

Lesson 43: Processing heterogeneous data

Table of Contents

[Lesson goals](#)

[Topic: Introduction to editing](#)

Concepts

[MajorityFilter](#)

[Expand](#)

[Shrink](#)

[Thin](#)

[Nibble](#)

[BoundaryClean](#)

[Merge](#)

[Mosaic](#)

Example

["Cleaning" heterogeneous data](#)

Exercises

[Edit grids](#)

[Mosaic and merge grids](#)

[Lesson summary](#)

[Lesson self test](#)

Goals

In this lesson, you will learn:

- how to use ArcView Spatial Analyst to clean up heterogeneous or noisy data

Topic 1: Introduction to editing

ArcView Spatial Analyst provides several operations to clean up heterogeneous or "noisy" data. These operations allow you to smooth borders; expand, shrink, or thin selected zones; and filter out small regions.

Grids created from data conversion of an image or other raster source often have many tiny regions, resulting in a grid with too much detail. If your analysis is concerned with the broader patterns in the data, you may eliminate some of the smaller regions.

Some modeling and analysis procedures produce grids that contain barely discernible patterns. An example would be a least-cost path that is only one cell wide. This path may not be a viable solution for some problems (e.g., wildlife corridors may require a wide path, greater than a single cell). You can expand the least-cost path

MajorityFilter

MajorityFilter is used to smooth data by evaluating all values in the input grid by its neighborhood. It replaces the value of each cell in a grid based on the majority value of the contiguous cells. If all values are the same in the neighborhood, except one anomaly (straggler), the anomaly is changed to match the majority of values.

Avenue syntax:

```
aGrid.MajorityFilter (diagonalNeighbors, halfOK)
```

Map Calculator Example:

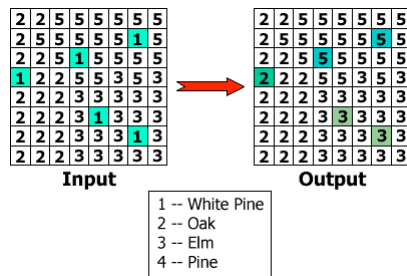
```
[sourceGrid].MajorityFilter (True, False)
```

If the diagonal neighbors argument is true, all eight contiguous neighbors will be used. Otherwise, only the four orthogonal neighbors are used.

If halfOK is true, only half the neighbors are needed to reach majority.

Majority = 3 of 4 or 5 of 8

Half = 2 of 4 or 4 of 8



MajorityFilter replaces cell values with the most frequent value in the neighborhood. It is used to smooth data by removing heterogeneous data

Expand

Expand widens specified zones in an input grid to a specified number of cells.

For example, you may be required to create a highway or wildlife corridor of a certain width. You could use Expand to widen or enlarge the existing corridor. You could also use Expand to create a buffer around a historical district, wildlife preserve, or a stream near a logging site.

Avenue syntax:

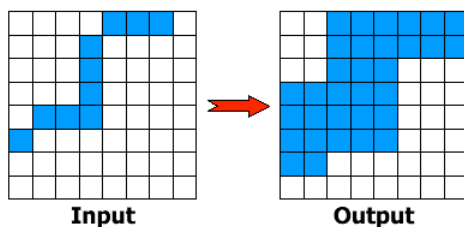
```
aGrid.Expand (aNumberOfCells, aZoneValueList)
```

- aNumberOfCells is the number of cells to expand the specified zones by.
- aZoneValueList is a list of zone values (e. g., {1,8,14}) required by the Expand request. The list contains numbers for each zone that will be expanded by a given number of cells. The list of numbers must be surrounded by the braces {list of numbers}.

Map Calculator example:

```
[inGrid].Expand(1, {1, 4, 6})
```

Values in braces { } denote a list



Expand enlarges a selected zone or list of zones

Shrink

Shrink makes zones smaller by contracting them by a specified number of cells. It is the opposite of Expand. You might use Shrink to remove a thin island zone inside another zone.

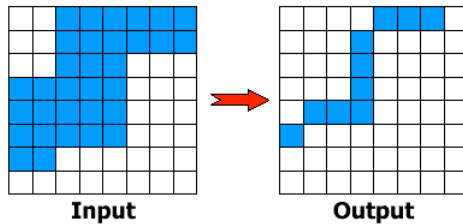
Avenue syntax:

```
aGrid.Shrink (aNumberOfCells, aZoneValueList)
```

- aNumberOfCells is the number of cells to shrink the specified zones by.
- aZoneValueList is a list of zone values (e.g., {1,8,14}) required by the Shrink request. The list contains numbers for each zone that will be shrunk by a given number of cells. The list of numbers must be surrounded by the braces {list of numbers}.

Map Calculator example:

```
[inGrid].Shrink(1, {1, 8, 14})
```



Shrink reduces selected zones by a specified number of cells

Thin

The purpose of Thin is to shrink a line feature down to a width of one cell. Its parameters control how the cells are chosen for thinning.

Thin may be used on a scanned map of elevation contours. If the scanner and base map resolution result in contour lines being five to ten cells wide, Thin reduces the number of cells used to represent the contours. After Thin, each contour line feature will be represented with single cell widths.

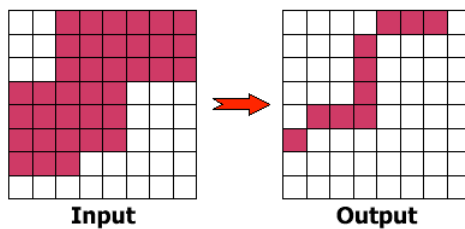
Avenue syntax:

```
aGrid.Thin (notBinary, doFilter, sharpenCorners, maxThickness)
```

- If notBinary is true, all values other than No Data are thinned. If it is false, only values greater than 0 are thinned.
- If doFilter is true, aGrid is filtered to smooth the boundaries between values. If it is false, no filter is applied.
- If sharpenCorners is true, rectangular corners and junctions are preserved. If it is false, all corners and junctions are smoothed.
- maxThickness can be a number or Nil that defines the maximum thickness, in map units, of linear features in aGrid. If maxThickness is set to Nil, a value 10 times the cell size of aGrid is used.

Map Calculator example:

```
[inGrid].Thin(True, False, True, 20)
```



Thin makes raster lines one pixel wide. It is similar to Shrink.

Nibble

Nibble replaces the values of cells in the input grid that correspond to No Data cells in the mask grid with the value of the nearest neighboring cell. Nibble is often used to eliminate noise, speckles, bad scan lines, or bad regions.

The most creative step in using Nibble is identifying the "bad" cells in the input data and creating a No Data mask for those cells. You can create this mask grid with the SetNull request.

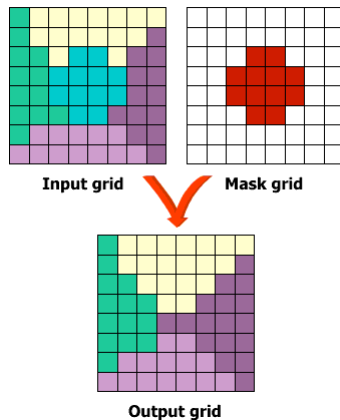
Avenue syntax:

```
aGrid.Nibble (maskGrid, dataOnly)
```

- maskGrid identifies the mask grid to be used.
- If dataOnly is true, only neighbors with values other than No Data are used. If it is false, any neighbor can be used regardless of its value.

Map Calculator example:

```
[inGrid].Nibble([maskgrd], True)
```



Nibble replaces bad cells identified by a mask grid. It uses nearest good values as replacements. No Data cells in the mask grid identify cells to be replaced

BoundaryClean

BoundaryClean cleans ragged edges between zones. It uses the "expand-then-shrink-then-expand-then-shrink" method to clean boundaries. Initially, zones with higher priority invade the neighboring lower priority zones by one cell in all eight directions. Each region takes its turn expanding. Then they all shrink back in the opposite order. The smallest region that can be maintained is a 3 x 3 block of cells.

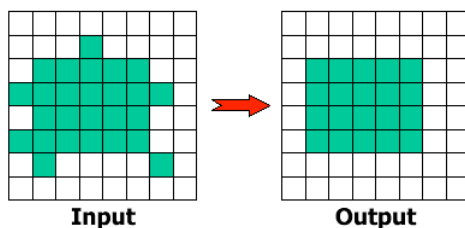
Avenue syntax:

```
aGrid.BoundaryClean (aGridSortTypeEnum, onceOnly)
```

- aGridSortTypeEnum is an enumeration defining the priority different zones have over each other during the process. These enumerations are:
 - #GRID_SORTTYPE_NOSORT--No sorting by size. Zones with larger values have a higher priority to expand into zones with smaller values.
 - #GRID_SORTTYPE_DESCEND--Sorts zones in descending order by size. Zones with smaller total area have a higher priority to expand into zones with larger total area.
 - #GRID_SORTTYPE_ASCEND--Sort zones in ascending order by size. Zones with larger total area have a higher priority to expand into zones with smaller total area.
- If onceOnly is true, the expanding and shrinking process is done only once. If it is false, the process is done twice, with the first one following the priorities set with aGridSortTypeEnum and the second reversing the priorities.

Map Calculator example:

```
[inGrid].BoundaryClean(#GRID_SORTTYPE_ASCEND , false)
```



BoundaryClean cleans ragged edges between zones for integer grids using an expand and shrink process. Priority can be set using zone value or zone area

Merge

The Merge request is normally used to merge adjacent grids.

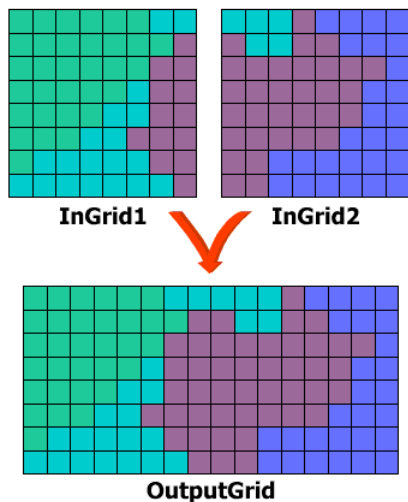
Avenue syntax:

```
aGrid.Merge ( aGridList )
```

A list of one or more grid themes is required for the Merge request. The list of grids must be surrounded by the braces {list of grids}.

Map Calculator example:

```
[inGrid].Merge([Grid1], [Grid2], [Grid3])
```



Merge merges or appends a list of grids. It determines priority of value assignment by order of input grids.

If a grid cell in the input grid is No Data, the value from the first grid in the list is output.

If all the grids are integer, the output is integer. If any input grid is floating point, the output is floating point.

Attributes are not preserved. Normally, merged grids would all contain the same data (e.g., soils). It is possible to merge a mixture of overlapping grids like soils, land use, and roads.

Merge does not interpolate values to fill gaps between input grids. The gap cells are assigned No Data values. However, it is possible to write some map algebra expressions to fill the gaps. The procedure varies depending upon if the data is continuous or discrete. For example, if you used Merge to combine four themes of continuous data as follows:

```
[Elevation].Merge( {[Elevation2], [Elevation3], [Elevation4]} )
```

you could write an expression like this one to fill in the gaps:

```
( [Elevation].isnull ).con( [Elev].FocalStats ( #GRID_STATTYPE_MEAN, NbrHood.MakeRectangle ( 5, 5, false ), false ), [Elev] )
```

In the above process of fixing gaps, if the cell in the merged grid is No Data (IsNull), the FocalStats request is used with a 5 x 5 neighborhood to interpolate the new value. If an input cell contains a value, the corresponding output cell will retain the same value.

The EucAllocation request may be used to fix gaps in discrete data. For example, if you used Merge to combine four land use themes as follows:

```
[Landuse1].Merge( {[Landuse2], [Landuse3], [Landuse4]} )
```

you could write an expression like this one to fill in the gaps:

```
[Landout].EucAllocation ( nil, nil, nil )
```

Mosaic

Mosaic merges an input grid with a list of other grids. Mosaic uses an algorithm (the Hermite cubic proximity algorithm) to smooth the area of overlap. The algorithm helps to make a smooth transition over overlapping areas.

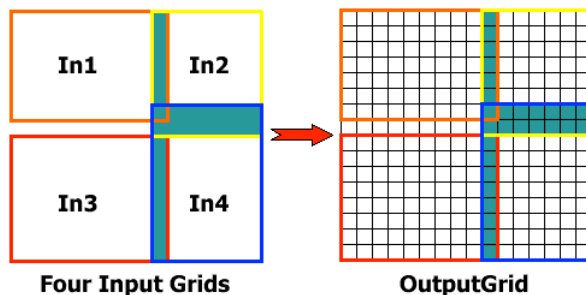
Avenue syntax:

```
aGrid.Mosaic ( aGridList )
```

A list of one or more grid themes is required for the Mosaic request. The list of grids must be surrounded by the braces {list of grids}.

Map Calculator example:

```
[inGrid].Mosaic({[Grid1], [Grid2], [Grid3]})
```



Mosaic merges a list of grids and smooths the overlapping areas. It is best used to merge surface grids, like elevation. Mosaic does not handle gaps in data

Mosaic can merge integer grids; however, it was designed to merge floating point grids of surface data (e.g., DEMs). If the input grids are integer, the output is truncated to integer.

The best results are usually obtained if all the input grids have the same cell size and are registered to each other. If the cell sizes do not match, then the input grids are resampled to the output grid resolution.

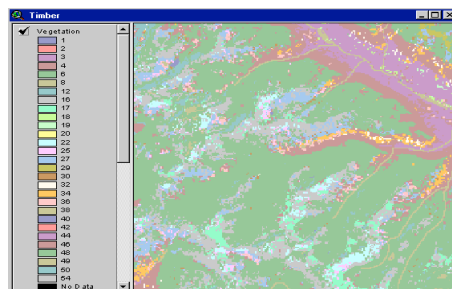
The resampling technique uses the nearest neighbor method, which is inappropriate for floating point grids. Therefore, if the cell sizes do not match, to avoid inappropriate resampling, you should use the Resample request and specify either the bilinear or cubic resampling methods.

Mosaic assigns a No Data value to any gaps. You may fix gaps using the same procedure discussed in the Merge concept

Example

"Cleaning" heterogeneous data

Daniel is a forester working for a timber company and is tasked with identifying areas suitable for growing timber. For his study, Daniel needs to create a generalized grid theme of vegetation for his analysis area. He has obtained a vegetation grid originally derived from satellite imagery. The problem with this theme is that it's too detailed for his study

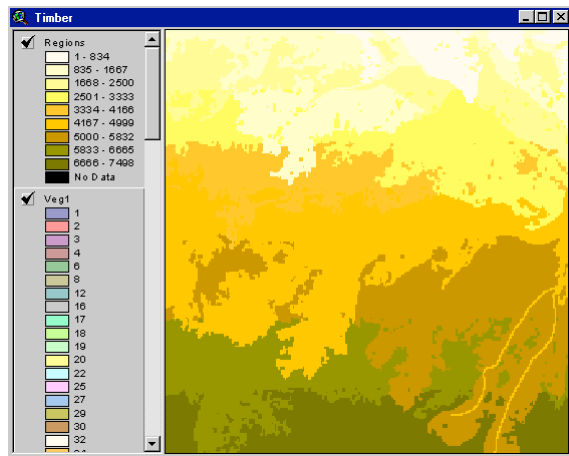


The Vegetation theme after MajorityFilter was used

Notice that many of the speckles have been removed. There are still quite a few areas that are too small for the study. Daniel does not want any regions that contain less than 20 cells to be used in the study. Instead, he wants these areas to be absorbed into the larger, neighboring regions.

Daniel next uses the RegionGroup request to identify regions. In the Map Calculator, he enters the following expression:

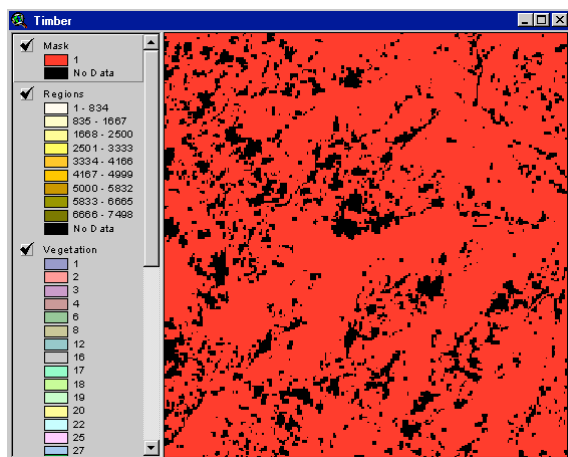
```
[Veg1].RegionGroup(TRUE,FALSE)
```



The Regions theme was created by using RegionGroup on the Veg1 theme.

With the regions identified, Daniel then uses the SetNull request to create a No Data mask to be used with the Nibble request. In the Map Calculator, he enters the following expression to create the [Mask] grid:

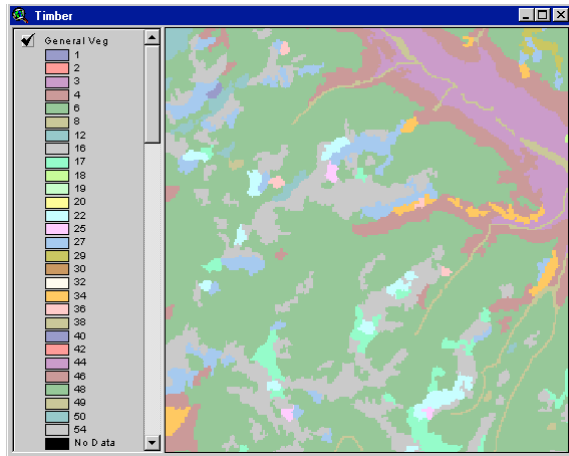
```
([Regions.Count] <= 20).SetNull (1.AsGrid)
```



Regions containing 20 or fewer cells will be set to No Data

Finally, Daniel uses the Nibble request with the newly created mask to create the final generalized vegetation theme. In the Map Calculator, he builds this expression:

```
[Veg2].Nibble ([Mask], False)
```



The cleaned vegetation theme. Note the much smoother appearance

Exercise

Edit grids

The objective of this exercise is to expose you to the ArcView Spatial Analyst registration and editing capabilities. In the scenario for this exercise, you are given a land cover grid derived from a satellite image. The grid has a problem because the land cover classes are represented by hundreds of disjointed regions, giving the data a speckled appearance. You will attempt to remove the speckles by merging the disjointed areas into homogeneous regions.

This type of editing can be time-consuming. It is an incremental and subjective process of trying a request on a dataset and deciding if the results are adequate. If you are not satisfied, you may delete the dataset and try another request or try the same request but with different options.

Many of the following steps use several Avenue requests. You will probably find it helpful to have the online help open as you issue these requests so you can learn how and why certain parameters are used.

If you have not downloaded the exercise data for this module, you should [download the data now](#).

Step 1 Start ArcView

Start ArcView and load the Spatial Analyst Extension.

Note: If you are running ArcView GIS 3.1, you see a Welcome to ArcView GIS dialog. Click Cancel to close this dialog.

If ArcView is already running, close any open projects.

Step 2 Open the project

From the File menu, choose Open Project. Navigate to the **dbases\lesson3** directory and open the project **I3_ex01.apr**.

Note: If you are running ArcView GIS 3.1, you see an Update I3_ex01.apr message box. Click No to dismiss this box.

When the project opens, you see a Redlands Land Cover Study view containing a Redlands image theme and a Land Cover grid theme. The Redlands image theme is turned on.

<..\graphics\t1exr1s2S2M1.cfm?CFID=847131&CFTOKEN=42944341>
<..\graphics\t1exr1s2S2M1.cfm?CFID=847131&CFTOKEN=42944341>

Step 3 Examine a grid before editing

Examine the satellite image and the Land Cover grid theme.

Turn on the Land Cover theme and make it active. Zoom in to its extent by clicking the Zoom to Active Themes button .

Visually compare the Land Cover theme with the image by turning the Land Cover theme off and on.

<..\graphics\t1exr1s3S3M1.cfm?CFID=847131&CFTOKEN=42944341>

<..\graphics\t1exr1s3S3M1.cfm?CFID=847131&CFTOKEN=42944341>

This grid was derived from the satellite image in an attempt to identify generalized land cover classes. The Land Cover grid has been clipped to the southeast quadrant of the Redlands study area.

Step 4 View a theme table before editing

With the Land Cover theme active, click the Open Theme Table button to open its theme table.

<..\graphics\t1exr1s4S4M1.cfm?CFID=847131&CFTOKEN=42944341>

<..\graphics\t1exr1s4S4M1.cfm?CFID=847131&CFTOKEN=42944341>

Examine the values and their corresponding land cover descriptions. In particular, notice Chaparral (1), Commercial / road (7), and Residential (6) because you will edit the noise and speckles to generalize these three areas.

From the project window, open the Chart of Land Cover. Examine this chart, which compares the cell counts for each land cover type.

Close the chart and the table.

Step 5 Generalize the data

After evaluating the display of the Land Cover theme, it is apparent that the seven classes are represented by hundreds of small regions. Some are as small as single cells intermixed together. The edits that you are about to perform will aggregate the dominant regions into larger, homogeneous areas.

In this step, you will group the Commercial / road areas (value of 7) with the Residential areas (value of 6) using the Con request.

Open the Map Calculator and enter a Con request on Land Cover where cells have a value equal to 7. If cells have a value of 7, convert them to 6; otherwise, output the same input value. Your expression should look like the one below:

([Land Cover] = 7).Con (6.AsGrid, [Land Cover])

When the new theme appears, close the Map Calculator. Change the name of the new theme to **Edit1** and turn it on. Turn off Land Cover and the Redlands image theme.

<..\graphics\t1exr1s5S5M1.cfm?CFID=847131&CFTOKEN=42944341>

<..\graphics\t1exr1s5S5M1.cfm?CFID=847131&CFTOKEN=42944341>

In the new grid, notice that the classes with values of 6 and 7 were merged into one class with a value of 6.

Step 6 Load a legend

Double-click on Edit1 to open the Legend Editor. Click the Load button.

In the file browser, navigate to the **dbasesa\lesson3** directory and select **I3_ex01.avl**. Click OK to load this legend. Click Apply in the Legend Editor, then close it.

<..\graphics\t1exr1s6S6M1.cfm?CFID=847131&CFTOKEN=42944341>

<..\graphics\t1exr1s6S6M1.cfm?CFID=847131&CFTOKEN=42944341>

At each step in the rest of this exercise, load this common legend file so that when you compare themes they will be displayed in the same colors.

Step 7 Use MajorityFilter

Next, you will eliminate some of the smaller speckles by changing their value to the majority value of cells in their neighborhood by using MajorityFilter.

In the Map Calculator, enter the MajorityFilter request on Edit1. Specify that eight diagonal neighbors (true uses all eight neighbors and false uses only the four orthogonal neighbors) should be considered using the majority rule (false for HalfOK means that more than half the cells must have the same value) to obtain a better smoothing. Your expression should look like the one below.

[Edit1].MajorityFilter (TRUE, FALSE)

When the new theme appears, close the Map Calculator. Load the common legend. Change the name of the new theme to **Edit2**, and turn it on. Turn off Edit1.

<..\graphics\t1exr1s7S7M1.cfm?CFID=847131&CFTOKEN=42944341>

<..\graphics\t1exr1s7S7M1.cfm?CFID=847131&CFTOKEN=42944341>

Many speckles are now gone and two dominant regions are starting to appear, one toward the top (values of 6) and the other toward the bottom (values of 1) of the grid.

<t1exr1s7RC1.cfm?CFID=847131&CFTOKEN=42944341>

<t1exr1s7RC1.cfm?CFID=847131&CFTOKEN=42944341>

Step 8 Use FocalStats

Compared to MajorityFilter, the FocalStats request has a stronger smoothing effect and an option to specify a desired neighborhood size and shape. You will use FocalStats next to smooth the data further.

In the Map Calculator, enter the FocalStats request on Edit2. Use the Majority enumeration and Make a Circle neighborhood that has a 2-cell radius. Your expression should look like the following:

[Edit2].FocalStats (#GRID_STATYPE_MAJORITY, NbrHood.MakeCircle(2, FALSE), FALSE)

When the new theme appears, close the Map Calculator. Load the common legend. Change the name of the new theme to **Edit3** and turn it on.

<..\graphics\t1exr1s8S8M1.cfm?CFID=847131&CFTOKEN=42944341>

<..\graphics\t1exr1s8S8M1.cfm?CFID=847131&CFTOKEN=42944341>

Turn off Edit2.

Notice the areas with No Data values that now appear. An unwanted side effect of FocalStats is that if there is no clear majority value in a neighborhood, then No Data is assigned. That is

why the Edit3 now has some black speckles (No Data values).

In the next step, you will convert all those No Data values to their old values from the Edit2 theme.

Step 9 Use Con and IsNull

In the Map Calculator, enter a nested expression using the Con and IsNull requests that sets the No Data cells in Edit3 back to their old Edit2 values. All other cells should remain the same as in Edit3. Your expression should look like the following:

[Edit3].IsNull.Con ([Edit2], [Edit3])

When the new theme appears, close the Map Calculator. Load the common legend. Change the name of the new theme to **Edit4** and turn it on.

../graphics/t1exr1s9S9M1.cfm?CFID=847131&CFTOKEN=42944341

../graphics/t1exr1s9S9M1.cfm?CFID=847131&CFTOKEN=42944341

Turn off Edit3.

Most of the speckles are now larger, multi-cell regions that are too large to be eliminated with MajorityFilter. In the next step, you will identify the smallest regions and dissolve them into the larger, neighboring regions.

Step 10 Use Nibble and RegionGroup

The Nibble request can be used to eliminate unwanted regions by replacing their values with those of other nearby cells. Before Nibble can be used, an edit mask must be created. The edit mask is a grid containing No Data values for the unwanted regions. You will create the edit mask by selecting regions that are composed of 10 or fewer cells and setting their values to No Data. The smaller regions will be nibbled up by neighboring larger regions.

Start by uniquely identifying each region and assigning unique ID values using RegionGroup. Then query for the smallest regions.

In the Map Calculator, use the RegionGroup request on the Edit4 theme to make regions. Your expression should look like this:

[Edit4].RegionGroup (TRUE, FALSE, NIL)

When the new theme appears, close the Map Calculator. Change the name of the new theme to **Region4Grid** and turn it on. Turn off Edit4.

Open the theme table for Region4Grid. Examine the Value, Count, and Link fields.

Pay attention to the Count field. Count is the number of cells that make up a region. Regions with 10 or fewer cells will be nibbled. Link is a relational key back to the Value field in the Edit4 theme table.

With Region4Grid active in the view, click the Query Builder button and search for regions that have a Count less than or equal to 10. Your expression should look like the following:

([Count] <= 10)

Click New Set.

Close the Query Builder and the Attributes of Region4Grid theme table. The cells that are highlighted belong to the smallest regions. These small regions will be used to create the No Data edit mask for the Nibble Request. The larger neighboring regions will nibble up the smaller regions.

In the Map Calculator, write an expression with the SetNull request to change regions with a Region4Grid.Count less than or equal (<=) to 10 and set all other values to 1. Your expression should look like the following:

([Region4Grid.Count] <= 10).SetNull (1.AsGrid)

When the new theme appears, close the Map Calculator. Change the name of the new theme to **Mask4Grid** and turn it on. Turn off Region4Grid and Edit4.

Notice all the No Data values that represent small regions made up of less than 10 cells. Next, you will use the mask in the Nibble request.

In the Map Calculator, use the Nibble request on Edit4 with the Mask4Grid as the edit mask.

Your expression should look like this:

[Edit4].Nibble ([Mask4Grid], FALSE)

When the new theme appears, close the Map Calculator. Load the common legend. Change the name of the new theme to **Edit5** and turn it on.

[../graphics/t1exr1s10S10M1.cfm?CFID=847131&CFTOKEN=42944341](http://t1exr1s10S10M1.cfm?CFID=847131&CFTOKEN=42944341)

[../graphics/t1exr1s10S10M1.cfm?CFID=847131&CFTOKEN=42944341](http://t1exr1s10S10M1.cfm?CFID=847131&CFTOKEN=42944341)

Notice that the smaller regions are now gone, their values replaced by neighboring cells.

Notice also there are two dominant zones, the Chaparral with a value of 1 and the Residential with a value of 6. In the next steps, you will work on expanding these two zones by further eliminating the number of smaller included regions.

t1exr1s10RC1.cfm?CFID=847131&CFTOKEN=42944341

t1exr1s10RC1.cfm?CFID=847131&CFTOKEN=42944341

Step 11 Use Expand

In the Map Calculator, use the Expand request to increase the size of Chaparral and Residential zones ({ 1, 6}) in the Edit5 theme by one cell. Your expression should look like the following:

[Edit5].Expand (1, { 1, 6})

When the new theme appears, close the Map Calculator. Load the common legend. Change the name of the new theme to **Edit6** and turn it on. Turn off Edit5.

[../graphics/t1exr1s11S11M1.cfm?CFID=847131&CFTOKEN=42944341](http://t1exr1s11S11M1.cfm?CFID=847131&CFTOKEN=42944341)

[../graphics/t1exr1s11S11M1.cfm?CFID=847131&CFTOKEN=42944341](http://t1exr1s11S11M1.cfm?CFID=847131&CFTOKEN=42944341)

Notice that the borders between Chaparral and Residential remained constant, but these two zones expanded out into other zones.

t1exr1s11RC1.cfm?CFID=847131&CFTOKEN=42944341

t1exr1s11RC1.cfm?CFID=847131&CFTOKEN=42944341

Step 12 Use Shrink

Shrink is the opposite of Expand. You will use Shrink to reduce the size of the other zones except Chaparral and Residential.

In the Map Calculator, use the Shrink request to reduce the size of zones { 2, 3, 4, 5}, in the Edit6 theme, by 1 cell. Your expression should look like the following:

[Edit6].Shrink(1, {2, 3, 4, 5})

When the new theme appears, close the Map Calculator. Load the common legend. Change the name of the new theme to **Edit7** and turn it on. Turn off Edit6.

[../graphics/t1exr1s12S12M1.cfm?CFID=847131&CFTOKEN=42944341](http://t1exr1s12S12M1.cfm?CFID=847131&CFTOKEN=42944341)

[../graphics/t1exr1s12S12M1.cfm?CFID=847131&CFTOKEN=42944341](http://t1exr1s12S12M1.cfm?CFID=847131&CFTOKEN=42944341)

t1exr1s12RC1.cfm?CFID=847131&CFTOKEN=42944341

t1exr1s12RC1.cfm?CFID=847131&CFTOKEN=42944341

Step 13 Use BoundaryClean

The BoundaryClean request iteratively applies Shrink and Expand to smooth edges between zones. It is possible to prioritize zones by value or area to enhance specific zones at the expense of others.

In the Map Calculator, use the BoundaryClean in the Edit7 theme with the NoSort enumeration. Make sure to enhance (false) the high-value, Residential zones at the expense of the low-value, Chaparral zones. Your expression should look like the following:

[Edit7].BoundaryClean (#GRID_SORTTYPE_NOSORT, FALSE)

When the new theme appears, close the Map Calculator. Load the common legend. Change the name of the new theme to **Edit8** and turn it on. Turn off Edit7.

../graphics/t1exr1s13S13M1.cfm?CFID=847131&CFTOKEN=42944341

../graphics/t1exr1s13S13M1.cfm?CFID=847131&CFTOKEN=42944341

t1exr1s13RC1.cfm?CFID=847131&CFTOKEN=42944341

t1exr1s13RC1.cfm?CFID=847131&CFTOKEN=42944341

Step 14 Use Nibble again

You will finish up by eliminating the remaining small regions and joining the attributes from the original Land Cover for a final clean land cover grid. You can then compare it with the original. You will use RegionGroup and Nibble one last time to eliminate all the small, left-over regions with cell counts less than 10.

In the Map Calculator, use the RegionGroup request on Edit8. Your expression should look like the one below.

[Edit8].RegionGroup (TRUE, FALSE, NIL)

When the new theme appears, close the Map Calculator. Change the name of the new theme to **Region8Grid** and turn it on. Turn off Edit8.

Look at the smaller regions.

With Region8Grid active, click the Query Builder button and search for regions that have a Count less than or equal to 10. Your expression should appear as follows:

([Count] <= 10)

Click New Set.

Make an edit mask using the small regions.

In the Map Calculator, write an expression with the SetNull request to change regions with a Region8Grid. Count less than or equal to 10 and set all other values to 1. Your expression should look like the following:

([Region8Grid.Count] <= 10).SetNull (1.AsGrid)

When the new theme appears, close the Map Calculator. Change the name of the new theme to **Mask8Grid** and turn it on. Turn off Region8Grid.

Notice that there are fewer small regions in this mask than the first one that you made earlier.

In the Map Calculator, enter the Nibble request on Edit8 with Mask8Grid as the edit mask.

Your expression should look like the following:

[Edit8].Nibble([Mask8Grid], False)

When the new theme appears, close the Map Calculator. Load the common legend. Change the name of the new theme to **Clean LandCover** and turn it on. Turn off Mask8Grid.

../graphics/t1exr1s14S14M1.cfm?CFID=847131&CFTOKEN=42944341

../graphics/t1exr1s14S14M1.cfm?CFID=847131&CFTOKEN=42944341

Notice that most of the smaller regions are now gone.

Finish by comparing Clean LandCover and the original Land Cover. Notice how the Clean LandCover has been generalized. The speckles, or noise, in the original Land Cover theme have been eliminated.

Step 15 Close the project

Close the project without saving any changes. You have completed this exercise

Exercise

Mosaic and merge grids

The objective of this exercise is to expose you to the ArcView Spatial Analyst tools for merging, or appending, grids together.

If you have not downloaded the exercise data for this module, you should [download the data now](#).

Step 1 Start ArcView

Start ArcView and load the Spatial Analyst Extension.

Note: If you are running ArcView GIS 3.1, you see a Welcome to ArcView GIS dialog. Click Cancel to close this dialog.

If ArcView is already running, close any open projects.

Step 2 Open the project

From the File menu, choose Open Project. Navigate to the **dbaseSA\lesson3** directory and open the project **I3_ex02.apr**.

Note: If you are running ArcView GIS 3.1, you see an Update I3_ex02.apr message box. Click No to dismiss this message.

When the project opens, you see a Western Mono Lake view containing four floating point grids of elevation. Some of the grids are overlapped and there is a slight gap between two of the grids.

[../graphics/t1exr2s2S2M1.cfm?CFID=847131&CFTOKEN=42944341](http://graphics/t1exr2s2S2M1.cfm?CFID=847131&CFTOKEN=42944341)

[../graphics/t1exr2s2S2M1.cfm?CFID=847131&CFTOKEN=42944341](http://graphics/t1exr2s2S2M1.cfm?CFID=847131&CFTOKEN=42944341)

Step 3 Use Mosaic

The Mosaic request can be used to merge a list of grids and smooth overlapping areas.

In the Map Calculator, use the Mosaic request on the Negit Island theme to merge it with the Mount Dana, Lundy Peak and Lee Vining grids. Your expression should appear as follows:

[Negit Island].Mosaic ({[Mount Dana],[Lundy Peak],[Lee Vining]})

Change the name of the new theme to **Mono Mosaic** and turn it on. Turn off all the other themes.

[../graphics/t1exr2s3S3M1.cfm?CFID=847131&CFTOKEN=42944341](http://graphics/t1exr2s3S3M1.cfm?CFID=847131&CFTOKEN=42944341)

[../graphics/t1exr2s3S3M1.cfm?CFID=847131&CFTOKEN=42944341](http://graphics/t1exr2s3S3M1.cfm?CFID=847131&CFTOKEN=42944341)

Notice that the four elevation themes have been merged together into one floating point grid theme. There is a slight horizontal gap assigned the No Data value in the middle of the grid.

You will repair this in the next step.

t1exr2s3RC1.cfm?CFID=847131&CFTOKEN=42944341

t1exr2s3RC1.cfm?CFID=847131&CFTOKEN=42944341

Step 4 Fix gaps in the floating point data

Gaps can be easily fixed using the FocalStats request using a 5 x 5 neighborhood. In the Map Calculator, use the FocalStats request on the Mono Mosaic theme for cells containing No Data values. Your expression should appear as follows:

[Mono Mosaic].IsNull.Con([Mono Mosaic].FocalStats (#GRID_STATYPE_MEAN, NbrHood.MakeRectangle (5, 5, false), false), [Mono Mosaic])

Change the name of the new theme to **Western Mono Lake** and turn it on. Turn off all the other themes.

[../graphics/t1exr2s4S4M1.cfm?CFID=847131&CFTOKEN=42944341](http://localhost:6080/graphics/t1exr2s4S4M1.cfm?CFID=847131&CFTOKEN=42944341)

[../graphics/t1exr2s4S4M1.cfm?CFID=847131&CFTOKEN=42944341](http://localhost:6080/graphics/t1exr2s4S4M1.cfm?CFID=847131&CFTOKEN=42944341)

Notice that the gaps have been removed.

Step 5 Visualize an elevation theme

A hillshade theme can be derived to better visualize the Western Mono Lake. Make the Western Mono Lake theme active, then from the Surface menu, choose Compute Hillshade. When the new theme appears, turn it on.

[../graphics/t1exr2s5S5M1.cfm?CFID=847131&CFTOKEN=42944341](http://localhost:6080/graphics/t1exr2s5S5M1.cfm?CFID=847131&CFTOKEN=42944341)

[../graphics/t1exr2s5S5M1.cfm?CFID=847131&CFTOKEN=42944341](http://localhost:6080/graphics/t1exr2s5S5M1.cfm?CFID=847131&CFTOKEN=42944341)

Close the Western Mono Lake view.

Step 6 Close the project

If you want to do the Challenge, go there now. Otherwise, close the project without saving any changes.

You have completed the lesson exercises

Summary

In this lesson you learned about how you can clean up heterogeneous or noisy data. ArcView Spatial Analyst includes several tools for this purpose:

- MajorityFilter--replaces cell values with the most frequent values in the neighborhood.
- Expand--enlarges a selected zone or list of zones.
- Shrink--reduces selected zones by a specified number of cells.
- Thin--makes raster lines one pixel wide.
- Nibble--replaces bad cells identified by a mask grid.
- BoundaryClean--cleans ragged edges between zones for integer grids.
- Merge--merges or appends a list of grids.
- Mosaic--merges a list of grids and smoothes the overlapping areas

Merge interpolates values to fill gaps between input grids.

Self test

3297	True
3298	False

2. Mosaic can merge only floating point grids.

3659	True
3660	False

3. Merge can be used only to append grids containing the same type of data, such as soils.

3299	True
3300	False

4. Floating point grids of different resolutions should be resampled using either the bilinear or cubic resampling methods before using Mosaic.

3663	True
3664	False

5. Ragged edges between zones can best be smoothed by using:

3285	Shrink
3284	Nibble
3283	BoundaryClean
3286	Thin

6. To reduce a selected zone by a specified number of cells, you could use:

3268	Nibble
3269	Merge
3267	Thin
3266	Shrink

7. The request that uses an expand-then-shrink-then-expand-then-shrink method to assist in editing is:

3293	Nibble
3292	BoundaryClean
3295	Merge
3294	Thin

8. Noise, speckles, bad scan lines, or bad regions in a grid are most appropriately eliminated with:

3281	Expand
3280	Nibble
3278	Shrink
3279	Thin

9. To smooth data by removing heterogeneous data, you could use:

3263	MinorityFilter
3264	Expand
3265	Merge
3262	MajorityFilter

10. To reduce line features in a grid, such as contour lines, down to the width of one pixel, you could use:

3275	Thin
3276	Nibble
3274	Shrink
3277	Merge