NASA/TM–20230013431

User's Guide for Creating a Wind Model for UAS Missions using WindNinja

Andrew Turner
National Institute of Aerospace, Hampton, Virginia

November 2023

NASA STI Program...in Profile

Since its founding, NASA has been dedicated to the advancement of aeronautics and space science. The NASA scientific and technical information (STI) program plays a key part in helping NASA maintain this important role.

The NASA STI Program operates under the auspices of the Agency Chief Information Officer. It collects, organizes, provides for archiving, and disseminates NASA's STI. The NASA STI Program provides access to the NASA Aeronautics and Space Database and its public interface, the NASA Technical Report Server, thus providing one of the largest collection of aeronautical and space science STI in the world. Results are published in both non-NASA channels and by NASA in the NASA STI Report Series, which includes the following report types:

- TECHNICAL PUBLICATION. Reports of completed research or a major significant phase of research that present the results of NASA programs and include extensive data or theoretical analysis. Includes compilations of significant scientific and technical data and information deemed to be of continuing reference value. NASA counterpart of peer-reviewed formal professional papers, but having less stringent limitations on manuscript length and extent of graphic presentations.
- TECHNICAL MEMORANDUM. Scientific and technical findings that are preliminary or of specialized interest, e.g., quick release reports, working papers, and bibliographies that contain minimal annotation. Does not contain extensive analysis.
- CONTRACTOR REPORT. Scientific and technical findings by NASA-sponsored contractors and grantees.
- CONFERENCE PUBLICATION. Collected papers from scientific and technical conferences, symposia, seminars, or other meetings sponsored or co-sponsored by NASA.
- SPECIAL PUBLICATION. Scientific, technical, or historical information from NASA programs, projects, and missions, often concerned with subjects having substantial public interest.
- TECHNICAL TRANSLATION. English-language translations of foreign scientific and technical material pertinent to NASA's mission.

Specialized services also include organizing and publishing research results, distributing specialized research announcements and feeds, providing information desk and personal search support, and enabling data exchange services.

For more information about the NASA STI Program, see the following:

- Access the NASA STI program home page at `http://www.sti.nasa.gov`
- E-mail your question to `help@sti.nasa.gov`
- Phone the NASA STI Information Desk at 757-864-9658
- Write to:
  NASA STI Information Desk
  Mail Stop 148
  NASA Langley Research Center
  Hampton, VA 23681-2199

NASA/TM–20230013431

User's Guide for Creating a Wind Model for UAS Missions using WindNinja

Andrew Turner
National Institute of Aerospace, Hampton, Virginia

National Aeronautics and
Space Administration

Langley Research Center
Hampton, Virginia 23681-2199

November 2023

Available from:

NASA STI Program / Mail Stop 148
NASA Langley Research Center
Hampton, VA 23681-2199
Fax: 757-864-6500

Abstract

This guide describes a methodology for creating a Wind Model for use in UAS mission planning. This method uses LiDAR data to create a raster elevation model for the WindNinja software package. The WindNinja software is run iteratively to create a set of xy-plane comma-separated lookup tables using batch processing and the ArcGIS® mapping software offered by Environmental Systems Research Institute, Inc. Python script. Details regarding the specific tools for creating a table database are outlined within this memorandum.

Contents

1 Introduction

A wind model can be used to predict hazardous flying areas for unmanned aircraft system (UAS) missions due to current or forecasted wind conditions. Current wind models offer low spatial resolution real-time and predicted measurements. These models are able to accurately determine conditions with a 1 km resolution. More recently, there has been a lot of interest in micro-scale weather models with high resolution that can provide output at the ground level in urban environments. Wind modeling software applications have been developed to accurately determine wildfire and particulate spread in urban areas. In order to accurately mitigate hazardous wind conditions for UAS missions, a microscale wind model must be used. The model requirements for the development of this method were a 1 meter resolution of wind speed.

In order to create the model, an area of interest (AOI) must be determined. Within this AOI, wind speed and wind direction, at a specified resolution, can be determined using the model and initial input conditions. The modeling method used within this guide creates a static set of lookup tables (LUTs) that can be parsed during a mission. More specifically, this guide provides a method for creating a set of numerically calculated horizontal xy-plane tables for a given range of altitude values and initial wind speeds and wind directions. A simulated three-dimensional flow field that can be extracted as a set of xy-plane LUTs is most desirable for this purpose due to the intensive computational power required to create the flow field in real time. While this method does not allow for wind forecasting, the lookup table data can be used to help predict hazards based upon a current wind condition input reading. The LUTs are created by the software package, WindNinja, which was designed by the Fire, Fuel, Smoke Science Program at the Rocky Mountain Research Station.[1]

In order to use the WindNinja software, all that is required is a raster elevation model of the AOI. The pixel resolution of the raster will directly affect the resulting output accuracy, therefore, having a raster pixel size similar to the desired wind value grid size is recommended. Every location in the raster elevation model must contain a value. Using the raster elevation model and the initial conditions of the wind speed and wind direction as inputs, an xy-plane shape file containing simulated wind speeds and wind directions at a specified resolution for the location can be extracted from WindNinja. The raster creation method implemented in this guide uses LiDAR data (.las files), also known as a LAS Dataset, that can be manipulated within the ArcGIS software to create the raster elevation model. The WindNinja GUI allows the user to extract a single xy-plane shapefile at a time, in order to automate this process for the entire model, this guide includes a python script for creating the entire set of LUTs for a given AOI using the WindNinja command line interface. This guide outlines the required data and the process for creating a set of LUTs for use within UAS mission operations.

As a note to the reader, this guide assumes the user has prior knowledge of how to use the ArcGIS® mapping software offered by Environmental Systems Research

---

[1] https://www.firelab.org/project/windninja

Institute, Inc., or GIS Software, as well as some understanding of the LiDAR point classification process and projected and geographic coordinate systems. Throughout the guide, footnotes have been added to link the official documentation for the tools as they are needed. At the time of this report, there exist online video tutorials for some of the individual processes as well.

2 Technology Used

Listed below are the software and the computer specifications used to create the WindNinja LUTs.

Computer Specifications:

Model: HP Z-Book™ brand laptop available from HP Inc.
Operating System: Windows 10™ x64 brand operating system available from Microsoft Corp.
Processor: Intel® Core™ i9-9880H CPU @ 2.30Ghz (8-Cores) brand processor available from the Intel Corp.
Installed RAM: 32.0 GB

Software:

1. WindNinja (version 3.5.3) wind prediction software, available from the U.S. Government Rocky Mountain Research Station (RMRS) Missoula Fire Science Laboratory under the following license: LICENSE

2. ArcGIS® (version 10.6.1) mapping software available from Environmental Systems Research Institute, Inc., which also provides the required mapping software extensions ArcGIS® Spatial Analyst™ and ArcGIS® 3D Analyst™ [2]

3. Python™ 2.7[3] and Python™ 3.x, programming language available from the Python Software Foundation

4. Windows Notepad++™ plain text editor available from Micorsoft Corp. (Any text-editor will work.)

---

[2] Must include licenses for the mapping extensions.

[3] Python™ 2.7 is included in version 10.6.1 of the mapping software used.

3 Creating a Wind Model

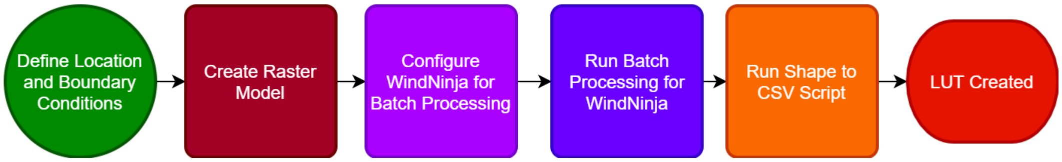


Figure 1: Overview of the Main Process for using WindNinja to create an xy-plane lookup table.

The first task in creating a wind model is to define the location and boundary conditions that the model will use. This includes an AOI, initial input wind conditions, output wind mesh resolution, and raster pixel resolution. Once this is completed, a raster elevation model must be created in order to obtain the numerically calculated wind shape file output from WindNinja. With the raster created, WindNinja can then be configured such that it will create an output file for the initial parameters defined in the first step. A Python script can be written to run the batch processing for all the permutations of initial input wind conditions and output z-values. After the batch processing completes, a script will clip, or cut-out, the WindNinja shape file output using the AOI and then convert the file into a comma separated text-file with longitude(x) and latitude(y) for each (x,y) position in the shape file and the corresponding wind speed and wind direction at each x,y position for each output shape file. These files are the final output and can be used by a UAS Weather Service as a LUT for real-time wind estimation.

3.1 Defining Locations and Boundary Conditions

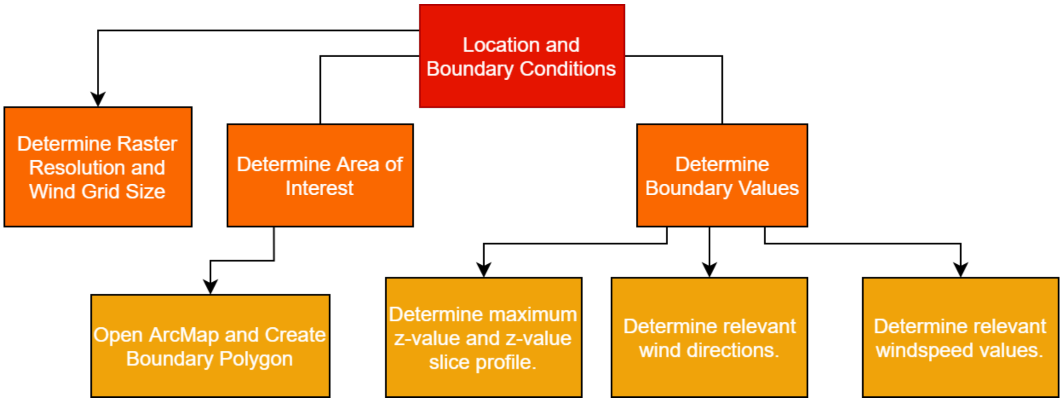


Figure 2: Flow Chart for defining wind model initial parameters.

In order to create the lookup tables necessary for the wind model, the first task is to define the location and boundary conditions. The location can be defined based on the available LiDAR data or pre-made raster data for the area. If there is already a raster elevation model of the area of interest, refer to section 3.2.1 for guidance on ensuring that no null values exist in the raster.

The next task is to define the conditions that the wind model will reference. These initial conditions include a range of wind speeds, wind directions, a z-value profile, and the height of the steady stream incoming wind. An example of these values is shown below.

Example Boundary Conditions:

Wind Speed: 1 - 15 (m/s) (+1 increments)
Wind Direction: 0, 45, 90, 135, 180, 225, 270, 315 (8 cardinal directions)
Z-Value Profile: 1m - 30m (+1m increments); 30m - 100m (+5m increments)
Steady Stream Wind Height: 10m

Some points to consider in choosing the conditions that are of interest include: the upper bound wind speed and maximum altitude allowed for UAV flight, average and maximum heights of the physical objects in the area of interest, and the height of local wind measurements.

With regard to the initial definitions, raster pixel size and wind model mesh size are the last two parameters to note. The raster pixel size is the value that determines the bin size for the LiDAR points. It will determine how many unique elevation values there will be within a given unit size. For example, a raster pixel size of 0.1m would mean that there are 10 unique values per meter. The wind model mesh size is the value used for the spacing of the wind speed and wind direction values in the two-dimensional xy-plane output from WindNinja. The z-value spacing is determined by the z-value profile definition.

The pixel size of the raster model should be as close to, but not less than, the average point spacing of the LiDAR data. The point spacing of the LiDAR data can be determined within the GIS software. If a LAS data set file has not already been created, skip the following information. How to create a LAS data set will be discussed in the next section. If a LAS data set already exists, the point spacing can be determined by following these steps.

Right-click the LAS data set file (*.lasd) within the **Catalog** window and then select *Properties*. Next, navigate to the *Files* tab; within this tab there is a table of the las files included in the dataset, the fourth column indicates the *Point Spacing*. If this column is blank, the statistics will need to be updated within the *Statistics* tab. Use the *Update* button in this tab to generate the statistics for the las files and then return to the *Files* tab. The *Point Spacing* values should now exist and can be used to determine the raster pixel size.

To summarize, before moving on to creating the raster model, these parameters will need to be defined:

1. Location of Interest
2. Existing LiDAR or Raster data coverage of location
3. List of Wind Speeds
4. List of Wind Directions
5. Z-Value profile
6. Height of Steady-Stream Wind measured by local anemometer
7. Maximum height of physical objects within location
8. LiDAR point spacing and Raster pixel size (these values can be determined later)
9. Grid size for WindNinja output shape file

Once these parameters have been defined, creating the raster elevation model is the next task.

3.2 Creating a Raster Elevation Model

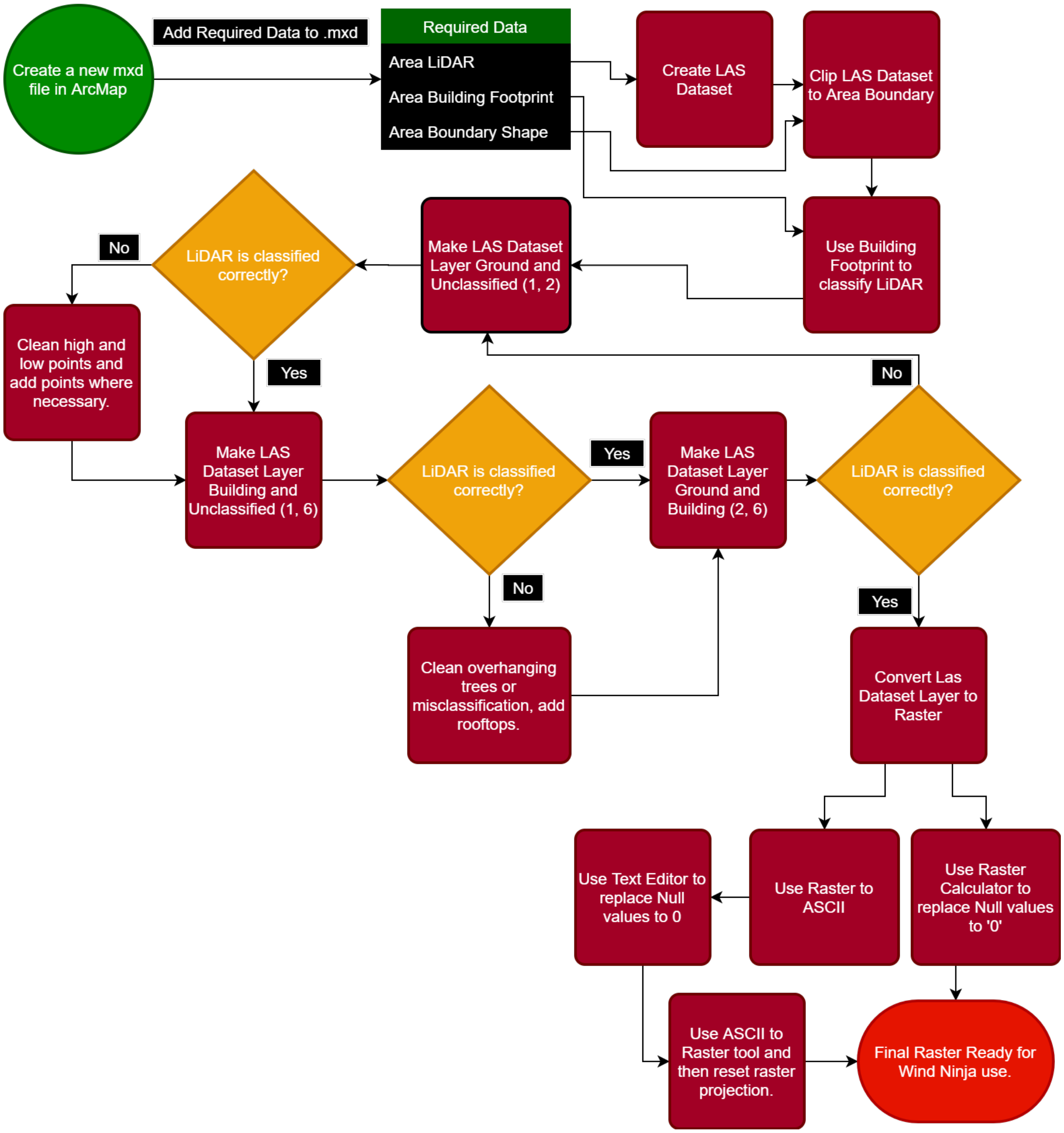


Figure 3: Flow Chart for creating a raster file within the GIS Software.

In order to start the raster elevation model creation process, open the GIS Software. package and create a new *.mxd file for area of interest. Initially, the required data will need to be added to the .mxd file. This data includes LiDAR data as a LAS dataset, building footprint shape file, and the AOI boundary shape file. If these files have already been created, they will likely have meta information stored within them that indicates the projection system being used to define where the data is located on Earth. It is very important to pick one projection and one set of units for the entirety of the process.[4]

NOTE: *There are many tools for editing LiDAR data. This process explains how to do this using the GIS Software.*

Add the existing *.las files to a new LAS dataset.[5] Next, create a new polygon shape[6] and edit it to create a bounding fence for the area of interest.[7] The last part of setting up the mxd file is adding the building footprint shapefile. If the area of interest does not have a building footprint shapefile, all the building points within the LiDAR data will need to be classified manually.

In order to enhance performance and conserve disk space, the LAS dataset can be clipped to the perimeter of the bounding polygon in order to reduce the amount of LiDAR data within the project. Use the *Extract LAS* tool to clip the LAS dataset to the polygon boundary.[8]

Now that the LiDAR data for the area of interest has been added into a new LAS dataset, the next part of the process involves classifying the LiDAR points that are of interest in the raster elevation model; the points that are of interest are the ground(2) and the building(6) points. Most of the points are initially unclassified(1). Every LiDAR point within the dataset can be classified based on what physical object it represents. The standard LiDAR classification numbers are listed below.

LiDAR point classifications:

1. Unassigned
2. Ground
3. Low Vegetation
4. Medium Vegetation
5. High Vegetation
6. Building
7. Noise
8. Model Key / Reserved

In order to re-classify the points of interest, a new LAS Dataset layer containing the unassigned(1) points and the classifications of interest, generally this is the ground(2) or building(6) points, will need to be created. This can be done using the *Make LAS Dataset Layer* tool.[9] This new layer will contain only the points classified

---

[4] Projections Toolset
[5] Creating a LAS dataset
[6] Create a New Shapefile
[7] Editing Shapefiles
[8] Extract LAS
[9] Make LAS Dataset Layer

as such. Toggling the original LAS dataset off within the **Table of Contents** window will allow the new layer to be edited using the LAS Dataset toolbar. In order to reclassify the points, ensure that the correct dataset is selected in the toolbar and that the zoom level is close enough that the LiDAR data points have rendered. The process for manual LiDAR point classification involves using the LAS Dataset Profile viewing tool within the LAS Dataset Toolbar to view a vertical cross-section of the points. Once the cross-section is selected a new window will appear called **Profile View**. Within this window, the toolbar contains point selection options and point editing options. The general process is to select all the points that need to be reclassified and change their classification using the **Edit** window, move to a new cross-sectional area, repeat until the entire area has been classified appropriately. It is a good practice that each cross-section overlaps the previous by a few feet in order to not skip points; it is also important to make the cross-section as thin as possible so that the points can more accurately be classified. The manual classification process can be lengthy and tedious depending on the location size, the objects located within the area of interest, the density of the LiDAR data, and whether or not the LiDAR data has already been classified or not. It is also important to note that any changes to the classification of the points are permanent. A new LAS dataset layer is just a filtered extension of the original set of points.

The following section in the creation of the raster elevation model is iterative:

As shown in Figure 3, create a new LAS Dataset layer containing only the unassigned points and the ground points. Methodically use the cross-section profile viewing tool to reclassify any unassigned points that should be ground or, oppositely, to reclassify any ground points that are not ground to unassigned.

Now, the buildings need to be classified. Create a new LAS Dataset layer containing only the unassigned points and the building points. The fastest way to classify the buildings is to use the building footprint shapefile to classify the points based upon the feature locations within the shapefile with the *Set LAS Class Codes Using Features* tool.[10] The accuracy of this process depends on the date at which the building data within the shapefile was last updated and the date that the LiDAR data was collected. If these two sets of data do not match, some buildings may need to be classified manually while others may need to be removed. This tool classifies all of the points located within the features to a specified LiDAR class. There may be trees, powerlines, noise, or other objects overhanging the buildings. Some buildings may have been demolished. All of the points that are not technically the building must be removed from the building class and set to unassigned. Alternatively, there may be buildings that are not included in the shapefile; therefore, it is best to check the entirety of the area once again to ensure accurate point classification. This can be done using the same process as before with the ground classification, using the edit options in the **Profile View** window after creating a cross-section.

With both the ground and the building points properly classified, create a new LAS Dataset layer containing only the ground and the building points. With this layer, it is important to do a quality check on the point classification. In order

---

[10] Set LAS Class Codes Using Features

to ensure that there are not any high, low, or other irregular points that are not ground or building, a visual inspection can be done using the *Elevation* button in the LAS Dataset Toolbar. This will display the LAS dataset's elevation values using a color-ramp. The resulting view is rendered as a TIN (Triangular Irregular Network), using this view, incorrectly classified points will be easier to see. The color-ramp can be modified so that certain elevation ranges have more color depth within the LAS dataset's properties. Use the beginning of the iterative process to repeat as necessary until the desired points have been appropriately classified.
This is the end of the iterative process.

According to Figure 3, the next task is to create the raster elevation model from the filtered LAS Dataset layer containing ground and building points. Determine the average point-spacing of the LiDAR data and choose the raster pixel size based on the previously discussed definition section. Use the *LAS Dataset to Raster* tool to extract a raster elevation model.[11] The parameters used to create the raster will depend on the LiDAR data's resolution. Use the recently created ground and building LiDAR layer as the input LAS Dataset, name the output raster and set the output location to the folder containing the location's data files. Use the *.tif file extension when naming. The value field is set to elevation and the interpolation type is set to binning. Binning is generally good enough when using LiDAR since the spacing between data points is generally small. If the LiDAR data happens to be very low density or low resolution, triangulation may be a better. With the binning option selected, the sub-options, Cell Assignment Type and Void Fill Method, are set to IDW and Linear respectively. These are the options that were used with greatest success in the development of this method. The next option is Output Data Type which should be set to FLOAT since integer values would cause a loss of accuracy in elevation values. Sampling type should be set to CELLSIZE; using this, the sampling value determines the size of a single pixel within the raster elevation model. The output file size and processing time increase as the sampling value decreases. Assuming all of the files imported into the map are in the correct coordinate systems, there should be no need to enter any values into the last option, Z Factor. This option is used to indicate any type of scaling that might need to occur due to a switch in linear units. The Z Factor will be left blank assuming all the units match the initial projection's units.

Running this tool will create a resulting raster elevation model that will be saved in the format *.tif. The raster creation tool creates a minimum bounding rectangle (MBR) for the bounding polygon of the given area. This process will create null values in the cells located beyond the polygon boundary due to the MBR. In order to use the raster within the WindNinja software package, these values must be replaced with a zero value instead of null. This process can be completed using two different methods depending on how the GIS Software tools interact with the raster. During the process of developing this method some glitches were encountered that required a different method of value replacement.

---

[11] LAS Dataset to Raster

3.2.1 Dealing with Null Values

The first method for replacing the null values in the raster is done using the *Raster Calculator* tool.[12] The algorithm for replacing the values uses a conditional statement. For each value within the raster, check to see if the value is null, if it is null, replace the value with "0", otherwise keep the current value. Within the Raster Calculator tool window, the expression for this algorithm is:

```
Con(IsNull("*.tif"),0,"*.tif")
```

Save the output raster as *_clean.tif within the same directory. After pressing 'OK' the process will complete and the resulting raster, *_clean.tif, is ready to be used within WindNinja. During the development of this method, the Raster Calculator tool began to throw an unknown error and would not complete the replacement. An alternative method is described next.

The second method for replacing the null values in the raster is accomplished by first converting the raster to an ASCII file and then manually replacing the null values with 0 using a text editor with find and replace functionality. Use the *Raster to ASCII* tool[13] to convert the raster elevation *.tif file to *.txt. Once the process is complete, this file can be opened in a text editor. It should be noted that if the file is extremely large (file size greater than 1 GB) it may require a special text editor to open. For the purposes of this guide, the text editor Notepad++ was used. The null values created from the LAS Dataset to Raster process have the value -9999 for the locations not defined within the bounding polygon. Simply do a find and replace to change the -9999 values to 0. Save the file and return to the GIS software to convert it back to a raster format. This can be accomplished with the *ASCII to Raster* tool.[14] Open the tool window and add the *.txt file as the input and save the output raster as *_clean.tif. Be sure to change the output data type to FLOAT so that none of the elevation data is lost in the conversion. The resulting output raster will need to be reprojected before it can be used within WindNinja. This can be done by right-clicking the newly created raster within the Catalog window and selecting the **Properties...** option. Inside the properties window, within the General tab, scroll to the Spatial Reference section and click the **Edit...** button. Choose the XY coordinate system that matches the project's coordinate system. The raster image can then be used within WindNinja to create output lookup tables.

---

[12] Raster Calculator

[13] Raster to ASCII

[14] ASCII to Raster

3.3 Configuring WindNinja for Batch Processing

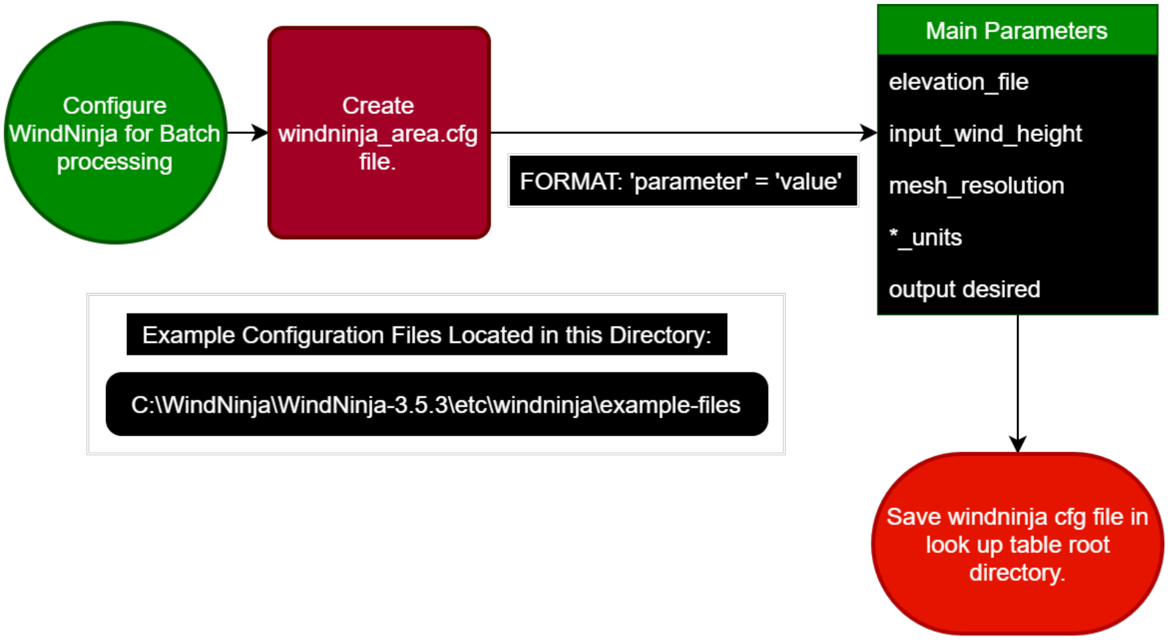


Figure 4: Configure WindNinja's parameters to desired output.

Now that the raster elevation model has been created, a WindNinja configuration file can be set up in order to run the table creation process via the Command Line. Using the configuration file and a batch script, the WindNinja shapefile output can be created iteratively using the boundary conditions defined earlier. These parameters are the raster elevation model, mesh size, initial wind speed, initial wind direction, input wind height, and output z-value for the shapefile. An example configuration file is included in the WindNinja installation directory on the local hard-drive.[15] To view all of the available configuration parameters, use the “*–help*” option with the command line executable WindNinja file[16] or use the developer's manual.[17]

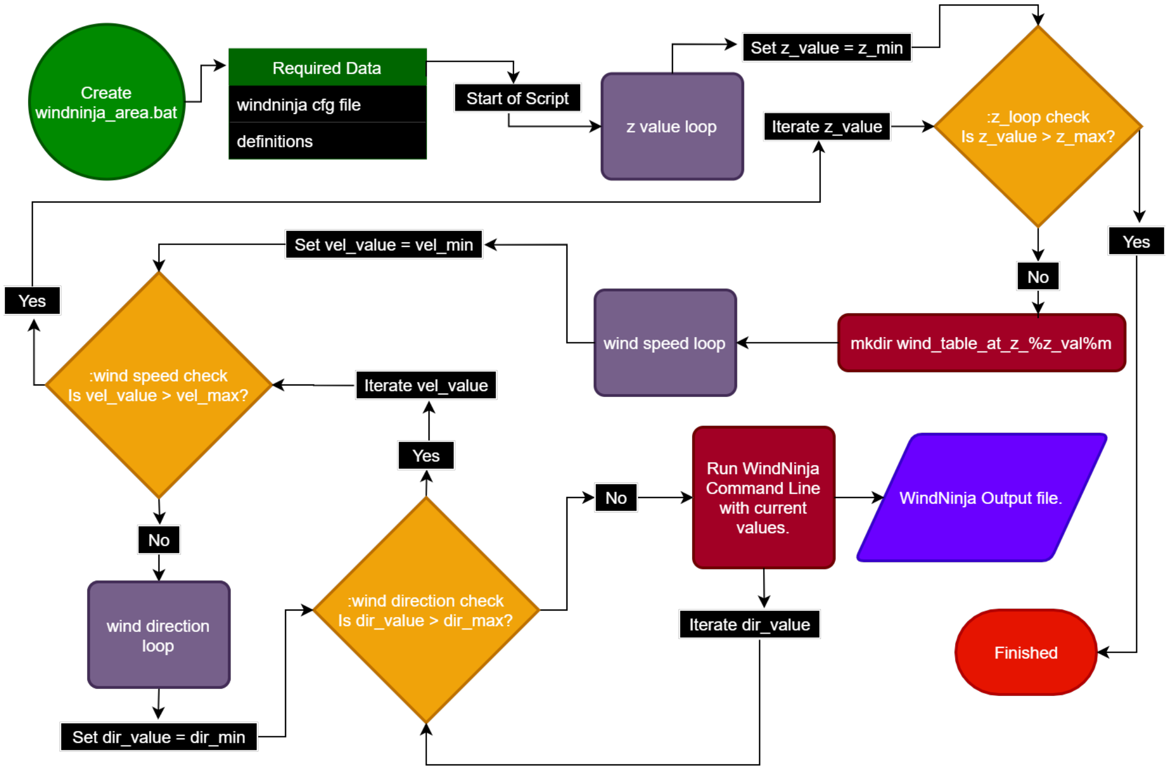
---

[15] C:/WindNinja/WindNinja-3.5.3/etc/windninja/example-files

[16] C:/WindNinja/WindNinja-3.5.3/bin/windninja_cli.exe

[17] WindNinja CLI Instructions

3.4 Batch Processing for WindNinja



Example WindNinja Command Line Execution using variables:

```
C:\WindNinja\WindNinja-3.5.3\bin\WindNinja_cli.exe wind_ninja_certain_1.cfg
--input_speed=%vel% --input_direction=%dir%
--output_wind_height=%z_val%
--output_path=C:\'path to look up tables'\wind_table_at_z_%z_val%m
```

Figure 5: Flow Chart for running WindNinja as a batch process.

After putting the necessary parameters into the configuration file, a windows batch script can be written in order to run WindNinja iteratively for the defined z-output, initial wind speed, and initial wind direction boundary conditions. The flow chart in Figure 5 describes this algorithm and provides an example of the instruction used to run an iteration of WindNinja using the command line interface.

3.5 Shape to CSV Script

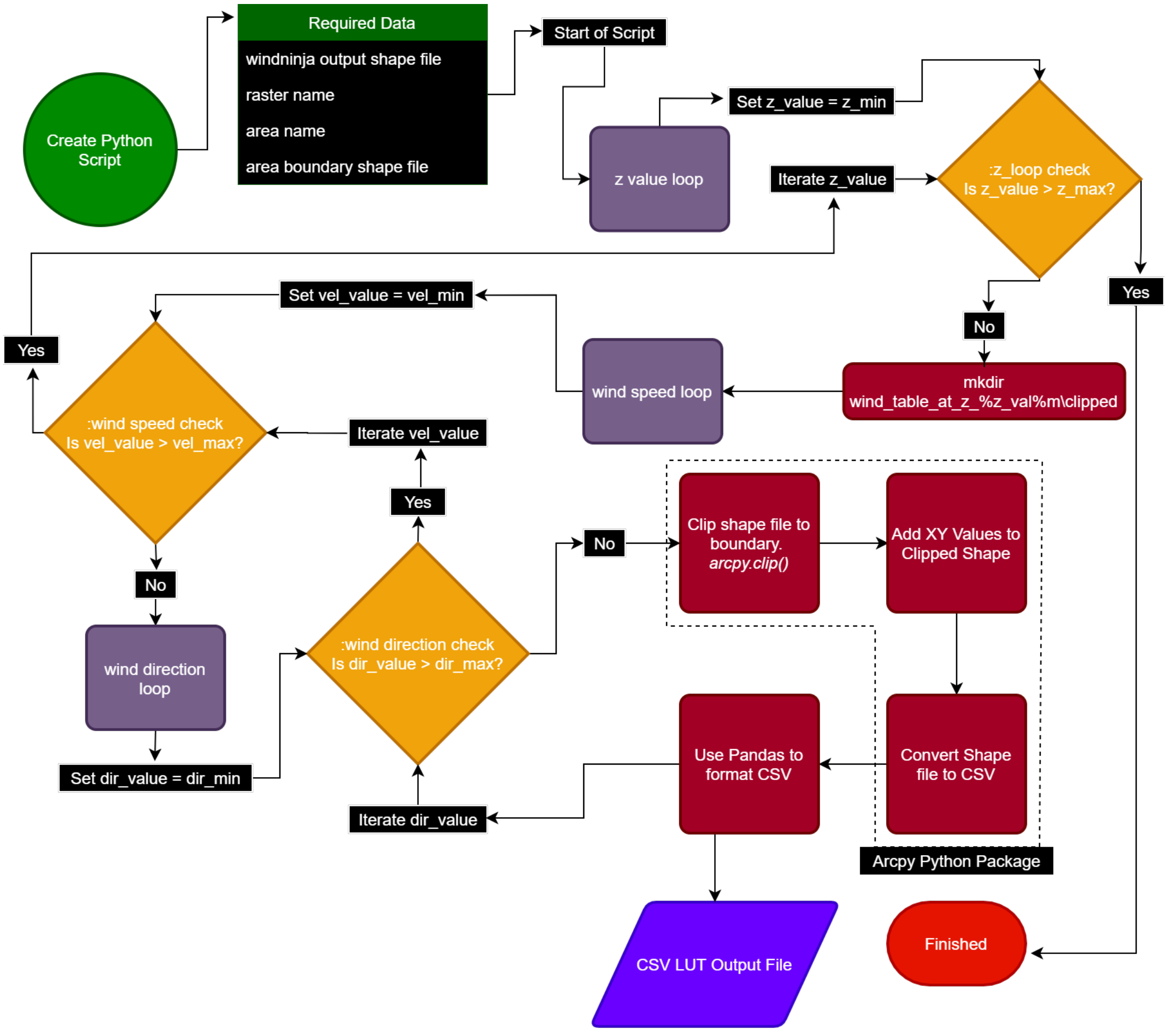


Figure 6: Flow Chart for a script that converts WindNinja output to usable csv file.

Once the batch process completes, there will be an associated folder for each of the z-values defined in the z-profile. The batch script used in the development of this method provides a total of 44 z-value folders as seen in the boundary condition definitions for the z-profile. Within each of these folders are the associated shapefiles for each of the 120 initial conditions for wind speed and wind direction outlined earlier in the Defining Locations and Boundary Conditions section. The resulting output shapefiles consist of an xy-plane with wind speed and wind direction for the given z-value output in the minimum bounding rectangle area of the raster elevation model. In order to create a more manageable data size for the final comma delimited lookup tables, the set of output shapefiles can be clipped to the AOI boundary polygon just as the LiDAR data was in the raster creation process. The GIS software tool *Clip*[18] can be used to trim shapefiles to a boundary polygon. The

---

[18] Clip

next task in creating the comma delimited table is extracting the x and y locations for each of the grid points in the tables to the desired output projection. This guide uses the *WGS84* projection since it uses degree decimal units corresponding to longitude in the x-direction and latitude in the y-direction. The *Add Geometry Attributes* [19] tool can append the x and y values to the shapefile's attribute table in a specified coordinate system. Finally, the *Export Feature Attribute*[20] tool can export a specified set of fields from the shapefile's attribute table and place them into an ASCII file with a specified delimiter. However, there is one issue in this process; the *Export Feature Attribute* tool appends two extra columns to the output ASCII file, these are the xy coordinates for the original projection of the shapefile. Using the *pandas*[21] python package, the two columns of xy positional data for the original projection can be removed.

---

[19] Add Geometry Attributes

[20] Export Feature Attribute

[21] pandas 0.24.2 documentation

4 Conclusion

Upon completion of the creation of the xy-plane lookup tables, it is important to note some of the key success and failure points within this methodology. Another relevant discussion is the process in which one would alter or change the method described in order to increase accuracy and optimize processing time. It should also be noted that there are many other software packages that can create numerical solutions for wind flow-fields. The method described in this guide is the most optimized approach known to the author in obtaining the resulting lookup tables at the time of development.

4.1 Assumptions

The assumptions made within this model will have an effect on the accuracy of the resulting tables. These assumptions are listed below.

1. The initial input flow-field wind height in known.
2. The incoming boundary layer wind flow is uniform.
3. Momentum is not conserved, only mass flow is conserved.
4. The Model does not contain trees or vegetation elements.
5. All structures extrude from the ground, no overhanging modeling.
6. No temperature effects are taken into account.
7. There exists a weather station for incoming flow-field measurements at height of the defined input wind.

4.2 Model Applicability

The main success of this model is that it has outlined a method for creating a set of lookup tables that can be used in UAS mission planning and during flight. These tables can be updated using other software that could provide a more accurate flow-field. Using this method, the final resulting tables can be incorporated into an algorithm for a UAS weather service for the end user's interface.

4.3 Caveats on Usage

The model has come up short in a few different areas that are notable. The first issue is that each of the wind tables include output wind speed and wind direction values in locations that are located within the building polygons. This information could be removed from the tables in future iterations; however, the addition of these points does not affect the user's ability to use the tables for a weather service system. Another issue is that the tables are static after their creation. As an example of why this is an issue, the tables would need to be completely remade if the location of interest has any changes in the existing buildings or terrain. Another thing that was noticed is that there are only 4 unique flow-fields for a given wind-speed. More specifically, for a given wind direction, the flow-field at 0 degrees matches that of 180 degrees only giving opposite direction values within the table.

4.4 Improvements

While there are specific improvements that could be implemented in the model, they are out of scope for this version of the User Guide. The WindNinja software package has an option to calculate the resulting flow-field given the initial conditions with either Mass or Mass and Momentum conservation. This guide describes a method that uses Mass conservation only. The processing time when using Mass and Momentum conservation is much greater than that of only Mass conservation. Including Momentum conservation would allow the software to predict areas of turbulence with much greater detail. It should also be noted that the table was created using whole number values for the Wind Speed and only eight cardinal directions. The process could be improved by including a higher resolution of wind speed and wind direction values into the process. There may also be a way to integrate local weather station measurements into the Wind Ninja process itself such that the initial conditions are set according to the weather station instead of running them iteratively. Including these weather stations in the WindNinja software could allow live table creation on the spot. There may also be a couple of methods for decreasing the amount of processing time required to create the resulting lookup tables. One method involves looking into the final process of the shapefile to CSV conversion. This process requires a redundant step in adding the XY values that could be improved by adjusting the built-in arcpy script *Export Feature Attribute*. More specifically, the adding and removing of the original coordinate system's xy values could be altered so that they are not added initially. It may also be possible to multi-thread the process for converting the output shapefiles to a comma delimited lookup table since it uses much less CPU and RAM resources than the initial WindNinja output processing.

4.5 Other Methods and Ideas

Some other ideas for creating the wind model were briefly explored in the development process. These methods include other wind modeling software applications and techniques. Listed below are some of the potential advantages and disadvantages for these methods.

QUIC-URB Software:
This software was developed by Los Alamos National labs for the purpose of creating a wind model for determining particle dispersion in urban environments. Some of the advantages of this software are that it allows import of shapefiles in order to create the urban model for the wind simulation. Given that there exists a shapefile for a set of physical objects in the area of interest, vegetation such as individual trees and forest areas can be defined to be used in the wind model creation. The disadvantages are that it does not account for the surface elevation, the surface is assumed to be flat. One of the biggest disadvantages of the software were in its

ability to output the required lookup tables via a batch process. Since the program uses a graphical user interface and is compiled as an executable, modifications to the output of the software were met with difficulties. However, this software could be useful in future wind modeling purposes.

Computational Fluid Dynamic (CFD) Model:
Using CFD to model the wind field is another method that was explored. Using a 3D model of the area, along with CFD software, highly accurate wind fields can be calculated. The main disadvantage of this method is the computer resources and time required to calculate and create the lookup tables. Calculating a single table using this method could take hours or days. It should also be noted that CFD modeling assumes a constant temperature for a single density fluid. The wind could be noticeably affected due to the gradient of the air's temperature and the physical composition of the atmosphere in a given location.