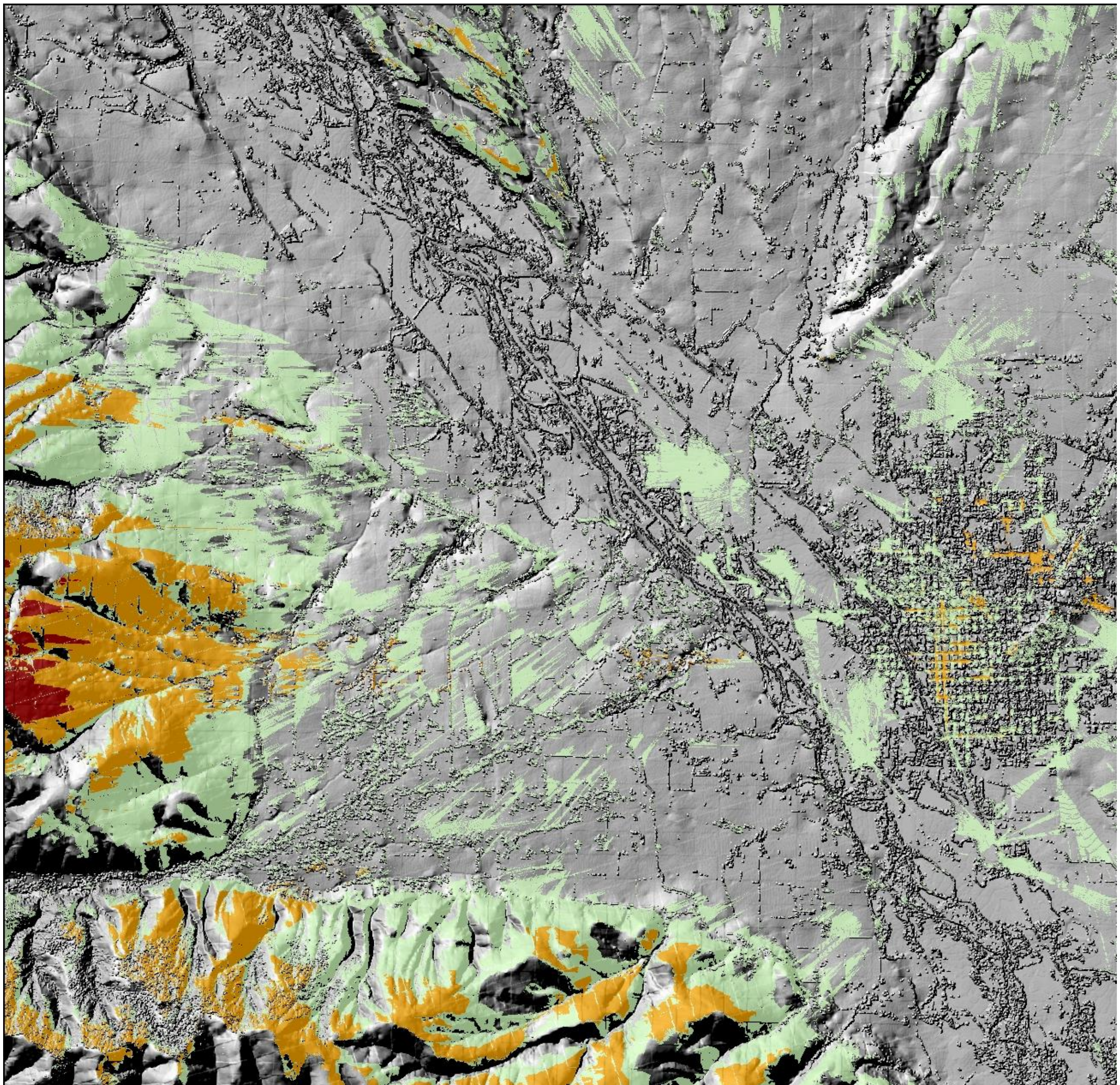
Regional Comprehensive Visual Sensitivity Assessment for Renewable Energy Facility Development



Introduction

Renewable Energy and Scenic Amenities

As interest in developing sustainable methods of energy production increases in the United States, rural regions with the climate and land base to accommodate large-scale renewable energy projects, such as wind farms and solar arrays are in a position to benefit economically. At the same time, research indicates that scenic and recreational amenities contribute significantly to economic growth and quality of life in many rural regions of the United States (Deller, Tsai, Marcouiller, & English, 2001; Jackson & Kuhlken, 2005; McGranahan, 1999; Smith & Krannich, 2000; Stewart, 2001). The scenic impacts of proposed renewable energy projects are a major reason many rural community members reject them (Wolsink, 2007). For rural jurisdictions to increase public support for these projects by mitigating scenic impacts, a comprehensive understanding of the visibility characteristics of their regional terrain is necessary.

To aid rural jurisdictions in mitigating visual impacts of large-scale renewable energy development, RGIS-PNW developed a methodology for creating a visual sensitivity model for a regional landscape based on cumulative viewshed analysis. This report describes the processes required to create the model.

Viewshed Analysis

A viewshed is a model of the visibility of terrain from an observer point. It is calculated from a raster representing elevation. To create a viewshed, observer points in the terrain are selected and lines of sight are drawn from each viewpoint to every cell. If there is no terrain between the two points, the target cell is classified as visible (Bolstad, 2002). Every cell in the raster is evaluated and assigned a Boolean value of visible or not visible.

Cumulative Visibility

Cumulative visibility analysis identifies the frequency with which locations in the landscape can be seen from other locations (Wheatley, 1995). A cumulative visibility surface is created by overlaying multiple viewsheds that have been calculated over the same terrain and summing the number of intersections across that terrain. The result is a model of cardinality, which identifies the number of times locations in the terrain are visible from a set of sampled observer points (DeFloriani & Magill, 2003). The result is a cumulative visibility surface revealing areas of low and high visibility to the observer points within that terrain.

Visual Sensitivity

To determine the visual sensitivity of the terrain to the population of a region, viewshed that are weighted based on population density at the observer point are used in a cumulative visibility analysis. By summing the weights of overlapping viewshed, rather than only summing the number of times a location can be seen from the observer points, the weighted visual sensitivity model identifies areas of low and high visibility to the population. In the final model, visually sensitive areas are identified by their higher weighted sums which indicate that these areas are visible to a larger portion of the regional population. The visual sensitivity model is a quantitative and objective measure of the visibility of the landscape to its residents.

To create an accurate visual sensitivity model, it is important to have a digital elevation model (DEM) that includes buildings and trees. These are difficult for rural jurisdictions to obtain. United States Geological Survey (USGS) DEMs do not include structures or vegetation. Lidar data can be used to create a top surface DEM that includes these features, but it is expensive to collect and therefore not widely available for rural jurisdictions. RGIS-PNW developed a process for creating a top surface DEM, which incorporates heights of buildings and trees. Esri ArcGIS and Visual Learning Systems' Feature Analyst are used to extract the features from 1m resolution National Agricultural Imagery Program (NAIP) orthorectified color aerial photographs (Miller, Nelson, & Hess, 2009) and then the average heights of these features are added to USGS 10m DEMs.

Feature Analyst automates feature extraction from raster data. The process begins by users digitizing training sets that represent the target features in raster imagery. Feature Analyst learns to identify target features in the raster by using the training sets to identify similar patterns of spectral signature, spatial association, size, shape, texture, pattern, and shadow. Additionally, multiple rasters representing spectral, elevation, and discrete data can be combined in the analysis to provide more information to the learning algorithm and aid the feature extraction process (VLS, 2005). After running an initial classification, the user digitizes errors of omission and commission providing the learning algorithm with new information, and again runs the extraction process. This iterative process continues until the user is satisfied with the results. The final extraction algorithm can be saved and run on multiple images.

System Requirements

Hardware: Calculating viewsheds is a processor intensive operation. A fast processor is advantageous for reducing the time needed to calculate the large number of viewshed required by the visibility model. A system that has a 3.0 Ghz or better processor and a minimum of 3GB of RAM is suggested. Large volumes of disk space are required for the model outputs. Free disk space of 500 GB or more is suggested to limit the need for file management during viewshed processing.

Operating System: Microsoft Window XP

Software: Esri ArcGIS 9.x or 10 with Spatial Analyst Extension, Visual Learning Systems Feature Analyst, and SpatialEcology.Com Hawth's Analysis Tools for ArcGIS

Data Requirements

National Agriculture Imagery Program (NAIP) 1m resolution color digital ortho quarter quads(DOQQs) or compressed county mosaics (CCM), 10m USGS Digital Elevation Model (DEM), Block Level Statistics and Boundaries, Block Group Census Statistics and Boundaries, city boundaries, roads, trails, and water bodies

Intended Audience

GIS Analysts with an intermediate level of GIS proficiency; experience using Spatial Analyst and Map Algebra; and a basic understanding of Feature Analyst, which should be acquired by reading the Feature Analyst Reference Manual prior to undertaking this methodology.

Methodology

Overview

The purpose of the visual sensitivity model is to identify the inherent visibility of a regional terrain to the resident population. The model is created by overlaying and summing multiple weighted viewsheds, which are generated from a large sample of observer points located on publicly accessible paths in the region of interest. Paths were chosen as the sampling frame because the population is most likely to be found on publicly accessible paths when traversing the landscape. Viewsheds are weighted to reflect the number of people residing in the area of the sampled observer point. The weight provides a relative indication of the number of people that can potentially view the landscape from that location. When the viewsheds are summed, the visibility of the entire terrain to the population is revealed. See figure 1 for a schematic of the methodology.

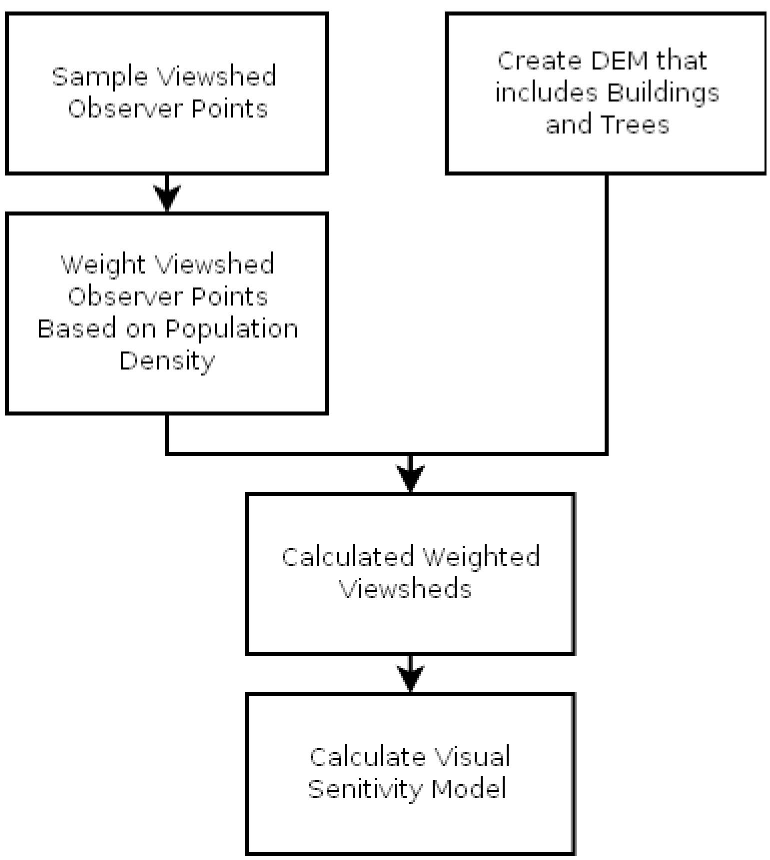


Figure 1. Schematic of Visual Sensitivity Model Workflow

To increase the accuracy of the viewsheds generated for the visibility model, the methodology requires the creation of a DEM that includes buildings and trees. ArcGIS and Feature Analyst are used to extract trees and buildings from 1m resolution NAIP color ortho aerial photographs. These features are assigned average heights and added to a 10m USGS DEM using the ArcGIS Spatial Analyst extension and Map Algebra. The final DEM is used to calculate the viewsheds.

This methodology requires a large amount of time to process the data. For example, extracting buildings and trees from a 6,000 km^2 area and processing the results for incorporation into a DEM can easily require 60 hours. For the same area, calculating 15,000 viewsheds, applying their weights, and summing the rasters can take up to 600 hours of CPU time. These times are based on computing with an Intel Core 2 Duo 2.66 Ghz processer and 3 GB of RAM. Although processing the viewsheds consumes large amounts of CPU time, much of the processing can be run in large batches. Development of a more efficient weighted viewshed tool that combines the tasks of calculating, weighting, and summing the viewsheds could reduce the amount of file management necessary to create the visual sensitivity model.

Weighted Visual Sensitivity Model Workflow

Sampling Viewshed Observer Points

Objective: Generate a sample of observer points, which will be used to create the viewsheds that will be used as input to visual sensitivity model.

Spatial Data: polyline feature classes of roads and trails covering the extent of the area of interest

Output: Point feature class representing a systematic sample of points on publicly accessible paths.

A systematic sample of observer points must be generated on all paths and nodes in the area of interest. Use Hawth's Analysis Tools for ArcGIS to generate the sample of observer points. Hawth's Tools contains a Convert Paths to Points Tool, which will generate points at regular intervals along polylines. Points are placed at the beginning of each polyline and then at each specified interval. Placing points at the beginning of each polyline will result in most nodes at intersections being sampled because in most cases paths networks are digitized using topological rules that require line segments to begin and end at intersections. Unfortunately, errors in interval spacing do occur. Points placed near the end of a segment may be too close to a point at the beginning of the next segment. In this case, an equal sampling interval over the entire network is not maintained (see Figure 2). Additionally, when two beginning points overlap, which is the case for all intersections, more than one sample point is located at the same spatial location. This results in duplicate sample points in many locations.

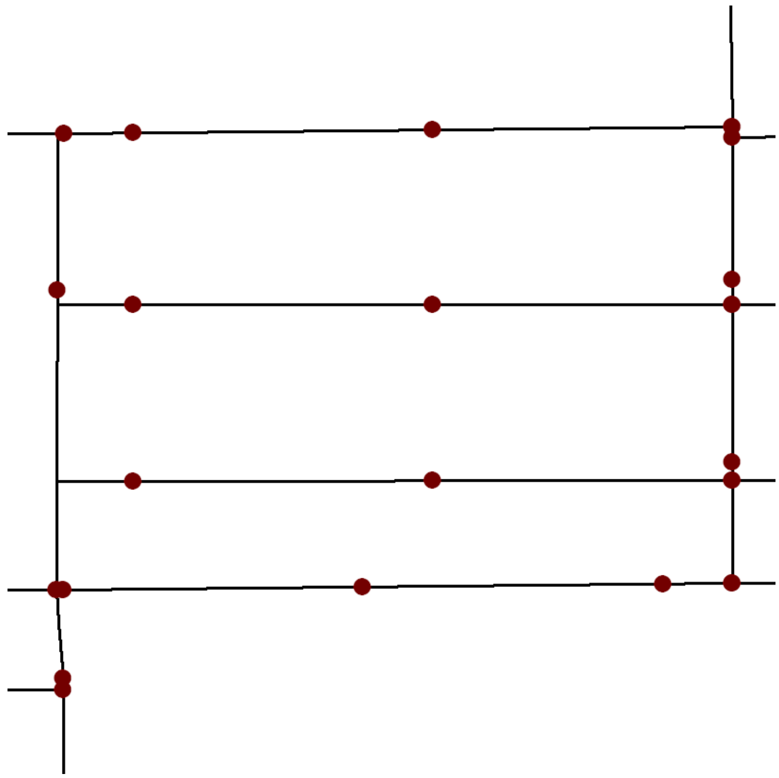


Figure 2. Output of the Convert Paths to Points Tool

To limit the number of sample points that are generated too close together, choose an appropriate distance interval between sampled points. The sampling distance should be slightly larger than the average block size in urbanized areas. For example, figure 3 shows an urban grid with an average block size of 100m, an appropriate sampling distance for the entire area of interest is 120m. This will reduce the number of points that are placed too close to intersections in urbanized areas (see Figure 3). Using average block size as the sampling interval, rather than a significantly larger value, helps to ensure that both urban and rural networks are sampled at the same interval. If a larger interval is chosen, urban road networks will be oversampled because of their greater number of intersection. To ensure a consistent sampling interval, points that are too close to each other and duplicate points must be removed from the data set.

Duplicate points can be removed using the Find and Delete Duplicate Records in a Shapefile Tool. This python script is available from the Esri support website or the Center for Spatial Information website (http://www.cwu.edu/~csi). It may be necessary to change the path in the following section of the python script to match your system configuration:

```
# Load required toolboxes...
gp.AddToolbox("C:/Program Files/Desktop10.0/ArcToolbox/Toolboxes/Data Management Tools.tbx")
```

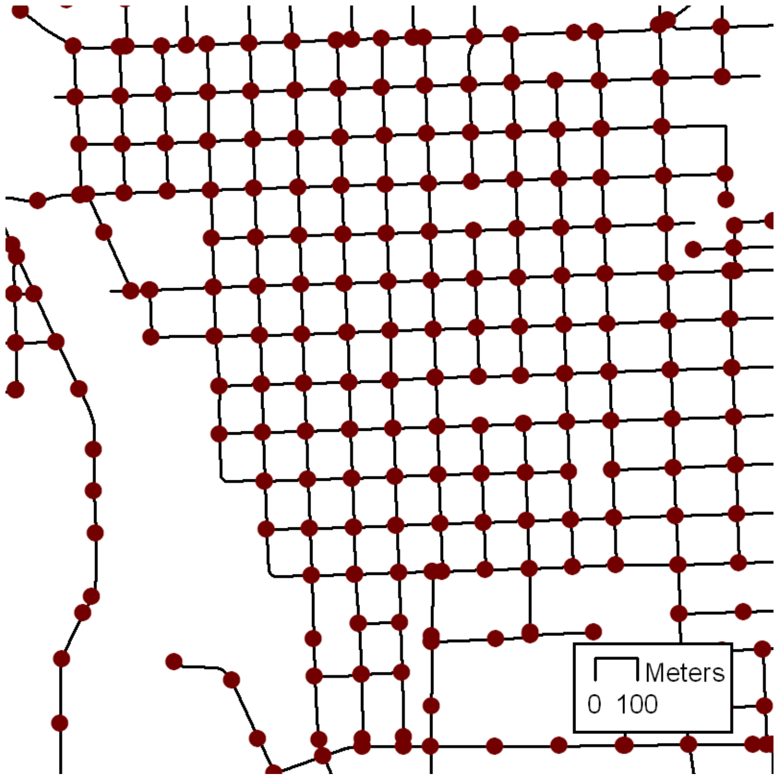


Figure 3. An urban road network with observer points sampled at 120m intervals.

The process of identifying and deleting points with proximities that are less than the sampling interval involves creating buffers around all points; identifing where they overlap; and deleting one of the points inside the overlapping buffers. First create a field containing a unique ID for each point in the observer points feature class, this unique ID will be used throughout the visual sensitivity modeling process. Use the Buffer Tool, in ArcToolbox to create a new buffer polygon feature class based on the observer points feature class. Make sure the unique ID of the points is retained in the buffer. The distance parameter for the Buffer Tool should be one third of the interval distance. This will identify points that are closer than the interval distance without identifying a large number of points that are only marginally closer than the interval distance. Save the output of the Buffer Tool to a feature dataset in a Geodatabase. Create a topology in the feature dataset and add the buffer polygon feature class and a Must Not Overlap rule. Validate the topology and add it to the ArcMap project. Topological errors where buffers overlap will be highlighted in red (see Figure 4).

The topological errors indicate locations where points are closer than the sampling interval and must be removed. To aid the editing process, create a relationship class between the points and buffers. When a relationship class exists, deleting a point will also delete the related buffer. This makes it clear that the error has been corrected. The relationship must be a composite relationship class based on the unique ID field that is present in both datasets. Visually inspect the polygon buffer feature class for topological errors. Where errors occur, delete one of the underlying points. Do not delete points that are at intersections. In cases where all points are at an intersection, use best judgment. Figure 5 shows the road network in Figure 4 after being cleaned of points that are too close together. Clean the entire observer point feature class.

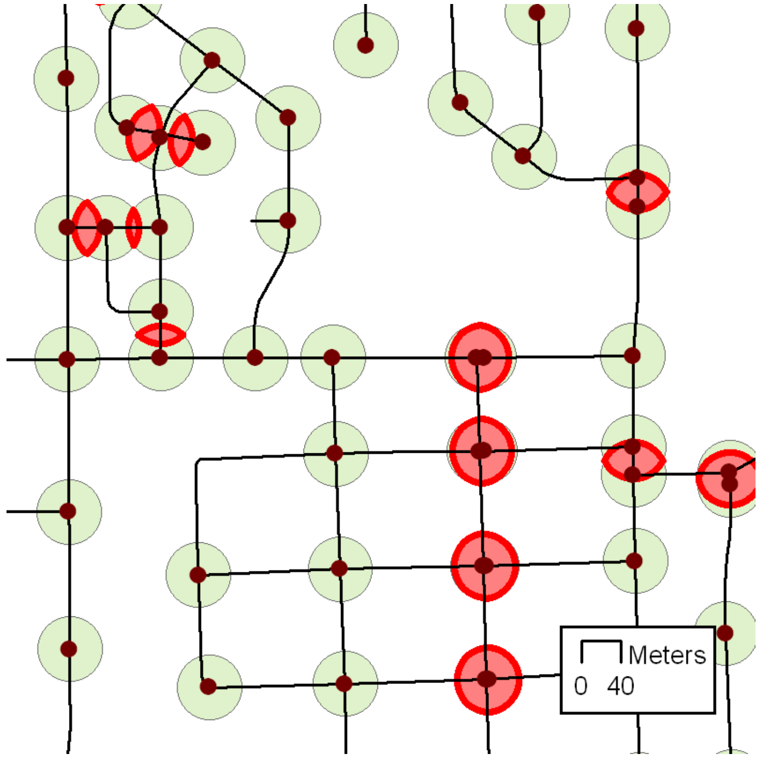


Figure 4. Overlap errors in buffer polygons indicating sampling interval errors.

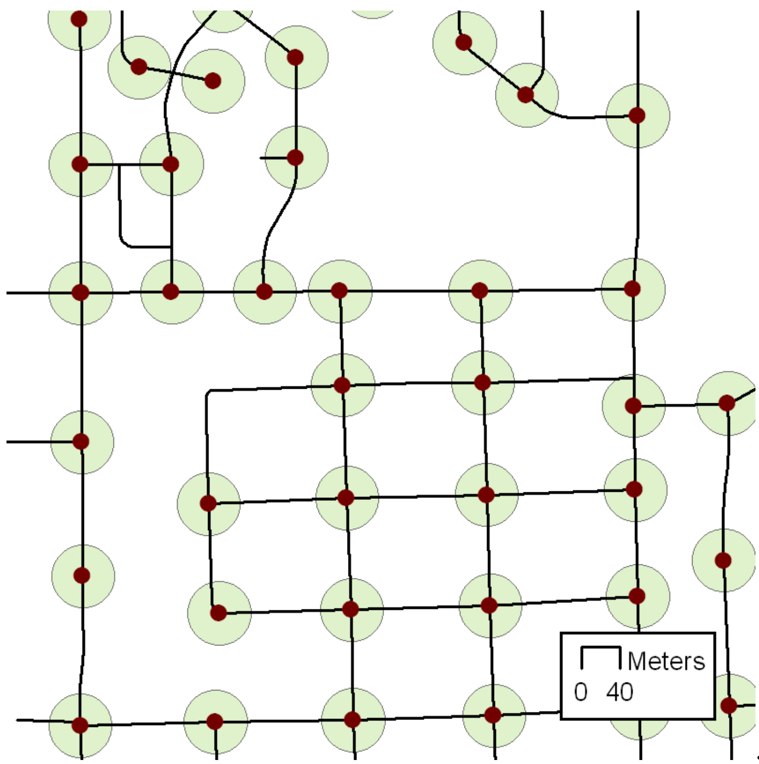


Figure 5. Sample observer point dataset cleaned of errors.

Weighting Viewshed Observer Points Based on Population Density

Objective: Assign a weight to each sampled observer point. The weight is based on population density at that location and will be applied to the viewshed that is generated from the point.

Spatial Data: Census Block boundaries and statistics, Census Block Group boundaries and statistics or populations estimates based on Census Block Groups, incorporated cities boundaries, and sampled observer points

Output: weighted sample observer points

In this step, Census Block Group data is used to calculate population density. Census Block Group was chosen as the unit of measure for population density because state agencies often use Block Group statistics for population estimates. This becomes more important toward the end of each decade, as the Census data becomes out dated. To calculate density, first, eliminate the areas in each Census Block Group that have zero population, this will result in a more accurate population density calculation. In many rural areas Census Blocks Groups cover large expanses of unpopulated land. Remove these areas from the Census Block Group feature class using the Census Block feature class. Select all Census Blocks that have zero population and export them to a new feature class. Use the feature class of Census Blocks with zero population blocks as input into the Erase Tool and erase those areas from the Block Group feature class. Populate a new field in the Block Group feature class with the area of the remaining block group polygons by using the calculate geometry function. Now calculate the population density in another new field by dividing the 'population' field by the 'area' field.

Now, the population density values must be transferred to the sample observer points. Some difficulties arise at this stage. Many census block group boundaries coincide with roads. This presents a problem because we are using roads as part of our sampling frame, so many observer points fall within two or more block group polygons. In these situations, the mean density of all Block Groups adjacent to the observer point is assigned. To accomplish this, use the buffer tool to create a new polygon feature class of 10m buffers around each observer point (see Figure 6). Intersect the buffer feature class with the Block Group feature class. The result will be a feature class of circular buffers divided into multiple polygons with different Block Group density values (see Figure 7). It is important to keep the buffers small to avoid inadvertently capturing Block Groups in the intersection that are not adjacent to the observer point.

Next, calculate the mean density for each observer point and convert it to a weight. To calculate the mean density for each buffer, use the Dissolve Tool to dissolve the buffers on the unique ID and make a selection to compute the mean statistic for the 'population density' field. This will result in a new feature class of buffer polygons for each observation point with an attribute table containing the unique ID and a mean population density for each buffer (see Figure 7). Use a join based on the unique ID to join the mean density values to the observer points. Be sure to export the joined data to a permanent feature class. To calculate the observer point weight in the new

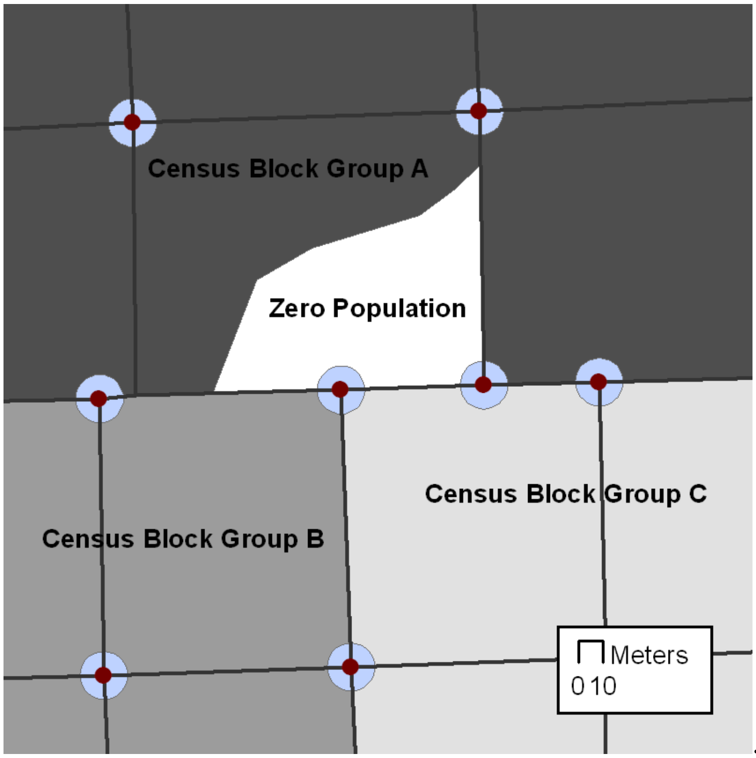


Figure 6. Observer point buffers overlaid on Census Block Group feature class.

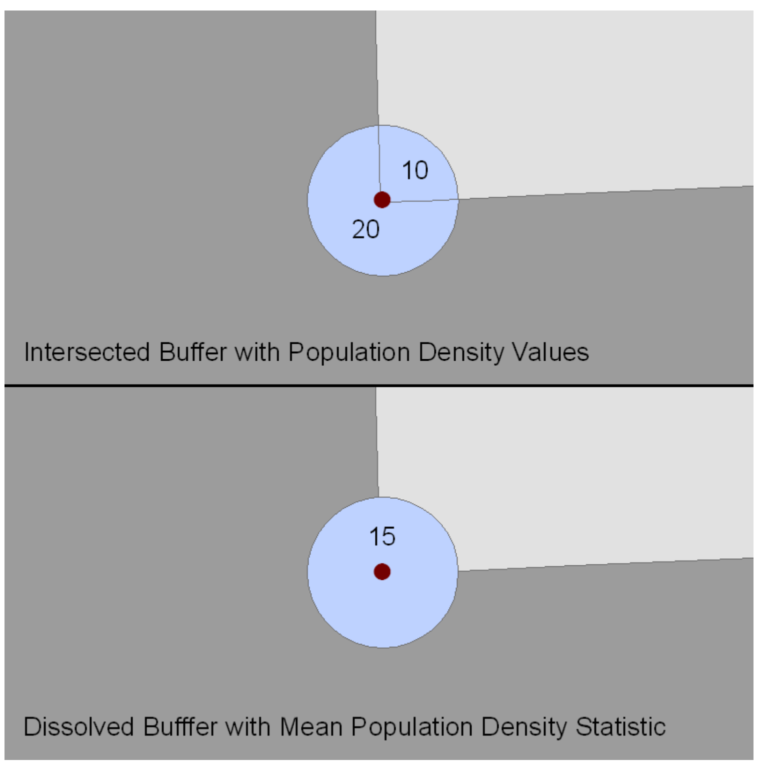


Figure 7. Examples of observer point buffers that have population density values.

feature class, create a field named ‘weight’ with a data type of double. Populate the weight field by using the field calculator to dividing the mean density value by the sum of the mean density field and multiply by 100. This creates a weight that equals the mean density values normalized to a 100 point scale. The sum of all records in the weight field should now be 100. The weight value will be used to weight the viewsheds that are created from the observer points.

Creating DEM that Includes Buildings and Trees

Objective: Create a DEM that includes buildings and trees.

Spatial Data: USGS DEM, 1 m resolution NIAP color ortho photographs, roads, water bodies

Output: A DEM that includes buildings and trees.

To increase the accuracy of viewshed calculations that are used in the visual sensitivity model, a DEM that includes buildings and trees is used. The approach taken in this methodology is to extract buildings and trees from 1 m resolution NAIP color ortho photographs using Feature Analyst and add their average heights to the DEM.

Preparing Imagery for Processing

First, the NIAP color ortho raster dataset must be prepared for processing. Feature Analyst uncompresses compressed rasters before operating on them. NAIP ortho photographs are often distributed in the compressed MrSID format. This adds to the processing time and requires additional disk space. To save time and disk space, convert the NAIP imagery to an uncompressed format (e.g., .img, .tif) before processing the data. This will generally result in a very large file. For example, a 2.5 GB MrSID file will uncompress to a 36 GB file. During processing, Feature Analyst generates temporary rasters that require twice the disk space of the original uncompressed image, so it is necessary to have free disk space on the drive where Feature Analyst is installed that is a least two times the size of the file being processed. Clipping large rasters into smaller tiles is recommended to avoid running out of disk space in the middle of processing or losing large amounts of data in the event of a fatal error or power outage during processing. ArcGIS 10 has a Split Raster Tool that can be used to tile large rasters. Another option is the Geospatial Data Abstraction Library (GDAL) and its gdal_retile.py command line utility. GDAL is an open source translator library for raster geospatial data formats that runs on Windows and Linux operating systems. It is available as part of FWTools package from MapTools.org. RGIS-PNW had good results running GDAL on the Ubuntu version of the Linux operating system.

Extracting Buildings and Tree Features from Ortho Photographs

Use Feature Analyst with ArcGIS to extract buildings and trees from 1m resolution NAIP color ortho photographs. First, it is necessary to develop training sets for buildings and trees prior to running the learning algorithm. This is accomplished by digitizing examples of these features found in the imagery. The training classes are used for Supervised Learning. The Supervised Learning process creates the classification algorithms. The algorithms are developed through an iterative training process that is called Hierarchical Learning. After the algorithm is run, feedback regarding correctly and incorrectly classified features is input into the algorithm and it is run again. This process can be repeated multiple times until the classification is acceptable. The algorithms are called Feature Models and they are used to extract the features of interest from all of the images. Feature Analyst allows for single-class feature extraction and multi-class feature extraction. In most implementations of the visual sensitivity methodology, feature extraction will take place over large regions with complex patterns of land cover. In these situations, target features of buildings and trees are extracted more accurately with a single-class approach. When developing workflows for extracting buildings and trees, RGIS-PNW was guided by Miller, Nelson, & Hess's (2009) work using Feature Analyst to extract impervious surfaces from high resolution color imagery.

Extracting building footprints with Feature Analyst using only color aerial photographs is challenging. In urban areas in particular, where there is a large diversity of building shapes and roof colors; driveways, sidewalks, roads, and parking lots that are adjacent to buildings with similar color roofs; and tree canopies covering portions of roofs provide great challenges. Most building classifications will result in a large number of errors of commission, especially errors where roads and sidewalks are classified as buildings. Attempts to refine the learning often result in classifications with unacceptable levels of errors of omission. This also occurs with reflections on water, but to a lesser degree. The workflow described as part of this methodology accepts initial errors that classify roads, sidewalks, and water as buildings; and mask out those errors at the end of the process using the roads and hydrology feature classes.

A workflow for extracting buildings from 1m NAIP color ortho photographs includes:

- Selecting three tiled images to be used for developing a training set. The images should include a sample of building types and environmental settings that is representative of the entire area of interest. For example, choose a mix of urban and rural development patterns in farmland and natural areas that contain residential and commercial buildings.
- Select one of the three images that have the greatest diversity of building types and roof colors, and the greatest diversity of building settings. Use Feature Analyst's drawing tools to digitize a training set from this image. The training set should include a representative sample of building types and colors in different settings.
- Run an initial single-feature supervised learning pass. Choose the preset feature selector for 'manmade features (>5m)' leaving all the default settings. Set the input bands to red, green, and blue spectral bands. Add the three bands of the raster a second time and set them to be texture bands. This seemed to marginally improve the accuracy of the classification. Although, it would be ideal have the output of the training be a raster dataset. There is no way to train the algorithm when the output is set to raster and no way to change the option after the Feature Model is completed and saved.
- Visually examine the resulting classification for errors of omission and edit the training set to include feature types that were not classified. At this stage it is better to edit the training set rather than identify errors of commission using Hierarchical Learning because more iterations of learning slow the processing time of the final model.
- Rerun the single-feature supervised learning pass. If the errors of omission are not still not acceptable edit the training set and run another learning pass.

- If the errors of omission are acceptable. Use the Hierarchical Learning tool to begin removing clutter by identifying errors of commission. It is useful to save the learning model after each learning pass. It may be necessary to return to it later if future learning passes do not improve the extraction algorithm.
- Repeat the Hierarchical Learning step until the results are satisfactory. Try to reduce the amount of errors of commission, while avoiding errors of omission. Allowing some streets and sidewalks to be identified as buildings is acceptable because they will be masked out later in the process.
- When the errors in the extraction process are acceptable, save the results as a Feature Model.
- Use Feature Modeler to run the model on another one of the three images selected at the beginning of the process.
- If the results are not satisfactory, use the Hierarchical Learning tool to begin removing clutter by identifying errors of commission and at this stage also identify errors of omission.
- When the results of this extraction are satisfactory save the Feature Model and run it on the final image tile that was selected.
- Use the Hierarchical Learning tool until the results are satisfactory, and save the Feature Model again
- Run the final version of the Feature Model on all of the image tiles that cover the area of interest. No color balancing is necessary if the images were originally part of a mosaic.
- The output will be a vector feature class of buildings for each input raster. Convert all the vector building polygon tiles to raster format. Reclassify the raster making areas of buildings equal to 1 and areas without buildings equal to 0.
- Mosaic all of the building raster tiles to a single raster dataset covering the entire area of interest.
- Use to NAIP imagery and field work to identify regions that have similar building heights. Digitize these regions in a new polygon feature class and assign each polygon an average building height. Be sure to use the same unit of measure as the elevation values in the USGS DEM. Convert the building height feature class to a raster dataset using the building heights values as the raster values.
- Use Map Algebra to multiply the building heights raster by the buildings raster. The result will be a raster that represents buildings with average heights assigned to each building.
- Create a roads mask raster dataset by buffering the line segments in the roads feature class. The buffer should equal the width of the average road right of way; including sidewalks.
- Create a water mask raster dataset by converting the water bodies feature class to a raster dataset.
- Use Map Algebra to combine the roads mask and water mask raster datasets into one mask raster dataset.
- Reclassify the mask raster dataset making the masked areas equal to 1 and the unmasked areas equal to 0.
- To eliminate errors in the building heights raster, where roads, sidewalks and water bodies were classified as buildings, use the mask raster dataset with a conditional statement in Map Algebra. Make the output raster equal 0 where the mask equals 1 and

have it remain the same value where the mask equals 0 (Example expression: raster = con(mask = = 1, 0, building heights raster)).
- The output is a cleaned building height raster (see Figure 8).
- Use Map Algebra to add the revised building heights raster to the USGS DEM.
- The output is a DEM that includes average building heights (see Figure 9).

Extracting trees from 1m NAIP color ortho photographs requires less data manipulation to achieve good accuracy, when compared to extracting buildings, especially in arid and semi-arid regions. There are several characteristics of trees that aid the object extraction process. During late spring and summer, most trees have fairly uniform color, texture, and shape. Trees cast shadows, which can be identified by the learning algorithm. There are generally more errors of commission than omission. Errors of commission occur most frequently where there are steep rock out croppings that cast long shadows or on the surface of eutrophic lakes that are ringed by trees. Errors of omission occur more frequently in areas of dispersed small conifers.

A workflow for extracting trees from 1m NAIP color ortho photographs includes:

- Selecting three tiled images to be used for developing a training set. The images should include a sample of tree types and environmental settings that is representative of the entire area of interest. For example, choose tiles with trees in riparian areas, dry grassland, agricultural areas, mountains, and urban areas.
- Select one of the three images with the greatest diversity of tree types and environmental settings. Use Feature Analyst drawing tools to digitize a training set from this image. The training set should include a representative sample of tree types in different environmental settings.
- Run an initial single-feature supervised learning pass. Choose the preset feature selector for ‘natural feature’ leaving all the default settings and set the input bands to red, green, and blue spectral bands. Add the three bands of the raster a second time and set them to be texture bands.
- Repeat the same iterative Hierarchical Learning procedures conducted in the building classification to extract tree features from all tiles covering the area of interest.
- Convert all of the tree polygon tiles to raster format.
- Mosaic all of the tree raster tiles to a single raster dataset covering the entire area of interest. Reclassify the tree raster dataset. Reclassify the raster; making areas of trees equal to 1 and areas without trees equal to 0.
- Conduct field work to identify average trees heights and use Map Algebra to multiply the tree raster by the average height of trees. Be sure to use the same unit of measure as the elevation values in the USGS DEM. The result will be a raster with values representing tree heights (see Figure 10).
- Create a DEM raster dataset that includes trees by using Map Algebra to add the values of the tree height raster dataset to the USGS DEM.
- The result is a tree height DEM (see Figure 11).

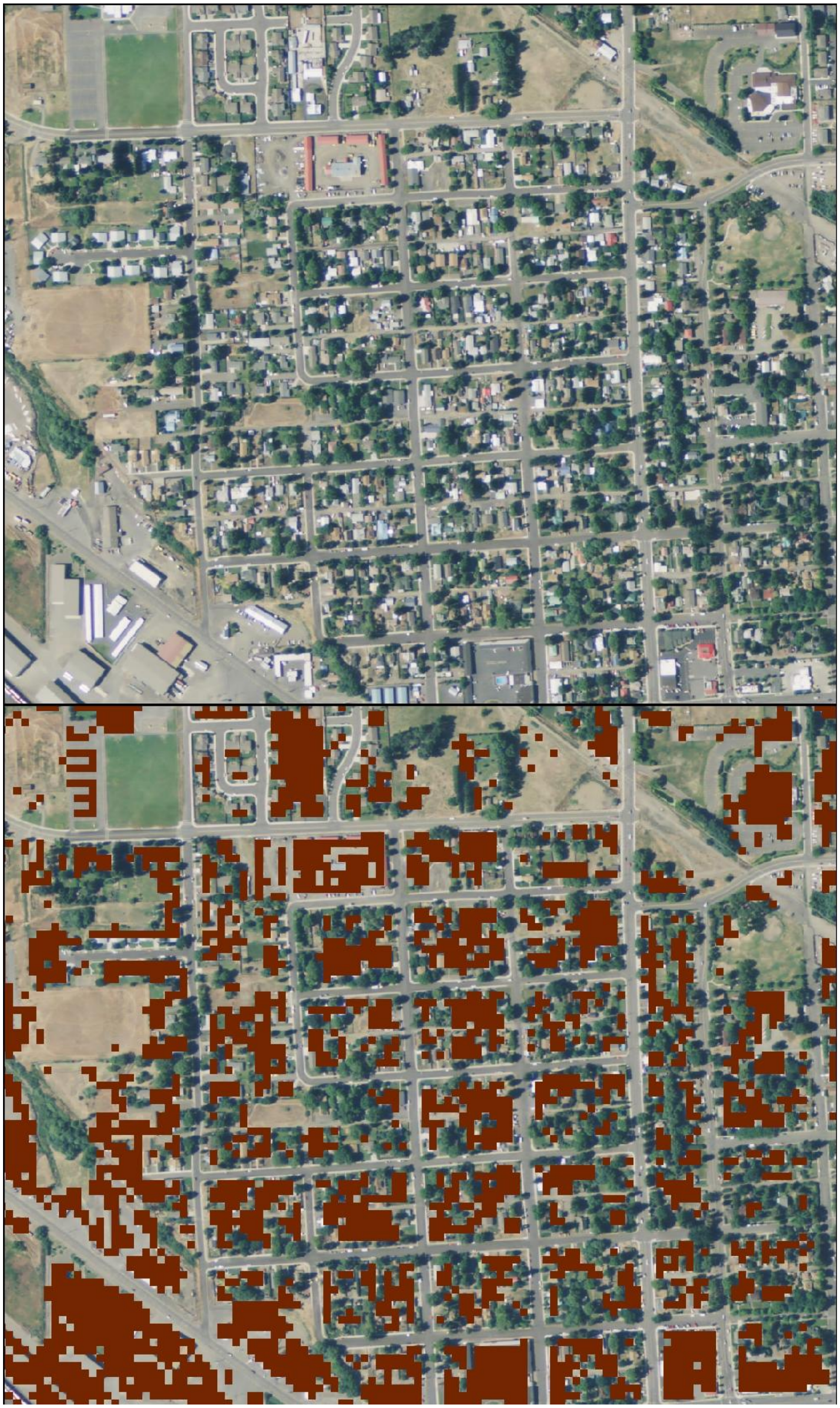


Figure 8. Example of buildings extracted from 1m resolution NAIP color ortho photographs.

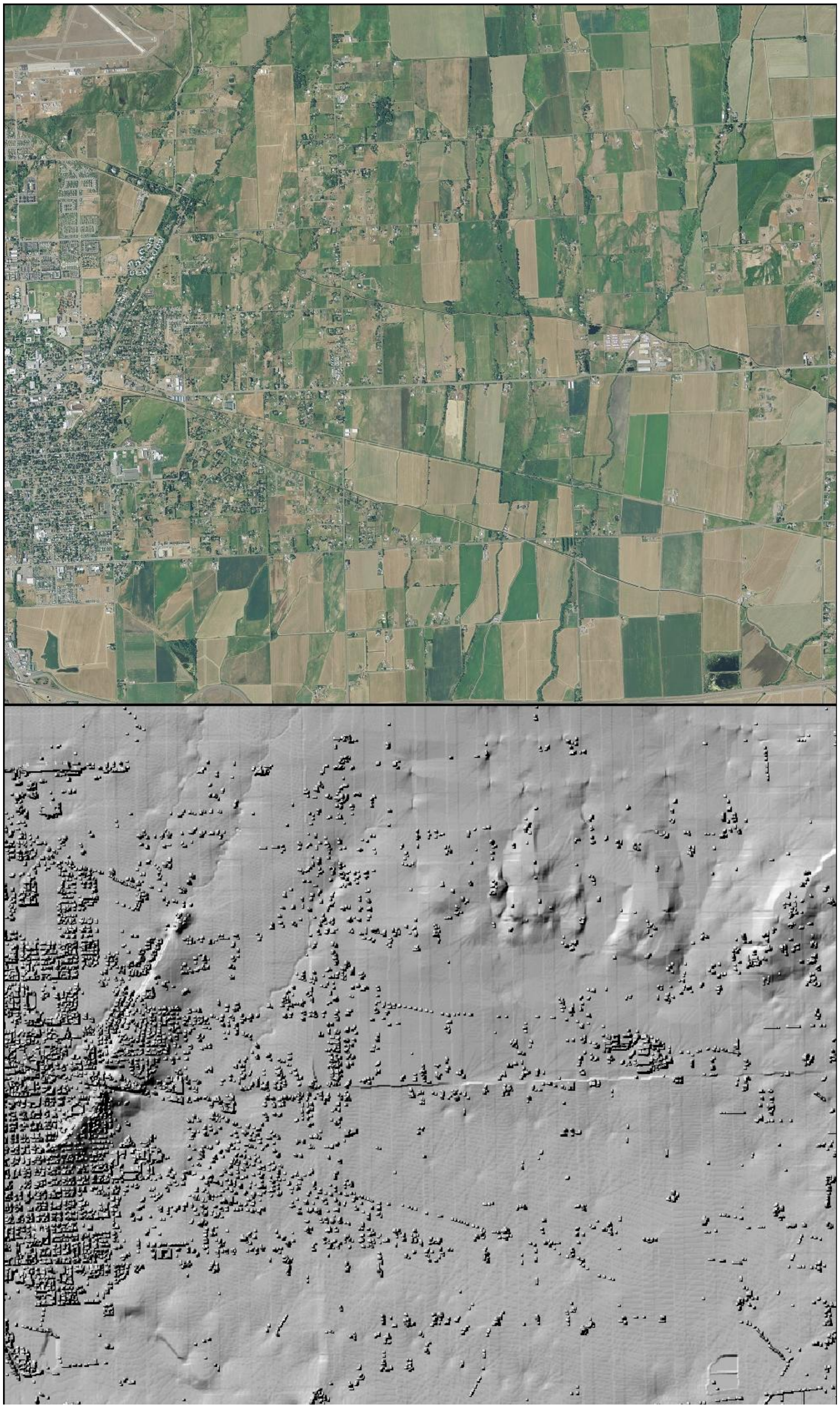


Figure 9. Example of DEM with buildings.

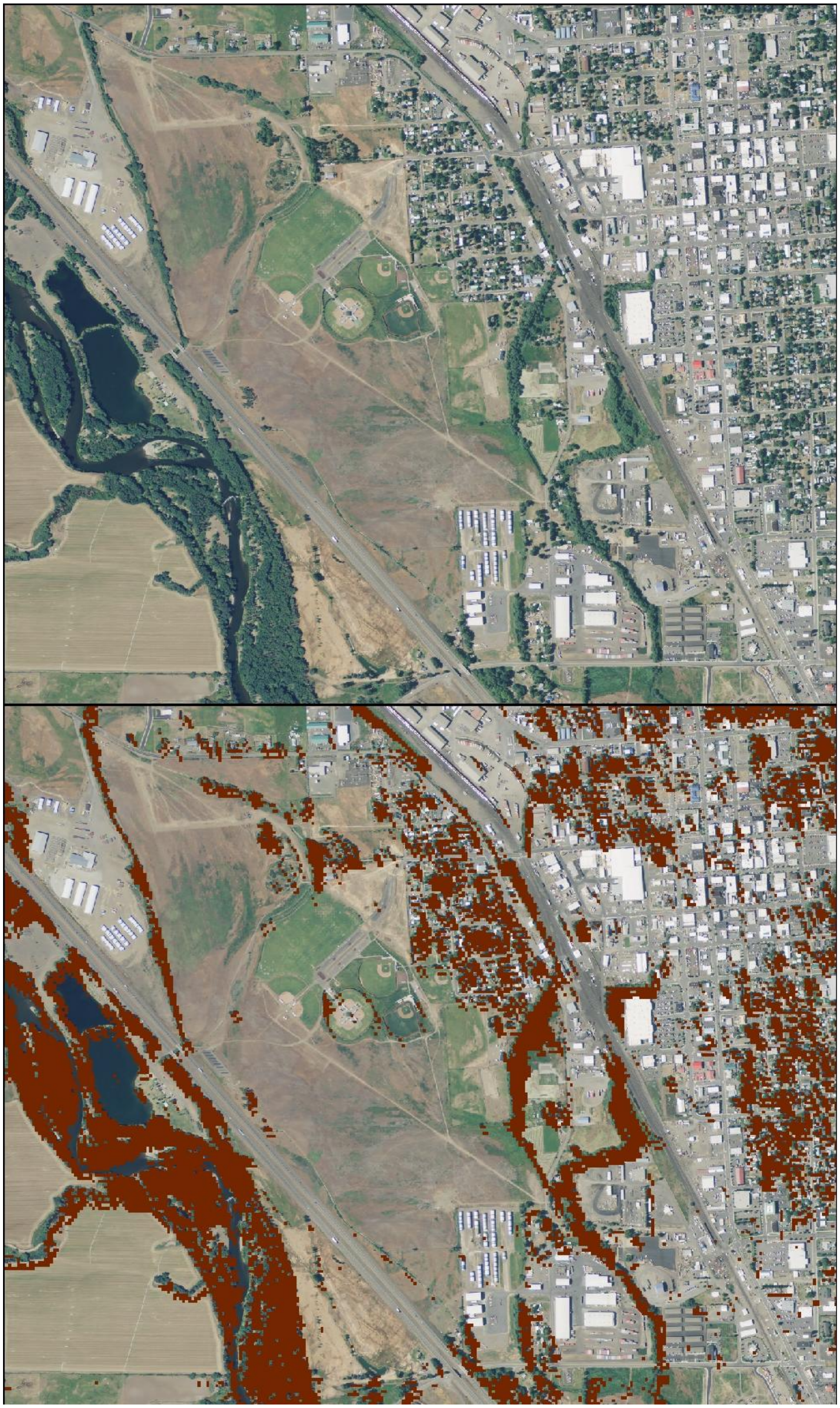


Figure 10. Example of trees extracted from 1m resolution NAIP color ortho photographs.

Figure 11. Example of DEM with trees.

To create the final DEM with buildings and trees, combine the tree DEM with the buildings DEM. Create a conditional statement in Map Algebra to add the trees DEM to the buildings DEM. Use a condition that states wherever the tree height DEM is greater than the buildings DEM the output raster should equal the tree height DEM; otherwise the output raster should equal the buildings DEM. (Example expression: raster = con(tree height DEM > buildings DEM, tree height DEM, buildings DEM)). The result is a DEM of the earth's surface including buildings and trees (see Figure 12).

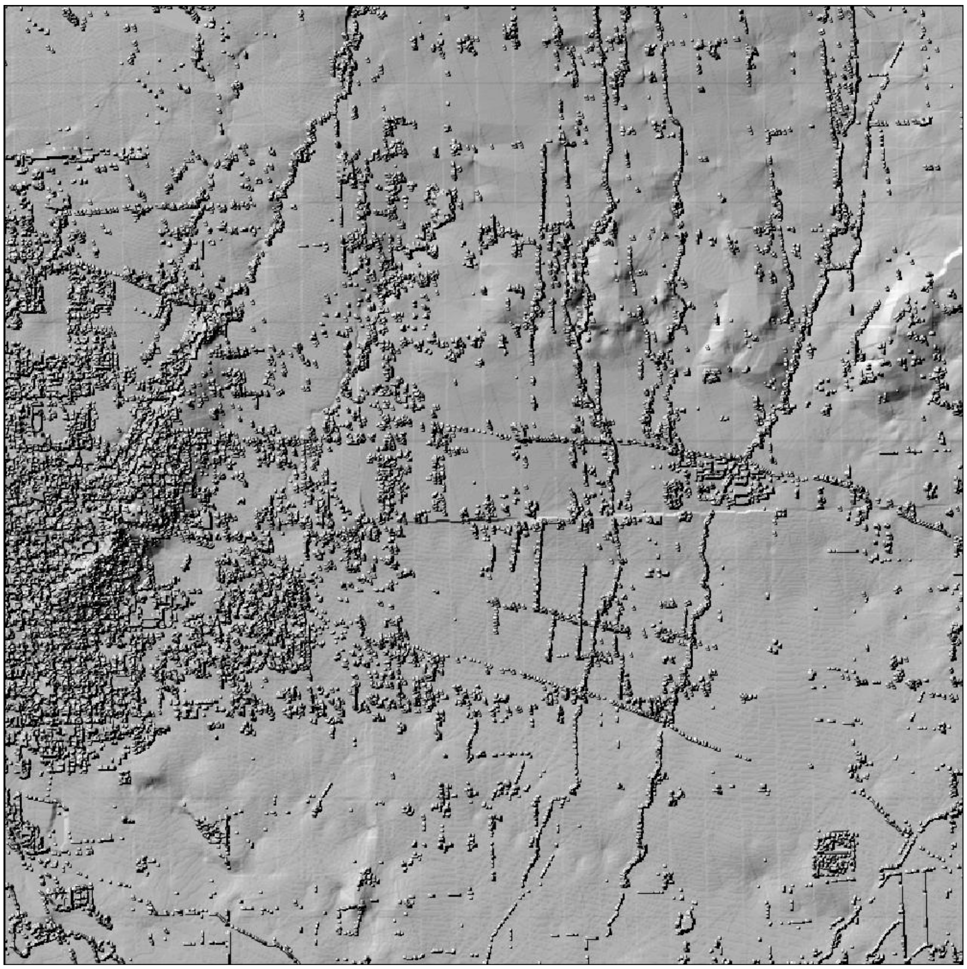


Figure 12. DEM that includes buildings and trees.

Calculating the Visual Sensitivity Model

Objective: Create a surface that represents variations in visibility across a regional terrain as viewed by the resident population.

Spatial Data: DEM with buildings and trees, USGS DEM, sample observer points, Census Block feature class.

Output: Weighted visibility surface.

To create the weighted visibility surface, viewsheds must be created for every observer point and overlaid and summed. The pixel values of each viewsheds must equal the weight of the observer point from which it was generated. Because the viewshed weights are based on population density, when the viewsheds are summed, the resulting raster shows the trends in visibility to the population across the terrain.

First, before the viewshed can be run, the DEM must be prepared. The surface elevation of the DEM cannot be used as the elevation of the observer points. The DEM contains trees that frequently over hang roads, if the raw DEM is used to generate the observer point elevations, observer points will frequently be assigned elevations that are on the tops of trees; not on the surface of the path. One option to resolve this is to create a field in the observer points feature class and use the original USGS DEM to populate it with bare earth elevations. The viewshed tool can use these elevations in the calculations, but this approach is problematic. Using this method the elevation of the observer points will frequently fall below the surface of the DEM from which the viewsheds are being calculated. This will cause an error in the viewshed algorithm and the viewshed will not be calculated. To resolve this situation the elevation values of the DEM must be lowered to the original surface elevation. To achieve this, use buffers around each observer point as masks to lower the elevation of the DEM in those locations. Create 15 m buffers around each observation point (see Figure 13). Convert the buffers to a raster dataset and reclassify the output to make the area of the buffers equal to 1 and the remaining areas equal to 0. In Map Algebra, use a conditional statement to make the buildings and trees DEM equal to the USGS elevation where the buffers exist (Example statement: raster = con(buffer == 1, USGS DEM, buildings and trees DEM)). It is important to not make the buffers too large; this will negate the work that was done to add buildings and trees to the USGS DEM. A buffer with a value of 15m will result in a circle with a radius of 30m. The resolution of the DEM is 10m, so when the buffers are converted to a raster dataset, at most 9 cells will represent the buffer. A larger buffer will mask out too many trees and buildings and a smaller buffer does not provide enough cells for the Viewshed Tool to consistently interpolate the observer point at the surface elevation. Do not create a spot elevation field in the observer points feature class. Allow the viewshed tool to interpolate the elevations from the DEM. This avoids ever having a situation, in which observer point elevations are below the DEM elevation. This will cause an error when calculating viewsheds and processing of viewsheds will stop.

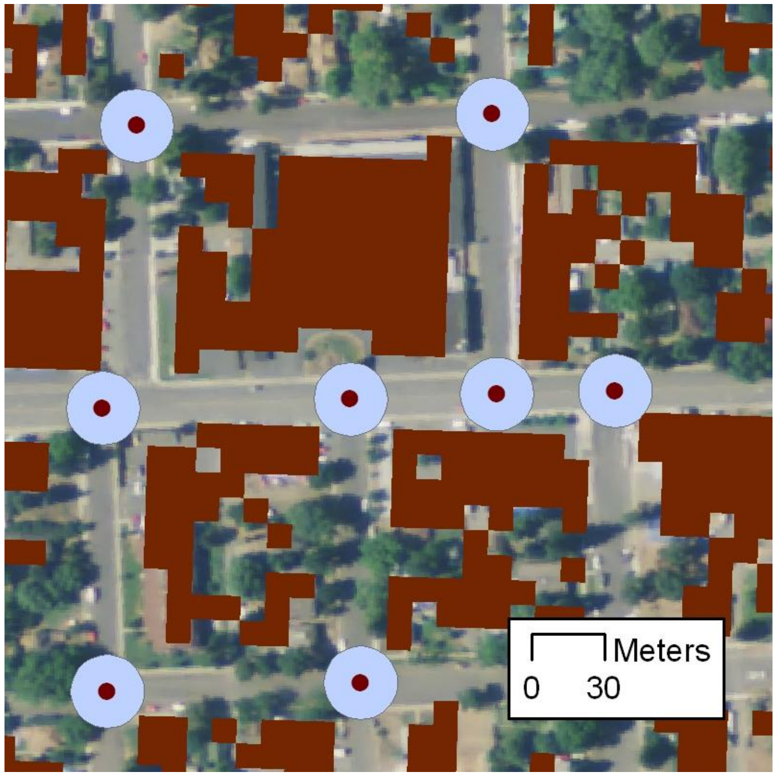


Figure 13. 15m buffers around observer points in relation to extracted building features.

Now, the observer point feature class must also be prepared before viewshed calculations can begin. By default, the Viewshed tool runs calculations from an elevation that is interpolated as the surface of the DEM, but model is intended to be an analysis from the height of an observer on the surface of the earth. So, an observer height for each point must be added to the observer points feature class in a fielded that must be named OFFSETA. The suggested observer height is 1.5 meters. The viewshed tool adds this value to the interpolated observer point elevation. Be sure to convert the observer height to the same units of measure as the DEM.

At this stage the viewshed calculations can begin. To assist the process of calculating weighted viewsheds, RGIS-PNW has developed a Visual Basic for Applications (VBA) Weighted Viewshed script that will calculate a viewshed for each point that is selected in the observer point feature class, apply the observer point weight to the resulting raster, name the raster using the unique ID value, and output all of the viewsheds to a single directory (see Appendix for the scipt). The script is configured to account for the curvature of the earth, but it does not contain a light refraction coefficient. To use the script, import it into the Visual Basic Editor and run it. Set the parameters in the dialog window that appears (see Figure 14).

Use the Weighted Viewshed script to calculate batches of viewsheds. The size of the batches depends on the size of your resulting viewsheds and your available disk space. Trial and error will help you estimate the number of viewsheds that can be calculated in one batch before too much disk space is used. It is important to leave free space that is equal to the disk space taken by the resulting viewsheds. This space is needed for the next step of the process, which is

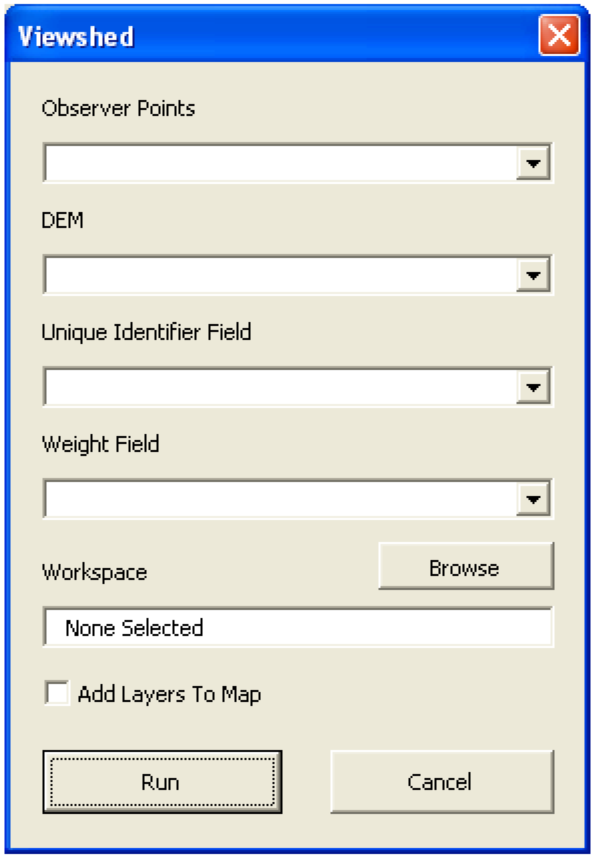


Figure 14. Weighted Viewshed tool dialog window.

summing the viewsheds. Select points from the observer point feature class for batchprocessing as you see fit. One suggestion is to first process batches for incorporated cities in the region of interest to create weighted visual sensitivity models for each. After processing the batches for incorporated cities, move on to processing batches for unincorporated areas. Use the observer point unique IDs to process them in sequential order. Before proceeding, be sure to create a new observer point feature class that does not have observer points that fall within incorporated city boundaries.

After the batches of weighted viewsheds have been created, they must be summed. Use the Weighted Sum Tool and leave the weight value field set to 1 to retain the original weight of the viewshed. The output will be a single raster that is the sum of all weighted viewsheds in the batch. The Weighted Sum tool will generate a large number of temporary files. The disk space required by the temporary files will equal the combined size of the input viewsheds. Be sure to set the scratch workspace of the Weighted Sum tool to a directory on a storage device that has enough free disk space. The rasters generated from each batch of viewsheds must be summed. Add the summed viewshed rasters using Map Algebra to create the final Visual Sensitivity Model (see Figure 15).

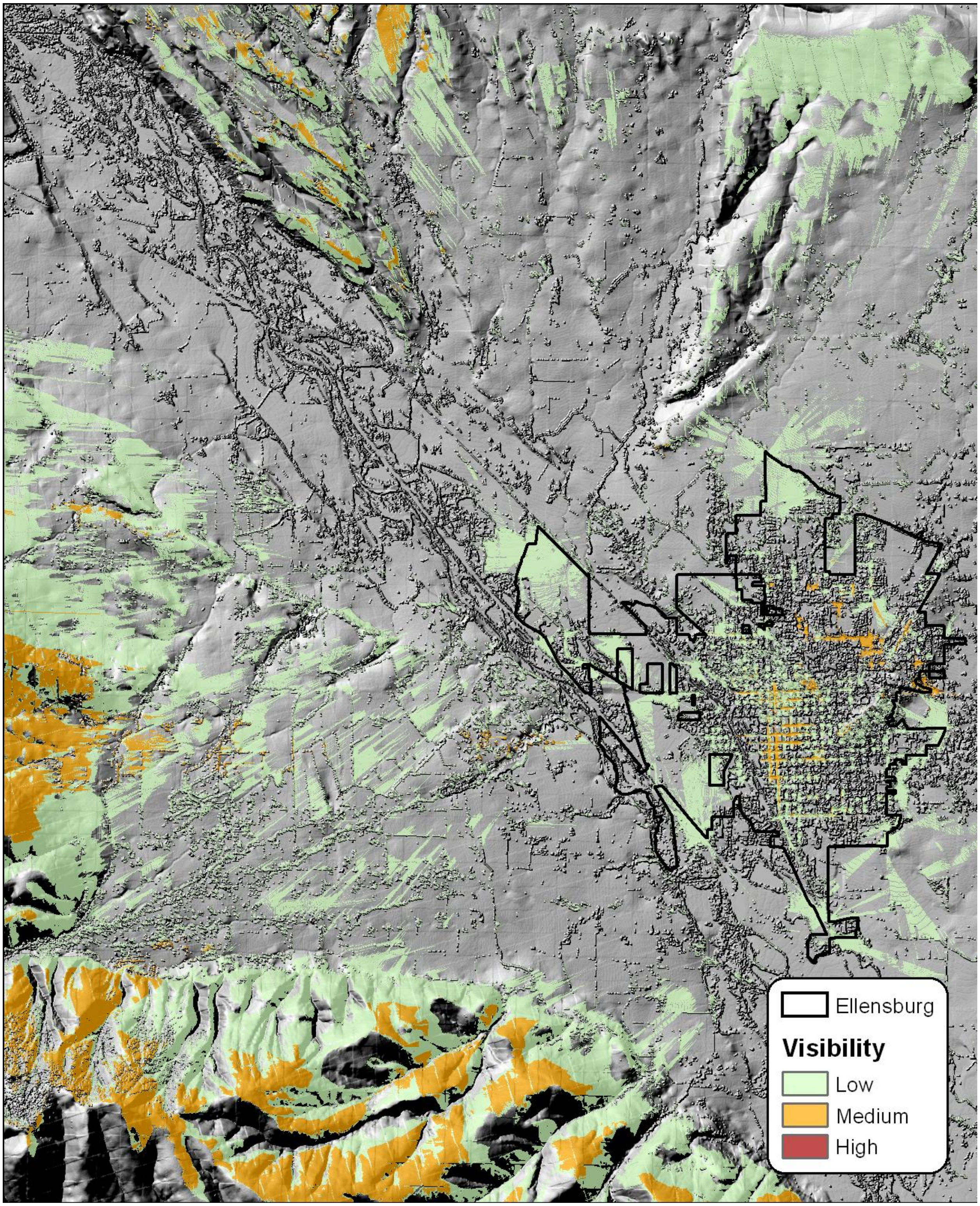


Figure 15. Visual Sensitivity Model for Ellensburg, Kittitas County, WA

Using the Visual Sensitivity Model for Siting Renewable Energy Projects

When using the Visual Sensitivity Model for analysis, it is important to remember the intent of the model. The weighted visibility model shows large scale trends in visibility across a regional terrain. It is not indented for fine determinations of the visibility of specific small scale objects, such as an individual house. The accuracy of the Visual Sensitivity Model is depends upon the accuracy of the elevation model that is used to calculate the viewsheds. There are points in the methodology where inaccuracies are introduced. Extracting buildings from the color imagery will always results in some errors of omission and commission. Building and tree heights are only averages of heights in different regions; therefore their blocking effects may not always be modeled accurately. In the model trees are objects that are solid from the surface upward, when in reality many trees form canopies that are not solid and exist 3 or more meters above the surface of the earth. Additionally, deciduous trees lose their leaves in winter making them even less of a blocking object. Regardless of these inaccuracies, the model should represent the visibility of the terrain more accurately than using a bare earth DEM. When using a bare earth DEM, visibility of locations will be overestimated because there are no features that block views. Using the visual sensitivity assessment methodology will result in a more conservative analysis of visibility because blocking features are incorporated into the DEM.

The visibility values in visual sensitivity model are weights based on population density that represent the relative visibility of terrain to residents. The values do not represent actual population numbers, additional analysis could be conducted to estimate the number of residents that can see a highly visible area. This could be accomplished by creating viewsheds from a sample of observer points in a highly visible area of terrain and overlaying the visible areas on Census Block Group Data. The proportion of visible area in each Census Block Group could be used to calculate the proportion of the population in each group that can see the highly visible area. Then, the proportions of population can be summed to estimate the total population that can see the area.

The Visual Sensitivity Model is useful for regional land use planning. It can aid decision-makers in designating zoned areas for large scale renewable energy projects, such as wind farms and solar arrays, prior to plans for their development. It is important to remember that the model is intended to objectively represent levels of visibility to the population across a terrain. The model does not account for local cultural preferences for the aesthetics of these projects. Ancillary data must be analyzed in conjunction with the visual sensitivity model to determine if viewing wind turbines or solar arrays is a positive or negative experience for residents. This can be accomplished through scenic preference surveys (Wherrett, 2000). Also, culturally important views should be considered in conjunction with scenic preferences when determining siting locations. This type of data can be obtained using spatial surveys that employ public participation mapping to gather local knowledge (Cordner, 2008).

References

Boldstad, P. (2002). GIS Fundamentals: A first text on geographic information systems. White Bear Lake, MN: Eider Press.

Cordner, D. (2008). *Critical community places: Assessing landscape meaning in Ellensburg, Washington* (Master's thesis). Central Washington University, Ellensburg, WA.

Cordner, D. (2007). The power of internet mapping: Kittitas County Open Mapping Project empowers citizens as they plan for the future. National Consortium for Rural Geospatial Innovations-Pacific Northwest. Retrieved from http://csi.gis.cwu.edu/~csi/ Research/Projects2006/KittitasFinal.pdf

De Floriani, L. & Magill, P. (2003). Algorithms for visibility computation on terrains: a survey. *Environment and Planning B: Planning and Design, 30,* 709-728.

Deller, S. C., Tsai, T., Marcouiller, D. W., & English, D. B. K. (2001). The role of amenities and quality of life in rural economic growth. *American Journal of Agricultural Economics, 83*(2), 352-365.

Jackson, P. L., & Kuhlken, R. (2005). *A rediscovered frontier: land use and resource issues in the New West*. Lanham, MD: Rowan & Littlefield.

McGranahan, D. A. (1999). *Natural amenities drive rural population change* (Agricultural Economic Rep. 781). U.S. Department of Agriculture, Food and Rural Economics Division, Economic Research Service.

Miller, J.E., Nelson, S.A., & Hess, G.R. (2009). An object extraction approach for impervious surface classification with very-high-resolution imagery. *The Professional Geographer, 61*(2), 250-264.

Smith, M. D. & Krannich, R. S. (2000). Culture clash revisited: Newcomer and longer-term residents' attitudes toward land use, development, and environmental issues in rural communities of the Rocky Mountain West. *Rural Sociology, 65*(3), 396-421.

Stewart, S. I. (2001). Amenity migration. In K. Luft, & S. MacDonald (Eds.), *Trends 2000: Shaping the future: Proceedings of the Fifth Outdoor Recreation and Tourism Trends Symposium* (pp. 369-378). Lansing, MI: Department of Park, Recreation, and Tourism Resources, Michigan State University.

Wherret, J.R. (2000). Creating Landscape Preference Models Using Internet Survey Techniques. *Landscape Research, 25*(11), 79-96.

Wheatley D. (1995). Cumulative Viewshed Analysis: A GIS-based method for investigating intervisibility, and its archaeological application. In G. R. Lock & Z. Stancic (Eds.), *Archaeology and GIS: A European Perspective* (pp. 5-13), London: Routledge,

Wolsink, M. (2007). Wind power implementation: The nature of public attitudes: Equity and fairness instead of "backyard motives". *Renewable and Sustainable Energy Reviews, 11*(6), 1188-1207.

Visual Learning Systems (VLS). 2010. Feature Analyst 5.0 for ArcMap Reference Manual. Missoula, MT: VLS. Retrieved from http://www.vls-inc.com/.

Weighted Viewshed Tool

```
'Weighted Viewshed Tool
'Script developed by Micheal Joslin for the Center for Spatial Information,
'Central Washington University
'Creates viewsheds from a selection of observer points in a feature class.
'Applies a weighted value to each resulting raster based on a field value.
'Names the raster based on a unique ID in a field.
'Outputs all rasters to a common directory.


Private Sub ComboBox1_Change()
Dim pMxDoc As IMxDocument
Dim pMap As IMap
Set pMxDoc = ThisDocument
Set pMap = pMxDoc.FocusMap
Dim p As Long
Dim n As Long

If ComboBox1.Text = "" Then

Else

'populate combobox3 - unique identifier field
Dim pFeatureLayer As IFeatureLayer
Set pFeatureLayer = ReturnLayer(pMap, ComboBox1.Text)
Dim pFeatureClass As IFeatureClass
Set pFeatureClass = pFeatureLayer.FeatureClass
Dim pFields As IFields
Set pFields = pFeatureClass.Fields
Dim pField As IField
Dim numFields As Long
numFields = pFields.FieldCount

ComboBox3.RemoveItem (0)

For p = 0 To numFields - 1
    Set pField = pFields.Field(p)
    ComboBox3.AddItem pField.name
Next p

'populate combobox4 - weight field
Dim pFeatureLayer2 As IFeatureLayer
Set pFeatureLayer2 = ReturnLayer(pMap, ComboBox1.Text)
Dim pFeatureClass2 As IFeatureClass
Set pFeatureClass2 = pFeatureLayer2.FeatureClass
Dim pFields2 As IFields
Set pFields2 = pFeatureClass2.Fields
Dim pField2 As IField
Dim numFields2 As Long
numFields2 = pFields2.FieldCount
```

```
ComboBox4.RemoveItem (0)

For n = 0 To numFields2 - 1
    Set pField2 = pFields2.Field(n)
    ComboBox4.AddItem pField2.name
Next n

End If

End Sub

Private Sub ComboBox2_Change()

End Sub

Private Sub ComboBox3_Change()

End Sub

Private Sub ComboBox4_Change()

End Sub

Private Sub CommandButton1_Click()

Dim pMxDoc As IMxDocument
Dim pMap As IMap
Dim pRasterLayer As IRasterLayer
Dim pRaster As IRaster
Dim pFeatureLayer As IFeatureLayer
Dim pLookoutDataset As IGeoDataset

Dim pFeatureSelection As IFeatureSelection
Dim pFeatureCursor As IFeatureCursor
Dim pFeature As IFeature

Dim pFeatureSelection2 As IFeatureSelection
Dim pQueryFilter As IQueryFilter
Set pQueryFilter = New queryFilter

Set pMxDoc = ThisDocument
Set pMap = pMxDoc.FocusMap
Dim pActiveView As IActiveView

Dim pStatusBar As IStatusBar
Dim i As Long
Dim j As Long
Dim k As Long

Set pStatusBar = Application.StatusBar
Set pProgbar = pStatusBar.ProgressBar

If ComboBox1.Text = "" Then
    MsgBox "Please select an observer points dataset"
    Exit Sub
```

```
End If

If ComboBox2.Text = "" Then
    MsgBox "Please select a DEM"
    Exit Sub
End If

If ComboBox3.Text = "" Then
    MsgBox "Please select a unique identifier field"
    Exit Sub
ElseIf ComboBox3.Text = "Please Select Observer Dataset" Then
    MsgBox "Please select a unique identifier field"
    Exit Sub
End If

If ComboBox4.Text = "" Then
    MsgBox "Please select a weight field"
    Exit Sub
ElseIf ComboBox3.Text = "Please Select Observer Dataset" Then
    MsgBox "Please select a weight field"
    Exit Sub
End If

If TextBox1.Text = "" Then
    MsgBox "Please specify a workspace"
    Exit Sub
ElseIf TextBox1.Text = "None Selected" Then
    MsgBox "Please specify a workspace"
    Exit Sub
End If

Set pRasterLayer = ReturnLayer(pMap, ComboBox2.Text)
Set pRaster = pRasterLayer.Raster

Set pFeatureSelection = ReturnLayer(pMap, ComboBox1.Text)

Set pFeatureSelection2 = ReturnLayer(pMap, ComboBox1.Text)

If pFeatureSelection.SelectionSet.Count = 0 Then
    MsgBox "Please at least one point"
    Exit Sub
End If

k = pFeatureSelection.SelectionSet.Count

pFeatureSelection.SelectionSet.Search Nothing, False, pFeatureCursor
Set pFeature = pFeatureCursor.NextFeature
For i = 0 To pFeatureSelection.SelectionSet.Count

If pFeature Is Nothing Then

Set pActiveView = pMap
pActiveView.Refresh

ViewshedForm.Caption = "Weighted Viewshed Tool"
```

```
Exit Sub

Else

j = i + 1
ViewshedForm.Caption = "Viewshed Processing " & j & " of " & k

UID = pFeature.Value(pFeature.Fields.FindField(ComboBox3.Text))
Weight = pFeature.Value(pFeature.Fields.FindField(ComboBox4.Text))

pFeatureSelection.Clear

pQueryFilter.WhereClause = "Point_ID =" & UID


pFeatureSelection.SelectFeatures pQueryFilter, esriSelectionResultNew, True

Set pFLayer = ReturnLayer(pMap, ComboBox1.Text)

Dim pFeatureLayerDefinition As IFeatureLayerDefinition
Set pFeatureLayerDefinition = pFLayer

Dim pSelLayer As IFeatureLayer
Set pSelLayer = pFeatureLayerDefinition.CreateSelectionLayer("Temp", True, "", "")

Set pLookoutDataset = pSelLayer

Dim pSurfaceOp As ISurfaceOp
Dim pEnv As IRasterAnalysisEnvironment
Dim pWS As IWorkspace
Dim pWSF As IWorkspaceFactory
Dim pOutRaster As IGeoDataset

Set pSurfaceOp = New RasterSurfaceOp
Set pEnv = pSurfaceOp
Set pWSF = New RasterWorkspaceFactory
Set pWS = pWSF.OpenFromFile(TextBox1.Text, 0)
Set pEnv.OutWorkspace = pWS
pEnv.SetCellSize esriRasterEnvMaxOf, pRaster

If pWS.Exists = False Then
    MsgBox "Please specify a valid workspace"
    Exit Sub
End If

Set pOutRaster = pSurfaceOp.Visibility(pRaster, pLookoutDataset,
esriGeoAnalysisVisibilityFrequencyUseCurvature)

Dim pRasterMakerOp As IRasterMakerOp
Set pRasterMakerOp = New RasterMakerOp

Dim pEnv2 As IRasterAnalysisEnvironment
Set pEnv2 = pRasterMakerOp
Dim pWS2 As IWorkspace
```

```
Dim pWSF2 As IWorkspaceFactory
Set pWSF2 = New RasterWorkspaceFactory
Set pWS2 = pWSF2.OpenFromFile(TextBox1.Text, 0)
Set pEnv2.OutWorkspace = pWS2
pEnv2.SetCellSize esriRasterEnvMaxOf, pOutRaster

Dim pWeightRaster As IGeoDataset
Set pWeightRaster = pRasterMakerOp.MakeConstant(Weight, False)

Dim pMathOp As IMathOp
Set pMathOp = New RasterMathOps

Dim pEnv3 As IRasterAnalysisEnvironment
Set pEnv3 = pMathOp
Dim pWS3 As IWorkspace
Dim pWSF3 As IWorkspaceFactory
Set pWSF3 = New RasterWorkspaceFactory
Set pWS3 = pWSF3.OpenFromFile(TextBox1.Text, 0)
Set pEnv3.OutWorkspace = pWS3
pEnv3.SetCellSize esriRasterEnvMaxOf, pOutRaster

Dim pOutRaster2 As IGeoDataset
Set pOutRaster2 = pMathOp.Times(pOutRaster, pWeightRaster)

If CheckBox1.Value = True Then
    Dim pRLayer As IRasterLayer
    Set pRLayer = New RasterLayer
    pRLayer.CreateFromRaster pOutRaster2
    pRLayer.name = "" & UID & ""
    pMap.AddLayer pRLayer
End If

Dim pWS4 As IWorkspace
Dim pWSF4 As IWorkspaceFactory
Set pWSF4 = New RasterWorkspaceFactory
Set pWS4 = pWSF4.OpenFromFile(TextBox1.Text, 0)

Dim pRaster2 As IRaster
Set pRaster2 = pOutRaster2

Dim pSaveAs2 As ISaveAs2
Set pSaveAs2 = pRaster2

Dim RasterName As String
RasterName = UID & ".tif"

pSaveAs2.SaveAs RasterName, pWS4, "TIFF"

Set pFeature = pFeatureCursor.NextFeature

pFeatureSelection2.Clear

End If

Next
```

```
End Sub
Private Function ReturnLayer(pMap As IMap, sLayerName As String) As ILayer

'Find a layer in a map document
'Return the layer or nothing if not found

Dim pEnumLayers As IEnumLayer
Dim pLayer As ILayer

Set pEnumLayers = pMap.Layers(Nothing, True)
pEnumLayers.Reset
Set pLayer = pEnumLayers.Next

Set ReturnLayer = Nothing

Do Until pLayer Is Nothing
If pLayer.name = sLayerName Then
Set ReturnLayer = pLayer
Exit Do
End If
Set pLayer = pEnumLayers.Next
Loop

End Function

Private Sub CommandButton2_Click()

Dim pD As IGxDialog
Dim pObj As IEnumGxObject
Dim pO As IGxObject
Dim pFil As IGxObjectFilter
Dim pSelect As GxSelection
Dim pFullName As IGxObject

Set pFil = New GxFilterContainers

Set pD = New GxDialog
pD.AllowMultiSelect = False
pD.StartingLocation = "c:\temp"
pD.RememberLocation = False
pD.Title = "Select Location"
pD.ButtonCaption = "Save To"
Set pD.ObjectFilter = pFil
pD.DoModalOpen 0, pObj
Set pO = pObj.Next

If Not pO Is Nothing Then

pLocation = pO.FullName
TextBox1.Text = pLocation

End If

Set pD = Nothing
```

```
Set pObj = Nothing
Set pO = Nothing
Set pFil = Nothing
Set pSelect = Nothing
Set pFullName = Nothing

Exit Sub

End Sub

Private Sub CommandButton3_Click()
Unload ViewshedForm
Set ViewshedForm = Nothing
End Sub

Private Sub Label1_Click()

End Sub

Private Sub TextBox1_Change()

End Sub

Private Sub UserForm_Click()

End Sub

Private Sub UserForm_Initialize()
Dim pMxDoc As IMxDocument
Dim pMap As IMap
Dim L As Long
Dim m As Long
Dim TableCollection As IStandaloneTableCollection

Set pMxDoc = ThisDocument
Set pMap = pMxDoc.FocusMap

'populate combobox1 - observer points
For L = 0 To pMxDoc.FocusMap.LayerCount - 1
    If TypeOf pMap.Layer(L) Is IFeatureLayer Then
        ComboBox1.AddItem pMxDoc.FocusMap.Layer(L).name
    End If
Next L

'populate combobox2 - DEM
For m = 0 To pMxDoc.FocusMap.LayerCount - 1
    If TypeOf pMap.Layer(m) Is IRasterLayer Then
        ComboBox2.AddItem pMxDoc.FocusMap.Layer(m).name
    End If
Next m

'populate combobox3 - unique identifier field
ComboBox3.AddItem "Please Select Observer Dataset"

'populate combobox4 - weight field
```

```
ComboBox4.AddItem "Please Select Observer Dataset"

TextBox1.Text = "None Selected"

End Sub
```