S9255
Reconstruction of 3D Building Models from Aerial LiDAR with AI

Dmitry Kudinov
Sr. Data Scientist
Esri Inc.
Leveraging the Power of Geography . . .

to Make Better Decisions

A Framework and Process

Geographic Knowledge

Data Management & Integration

Visualization & Mapping

Analysis & Modeling

Planning & Design

Decision-Making

Action

Measuring

Analyzing

Understanding

Collaborating

Leveraging the Power of Geography . . . to Make Better Decisions
GIS Is Advancing Rapidly
Integrating and Leveraging Many Innovations

Data
Computing
GIS Innovation

Web GIS
Easier, Open, and Accessible

Expanding the Power of GIS
GIS Is Advancing Rapidly
Integrating and Leveraging Many Innovations

Web GIS
Easier, Open, and Accessible

Data
Remote Sensing
Scientific Measurements
Demographics
Drones
Imagery
Traffic
3D
Crowdsourcing
Real-Time
IoT
Full-Motion Video

Computing
Mobile
Big Data
Faster
Cloud
SaaS
Web Services
Distributed Computing
Microservices
Containerization
Virtualization

GIS Innovation
Real-Time
Data Exploration
Scripting
3D Visualization
Apps
Smart Mapping
Analytics
Predictive Modeling
Content
Distributed Architecture
Geospatial AI

Expanding the Power of GIS
GIS Is Advancing Rapidly
Integrating and Leveraging Many Innovations
GIS Is Advancing Rapidly
Integrating and Leveraging Many Innovations

GIS Innovation
- Web GIS
- Distributed Architecture
- Content
- Real-Time Data Exploration
- Analytics
- Imagery
- Scripting
- 3D Visualization
- Smart Mapping
- Apps
- Geospatial AI
- Predictive Modeling
- Networks
- Virtualization
- Containerization
- SaaS
- Cloud
- Faster
- Big Data
- IoT
- Full-Motion Video
- Real-Time
- Traffic
- Drones
- Imagery
- Demographics
- Lidar
- Mobile
- Distributed Computing
- Microservices
- Networks
- Remote Sensing
- Scientific Measurements
- Weather
- Crowdsourcing
- Easier, Open, and Accessible

Expanding the Power of GIS
Deep Learning, Machine Learning, & Data Science

ArcGIS Includes Machine Learning . . . and Integrates Deep Learning & Data Science

New and Improved

- Clustering
- Prediction
  - Classification
  - Regression
  - Interpolation
- Object Identification

Coming

- Feature Extraction
- Site Selection
- Event Prediction
- Image Analysis

Deep Learning

Multiple Frameworks & Platforms

- CNTK
- TensorFlow
- scikit-learn
- Microsoft
- IBM Watson
- Amazon

Data Science

Open Science Libraries

- R Integration
- pandas
- SAS
- Jupyter
3D models of cities: valuable and expensive

- Third dimension is important for urban planning, design and aesthetics, insurance, taxation, safety, damage management, etc.

- Creating accurate 3D building models at scale is expensive and manually intensive.

- Common source:
  - Airborne LiDAR, and
  - Triangulated 3D meshes from oblique imagery.
Two approaches to creation and maintenance:

1. **High fidelity models of historical buildings and cityscape features which are considered stable and never / rarely undergo any modifications.**
   - Manually crafted models,
   - Often have designated budgets for creation,
   - Rarely updated.

2. **Schematic-like models of commercial, industrial, residential zones which develop and change often.**
   - Have the largest area,
   - Need to be re-evaluated periodically for taxation and regulatory purposes,
   - Must be evaluated first and fast in case of a natural disaster, e.g. earthquake,
   - The process must be quick, accurate enough, and **cost effective**.
Unlabeled point clouds and continuous meshes

• LiDAR point clouds always have X-Y-Z, but sometimes may come with additional attributes like Intensity and RGB.

• 3D triangulated meshes, although have much lower vertex density than LiDAR, often have high-resolution RGB textures attached.

- Neither sources have building points/faces labeled.
- How to extract buildings from such sources?
Case Study: Miami-Dade County project

1. Raw data source: airborne LiDAR ~15 points per square meter resolution.
2. Point cloud is rasterized to a single channel raster, with values representing the height above the local ground elevation (Normalized Digital Surface Model / nDSM).
3. Human editors manually digitize 2D roof segment polygons around buildings from the nDSM raster.
4. ArcGIS Pro is used to automatically extrude the complex building shapes out of manually digitized roof segments.
Case Study: Miami-Dade County project

- **Step 3**: Human editors manually digitize 2D roof segment polygons around buildings from the nDSM raster.
  - Over 3,000 man hours were spent on digitizing about 213,000 roof segments covering the area of 200 square miles.
  - The average speed for a human editor is ~70 roof segments per hour.
Case Study: Miami-Dade County project

Can we make the process more efficient?

- Reduce the amount of manual labor,
- Increase the productivity,
- Improve the quality of 3D building models,
- Reduce the cost of 3D content acquisition.
Case Study: Miami-Dade County project

- Using Mask R-CNN for helping human editors with the Step 3:
  - Automatic detection and classification of roof segment masks in the input nDSM raster.
  - All seven roof types are detected.

- Although not as accurate as humans, it is much faster: 60,000 (!) roof segment masks per hour from a single Nvidia GP100 GPU.

- Raw predictions masks are regularized using automated tools before the extrusion.
Using ArcGIS Pro:
- To convert Point Cloud into nDSM,
- To create Training and Validation sets,
- To run inferencing and digest results,
- To perform the 3D multipitch extrusion and procedural texture application,
- To calculate floor count and square footage,
- To allow for manual high-fidelity edits of the resulting 3D models,
- To publish resulting models as a 3D Scene Service.

Using ArcGIS Online / Portal to host and manage access for multiple clients and applications.

Case Study: Miami-Dade County project
Demo

Miami-Dade County

- Training Data Creation
- Inferencing
- 3D extrusion
- 3D Web Scene Service
But there are other ways to work with point clouds...

Today ArcGIS allows for reconstruction of buildings directly from point clouds using traditional algorithms and released GP Tools:

To get building rooftop classified points:

1. **ClassifyLASGround** (if ground not already classified)
2. **ClassifyLASBuilding**

To get building footprints:

3. **LASPointStatisticsAsRaster**
   - with LAS layer filtered on class 6 (building)
   - using the ‘Most Frequent Class Code’ option
4. **RasterToPolygon**
   - Turn off the Simplify polygons option

5. **EliminatePolygonPart** to remove small holes (could alternately have performed some manipulation on the raster side for this)
6. **RegularizeBuildingFootprint** to straighten things out.

To extract shells:

7. **LASDatasetToRaster** with input LAS layer filtered on class 2 points to make DEM
8. **LASBuildingMultipatch**
But there are other ways to work with point clouds…

Such models contain a large number of faces and are extremely hard to edit manually after, so it’s better to have them produced of the highest quality possible.
But there are other ways to work with point clouds…

…also relies heavily on accuracy of the labels assigned to points in the source point cloud:

- Ground / Water,
- Buildings,
- Vegetation / everything else.

- Traditional deterministic tools like **ClassifyLASBuilding** have a hard time working in areas with lots of vegetation around buildings
But there are other ways to work with point clouds…

…also relies heavily on accuracy of the labels assigned to points in the source point cloud:
- Ground / Water,
- Buildings,
- Vegetation / everything else.
- Traditional deterministic tools like **ClassifyLASBuilding** have a hard time working in areas with lots of vegetation around buildings
But there are other ways to work with point clouds…

…also relies heavily on accuracy of the labels assigned to points in the source point cloud:

- Ground / Water,
- Buildings,
- Vegetation / everything else.

- Traditional deterministic tools like **ClassifyLASBuilding** have a hard time working in areas with lots of vegetation around buildings.
Can we use DL to label point clouds?

Deep Learning and Point Clouds, feature learning from irregular domains:
- Harder to deal with because point clouds are irregular and unordered, direct use of Convolutions does not work.
- Good news: multiple developments, DL architectures, and papers in recent years: PointNet, Graph Convolutional networks, Deep Sets, PointCNN, etc.
PointCNN and LiDAR point clouds

- Trained on 1.8B X-Y-Z points from Amsterdam.

- 0.97 accuracy on Validation set after 6.5 hours of training on QUADRO V100.

- Tested on city of Utrecht.
PointCNN and LiDAR point clouds

- Trained on 1.8B X-Y-Z points from Amsterdam.
- 0.97 accuracy on Validation set after 6.5 hours of training on QUADRO V100.
- Tested on city of Utrecht.
PointCNN and LiDAR point clouds

- Trained on 1.8B X-Y-Z points from Amsterdam.
- 0.97 accuracy on Validation set after 6.5 hours of training on QUADRO V100.
- Tested on city of Utrecht.
We used PointCNN to classify a point cloud for better results

- Much lower noise level in RANSAC reconstructions created with point cloud labeled by PointCNN model.

- PointCNN segments point cloud into multiple classes in a single pass.

- Only 3.5M trainable parameters.
Then we can work with 3D Meshes, right?

1. Feeding all the input mesh vertices, with additional Monte-Carlo sampled points to PointCNN for segmentation.

2. Applying segmentation back to the mesh.

3. Boundary condition resolution on the way from point cloud to triangulated mesh.

4. Better results with face-normal vectors and RGB features.

5. Works OK even if was trained on a true LiDAR point cloud(!).
Mask R-CNN and PointCNN: what to look for?

- **Mask R-CNN sensitivity / bias:**
  - Architectural styles
  - LiDAR scanner: point density

- **PointCNN sensitivity / bias:**
  - LiDAR scanner: point density, intensity, RGB
  - Sampling technique when segmenting 3D Meshes

- Want a universal model?...
- …Then bring more training samples.
- …BTW, synthetic training data is an option too.
Want to learn more?

https://goo.gl/3uaRJi
ArcGIS as the primary Spatial Data-Science Platform

- **Power of GIS:**
  - Creation of high quality Training sets using Desktop and Cloud products.
  - Unmatched tools for QA of Spatial DL models.
  - Direct import of the inference results into the platform as raster or feature classes, or hosted services.

- **Full integration:**
  - Hosting DL models as Portal items and services.
  - Integration with Deep Learning frameworks for local and Cloud GPU inferencing.

- **Products:**
  - ArcGIS Pro 2.3
  - Image Server and Raster Analytics 10.7
  - Python API 1.6
  - More to come…