Efficient object classification from airborne LiDAR

Jorge Martínez Sánchez

Centro de Investigación en Tecnoloxías da Información (CiTIUS)

Universidade de Santiago de Compostela

Doctoral Meeting. CiTIUS. March 2017.

Table of contents

Table of contents

Problem Description

Provide updated information of the Earth's surface:

- Terrain
- Vegetation
- Buildings
- ...

Object classification (landinfo.com)

Fast enough to be used in emergency services

- Fire-extinguishing
- Rescue missions
- ...

MH-60S Seahawks (aviafora.com)

Accurate classification is a studied topic but, what about performance?

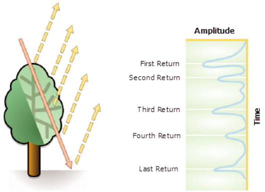
LiDAR Technology

Light **D**etection **A**nd **R**anging

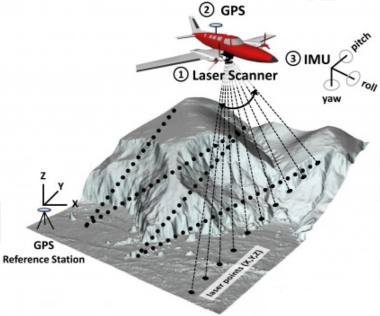
Emit a beam and measure the time and intensity to return to its source

Advantages over aerial imagery:

- Records elevation (**3D** point cloud)
- Tolerant to atmospheric conditions
- Beam penetration on vegetation
- Reduced cost on large areas



Beam penetration (arcgis.com)



Scanning overview (kfki.de)

General Overview

Unsupervised object classification

PREPROCESSING → SEGMENTATION → CLASIFICATION → Ground / Vegetation / Buildings / Roads

$\overrightarrow{F} = \{x, y, z, I, rn, nor\}$

Feature vector

$\overrightarrow{F}, \overrightarrow{n}$

$\overrightarrow{F}, \overrightarrow{n}, group$

$\overrightarrow{F}, \overrightarrow{n}, group, type$

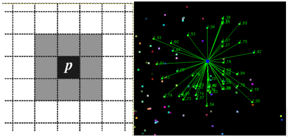
Some critical parts parallelized with OpenMP

Table of contents

Data Organization

Irregularly spaced point cloud

Expensive neighbor search

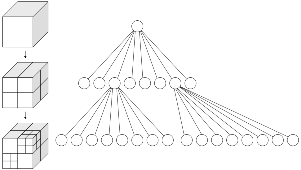


Neighbors search 2D vs 3D

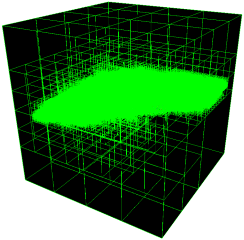
$\rightarrow$ **3D Space partitioning**: octree

Split into 8 children recursively

Neighbors search in $O(\log n)$ time



Octree diagram (wikipedia.org)



Partitioned point cloud

Normal Vector Estimation

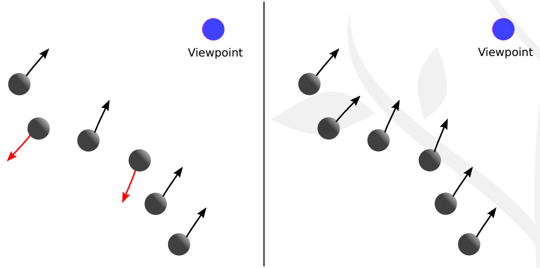
Calculated directly from the point cloud with PCA

Neighborhood definition: 3D voxel with radius r

Problem: orientations given by PCA are not consistent

$\rightarrow$ **Correct orientations**

- Airborne LiDAR is 2.5D not full 3D
- Artificial viewpoint = point with highest Z
- Each point P must satisfy $\overrightarrow{n_P} \,.\, \overrightarrow{dir_{PV}} > 0$



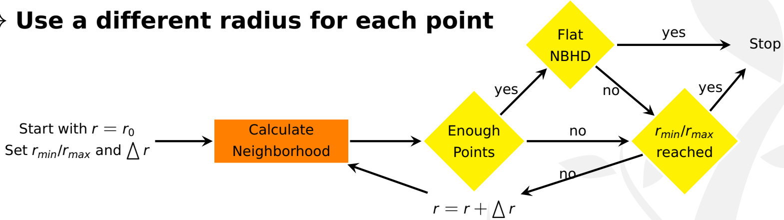
Normal vector orientations: Inconsistent in red (left) Corrected (right) result

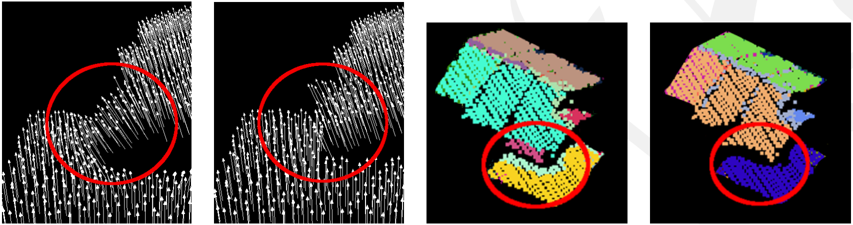
Adaptive Neighborhood

What voxel radius? $r_0 = 1.5 \times \mathrm{APS}$ (Average Point Spacing)

Same for all points? Bad when density varies or at spatial boundaries

$\rightarrow$ **Use a different radius for each point**





(a) (b) (c) (d)

(a) Normals with fixed r (b) Normals with adaptive r (c) Segmentation with fixed r (d) Segmentation with adaptive r

Table of contents

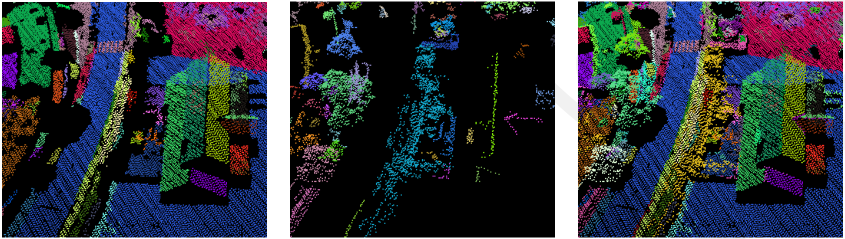
Two-Phase Region-Growing

Simple and relative low computational load

Segment planar and non-planar objects **independently**:

- 1^{o} Phase: segment planar objects
  Point features: $x, y, z, I, \overrightarrow{n}$
- 2^{o} Phase: segment non-planar objects
  Point features: x, y, z, I

Final result = combination of both partial segmentations



(a) (b) (c)

Results of the segmentation: (a) First phase. (b) Second phase. (a) Final result

Epicenter Point

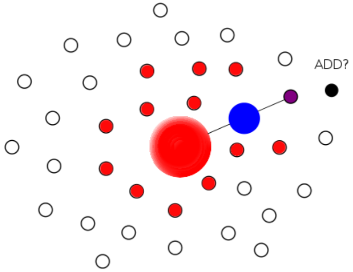
In the growing process, when is a new point added into a region?

Epicenter: Value between centroid region and last added point

Epicenter for a feature x

$$Epicenter_x = R_{\overline{x}} * \sigma + p_x * (1 - \sigma)$$

R = Region, $R_{\overline{x}}$ = Centroid of the region, p_x = last added point, $\sigma \in [0, 1]$ = expansion value



Unsegmented (white), inside R (red), $R_{\overline{x}}$ (big red), Epi_x (blue), p_x (purple), candidate (black).

Controlling expansion

Tunable by σ

$\sigma = 1 \rightarrow$ By centroid

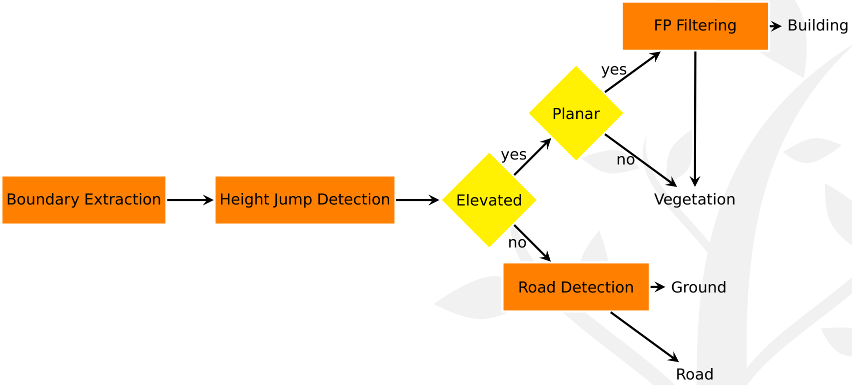
$\sigma = 0 \rightarrow$ By last added

$\sigma = 0.5 \rightarrow$ Trade-off

Table of contents

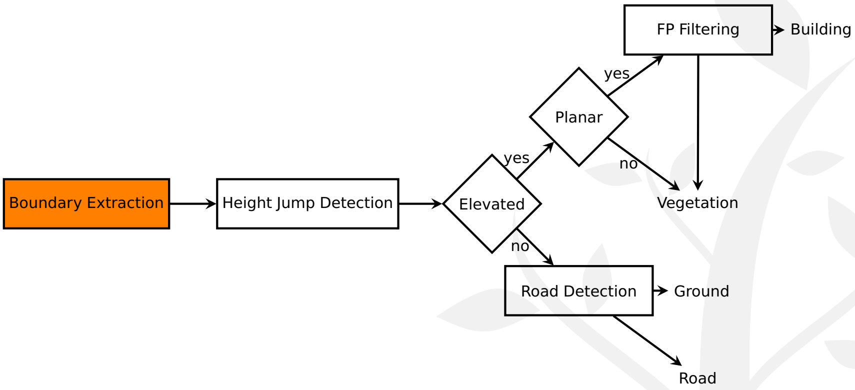
Rule-Based Classification

Four classes: ground, vegetation, building, road.



Scheme of the ruled-based classification

Boundary Extraction

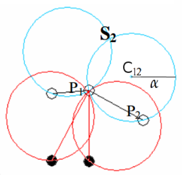


Boundary Extraction

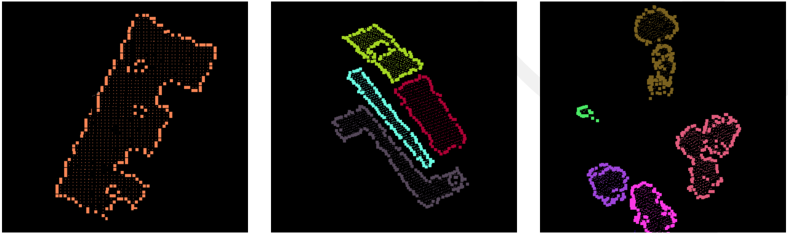
Alpha shapes (Wei, 2008)

Draw a circle using two points and a radius α, if it is empty $\longrightarrow$ boundaries

- Advantages:
  - Avoid costly TIN creation
  - Extracts inner boundaries
  - Fully parallelizable
- Disadvantages:
  - Unordered boundaries (polygon area?)



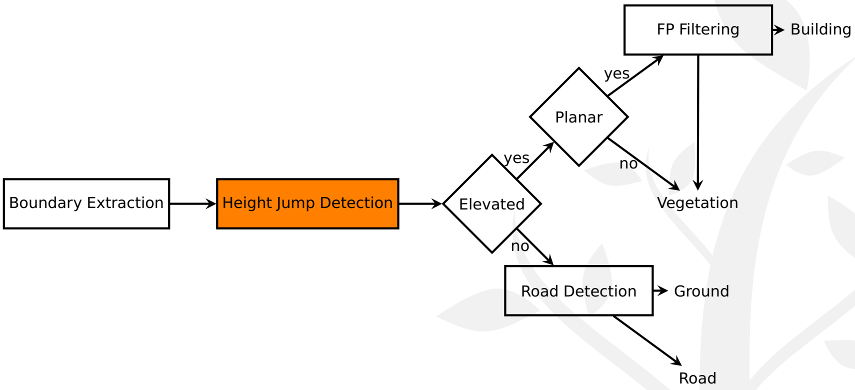
Alpha Shapes judgement



(a) (b) (c)

Clusters and their boundary points (a) Vaihingen building (b) Alcoy building (c) Clusters of trees

Height Jump Detection

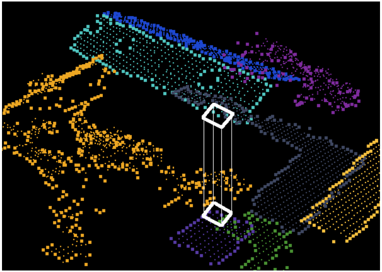


Height Jump Detection

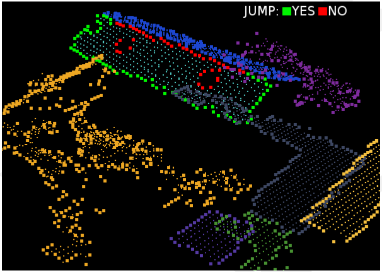
1. Identify neighbors of clusters (MBB overlap)
2. Examine heights of **neighbors at boundaries**

   Determine in which boundaries there are jumps

   If jump ratio is high $\longrightarrow$ cluster is tagged as elevated



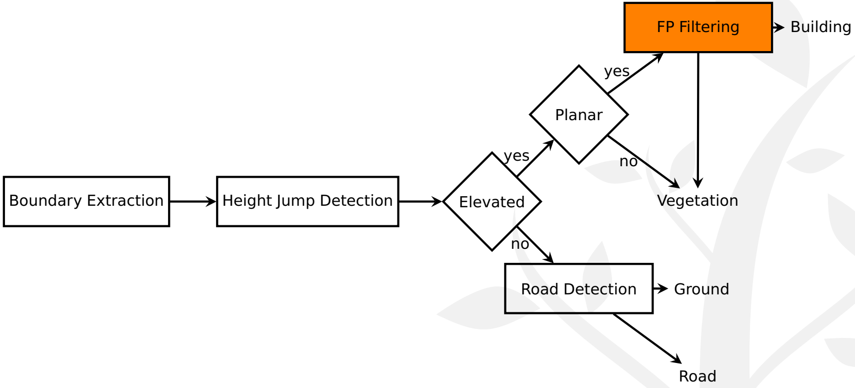
(a)



(b)

(a) Comparing heights inside the boundary window (b) Boundaries that jump (green) and do not jump (red)

Building Filtering



Building Filtering

1. Remove false positives

   -Percent of non-first returns

   Low on buildings

   -Normal vector orientation

   Low z component on walls

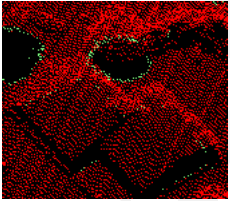
   -Area estimation

   Minimum area of 15 m^2

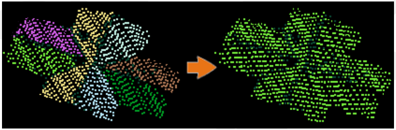
2. Refinement

   -Merge roof planes

   -Recover joint points

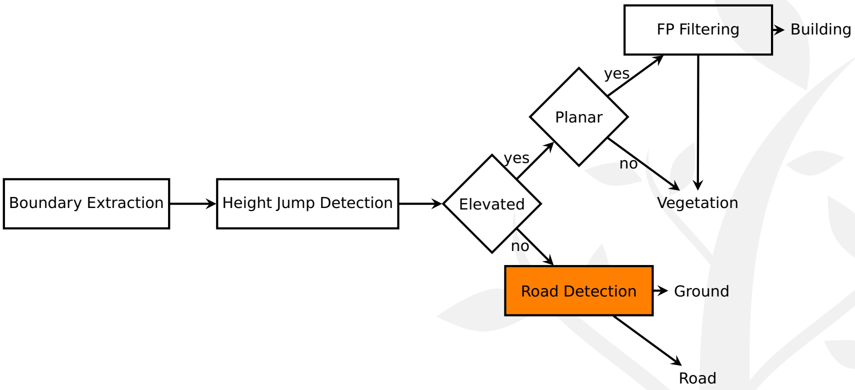


First returns (red) Second returns (green)



Merging roof planes

Road Detection



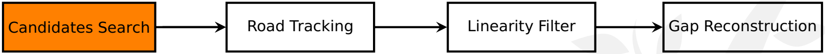
Road Detection

Road assumptions: planar, low intensity, linear, connected.

Point cloud from above colored by intensity

Candidates Search → Road Tracking → Linearity Filter → Gap Reconstruction

Road Detection: Candidates Search

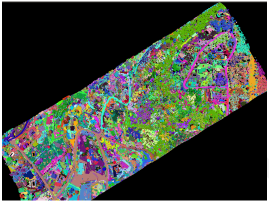


Road Detection: Candidates Search

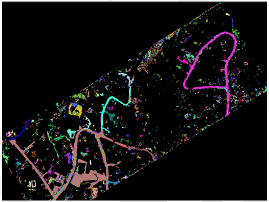
Octree-split approach

1. Divide octants until points share same plane and intensity
2. Map candidate points with the segmented points

(a)



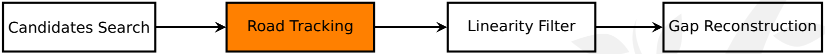
(b)



(c)

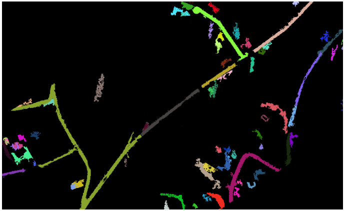
(a) Candidate road points (b) Previously segmented cloud (c) Mapped segmented groups

Road Detection: Road Tracking

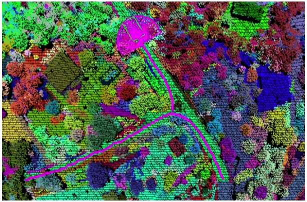


Road Detection: Road Tracking

Problem: Isolated non-road groups, road sections are scattered

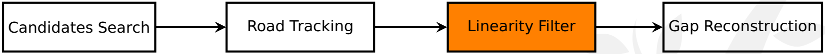



Tracking: set of rectangles that follows point density inside the group



- Non-maximum suppression
- Pending paths queue
- Avoid back steps

Road Detection: Linearity Filter



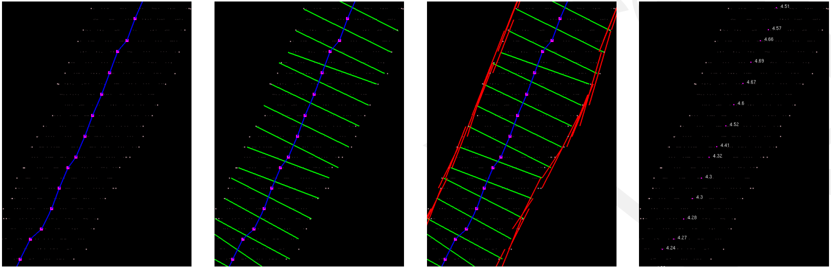
Road Detection: Linearity Filter

Segmented groups must have:

- Parallel boundaries
- Constant width

Easy to examine given the group direction

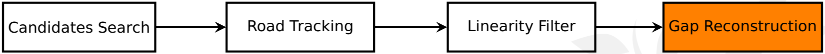
Width = distance from centerline to boundary line



(a) (b) (c) (d)

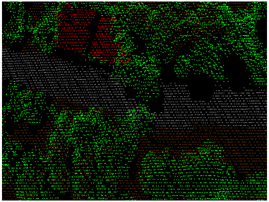
(a) Road centerline (b) Perpendiculars to the centerline (c) Best fit lines at boundaries (d) Widths along the centerline

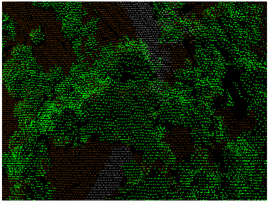
Road Detection: Gap Reconstruction

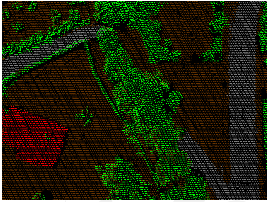


Road Detection: Gap Reconstruction

Problem: Vegetation occlusions create gaps







Active Contours: road reconstruction.

Contour points guided by:

- Internal force: model parameters
- External force: points intensity (high gradient on boundaries)

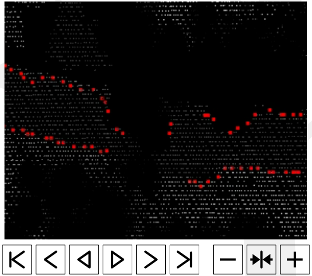


Table of contents

Problems

Problem: Lack of consensus in evaluation

- Evaluation of rasterized 2D image / all 3D points
- Usage of public / non-public datasets
- Usage of same / various LiDAR lasers

Problem: Lack of ground truth

- 3D manual labeling
- X00.000 points
- Field survey?

Only one public* benchmark available for whole cloud classification

ISPRS 3D Semantic Labeling Contest (Vaihingen)

1 dataset with 2 small city blocks, 1 LiDAR laser

Dataset is public, but **ground truth do not**

Results must be submitted for external evaluation.

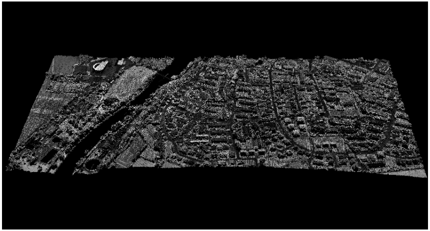
Unsolved topic!

Dataset

Urban area

Public (ISPRS). Vaihingen, Germany

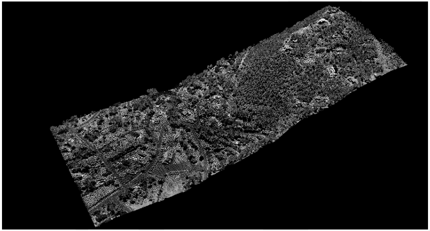
aprox. 4 points/m^2



Rural area

Private (INAER). Alcoy, Spain

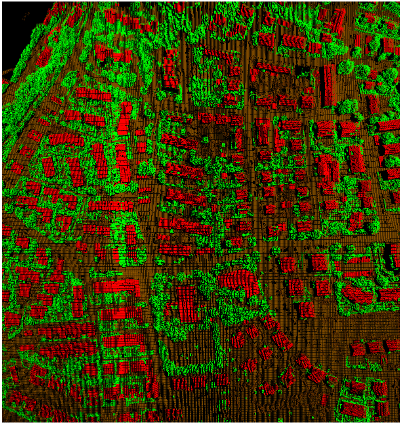
aprox. 9 points/m^2



Measurements

- Segmentation ratio: $\frac{100 \times \sum \text{Points in clusters with size} > 20}{\text{Total points}}$
- Building detection: Completeness $\frac{100 \times \text{TP}}{\text{TP}+\text{FN}}$ Correctness $\frac{100 \times \text{TP}}{\text{TP}+\text{FP}}$
- Execution time on an Intel® Core™ i7-4790 at 3.60 GHz

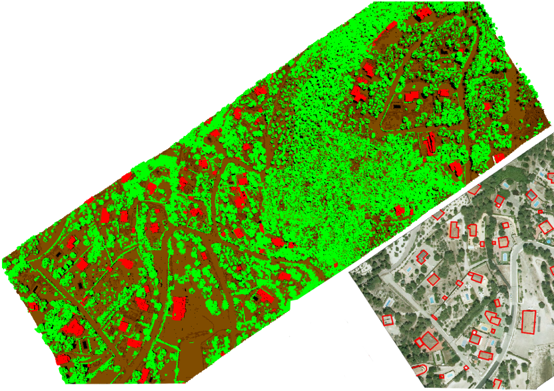
Results I: Vaihingen

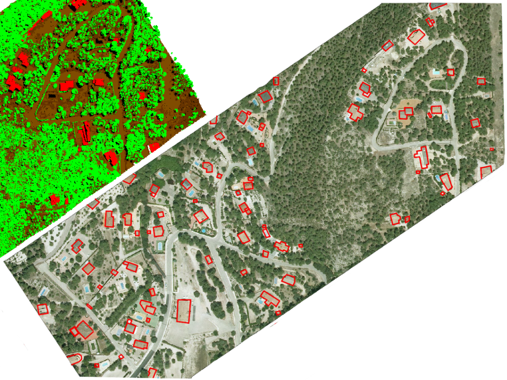


Segmentation ratio	91.46%
Overall time	42.05 s

Labeled buildings	227
Completeness	96.46%
Correctness	98.68%

Results II: Alcoy



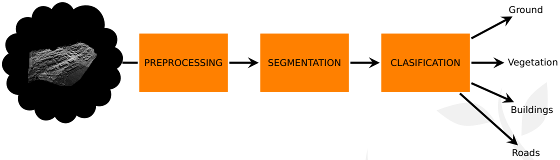


Segmentation ratio	94.33%
Overall time	40.70 s

Labeled buildings	66
Completeness	100%
Correctness	84.85%

Summary

Fast object classifier for airborne LiDAR



Low execution time

- Manage data adequately
- Avoid inefficient algorithms
- Parallelization with OpenMP

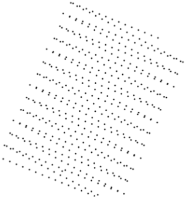
2.5 million points processed in $\sim$**40 s** in a desktop computer.

Future Work

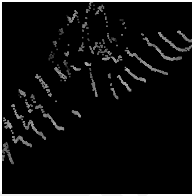
Add more classes: parking lots, power lines, cars...

Add a ground filter to reduce segmentation time

Processing the cloud in **real-time** (laser scan lines)



(a) Seesaw scanning pattern



(b) Visible scan lines on cloud

Problem: raw data recording is not open access.
Handled by proprietary software of laser manufacturers.

Thank you for your attention