# ArcGIS 3D Analyst: Lidar Classification and Feature Extraction

Lindsay Weitz

Khalid Duri

Clayton Crawford

2021 ESRI USER CONFERENCE

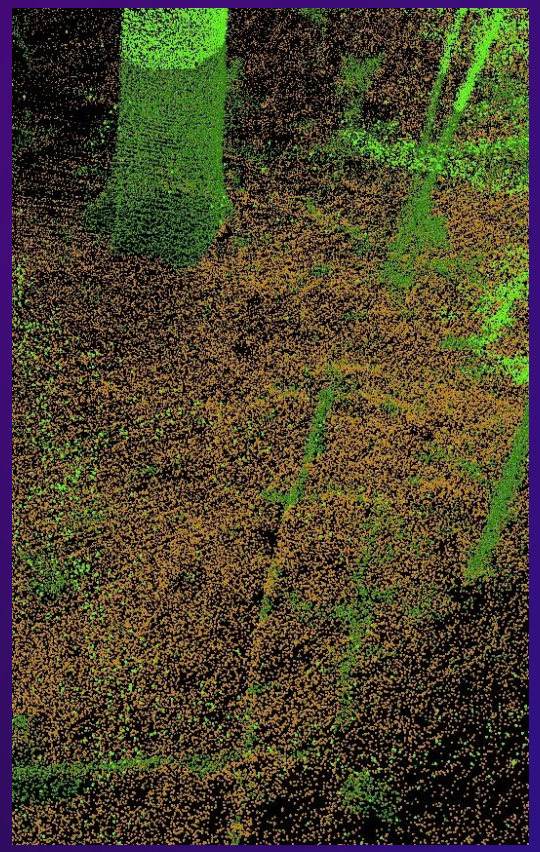
## Workshop Overview

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

# ArcGIS Supports

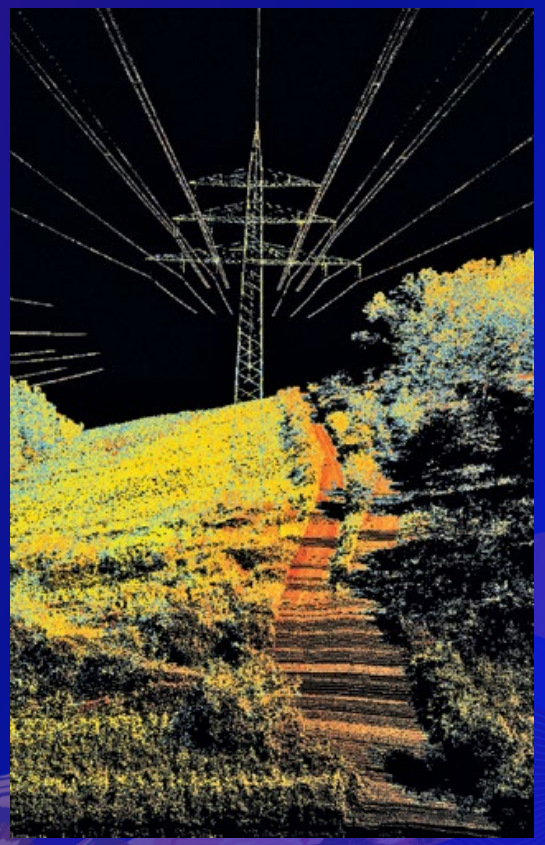
Airborne

Terrestrial



Mobile

Drone/UAV



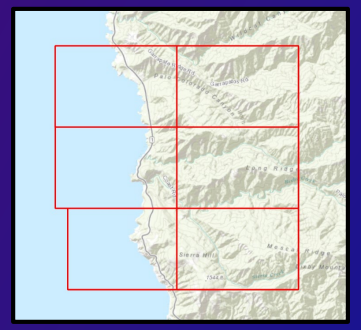
# Data formats and management

- Formats
  - LAS / zLAS / LAZ
- Management

## LAS Dataset

ARRA-CA_CentralCoast-Z4_2010_000070.las
ARRA-CA_CentralCoast-Z4_2010_000172.las
ARRA-CA_CentralCoast-Z4_2010_000293.las
ARRA-CA_CentralCoast-Z4_2010_000331.las
ARRA-CA_CentralCoast-Z4_2010_000407.las
ARRA-CA_CentralCoast-Z4_2010_000429.las

→ BigSur.lasd →



## Mosaic Dataset

MA_ME_MA_QL2_UTM19_L1_2015_000074.las
MA_ME_MA_QL2_UTM19_L1_2015_000075.las
MA_ME_MA_QL2_UTM19_L1_2015_000089.las
MA_ME_MA_QL2_UTM19_L1_2015_000174.las

→ WorcesterMA.lasd →


## Point Cloud Scene Layer

WorcesterMA.lasd

↓

Worcester_SL_package.slpk

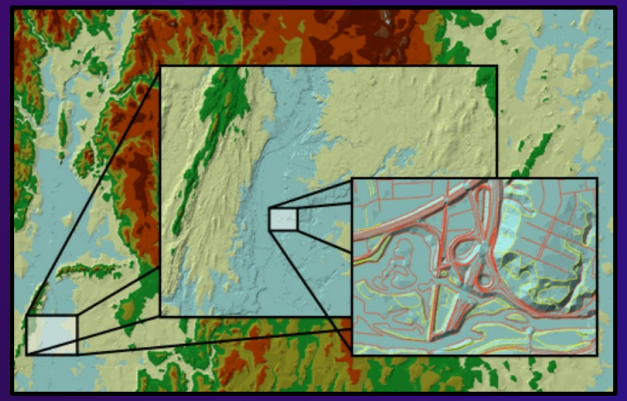
↓

| | | |
|---|---|---|
| ☐ | Worcester_SL_package | Scene Layer (hosted) |
| ☐ | Worcester_SL_package | Scene Layer Package |

## Terrain Dataset

- LiDAR_Bootcamp_Demos.gdb
  - BigSur_FD
    - BigSur_Masspoints
    - BigSur_Streams_Z
    - BigSur_StudyArea
    - BigSur_Terrain

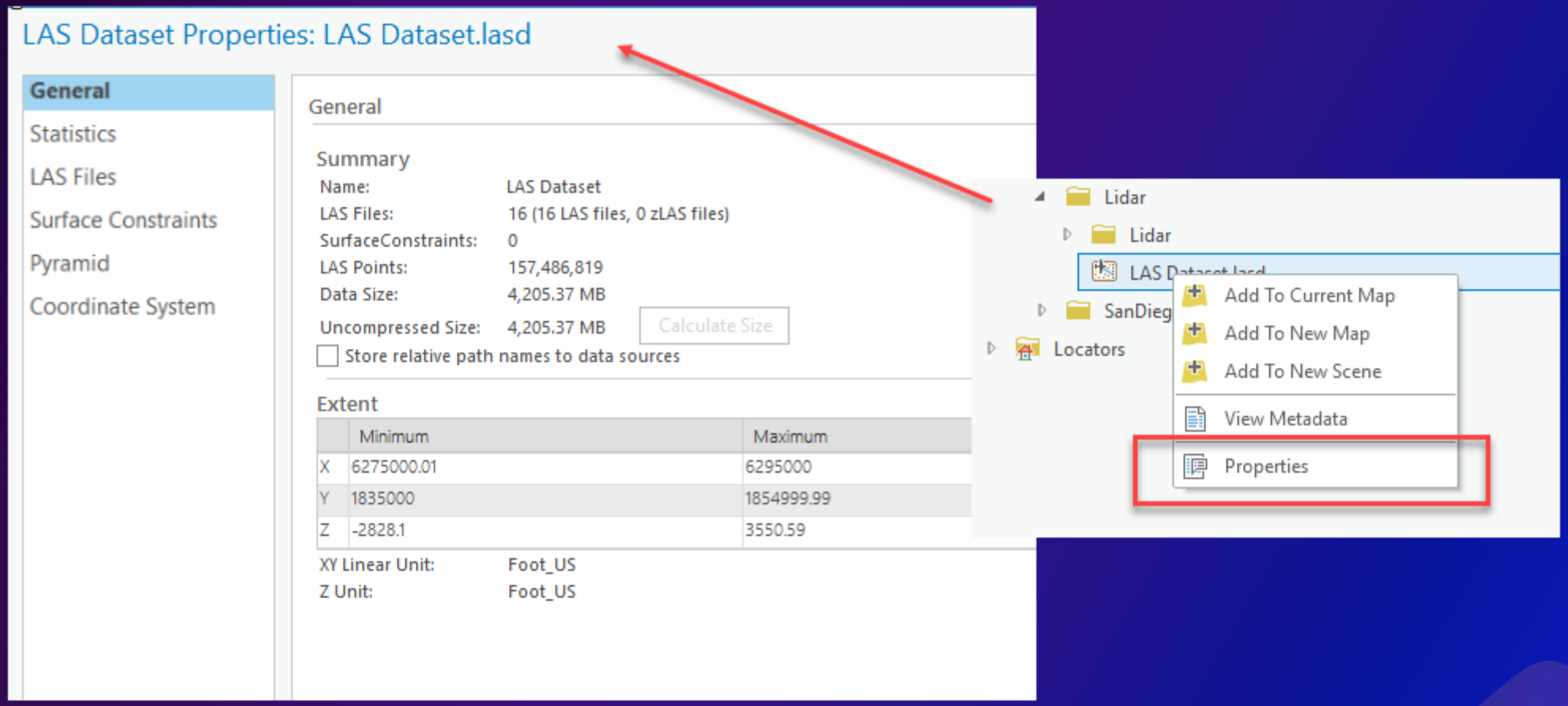
→



# Lidar data with a LAS dataset

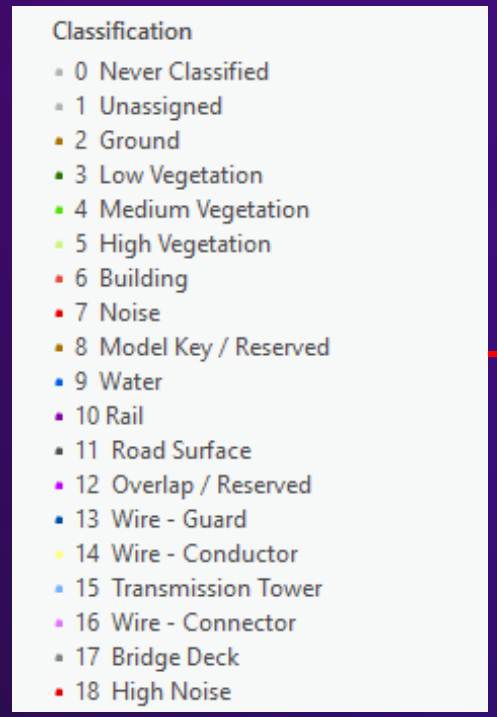
- Direct read of LAS or zLAS format lidar
- File based
- QA/QC tools
- Stores references to LAS/zLAS files on disk
- Optionally reference breakline and control point data
- Treats a collection of LAS/zLAS files as one logical dataset (“Project”)

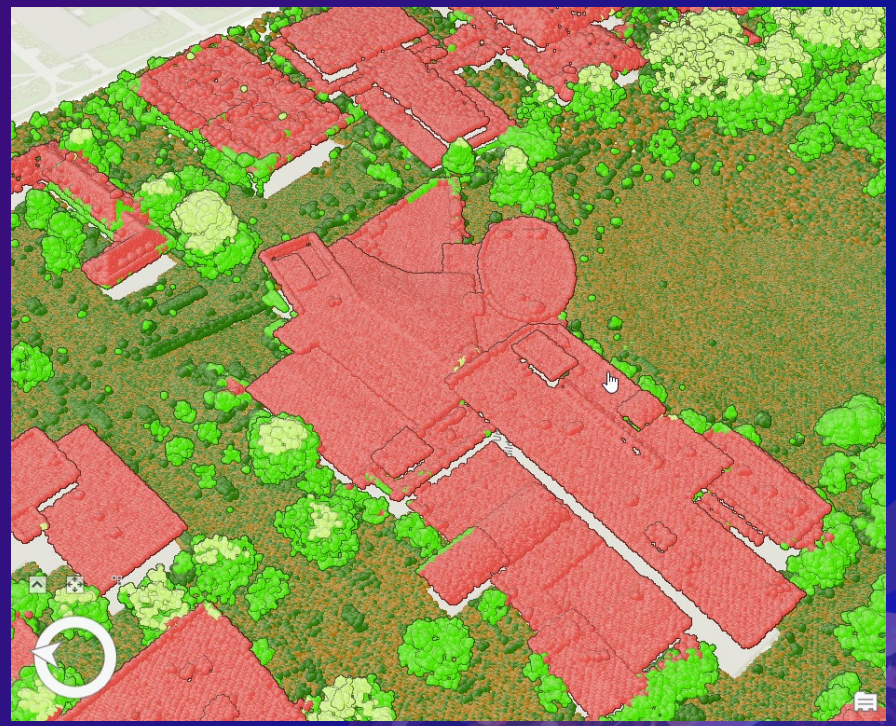
# QA/QC: LAS Dataset Properties



# Classification codes

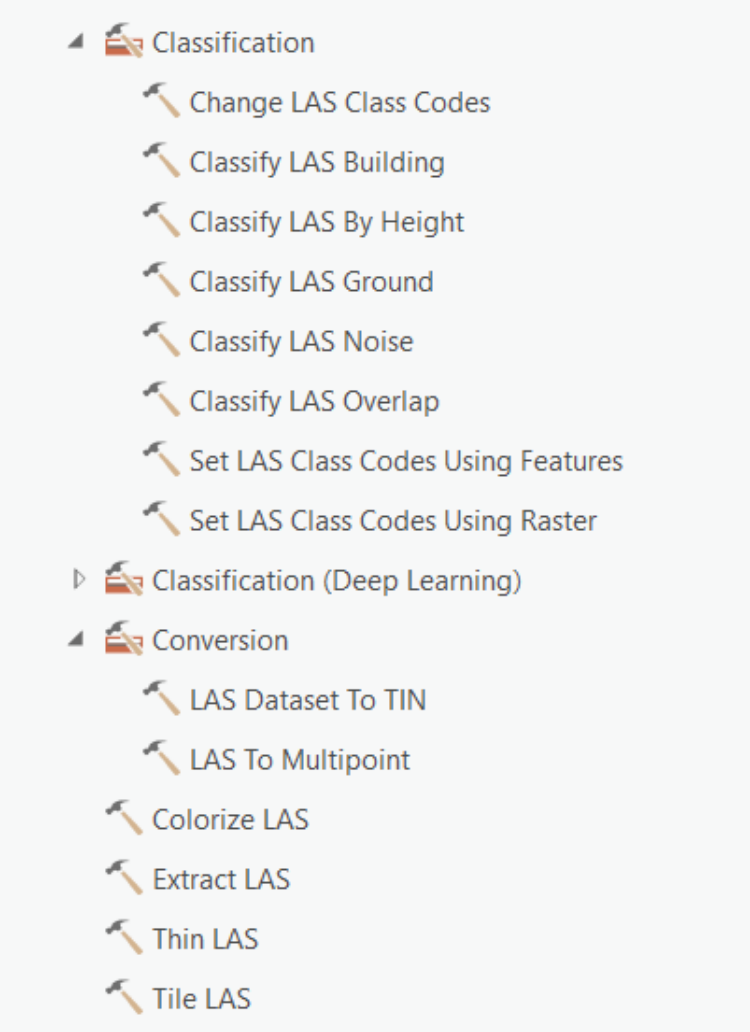
- Every lidar point can have a classification that defines the type of object that has reflected the laser pulse.
- Lidar points can be classified into a number of categories.
- The different classes are defined using numeric integer codes in the LAS files.



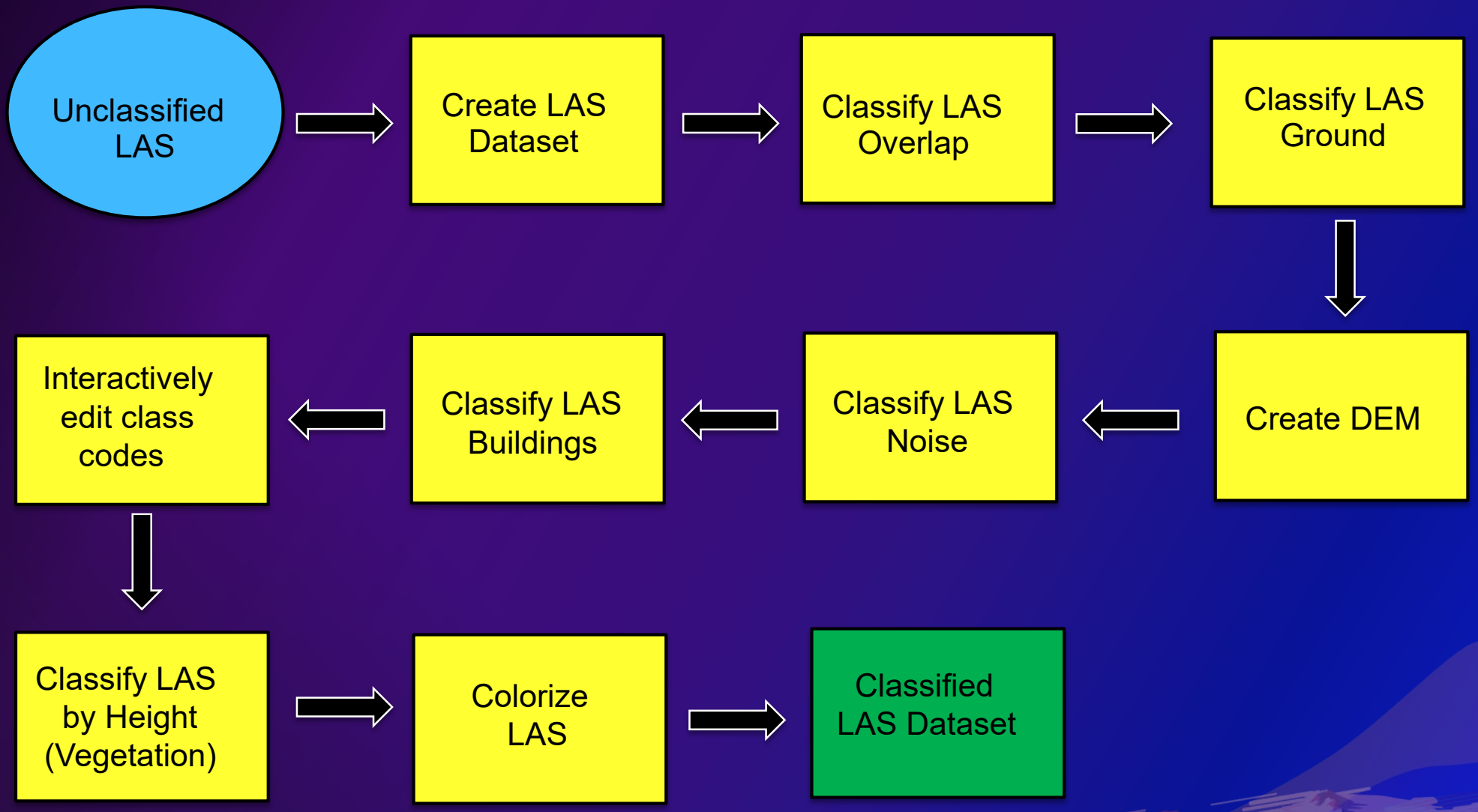



# Tools for managing lidar

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



# Example Workflow Classifying and Processing Lidar



# Rule-Based Classification

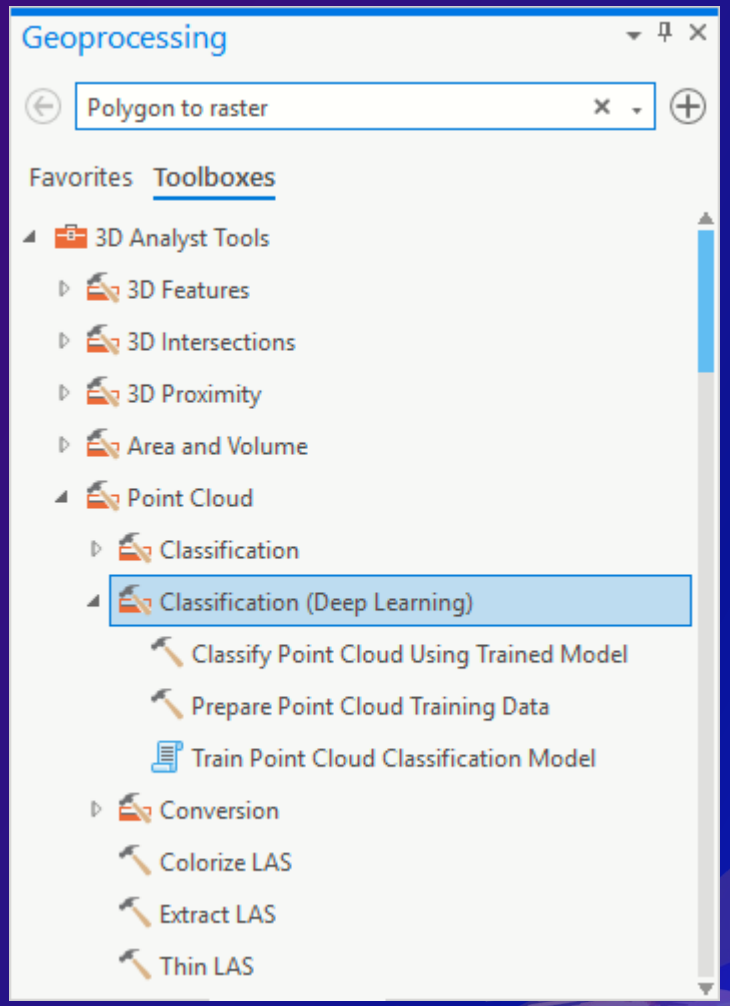
Lindsay Weitz

# Deep Learning for Point Cloud Classification

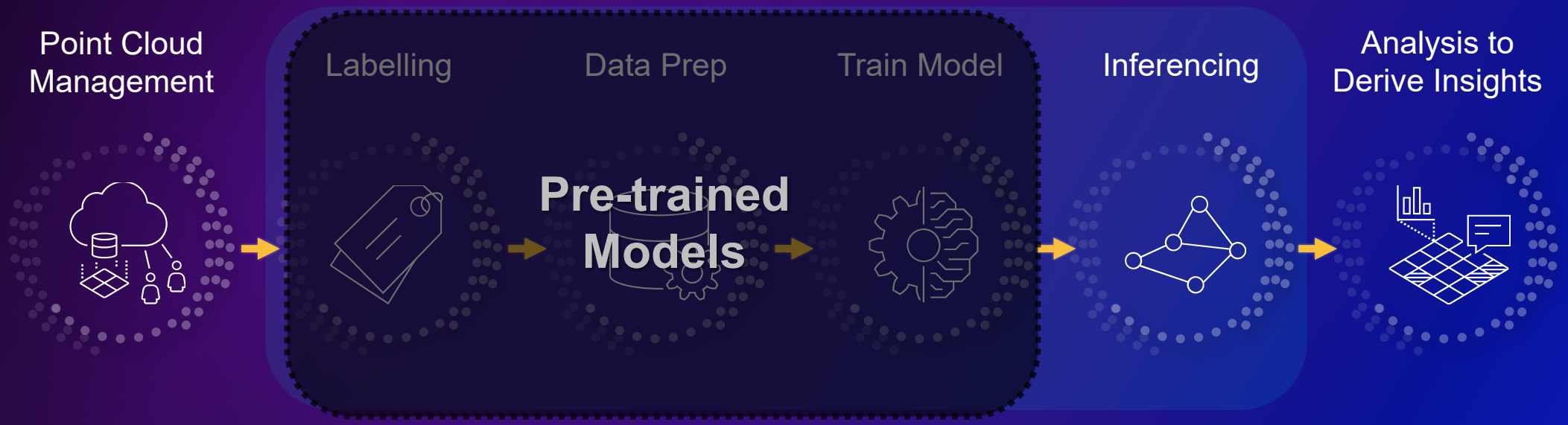
Overview of Basic Requirements & Best Practices

# 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



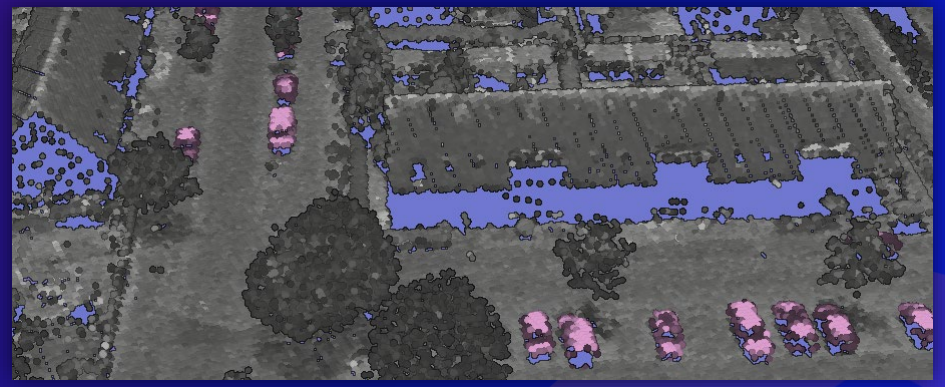
# Overview of Deep Learning Workflow




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

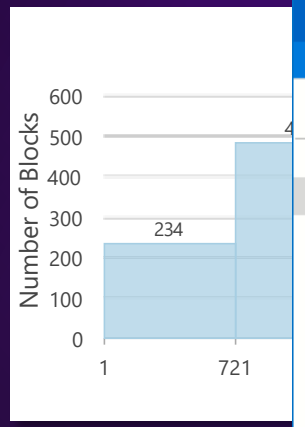
# Understanding the Training Dataset

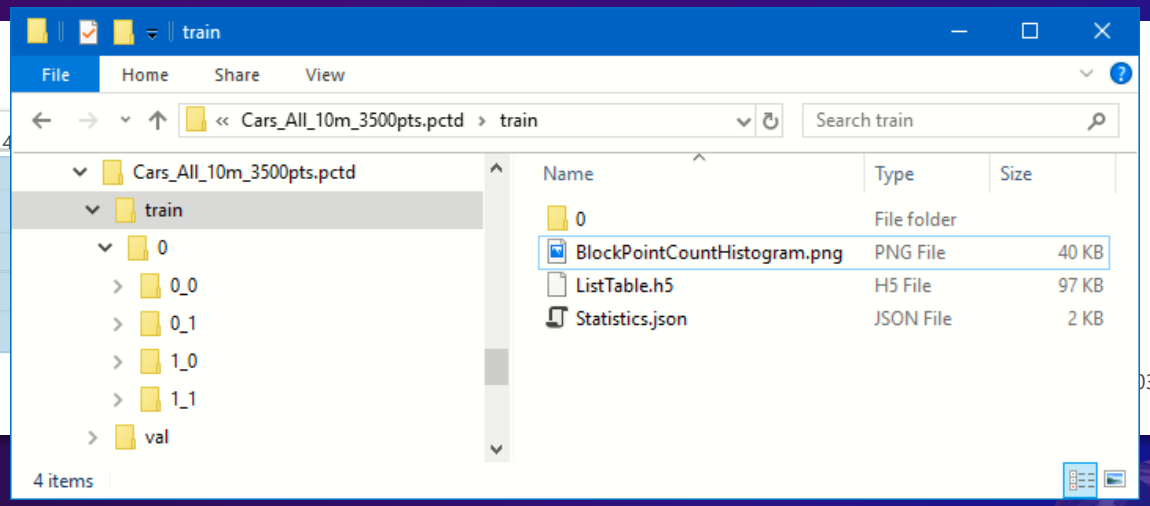
- Training dataset comprised of:
  - Data used for training the model
  - Data used for validating the model being trained
- Objects of interest must be:
  - Accurately classified
  - Captured in varied representations
- Point cloud should reflect the properties of data the model will be used on
  -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

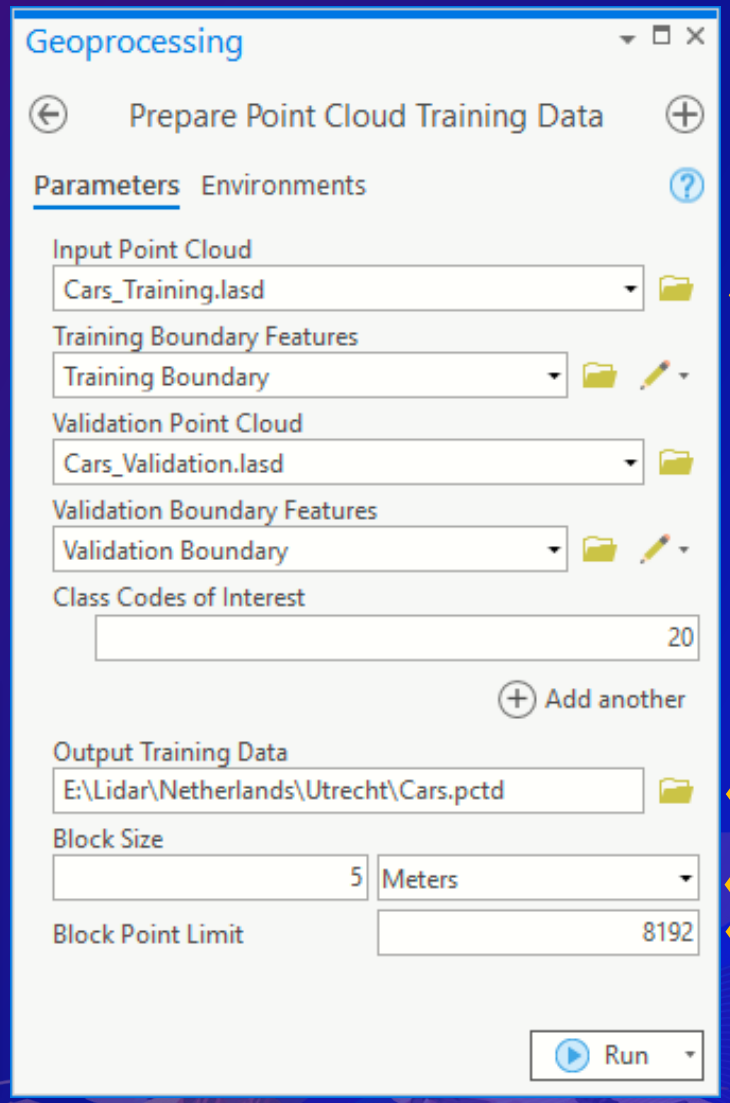


# Creating the Training Dataset

- 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

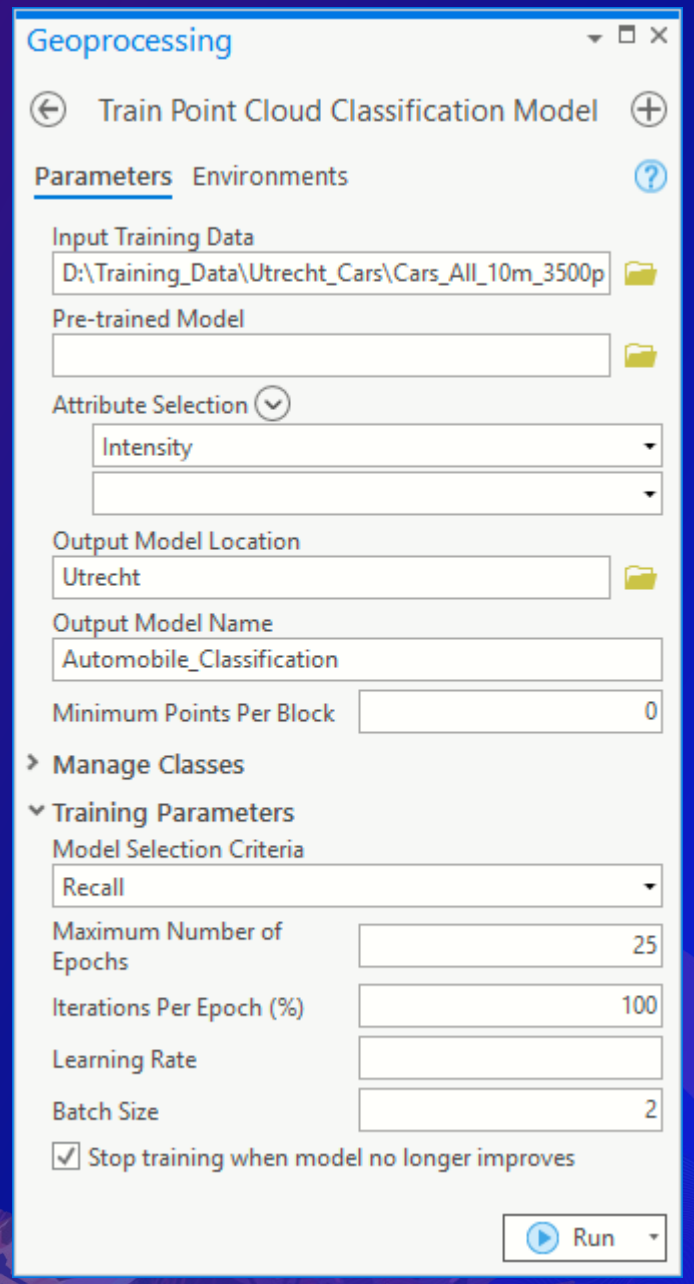
- **Epoch:** The number of times the training data is processed.
- **Iteration:** The instances within an epoch when training data is processed
- **Batch Size:** The number of blocks processed in each iteration
- **Checkpoint:** The model weights at the end of the epoch
- **Training Loss:** A measure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

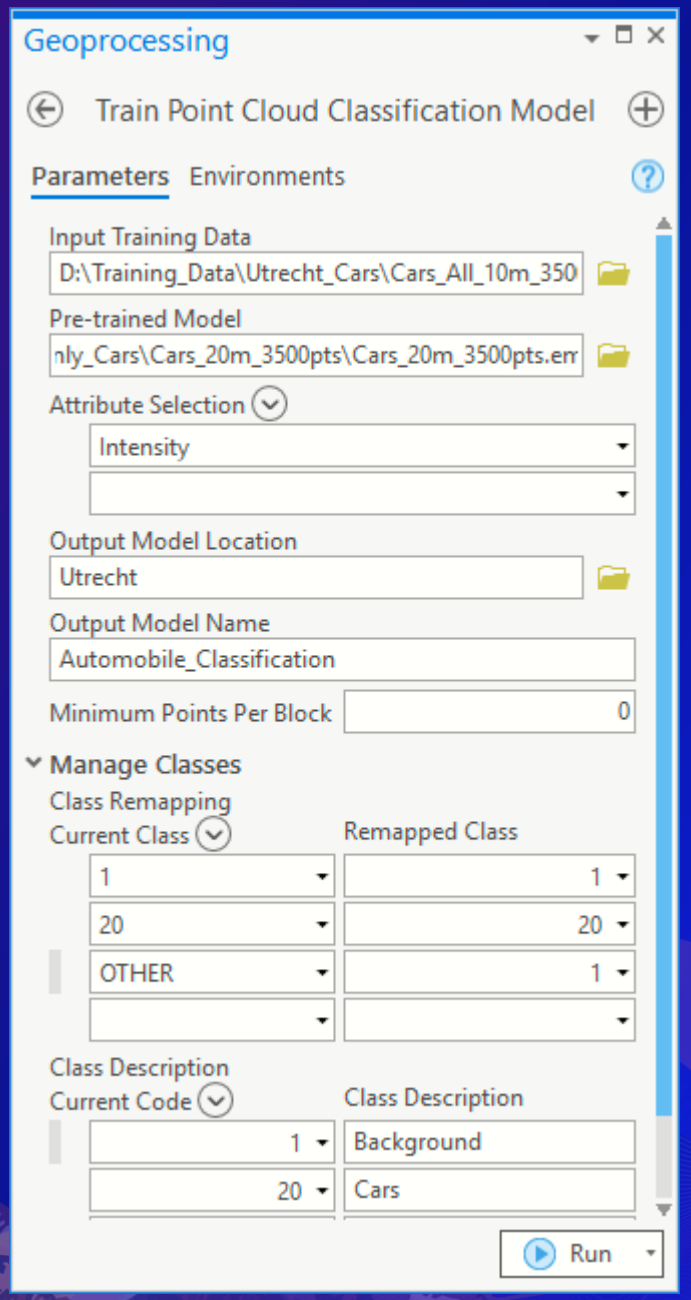
- Purpose of training is to develop and iteratively refine model weights through multiple epochs
- Classes can be remapped to simplify training
- Model metrics are reported per epoch to help understand how the model is performing
  - **Overfitting:** Model's training loss is low while validation loss 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 | 0.45975 | 00:11:04 |



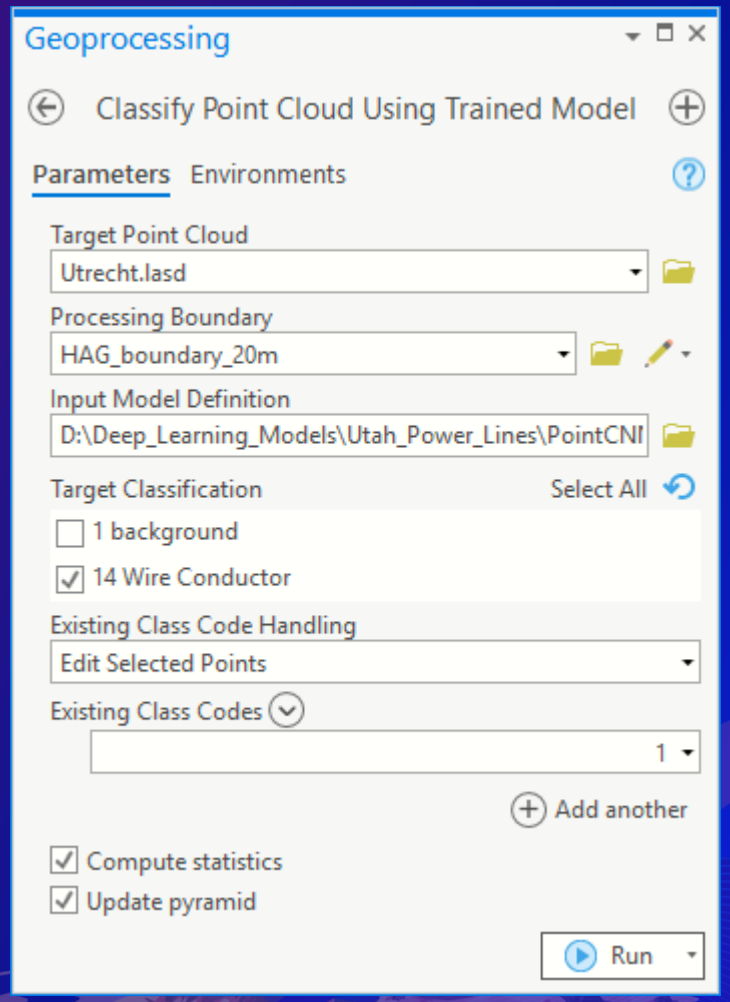
# Training the Classification Model: Transfer Learning

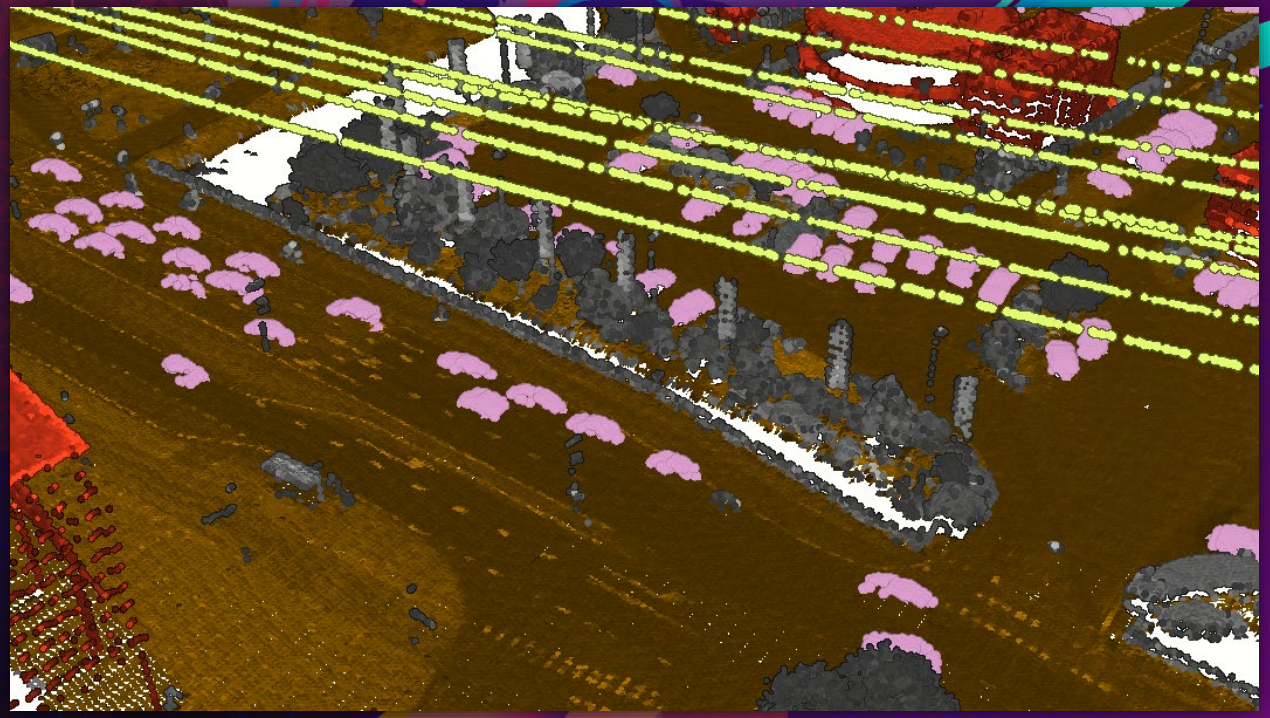
- Use model weights from a pretrained model to create a more refined model with less effort
- Training data must match attributes used in the pretrained model
- Training data must match class codes in pretrained model. Classes that differ must be remapped to those in pretrained model.



# 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

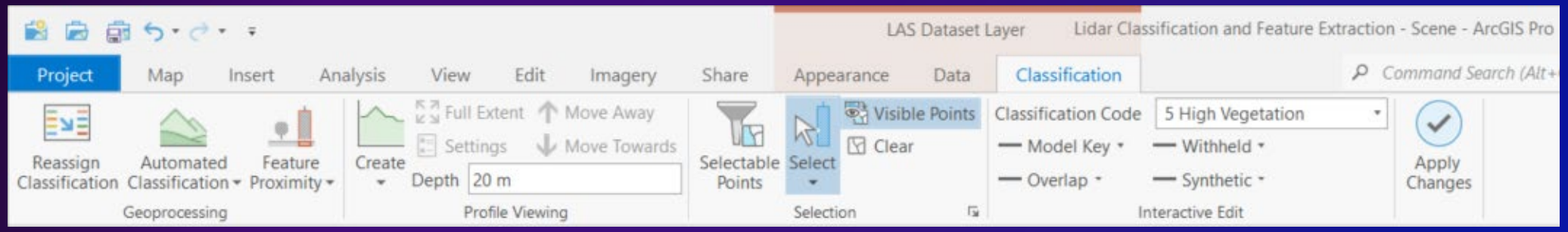
# Manual Classification

# 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



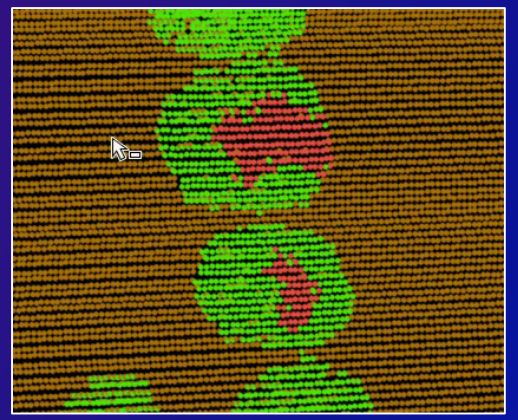

1. Select points of interest

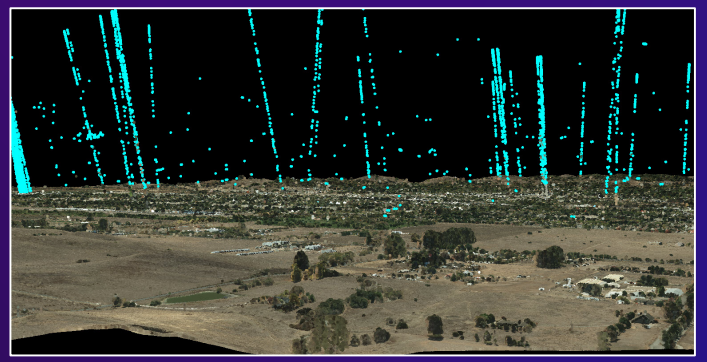
2. Set desired class code and flag settings

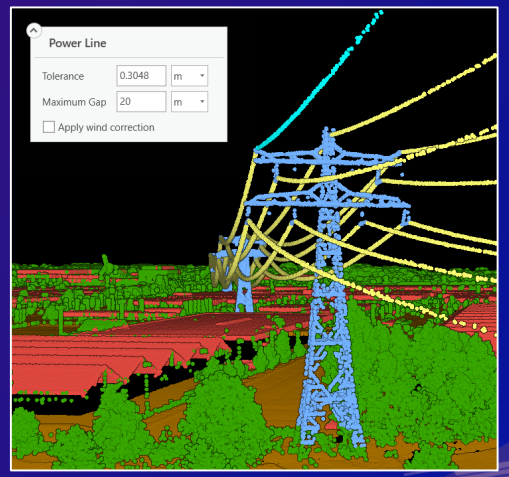
3. Apply changes

# 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





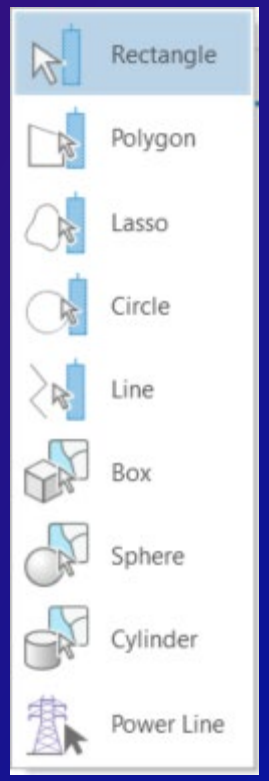


# Selection Tools

Basic primitives
(maps and scenes)

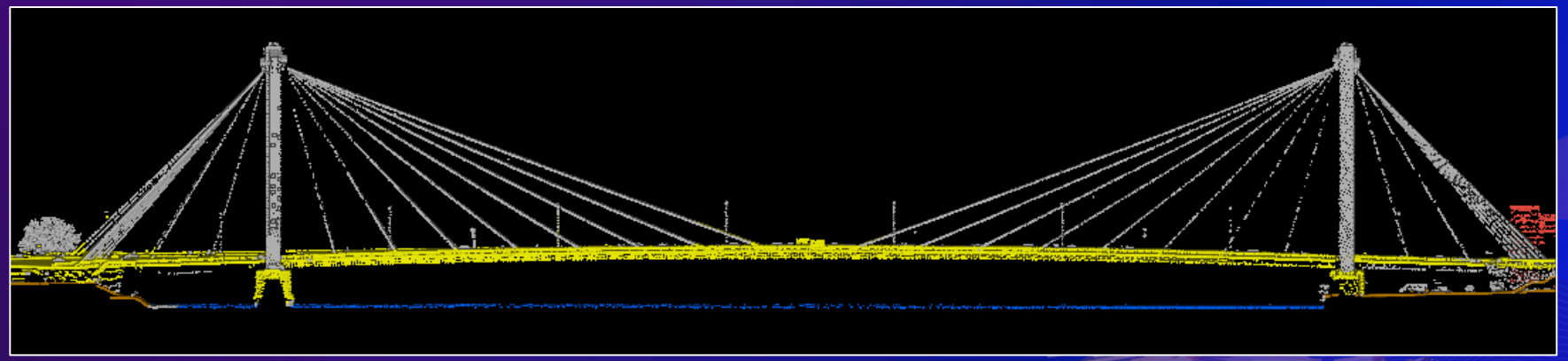
Volumetric
(scenes only)

Feature specific
(scenes only)



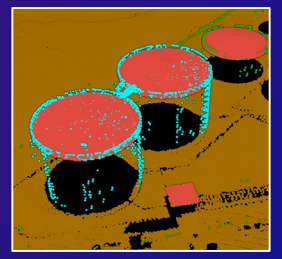
# Profile View

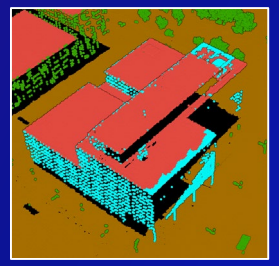
- Slice of data
- All points within slice are displayed (no thinning)
- Only in 3D scene
  - It’s implemented as a scene display mode rather than its own type of view
- Other data in scene included
  - Benchmark points
  - Breaklines

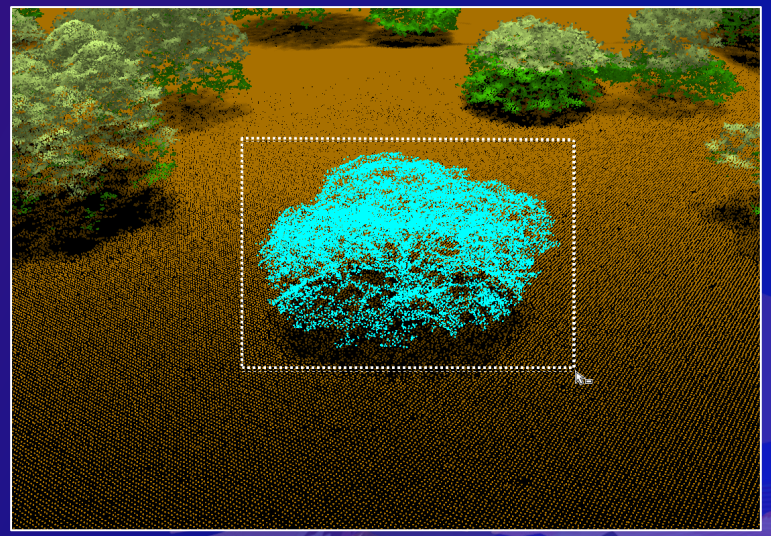


# Selection Qualifiers

- 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





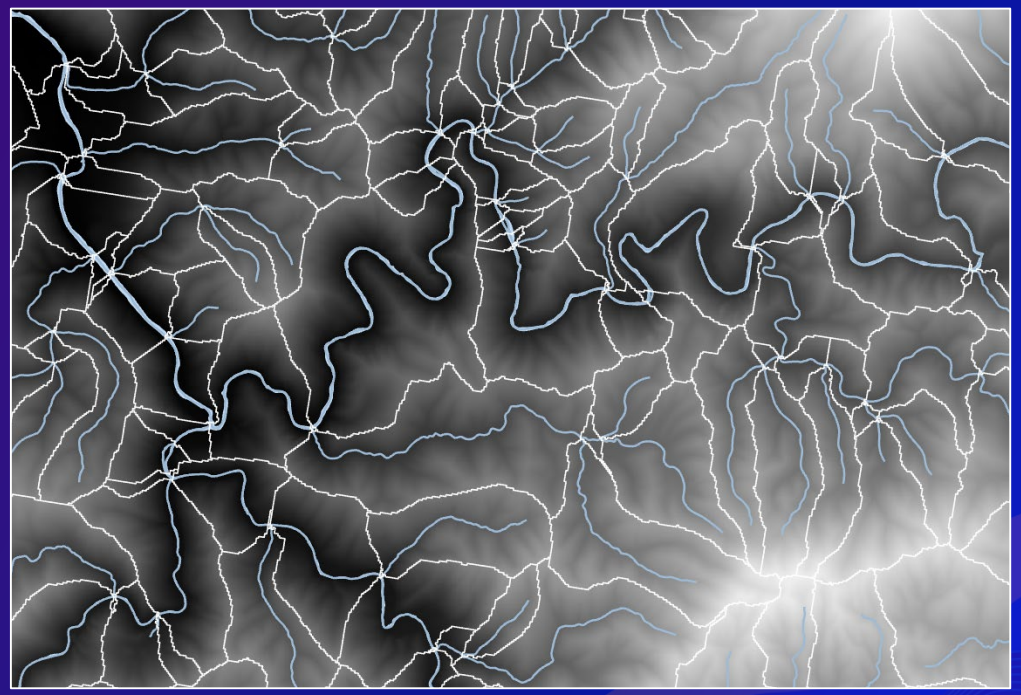


# Demo

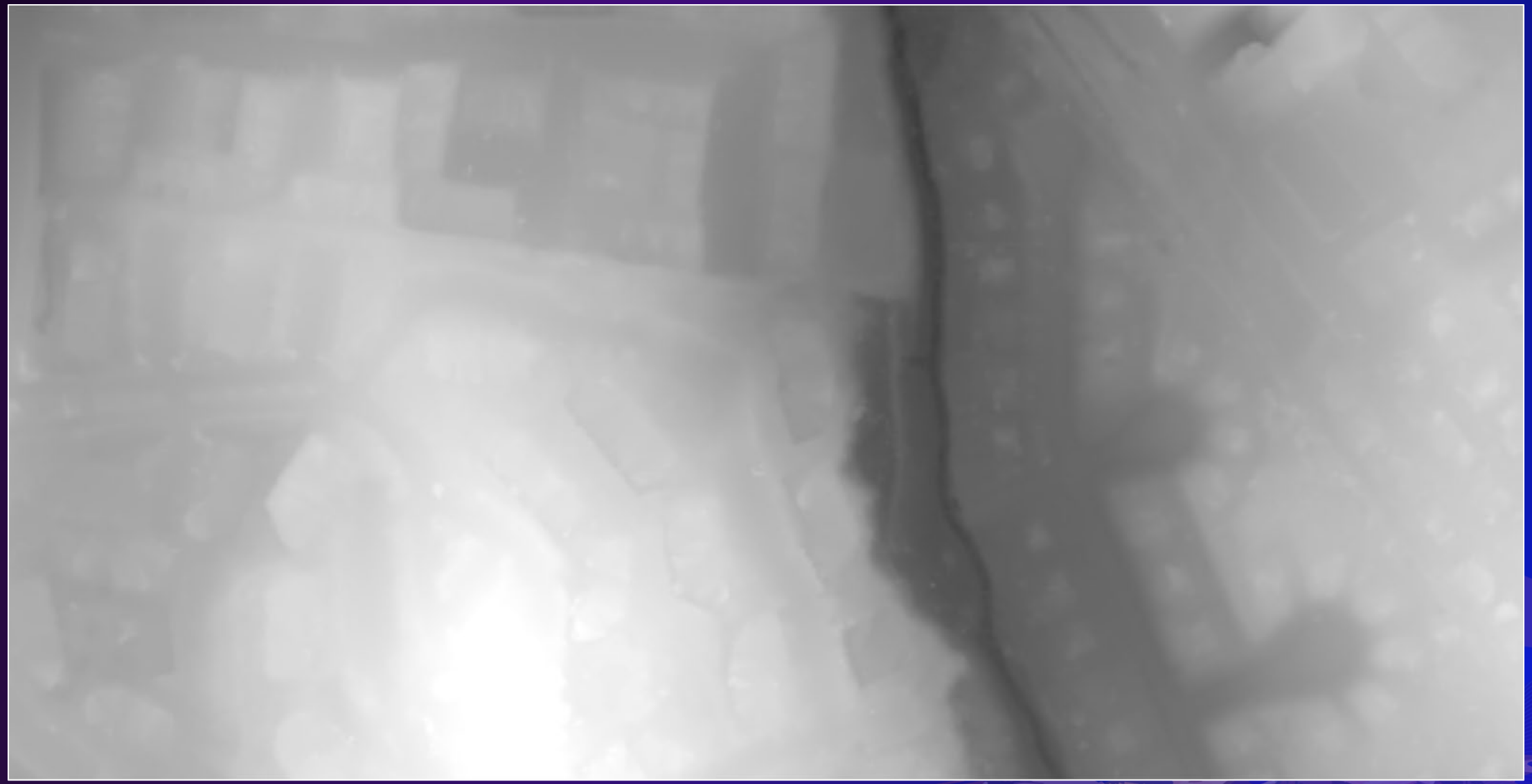
# Feature Extraction

# Surface Based Derivatives

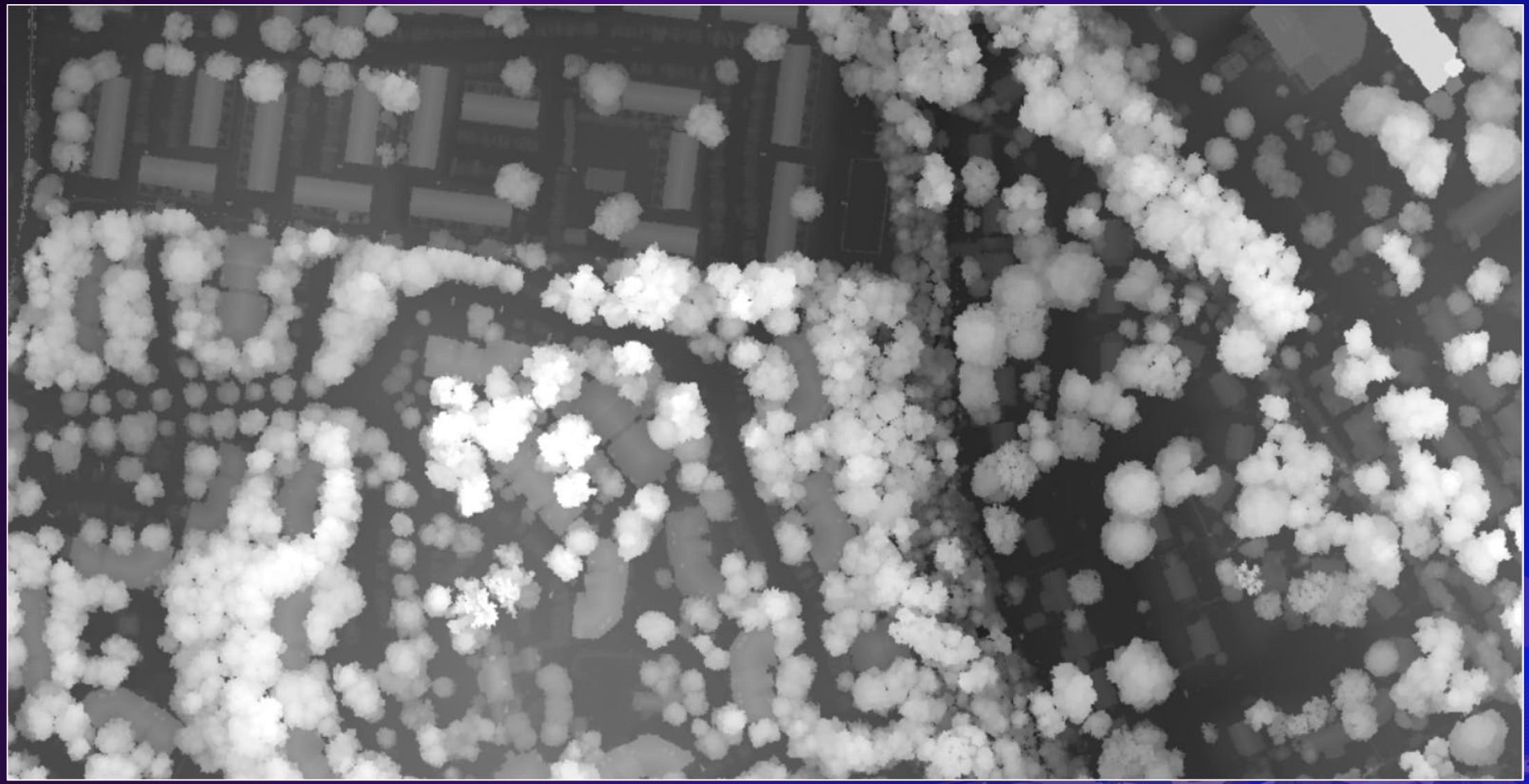
- DTMs / DEMs and DSMs
  - Can provide height info for vector feature data
  - Raster based feature extraction techniques
  - Surface derived features:
    - Contours
    - Watershed boundaries
    - Stream networks
    - Floodplain boundaries
    - Etc.
- Normalized height models
  - nDSM, CHM
  - Height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



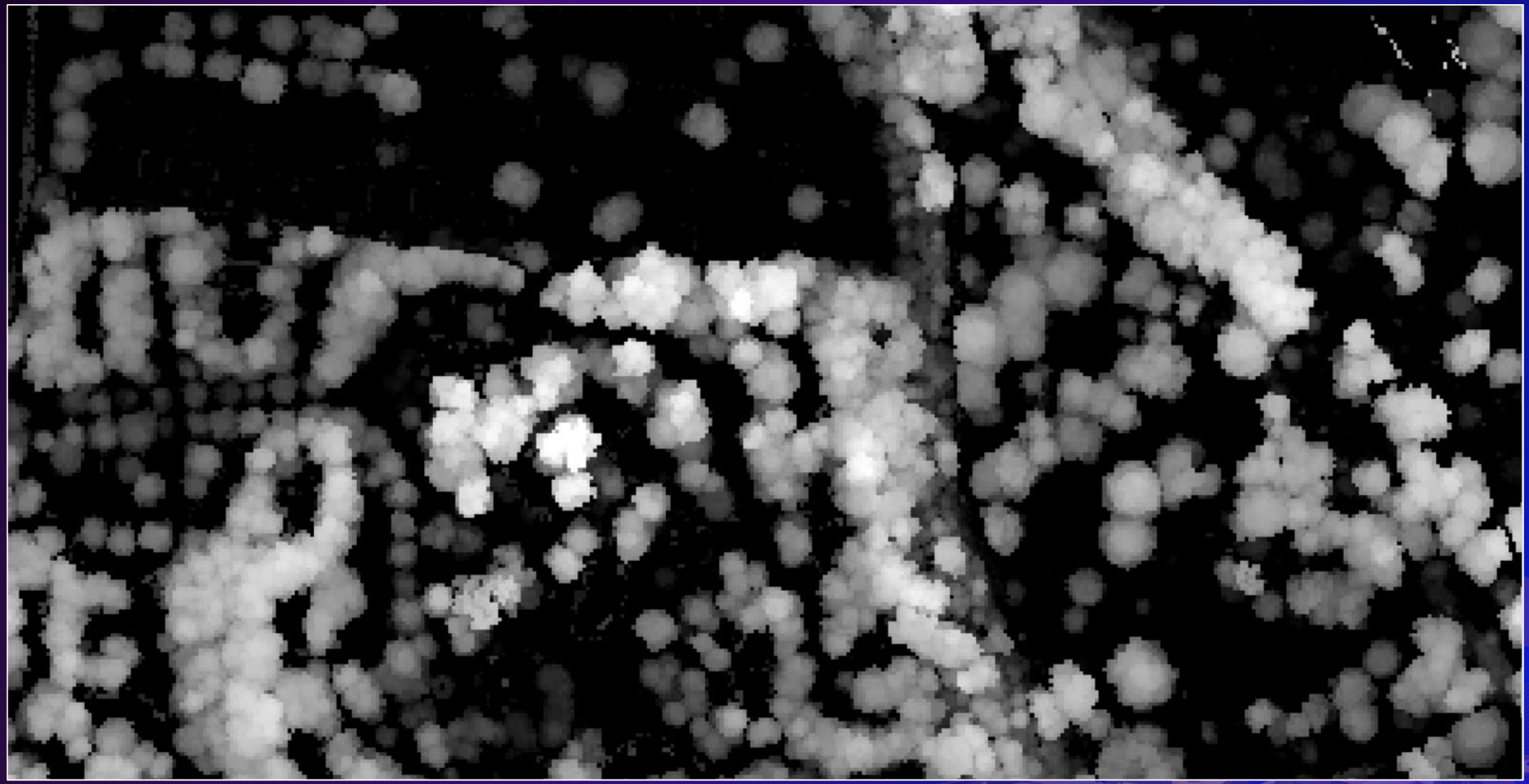
# DEM / DTM : Ground



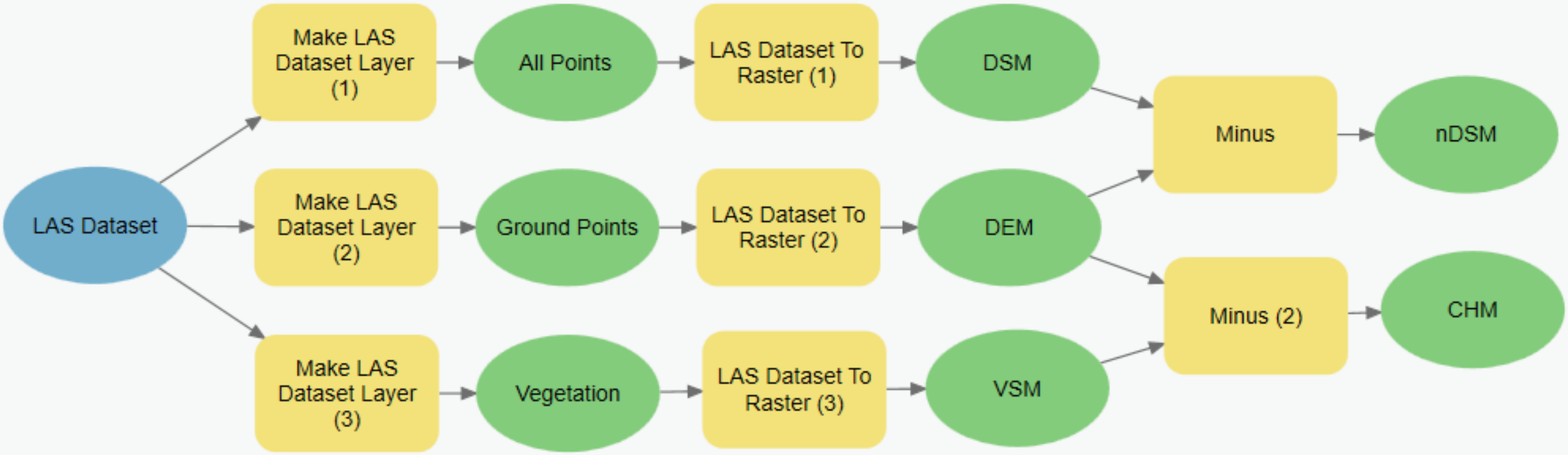
# DSM : Ground + Buildings + Vegetation



# CHM : Vegetation / Canopy Height

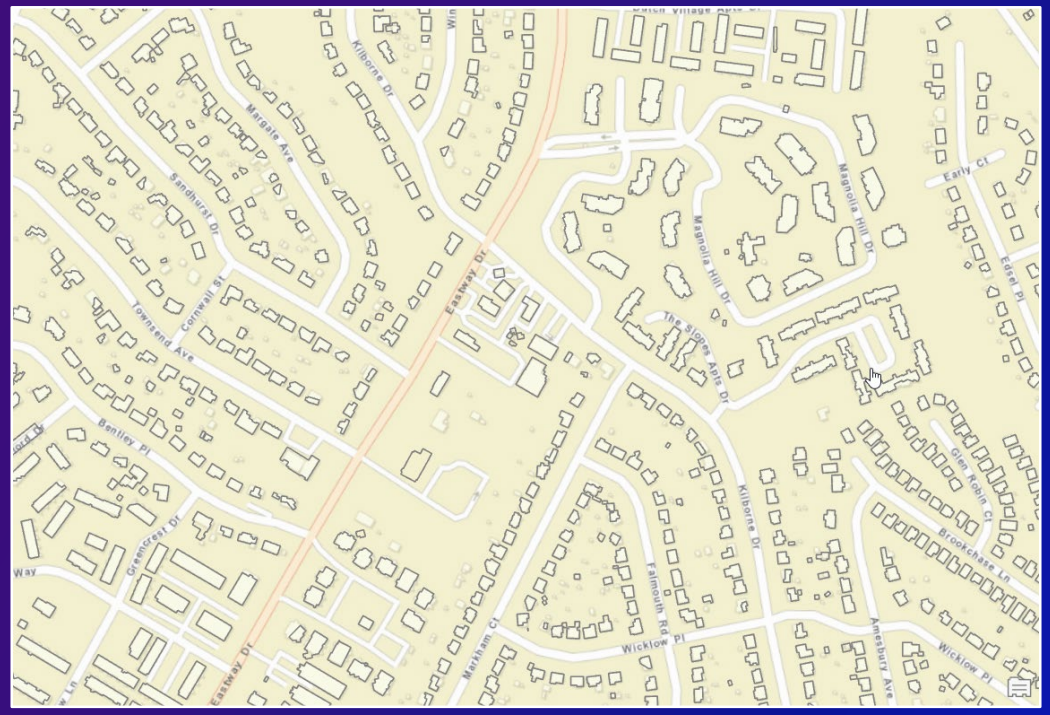


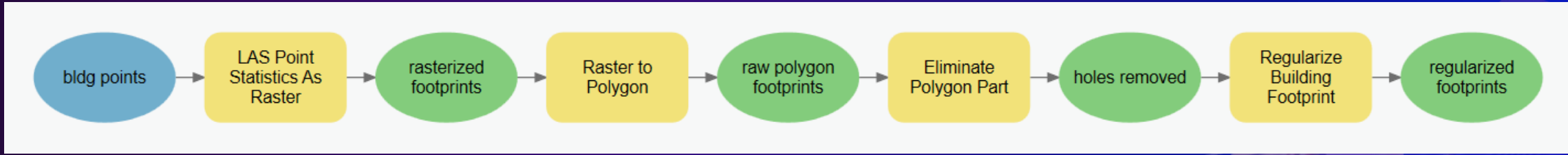
# Surface Creation – Automation with Model Builder

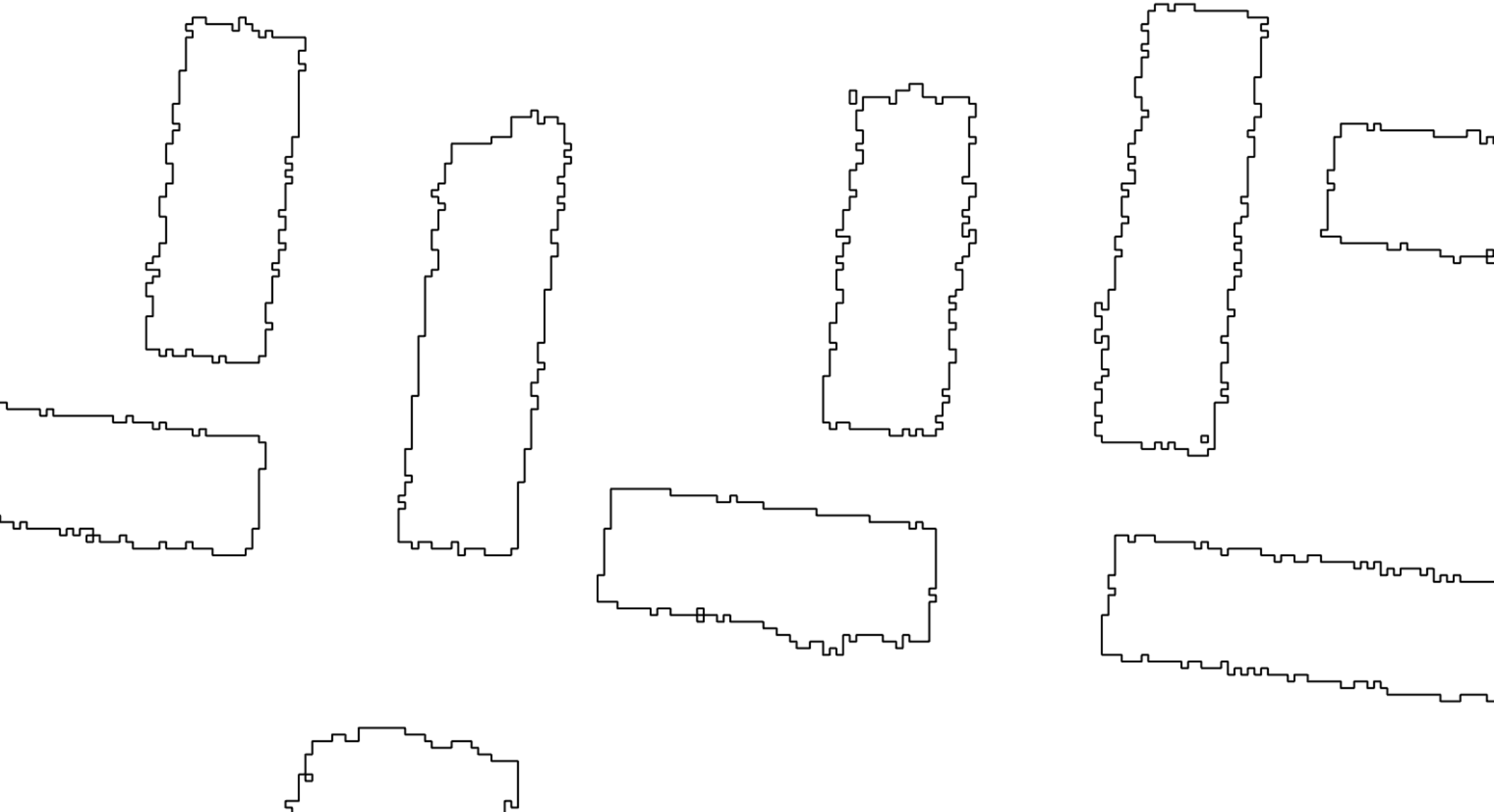


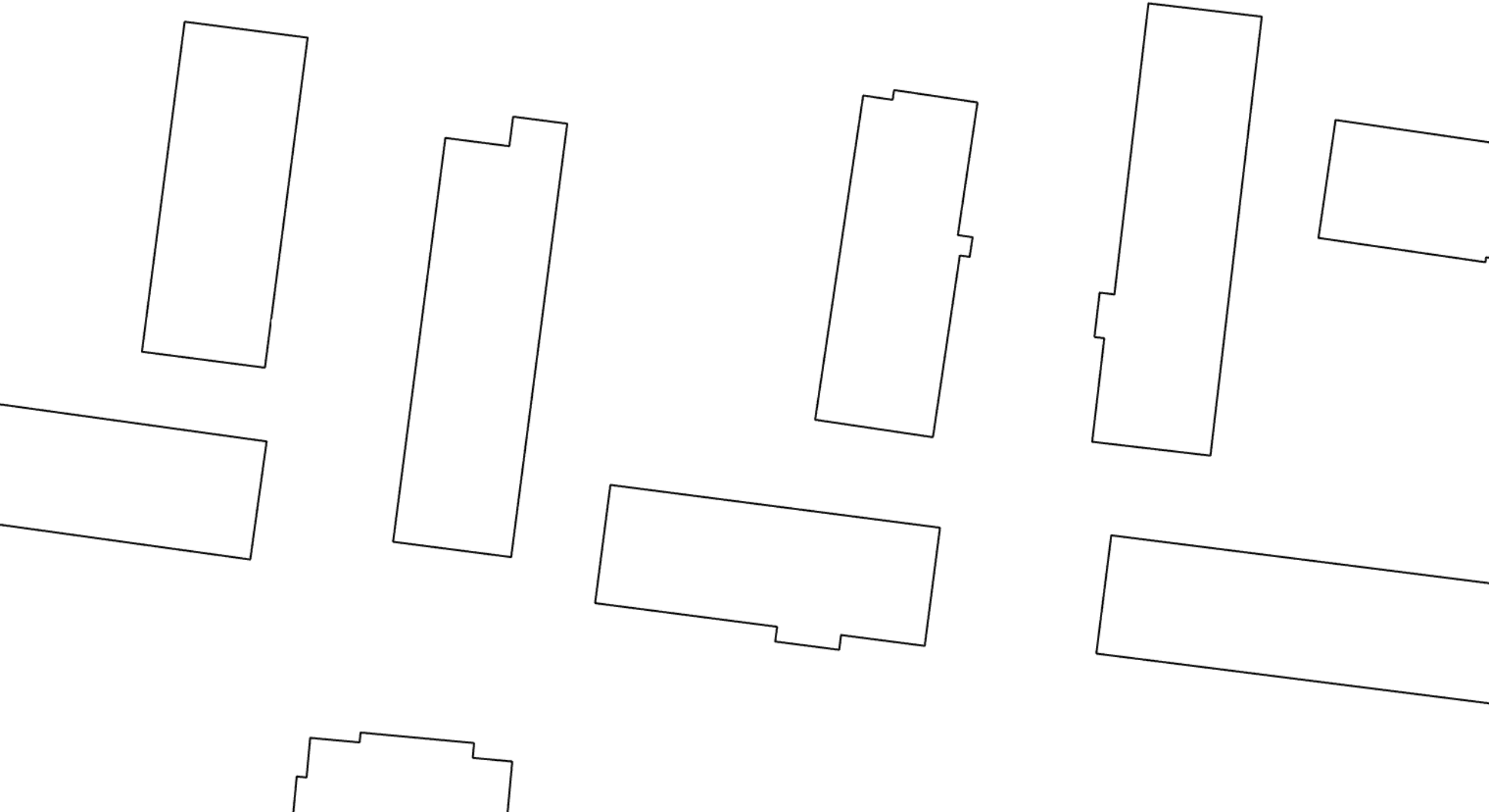
# Building Footprints

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





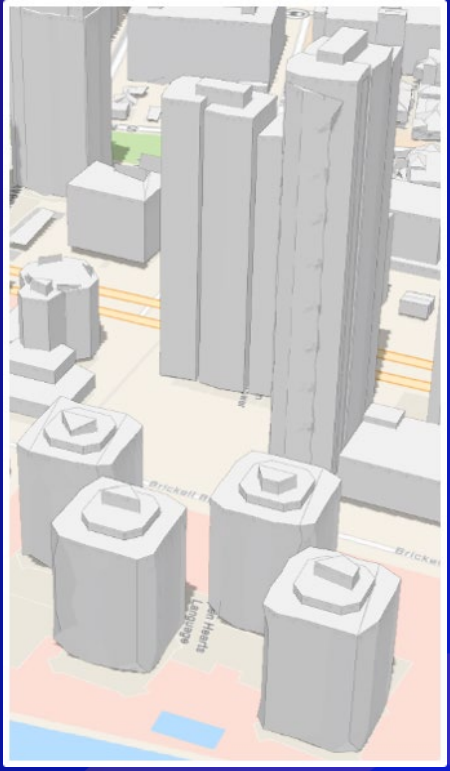


# 3D Building Shells

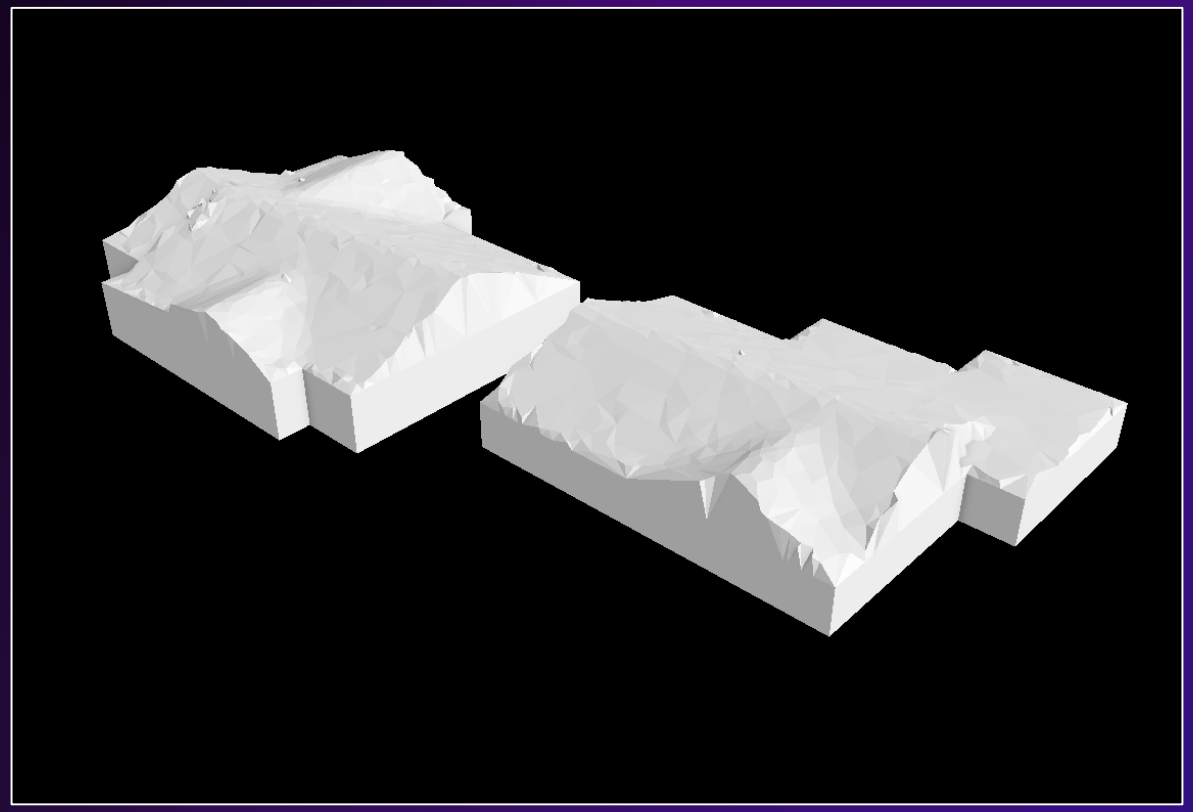
- Simple extrusion (LOD1)
  - Height obtained from lidar
  - Flat roofs
- Rooftop forms (LOD2)
  - Triangulated rooftops vs. Procedurally generated
- Building sides
  - Majority of aerial lidar does not reliably capture sides
  - As consequence sides are modeled using simple vertical walls

LOD1
Simple Extrusion

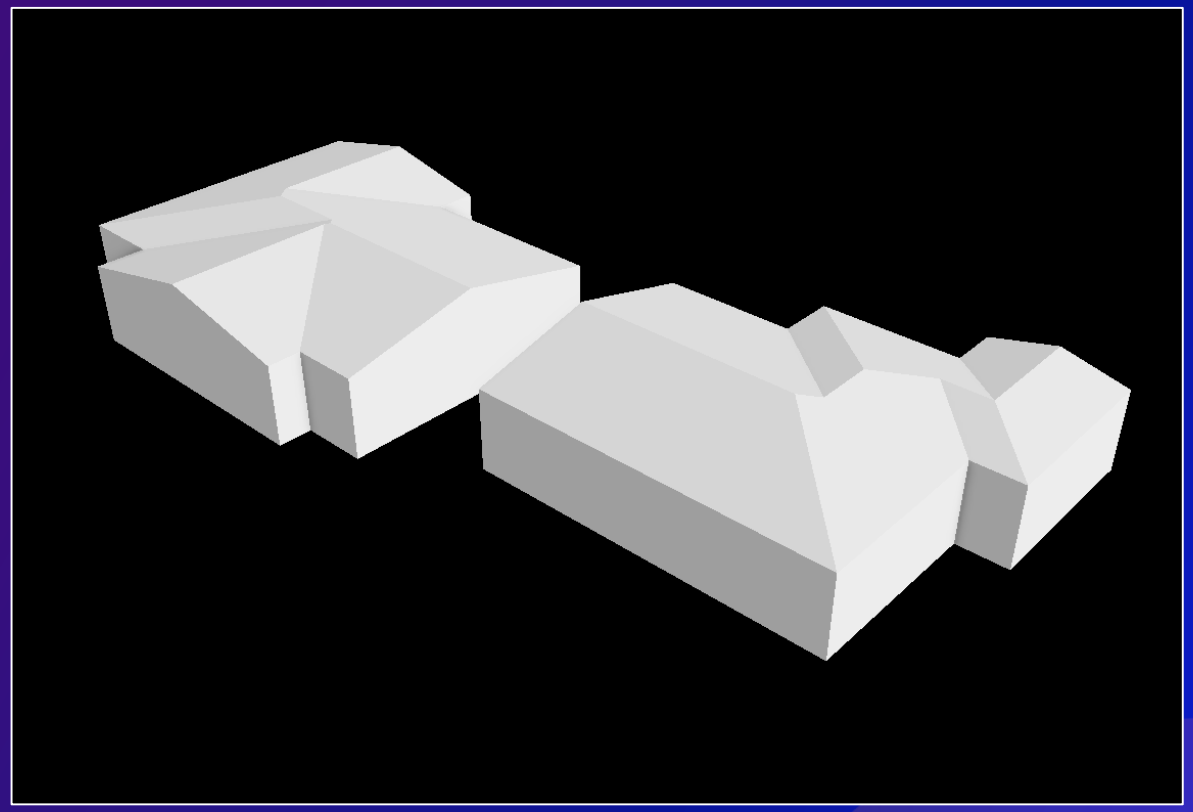

LOD2
Roof Form



# 3D Building Shells

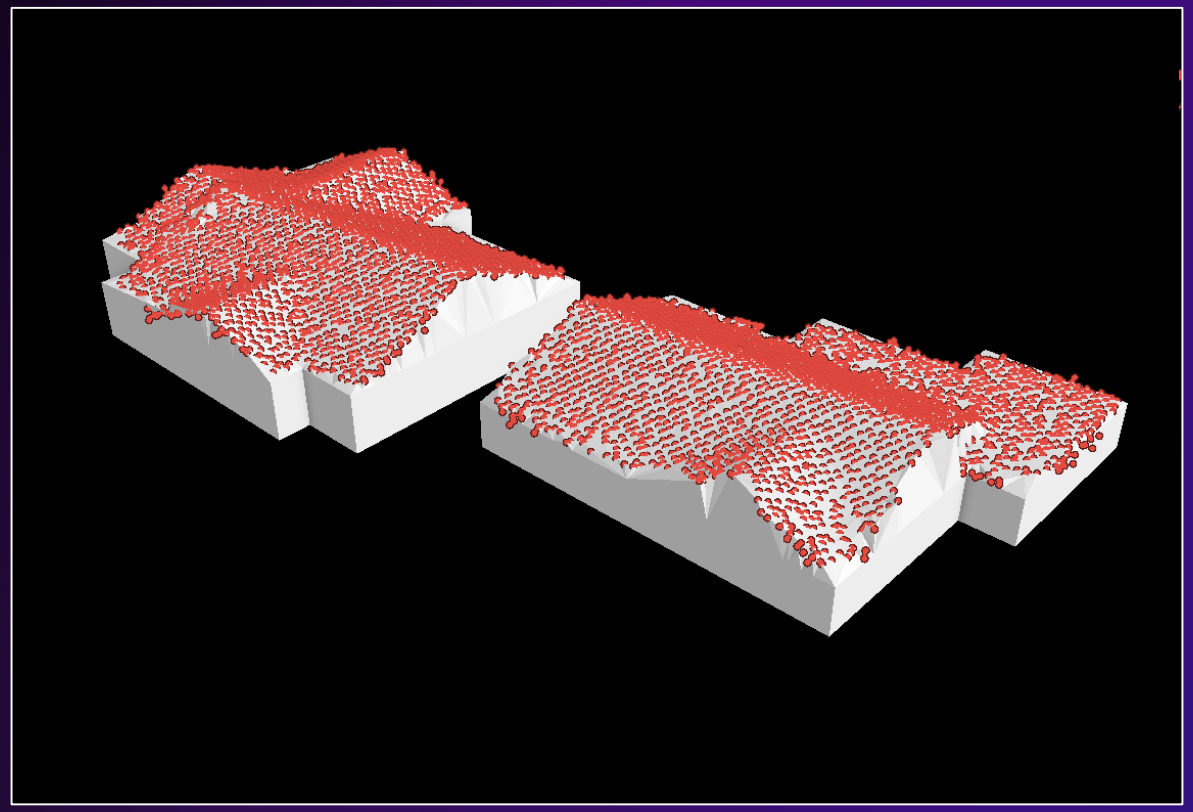


LAS Building Multipatch tool
(triangulation)

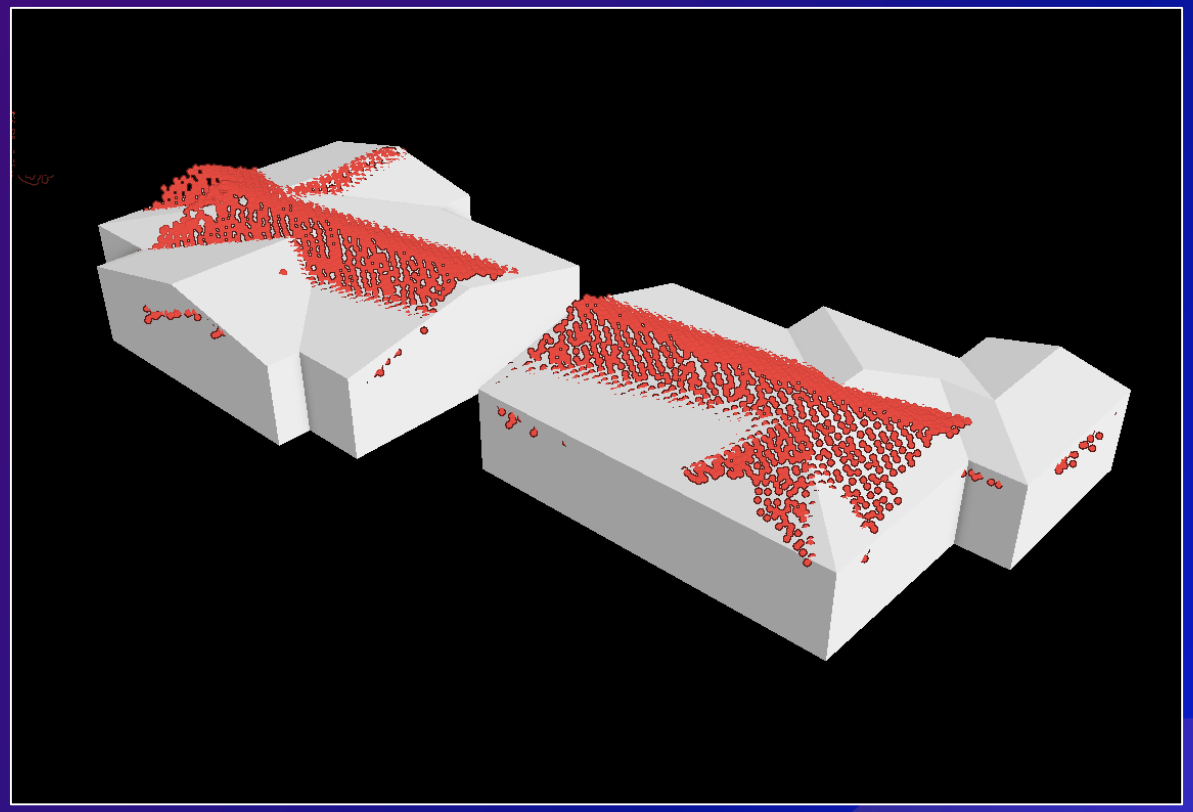


3D Basemaps Solution
(procedural symbology)

# 3D Building Shells



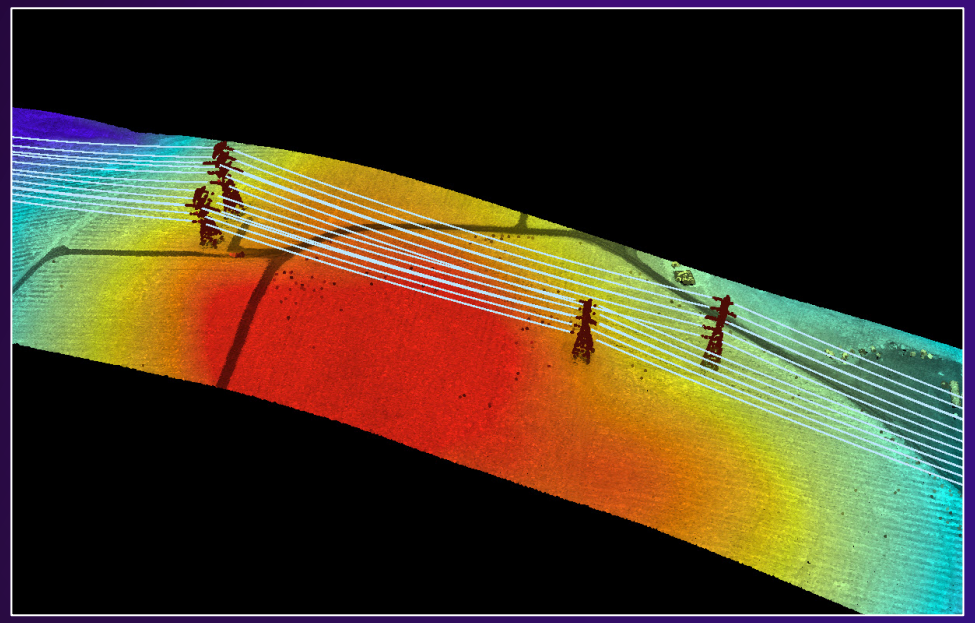
LAS Building Multipatch tool
(triangulation)

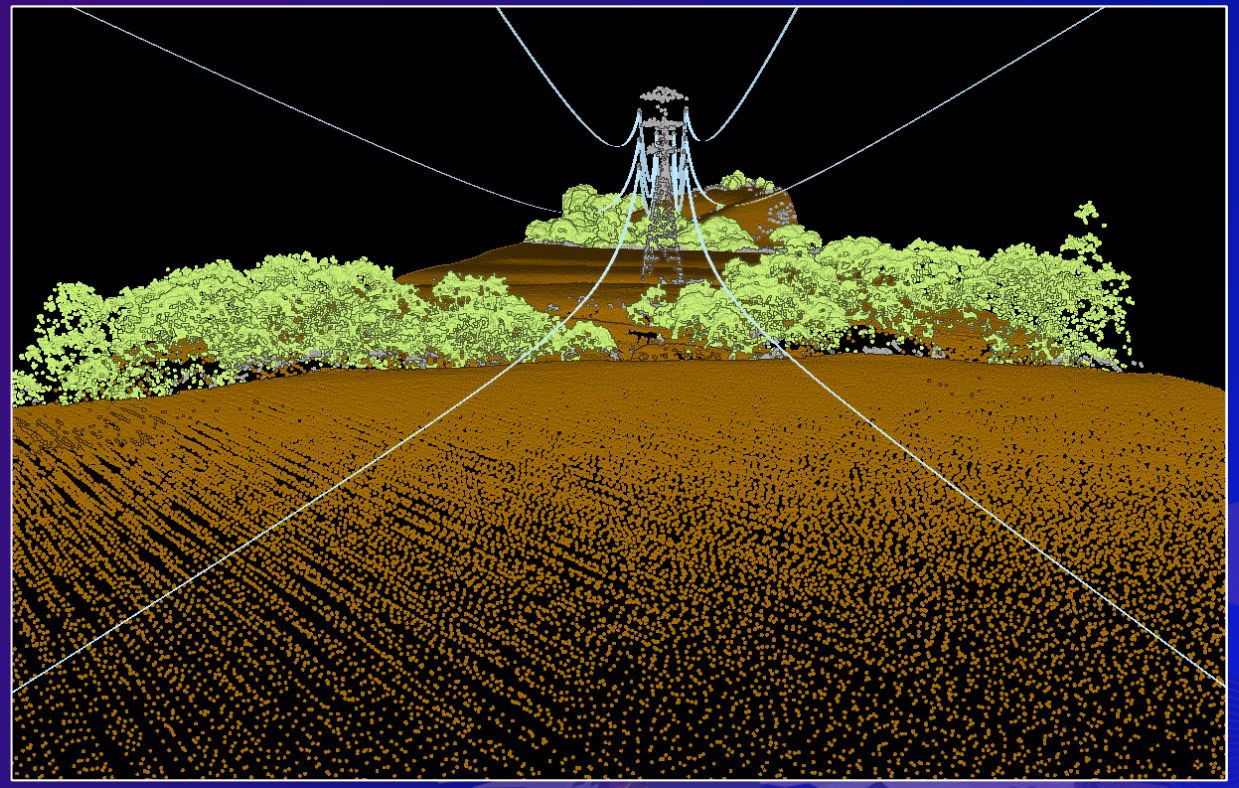


3D Basemaps Solution
(procedural symbology)

# 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 an 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

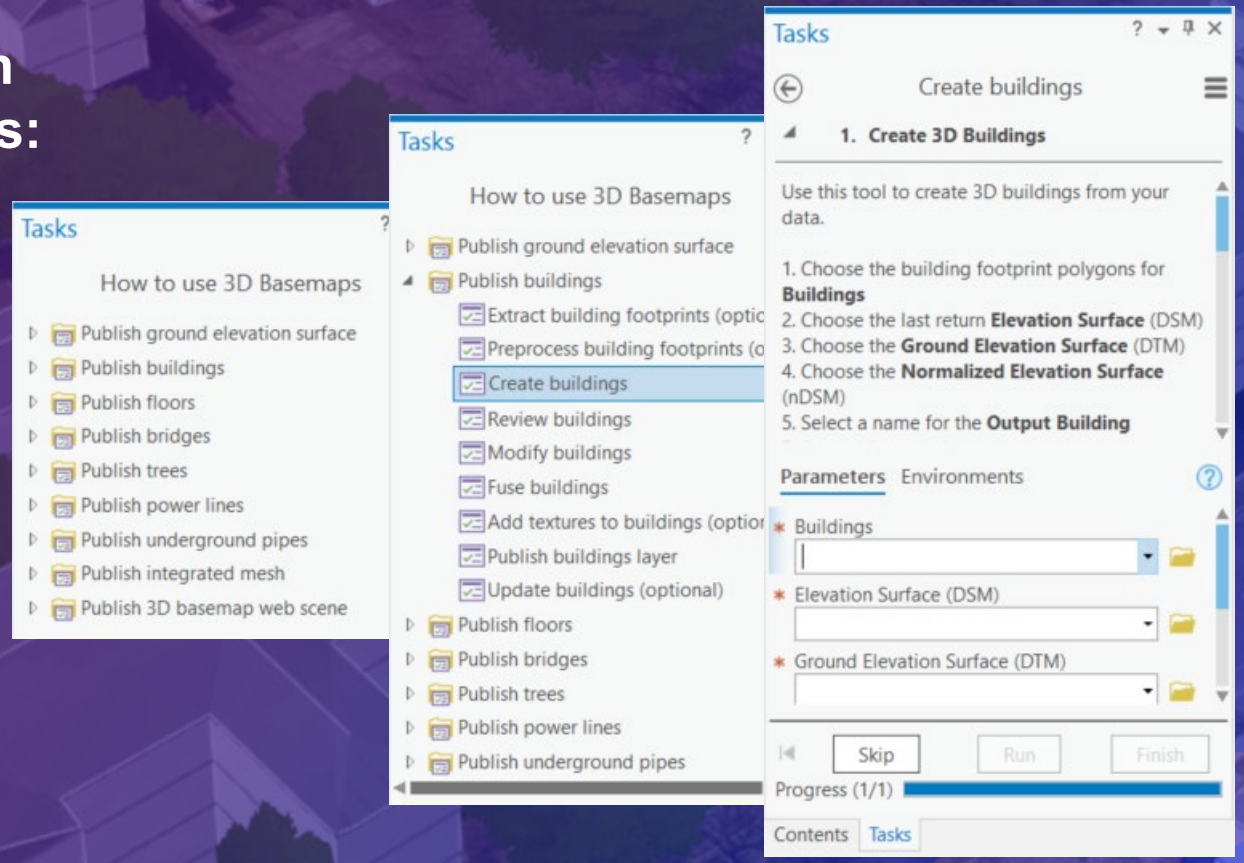
Publish bridges

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®