

Topic 4

Spatial Analysis

4.1 Classes of Analytical Operations

As discussed in Topic 3, vector-based systems identify three basic map features—points, lines and polygons. These features are suitable for characterizing discrete spatial objects, such as light poles, streets and property boundaries. However, continuous gradients, such as an elevation surface or a proximity map are poorly represented as contour lines that generalize detailed data into a set of intervals used for display.

The introduction of a grid-analysis frame provides a structure for storage and processing of a fourth map feature—a surface. The grid-based construct enables display and processing of geographic space as a continuum. Its base spatial unit is a “grid cell” defined by the column and row coordinates of an imaginary grid superimposed over an project area that is used to identify...

Points— a single cell,

Lines— a set of linearly connected cells,

Polygons— all cells identifying the edge and interior of a spatial object, and

Surfaces— all cells within a project area with a value assigned to each that indicates the presence by feature type (discrete object) or the relative variable response (continuous gradient).

Also, the consistency of the analysis frame provides a foothold for a robust set of advanced map analysis operations, as well as a generalized analytic structure. For example, data base packages typically contain a couple of dozen analytic operations, yet these systems can be used to create models for such diverse applications such as address lists, payroll, inventory control, or commitment accounting. Once developed, the logical sequencing of data base commands is fixed into menus for easy end-user applications.

This flexible structure provides for dynamic simulation as well as data base management. For example, a typical spreadsheet package allows users to define interrelationships among important factors in a system, such as a financial model of a company's production process. By changing specific values in the model, the impact of several fiscal scenarios can be simulated.

Data base management and spreadsheet packages have revolutionized the handling of non-spatial data. The potential of computer-assisted map analysis promises a similar revolution for spatial data processing—sort of a spatial spreadsheet based on map variables. From this perspective, four classes of spatial analysis operations can be identified...

Reclassifying map categories by simply reassigning map values without creating any new spatial information,

Overlaying two or more maps that results in delineation of new spatial information based on spatial coincidence,

Measuring distance and connectivity to generate entirely new information by characterizing the relative positioning of map features, and

Characterizing neighbors to summarize the conditions occurring in the general vicinity of map locations.

The reclassifying and overlaying operations based on point processing are the backbone of current GIS applications, allowing rapid updating and examination of mapped data. However, other than the significant advantages of speed and ability to handle tremendous volumes of data, these capabilities are similar to those of traditional processing approaches. However, map-wide overlays, distance and neighborhood operations identify more advanced analytic capabilities that are not part of the paper map nor traditional desktop mapping legacies.

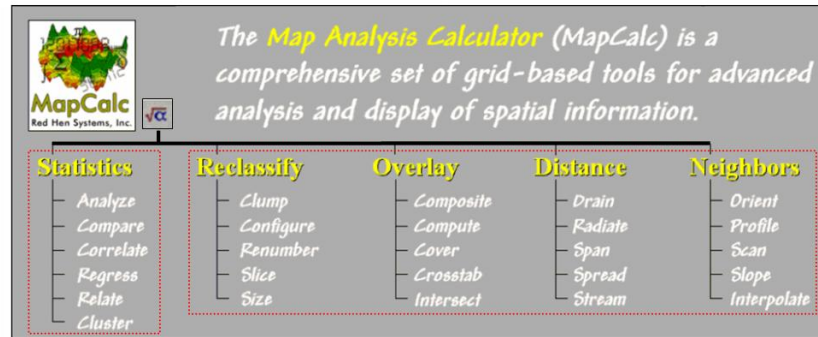


Figure 4-1. MapCalc operations for spatial analysis and statistics.

The grid-based map analysis capabilities in the MapCalc system used in this book are divided into two broad classes—spatial analysis and spatial statistics (see figure 4-1). **Spatial Analysis** tools derive new maps based on spatial context, patterns, surface configuration, proximity, connectivity and flows. They include:

✓Geographic Search ✓Weighted Proximity Analysis ✓Slope and Aspect ✓Visual Exposure ✓Optimal Path
 ✓Local, Regional and Map-Wide Overlay ✓Nearby Neighbor Summary ✓Edge, Shape and Pattern
 Characterization ✓Narrowness ✓Logical, Ranking, Rating and Mathematical Combination ✓Confluence
 Surfaces ✓Propensity Analysis ✓Travel-Time Surfaces ✓Competition Analysis.

Spatial Statistics tools investigate statistical patterns and relationships among maps. They include:

✓Descriptive Statistics ✓Normality Tests ✓Point Density Maps ✓Geographic Trend Analysis ✓Spatial
 Interpolation ✓Spatial Autocorrelation ✓Scatter Plots and Variograms ✓Residual Analysis ✓Error Mapping
 ✓Coincidence Statistics ✓Change Surfaces ✓Comparison Tests ✓Spatial Correlation and Dependency
 ✓Multivariate Regression ✓Clustering ✓Map Similarity.

The remaining topics in this book investigate several geo-business applications utilizing spatial analysis and spatial statistics tools. It is important to note that most modern GIS systems support many, if not all, of the fundamental operations used in the examples. The goal of this book isn't to train you in MapCalc software procedures (vocational) but to introduce spatial reasoning skills through hands-on experience in applying grid-based map analysis concepts (educational).

4.2 Creating Travel-time Maps

One of the most intriguing classes of spatial analysis operations involves measuring distance and connectivity. The ability to identify and summarize areas around a map feature (a.k.a. *buffering*) is a fundamental analysis tool in most desktop mapping systems. A user selects one or more features then chooses the buffering tool and specifies a radius distance. Figure 4-2 shows a **simple buffer** of 1-mile about a store (Kent's Emporium) that can be used to locate its "closest customers" from a geo-registered table of street addresses.

This information, however, can be misleading as it treats all of the customers within the buffer as the same. Common sense tells us that some street locations are closer to the store than others. A **proximity buffer** provides a great deal more information by dividing the buffered area into zones of increasing distance. It is like tossing a rock into a pond (store location) with the ripples generated indicating increase distance. While the concept is simple, the construction of a proximity buffer in a traditional mapping system involves a tedious cascade of commands.

Calculating a Simple Buffer (Desktop Mapping System)

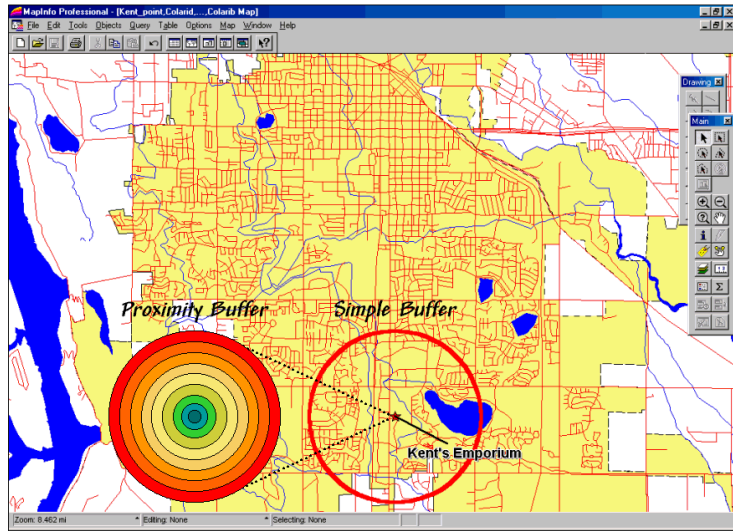


Figure 4-2. A simple buffer identifies the area within a specified distance.

Yet even a proximity buffer lacks the spatial specificity to determine effective distance considering customers rarely travel in straight lines from their homes to a store. Movement in the real world is seldom straight but our traditional set of map analysis tools assumes everything travels along a straightedge connecting two points. As most customers travel by car, a procedure is needed to generate a buffer reflecting movement along the streets.

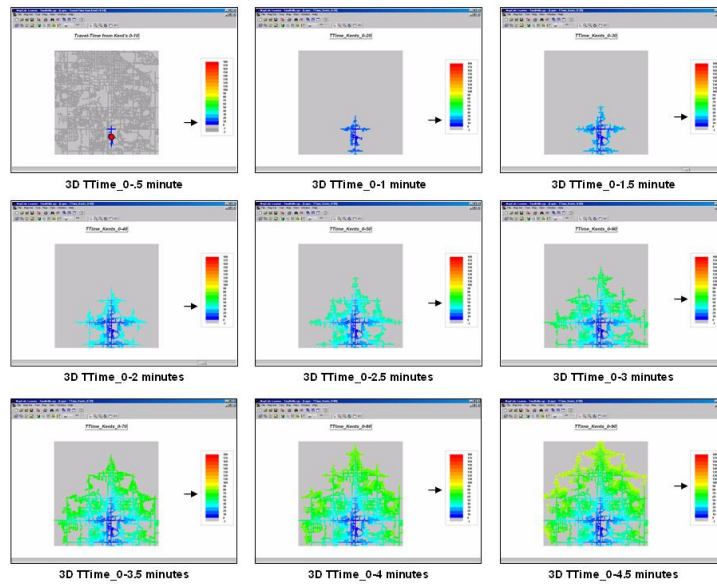


Figure 4-3. Travel-time is computed as a series of increasing waves moving away from a starting location that are constrained by the street network.

This process is best conceptualized by imagining the street network is an interconnecting series of canals. When a rock is tossed into the canal, a series of waves propagates outward but are constrained to the pattern of the streets. Figure 4-3 shows the wave-front's movement in half-minute increments. The procedure simulates driving a car along all possible pathways away from the store, keeping track of the least time to any location along the street network (optimal path).

Figure 4-4 outlines a grid-based procedure for calculating an *effective buffer* based on travel-time from Kent’s Emporium. The first step is to export the store and street locations from a desktop mapping system to a grid-based one. Inset (a) depicts the superimposition of a 100-column by 100-row analysis grid over the area of interest. In effect, the data exchange “burns” the store location into its corresponding grid cell (inset b). Similarly, cells containing primary and residential streets are identified (inset c) in a manner analogous to a branding iron burning the street pattern into another grid layer.

Calculating an Effective Buffer (Grid Analysis System)

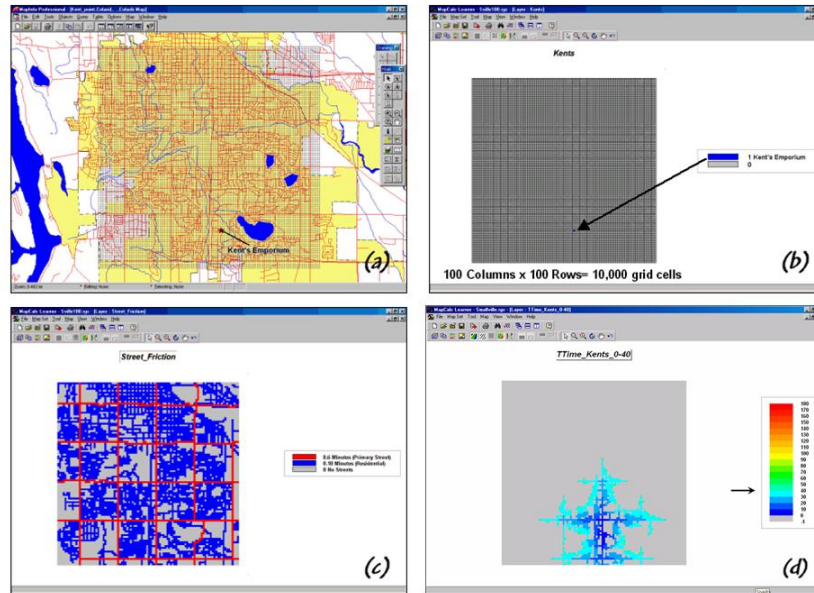


Figure 4-4. An effective buffer characterizes travel-time about a map feature.

Inset (d) shows an effective travel-time buffer derived from the two grid layers. The Store map identifies the starting location and the Streets map identifies the relative ease of travel. Primary streets are the easiest (.1 minute per cell), secondary streets are slower (.3 minute) and non-road areas can’t be crossed at all (infinity). The result is a buffer that looks like a spider’s web with color zones assigned indicating travel-time from the store (effective proximity buffer).

While the effective proximity buffer contains more realistic information than the simple proximity buffer, it isn’t perfect. The calibration for the friction surface (inset c) assumes an “average speed” for the street types without consideration of one-way streets, left-turns, school zones and the like. Network analysis packages are designed for such detailed routing. Grid analysis packages, on the other hand, are not designed for navigation but for map analysis. As in most strategic planning it involves a statistical representation of geographic space.

In addition, travel-time analysis is affected by artificial edges of the project area. Some of the streets within the analysis window were designated as infinitely far away because their connectivity is broken by the analysis window’s border. Also, if the grid cells are large, false connections can be implied by closely aligned yet separate streets.

Generally speaking, the analysis window should extend a bit beyond the specific area of interest and contain cells that are as small as possible. The rub is that large grid maps exponentially affect performance. While the 10,000-cell grid in the example took less than a second to calculate, a 1,000,000-cell grid could take a several minutes. Larger maps also require more storage and adversely affect the transfer of information between grid systems and desktop mapping systems. A user must weigh the errors and inaccuracies of a simple buffer against the increased requirements of grid processing.

Effective Proximity2 Buffer (MapCalc)

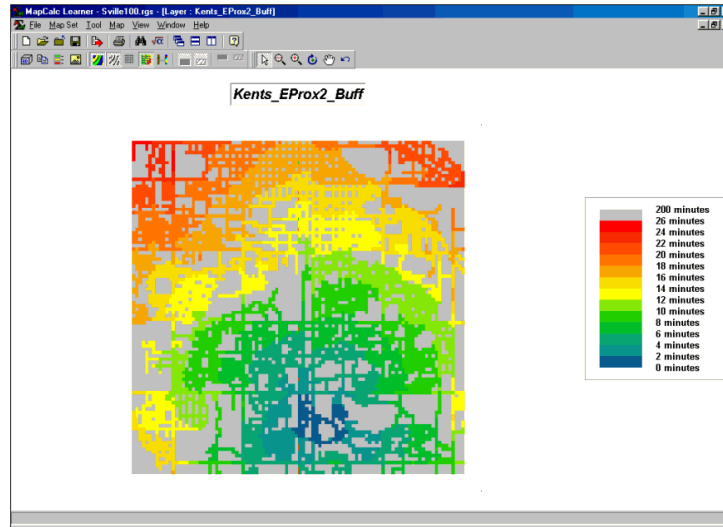


Figure 4-5. The reach of the travel-time buffer can be extended to the entire analysis grid.

Figure 4-5 extends the reach to encompass the entire analysis grid. Note that the farthest location from the store appears to be about 20 minutes away and is located in the northwest corner. While the proximity pattern has the general shape of concentric circles, the effects of different speeds tend to stretch the results in the directions of the primary streets.

Note: see <http://www.innovativegis.com/basis/MapAnalysis/Default.html>, Topic 14, *Deriving and Using Travel-Time Maps for more information on travel-time analysis.*

4.3 Transferring Grid and Vector Data

The previous section described a procedure for calculating travel-time grid surfaces. It involves establishing an appropriate analysis grid then transferring the point, line or polygon features that will serve as the starting point (e.g., a store location) and the relative/absolute barriers to travel (e.g., a street map). The analytical operation simulates movement from the starting location to all other map locations and assigns the shortest distance respecting the relative/absolute barriers. If the relative barrier map is calibrated in units of time, the result is a travel-time map that identifies the time it takes to travel from the starting point to any map location.

The information derived in the grid package is easily transferred to a desktop mapping system as standard tables, such as ArcView's *.SHP* or MapInfo's *.TAB* formats. In a sense, the process simply reverses the "burning" of information used to establish the Store and Street layers. A pseudo grid is generated that represents each cell of the analysis grid as a polygon with the grid information attached as its attributes. The result is polygon map with an interesting spatial pattern—all of the polygons are identical squares that abut one another.

Transferring and classifying the pseudo grid polygons into travel-time intervals generated the large display in figure 4-6. Each polygon grid cell is assigned the color interval corresponding to its distance and displayed as a backdrop for the line work of the streets, water and other vector features.

Transferring Travel-Time Information

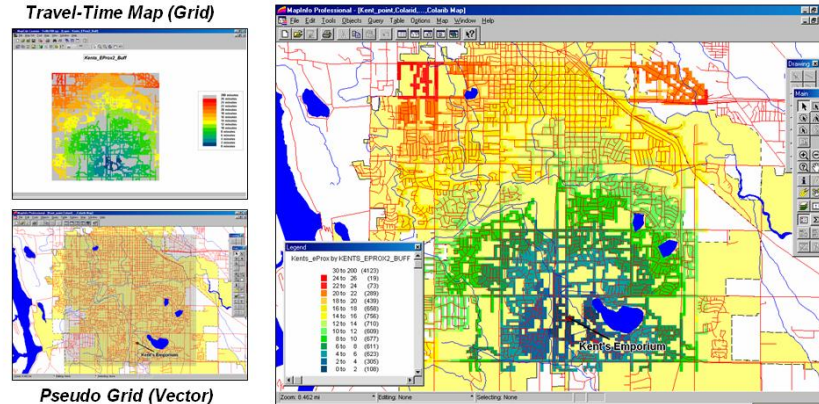


Figure 4-6. The travel-time map can be imported into most generic desktop mapping systems by establishing a pseudo grid.

More importantly, the travel-time values themselves can be merged with any other map layer, such as “appending” the file of the store’s customers with a new field identifying their effective proximity. The small map in the top-left of figure 4-7 is a display of the transferred travel-time map. Recall that in a vector system this map is stored as a pseudo grid with a separate polygon representing each grid cell—100 columns times 100 rows= 10,000 polygons in this example. The cells serve as a consistent parceling of the project area and any information derived during grid processing is simply transferred and appended as another column to the pseudo grid’s data table.

Integrating Travel-Time Information

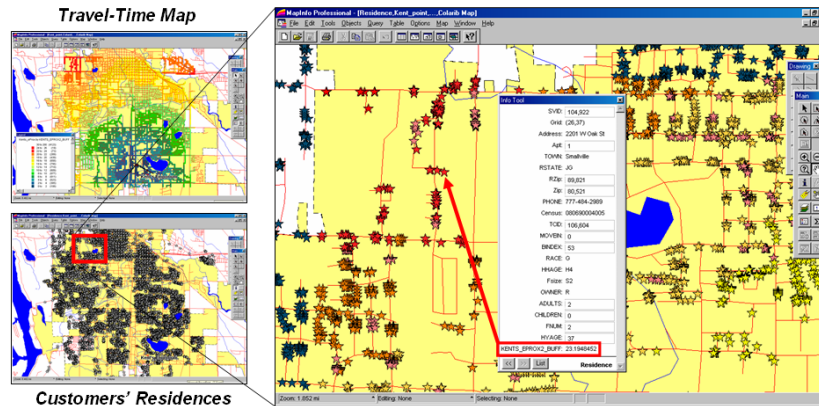


Figure 4-7. Travel-time and customer information can be joined to append the effective distance from a store for each customer.

But how is this grid information integrated with the data tables defining other vector layers? For example, one might want to assign a computed travel-time value to each customer’s record identifying residence (spatial location) and demographic (descriptive attributes) information. The small map in the bottom-left of figure 4-7 depicts the residences of the customers of Kent’s Emporium’s as point locations derived by address geo-coding. At this point we have a map containing a set of polygons (grid cells) with travel-time information from Kent’s and another containing set points locating its customers.

Most desktop mapping systems provide a feature for “spatially joining” two tables. For example, MapInfo’s “Update Column” tool can be used for the join as specified as “...where object from table <Sville100> contains object from table <Residences>”— Sville100 is the pseudo grid and Residences is the point map (see figure 2). The

procedure determines which grid cell contains a customer point then appends the travel-time information for that cell (*K_TTime*) to the customer record. The process is repeated for all of the customer records and the transferred information becomes a permanent attribute in the Residences table.

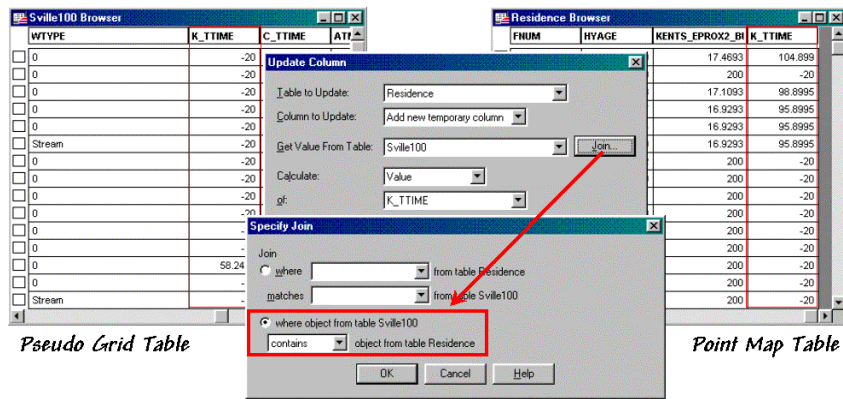


Figure 4-8. A “spatial join” identifies points that are contained within each grid cell then appends the information to point records.

The result is shown in the large map on the right side of figure 4-7. The stars that identify customers’ residences are assigned “colors” depicting their distance from the store. The “info tool” shows the specific distance that was appended to a customer’s record. At this point, the derived travel-time information is fully available in the desktop mapping system for traditional thematic mapping and geo-query processing.

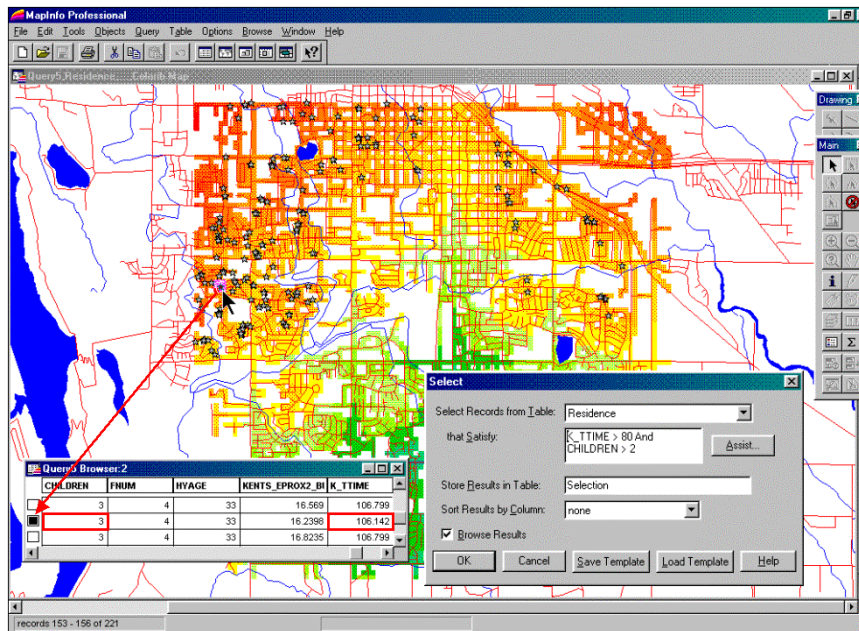


Figure 4-9. The appended travel-time information can be utilized in traditional geo-query and display.

For example, the updated residence table can be searched for customers that are far from the store and have more than three children. The dialog box in the lower right corner of figure 4-7 shows the specific query statement. The result is a selection table that contains just the customers who satisfy the query. The map display in the figure plots these customers and shows a “hot link” between the selection table and one of the customers with three children who live 10.6 minutes from the store.

The ability to easily integrate travel-time information greatly enhances traditional descriptive customer information. For example, large families might be a central marketing focus and segmenting these customers by travel-time could provide important insight for retaining customer loyalty. Special mailings and targeted advertising could be made to these distant customers.

Applications that benefit from integrating grid-analysis and geo-query are numerous. Initially, the analytical capabilities were limited to large and complex GIS systems requiring custom application development. However more recently, grid-based extensions to desktop mapping systems have begun to bridge the processing gap. Special-purpose buttons based on grid data, such as effective distance, density mapping, and spatial interpolation and 3-D display are becoming commonplace.

As awareness of grid-analysis capabilities increases and applications crystallize, expect to see more map analysis capabilities and a tighter integration between the grid and vector worlds. In the not so distant future all GIS systems will have a travel-time button and wizard that steps you through calculation and integration of the derived mapped data.

4.4 Creating Visual Exposure Maps

Another example of spatial analysis involves assessing visual connectivity. Like travel-time, it links a location to other map locations but in this instance the procedure determines whether the location are seen or not seen. This information has numerous uses particularly in real estate and land development.

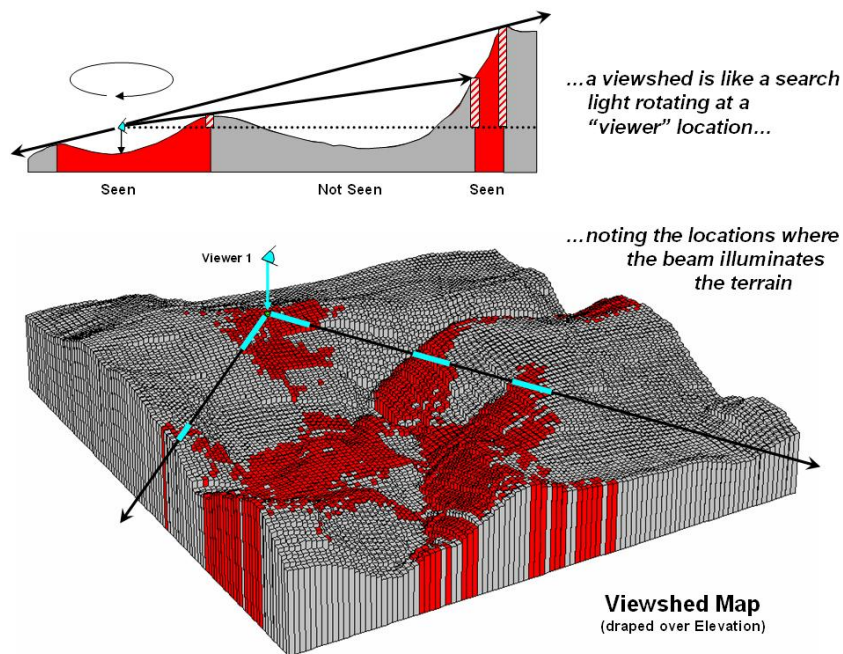


Figure 4-10. Identifying a viewshed is analogous to noting the areas illuminated by a search light.

The algorithm uses a series of expanding rings to determine relative elevation differences from the viewer position to all other map locations. Elevation differences that are less than those in previous rings are not seen. The top portion of figure 4-10 illustrates the procedure. The ratio of the elevation difference (*rise* indicated as striped boxes) to the distance away (*run* indicated as the dotted line) is used to determine visual connectivity. Whenever the ratio exceeds the previous ratio, the location is marked as seen (red); when it fails it is marked as not seen (grey).

To conceptualize the procedure, imagine a searchlight illuminating portions of a landscape. As the searchlight revolves about a viewer location the lit areas identify visually connected locations. Shadowed areas identify locations that cannot be seen from the viewer (nor can they see the viewer). The result is a *viewshed map* as shown draped over the elevation surface in figure 1. Additional considerations, such as tree canopy, viewer height and view angle/distance, provide a more complete rendering of visual connectivity.

The top portion of figure 4-11 shows the viewsheds from three different viewer locations. Each map identifies the locations within the project area that are visually connected to a specified viewer location. Note that there appears to be considerable overlap among the “seen” (red) areas on the three maps. Also note that most of the right side of the project area isn’t seen from any of the locations (grey).

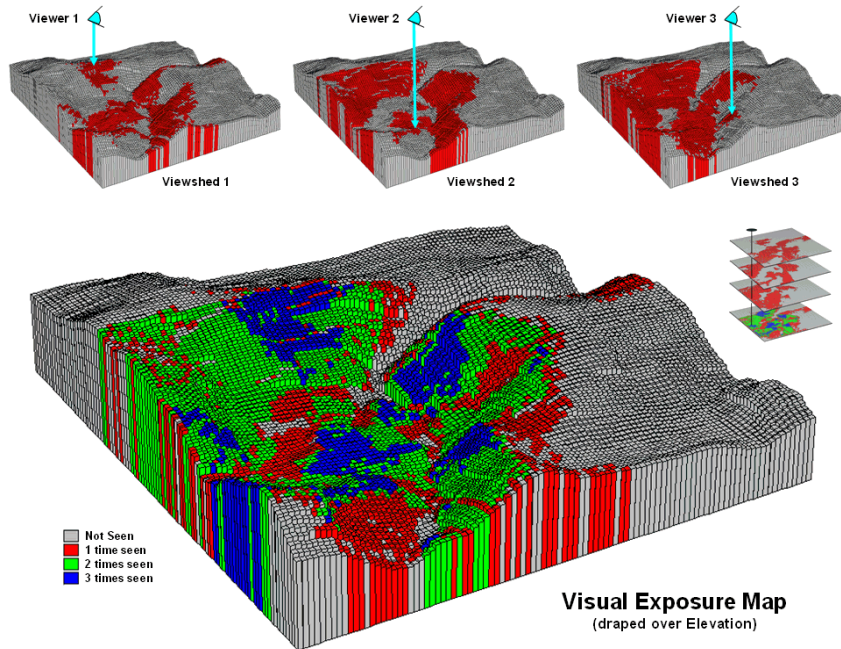


Figure 4-11. Calculating a visual exposure map.

A *visual exposure* map is generated by noting the number of times each location is seen from a set of viewer locations. In figure 4-11 this process is illustrated by adding the three viewshed maps together. The resulting visual exposure map in the bottom of the figure contains four values—0= not seen, 1= one time seen, 2= two times seen and 3= three times seen—forming a relative exposure scale.

The top portion of figure 4-12 shows the result considering the entire road network as a set of viewer locations—sort of an elongated eyeball composed of hundreds of individual grid cells. In addition, the different road types are weighted by the number of cars per hour. In this instance the total “number of cars” replaces the “number of times seen” for each location in the project area.

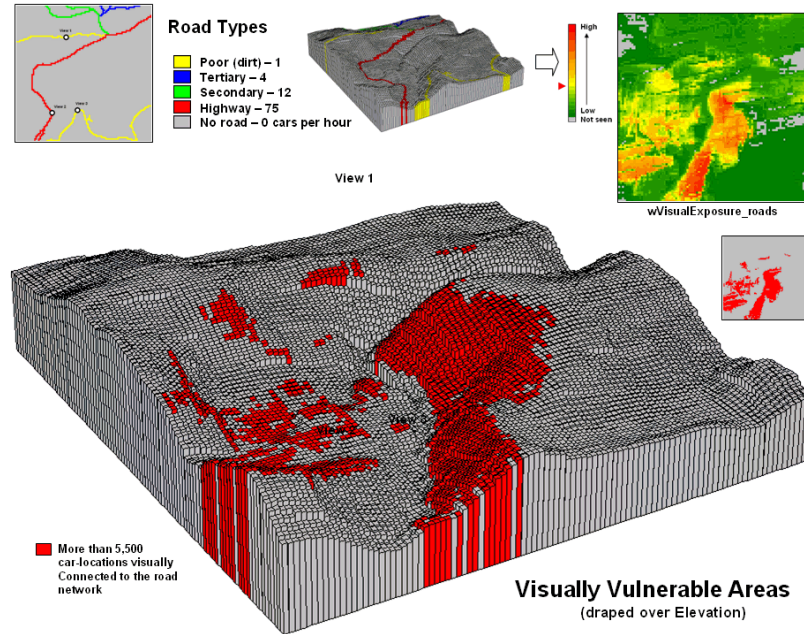


Figure 4-12. Calculating a visual vulnerability map.

The effect is that extra importance is given to road types having more cars yielding a *weighted visual exposure map*. The relative scale extends from 0 (not seen; grey) to 1 (one car-location visually connected; dark green) through 12,614 (lots and lots of visual exposure to cars; dark red). In turn, this map was reclassified to identify areas with high visual exposure—greater than 5,500 car-locations (yellow through red)—for a map of *visual vulnerability*.

A visual vulnerability map can be useful in planning and decision-making. To a resource planner it identifies areas that certain development alternative could be a big “eyesore.” To a backcountry developer it identifies areas whose views are dominated by roads and likely a poor choice for “serenity acres.” Before visual analysis procedures were developed, visceral visions of visual connectivity were conjured-up with knitted-brows focused on topographic maps tacked to a wall.

Today, detailed visual vulnerability assessments, as well as travel-time analyses, are easily derived. These and a host of other analytical capabilities illustrate how advanced spatial analysis operations are radically changing the business information environment.

Note: see <http://www.innovativegis.com/basis/MapAnalysis/Default.html>, Topic 15, *Deriving and Using Visual Exposure Maps* for more information on visual analysis.

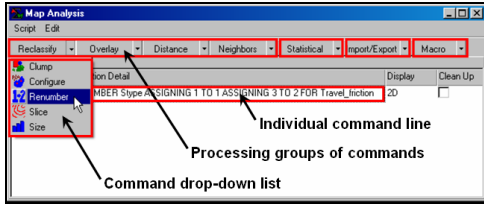
4.5 Exercises

Access *MapCalc* using the *GB_Smallville.rgs* by selecting **Start** → **Programs** → **MapCalc Learner** → **MapCalc Learner** → **Open existing map set** → **GB_Smallville.rgs**. The following set of exercises utilizes this database.




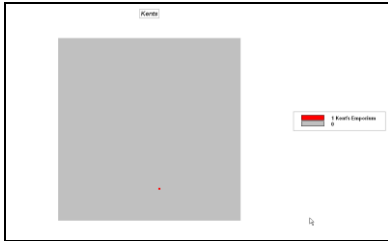
Pressing the *Map Analysis* button provides access to all of the analytical operations in MapCalc—Reclassify, Overlay, Distance, Neighbors and Statistical. Note the organization of the map analysis menu as shown in the following figure. Press the *close* button in the upper right corner of the window to close it.


4.5.1 Calculating Simple Proximity

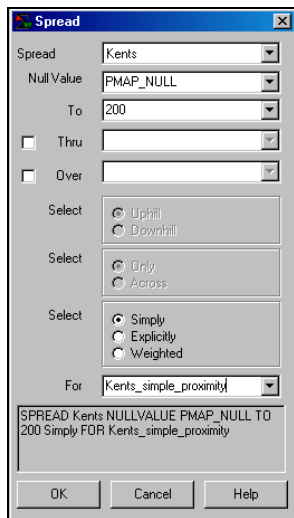



The map analysis window is used to process maps and develop analytical models. The following sequence of commands determines the travel-time from Kent’s Emporium to all locations along the street network in the Smallville project area.

 Use the View button to obtain a display of the Kent’s Emporium location (lower center).



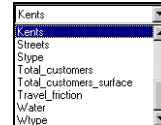
 Click on the *Map Analysis* button and select **Distance**→ **Spread** to pop-up the *Spread* dialog box. Note that the command window is composed of a series of “hot-fields” you will complete to perform desired processing.





 Before entering anything, press the *Help* button to get information on the Spread command. The online help information contains a brief description of the command, an example


of its use and a detailed description of the commands input fields.


To generate a simple proximity map from Kent’s Emporium entering the following (leave the default settings for input fields not mentioned)—

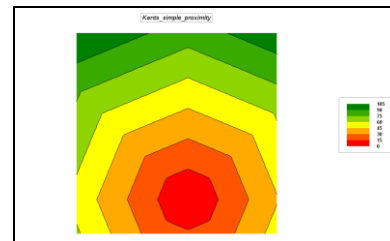
 Click on the *down arrow* next to the top hot-field and select **Kents** from the drop-down list of maps currently in the database. This identifies Kent’s as the starting point for calculating proximity as it is the only non-zero value embedded in the “starters” map.


 Enter **200** (up to 200 cells away) as the extent of the proximity reach. Since the analysis area is only 100cols by 100rows a reach of 200 insures that proximity will be calculated throughout the project area.

 Enter the map aname **Kents_simple_proximity** for the new map that will be created.

 Note that as you were entering command parameters the command line at the bottom kept track of your input.

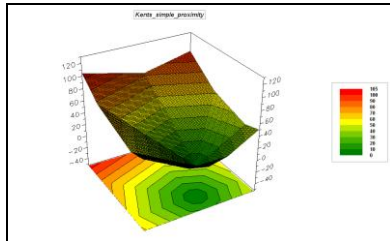
 Press **OK** to create and display the simple proximity map.



 The default display is *2D, Lattice* with *seven equal ranges* and *color ramp* from red to green with a mid-range inflection color of yellow.

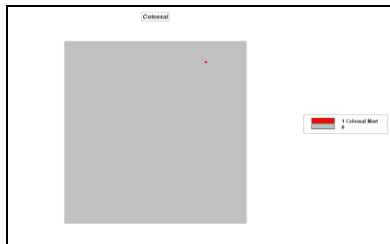
Based on your experience from the Topic 3’s exercises, use the **Shading Manager** to change the display to *3D, Lattice* with *eleven equal ranges* and *color ramp* from red to green with a mid-range inflection color of yellow.

Weight	Min	Max	Count	Area	% of Total Area	Color	Label
100	100	37	531	0.37	0.01	Red	On
80	100	396	560	4	0.01	Orange	Off
60	80	885	1,241	8.6	0.02	Yellow-Orange	Off
40	80	1036	1,486	10	0.02	Yellow	Off
20	80	1146	1,647	11	0.02	Light Green	Off
0	80	1436	2,050	14	0.03	Green	Off
0	40	1547	2,220	15	0.03	Light Green	Off
0	30	1340	1,923	13	0.02	Light Green	Off
0	20	1076	1,544	11	0.02	Light Green	Off
0	10	891	1,202	8.4	0.02	Light Green	Off
0	0	281	403	2.8	0.00	Dark Green	Off

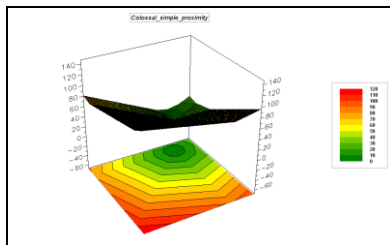


Note the “bowl-shape” of the proximity surface with increasing distance values the farther away from the starting location.

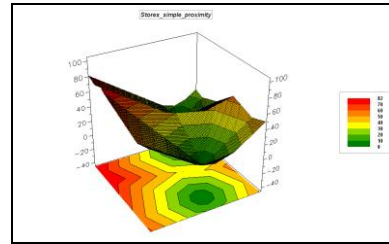
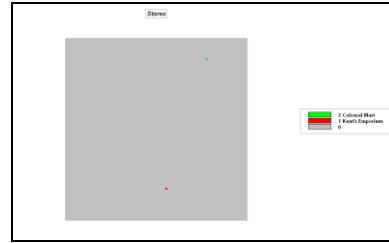
Use the *View* button to display the map of **Colossal Mart**'s location.



Click on the Map Analysis button, select **Distance** → **Spread** and complete the dialog box as before except specifying **Colossal** in place of **Kents**.



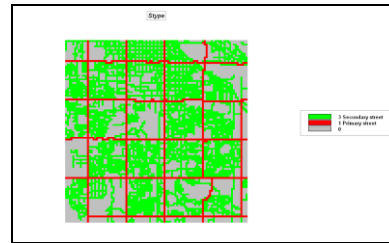
Repeat the processing, however this time use the **Stores** map identifying both Kent's Emporium and Colossal Mart.



Explain the “double-dip” shape of the proximity surface.

4.5.2 Creating Travel-Time Maps

Use the *View* button to obtain a display of the street type map (*Stype*). Note that there are two types of streets in the project area—Primary and Secondary.



It is assumed that movement along secondary streets is three times slower than movement along primary streets.

Click on the Map Analysis button, select **Reclassify** → **ReNUMBER** and complete the dialog box to generate the command line...

RENUMBER Stype ASSIGNING 1 TO 1
ASSIGNING 3 TO 2 FOR Travel_friction

From the drop-down list of maps select **Stype**.

Enter the “assignment” phrase for the primary streets that

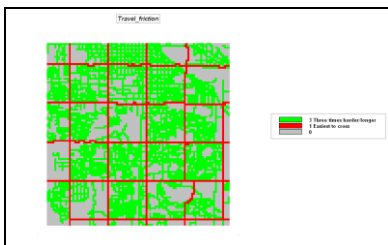
assigns the new value of 1 (easiest to cross; relative friction) to the old value of 1 (street type 1) and press the *Add* button to register the assignment.

Enter the “assignment” phrase for the secondary streets that assigns the new value of 3 (three times more difficult to cross) to the old value of 2 (street type 2) and press the *Add* button to register the assignment.

The accumulative assignment list should have two entries.

Enter the map name **Travel_friction** for the new map that will be created. The completed renumber command window should like the one below.

Press the **OK** button to create the travel friction map.



Click on the *Map Analysis* button, select **Distance**→ **Spread** and complete the dialog box as before specifying **Kents** as the “starter” map and **Travel_friction** as the “Thru” map indicating relative ease of travel— 1= easiest, 3= three times longer and 0= can’t go (absolute barrier to vehicular travel). The completed

dialog box should look like the one below. If so, then press **OK** to generate the travel-time map from Kent’s.

Display the map as 2D, Grid with seven Equal Ranges and color ramp from red to green with a yellow mid-range inflection point (see below).



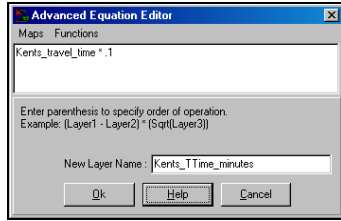
Moving the cursor about the map shows the relative travel-time. Locations with the value 200 assigned are “200 or more units away” (dark green) that are actually infinitely far away for a car as they are not part of the street network. The farthest away location reachable by car is in the upper-left corner (NW) reporting 171 base cells away.

Field study shows that it takes on the average one-tenth of a minute (6 seconds) to traverse a primary street grid cell. The knowledge that the base value of 1 cell is equated to .1 minute can be used to recalibrate the map to “minutes” away from Kent’s.

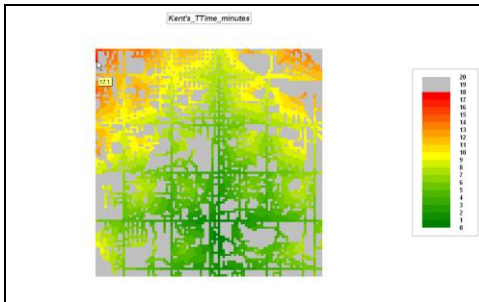
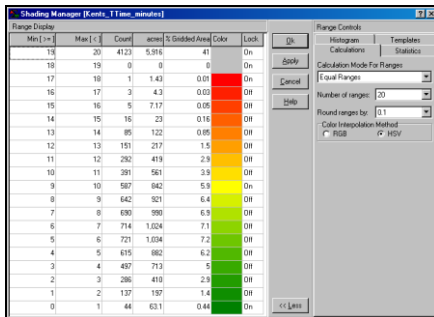


Click on the *Map Analysis* button, select **Overlay**→ **Calculate** and complete the dialog box as shown below. Click on the “Maps” menu

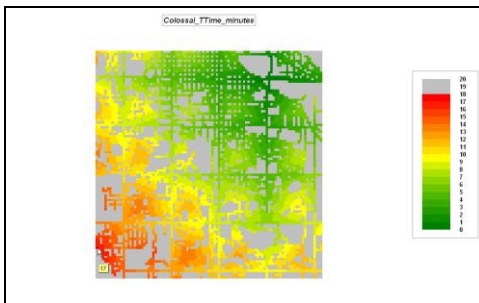
item to pop-up a list of current maps and select **Kents_travel_time**. Complete the equation to read “Calculate Kents_travel_time * .1” and name the output map **Kents_TTime_minutes**.




Display the map as 2D, Grid with seven Equal Ranges and color ramp from red to green with a yellow mid-range inflection point (see below).




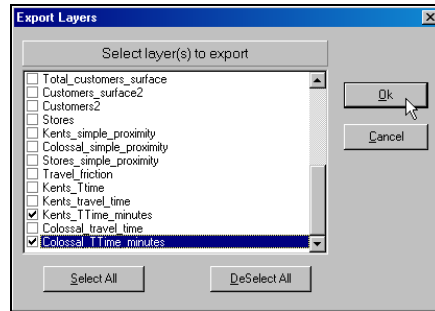
If time permits, repeat the processing to create a travel-time map from Colossal Mart.



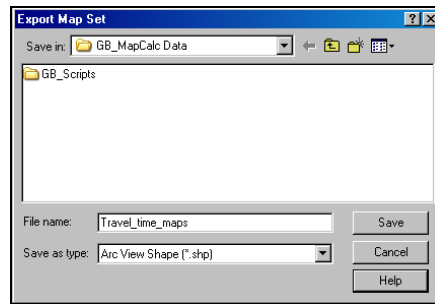
4.5.3 Transferring Maps

 From MapCalc’s main menu select **File** → **Export** → **Regions** to begin the export wizard.

 In the *Export Layers* window press the **Deselect All** button to clear the export list. Click on the **Kents_TTime_minutes** and **Colossal_TTime_minutes** maps to select them for export. Press **OK**.



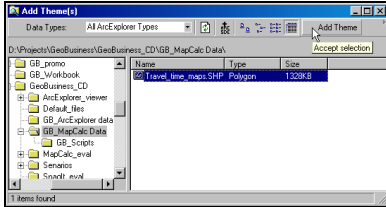
Specify **Travel_time_maps** as the output file name and **ArcView Shape (*.shp)** as the file type. Press **Save**.



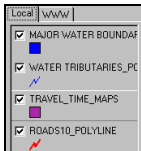
Access *ArcExplorer* using the *Sville.aep* database by selecting **Start** → **Programs** → **ESRI** → **ArcExplorer 2.0** → **ArcExplorer**. From ArcExplorer’s main menu, select **File** → **Open** and specify **...\\Sville.AEP** as the project to open. Press the **Open** button to plot the data.

Refer to the exercises with Topic 2 for guidelines on data display using ArcExplorer. The basic procedures for displaying the exported maps are outlined below.

 Press the **Add Theme** button.

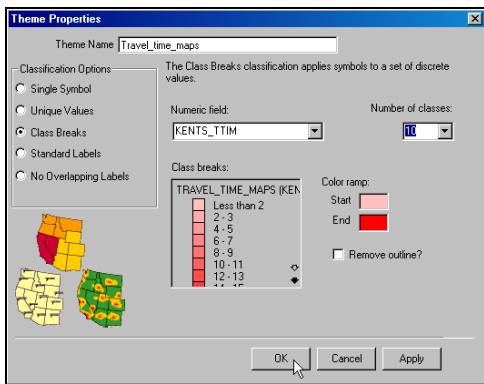


Navigate to the **Travel_time_maps.shp** file you stored in the ... \GB_MapCalc Data folder and click on the **Add Theme** button.

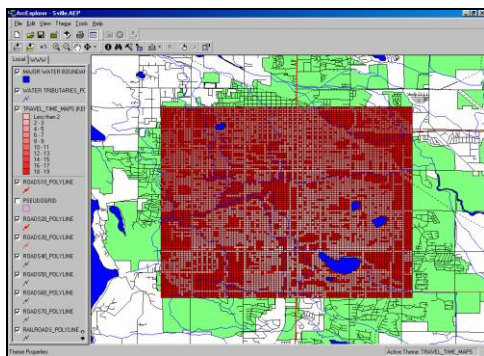


View the **Travel_time_maps** layer by clicking on its **checkbox**. **Click and drag** the map below the Water Tributaries layer as shown.

Double-click on the **Travel_time_maps** layer to pop-up its *Theme Properties* dialog box.

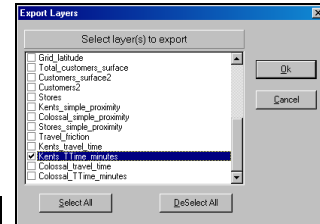


Select **Class Breaks** as the classification option, specify **KENTS_TTIM** as the numeric field and set **10** as the number of classes. Press **OK** to generate a thematic map display of the transferred data.



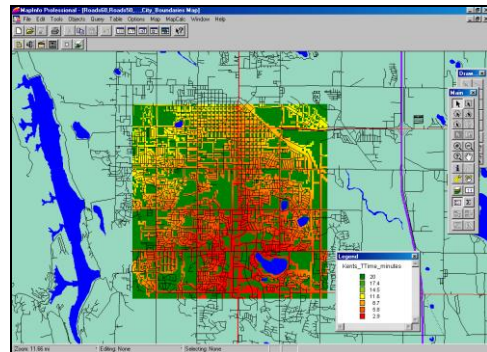
If you have MapInfo installed on your computer the data transfer is much easier and retains the color ramp sets you developed in MapCalc.

From MapCalc’s main menu select **File** → **Export** → **Send to MapInfo** to begin the export wizard.



Press the **Deselect All** button to clear the selections and click on **Kents_TTime_minutes** map to select it for export. Press **OK** to continue.

The data will be transferred and MapInfo accessed to automatically export/import/display the selected map layers.

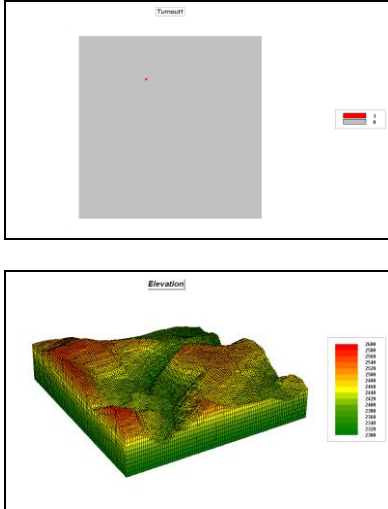


Exit ArcExplorer and access the MapCalc system.

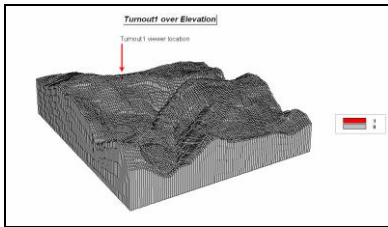
4.5.4 Creating Visual Exposure Maps


Press the *Open Existing File* button, choose to close the **GB_Smallville.rgs** database then open the **GB_Bighorn.rgs** one.

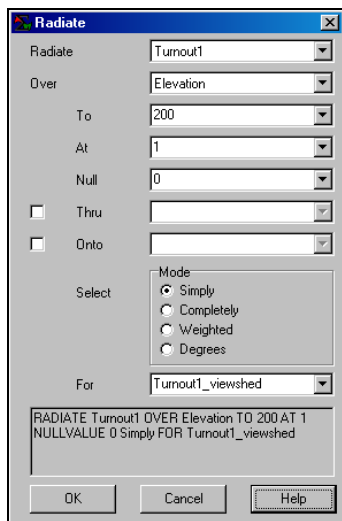
use the *View* button to generate a display of the **Turnout1** map that locates a point in the northwest portion of the project area. This point will serve as a “viewer” location for generating a viewshed map. Repeat to generate a display of the **Elevation** surface that will serve as barriers to visual connectivity.



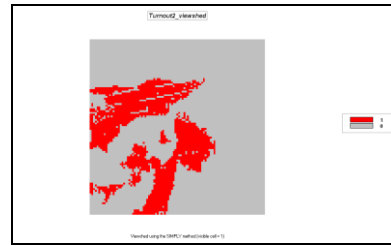
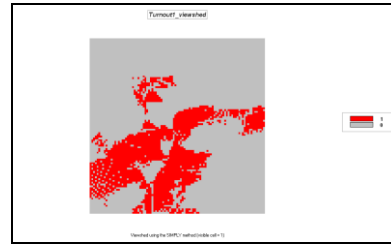
Generate a display draping the Turnout1 viewer location on the Elevation surface by selecting **Map → Overlay → Turnout1** from the main menu.




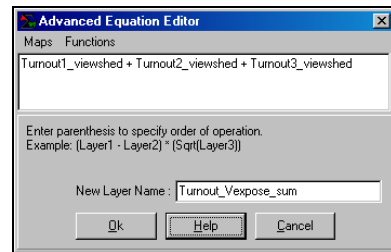
 Click on the *Map Analysis* button and select **Distance → Radiate**. Complete the dialog box as shown below specifying the **Turnout1** map as the viewer, **Elevation** as the barrier surface, 200 (cells away) reach and **Turnout1_viewshed** as the output map.




Repeat the similar processing to generate **Turnout2_viewshed** and **Turnout3_viewshed** maps.

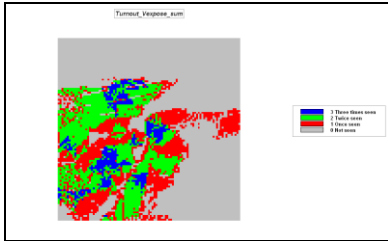



 Click on the *Map Analysis* button and select **Overlay → Calculate**. Enter the following equation to compute a visual exposure surface (number of times seen) using the three viewshed maps. Click on the menu item **Maps** to get a listing of the current maps for selecting the appropriate maps.

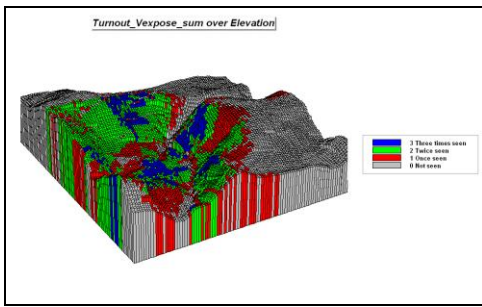



 Display the **Turnout_Vexpose_sum** map as 2D, grid, and discrete (*Data Type* button). Double-click on the map's legend and enter the labels as a value of 0= not seen (grey), 1= seen

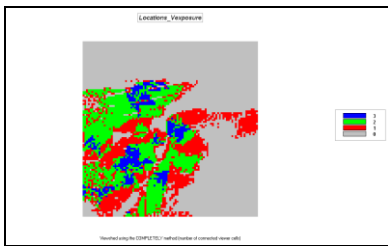
once (red), 2= seen twice (green) and 3= seen three times (blue).




 Use the View tool to display the **Elevation** surface then drape the *Turnout_Vexpose_sum* map **Elevation** surface by selecting **Map** → **Overlay** → **Turnout_Vexpose_sum** from the main menu.

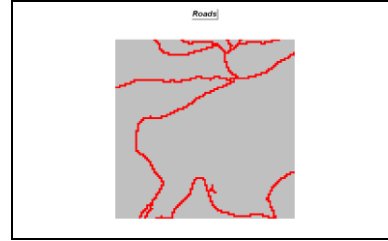


 Re-access the *Radiate* command (**Distance** → **Radiate**) and complete the following to generate a visual exposure map (number of times seen) in a single command line.




Note that this map is identical to the one created by adding the individual viewsheds—*number of times seen*.

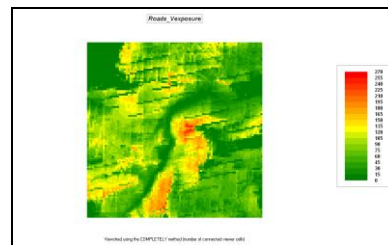
 Use the View tool to display the **Roads** map.



Category	Count	Area	% Gridded Area	Color
1 Roads	621	138	6.3	Red
0	9180	2,042	94	Grey


Double-click on the legend to pop-up its Shading Manager. Note that there are 621 viewer cells (*Count* column).

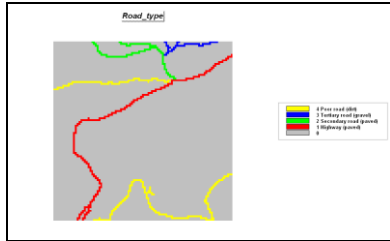
 Re-access the *Radiate* command (**Distance** → **Radiate**) and complete the following to generate a visual exposure map (number of times seen) from the entire road network.




Display the resultant map as 2D, Grid with 18 Equal Ranges from green to red with a mid-range yellow inflection point. This map reports the number times each map location is seen from the set of cells defining the road network. Values range from 0 (not seen at all) to 270 times—indicating over 40% of the road network

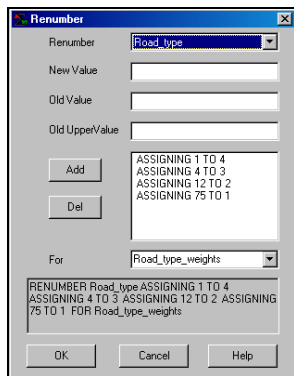
sees the highest value locations at the center of the project area ($270/621=43\%$).


 Use the **View** tool to display the **Road_type** map.

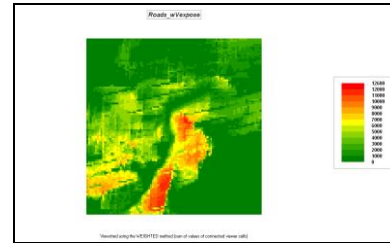
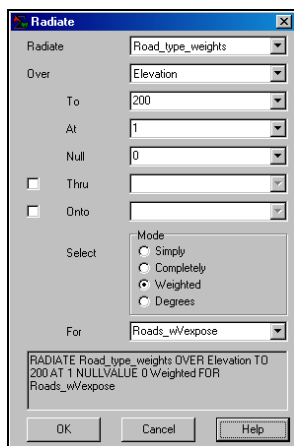



This map depicts the road network by the type of road and can be calibrated to represent the relative amount of traffic.

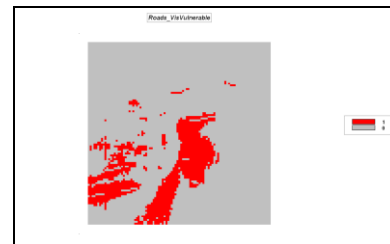
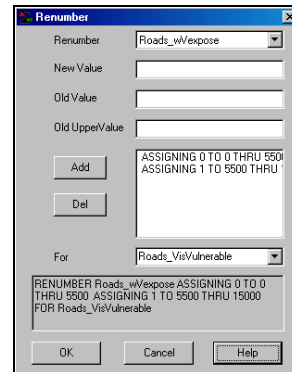
 Use **Reclassify** → **Renumber** to calibrate the road type map for relative traffic flows.



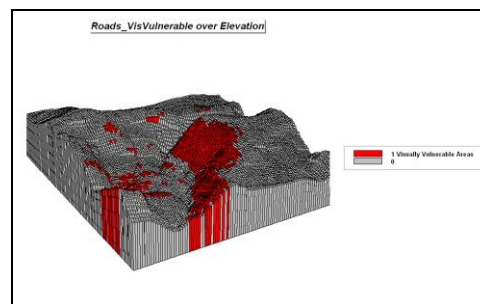
 Use **Distance** → **Radiate** to generate a weighted visual exposure map to roads that recognizes increased visual exposure for locations connected to heavily used roads.



 Use **Reclassify** → **Renumber** to isolate areas with very high visual exposure to roads.



Drape the **Roads_VisVulnerable** map on the 3D display of the Elevation surface (remember how?).



On your own, generate a visual exposure surface using the Water as the viewer map then isolate areas with relatively high visual exposure to water (>40 cells seen) and drape the resultant map on the Elevation surface.

