

ArcGIS 3D Analyst: Lidar Classification and Feature Extraction

Lindsay Weitz

Khalid Duri

Clayton Crawford



esri

THE
SCIENCE
OF
WHERE®

2021 ESRI USER CONFERENCE

Workshop Overview

- Lidar basics
- The LAS dataset
- Rule-based classification techniques
- Deep learning classification techniques
- Manual classification techniques
- Feature extraction from lidar

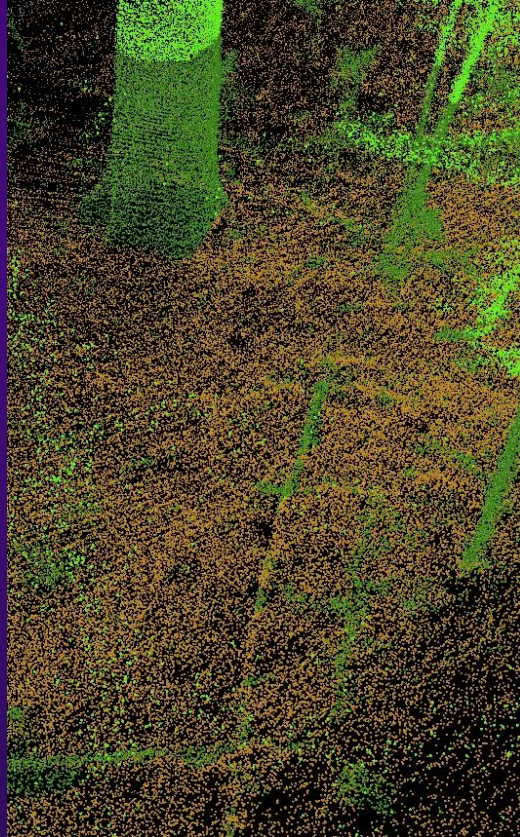


ArcGIS Supports

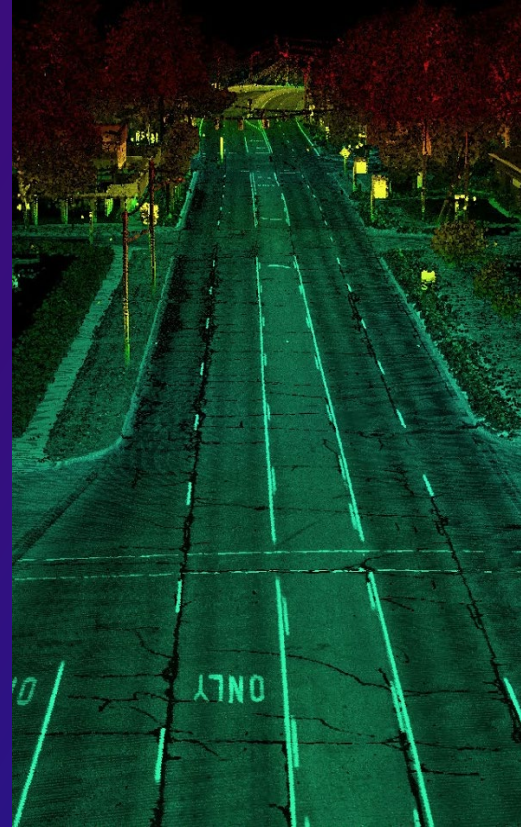
Airborne



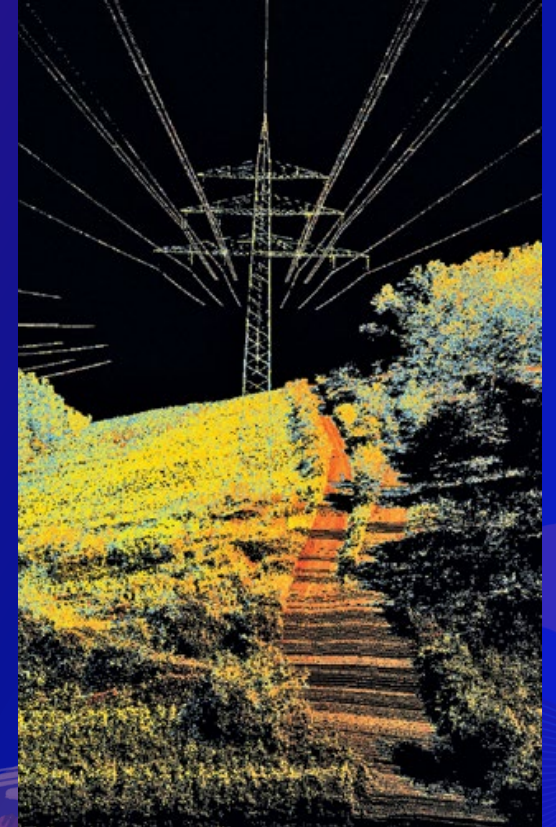
Terrestrial



Mobile



Drone/UAV

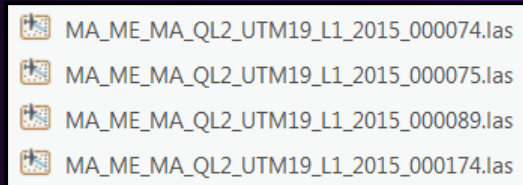
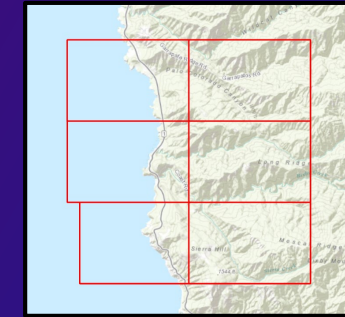
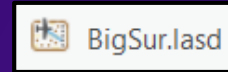


Data formats and management

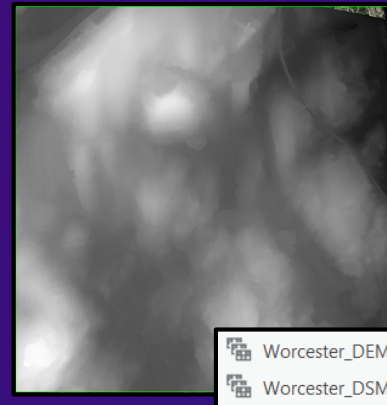
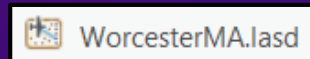
- Formats
 - LAS / zLAS / LAZ
- Management



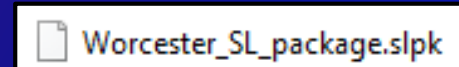
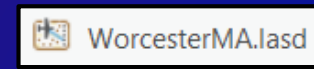
LAS Dataset



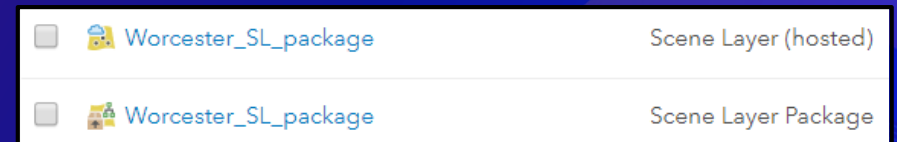
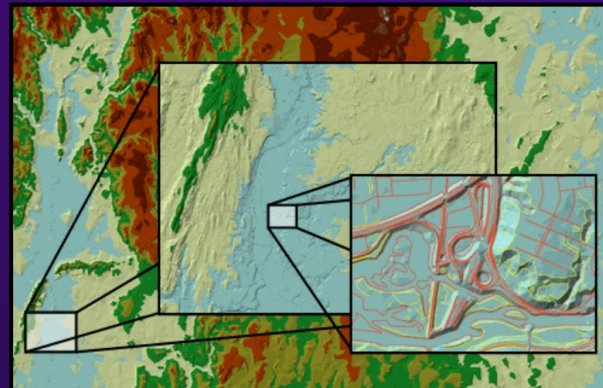
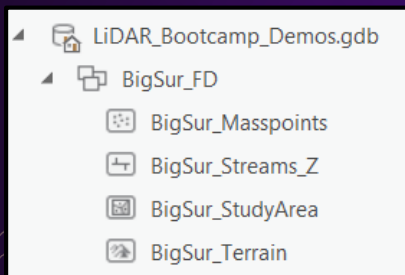
Mosaic Dataset



Point Cloud Scene Layer

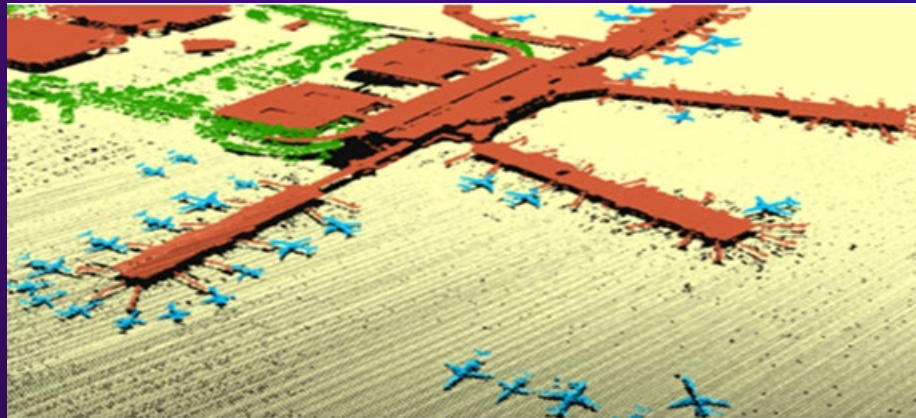


Terrain Dataset



Lidar data with a LAS dataset

- Direct read of LAS or zLAS format lidar
- File based
- QA/QC tools
- Stores references to LAS/zLAS files on disk
- Optionally reference breakline and control point data
- Treats a collection of LAS/zLAS files as one logical dataset (“Project”)



QA/QC: LAS Dataset Properties

LAS Dataset Properties: LAS Dataset.lasd

General

Summary

Name: LAS Dataset
LAS Files: 16 (16 LAS files, 0 zLAS files)
SurfaceConstraints: 0
LAS Points: 157,486,819
Data Size: 4,205.37 MB
Uncompressed Size: 4,205.37 MB [Calculate Size](#)
☐ Store relative path names to data sources

Extent

	Minimum	Maximum
X	6275000.01	6295000
Y	1835000	1854999.99
Z	-2828.1	3550.59

XY Linear Unit: Foot_US
Z Unit: Foot_US

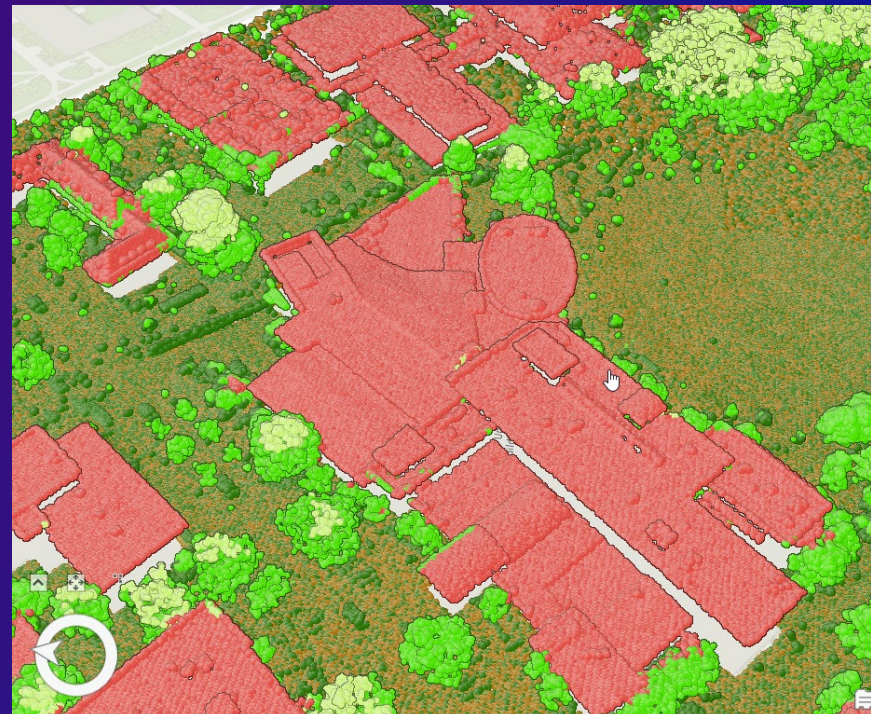
Context Menu:

- Add To Current Map
- Add To New Map
- Add To New Scene
- View Metadata
- Properties**

Classification codes

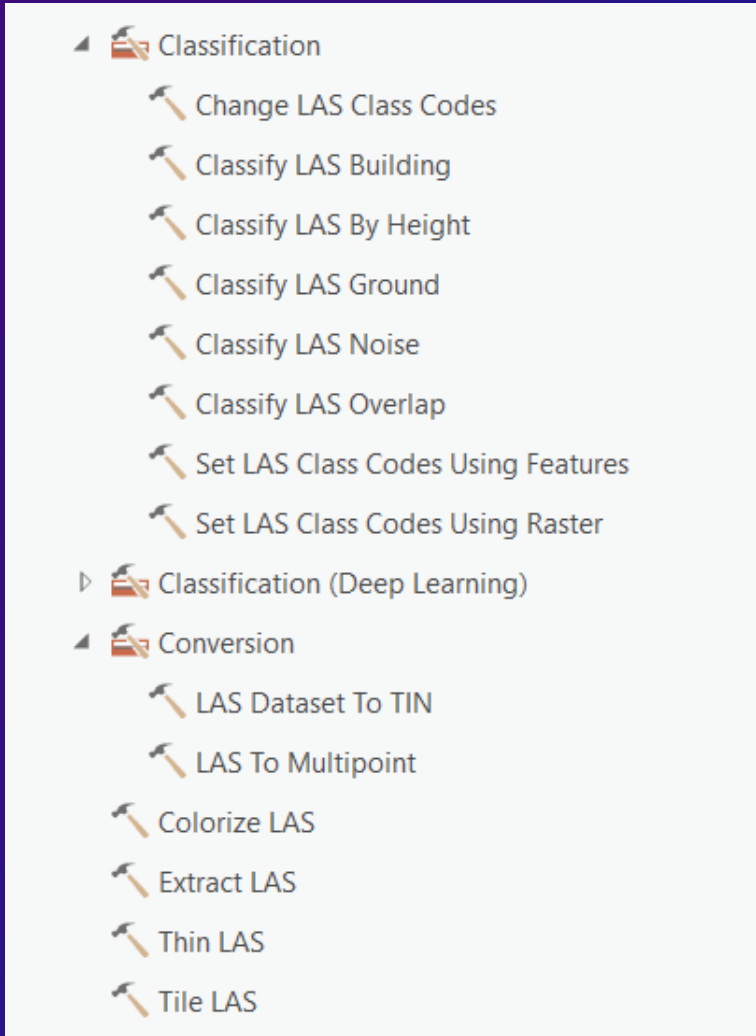
- Every lidar point can have a classification that defines the type of object that has reflected the laser pulse.
- Lidar points can be classified into a number of categories.
- The different classes are defined using numeric integer codes in the LAS files.

Classification	
0	Never Classified
1	Unassigned
2	Ground
3	Low Vegetation
4	Medium Vegetation
5	High Vegetation
6	Building
7	Noise
8	Model Key / Reserved
9	Water
10	Rail
11	Road Surface
12	Overlap / Reserved
13	Wire - Guard
14	Wire - Conductor
15	Transmission Tower
16	Wire - Connector
17	Bridge Deck
18	High Noise

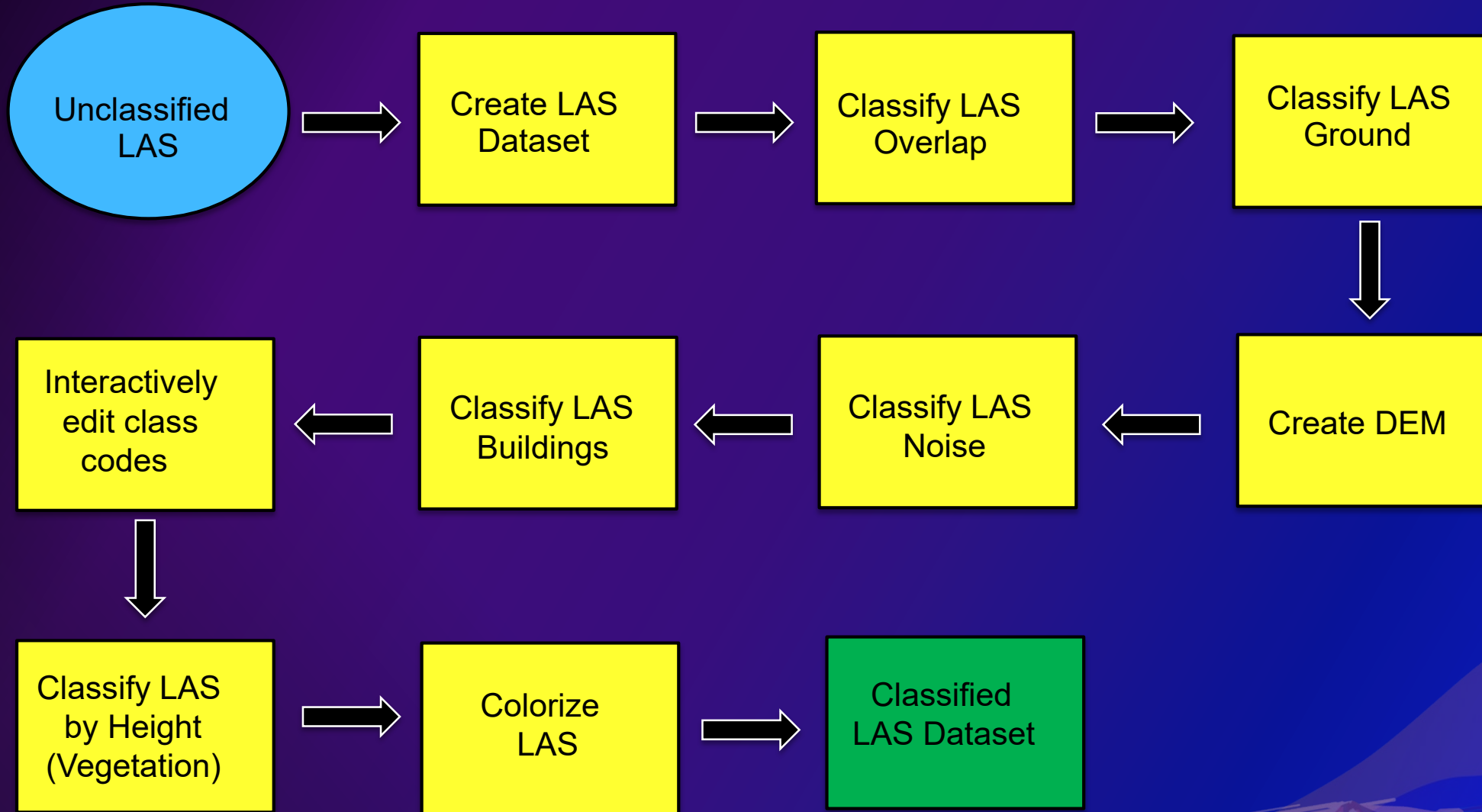


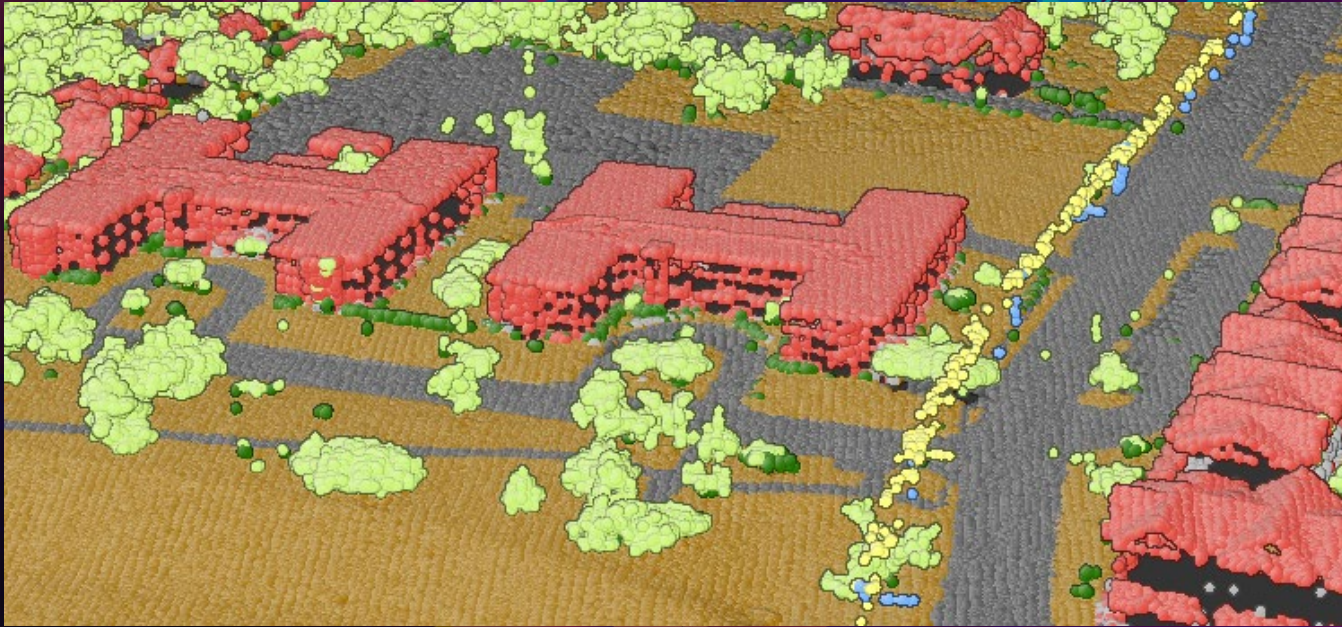
Tools for managing lidar

- **Extract LAS**
- **Tile LAS**
- **Thin LAS**
- **Colorize LAS**
- **Convert LAS**
- **Set LAS Class Codes Using Features**
- **Set LAS Class Codes Using Raster**

- 
- A screenshot of a software application's menu for lidar management. The menu is organized into three main sections: Classification, Classification (Deep Learning), and Conversion. Each section contains several tool options, each preceded by a small icon of a hammer and a flag. The background of the menu is white, and the text is black. The icons are small and stylized.
- Classification
 - Change LAS Class Codes
 - Classify LAS Building
 - Classify LAS By Height
 - Classify LAS Ground
 - Classify LAS Noise
 - Classify LAS Overlap
 - Set LAS Class Codes Using Features
 - Set LAS Class Codes Using Raster
 - Classification (Deep Learning)
 - Conversion
 - LAS Dataset To TIN
 - LAS To Multipoint
 - Colorize LAS
 - Extract LAS
 - Thin LAS
 - Tile LAS


Example Workflow Classifying and Processing Lidar





Rule-Based Classification

Lindsay Weitz

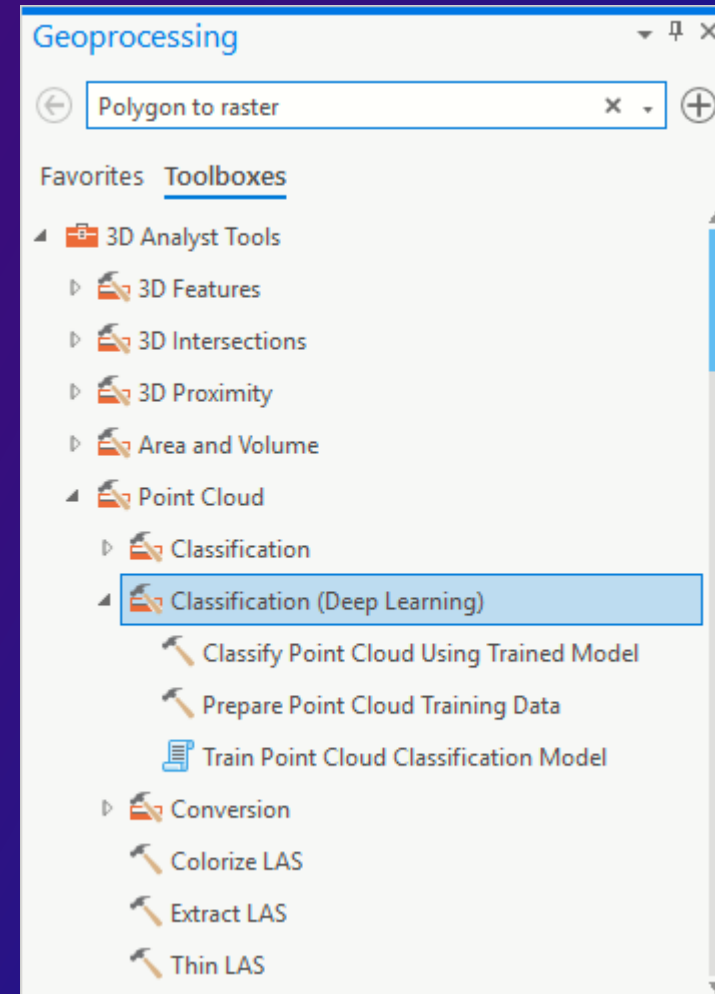


Deep Learning for Point Cloud Classification

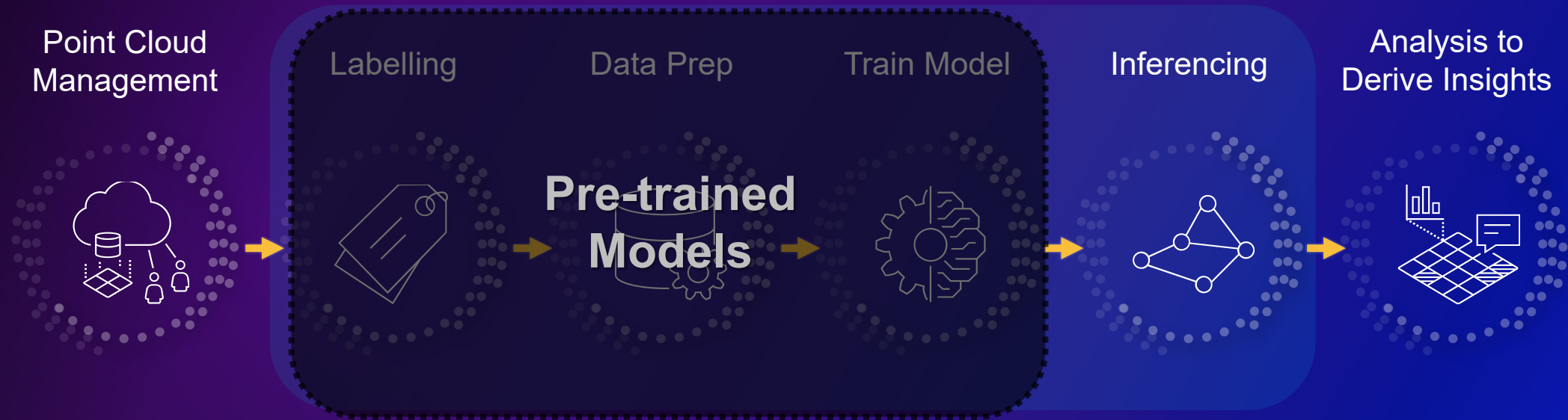
Overview of Basic Requirements & Best Practices

Getting Started with Deep Learning for Point Cloud Classification

- ArcGIS Pro 2.8
- 3D Analyst extension
- ArcGIS Deep Learning Framework
 - NVIDIA GPU (recommended)
 - CUDA Compute Capability of 6.0
 - 8 GB dedicated RAM

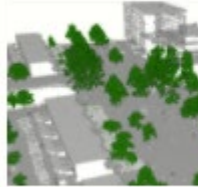


Overview of Deep Learning Workflow

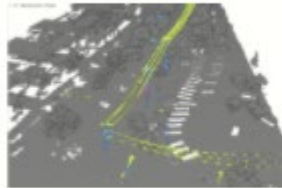



- Eliminates:
 - Need for extensive labeling
 - Need for extensive training
 - Massive compute requirements

Tree Point Classification



Power Line Classification

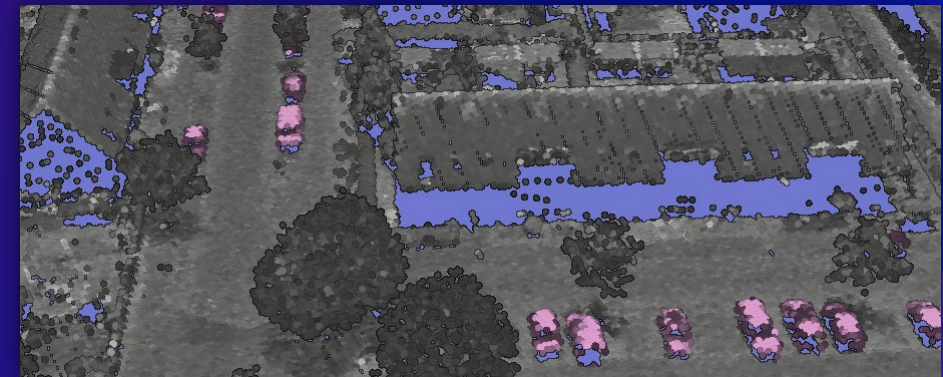
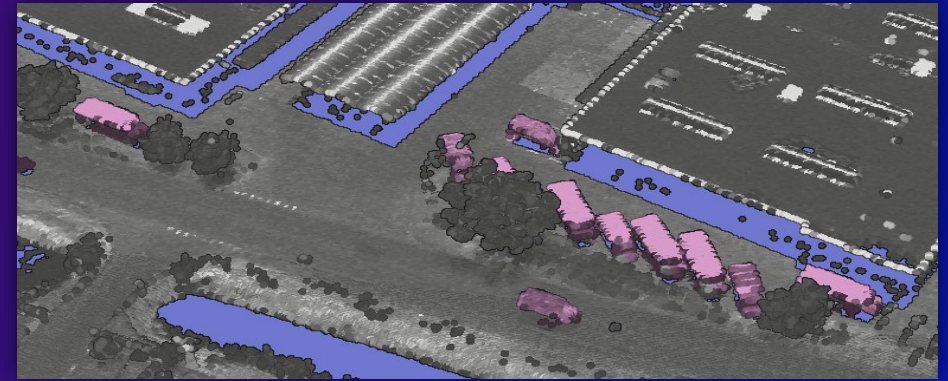


 Deep Learning Package By [esri_analytics](#)

Deep learning model to classify wire conductors, distribution towers, and wire structure connectors from a point cloud dataset. This model is targeted for distribution wires.

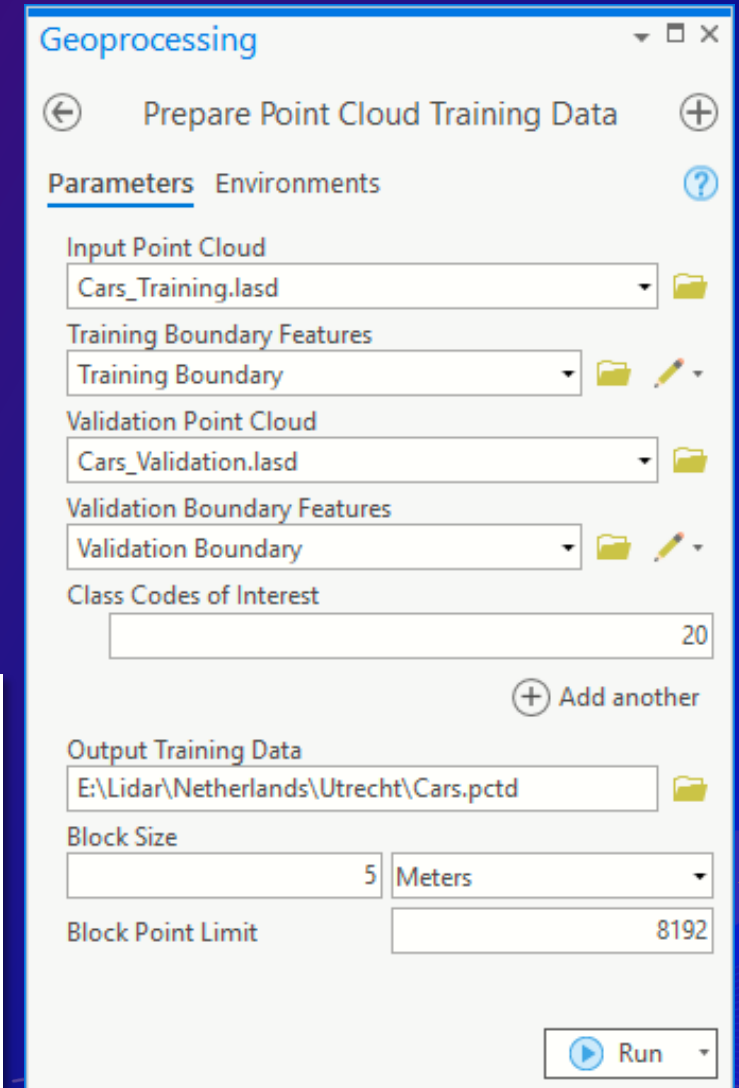
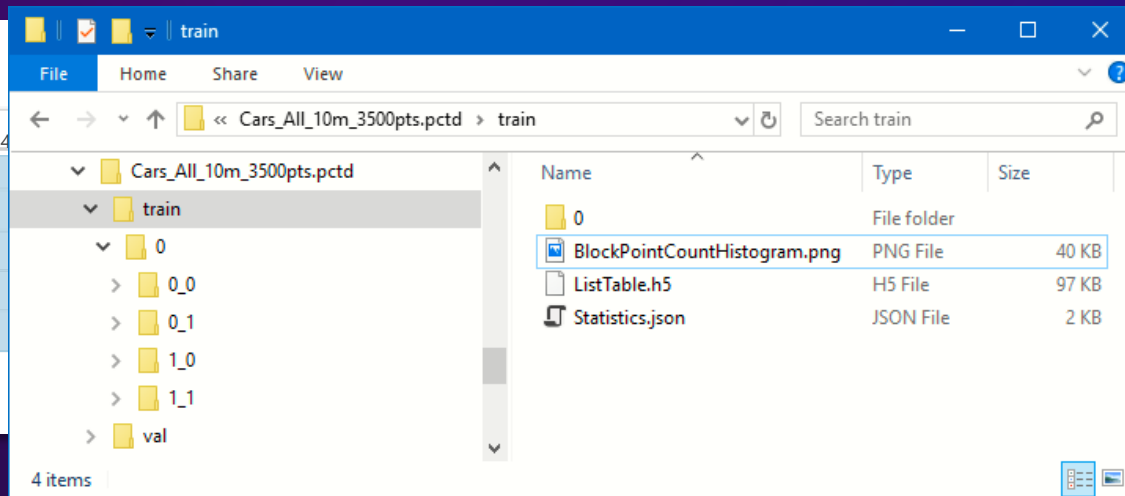
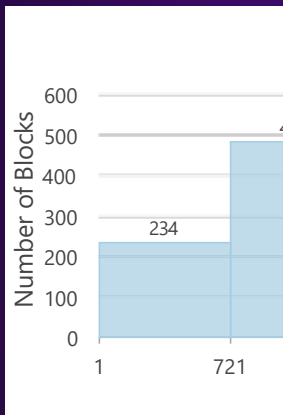
Understanding the Training Dataset

- Training dataset comprised of:
 - Data used for training the model
 - Data used for validating the model being trained
- Objects of interest must be:
 - Accurately classified
 - Captured in varied representations
- Point cloud should reflect the properties of data the model will be used on
 - Point density
 - Type of collection (e.g. terrestrial scan vs aerial scan)
 - Attributes (RGB, intensity, return number, etc...)



Creating the Training Dataset

- Source point cloud converted to overlapping “**blocks**”
- Free storage space should be 8x size of point cloud
- Block size should capture object of interest
- Block point limit helps manage GPU memory usage
 - Limit should minimize blocks with over-flow
 - “**LAS Point Statistics As Raster**” can provide preliminary estimate of points per desired block size



Training the Classification Model: Key Terms

- **Epoch:** The number of times the training data is processed.
- **Iteration:** The instances within an epoch when training data is processed
- **Batch Size:** The number of blocks processed in each iteration
- **Checkpoint:** The model weights at the end of the epoch
- **Training Loss:** A measure of the error when the model trained in an epoch is applied to the training data
- **Validation Loss:** A measure of the error when the model trained in an epoch is applied to the validation data

Check out the “Deep Dive into Deep Learning” workshop for more information.



Training the Classification Model

- Purpose of training is to develop and iteratively refine model weights through multiple epochs
- Classes can be remapped to simplify training
- Model metrics are reported per epoch to help understand how the model is performing
 - **Overfitting:** Model's training loss is low while validation loss is high, indicating it is not able to generalize sufficiently from the training data to properly classify new data.
 - **Underfitting:** Model's training loss is high, indicating it is not able to learn from the training data.

Epoch	Training Loss	Validation Loss	Accuracy	Precision	Recall	F1-Score	Time
0	1.11217	1.6989	0.60814	0.33033	0.2836	0.25811	00:11:00
1	1.76852	1.51276	0.70874	0.4513	0.38471	0.36054	00:11:04
2	0.627585	2.45031	0.74518	0.52373	0.46515	0.43134	00:11:04
3	0.605376	0.816826	0.74196	0.54228	0.49684	0.45464	00:11:04
4	1.07499	0.724953	0.74472	0.58966	0.49751	0.45075	00:11:04

Geoprocessing

←

Train Point Cloud Classification Model

+

Parameters

Environments

?

Input Training Data

D:\Training_Data\Utrecht_Cars\Cars_All_10m_3500p

Pre-trained Model

Attribute Selection

Intensity

Output Model Location

Utrecht

Output Model Name

Automobile_Classification

Minimum Points Per Block

0

> Manage Classes

▼ Training Parameters

Model Selection Criteria

Recall

Maximum Number of Epochs

25

Iterations Per Epoch (%)

100

Learning Rate

Batch Size

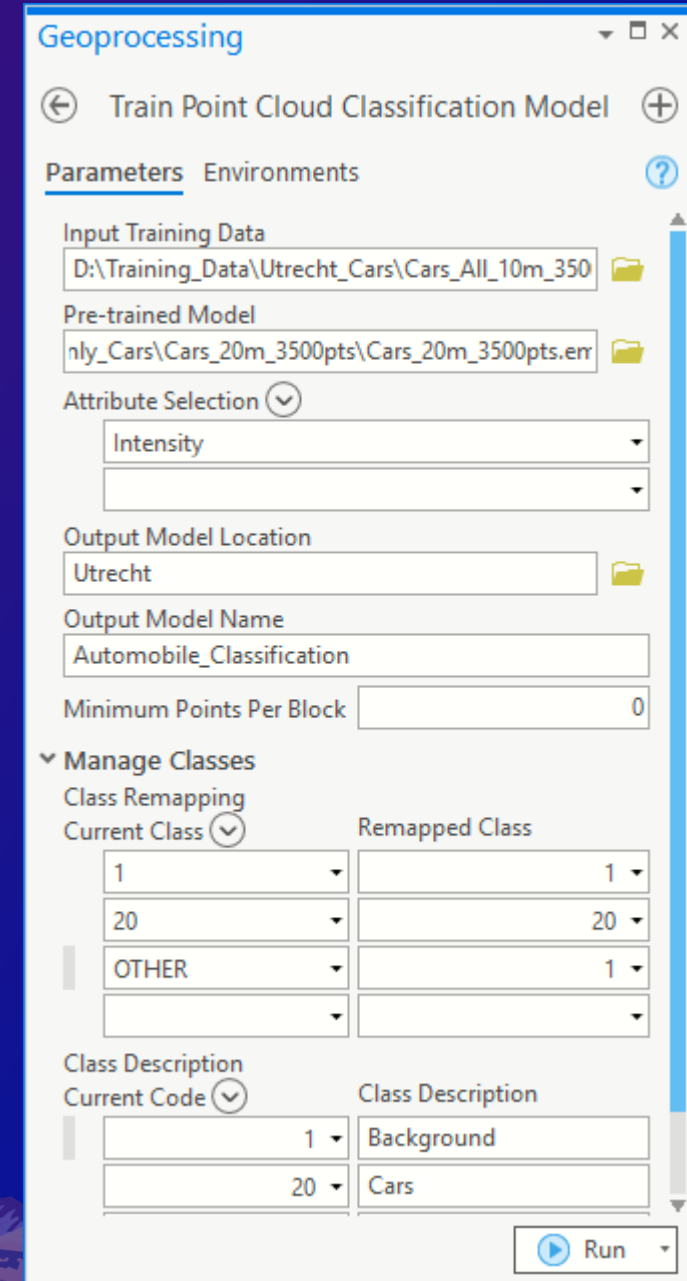
2

☒ Stop training when model no longer improves

Run

Training the Classification Model: Transfer Learning

- Use model weights from a pretrained model to create a more refined model with less effort
- Training data must match attributes used in the pretrained model
- Training data must match class codes in pretrained model. Classes that differ must be remapped to those in pretrained model.



The screenshot shows the 'Train Point Cloud Classification Model' tool in the Geoprocessing environment. The tool is configured with the following parameters:

- Input Training Data:** D:\Training_Data\Utrecht_Cars\Cars_All_10m_350
- Pre-trained Model:** nly_Cars\Cars_20m_3500pts\Cars_20m_3500pts.ern
- Attribute Selection:** Intensity
- Output Model Location:** Utrecht
- Output Model Name:** Automobile_Classification
- Minimum Points Per Block:** 0

The **Manage Classes** section is expanded, showing the **Class Remapping** table:

Current Class	Remapped Class
1	1
20	20
OTHER	1

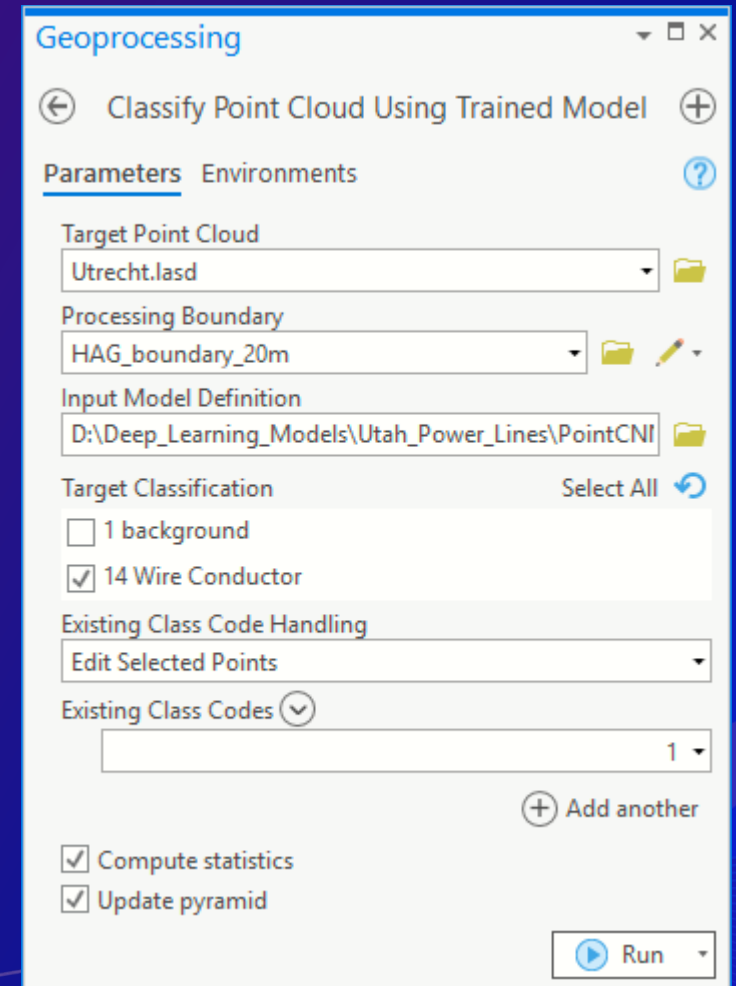
Below the remapping table is the **Class Description** table:

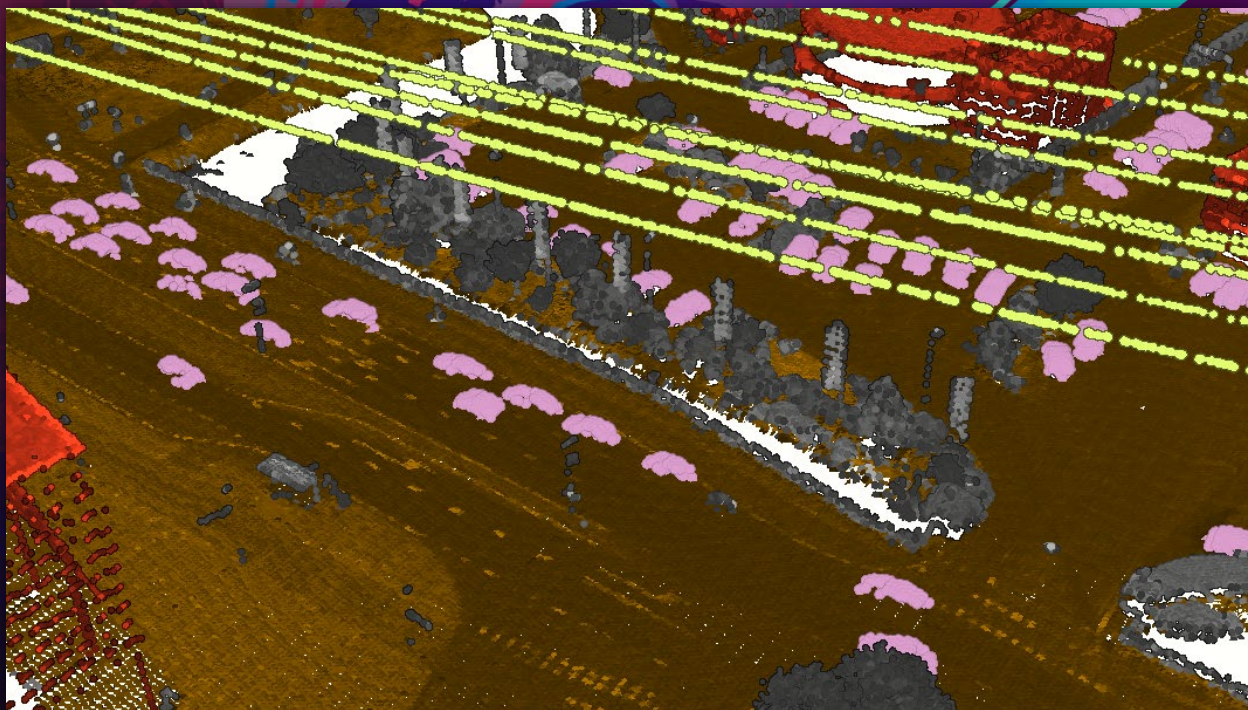
Current Code	Class Description
1	Background
20	Cars

A **Run** button is located at the bottom right of the tool interface.

Classifying a Point Cloud Using a Trained Model

- Target point cloud must have attributes used by pretrained model
- Best results obtained when point cloud has similar characteristics to training data
 - Scan type (terrestrial vs aerial scan)
 - Originating source (lidar vs photogrammetric)
 - Average point spacing & coverage of objects
 - Level of noise/error
- Use boundary polygon to limit extent of data being classified whenever possible





Classifying Transmission Power Lines

Khalid Duri

Manual Classification

The background is a vibrant, abstract composition. It features a deep purple base color. Overlaid on this are several organic, wavy shapes in shades of bright pink and magenta, primarily concentrated in the top-left and bottom-right corners. A thin, bright yellow arc curves from the middle-right towards the top-right. In the bottom-right corner, there is a stylized, high-contrast illustration of a mountain range or a rugged coastline in white and light orange. The overall aesthetic is modern and digital.

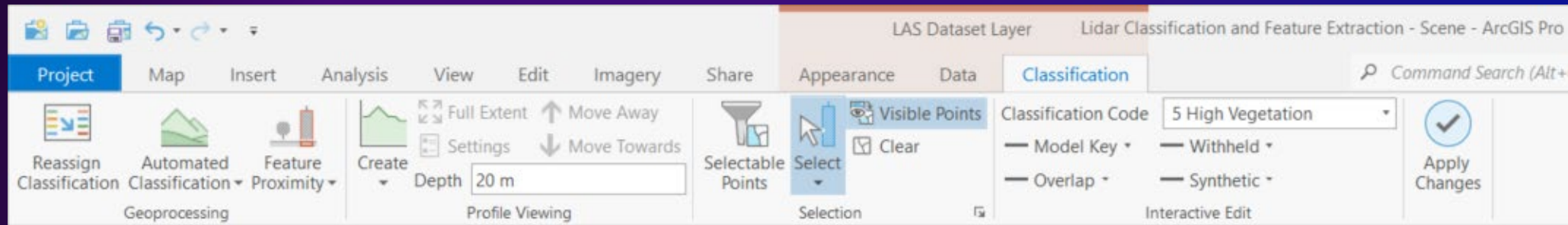
Manual Classification: Purpose

- To have fun 😊
- Born of necessity
- Clean up after automated classification
- Create training data for deep learning



Manual Classification: Ribbon and Tools

Classification tab for LAS dataset layer



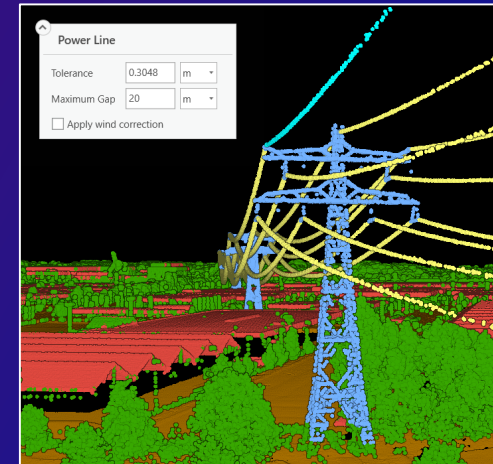
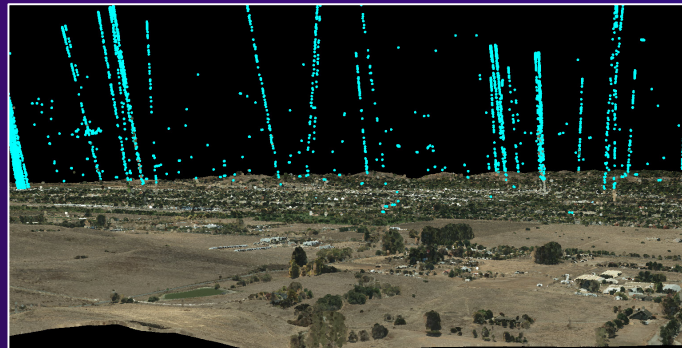
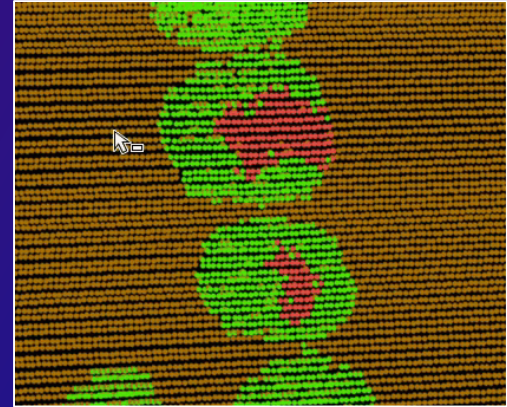
1. Select points of interest

2. Set desired class code and flag settings

3. Apply changes

Display Environment: Map vs Scene

- Map
 - 2D only
 - No profile view
 - Need to be zoomed to 100% point display
- Scene
 - Can operate in 2D and 3D
 - More selection tools and options
 - Profile view

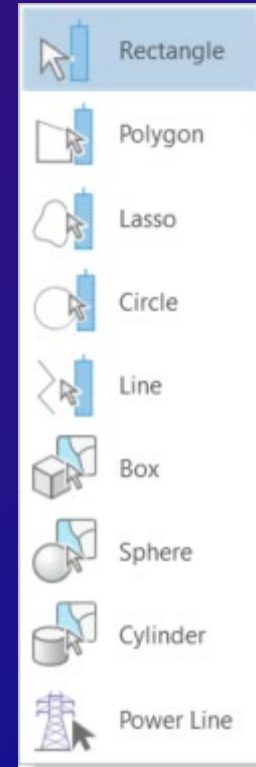


Selection Tools

Basic primitives
(maps and scenes)

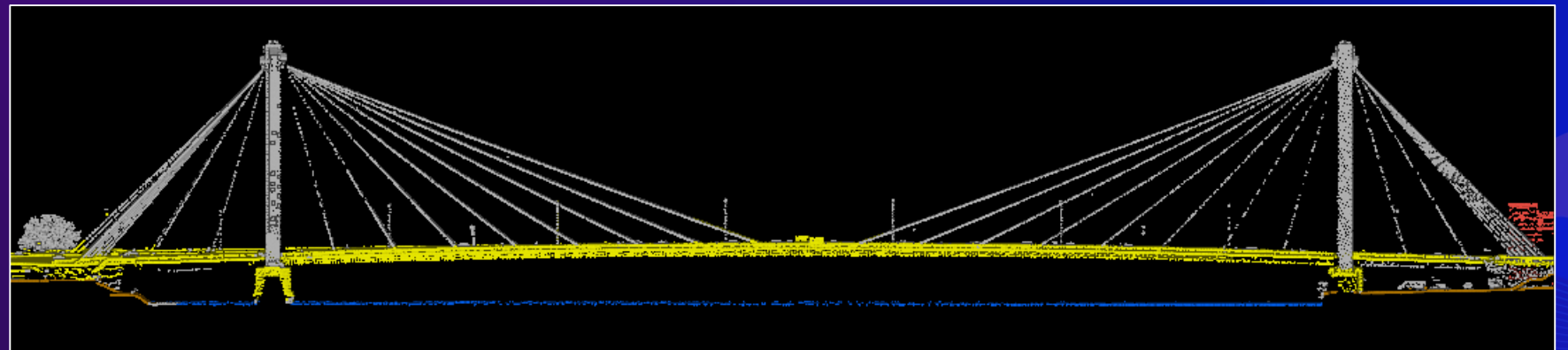
Volumetric
(scenes only)

Feature specific
(scenes only)



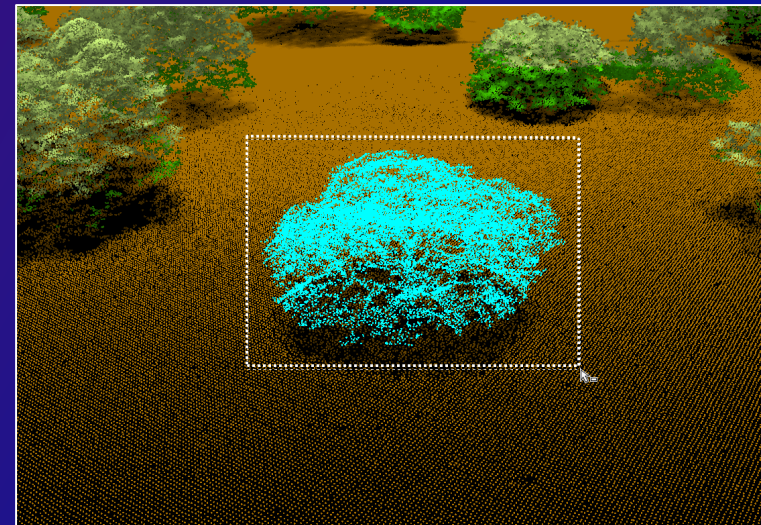
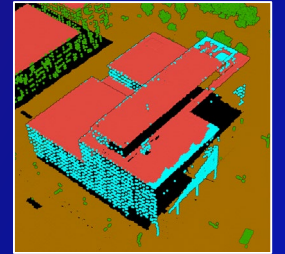
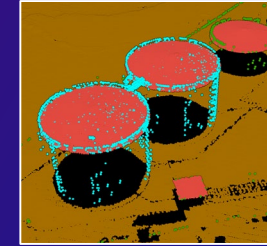
Profile View

- Slice of data
- All points within slice are displayed (no thinning)
- Only in 3D scene
 - It's implemented as a scene display mode rather than its own type of view
- Other data in scene included
 - Benchmark points
 - Breaklines



Selection Qualifiers

- **Selectable class codes**
 - Controls which points can be selected based on their class code(s) rather than just their visibility
- **Visible points toggle**
 - Enables selection of only points which are visible vs. all that are within selection graphic regardless of whether they are visible or obscured by other points





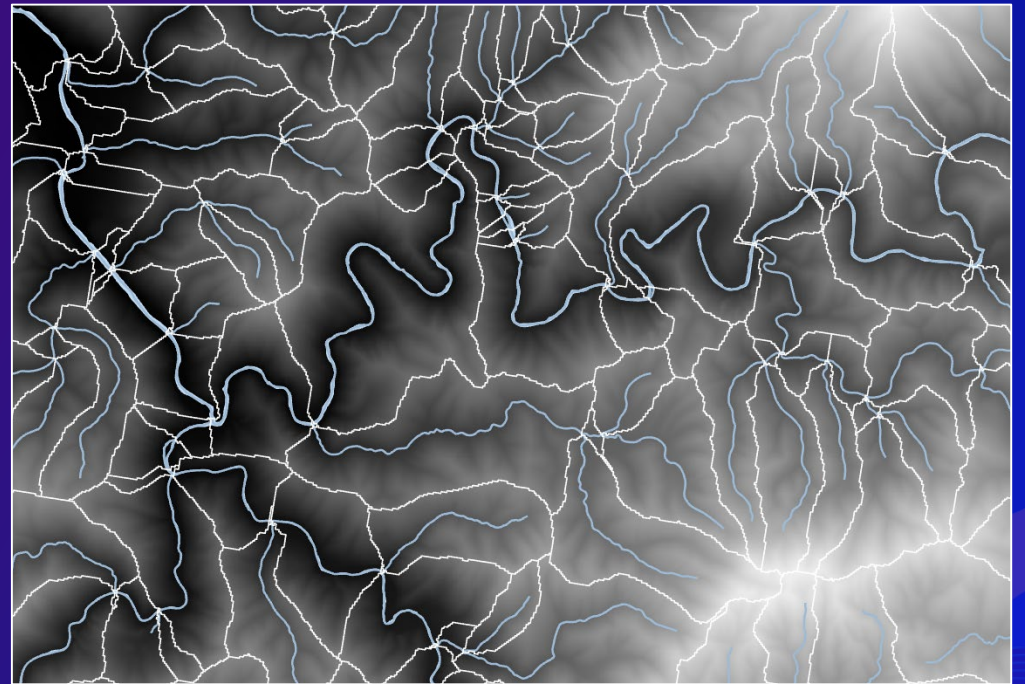
Demo

Feature Extraction

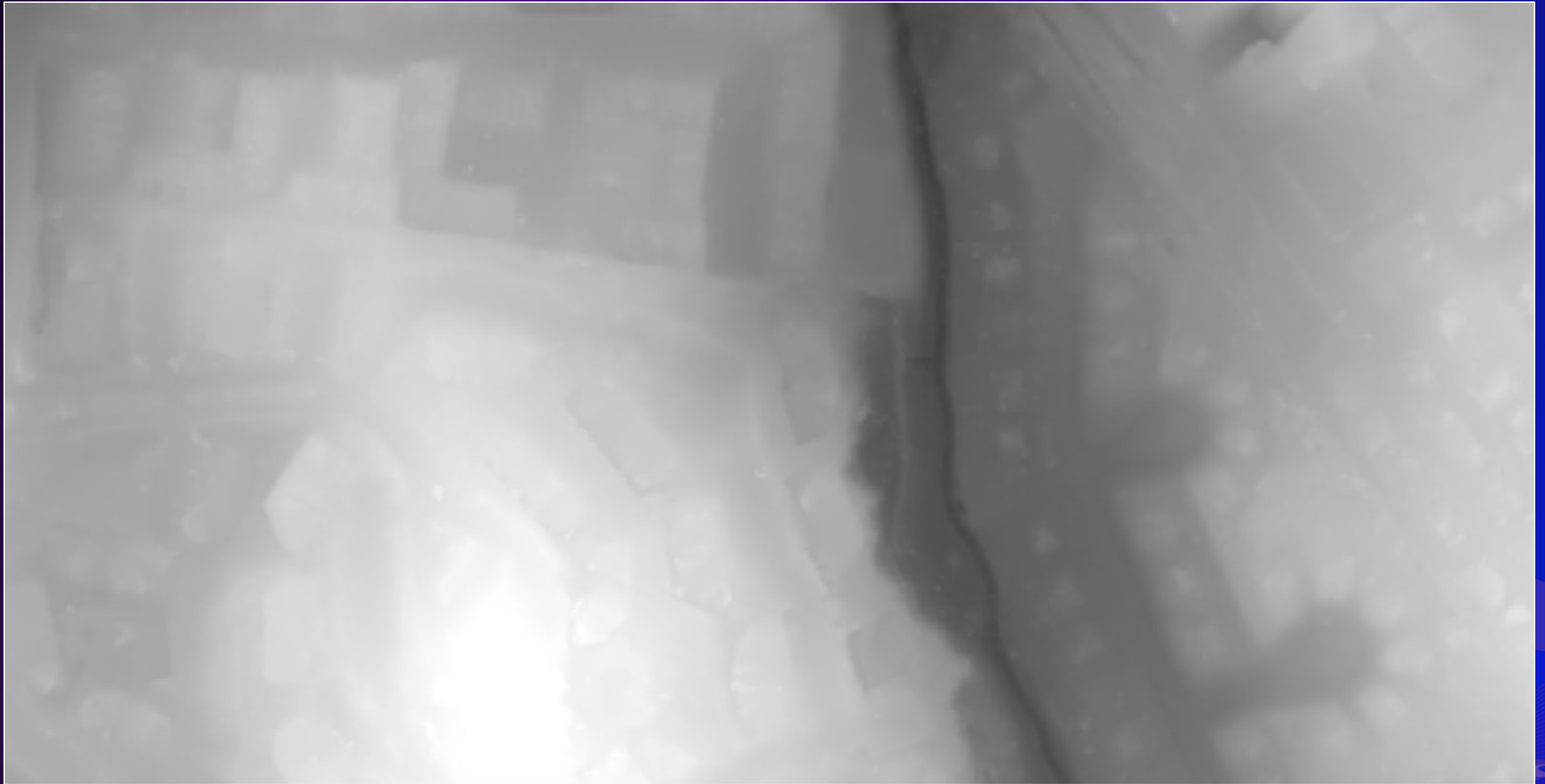
The background is a vibrant, abstract composition. It features a deep purple base color. In the top-left corner, there are layered, wavy shapes in shades of pink, magenta, and orange. A thin, bright yellow arc curves from the right side towards the center. On the right side, there are more complex, layered shapes in orange and pink, some with darker, almost black, internal details that resemble a stylized landscape or topographical map. The overall effect is modern and digital.

Surface Based Derivatives

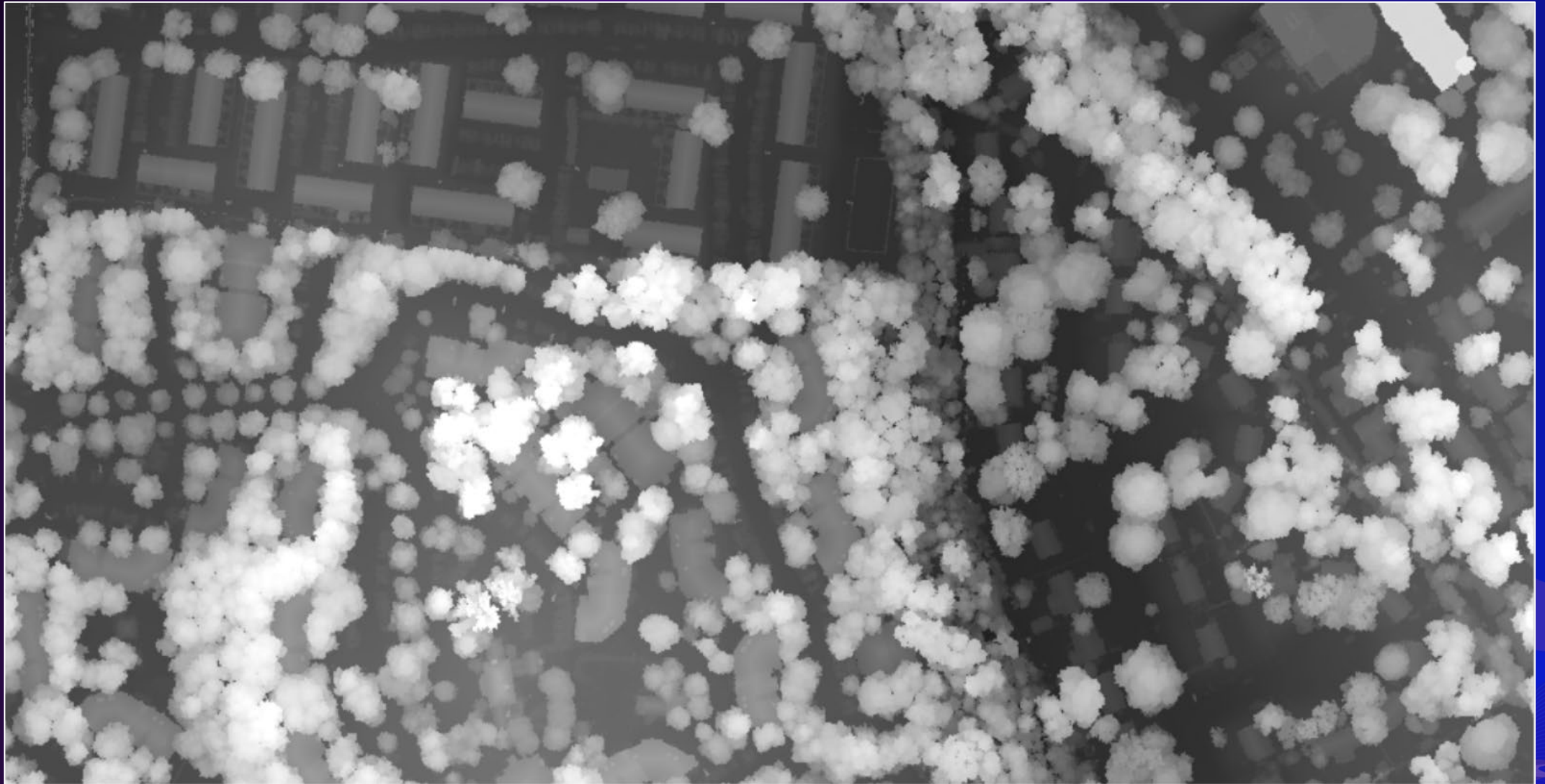
- **DTMs / DEMs and DSMs**
 - Can provide height info for vector feature data
 - Raster based feature extraction techniques
 - Surface derived features:
 - Contours
 - Watershed boundaries
 - Stream networks
 - Floodplain boundaries
 - Etc.
- **Normalized height models**
 - nDSM, CHM
 - Height relative to ground



DEM / DTM : Ground



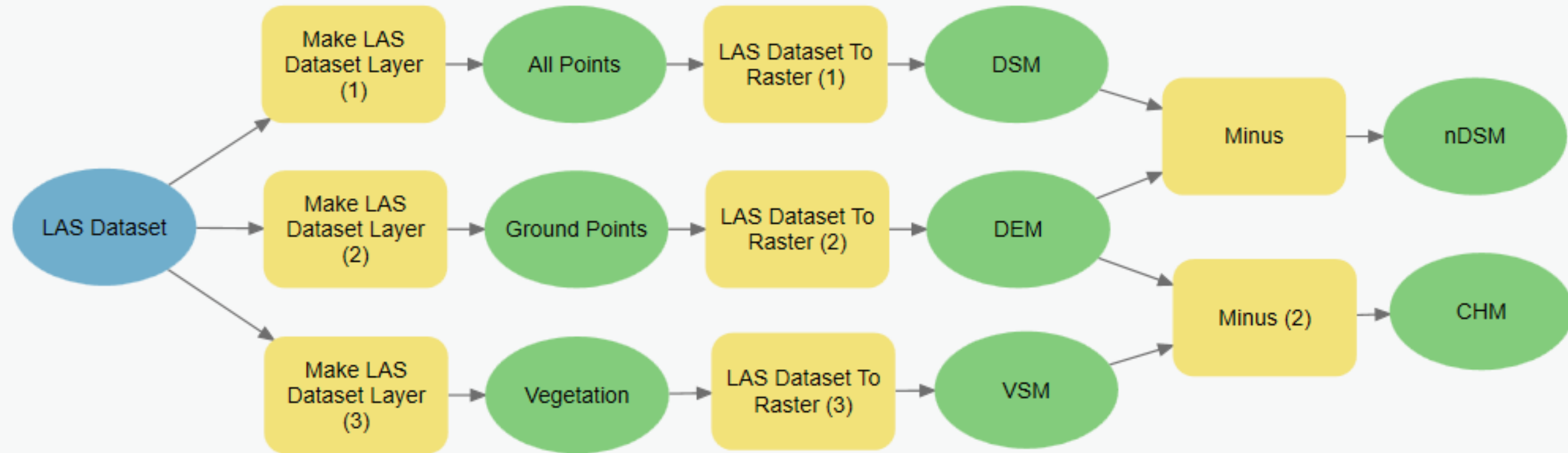
DSM : Ground + Buildings + Vegetation



CHM : Vegetation / Canopy Height

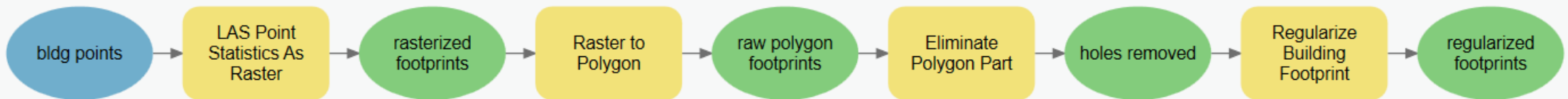


Surface Creation – Automation with Model Builder



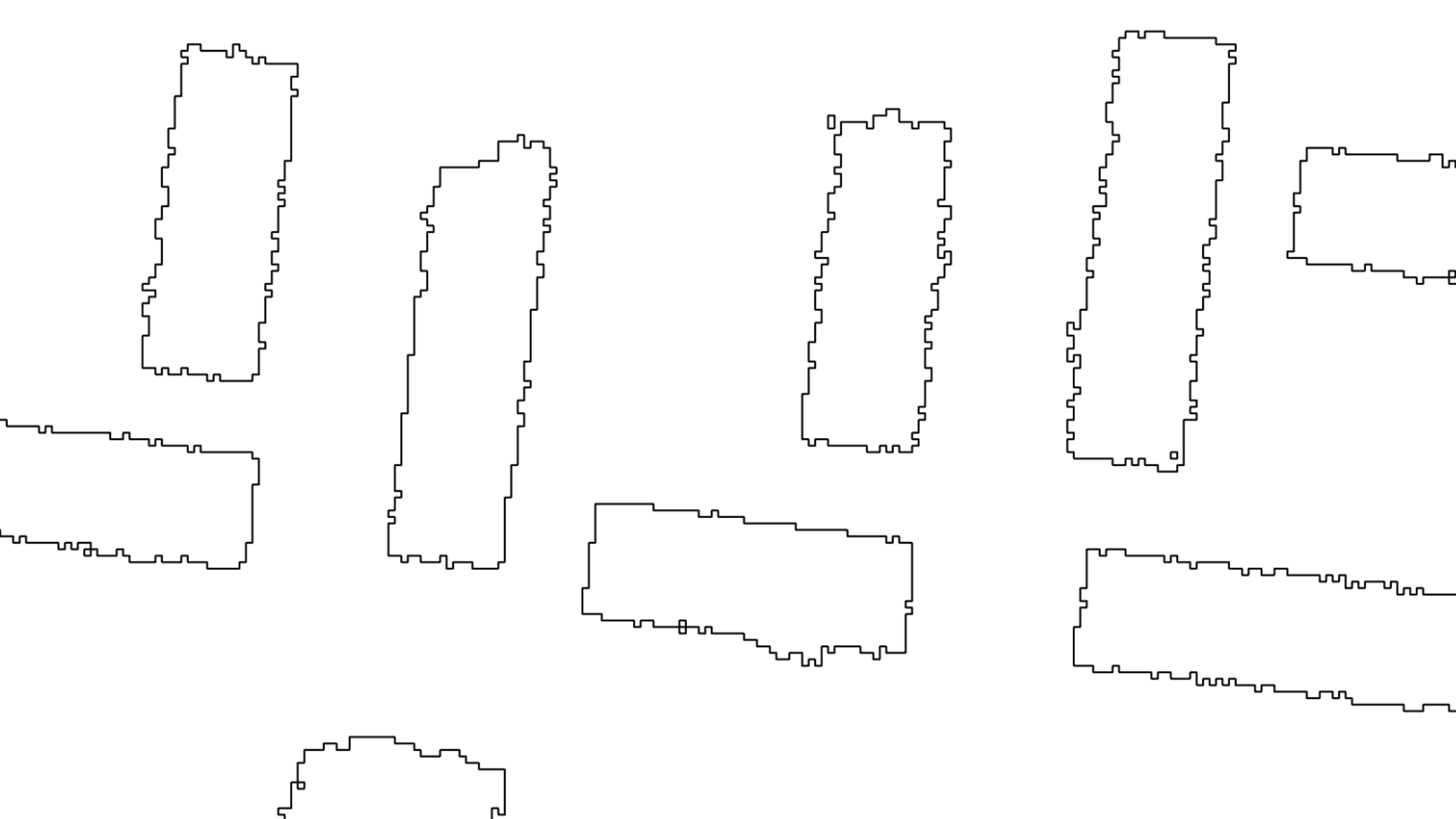
Building Footprints

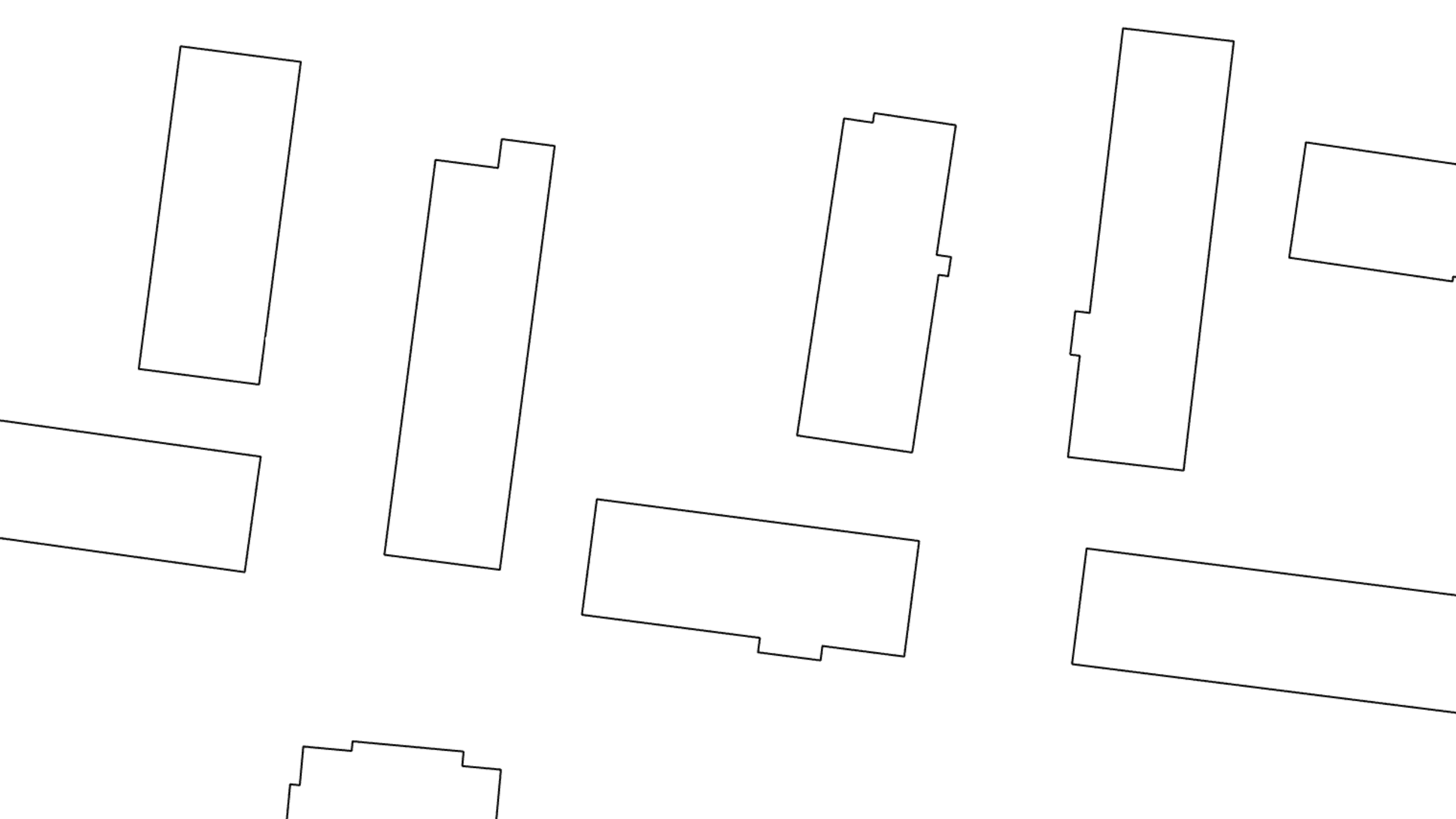
- Rasterize building points
- Vectorize raster
- Remove small holes
- Regularize polygons











3D Building Shells

- **Simple extrusion (LOD1)**
 - Height obtained from lidar
 - Flat roofs
- **Rooftop forms (LOD2)**
 - Triangulated rooftops vs. Procedurally generated
- **Building sides**
 - Majority of aerial lidar does not reliably capture sides
 - As consequence sides are modeled using simple vertical walls

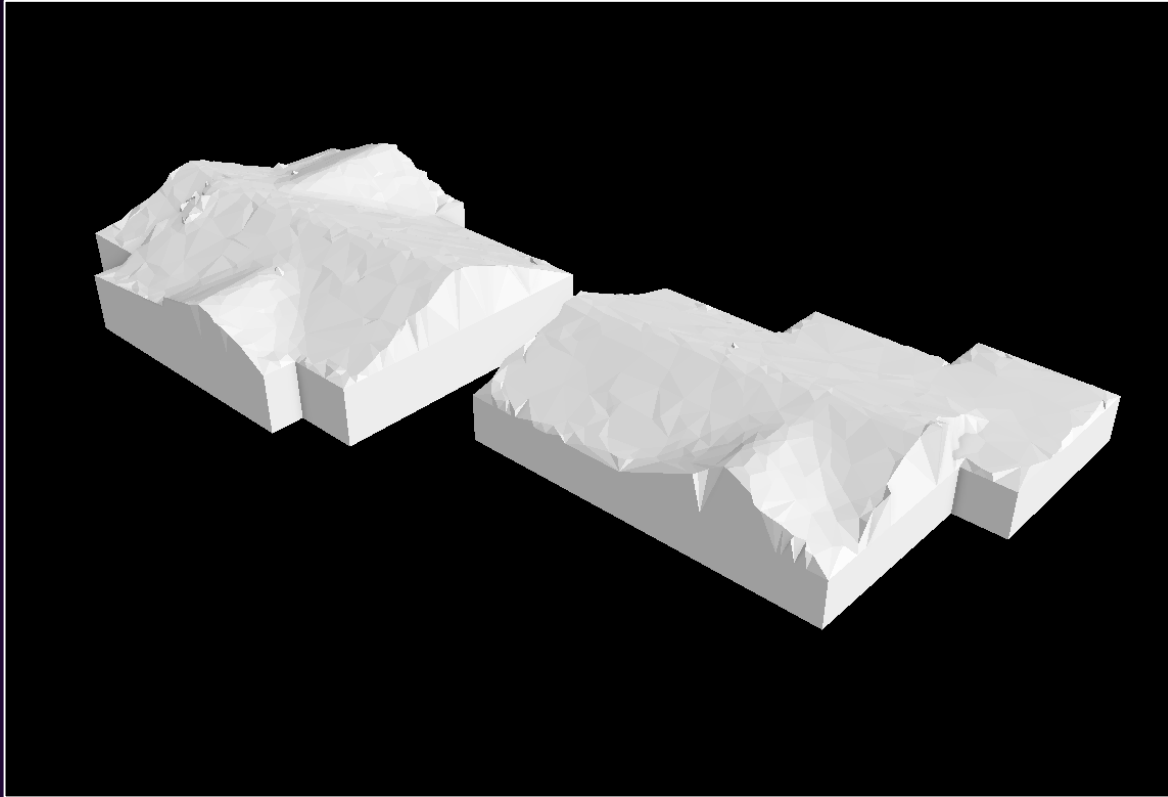
LOD1
Simple Extrusion



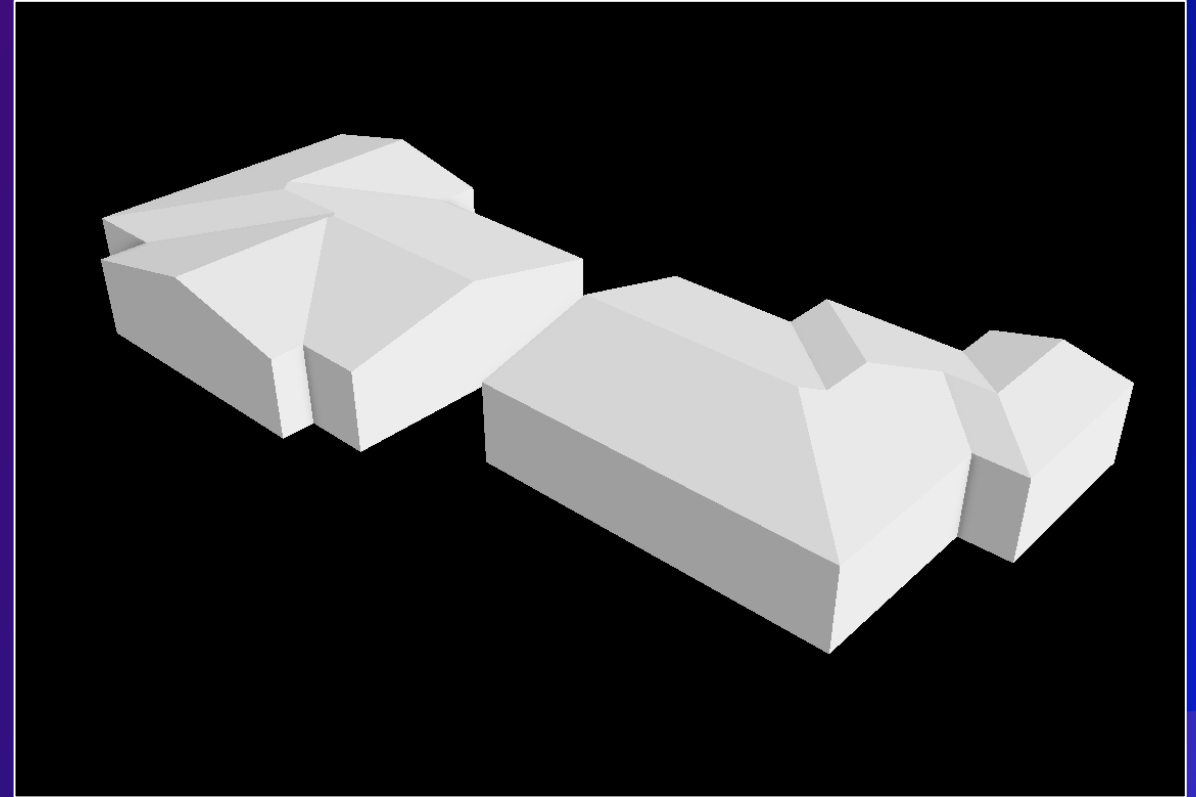
LOD2
Roof Form



3D Building Shells

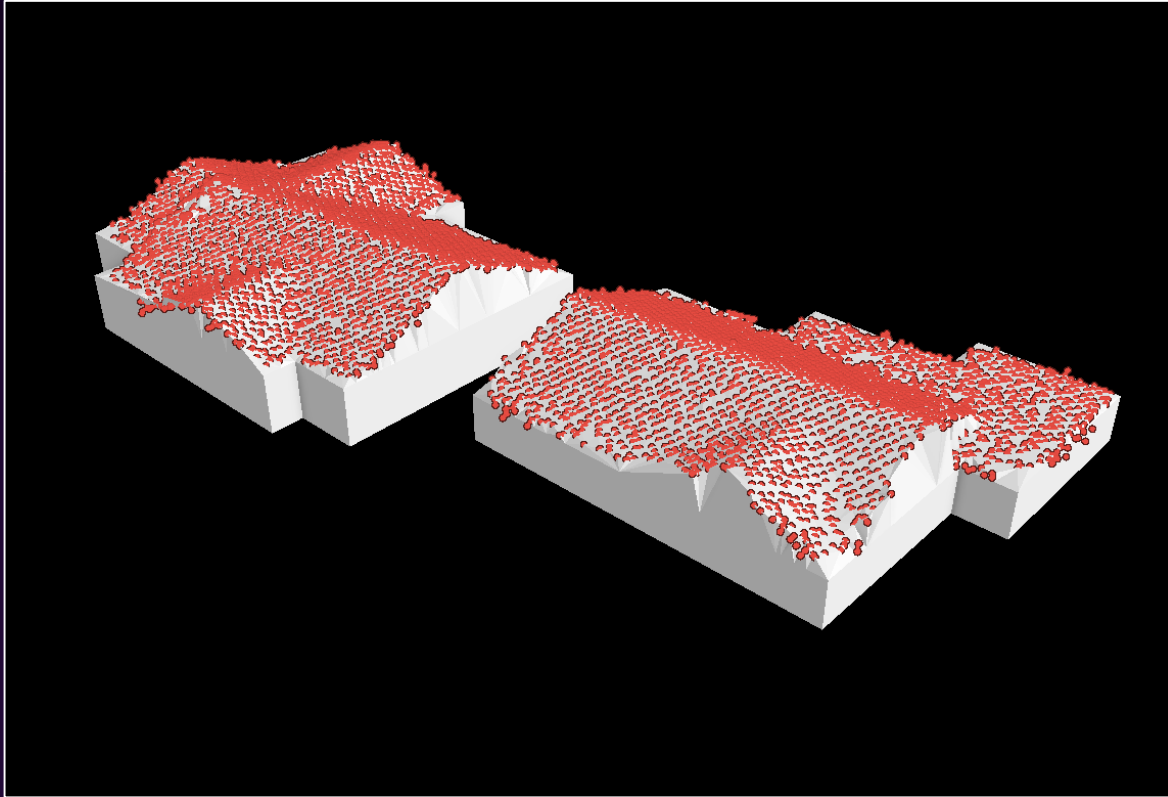


LAS Building Multipatch tool
(triangulation)

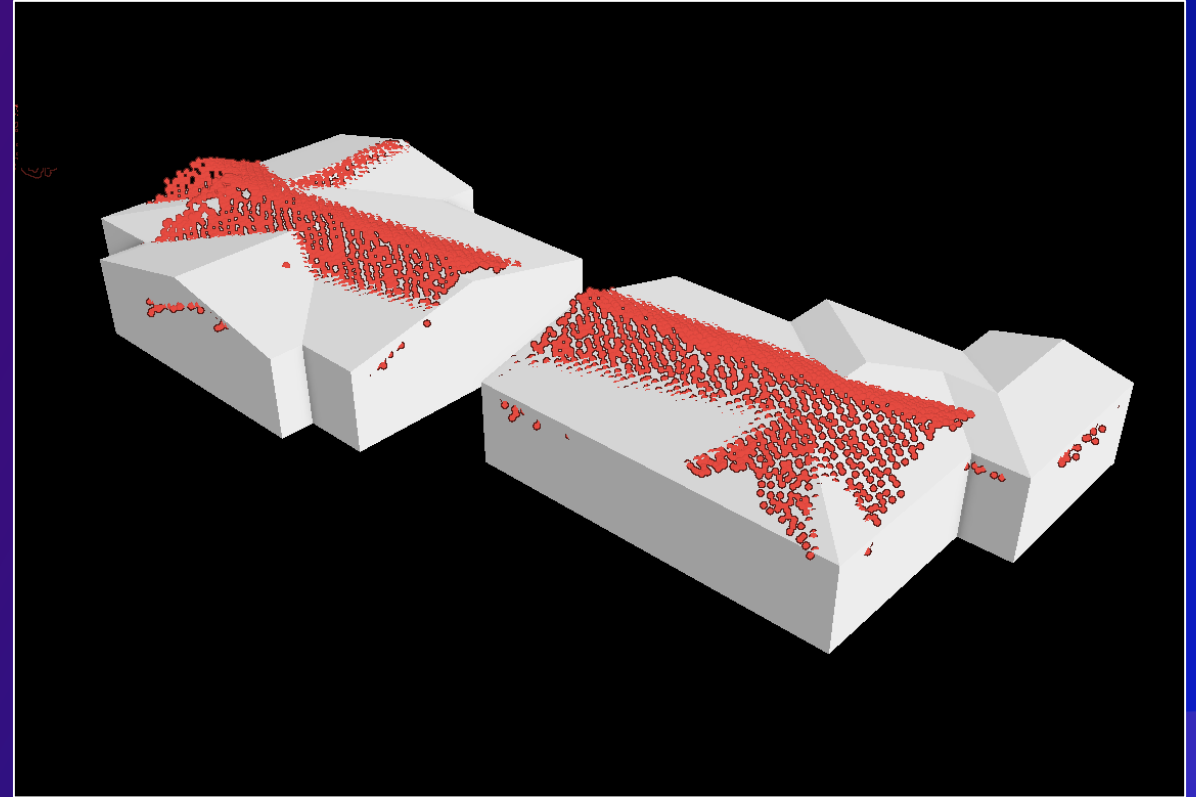


3D Basemaps Solution
(procedural symbology)

3D Building Shells



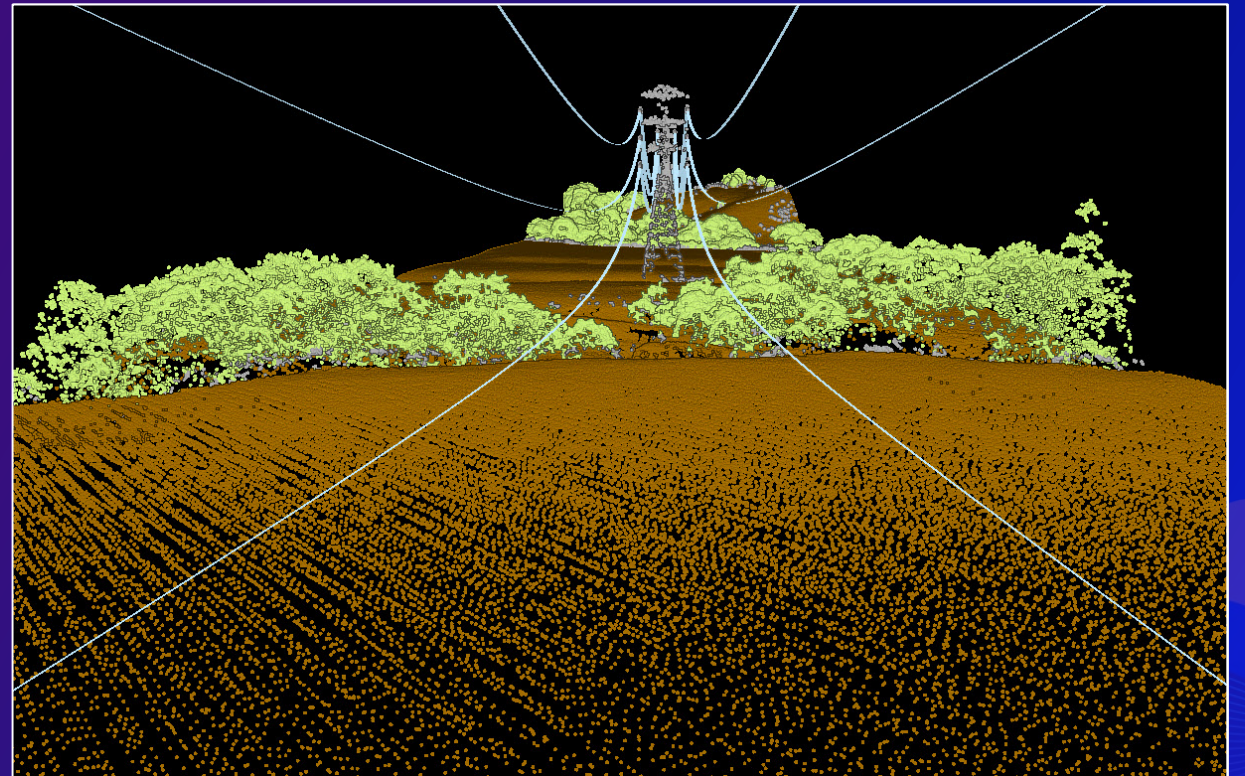
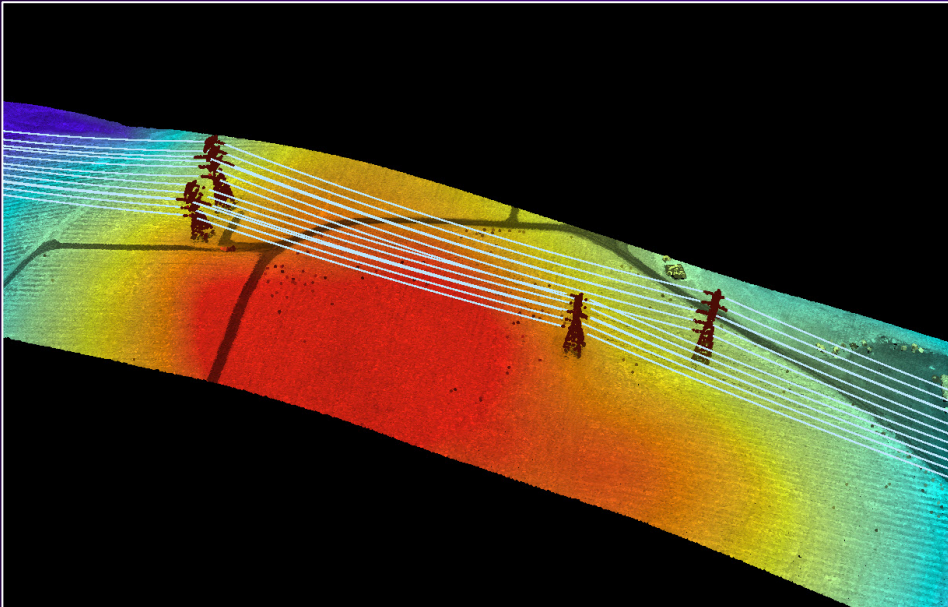
LAS Building Multipatch tool
(triangulation)



3D Basemaps Solution
(procedural symbology)

Powerlines

- Extract Powerlines From Point Cloud geoprocessing tool
- Need tight point spacing and good positional accuracy
 - E.g., 0.3 meter spacing or better
- Meant for larger transmission lines





Demo

3D Basemaps Solution

- Task driven workflows for creating and publishing a standard set of 3D layers:

Get to know 3D Basemaps

Publish data for your 3D
Basemaps

Publish ground elevation surface

Publish buildings

Publish floors

Publish trees

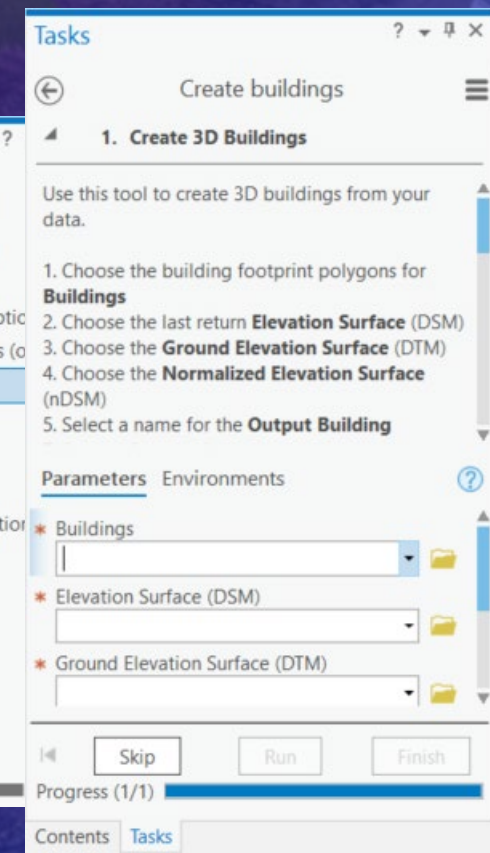
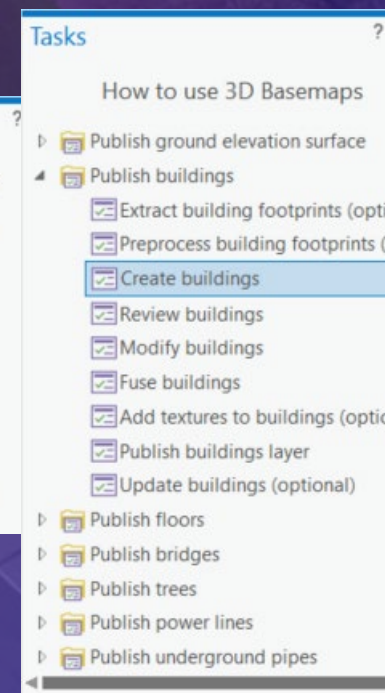
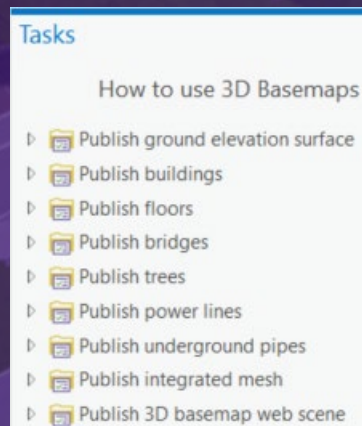
Publish bridges

Publish power lines

Publish underground pipes

Publish integrated mesh

Publish 3D Basemaps



<https://doc.arcgis.com/en/arcgis-solutions/latest/reference/introduction-to-3d-basemaps.htm>



Questions?



esri®

THE
SCIENCE
OF
WHERE®