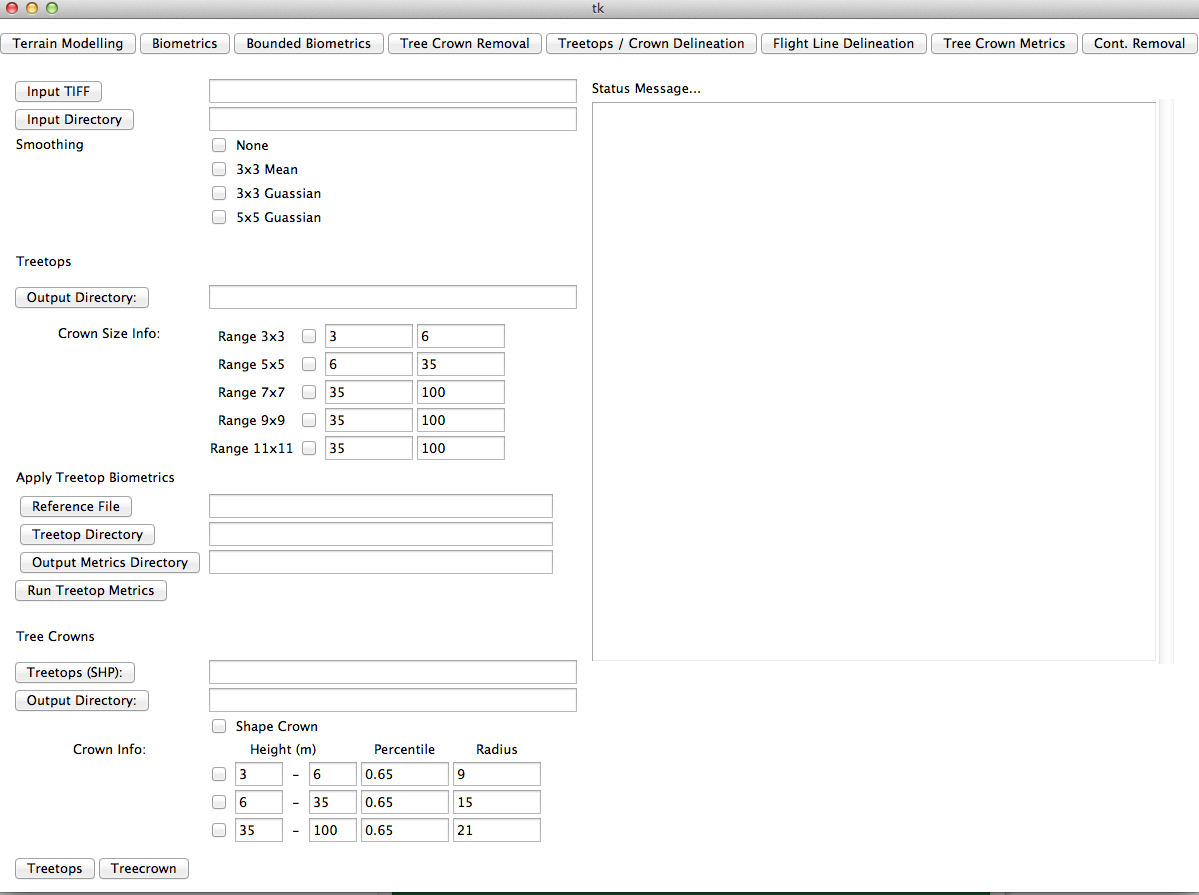
Treetops



Input:

* CHM in the form of a TIFF file
* Crown Size Information given for up to 5 discrete elevations ranges as selected by the user. The user maps the desired elevation ranges to search grids of size 3x3, 5x5, 7x7, 9x9, and 11x11.
* One of the following smoothing strategies:
  + None – no smoothing is applied
  + 3x3 Mean – each pixel is replaced by the mean elevation of the 3x3 grid centered at that pixel
  + 3x3 Gaussian – apply 3x3 Gaussian filter with variance of 1
  + 5x5 Gaussian – apply 5x5 Gaussian filter with variance of 2

|  |  |  |
| --- | --- | --- |
| 0.0751136 | 0.123841 | 0.0751136 |
| 0.123841 | 0. 204180 | 0.123841 |
| 0.0751136 | 0.123841 | 0.0751136 |

3x3 Gaussian Matrix (variance 1)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 0.01247 | 0.02641 | 0.03391 | 0.02641 | 0.01247 |
| 0.02641 | 0.05592 | 0.07180 | 0.05592 | 0.02641 |
| 0.03391 | 0.07180 | 0. 0921 | 0.07180 | 0.03391 |
| 0.02641 | 0.05592 | 0.07180 | 0.05592 | 0.02641 |
| 0.01247 | 0.02641 | 0.03391 | 0.02641 | 0.01247 |

5x5 Gaussian Matrix (variance 2)

Algorithm:

The algorithm works by visiting each pixel and determines if the pixel under consideration satisfied the conditions for becoming a treetop.

Determining if a pixel is a treetop:

1. Apply the desired smoothing technique to the CHM.
2. Apply small perturbations to each pixel value. (This has been copied from Raf’s code. Its purpose is to reduce the probability of neighboring pixels having the exact same value.)
3. Visit each pixel p in the smoothed CHM.
4. Determine the crown size information p based on its elevation.
5. If p is the highest pixel in the grid of size k centered around p in the smoothed CHM then the pixel is determined to be a treetop. (**Note:** p may not be the highest pixel in the original (non-smoothed) CHM. To ensure the treetop is always chosen as the highest pixel in the non-smoothed CHM the tree top is assigned to the highest pixel in the 3x3 window centered around p in the original CHM.)

Tree Crowns

The tree crown delineation algorithm is based on the algorithm given by Tiede et al. [1]. Tiede et al. [1] give a tree top determination algorithm that uses a dynamic search radius determined by the crown-width (). The crown-width is defined as the following function of tree-height ()

where and are constants. Our tree crown delineation algorithm, similar to the treetop algorithm, uses discrete search radiuses as determined by user input.

Input:

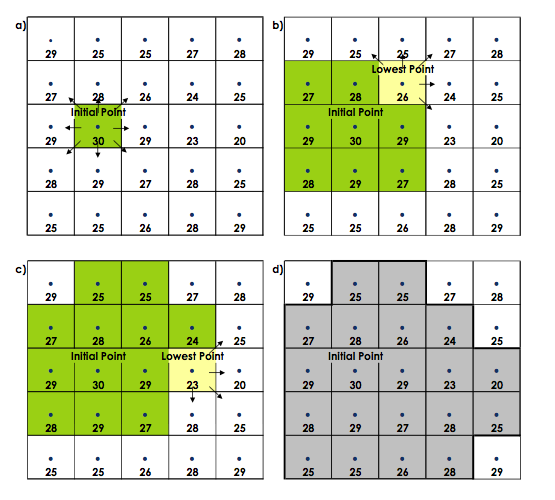
* CHM in the form of a TIFF file.
* Treetops in the form of a shape file representing the seed points for the tree crown delineation.
* Shape Crown checkbox that specifies whether the algorithm will attempt to shape the crown.
* Crown properties for up to 3 discrete elevation ranges. The properties are:
  + Percentile – any raster pixel with elevation < the defined percentile of the treetop height will be considered too low to belong to the tree crown. This is used as a stopping condition in the algorithm.
  + Radius – defines the maximum tree crown radius. This is used to define the size of the window centered around the treetop for which the tree crown is delineated.

Algorithm:

1. A smoothed CHM is generated base on the chosen smoothing strategy.
2. The algorithm visits each treetop in the treetop file as specified by the user. The seed point is defined to be the highest point in the smoothed CHM in the 3x3 grid surrounding the pixel corresponding to the treetop. This is done so that the seed point is the highest in the smoothed CHM.
3. The crown is built using a region-growing algorithm starting at the seed pixel.
   1. Every nearest neighbor point is compared with the initial point. Nearest neighbors are defined as the 8 neighboring pixels.
   2. Each neighbor that satisfies the following criterion is added to the crown
      1. Lower elevation
      2. Elevation > specified percentile of the associated treetop height
      3. Point is within defined search radius
   3. This process is repeated with each neighbor added to the crown as the seed pixel until no new points satisfy the criterion of b.

Example

The following figure provides an example of how the algorithm would run. Suppose that the percentile here is 60% and therefore only accept pixels into the crown that have an elevation of at least 18. We begin with the initial point in a). We add each neighbor of a) to the crown because each point has a lower elevation that. We repeat with this process in b) checking if its neighboring pixels not in the crown can be added. All neighboring pixels <= 26 and >= 18 are added. This process is again repeated in c) and the final tree crown is show in d).



Bibliography

1. Dirk Tiede, Gregor Hochleitner, and Thomas Blaschke. **A full GIS-based workflow for tree identification and tree crown delineation using laser scanning.**