2.0 METHODOLOGY

As discussed above, the methodology can be divided into three major tasks: (1) DEM processing — delineating streams and watersheds and computing flowlength, (2) generating HRAP cells and properly defining their geographic position relative to the streams and watersheds, and (3) merging DEM data with HRAP cells to determine the contributing area and mean travel length from each HRAP cell to the watershed outlet(s). Figure 2.1 summarizes these tasks and provides a more detailed listing of the steps required to accomplish each task.

2.1 OBTAIN AND PROCESS DIGITAL ELEVATION MODEL

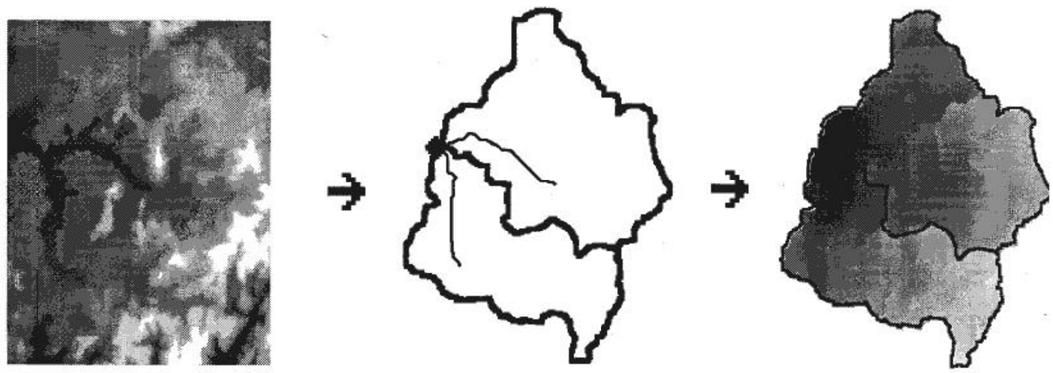
2.1.1 Create a "Hydrologic" DEM for the Region of Interest

2.1.1.1 Locating Study Region

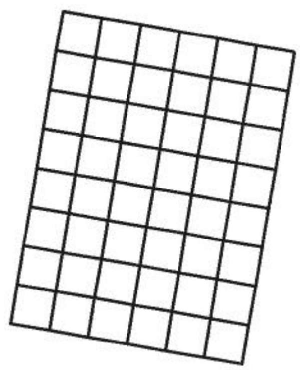
Working with 3" DEMs consumes a considerable amount of disk storage (approximately 9.8 megabytes for a 1° by 1° block in raw form) and processing time; therefore, several steps are recommended to eliminate large portions of DEM data outside of the study area.

Since watersheds most often define the boundaries of a hydrologic study, an easy first step in establishing a digital reference frame is to identify and cut out USGS Hydrologic Cataloging Units (HUCs) of interest. A hydrologic unit is uniquely identified by an eight digit code. The first two digits of this code identify the hydrologic region, the third and fourth digits identify the hydrologic subregion, the fifth and sixth digits identify the accounting unit, and the seventh and eighth digits identify the cataloging unit. An Arc/Info coverage of the HUCs in the United States at 1:250,000 scale (huc250) contains the first eight digits (HUC), the first six digits (HUC6), the first four digits (HUC4), and the first two digits (HUC2) of the hydrologic unit codes as separate attributes. ArcViewII was used to identify the Tenkiller basin by viewing the huc250 coverage of United States basin outlines and zooming in on the region of interest. Simply double-clicking on the watershed of interest yielded the unit code for the Tenkiller basin (HUC = 11110103). Figure 2.2 illustrates different levels of HUC classification. The Tenkiller basin, with its unique cataloging unit, is highlighted.

1. Obtain and Process DEM



2. Generate a Coverage of Rainfall Cells and Transform to the Chosen Coordinate System



3. Merge Processed DEM and HRAP Cells to Create Zones of Uniform Precipitation - Extract Statistics for DEM Cells in These Zones

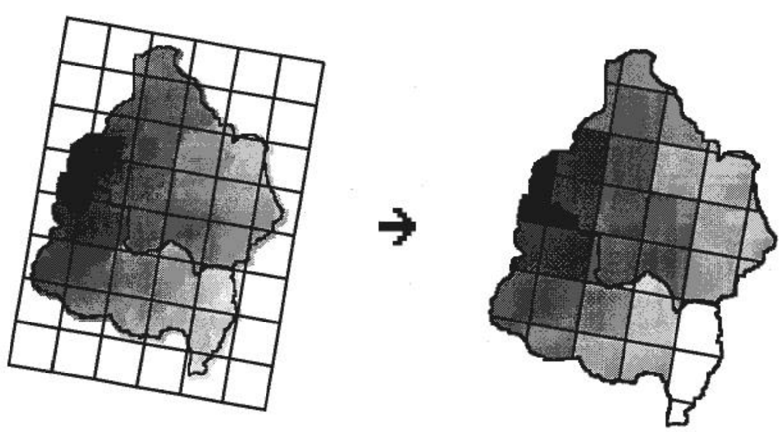


Figure 2.1: Procedure Overview

1. Obtain and Process DEM

Create a "hydrologic" DEM for the region of interest

1 Identify and cut out HUCS of interest
2 Create a buffer around the HUCS to extract relevant portions of DEM's
3 Obtain 3" DEM's from Internet and convert to ARC/INFO Grid
4 Transform buffered HUCS and DEM to a common map projection
5 Clip the relevant DEM region with buffered HUCS
6 Create "Hydrologic" DEM (i.e., FILL)

Process DEM for stream and watershed delineation.

1 Compute Flowdirection
2 Compute Flowaccumulation
3 Identify "streams" by threshold
4 Label stream links
5 Generate point coverage of relevant gaging stations
6 Identify outlet cells
7 Delineate watersheds
8 Convert Grids to Coverages as needed

Process DEM for Travel Length or Travel Time Parameter

1 Compute Flowlength (on sub-watershed or watershed level)
2 Compute integrated "time-index" parameter (includes computation of slope and FLOWACCUMULATION on a sub-watershed level)

2. Generate a coverage of rainfall cells in the chosen coordinate system.

1 Given the vertical and horizontal extent of the study area in latitude and longitude, write a file of HRAP coordinates that will cover the study area.
2 Transform the HRAP coordinates into geocentric coordinates and write a file that can be used to generate a polygon coverage.
3 Write a file of polygon ID's and HRAP coordinates that can be attached to the polygon attribute table (PAT) of the HRAP coverage.
4 Generate a polygon coverage.
5 Transform the geocentric HRAP polygon coverage into the common coordinate system.
6 Attach HRAP coordinates as attributes to the PAT of the HRAP coverage.

3. Merge processed DEM and HRAP Cells to Create Zones of Uniform Precipitation — extract statistics for DEM cells in these zones..

1 Intersect HRAP cells with sub-watershed boundaries to create homogeneous precipitation zones.
2 Convert the result of step 1 to a grid of zones.
3 Compute statistics for values defined in the parameter grids (i.e. flowlength or "time-index" grids) based on zones defined in 2.
4 Establish Info Relates between sector_cov.pat and the statistics files.
5 Write HRAP cell parameters to a file that can be used as input to modClark: (i.e. write: hrapx, hrapy, mean flowlength to outlet, cell area).

Figure 2.1(cont.) Step Summary

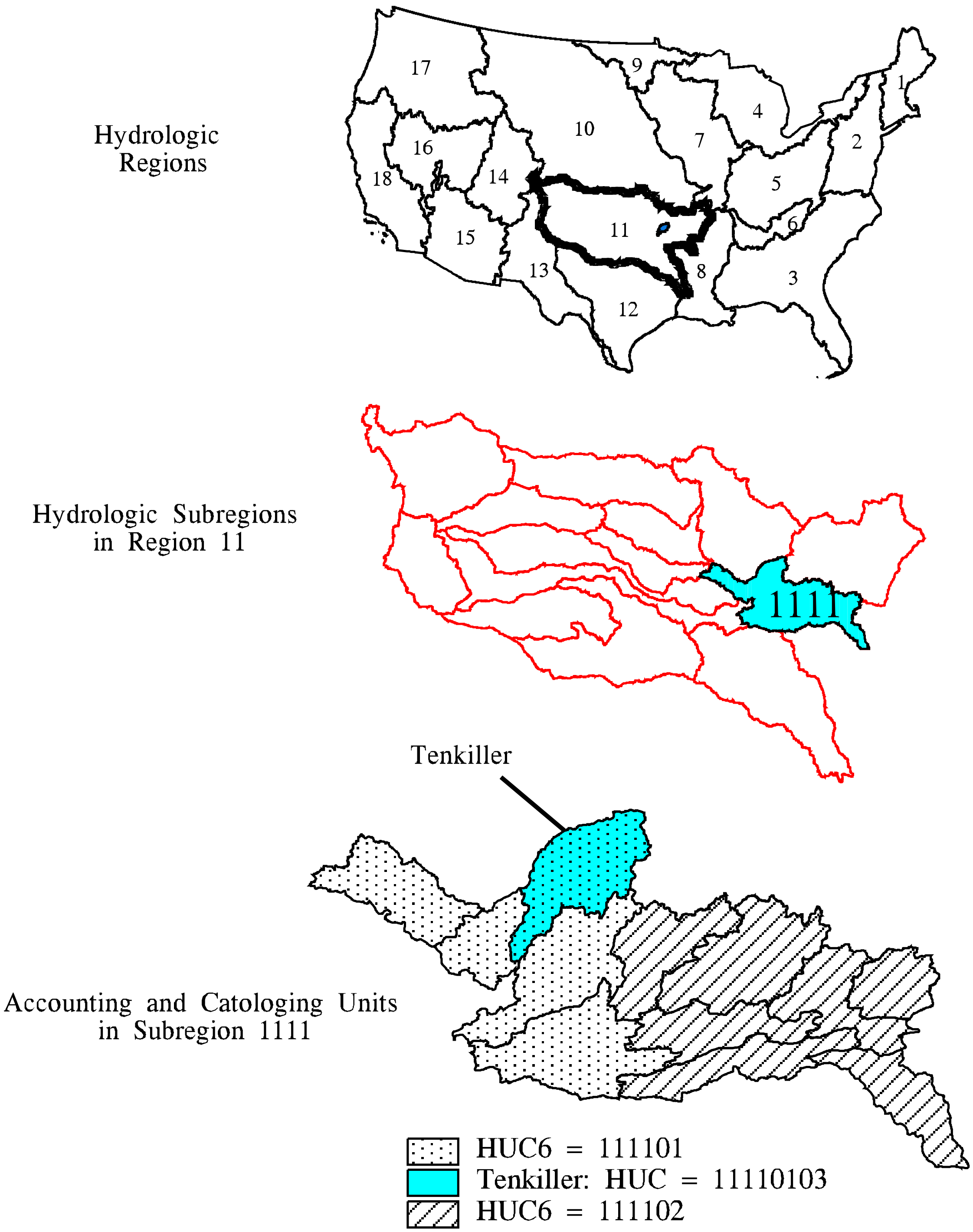


Figure 2.2: Identifying the Tenkiller Hydrologic Cataloging Unit

Although the ArcViewII Query Builder used to generate Figure 2.2 was useful for creating displays of selected polygons (watersheds), the actual coverage (huc250) remained unchanged at this point. To create a new HUC coverage containing only the Tenkiller basin, the Arc Reselect command was used. For this study, Reselect extracted a set of map features from the input coverage huc250 based upon the eight digit HUC attribute to produce an output coverage huctk

RESELECT <in_cover> <out_cover> {in_feature_class}

Arc: **reselect huc250 huctk poly**

Reselecting POLYGON features from HUC250 to create HUCTK

Enter a logical expression. (Enter a blank line when finished.)

>: **res huc = 11110103**

>: ~

n

n

Next, a buffer was placed around the coverage huctk to ensure that all relevant portions of the DEM were extracted. Because a watershed boundary delineated from a DEM does not exactly coincide with the HUC boundary, creating a buffer around huctk ensures that all DEM cells draining to Tenkiller Reservoir are selected. The Buffer command in Arc/Info was used for this purpose. Figure 2.3 illustrates a 10 km buffer generated for the Tenkiller HUC with the syntax below. Figure 2.3 also shows RF1 features within the Tenkiller basin.

BUFFER <in_cover> <out_cover> {buffer_item} {buffer_distance} {fuzzy_ tolerance} {LINE | POLY | POINT |NODE}

Arc: **buffer huctk huctkbuff # # 10000.0 # poly**

The necessary buffer size cannot be predicted with certainty. In this example, a 10,000 meter buffer was used, but comparison between the Tenkiller HUC boundary and the Tenkiller boundary delineated from the DEM in a later step reveals that a 5,000 meter buffer would have been adequate for Tenkiller.

2.1.1.2 Obtaining Digital Elevation Models

3" DEMs can be obtained from Internet and converted into Arc/Info Grid format. Using Mosaic, the DEMs can be reached by choosing "Open URL" from the "File" menu and entering the address — http://sun1.cr.usgs.gov/eros-home.html — this is the “EROS Home Page.” At the “Home Page,” select “US Geodata (FTP File Access).” At this point, DEM files may be downloaded “via State”, “via Graphics”, or “via Alphabetical

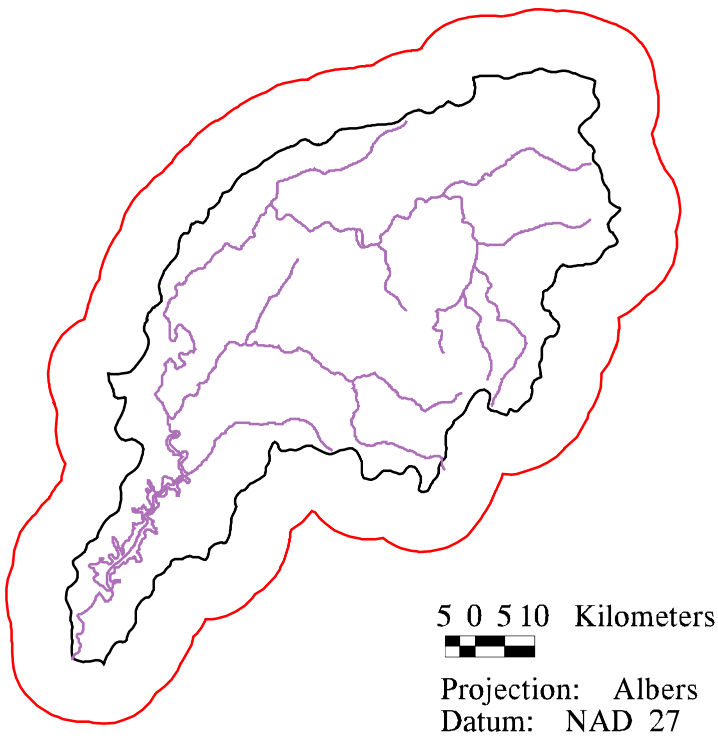


Figure 2.3: Buffered Tenkiller HUC with RF1 Features

List." In any case, the user needs to know the names of the names of the USGS 1:250,000 Index Maps that cover his or her study area. The 1° by 1° blocks downloaded for the Tenkiller analysis were Fort Smith (east and west) and Tulsa (east and west). 3" DEMs may also be obtained by anonymous ftp at edcftp.cr.usgs.gov in the directory /pub/data/DEM/250. DEM files can be downloaded in compressed or uncompressed form. Compressed files can be uncompressed using the UNIX utility gunzip. For example,

Unix: **gunzip -v tulsae.gz**

The United States Geological Survey distributes DEM files in a format that is imcompatible with Arc/Info because the record lengths are too long. A file's record length can be modified using the UNIX command dd. With the syntax printed below, dd adds a carriage return at the end of every 1024 bytes.

Unix: **dd if=tulsae of=tulsaedel ibs=4096 cbs=1024 conv=unblock**

{if — input file; of — output file; ibs — input block size in bytes;

cbs — conversion buffer size; conv=unblock — convert fixed length records to variable length}

A DEM file with appropriate delimiters can be converted into an Arc/Info Grid using the Arc command demlattice.

DEMLATTICE <in_dem> <out_lattice> {USGS | TAME} {z_factor}

Arc: **demlattice tulsaedel tulsaeg USGS**

For Tenkiller, the grids tulsaeg, tulsawg, fsmitheg, and fsmithwg were merged into one grid.

Arc: **grid**

Grid: **totgrid = merge (tulsaeg, tulsawg, fsmitheg, fsmithwg)**

2.1.1.3 Projecting the DEM

In order to extract relevant portions of the DEM for processing, the buffered Tenkiller HUC (huctkbuff) and the DEM grid (totgrid) were transformed to a common map projection. For this step, HUCs (already defined in the standard Albers projection) could be approximately transformed from NAD27 to WGS72; however, this transformation is unnecessary because horizontal shifts due to converting between two ellipsoidal datums are very small relative to the size of the buffer. Therefore, the only transformation required in this step was to project the DEM grid from geographic coordinates into the common Albers projection.

(*) PROJECT (<grid>, {projection_file}, { NEAREST | BILINEAR | CUBIC }, {out_cellsize})

Grid: **totalbg = project (totgrid, geoalb.prj, bilinear, 100.0)**

With this syntax, the projection parameters are specified in the file geoalb.prj. The contents of this file are listed below:

```
input
projection geographic
units ds
datum wgs72
parameters
output
projection albers
units meters
datum wgs72
parameters
29 30 00
45 30 00
-96 00 00
23 00 00
0.0
0.0
end
```

During projection, Arc/Info resamples the elevation points at 3” spacing to generate a grid (totalbg) with uniform spacing of 100 meters. This cell size was chosen to make subsequent area computations simple. Arc/Info offers nearest, bilinear, and cubic resampling algorithms. Either the bilinear or cubic algorithms, in which a weighted distance average of the values in surrounding input cells is taken to determine the value of an output cell, are most appropriate for continuous data such as DEMs. The option for bilinear resampling was specified in this case. When a new elevation point is determined with bilinear sampling, the four nearest elevation values to the desired point in the output projection are used. The units of the input DEM grid are decimal seconds and the units of the output projection are meters. Decimal seconds (DS) and decimal degrees (DD) are common units for geographic coordinates. The conversion is simply, DS = DD * 3600.

Next, the desired region of DEM cells were clipped with the buffered Tenkiller HUC. The Grid Setwindow command reduces the analysis window to the mapextent of a specified coverage (huctkbuff). The mapextent of a coverage is always rectangular in shape. The snap_grid is totalbg which means that any grid created in this analysis window is aligned with totalbg.

SETWINDOW <GRID | COVERAGE | xmin ymin xmax ymax | *> {snap_grid}

Grid: **setwindow huctkbuff totalbg**

Once the analysis window was set, a new grid was created (tkalbg) that contains the values of totalbg within the analysis window.

Grid: **tkalbg = totalbg**

These two commands reduced the grid totalbg containing 4,091,049 data points to a more manageable 1,340,932 data points.

2.1.2 Filling Sinks

Raw DEMs downloaded from Internet may contain one or more sinks. A sink is a grid cell or group of cells that has an elevation no higher than each of its immediate neighbors. Sinks need to be removed for runoff modeling with current Arc/Info algorithms. The removal of sinks creates a so-called "hydrologic" DEM. Two methods for removal of sinks available in Arc/Info are the Grid Fill command and the Arc Topogrid command. The Fill command simply alters the elevation of the sink point(s) to the height of its/their boundary cell(s) with the lowest elevation. The Fill command should be performed after grid projection because data resampling during the projection creates artificial sinks. The Fill command was used in the Tenkiller analysis.

FILL <in_grid> <out_grid> {SINK | PEAK} {z_limit} {out_dir_grid}

Grid: **fill tkalbg tkalbgf**

The Arc command Topogrid provides a more refined method for generating a hydrologically correct grid. Following Hutchinson, 1989, Topogrid uses a more sophisticated drainage enforcement algorithm to fill sinks and it can incorporate digital streamline data into its interpolation scheme. Topogrid is written to generate a hydrologic DEM given a point coverage of elevation values (expected to be more sparse than a DEM with sinks) or a line coverage of elevation contours. Although cumbersome, Topogrid can be used to modify raw USGS DEMs, enforcing drainage and forcing delineated streams to coincide more closely with digitized streams. Elevation points in USGS DEMs are rounded to the nearest meter, often resulting in large flat areas in stream valleys for which no accurate flow direction can be inferred. By incorporating digital streamline data, Topogrid more accurately represents the land surface in stream valleys. Use of Topogrid requires obtaining digital streamline data, obtaining DEM or hypsography data, transforming these data into a common coordinate system, and making sure that all stream arcs are oriented downstream. Experiments using Topogrid in the Allegheny watershed were successful on a region of approximately 90,000 cells but unsuccessful on a larger region of approximately 530,000 cells. Because of difficulties working with large numbers of elevation points, Topogrid was not used in the Tenkiller analysis.

An AML script could have been used to execute the Arc/Info and UNIX commands described in this section. The box below contains a simple script that could have been used to create a hydrologic DEM for the region of interest. Of course, a more sophisticated AML with error checking, flexibility to accommodate an arbitrary number of DEM files, input files located in different directories, messages to the user, menus, etc., could be written.

Table 2.1: Summary Script for Creating a Hydrologic DEM

```
reselect huc250 huctk poly  /* Select relevant HUCs
  res huc = 11110103
  ~
  n
  n
buffer huctk huctkbuff # # 10000.0 # poly
/*  Assuming that 4 DEM files have been downloaded from internet
/*  using Mosaic, Netscape, ftp, etc. and given the names
/*  tulsae.gz, tulsaw.gz, fsmithe.gz, and fsmithw.gz
&sv dem1 = tulsae; &sv dem2 = tulsaw
&sv dem3 = fsmithe; &sv dem4 = fsmithw
&sv count = 1
/*  Uncompress, reblock, and import all of the DEM files into Arc/Info format.
&do &while %count% le 4
   &sv filename = [value dem%count%]
   &sys gunzip -v %filename%.gz
   &sys dd if=%filename% of=%filename%del ibs=4096 ~
   cbs=1024 conv=unblock
   demlattice %filename%del %filename%g USGS
   &sv count = %count% + 1
&end
grid    /* Start Grid subprogram.
totgrid = merge (%dem1%g, %dem2%g, %dem3%g, %dem4%g)      /* Merge DEM
grids.
totalbg = project (totgrid, geoalb.prj, bilinear, 100.0)
setwindow huctkbuff totalbg    /* Set analysis window.
tkalbg = totalbg    /* Define a new grid of reduced size.
fill tkalbg tkalbgf    /* Fill sinks.
q    /* Quit grid.
&return
```

2.2 PROCESS DEM FOR STREAM AND WATERSHED DELINEATION

Arc/Info Grid functions can be used to delineate stream networks and watersheds. The first two steps in the watershed delineation process involve assigning a "flow direction" and "flow accumulation" to each grid cell. The Flowdirection function assigns a unique value to a cell indicating to which of its eight neighboring cells it will flow — assuming that water will only flow in one of eight possible directions. Figure 2.4a depicts this eight direction pour point model used to define flow direction. The Flowdirection function takes a grid of terrain elevations as input and produces a grid of flow directions as output as shown in Figure 2.4b and c . By assuming a flow direction to and from each cell, a drainage network has been developed (Figure 2.4d). The number of cells that are upstream of any given cell in a drainage network is termed the Flowaccumulation. The Flowaccumulation function takes the Flowdirection grid as input and stores in each output cell the number of cells that flow into that cell. Streams or major drainage paths can be defined as cells with high Flowaccumulation and a threshold Flowaccumulation value can be used to delineate a stream network. The density of the stream network will depend on the chosen threshold (Maidment and Mizgalewicz, 1993). The syntax for Flowdirection, Flowaccumulation, and stream delineation are:

Grid: **tkfd = flowdirection(tkalbgf)**

Grid: **tkfa = flowaccumulation(tkfd)**

Grid: **tkst = con (tkfa > 10000, 1)**

In this example the threshold chosen to define "streams" was 10,000, a somewhat arbitrary choice. In words, the third command can be translated as — define a grid named tkst and if the corresponding cell in tkfa has a value greater than 10,000 then assign a value of 1 to the cell in tkst; otherwise, assign the cell NODATA.

2.2.1 Creating a Point Coverage of Watershed Outlets

The Grid Watershed command traces upstream from selected outlet points in a drainage network and identifies all cells flowing to specified outlets. The Watershed command

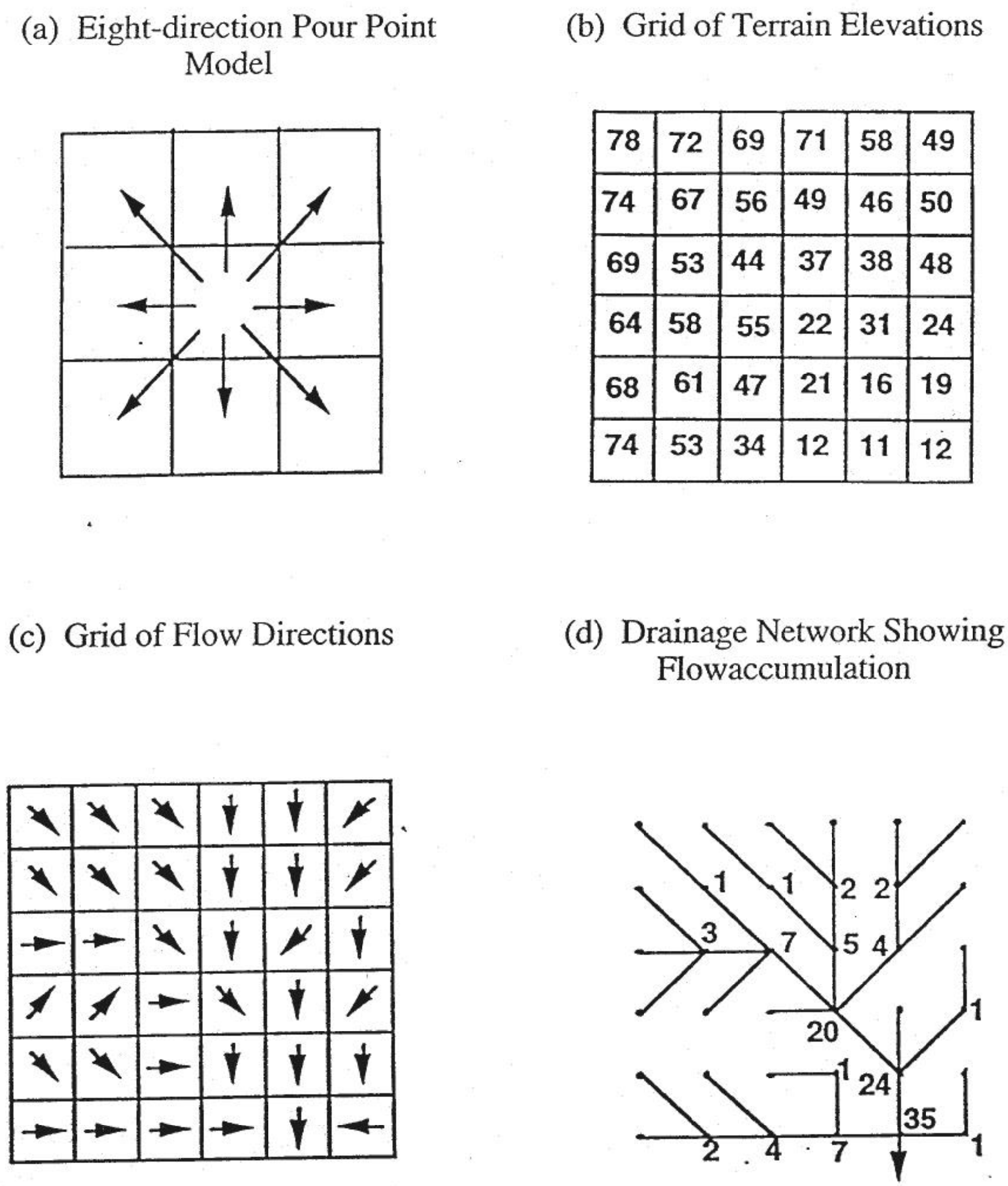


Figure 2.4: Creating a Drainage Network

takes a grid of Flowdirection and a grid of outlet cells as input and produces a grid in which each cell contains the value of the outlet cell to which it flows. Therefore, each cell in the grid of outlet cells must contain a unique value. Coordinates of flow measurement locations provided by the Hydrologic Engineering Center were used to select outlet cells for the division of Tenkiller into subwatersheds. Table 2.2 lists these flow measurement locations.

Table 2.2: Flow Measurement Locations

Location ID	Name	Latitude	Longitude
1	Illinois R. Near Watts, OK	36.1300	-94.5700
2	Illinois R. Near Tahlequah, OK	35.9214	-94.9208
3	Baron F. at Eldon, OK	35.9211	-94.8383
4	Tenkiller Ferry Reservoir	35.5967	-95.0492

As a first step, a point coverage of these outlets was generated.

Arc: **generate tkout**

Generate: **input tkout.gen**

Generate: **points**

Generate: **quit**

The ASCII file used as input (tkout.gen) is listed below. The file contains ID#, longitude, and latitude.

```
1,-94.5700,36.1300
2,-94.9208,35.9214
3,-94.8383,35.9211
4,-95.0492,35.5967
end
```

The point coverage (tkout) was then projected from geographic coordinates into the common Albers projection described in Table 1.2. The datum for the outlet point coordinates was not provided, although they were almost certainly taken from a map with an ellipsoidal rather than a spherical datum; the WGS 72 datum was assumed. As discussed in Section1.4 **,** the magnitude of the errors associated with this assumption did not affect further analysis. The point coverage was projected using:

Arc: **project cover tkout tkoutalb albdd.prj**

Arc: **build tkoutalb point**

The projection file albdd.prj is the same as the projection file geoalb.prj listed in Section2.1.1.3 except the units on the input file are decimal degrees (DD) instead of decimal seconds (DS).

2.2.2 Selecting Grid Cells for Watershed Outlets

In order to delineate watersheds, the outlet points need to be grid cells that lie on the stream network (tkst). Because the outlets in the point coverage tkoutalb do not fall precisely on the stream network, a simple conversion of the point coverage tkoutalb to a grid was not sufficient. An interactive procedure was used to select outlet cells that do lie on the stream network. Also, the values in different outlet cells needed to be unique; therefore, the values for outlet cells were selected from a grid of stream "links," not the grid tkst in which all cells contain the value 1. Given a gridded stream network and a Flowdirection grid, the Streamlink function produces an output grid of the stream network such that each cell in a given stream reach (or link) contains a unique value.

Grid: **tklink = streamlink (tkst,tkfd)**

Before stream outlet points could be selected, the stream network and outlet points were displayed on the screen.

Grid: **display 9999** {If no display}

Grid: **mape tklink**

Grid: **gridpaint tklink** {paint streams}

Grid: **points tkoutalb** {display outlet points}

The mapextent was reduced to the smallest window that contained all outlet points.

Grid: **mape ***

Define the box

Grid: **clear**

Grid: **gridpaint tklink** {refresh streams}

Grid: **points tkoutalb** {refresh outlet points}

In order to zoom in on multiple outlet cells at once, at a high enough resolution to resolve individual cells, the PAN/ZOOM feature of the Arc/Info graphics window was used to create four new windows. An alternative to manually creating each new window is to execute an AML that automatically displays the vicinity of outlet cells. The code for an AML called make_win.aml that automatically creates new windows in the vicinity of outlets is presented in the Appendix. After the new graphics windows were created, the Gridpaint and Points display commands were repeated again to redraw the same features in the new windows.

Grid: **gridpaint tklink**

Grid: **points tkoutalb**

The Selectpoint function allows choice of outlet points. With the * option, cells can be selected with the mouse.

Grid: **tkallout = selectpoint (tklink,*)**

Define the points <9 to END>

A new grid (tkallout) with four cells was created. With a large number of outlets, it might be too difficult to define all the outlet cells in one application of the Selectpoint function. In that case, several outlet grids created with repeated application of the Selectpoint function may be combined using the Merge function. For Tenkiller, the new grid tkallout contains four cells selected from the stream network that were closest to the outlet points in tkoutalb; all other cells in tkallout contain NODATA. The values of outlet cells in tkallout correspond to values in the streamlink grid tklink — these values are 85, 86, 113, and 127. The downstream end of the links with these values correspond to the flow measurement locations 1, 2, 3, and 4 in Table 2.2 . If tkst were used as the input grid to Selectpoint, all the outlet cells would have the same value (1) and the Watershed function would delineate only one watershed.

2.2.3 Delineating Watersheds

With the grid of outlet points properly defined, subwatersheds were delineated.

Grid: **tksubsheds = watershed (tkfd,tkallout)**

Each of the cells in tksubsheds has one of four values, inherited from the cell values in tkallout, identifying the watershed in which that particular cell is located. The values for

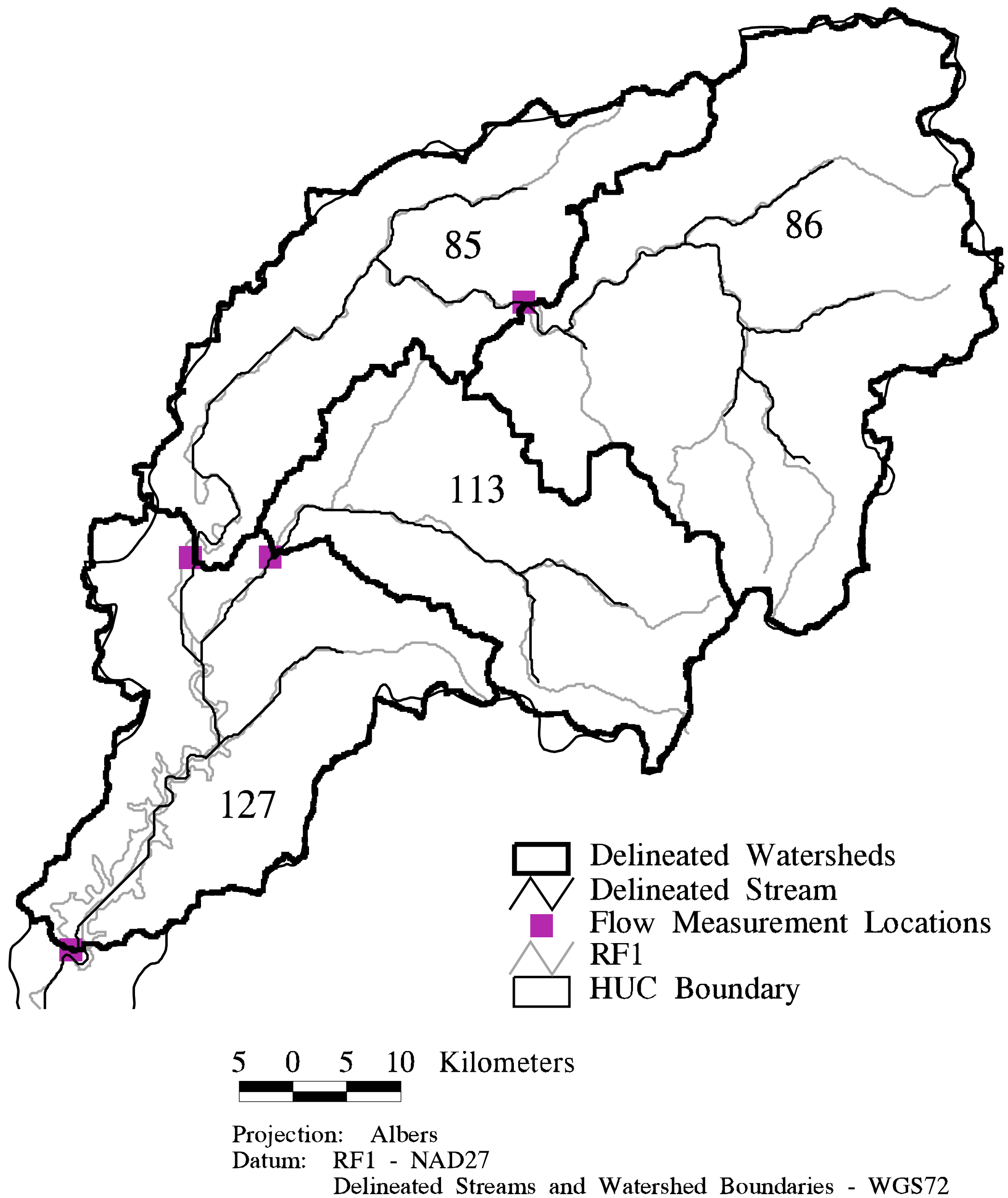


Figure 2.5: Delineated Tenkiller Streams and Watersheds with RF1 Streams and HUC Boundary

each of the Tenkiller watersheds are listed in the value attribute table (VAT) below. One record is listed for each value in the subwatershed grid. The count column lists the number of cells in the grid that contain a particular value; the watershed area (km^2) is equal to count times 0.01 since each DEM cell is 0.01 km^2 in area.

Arc: **list tksubsheds.vat**

Record	VALUE	COUNT
1	85	84515
2	86	155692
3	113	82495
4	127	93628

Figure 2.5 ws delineated Tenkiller streams and watersheds. HUC boundaries and RF1 streams are also included for comparison. Once again, the values 85, 86, 113, and 127 used to label subwatersheds correspond to numbers selected from the Streamlink grid tklink.

Selecting outlet cells in the stream network that are closest to the gaging stations in tkalbout could be automated; however, some degree of engineering judgement may be required. For example, at a stream confluence the user may want to delineate one branch of the stream while the computer might identify a stream cell that is downstream of both branches as the closest to an outlet point, resulting in an incorrect delineation. An example of the need for user judgement or the need for additional information arose in the Tenkiller delineation. The position of the furthest downstream station (Tenkiller Dam) occurs just upstream of a large bend in the delineated stream. This station, along with the stream network delineated from the DEM and streams from EPA's RF1 files is shown in Figure2.6. The RF1 file shows that the outlet should be selected above the stream bend — information that could not be obtained from the DEM alone. The original delineation for Tenkiller was made without this knowledge and is indicated by the lighter boundary in Figure 2.6 labeled "alternate delineation." The difference in area between the two delineations is 25 km^2 or 0.6% of the total watershed area.

2.2.4 Creating a Vector Coverage of Watersheds and Streams

In this procedure, it is necessary to make a raster to vector conversion on the watershed grid so that an intersection can be made with a coverage of rainfall cells in a

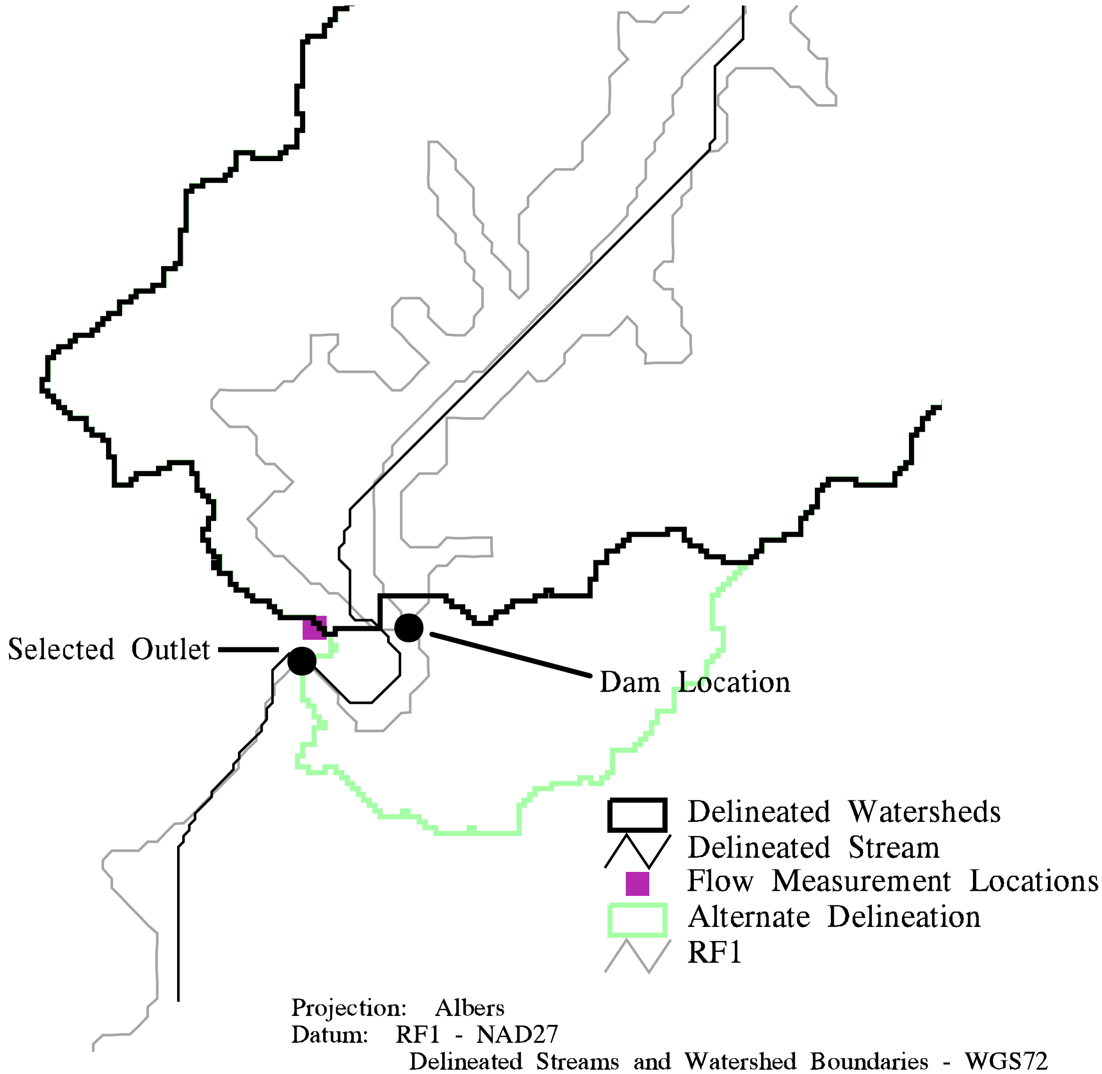


Figure 2.6: Close-up of Dam Location

later step. Raster to vector conversion is also useful for display purposes. Two functions used to convert raster data into vector data are Gridline and Gridpoly. A newer function called Streamline is also available.

Grid: **tkstc = gridline (tkst)**

Grid**: tksubshedsc = gridpoly (tksubsheds,0.0)**

The VALUE item from tksubsheds.vat gets stored as item GRID-CODE in the polygon attribute table of tksubshedsc (tksubshedsc.pat). The second argument to the Gridpoly function specifies a weed tolerance of zero. Weed tolerance is the minimum distance between arc vertices in map units. Choice of weed tolerance is rather subjective. When the weed tolerance is less than the size of one cell, cell boundaries are precisely maintained. With the weed tolerance chosen, there is a potential for small errors in converting cells to polygons if two or more cells identified as belonging to the same watershed do not share a cell-side. An example of this error showed up during the raster to vector conversion at Tenkiller. Figure2.7 shows a close-up of three cells in the grid tksubsheds that drain to the gray watershed (113) but are surrounded on all sides by cells of another watershed (85).

During raster to vector conversion with Gridpoly, these three cells formed their own polygon. As a result, tksubshedsc.pat had one more row entry than expected. Two rows contain the GRID-CODE 113 as shown below.

Arc: **list tksubshedsc.pat**

Record	AREA	PERIMETER	TKSUBSHED S2C#	TKSUBSHEDS 2C-ID	GRID-CODE
1	- 4163299584.0	520200.000	1	0	-9999
2	1556919808.0	297599.96	2	1	86
3	845149888.0	260599.984	3	2	85
4	824919808.0	216999.969	4	3	113
5	30000.0	800.000	5	4	113
6	936280000.0	244200.000	6	5	127

In this listing, area has units of meters squared and perimeter has units of meters. The polygon with GRID-CODE = -9999 is the "universe" polygon or the polygon outside of all watersheds. An Avenue code is being developed to fix the problem of small dangling polygons illustrated in Figure2.7. For now, the small replicate polygon (113) does not impair further execution or affect the final results of this analysis.

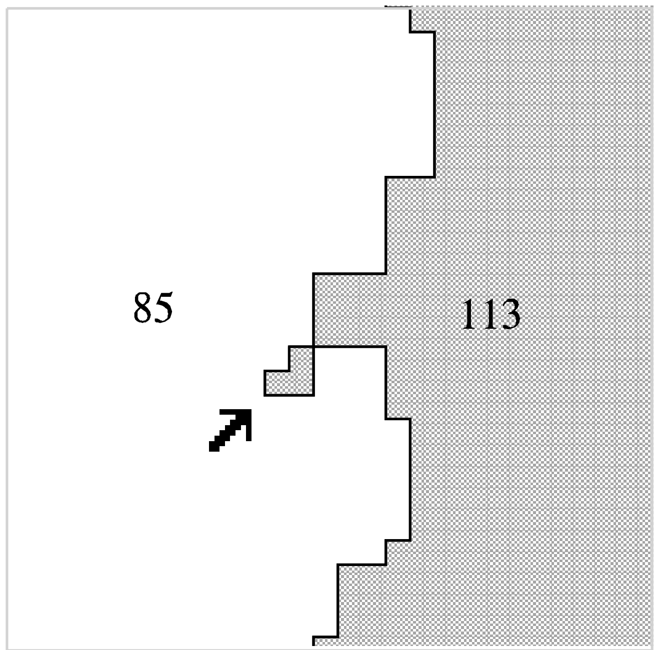


Figure 2.7: A Few Misplaced Cells During Raster to Vector Conversion

The script printed in the box below summarizes the steps described in this section.

Table 2.3 Summary Script for Stream and Watershed Delineation

```
/* Generate a point coverage of flow measurement locations.
/* This is done before all grid processing so that it is not
/* necessary to enter, exit, and re-enter the Grid subprogram.
generate tkout    /* Start Generate subprogram.
input tkout.gen    /* Requires input file of flow measurement locations.
points
quit    /* End Generate subprogram.
project cover tkout tkoutalb albdd.prj
build tkoutalb point
grid    /* Start Grid subprogram.
tkfd = flowdirection(tkalbgf)
tkfa = flowaccumulation(tkfd)
tkst = con (tkfa > 10000, 1)    /* Identify streams.
tklink = streamlink (tkst,tkfd)
&if [extract 1 [show display]] ne 9999 &then    /* If there is not display window
  display 9999    /* then bring one up.
mape tklink
gridpaint tklink
points tkoutalb
mape *    /* Zoom in on an area that contains all four outlets.
clear
gridpaint tklink    /* Paint streams in white.
points tkoutalb    /* Display outlet points.
/* Use the PAN/ZOOM menu to manually create a closeup window for each outlet
/*or type &r make_win tklink tkoutalb
gridpaint tklink
points tkoutalb
tkallout = selectpoint (tklink,*)    /* Select outlet cells.
/* Define the points <9 to END>
tksubsheds = watershed (tkfd,tkallout)
/* Create Arc coverages from grids for future use.
tkstc = gridline (tkst)
tksubshedsc = gridpoly (tksubsheds,0.0)
q   /* Exit Grid subprogram.
&return
```

2.3 PROCESS DEM FOR TRAVEL LENGTH OR TRAVEL TIME PARAMETER

To facilitate distributed modeling with a NEXRAD rainfall cell as the basic modeling unit, a measure of the flow distance or flow time from a rainfall cell to the point at which the flow is to be predicted (most often a watershed outlet) is needed. The development of a drainage network described in Section 2.2 can easily be extended to the computation of travel length from any cell in a DEM grid to a given outlet cell. Arc/Info offers the grid function Flowlength which computes, for each cell in a watershed, the length along the drainage network to either the lowest point in that watershed (downstream option) or to the highest point in that watershed (upstream option). The Flowlength function takes a grid of Flowdirection as input and also provides options for specifying a weight grid. Flowlength computations account for the fact that although all cells in the grid data model are of equal size, three different flow path lengths are possible through a given cell depending on which of eight possible directions the flow enters and leaves the cell. Although a cell-based drainage network clearly does not represent the true flow path that a water droplet falling on the land surface follows to the outlet, it provides a good conceptual model for computing relative travel lengths.

In addition to travel length, the Flowlength option to specify a weight grid opens up further possibilities for estimating a travel time or at least a travel time index for each cell. Maidment *et al.* (1995) proposed assigning a velocity to each cell and computing travel time to the outlet using (1/cell velocity) as the weight grid. Further work by Maidment *et al.* suggested that a model, conceptually similar to Manning's channel flow equation, would be suitable for assigning cell velocities.

$$v = v_{avg} * (S^b A^c) / (S^b A^c)_{avg} \qquad (2.1)$$

S is local slope at a cell; A is upstream drainage area to a cell; b and c are coefficients determined by calibration (values b = c = 0.5 proved to be appropriate); and the subscript avg denotes an arithmetic average over the cells in a particular watershed. It follows from Equation 2.1 that by computing $1/S^b A^c$ for each cell and using this as a weight grid, the Flowlength function would yield a grid of travel time index values. At this point, the physical significance of a time index parameter has not been fully explored. In its draft version, the modClark formulation only requires a travel length measure for each rainfall cell as input; however, the procedure to compute a time index grid has been included in the codes in the Appendix to this report.

Flow_length.aml, presented in the Appendix, computes the travel length from each cell in a subwatershed to the outlet of that subwatershed given an input grid defining subwatersheds and the Flowdirection grid for the same area. A filled DEM is an optional input, required if a time index parameter is to be computed. The names of required input grids are passed to flowlength.aml at the command line. The command syntax used to run flowlength.aml for the Tenkiller watersheds was as follows:

Arc: &r <aml_name> <subshed_grid> <dir_grid> {filled_dem}

Arc: **&r flow_length tksubsheds tkfd tkalbgf**

The main loop in flow_lengtha.aml that controls which cells in the Flowdirection grid are currently being processed uses a programing concept called a Cursor to access individual records in the value attribute table (VAT) of the subwatershed grid tksubsheds.vat — a copy of this table was printed in Section 2.2.3. A Cursor is a mechanism in INFO for accessing a selected set of records in a table. As the Cursor points to each subwatershed (record) in tksubsheds.vat, a conditional statement is tested — if a cell is in the current subwatershed, its Flowdirection is passed to the Flowlength function. Temporary grids are created that contain the Flowlength values for each subwatershed. The names of the temporary grids are stored as character strings; these strings are later concatenated to serve as an argument to the Grid function Merge. The end result is flmerge_grid, a grid of flowlengths for all subwatersheds. Flow_length.aml also graphically displays the results of the Flowlength computations as each subwatershed is processed. The structure of flow_length.aml was modeled after an AML written by Beavers (1993). This structure is useful for automating any Grid functions that need to be performed on individual zones or watersheds. Flowlength.aml calls a "canned" program and its associated menu described by Beavers (1993), msworking2.aml and msworking2.menu; these programs only supply messages to the user and do not affect grid processing. In addition, slope.aml and time_weight.aml may be called in the main loop of flow_length.aml to compute a time index parameter for each cell. Since a time index was not computed for this report, the relevant lines in flow_length.aml have been commented out. Figure2.8 illustrates several steps in the creation of flmerge_grid and shows a gray-shaded image of the final product.

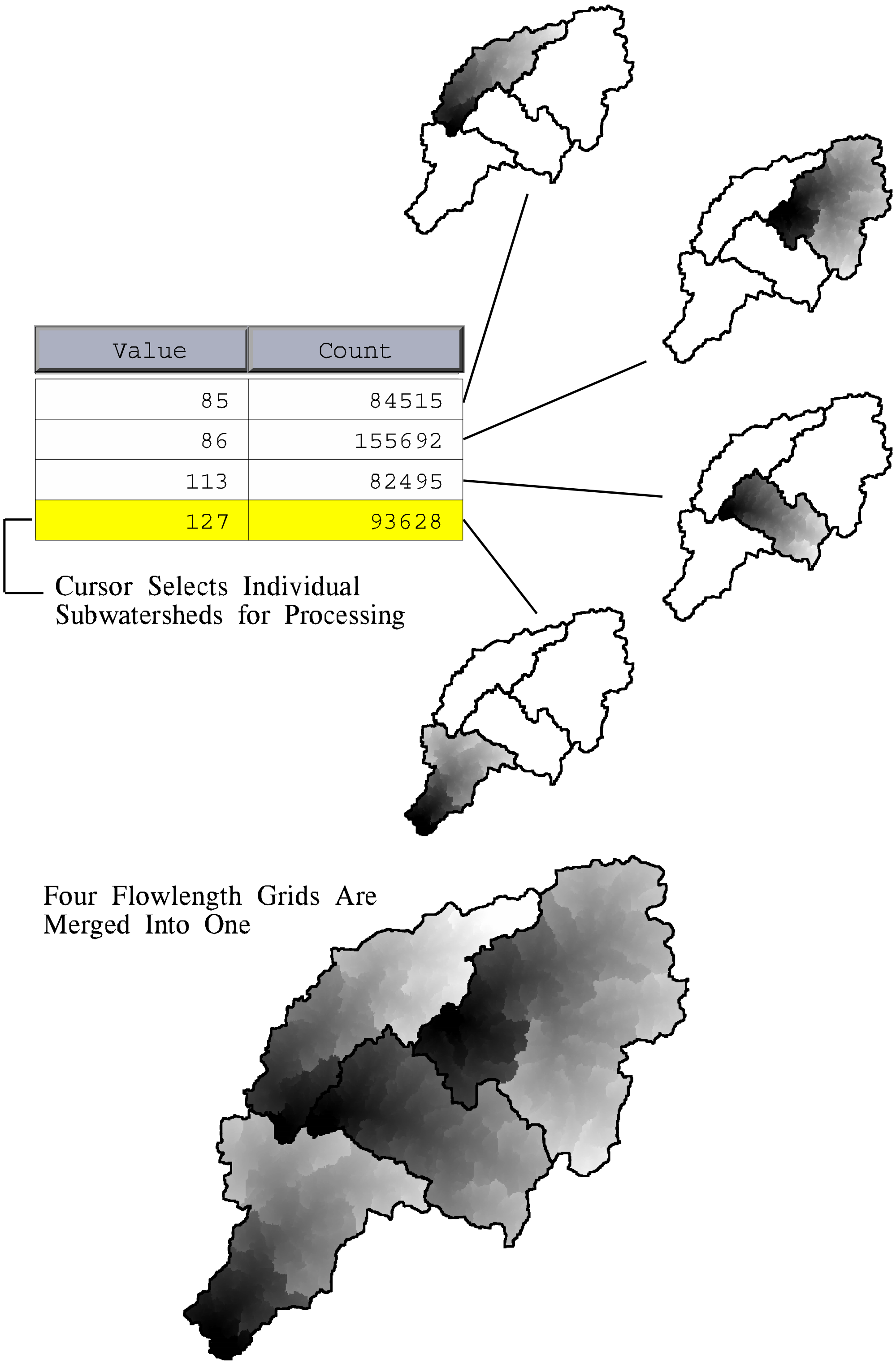


Figure 2.8: Computing Flowlength for Each Sub-basin