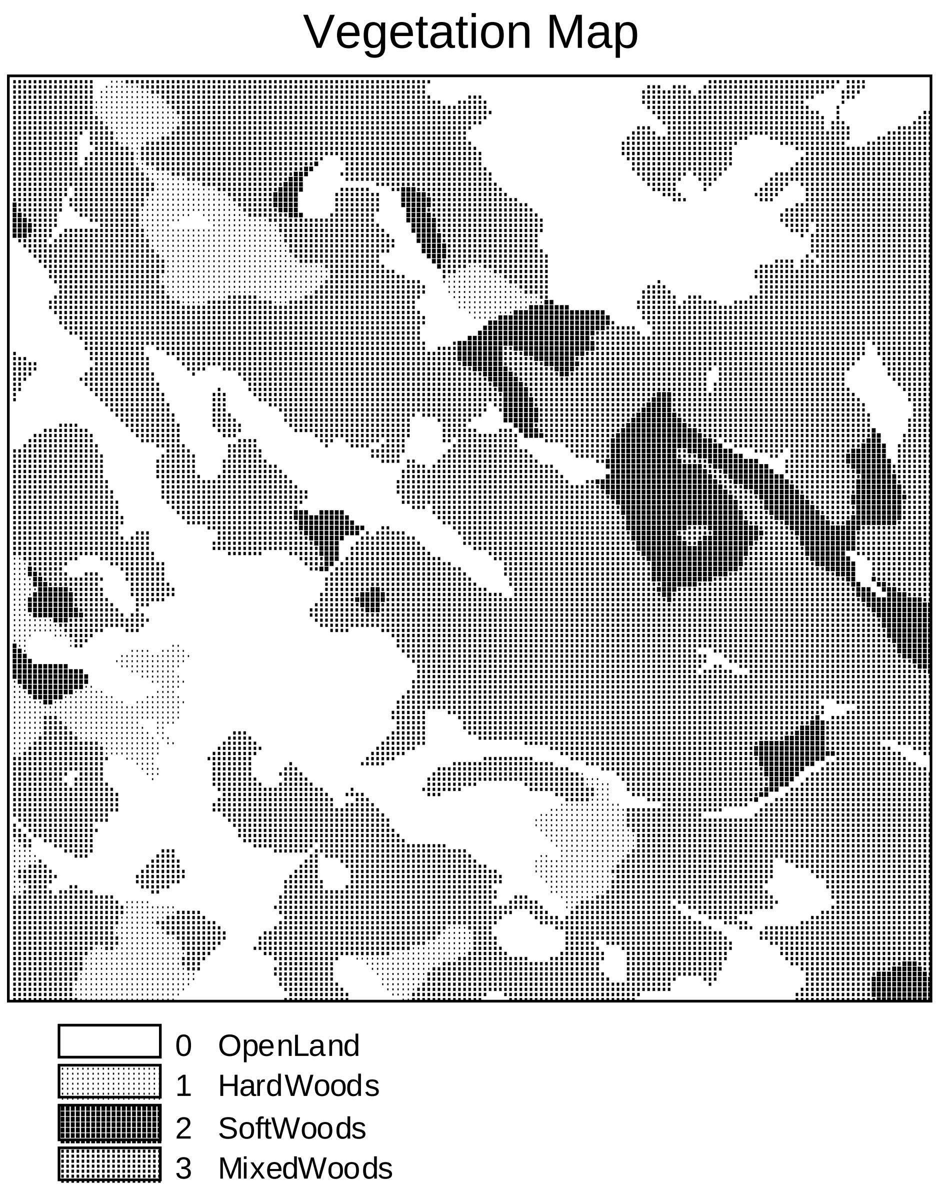
EXTENDED NEIGHBORHOOD OPERATIONS

1. Cartographic smoothing and predominant type determination
   - areal
   - surficial
     a. fill up basins and valleys
     b. wear down peaks and ridges
2. Determine distance to nearest non-null value location (`FocalProximity`)
   - specifying a distance, say *d*, constrains the search neighborhood and is used as the default distance if no location is found
   - known as buffer zone (corridor, image dilation, within)
   - useful in search operations
3. Determine bearing (1 to 360) of nearest non-null value location (`FocalBearing`)

4. Determine value of nearest non-null value location rather than distance or bearing (`FocalNeighbor`)
   - Ex: which fire station should respond to a call?
   - resulting decomposition of space is *Voronoi diagram* (also *Thiessen polygon*)
   - a form of cartographic interpolation (i.e., an estimate of unknown values) in that each null valued location is set to the nearest of a specified set of known values
   - like discrete interpolation
5. Determine weighted average value of nearest non-null value locations (`FocalGravitation`)
   - weight each known value by the inverse of the square of its distance from the neighborhood focus
   - a better estimate than `FocalNeighbor`
   - influence like gravity
   - like continous interpolation

AREAL SMOOTHING EXAMPLE

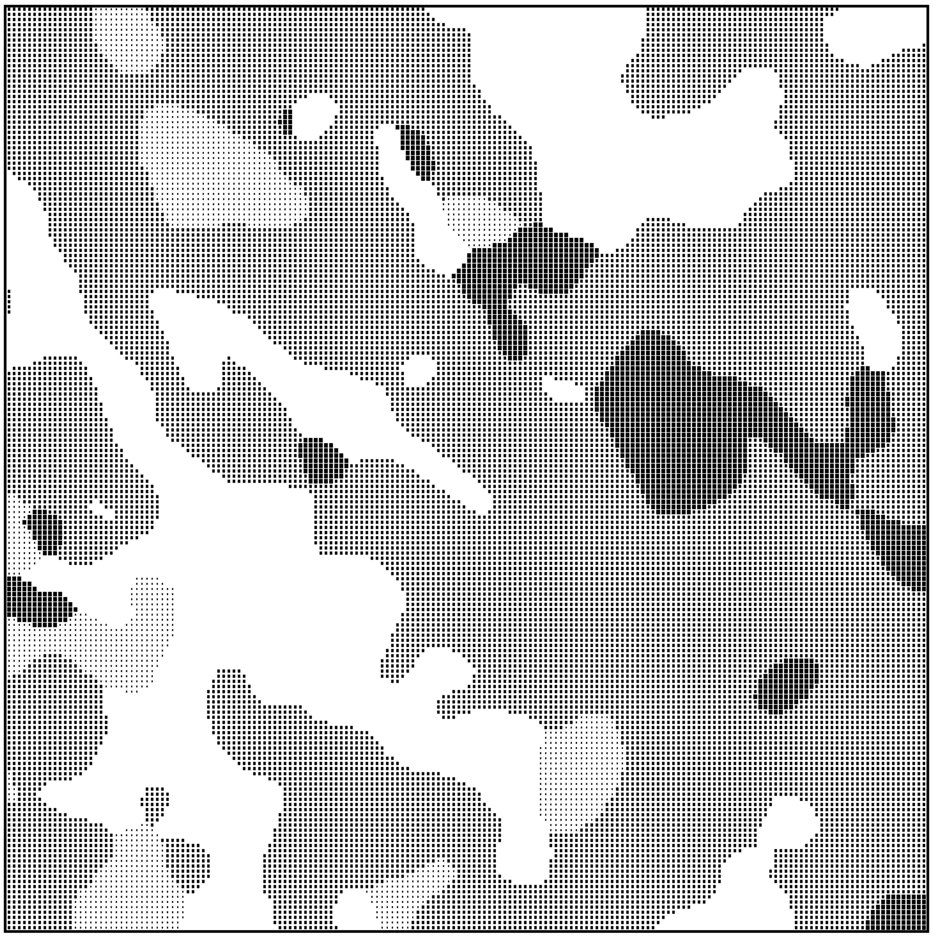
Figure 1-11 from [Toml90]



Ex: Figure 5-15 from [Toml90]

```
SmoothVegetation = FocalMajority of
                    Vegetation at …100
```

SmoothVegetation Map

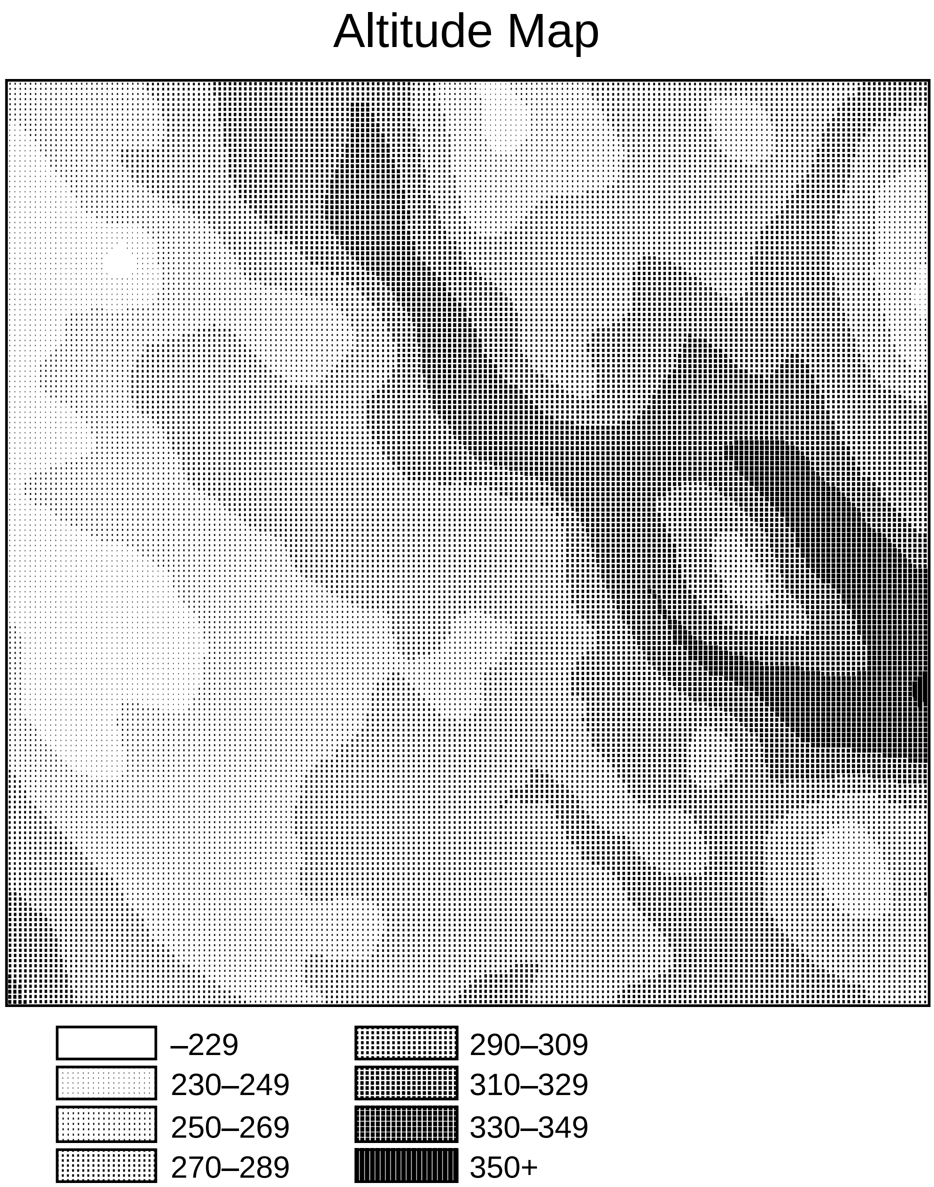


0 OpenLand
1 HardWoods
2 SoftWoods
3 MixedWoods

- Indicates the most frequent vegetation zone within 100 meters of each location

SURFICIAL SMOOTHING EXAMPLE

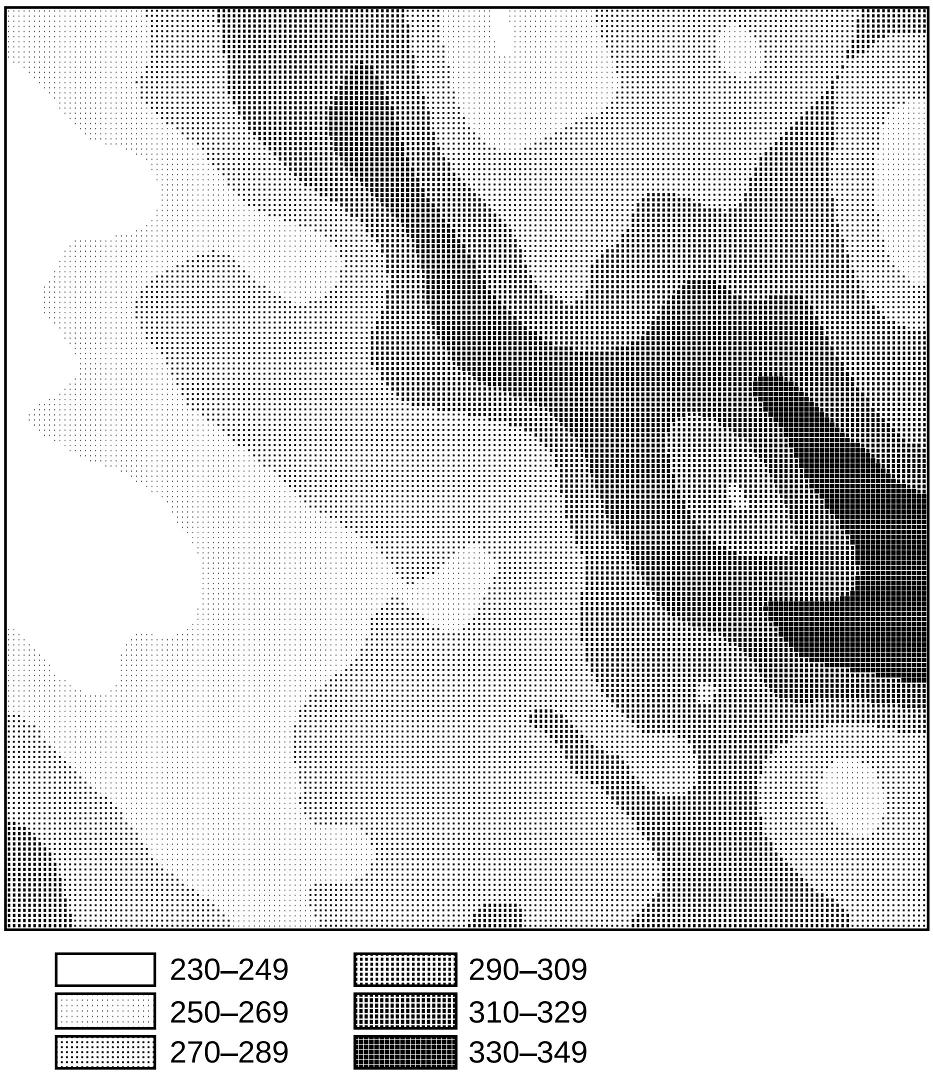
Figure 1-9 from [Toml90]



Ex: Figure 5-16 from [Toml90]

`SmoothAltitude` = `Altitude` and repeatedly apply:
`SmoothAltitude` = `FocalMean of SmoothAltitude`
`at …1000`

SmoothAltitude Map

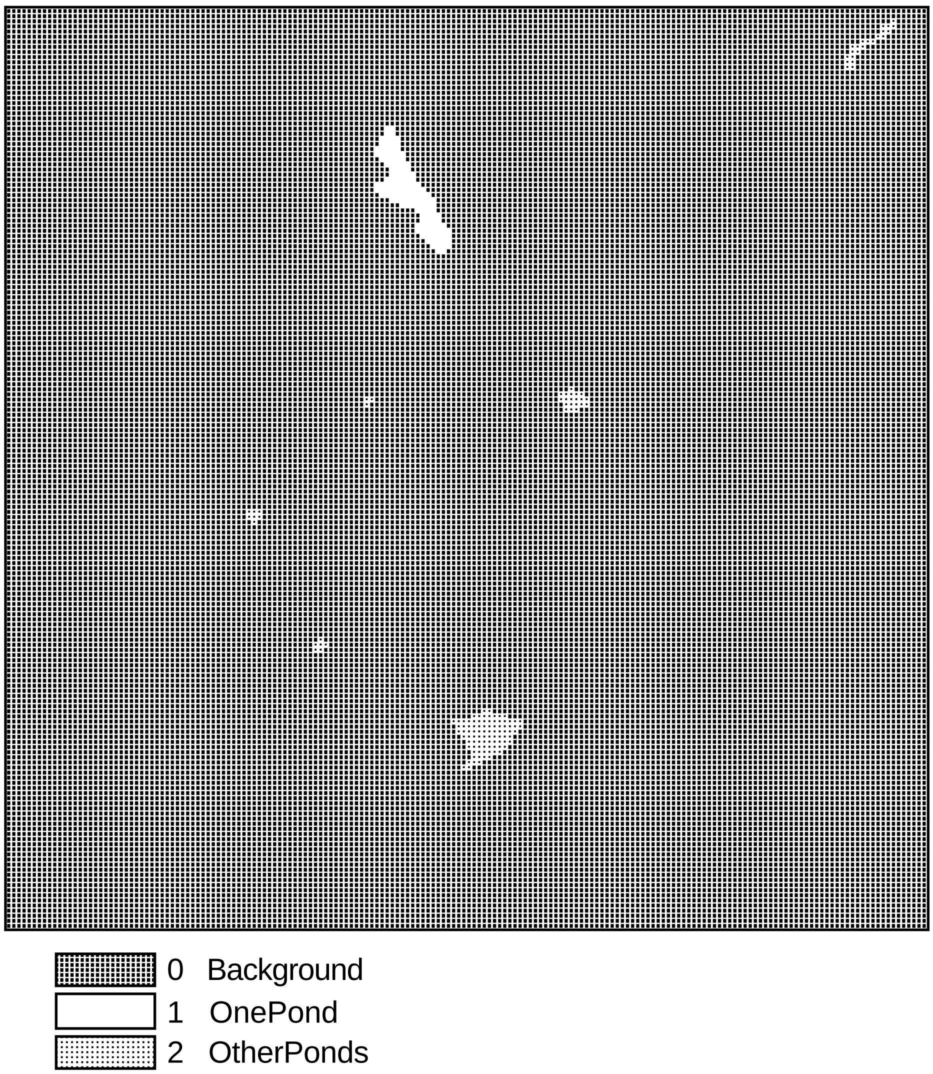


- Repeatedly average the altitude over a radius of 1000

`FocalProximity` EXAMPLE

Figure I-1 from [Toml90]

WhichPond Map

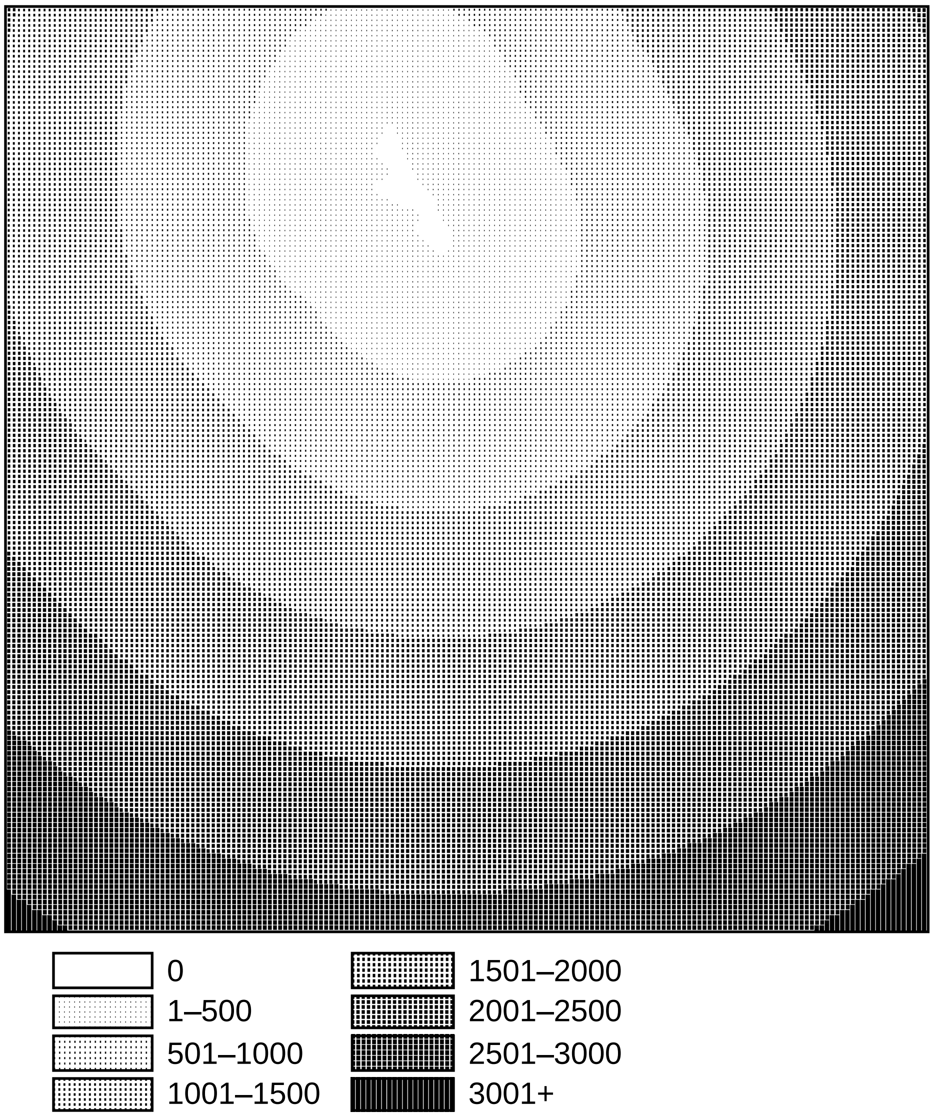


- Distinguishes one of the ponds from the rest of the ponds and the non-pond areas

Ex: Figure I-2 from [Toml90]

```
ThisPond = LocalRating of WhichPond
           with –0 for 0 2 with 0 for 1
ThisFar  = FocalProximity of ThisPond at …
```

ThisFar Map

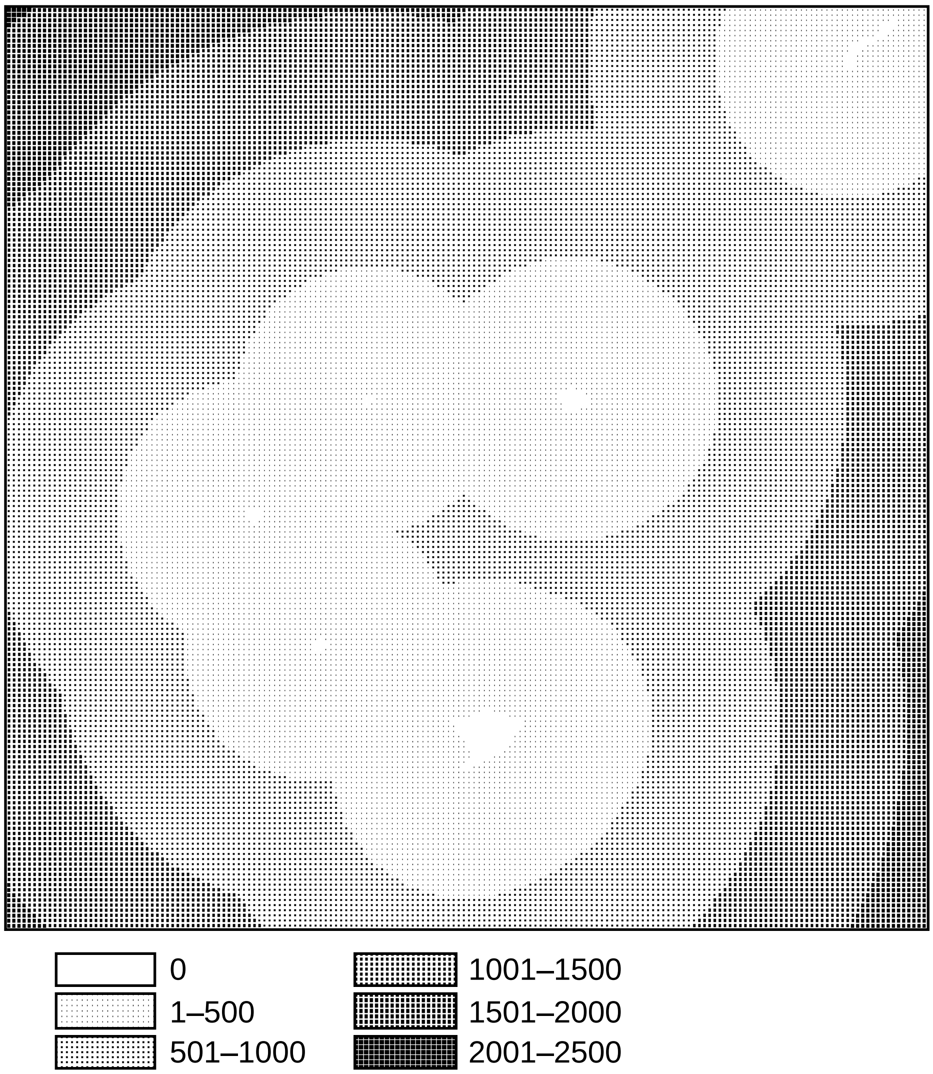


- ThisFar should be followed by a LocalRating operation to get the ranges and ThisFar Map
- Indicates ranges of distance to a distinguished pond

Ex: Figure I-3 from [Toml90]

```
ThosePonds = LocalRating of WhichPond
             with –0 for 0 1 with 0 for 2
ThatFar    = FocalProximity of ThosePonds at …
```

ThatFar Map



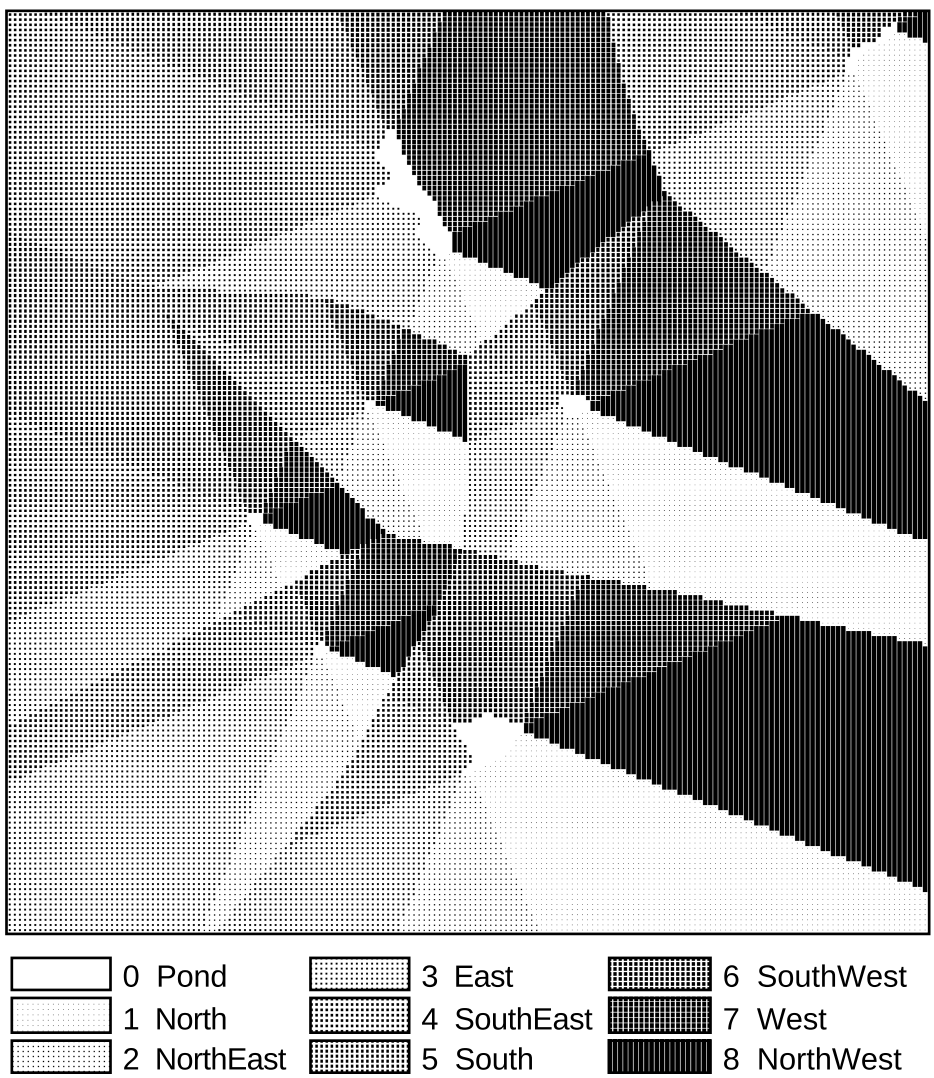
- ThatFar should be followed by a LocalRating operation to get the ranges and ThatFar Map
- Indicates ranges of distance to a group of 6 ponds

`FocalBearing` EXAMPLE

Figure 5-19 from [Toml90]

```
PondBearing =     LocalRating of WhichPond with –0 for 0
PondBearing =     FocalBearing of PondBearing
PondBearing =     LocalRating of PondBearing
            with 1 for 338…360 for 1…22 with 2 for
23…67
            with 3 for 68…110 with 4 for 111…157
            with 5 for 158…202 with 6 for 203…247
            with 7 for 248…290 with 8 for 291…337
```

PondBearing Map



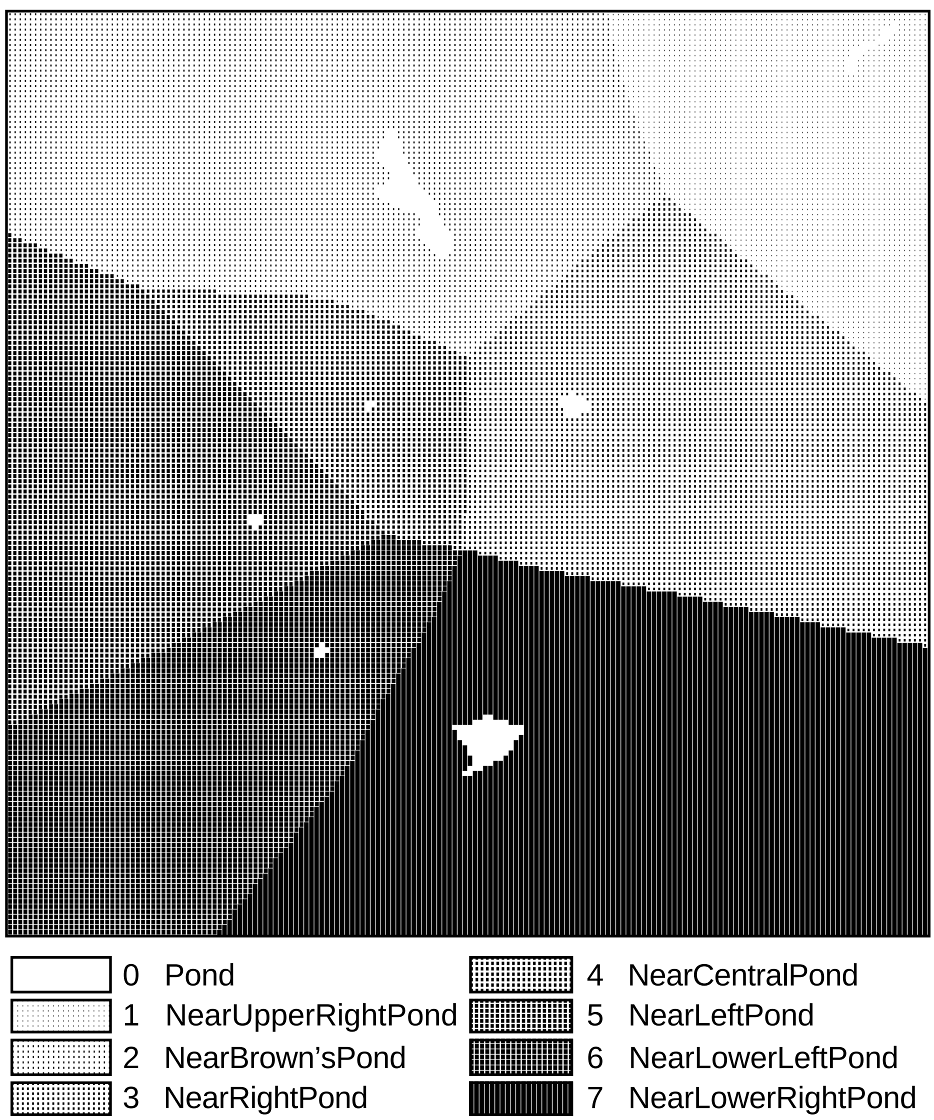
- Indicates direction to nearest pond from each location

`FocalNeighbor` EXAMPLE

Figure 5-20 from [Toml90]

```
EveryPond   = LocalRating of WhichPond with –0
              for 0
EachPond    = FocalInsularity of EveryPond
NearestPond = FocalNeighbor of EachPond
NearestPond = LocalRating of WhichPond with 0
              for 1 2 with NearestPond for 0
```

NearestPond Map



- Indicates nearest pond to each location (i.e., Voronoi diagram)

NEIGHBORHOODS BASED ON VISUAL CONTACT (RADIATING)

- Visual contact occurs between *a* and *b* if an unobstructed line of sight exists between the vertical positions of *a* and *b*
- Distance from neighborhood focus is measured over unobstructed lines of sight
- Obstruction of light forms the basis of the distance calculation
- Locations can either transmit or receive
- Assume neighborhood focus contains a light source (can have more than one focus)
- Applications

  1. surveillance

  2. siting of unsightly facilities

- Need additional information (in form of map layers and similar to attribute data)

    1. vertical position of each grid square (*surface* map layer)
    2. relative vertical position value for a neighborhood focus—i.e., the light sources (*transmission* map layer)
        - this layer enables controlling what can be seen
        - therefore, light sources are placed in these spots
    3. relative vertical position values for the obstacles (*obstruction* map layer)
        - e.g., buildings, vegetation, trees, etc.
        - line of sight cannot pass through these locations to further locations at lower vertical positions
    4. relative vertical position values for establishing the maximum vertical positions at which light (i.e., a line of sight) can be received (*reception* map layer)
        - reception region indicates the maximum 3D volume that could be illuminated by the light source
            a. cone-shaped for a covered light source
            b. dome for a light source with no cover
            c. extent is a function of intensity of light source
        - reception layer records the relative elevation value of the highest location visible at each grid cell and it could have a value even for the grid cell containing the corresponding light source
        - ideally, 1 layer for each light source which are merged

- Ex: Where should light sources be placed to assure total illumination?

ILLUSTRATION OF RADIATING LAYERS

Figure 5-23 from [Toml90]

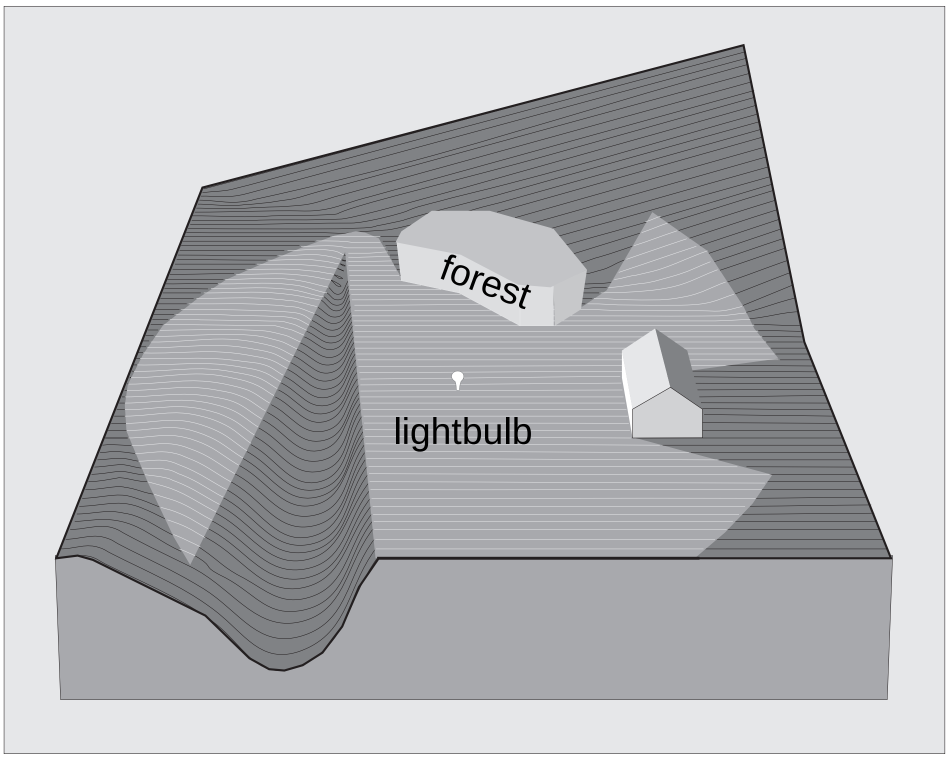


ILLUSTRATION OF RADIATING LAYERS

Figure 5-23 from [Toml90]

- Field of illumination of lightbulb is circular with a range specified by the reception map layer

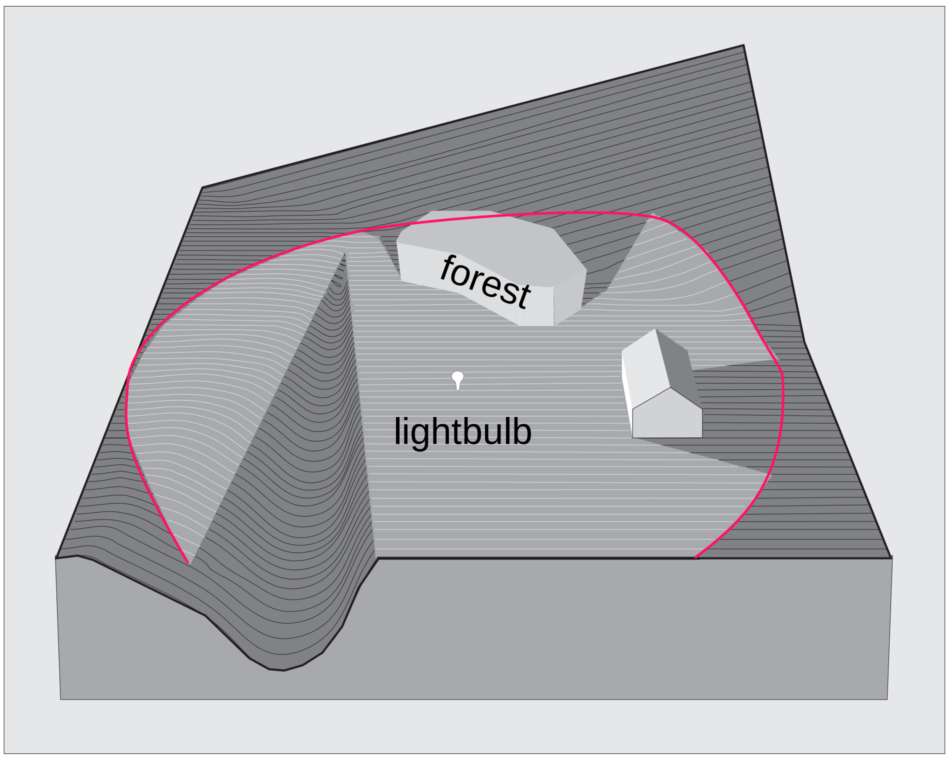
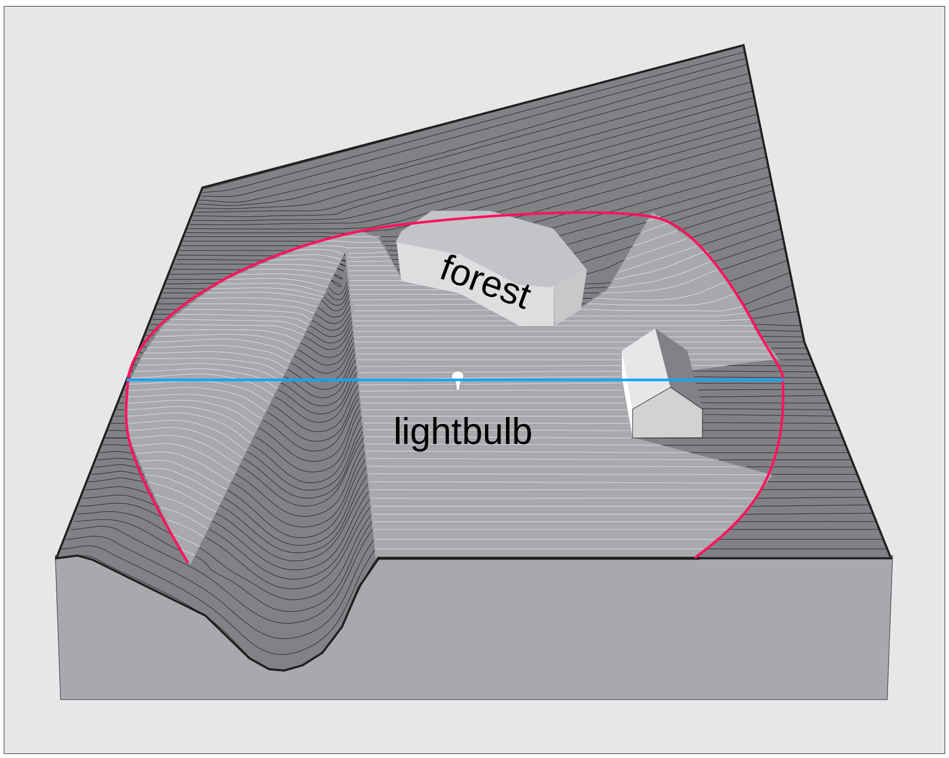
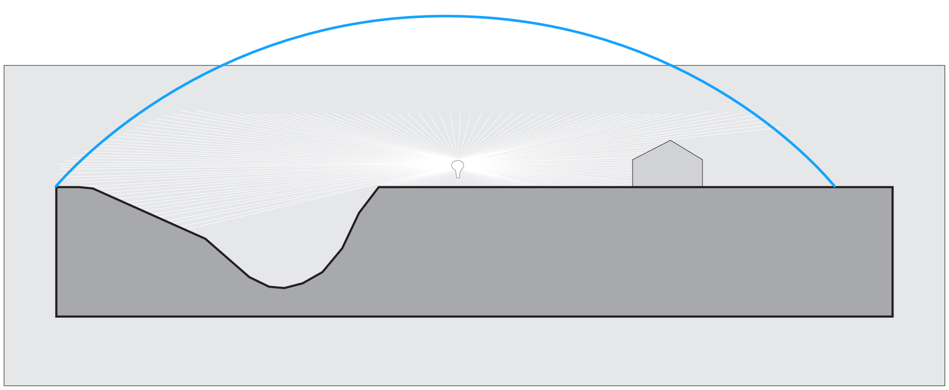


ILLUSTRATION OF RADIATING LAYERS

Figure 5-23 from [Toml90]

- Field of illumination of lightbulb is circular with a range specified by the reception map layer





- Cross-sectional view at the light source

VISUAL CONTACT EXAMPLE

- Sequence of operations

```
Steeple1 = LocalRating of Development
           with –0 for 0 1 2 3 5
Steeple2 = FocalInsularity of Steeple1
Steeple3 = LocalRating of Steeple2
           with 0 for … with 1 for 13
Steeple4 = FocalMaximum of Steeple3 at … radiating
           on Altitude
SteepleView    =   LocalRating of Development on
Steeple4
           with 1 for 1 … 5 on 1 with 2 for 0 on 1
           with 3 for 1 … 5 on 0 with 4 for 0 on 0
```

- Interpretation of operations

`Steeple1:` nullify all but public buildings

`Steeple2:` assign a unique number to each public building

`Steeple3:` nullify all but building 13 which has a steeple

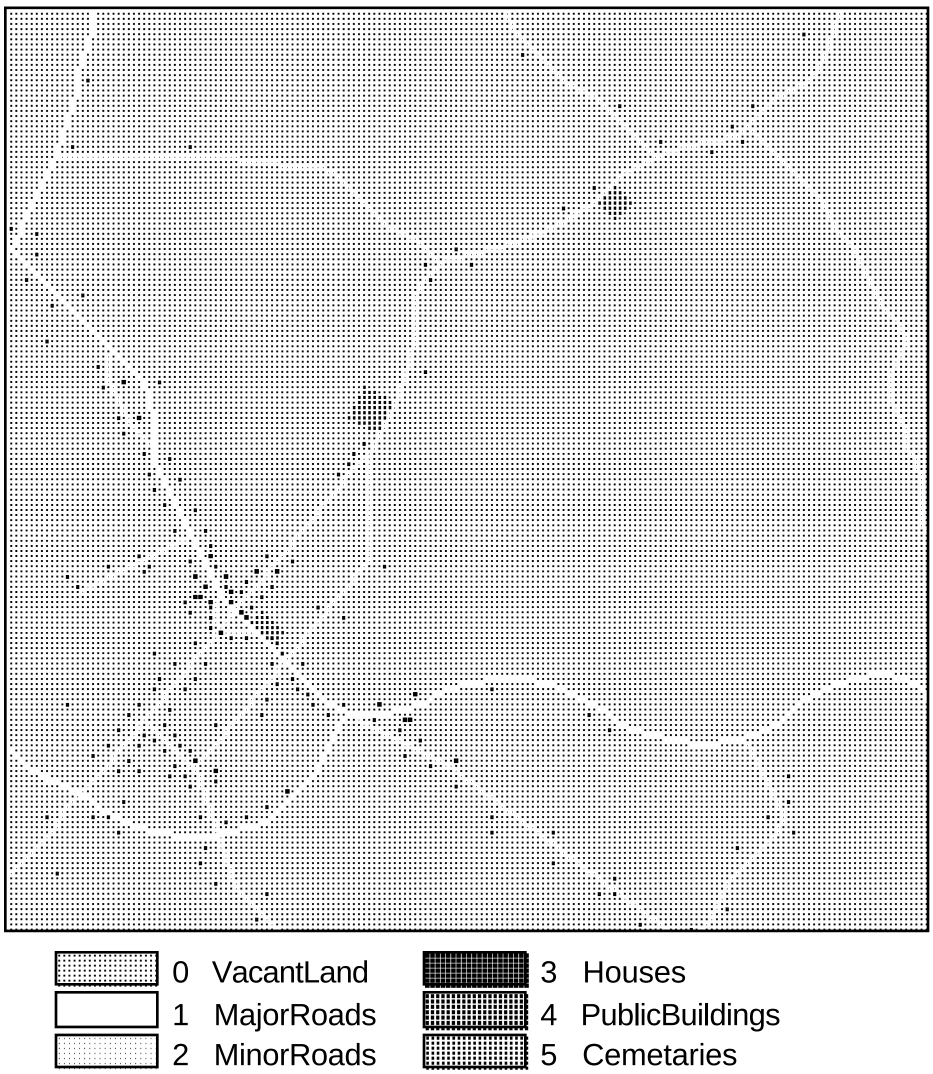
`Steeple4:` set every location to the maximum value visible from it using `Altitude` as the surface layer

1. all locations have value of 1 (steeple) or 0 (not a steeple)
2. neighborhood is unbounded since “at …”
3. `FocalMaximum` determines the maximum value that is visible from each location
   - steeple is valued 1; all other locations are 0
   - visibility is symmetric—i.e., if steeple is visible from a location, then the location is visible from the steeple

`SteepleView:` classify the land based on the combination of visibility and development (vacant or not)

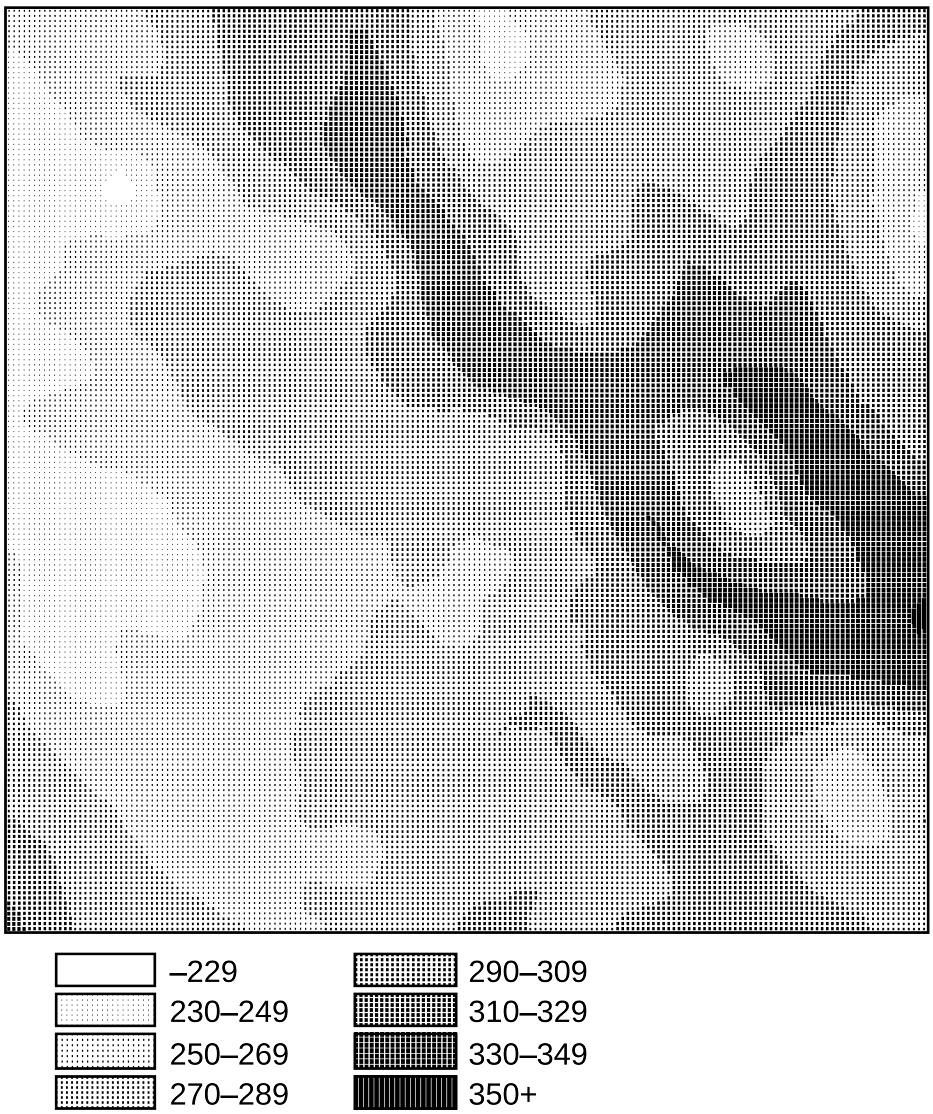
DEVELOPMENT MAP

Figure 1-12 from [Toml90]



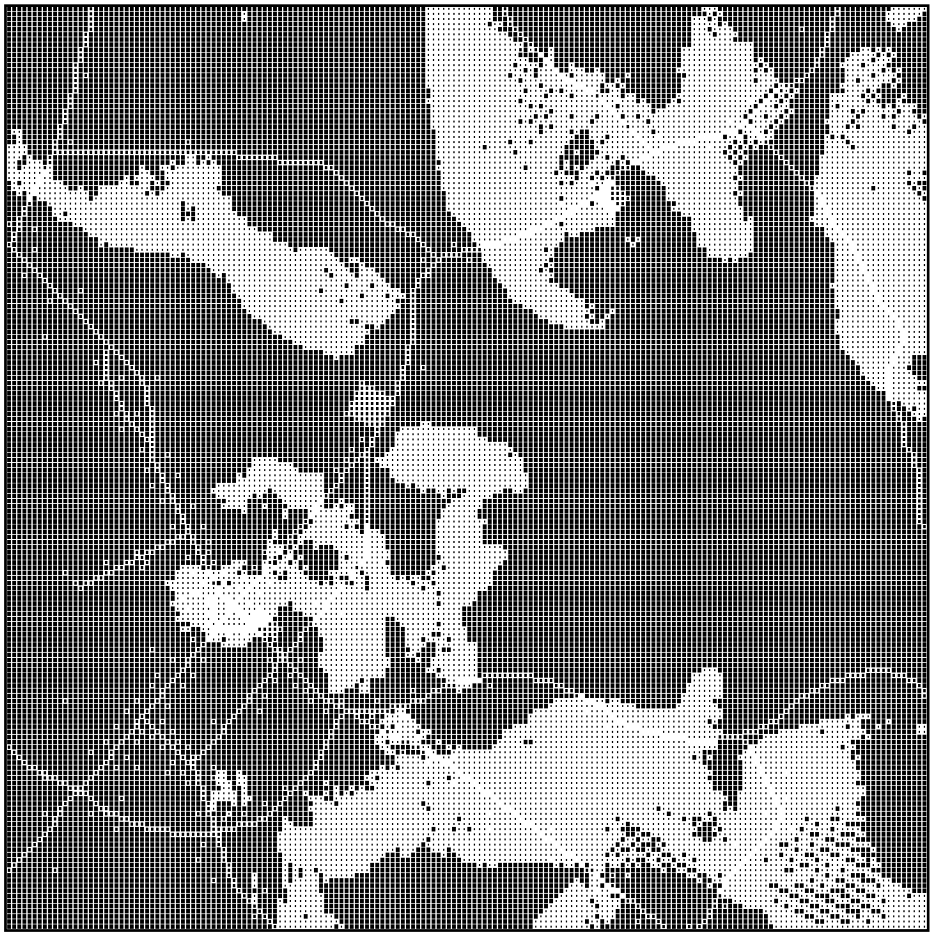
ALTITUDE MAP

Figure 1-9 from [Toml90]



STEEPLEVIEW MAP

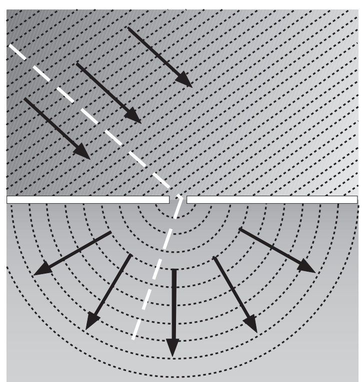
Figure 5-24 from [Toml90]



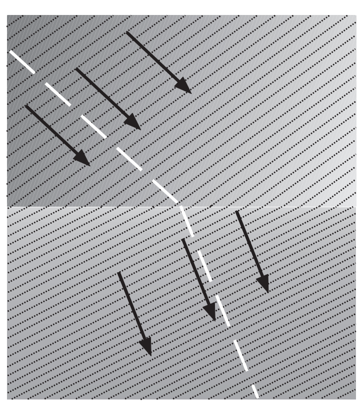
1	DevelopedLandWithinSight
2	VacantLandWithinSight
3	DevelopedLandOutOfSight
4	VacantLandOutOfSight

NEIGHBORHOODS BASED ON ACCESSIBILITY (SPREADING)

- Minimum distance between two locations is not necessarily the length of the straight line between them (i.e., Euclidean distance)
  1. distance is measured in terms of lengths of the links traversed in going from the neighborhood focus to the location
  2. account for obstructions
     - need to move above (vertical)
     - need to move around (analogous to diffraction of light)



     - trafficability in sense that need to change speed (analogous to refraction of light)



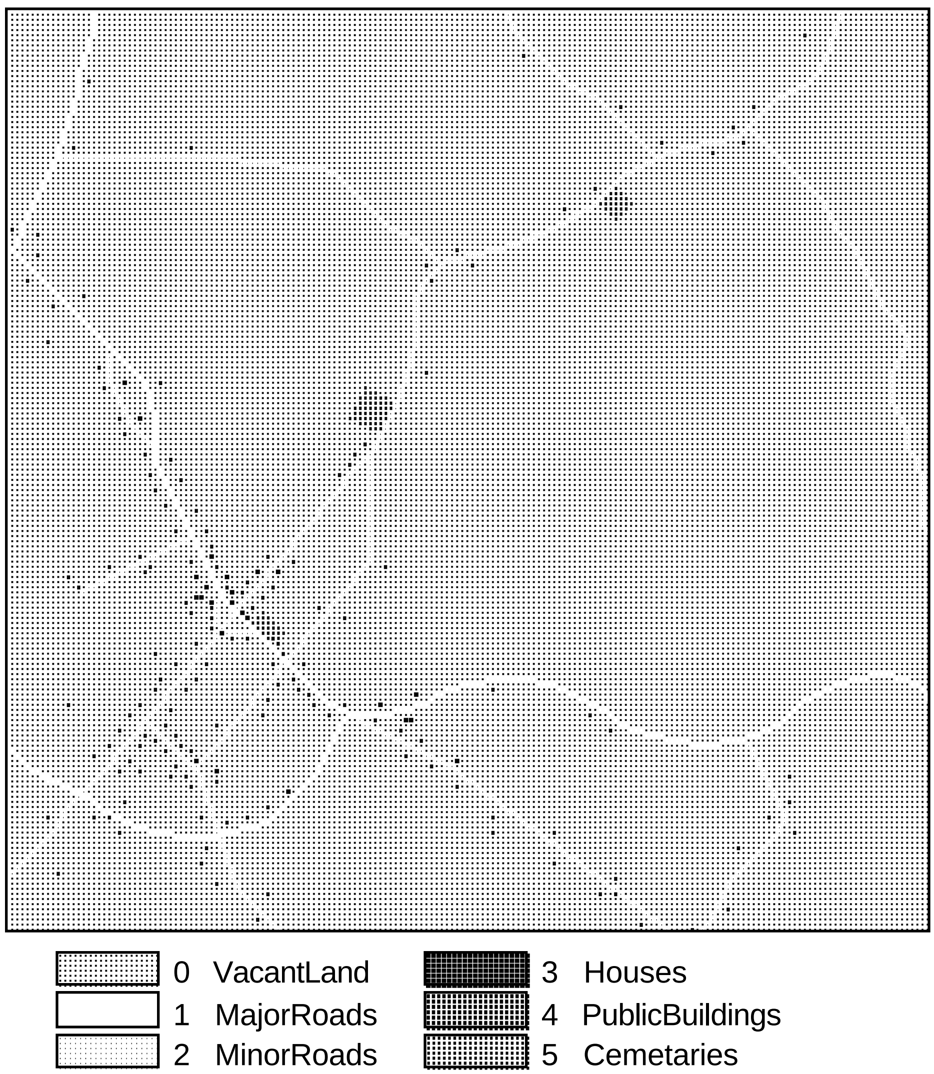
- Need additional information (in form of map layers)

  1. incremental cost of proceeding through the location (*friction* map layer)
     - like an impedance factor
     - number of incremental units of travel cost (e.g., time, money)
  2. vertical position of the location (*surface* map layer)
     - effectively increases actual distance between a location and its neighbor
  3. reachability information (*network* map layer)
     - indicate which of 8 adjacent neighbors is reachable
  4. if no additional map layers
     - distance is minimum of the sum of the lengths of the links traversed from neighborhood focus to the locations
     - use minimum since often more than one path

SPREADING EXAMPLE

Figure 1-12 from [Toml90]

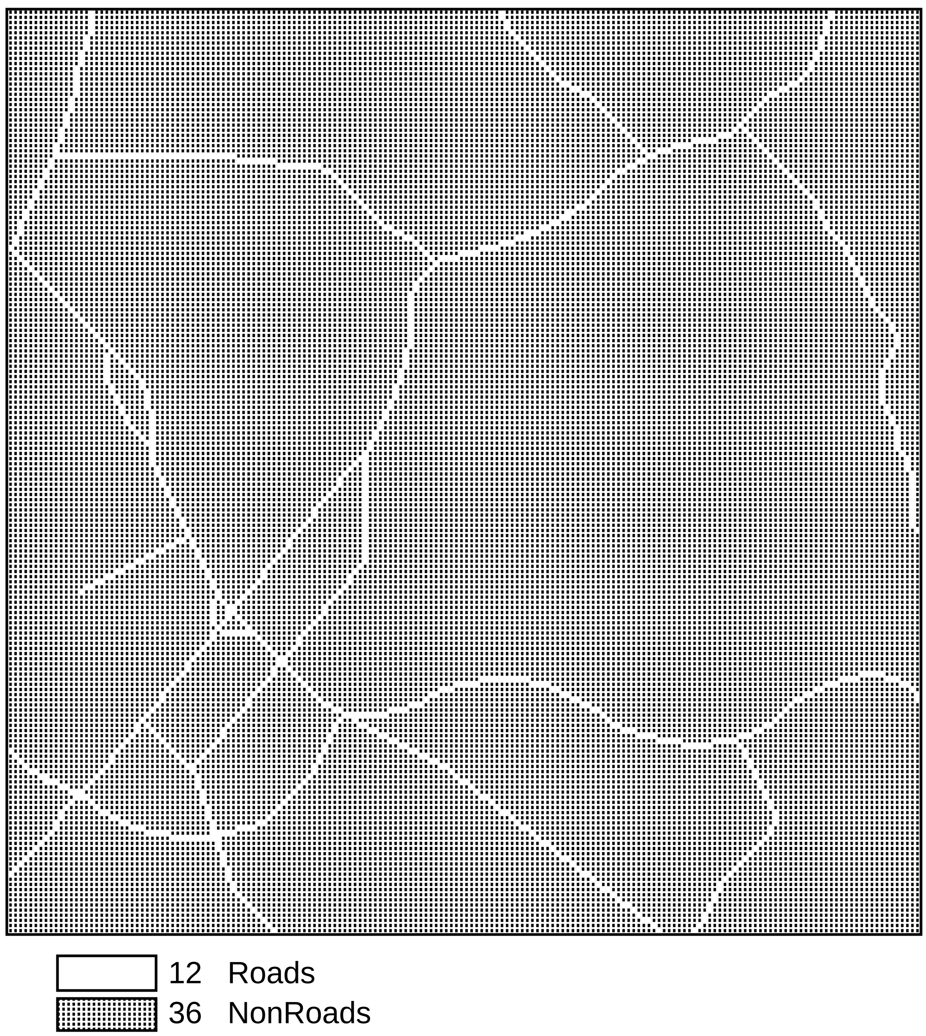
Development Map



Ex: Figure 4-2 from [Toml90]

```
Mobility = LocalRating of Development with
           12 for 1 2 with 36 for 0 3 4 5
```

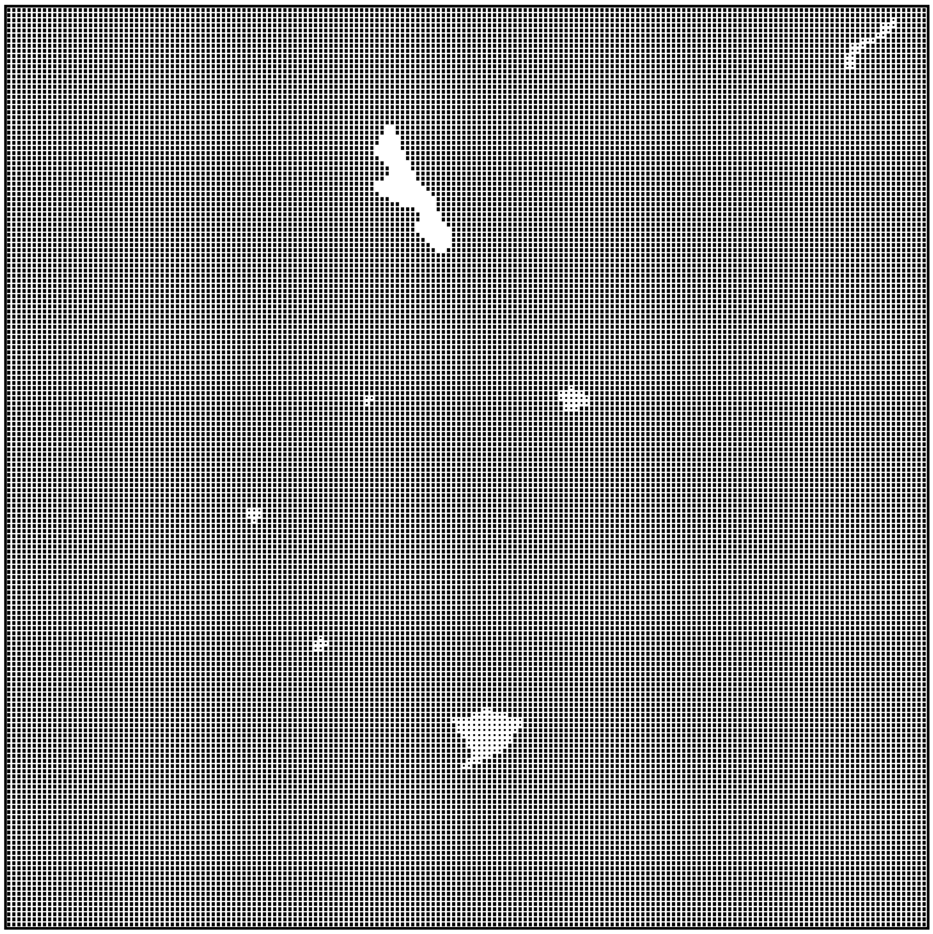
Mobility Map

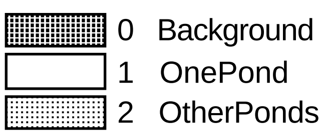


- Indicates locations lying along major or minor roads

Ex: Figure I-1 from [Toml90]

WhichPond Map



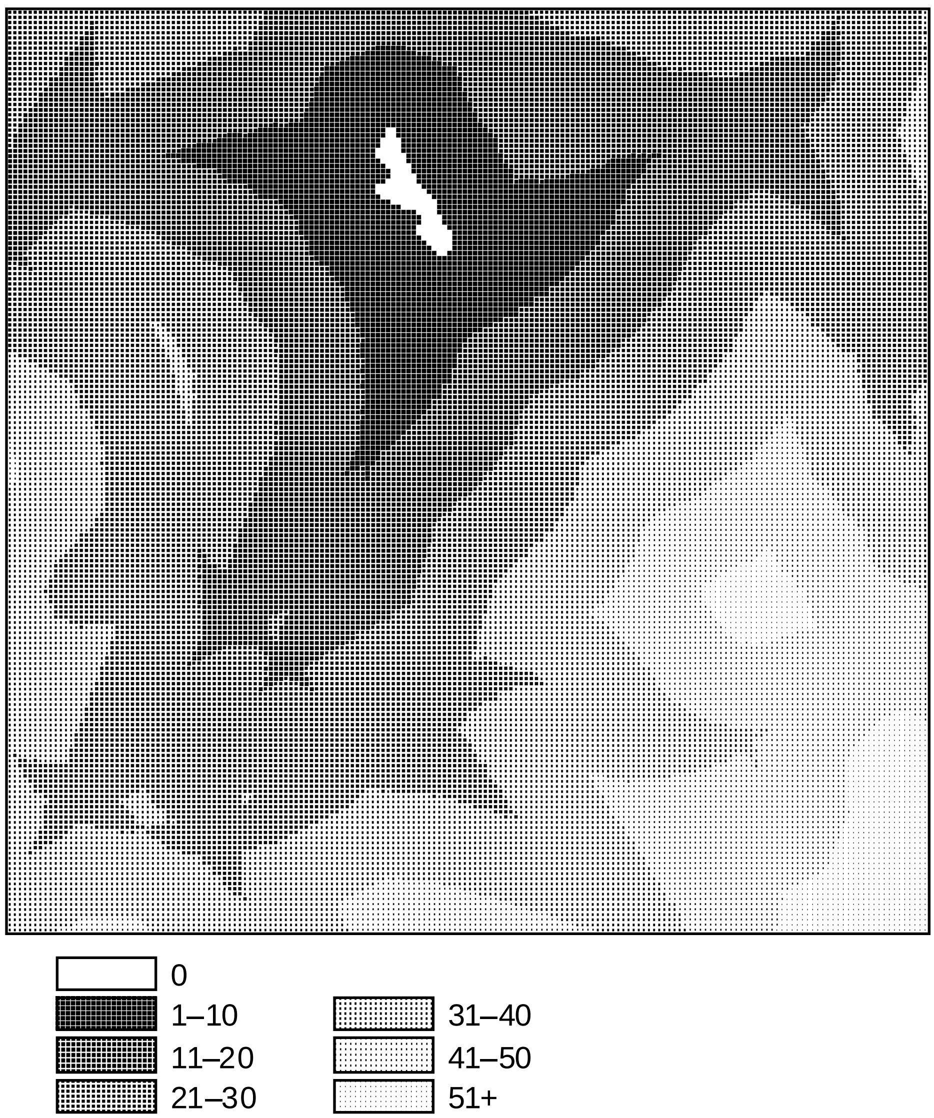


- Distinguishes one of the ponds from the rest of the ponds and the non-pond areas

Ex: Figure 5-30 from [Toml90]

```
Brown’s  = LocalRating of WhichPond
           with –0 for 0 2 with 0 for 1
ThisTime = FocalProximity of Brown’s spreading
           in Mobility at …
ThisTimeMap = LocalRating of ThisTime with … for …
```

ThisTime Map



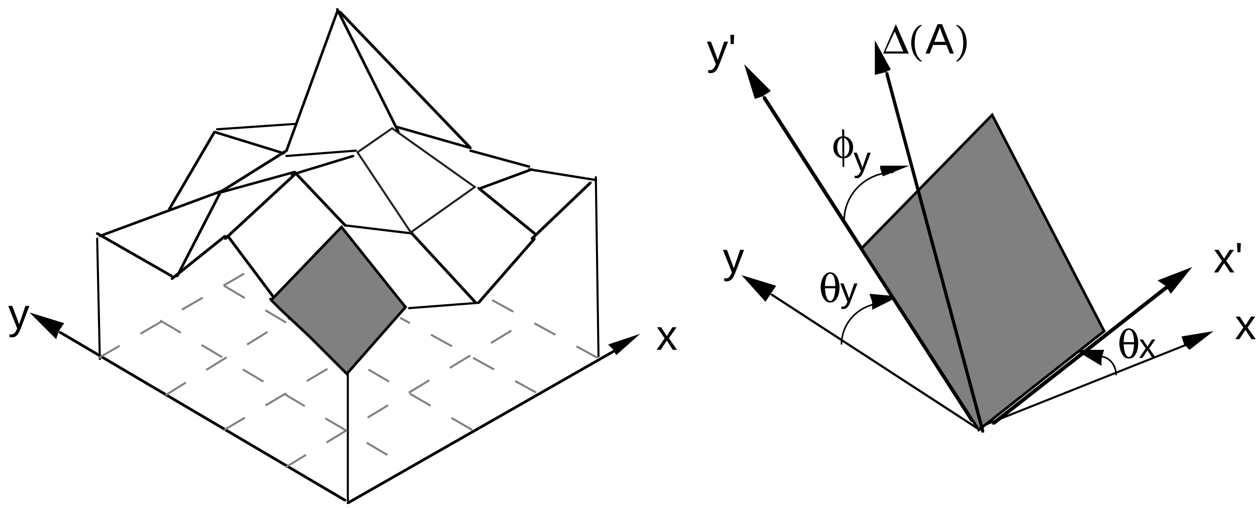
- Indicates estimated walking time from each location to a distinguished pond
- Mobility helps to indicate that walking along a road gets you to the destination faster than walking on a non-road (as the cost is lower!)

INCREMENTAL OPERATIONS

- When contents of grid squares are interpreted to contain data other than a square region
  1. infer a new map layer of appropriate type, OR
  2. compute a new map layer based on inferred values
  3. non-inference operations can also take surface information into account
- Point data have no additional operations
- Lineal data
  1. infer nature of line through location (`IncrementalLinkage`)
  2. total length of inferred lines through location (`IncrementalLength`)
- Areal data
  1. infer shape of areal element (`IncrementalPartition`)
  2. area of inferred areal element (`IncrementalArea`)
  3. length of inferred edges between the areal element and its immediate neighbors that are not in the same zone (`IncrementalFrontage`)

GRADIENT AND ASPECT

1. Assume a tilted quadrilateral is associated with each cell



2. Δ(A)= Gradient of tile A = the maximum slope of A
   - component of slope on y = $\tan \theta_y$
   - component of slope on x = $\tan \theta_x$
   - $\Delta(A) = \sqrt{\tan^2 \theta_x + \tan^2 \theta_y}$

3. Computation of aspect (direction of maximum slope Δ(A)) of tile A with respect to y

$$\phi_y = \arcsin \frac{\tan\theta_y}{\Delta(A)} = \arccos \frac{\tan\theta_x}{\Delta(A)}$$

4. Estimation of θ
   - use a 3 x 3 window centered at point (x,y)
   - d = ground separation between centers of adjacent
   - $\tan \theta_x$ = (z(x+1,y)–z(x–1,y)) / (2·d)
   - $\tan \theta_y$ = (z(x,y+1)–z(x,y–1)) / (2·d)

INCREMENTAL SURFACE OPERATIONS

- Requires a surface map layer
- Additional map layer can specify areal conditions under the surface—if absent, then assume all one zone
- Applicable to all types of surfaces (not just topography)

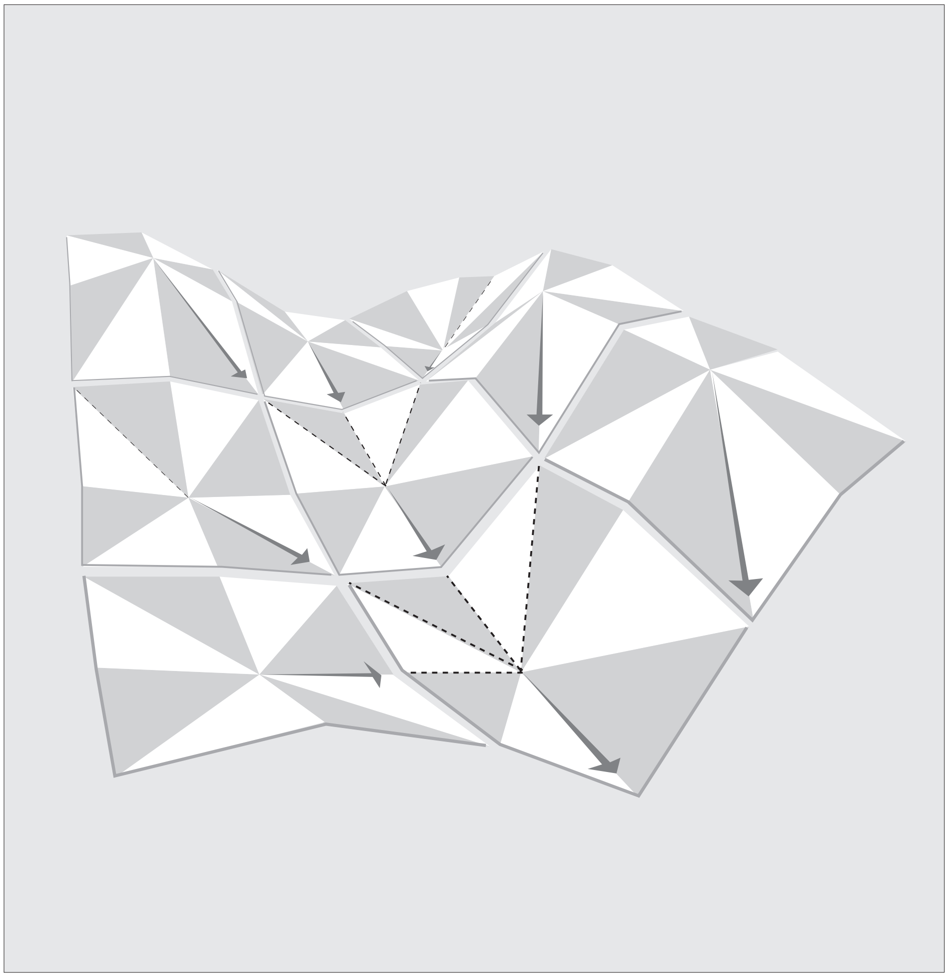
- Operations

  1. volume under the surface for each location (`IncrementalVolume`)
  2. infer a slope from the surface layer value and all immediate neighbors with same zonal value (`IncrementalGradient`)
     - in degrees with 0 being horizontal and 90 being vertical
  3. compass direction of steepest descent for plane inferred by `IncrementalGradient` (`IncrementalAspect`)
     - range from 0 (denoting nonsloping value) to 360 degrees
  4. identity of immediate upstream neighbors on plane inferred by `IncrementalGradient` (`IncrementalDrainage`)
     - neighbors from which the location is in a direction of steepest descent
     - result is an 8 bit vector corresponding to the 8 directions
     - also known as a *watershed*

- `IncrementalGradient` and `IncrementalAspect` can be used to measure rate of change or direction of change in any quantity

ILLUSTRATION OF SURFICIAL DRAINAGE

Figure 5-11 from [Toml90]



- Arrows indicate the direction(s) of steepest or downstream descent from each location on a surface toward one or more of its immediate neighbors
- Broken lines indicate the upstream directions associated with each of these neighbors

CARTOGRAPHIC MODELING

1. Descriptive
   - *"what is?"* or *"what could be?"*
   - analytic methods
     a. decompose data into finer levels of meaning
     b. objective
     c. ferret *"what is significant?"*
   - synthetic methods
     a. recompose data to discover new meaning
     b. expose significant facts in the data
     c. express a meaning attached to these facts
     d. subjective
     e. answer *"how something is significant?"*
        - what makes it significant
        - value judgement of its significance
2. Prescriptive
   - *"what should be?"*
   - cartographic allocation problems
     a. atomistic—function of individual locations
     b. holistic—function of neighborhood with similar properties

ANALYTIC METHODS

1. Position of a cartographic condition
   - measurements indicating its absolute position (e.g., centroid)
   - measurements indicating its relative position
     a. in terms of distance or direction
     b. similar to integral calculus
        - Ex: `‘FocalProximity of X spreading in FRICTIONLAYER’` is similar to integration as the result is like the area under a curve
        - `IncrementalGradient` yields slope which is like differentiation
        - applying `IncrementalGradient` to result of `FocalProximity` should return something similar to `FRICTIONLAYER`

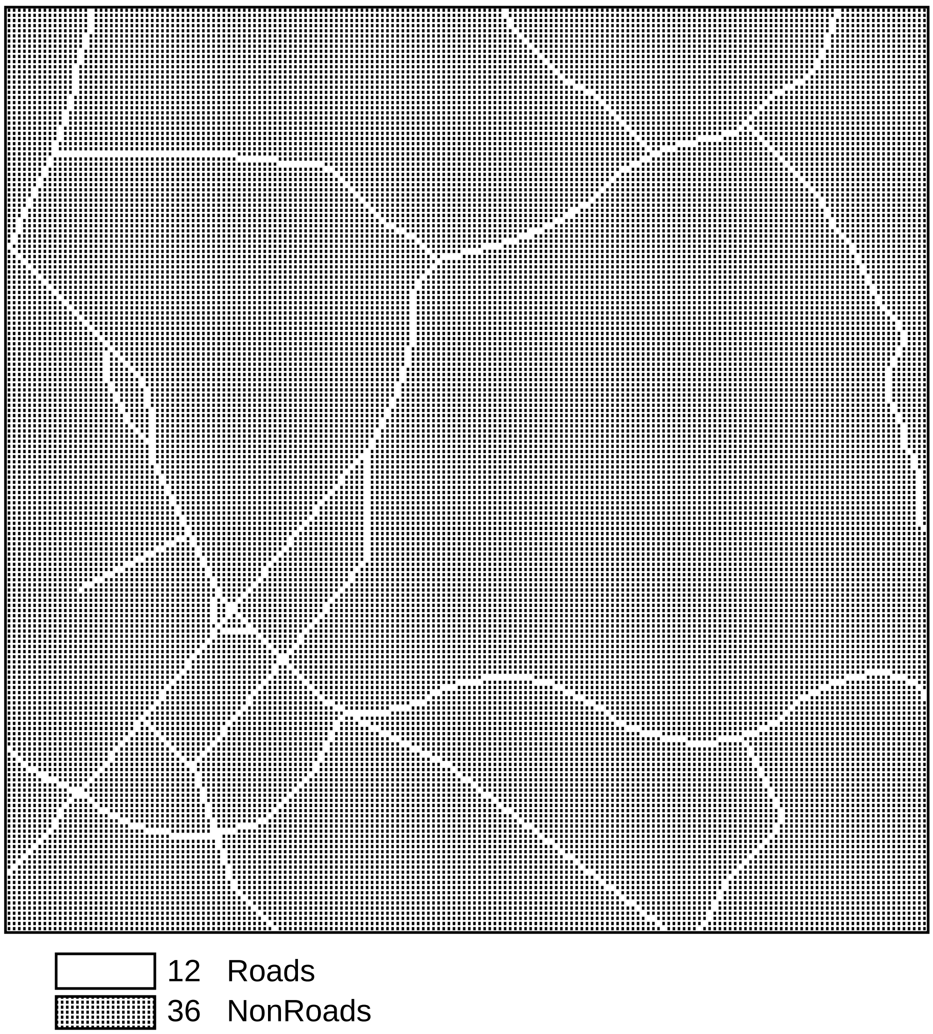
          (i) ‘similarity’ rather than exact match because of the two-dimensional nature of the `X` layer

          (ii) if the `X` layer was one-dimensional, then the result would be identical to `FRICTIONLAYER`

Ex: Figure 4-2 from [Toml90]

```
Mobility = LocalRating of Development with
           12 for 1 2 with 36 for 0 3 4 5
```

Mobility Map

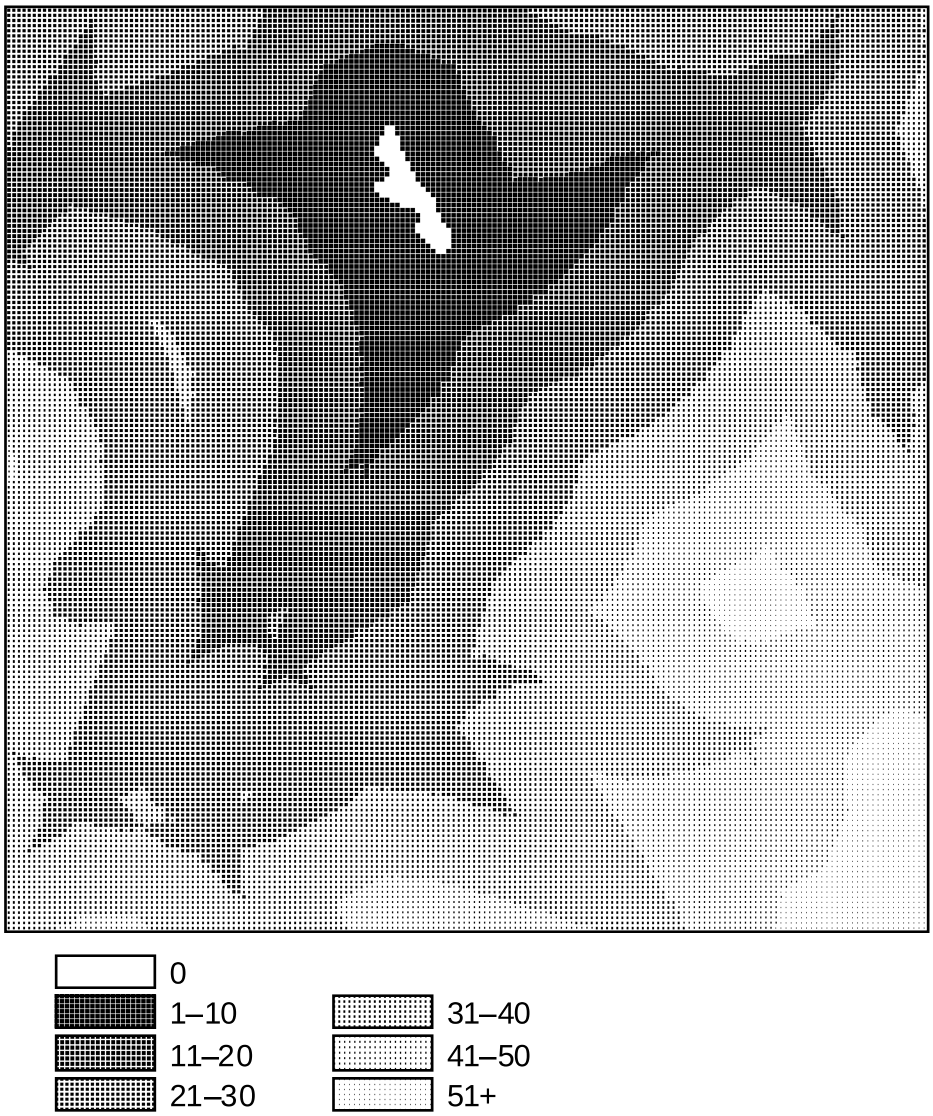


- Indicates locations lying along major or minor roads

Ex: Figure 5-30 from [Toml90]

```
Brown's  = LocalRating of WhichPond
            with –0 for 0 2 with 0 for 1
ThisTime = FocalProximity of Brown's spreading
           in Mobility at …
ThisTimeMap = LocalRating of ThisTime with … for …
```

ThisTime Map

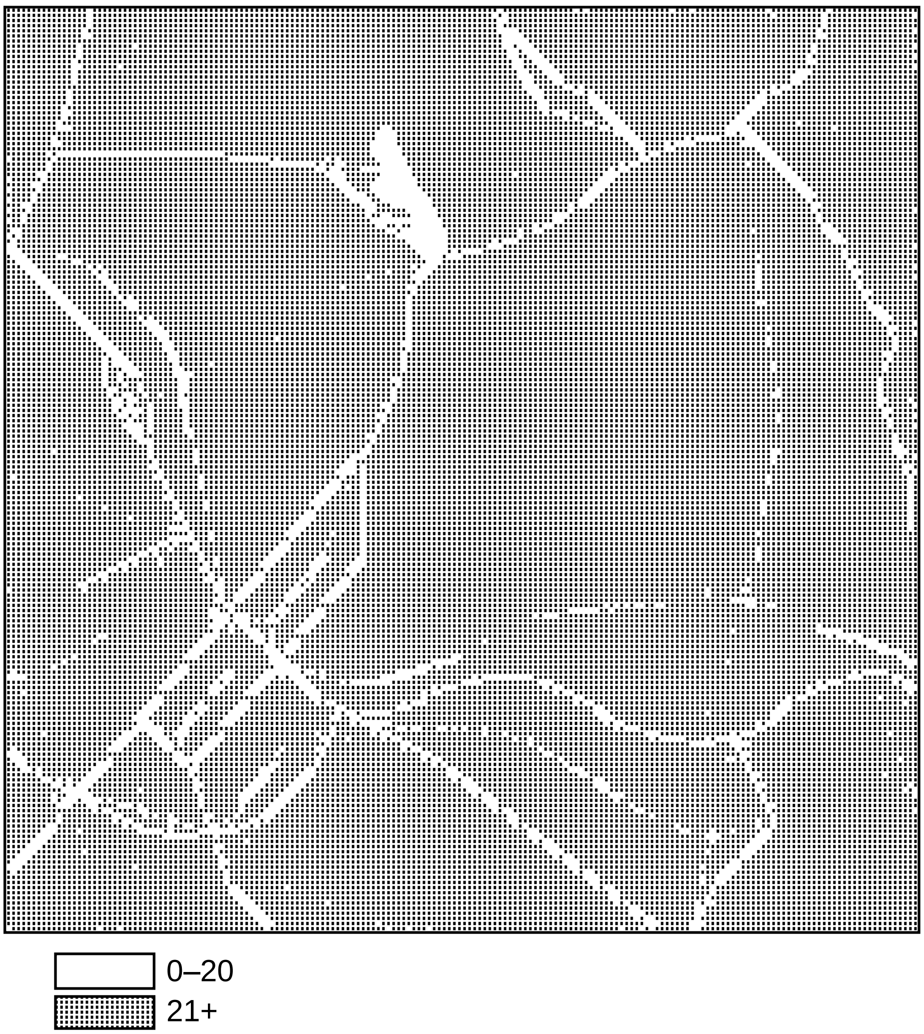


- Indicates estimated walking time from each location to a distinguished pond
- Mobility helps to indicate that walking along a road gets you to the destination faster than walking on a non-road (as the cost is lower!)

Ex: Figure 7-1 from [Toml90]

```
InferredMobility = IncrementalGradient of
                   ThisTime
```

InferredMobility Map



- Indicates change in cumulative travel time to a distinguished pond (inverts the effect of `FocalProximity Spreading` in operation)
- Areas in white in addition to the roads represent paths along which other ponds are equally accessible

2. Cartographic form

- in terms of size and shape
- depends on nature of geographic feature being modeled

  a. point data has no shape but shapes of point clusters may be meaningful

  b. lineal data
     - length

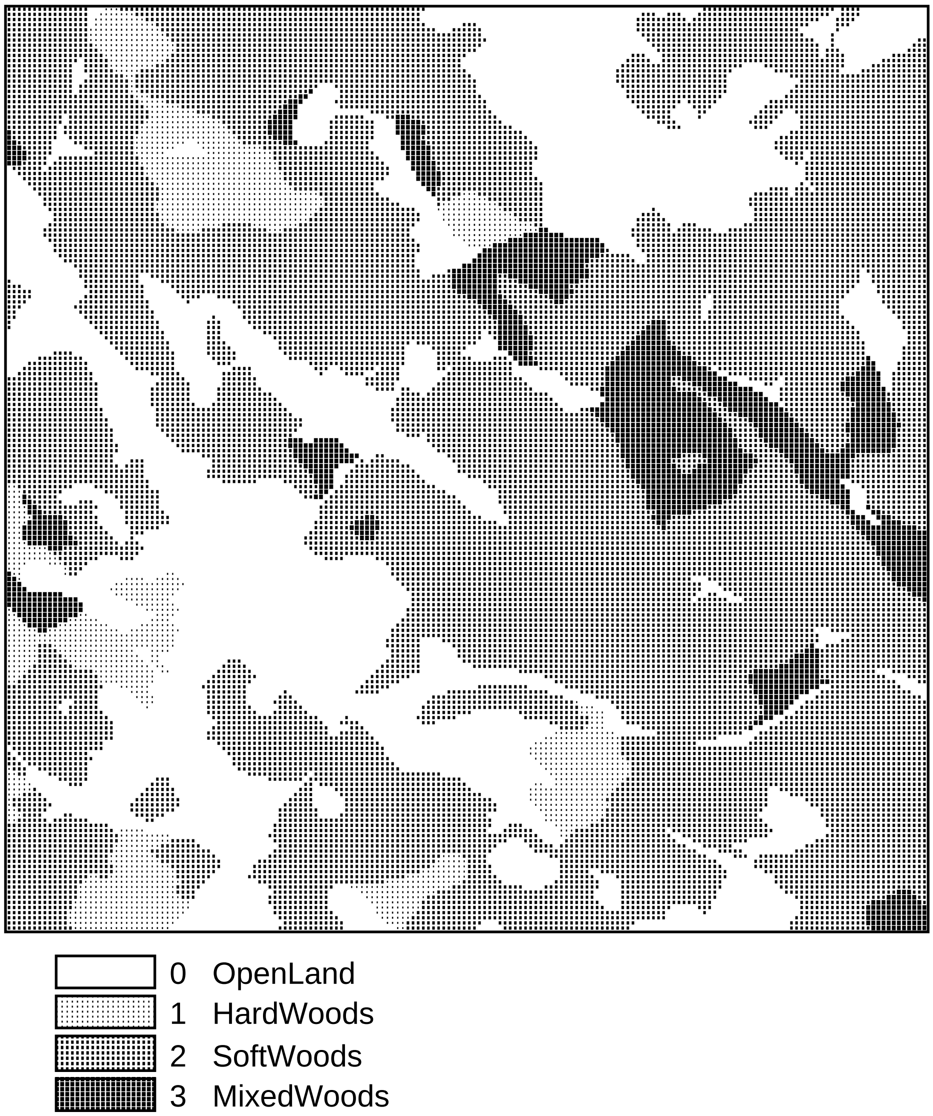
  c. areal data
     - roundness
     - genus
     - number of holes
     - etc.

  d. surface data
     - topographic inflection (slope of slope)
     - shadiness
     - narrowness
     - etc.

CARTOGRAPHIC FORM EXAMPLE

Figure 1-11 from [Toml90]

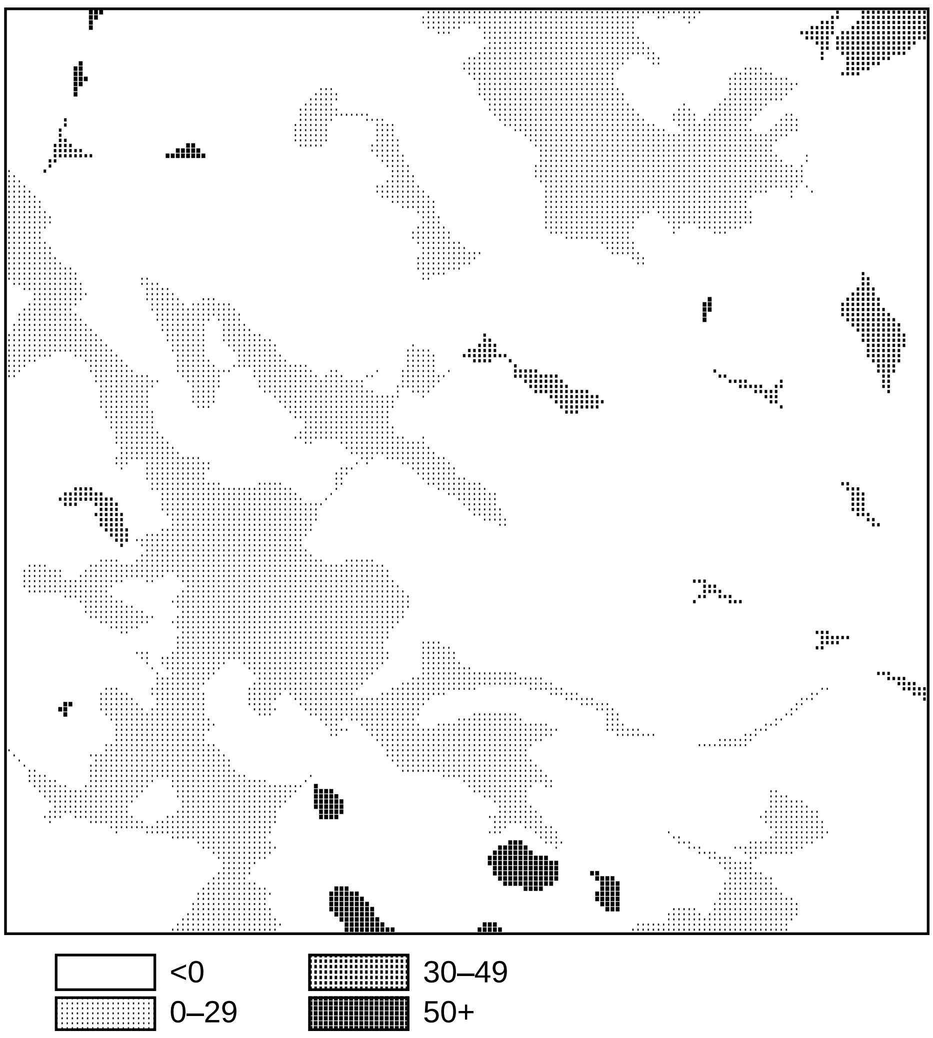
Vegetation Map



Ex: Figure 7-6 from [Toml90]

```
EveryField = LocalRating of Vegetation with –0 for 1 2 3
EachField  = FocalInsularity of EveryField
Area       = IncrementalArea of EveryField
Area       = ZonalSum of Area within EachField
Area       = LocalRoot of Area and 2
Area       = LocalProduct of Area and 354
Perimeter  = IncrementalFrontage of EachField
Perimeter  = ZonalSum of Perimeter within EachField
HowRound   = LocalRatio of Area and Perimeter
```

HowRound Map



- Indicates the degree to which each location's areal form is round $\equiv 354 \cdot \sqrt{zone\_area / zone\_perimeter}$

SYNTHETIC METHODS

1. Formulation phase
   - subjective and difficult to pinpoint
2. Implementation phase
   - less subjective but still subjective when compared to analytic
   - how to express subjective judgement
     a. `LocalRating`: characterize or filter a set of implications
     b. `LocalCombination`: synthesize all the values in a way that avoids (or defers) making an explicit judgement because of its combinatorial nature
     c. use a common statistic
        - `LocalMajority`, `LocalMinority`, `LocalVariety` can be used with all types of measurements
        - can combine above with `LocalMaximum` and `LocalMinimum` for all but nominal measurements
        - `LocalSum`, `LocalDifference`, and `LocalMean` can be used with range and ratio measurements
        - `LocalProduct`, `LocalRatio`, etc. can only be used with ratio measurements

PRESCRIPTIVE METHODS

1. Cartographic allocation problems
   - selection of locations to satisfy some criteria
   - result in a solution of a problem
   - investigating *“how things will be”* in contrast with *“how things are”* as in descriptive methods
2. Types of allocation problems
   - atomistic—in terms of individual locations
   - holistic—in terms of an entire geographic space

ATOMISTIC ALLOCATION

1. Function of individual locations
2. Precise criteria
   - e.g., air quality must be above a certain value in given regions
3. Solution involves local operations
4. Usually many solutions
   - implies problem is underconstrained
   - constrain the criteria and iterate

HOLISTIC ALLOCATION

1. Function of a neighborhood whose elements have similar properties

2. Criteria

   • varied (e.g., minimum cost paths, connectedness, size)

   • imprecise

   • vague (e.g., roundness, sparseness)

   Ex: locate a site for an airport

   • examine topological criteria for positioning of runways

   • shape and grouping pattern are as important as location

3. Solution involves focal and zonal operations

4. Imprecision and vagueness influence solution process

   • use heuristics to establish a starting point or existing condition

   • iterate