Spatial Extent: Image Aggregation

What it does:

- Resamples images from high to low resolution

- Combines multiple high-resolution pixels into one low-resolution output pixel for each pixel in the output image

- Uses pluggable aggregation methods to compute output pixel values

Spatial Extent: Image Aggregation

How it works:

• A user-specified template image describes the extent, resolution, and other geospatial characteristics of the output image

• A user-specified high-resolution input image supplies pixels to aggregate into the output image

• Computes the number of input image pixels comprising one output image pixel – the kernel size; ensures the kernel size is odd so it is always centered on an output pixel

• Computes the value of each output image pixel by retrieving the kernel of pixels from the input image and applying the user-specified aggregation method

• Available aggregation methods: mean, minimum, maximum

Spatial Extent: Image Aggregation

To run:

• *aggregate.py* is a Python script

• GDAL and its Python bindings must be installed; if you have can run gdal_info, you should have everything aggregate.py needs

• Command summary:

aggregate.py *input_image template_image output_image* [options]

• Command options:

• -h	print usage statement
• -l	list all available aggregation methods
• -m *aggregation_method*	specify the aggregation method
• -v	print diagnostic messages

• The *aggregation_method* provided to the -m option must be one of the aggregation method names listed by the -l option; in other words, use -l to see the list of methods, then specify one from the list with -m

Spatial Extent: Image Aggregation

To write an aggregation method:

1. Copy an existing aggregation method file (e.g. AggMethodMin.py) to a new file (e.g. AggMethodStdDev.py)

2. Change the class name to match the file name, without the ".py" of course "class AggMethodMin" becomes "class AggMethodStdDev"

3. Modify the getName() method to return the name of the aggregation method "return StdDev"

4. Implement the aggregation algorithm in the body of the "aggregateKernel()" method
   - aggregateKernel() is passed a list of pixel values that are the kernel for the current output pixel
   - Return a single value that is the result of the aggregation method applied to the kernel

The aggregate script and AggMethod base class manage everything else

Spatial Extent: Image Aggregation

To add an aggregation method:

• Put it the directory with aggregate.py; that's it

Behind the scenes:

• aggregate.py looks for other ".py" files in the directory in which it resides

• aggregate.py *imports* those files and instantiates the classes within

• When instantiated, each aggregation method class adds itself to a master list of aggregation methods; this happens in the base class, AggMethod, so no extra coding is required