Arc Hydro: DEM from LAS Workflow and Tools

esri | THE SCIENCE OF WHERE™

Table of Contents

1.0 Introduction

Arc Hydro consists of a data model, toolset, and workflows developed over the years to support specific geographic information system (GIS) implementations in water resources. The initial implementation of Arc Hydro was in 2002 with the data model, Arc Hydro book published by Esri Press, and an initial set of about 30 tools. Since then, Arc Hydro has been used in many projects, and in the process, new tools and workflows have been developed. There are more than 300 Arc Hydro tools now, and they continue to be expanded based on work in specific implementations.

This document describes key steps for using Arc Hydro tools for creating a DEM from LAS points, according to the methodology developed by Dr. David E. James, USDA/ARS National Laboratory for Agriculture and the Environment, Ames, Iowa.

1.1 Document History

Table 1. Document Revision History

Version	Description	Date
1	First version (GLO)	11/2020
2	New parameter Buffer Distance for Clipping	4/2021

1.2 Recommended reading

This is one of several Arc Hydro "how to" documents. The following are the other suggested reading, in the order of importance:

1. Arc Hydro - Project Development Best Practices
2. Arc Hydro - ArcGIS Pro Project Startup Best Practices
3. Documentation on the ACPF method
4. Arc Hydro - Overview of Terrain Preprocessing Workflows
5. Arc Hydro – Identifying and Managing Sinks

2.0   Solution Overview

The process for creating a partially hydro-conditioned DEM from a LiDAR point cloud described here was developed by Dr. David E. James (see his Google Scholar page here). Several out-of-the-box tools are executed to complete the process. The basic steps are:

1. Build a terrain dataset from LAS data

    a. A new file geodatabase and child feature dataset are created. The feature dataset will store a multipoint feature class and a terrain dataset.

        i. Only class 2 (bare earth) points are loaded into the multipoint feature dataset. Point spacing is hardcoded to 1.

    b. If break lines and/or polygons are provided, these are copied into the feature dataset.

    c. The terrain dataset is created.

        i. 3D points are added to the terrain.

        ii. Pyramid levels are added to the terrain.

        iii. If break lines and/or polygons were provided, they are added to the terrain.

    d. The terrain surface is generated.

2. Create a DEM raster from the terrain dataset

    a. If a limiting polygon is provided, it will be used to clip the output DEM.

        i. When a limiting polygon is provided, an optional parameter "Buffer Distance for Clipping" is available. The buffer distance entered will be applied to the limiting polygon before it is used to clip the DEM. Including a buffer can be useful for avoiding edge effects during subsequent DEM processing steps.

    b. DEM float raster is created with the user-specified resolution and Natural Neighbors interpolation method.

    c. Optionally, create an integer version of the DEM by scaling the DEM values by 100 (conversion from meters to centimeters).

    d. Create a DEM without 1-cell sinks:

        i. Identify "real" sinks using the Sinks tool,

ii.   Replace sink locations with No Data,

iii.   Fill any sinks outside of the No Data values (i.e., 1-cell sinks), and

iv.   Replace the No Data values with the data from the original DEM raster.

If the LiDAR point cloud is packaged in the LAZ format, users will need to complete a perquisite step to convert these to LAS format, using the out-of-the-box tool, Convert LAS.
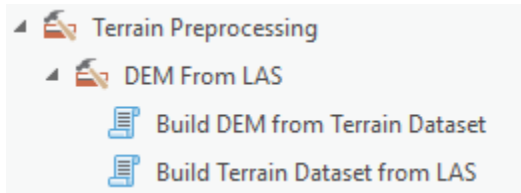

Figure 1. Arc Hydro "DEM From LAS" toolset (Arc Hydro Tools Python toolbox).

The tools discussed in this document are in the Arc Hydro Tools Pro toolbox supporting ArcGIS Pro 2.6 and later.

2.1   Overview of Arc Hydro tools involved in LAS to DEM Conversion

This section presents a list of tools used in the conversion of LAS files to a DEM raster. Tools in this section are presented in the order they appear in the Arc Hydro Tools Python Toolbox and do not represent the order in which tools are used. Not all the tools are used in all situations. The subsequent sections will explore specific workflows and which tools are used for them.

Toolset	Step	Tool	Description
(Core Software Tool) Conversion Tools / LAS	Convert LAZ files to LAS	Convert LAS	This is an out-of-the-box tool that converts LAZ files found in directory to LAS files.
Terrain Preprocessing / DEM From LAS	Build DEM from Terrain Dataset	Build DEM from Terrain Dataset	Creates a DEM raster from the terrain dataset generated in the previous step. Optionally, an integer DEM raster can be created by scaling the units by 100. The choice of cell size (1, 2, 3, 4, or 5m) defines the level of generalization applied based on terrain dataset pyramid levels.
	Build Terrain Dataset from LAS	Build Terrain Dataset from LAS	Creates the new file geodatabase and the child feature dataset that will store the LAS data. Points are imported to the feature

|			dataset from the LAS files. Optionally, 3D polylines and 3D polygons will be added to the feature dataset. Finally, the terrain dataset is built from the imported features.

3.0 Use Cases

3.1 Use Case 1: Creating a DEM without beak lines or break polygons

1. Convert LAS (Core software tool)
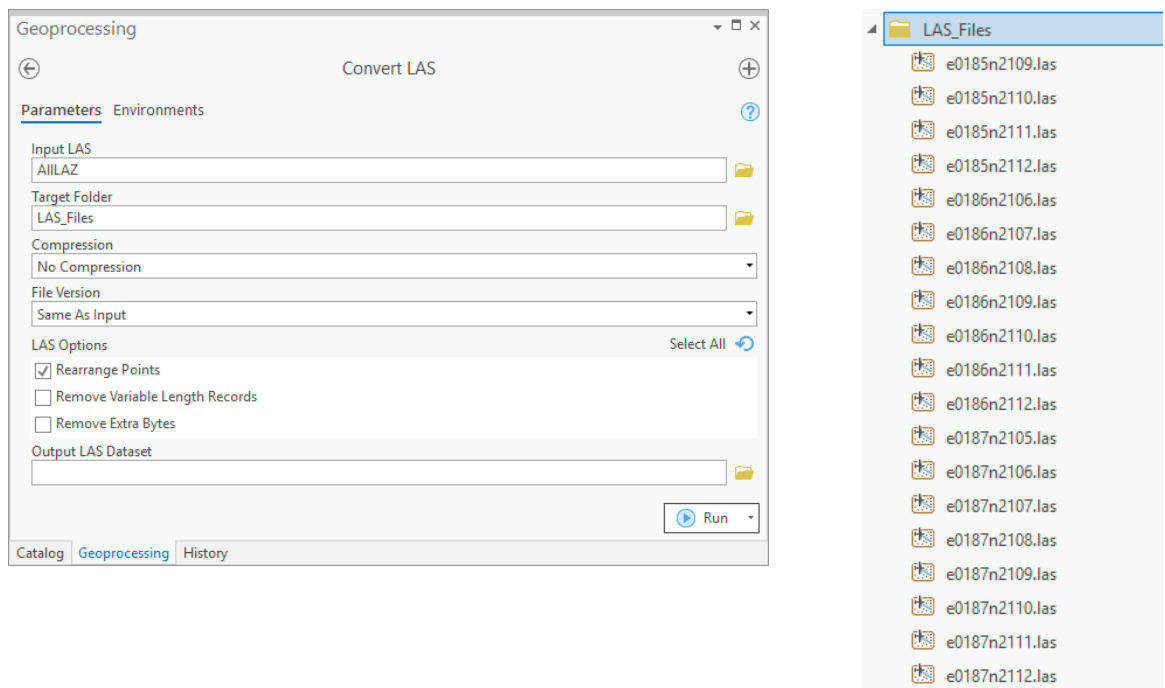
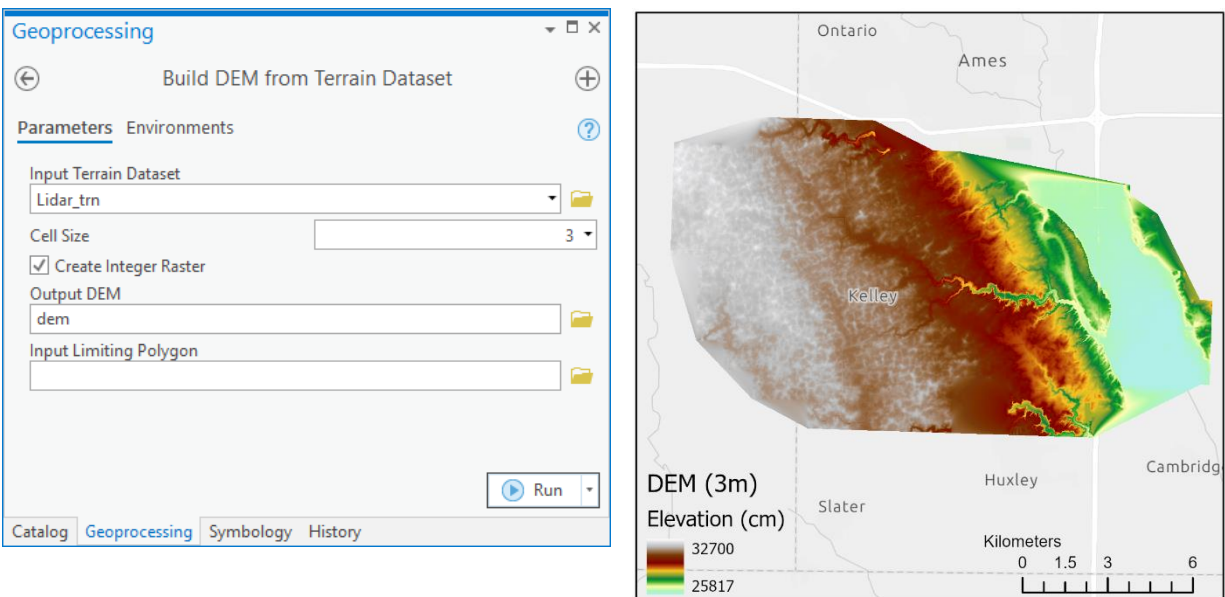This step is not necessary if the input files are already in LAS format.

2. Build Terrain Dataset from LAS
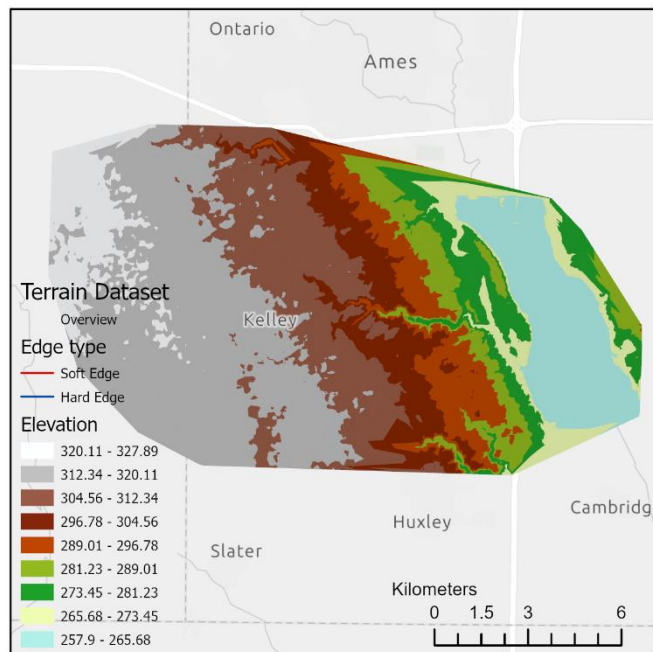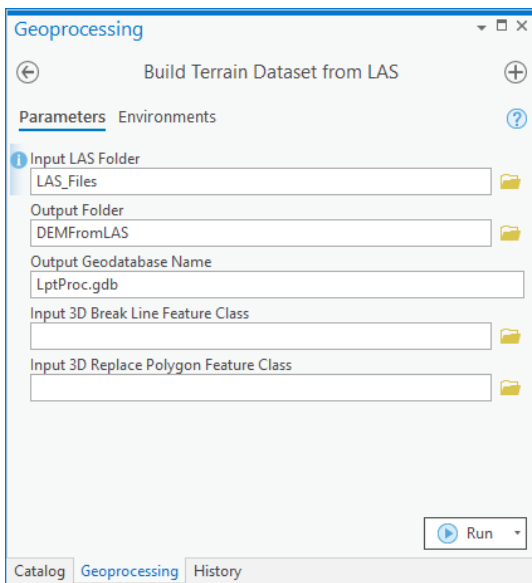
From Python:

```
import buildterraindatasetfromlas
build_terrain = buildterraindatasetfromlas.BuildTerrainDatasetfromLAS()
build_terrain.executeAH(r'C:\Documents\ArcGIS\Projects\DEMFromLAS\LAS_Files',
r'C:\Documents\ArcGIS\Projects\DEMFromLASPy', 'LptProc.gdb', None, None)
```

3. Build DEM from Terrain Dataset

Note that the DEM created is of integer type (centimeter vertical units). **Users should keep in mind that the tools does *not* alter the projection file related to the raster.** Users will have to manually update the proejction information for the output raster to reflect the change to vertical units of centimeters.

From Python:

```
import builddemfromterraindataset
build_dem = builddemfromterraindataset.BuildDEMfromTerrainDataset()
build_terrain.executeAH('Lidar_trn', 3, False,
r'C:\Documents\ArcGIS\Projects\DEMFromLASPy\dem', None)
```
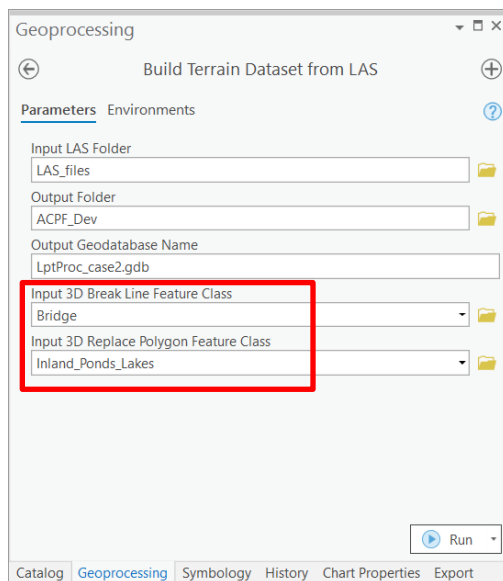
To include a limiting polygon:

```
build_terrain.executeAH('Lidar_trn', 3, False,
r'C:\Documents\ArcGIS\Projects\DEMFromLASPy\dem', 'HUC_boundary', '500 Meters')
```

3.2. Use Case 2: Creating a DEM with beak lines and break polygons

1. Convert LAS (Core software tool) using the same process as shown in <u>Use Case 1</u>.

2. Build Terrain Dataset from LAS

Since the majority of the output terriain dataset will look identical to that in Use Case 1, the example output above focuses on the locations of the break features. The top right panel depicts the break polyline, used to enforce elevation underlying a bridge. The bottom right panel depics the break polygons, used to enforce the elevation underlying inland ponds and lakes. In the left panels, the break polyline and break polygons are shown in the terrain dataset as hard edges (blue lines).

From Python:

```
import buildterraindatasetfromlas
build_terrain = buildterraindatasetfromlas.BuildTerrainDatasetfromLAS()
build_terrain.executeAH(r'C:\Documents\ArcGIS\Projects\DEMFromLAS\LAS_Files',
r'C:\Documents\ArcGIS\Projects\DEMFromLASPy', 'LptProc_case2.gdb', 'Bridge',
'Inland_Ponds_Lakes')
```

3. Build DEM from Terrain Dataset using the same process as shown in Use Case 1.
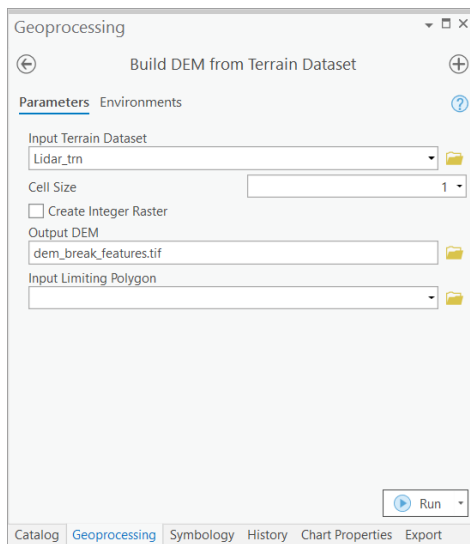
Notice that in this example, a floating point (meter) DEM raster was created. The example output above focuses on the locations of the break features. The left panels show how constant elevation values were forced along the break polyline and within the break polygons.

From Python:

```
import builddemfromterraindataset
build_dem = builddemfromterraindataset.BuildDEMfromTerrainDataset()
build_terrain.executeAH('Lidar_trn', 1, False,
r'C:\Documents\ArcGIS\Projects\DEMFromLASPy\dem_break_features.tif', None)
```

4.0 Appendix

4.1 Processing Times

The information in this appendix is provided for reference only. The actual performance will vary greatly depending on the size and complexity of the underlying input data and function parameters. The numbers provided here are to be used as a relative measure of performance – which functions are "faster" and which are "slower". Remember that raster processing tends to be non-linear for large rasters and that doubling the size of the raster will usually result in more than doubling the time it takes to process it.

The hardware used for processing will also have an impact on the performance. Fast hard-drive (HD) subsystem (preferably SSD) can double the improvement in performance compared to traditional HD. Make sure that both the scratch and permanent storage are pointing to the fast HD. For other processing suggestions, refer to "Arc Hydro - Project Development Best Practices" document.

The following results were collected using a machine with 16 GB of RAM. The largest extent of the testing dataset, located in the southwest Iowa, included:

- 153 LAS files

- ~1 million points

- Derived DEM raster (3m resolution) with 6000 x 4000 cells

Tool	Runtime
Convert LAS	1 minute, 30 seconds
Build DEM from Terrain Dataset (integer output)	5 minutes, 30 seconds
Build DEM from Terrain Dataset (float output)	4 minutes, 30 seconds
Build Terrain Dataset from LAS	4 minutes, 30 seconds