THE IQMULUS, URBAN SCENARIO

IQmulus Workshop Bergen

Setember 21, 2016

Roderik Lindenbergh (TU Delft), Mathieu Brédif (IGN), Jan Böhm (UCL), Michel Krämer, Frank Michel (Fraunhofer) and colleagues,

with a contribution from Jinhu Wang (TU Delft)

OVERVIEW

- Motivation and Overview of the Urban Scenario
- Some details on the algorithms within the Scenario
- Processing, results + scalability
- Ongoing extensions
- The Future

SEVENTH FRAMEWORK PROGRAMME

More and more large point cloud data sets

- AHN2 and AHN3 airborne laser scan data sets covering the entire Netherlands
- Continuous TLS
- Here: Laser mobile mapping data

How to process such massive point clouds efficiently?

- **Cloud computing**
  - Many scenario's
  - Low cost comparing to HPC (high performance computing)
- **Level of Detail (LOD) processing**
  - Process only what needs to be processed
  - Maintain flexibility between different LODs

URBAN SHOWCASE 2

URBAN SHOWCASE 2

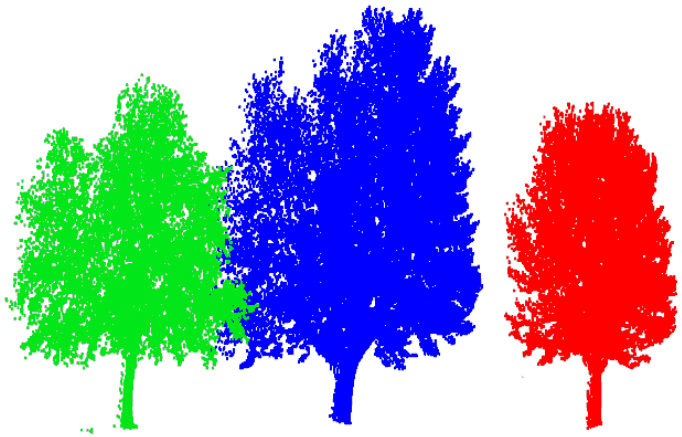
Showcase title:

Individual tree extraction from urban LMMS data

Showcase objective:

Identify single trees in mobile mapping point clouds sampling urban environments by:

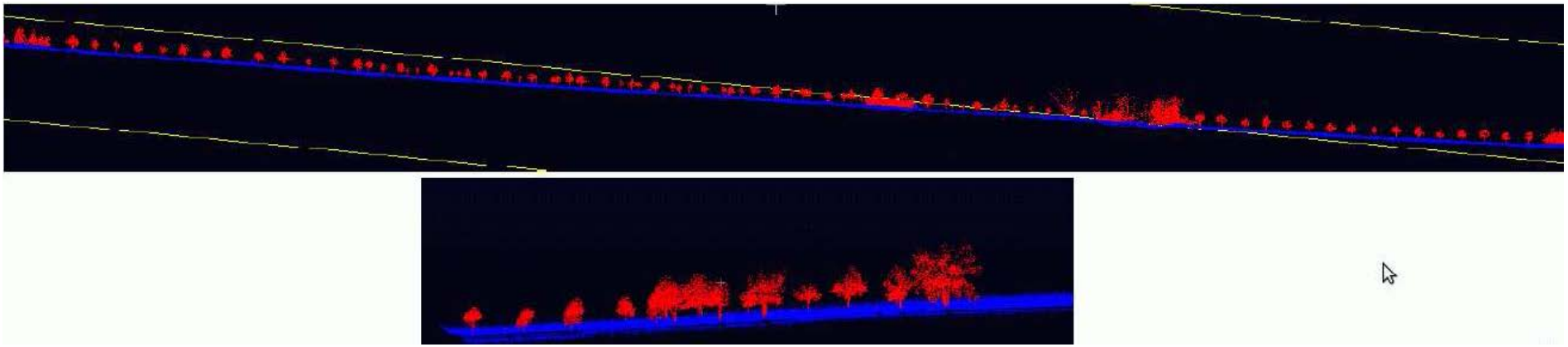
- reporting their **locations**
- outputting all point cloud points sampling trees, organized at individual tree level



Need for tree data

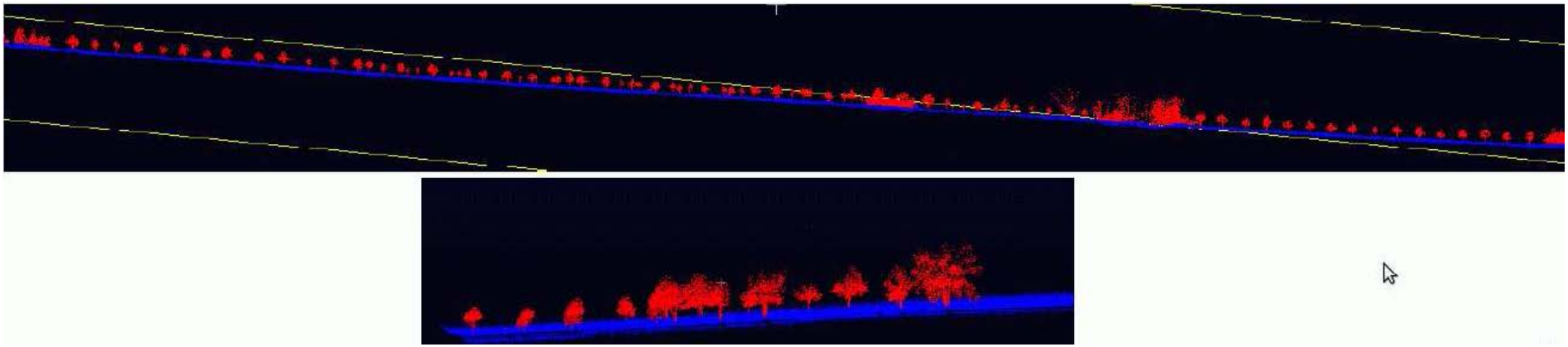
Individual **tree parameters** are high in demand by local and regional authorities for:

- **Street inventory** management and city climate assessment
- **Tree cadaster** and biomass monitoring requirements
- Clearance regulations and **hazard protection**



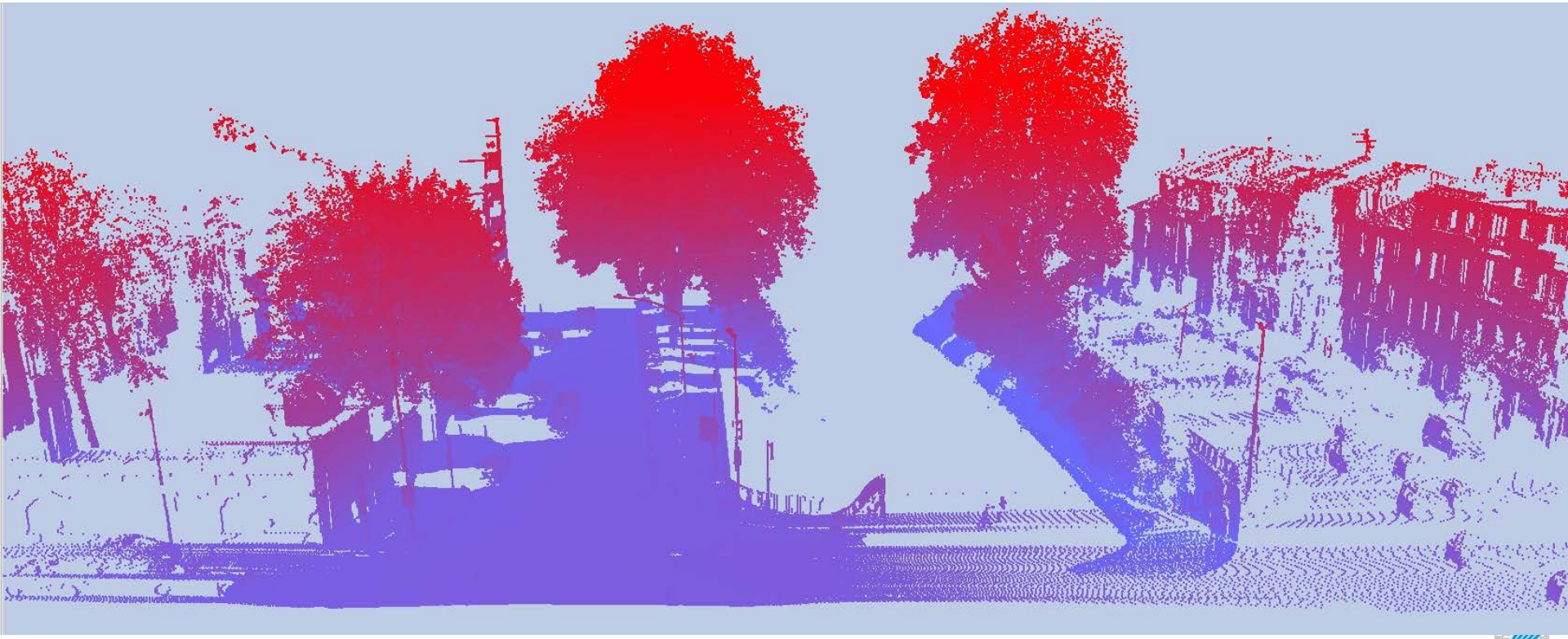
Tree extraction challenges

- Mobile mapping systems acquire point clouds at up to 1.000.000 pts/sec
- Ideally, point clouds acquired during the day are processed over night
- Trees appear in a *variety of sizes and shapes*, are often only *partly scanned*, at different distances from the road, and may be *close to facades, people, cars, street poles*, etc.



US2 INPUT DATA

- 10 km of Mobile Mapping data sampling **Toulouse**
- Provided by **IGN** (Stereopolis system)
- Acquisition time: 121 GB in **2 hours**
- Number of tiles: 517 tiles of ~ 3 million points each



SEVENTH FRAMEWORK PROGRAMME

US2 IN THE DOMAIN-SPECIFIC LANGUAGE

```
for each [PointCloudCollection] do
    # 35 C++ (UCL)
    apply Resampling
        using resolution: 0.3

    # 66 C++ (IGN)
    apply PointCloudDimensionality
        using minSizeNeighourhood: 16
            and maxSizeNeighourhood: 128
            and windowSize: 0

    # 59 C++ (IGN)
    apply MultiObjectClassification

    # 119 C++ (TUDelft)
    apply ConvertPlyToXYZ

    # 26 C++ (TUDelft)
    apply IndividualTreeDetection

    store
end

generate Visualization
```

For all files

Resample point cloud

Calculate point cloud dimensionality

Unroll and parallelize for loop

Classify points

Format conversion

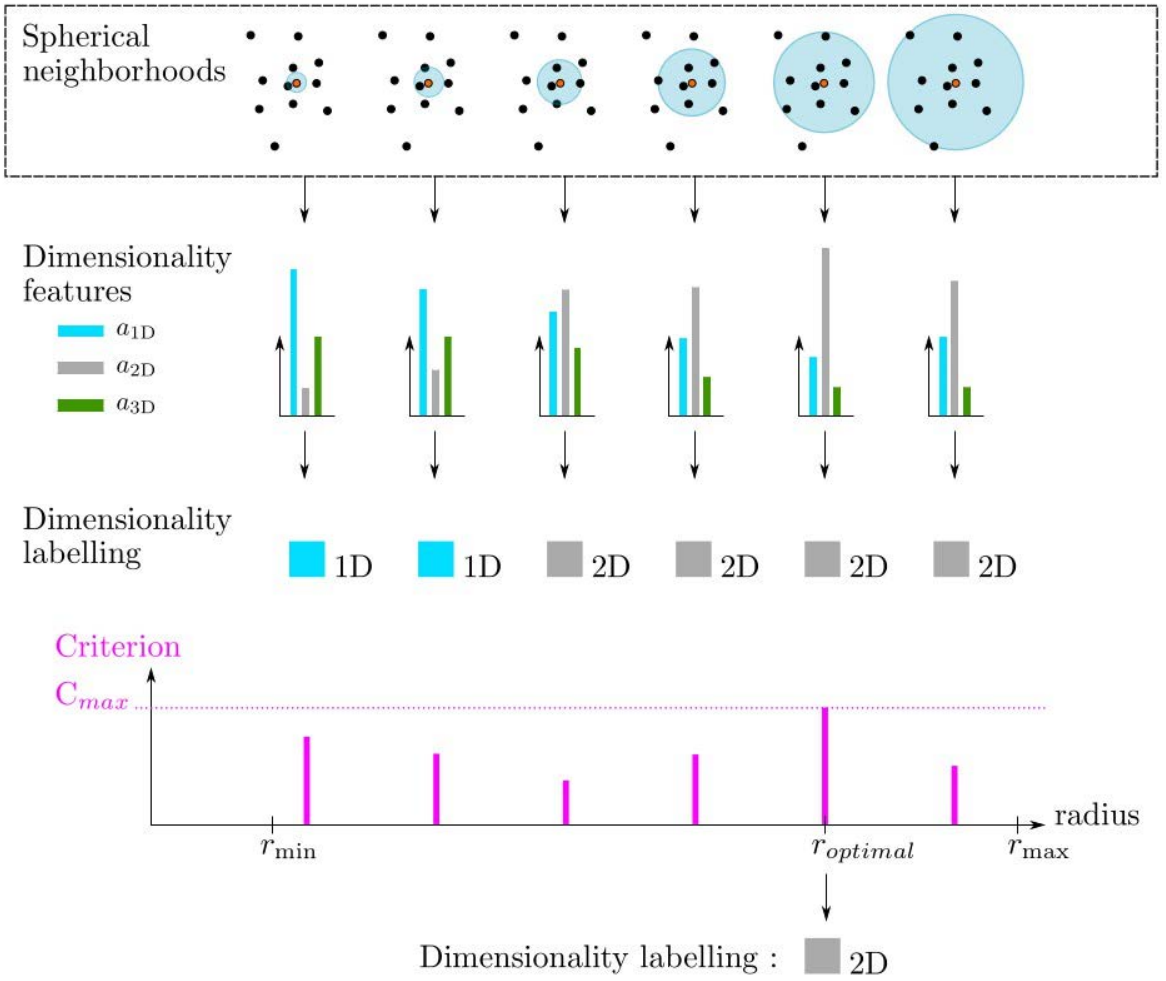
Detect trees

Store detected trees

Prepare visualisation

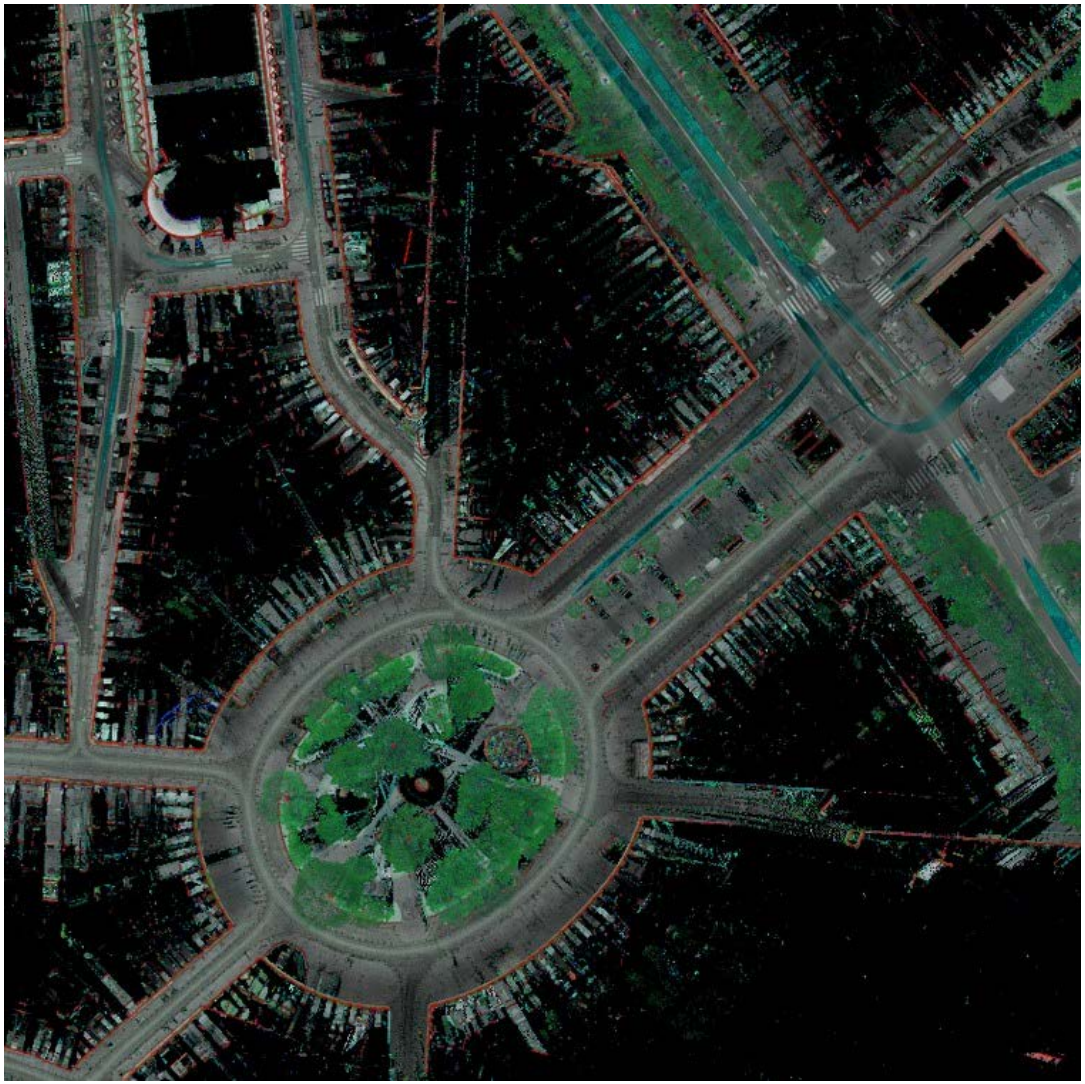
DIMENSIONALITY ANALYSIS

- **Dimensionality features (PCA) [Demantké 2011]**
  - Linear (1D)
  - Planar (2D)
  - Scattered (3D)
- **Adaptive neighborhood**
  - Automatic selection of the most meaningful neighborhood size !
- **Computing time**
  - <1min / million points



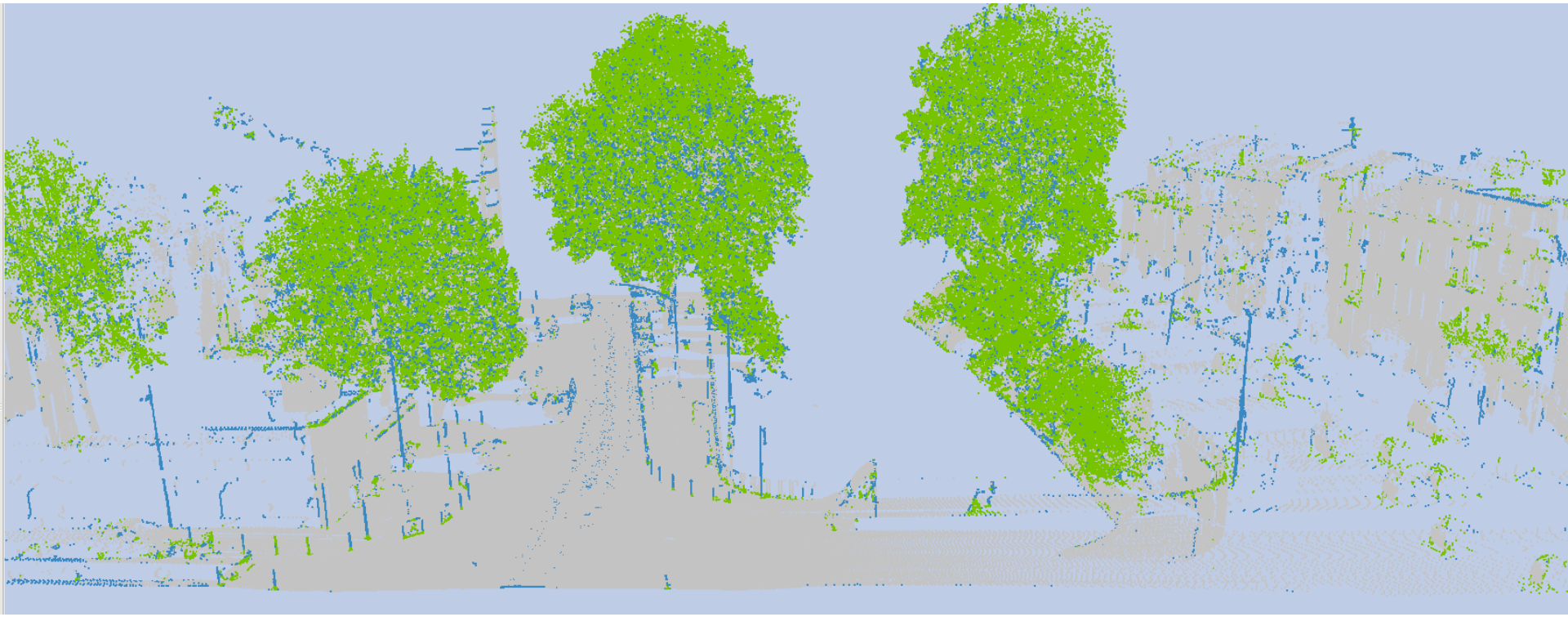
DIMENSIONALITY ANALYSIS

- Vertical Planar
- Horizontal Planar
- Vertical Linear
- Horizontal Linear
- Scattered



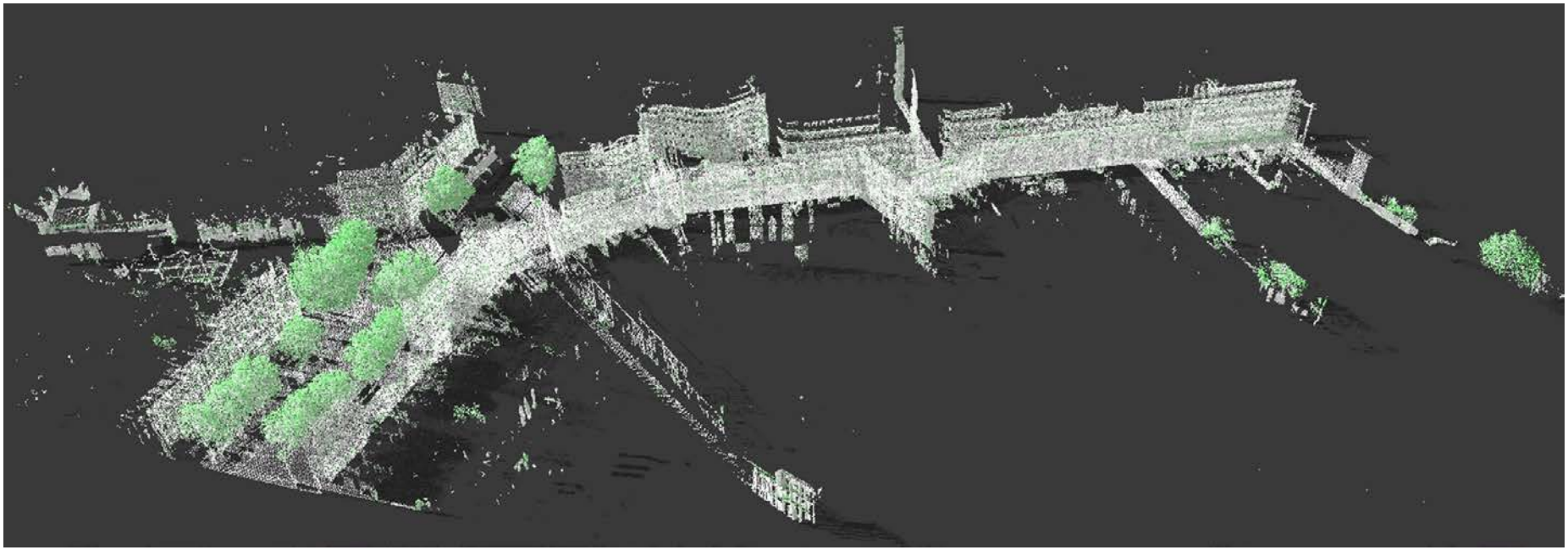
DIMENSIONALITY ANALYSIS

Linear (1D) Planar (2D) Scattered (3D)



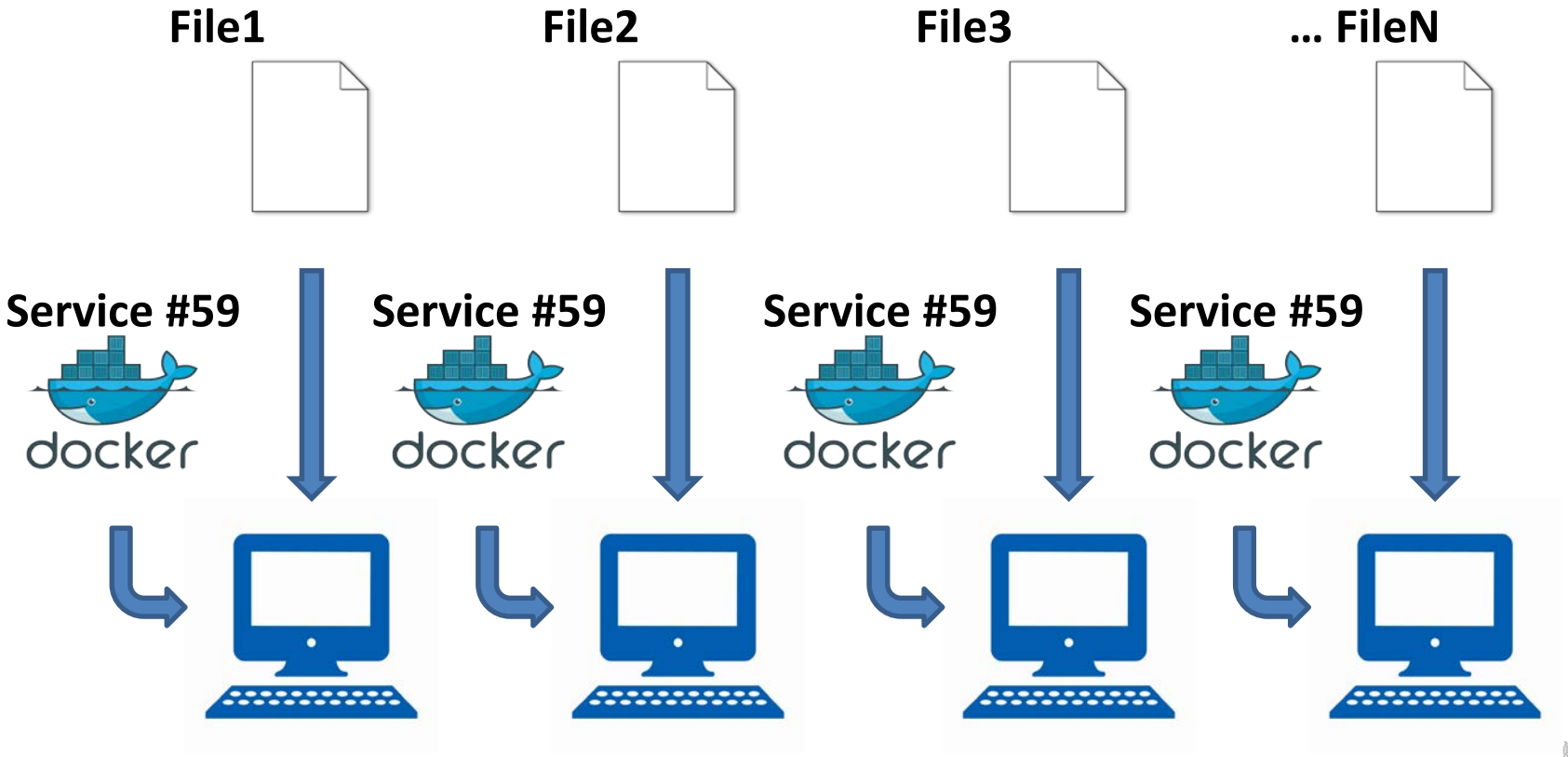
TREE/NON TREE CLASSIFICATION

- Feature extraction based on x,y,z and **dimensionality features** [Weinmann 2014]
- **Random Forest** supervised classifier
- **Regularization** : majority voting within the optimal dimensionality neighborhood
- Computing time : <1min / million points, ~80% accuracy



DISTRIBUTING OVER THE INFRASTRUCTURE

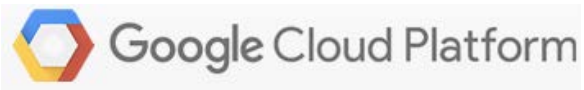
- Streaming Processing: read single data point in loop
- Single CPU implementation
- Parallelizing over multiple files on separate nodes



APACHE SPARK

- A general computing engine for large-scale data processing

  - Java, Scala, Python, R support
  - Build-in machine learning libraries (for classification applications)
  - Easy to deploy



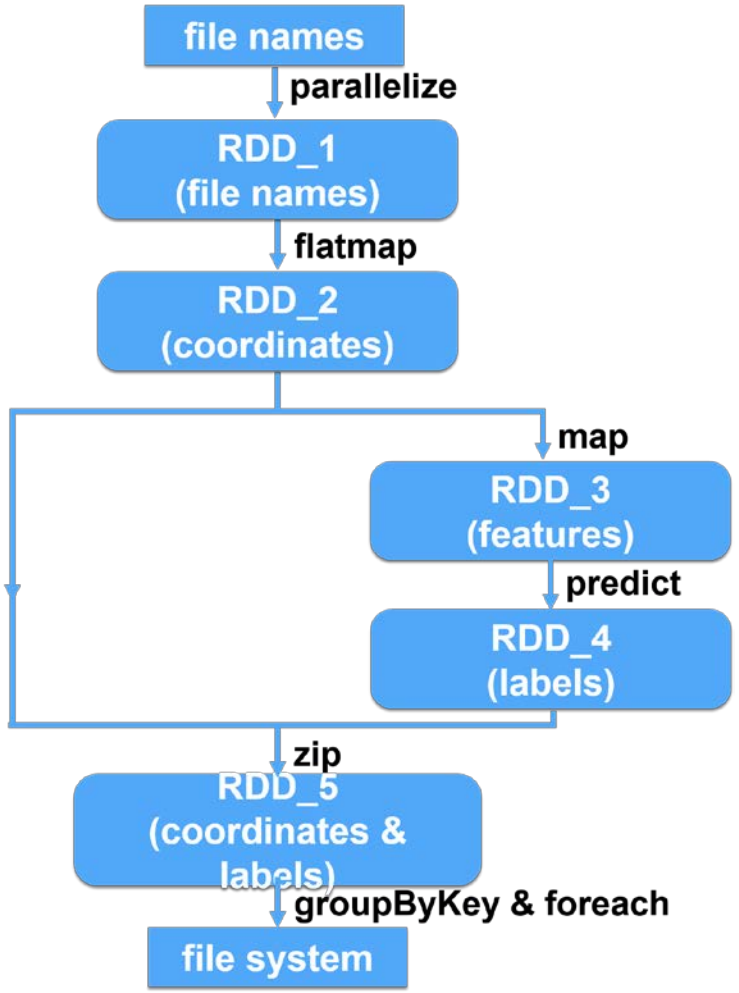
SPARK PROCESSING STEPS

RDD: Resilient distributed Dataset

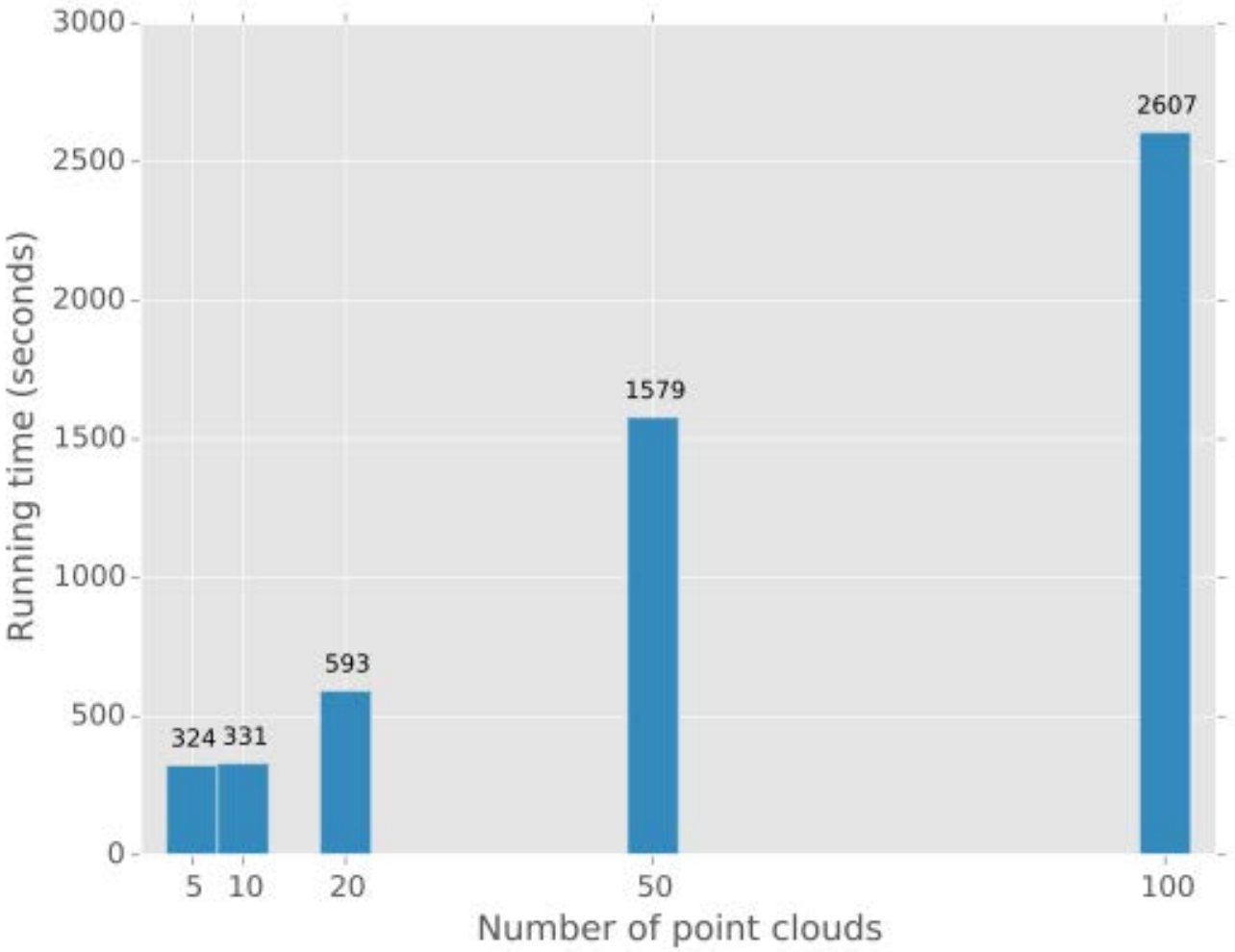
Spark Processing strategy:

Manipulate RDD's by

- Creating new RDD's
- Operate on RDD's
- Transforming existing RDD's

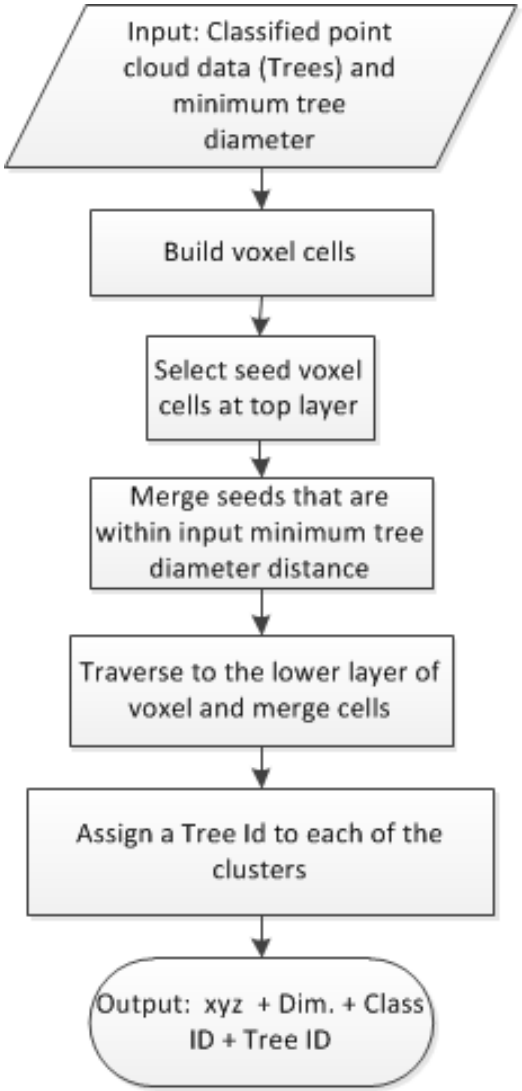


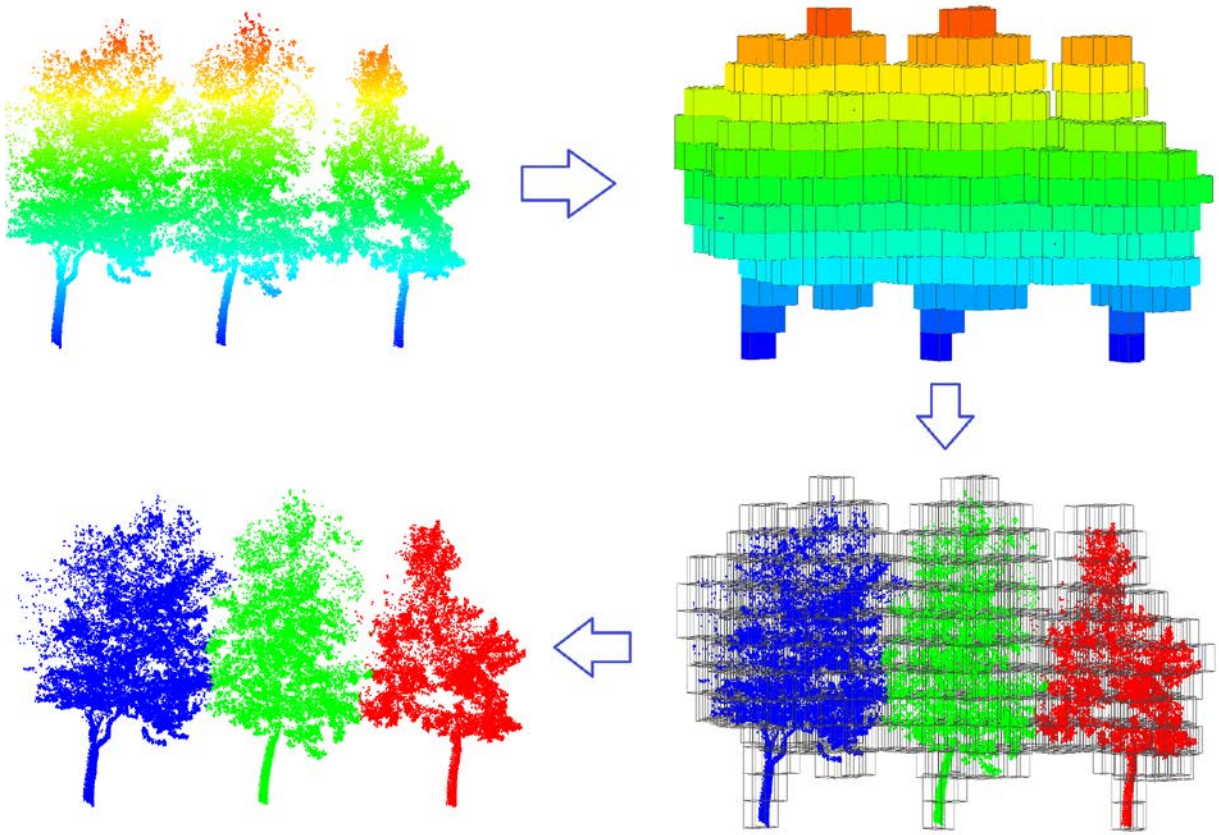
SCALABILITY EXAMPLE



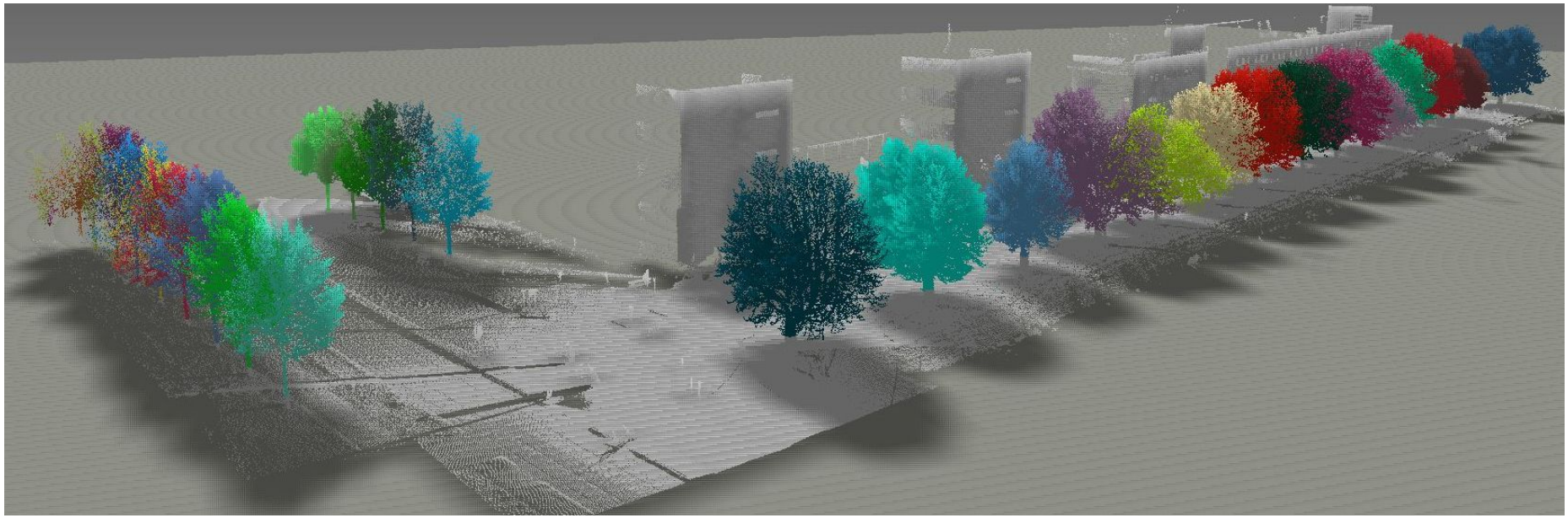
Recall: A point cloud tile has 3 million points

TREE INDIVIDUALIZATION





TREE INDIVIDUALIZATION VISUALIZATION



DETECTED TREES IN GOOGLE EARTH

4000+ tree locations automatically identified by the US2 workflow.

PROCESSING THE US2 DATA

Time taken on **desktop computer**:

~52 hours / 1 core /
~6 minutes per tile

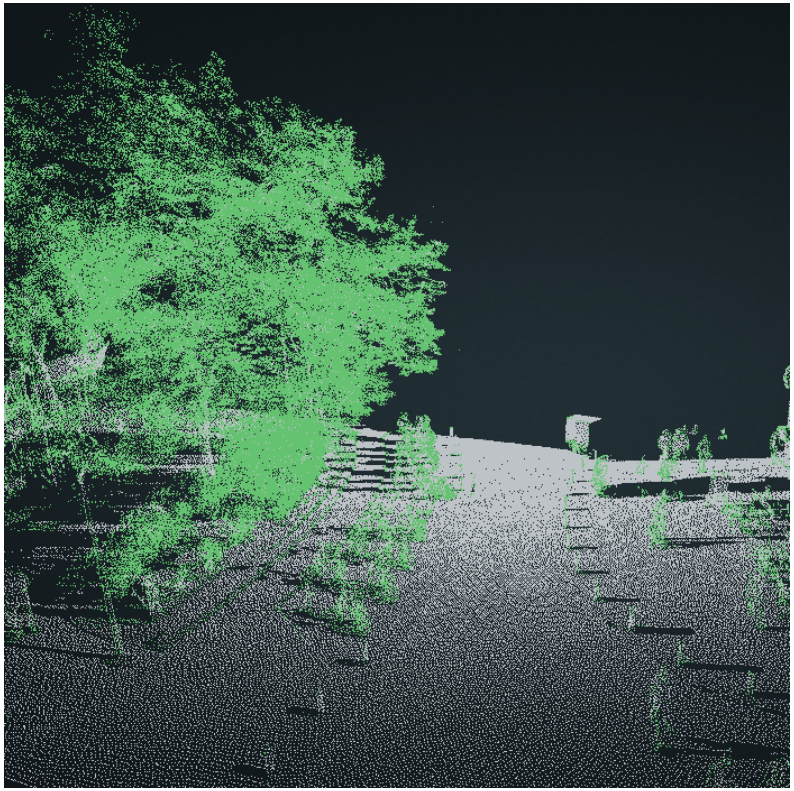
Time taken on **Fraunhofer cloud**:

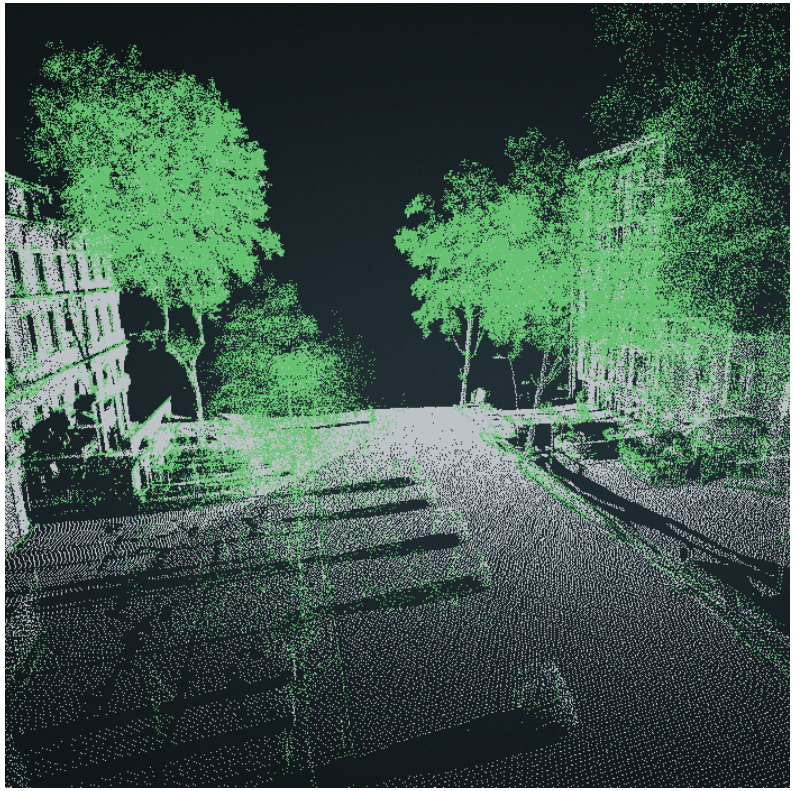
~5 hours / 12 nodes /
~7 minutes per tile + overhead for scheduling

SEVENTH FRAMEWORK PROGRAMME

WP5: VISUALIZATION USING POTREE

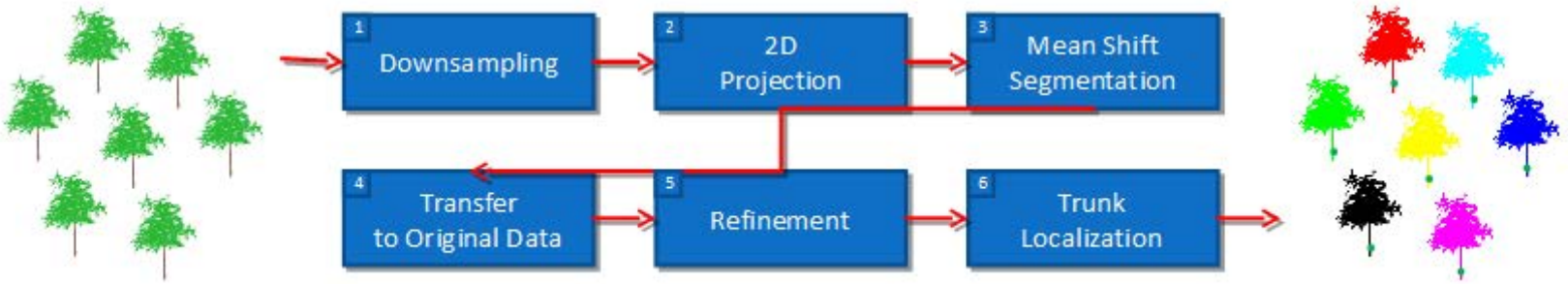
- Organize output data in POTREE octree implementation





- Different data access strategies for **fat** and **thin** client

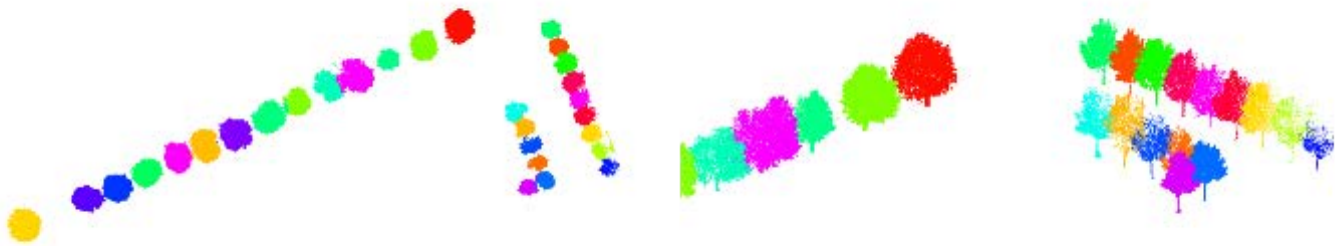
Alternative Tree Individualization: Method



Proposed framework:

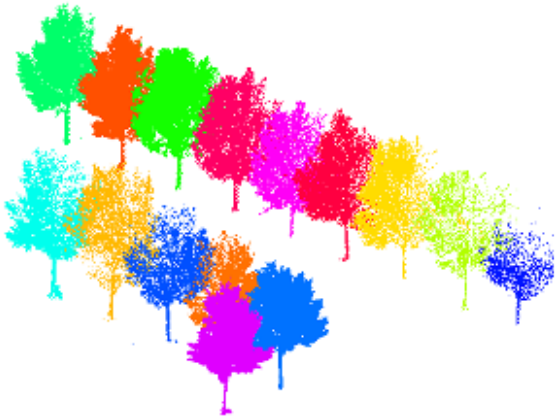
- Efficient approach for individual tree extraction
- Scalability towards the processing of large-scale datasets
- No voxelization, i.e. processing on point-level

GEOBIA2016 SOLUTIONS & SYNERGIES

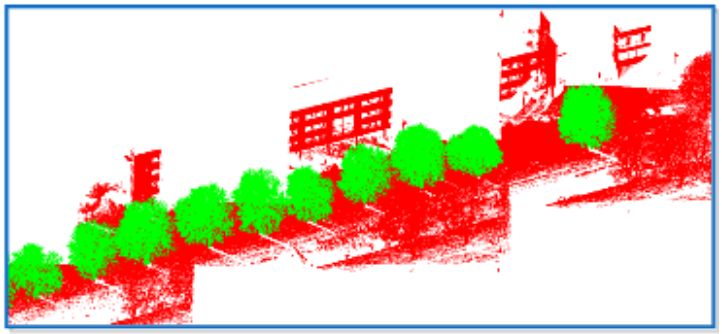


Alternative Tree Individualization: Results

- **Advantages of the proposed framework:**
  - Derived segmentation results are comparable to the state-of-the-art [Gorte et al., 2015]
  - Simple and straightforward approach without voxelization
  - < 1min on a notebook of medium performance (Intel Core i5-2410M, 2.3GHz, 4GB RAM, Matlab)



- **Limitations of the proposed framework:**
  - Dependence on the quality of the previous separation between *tree points* and *other points*
  - The selection of the window size relies on prior knowledge (→ expected size of trees)

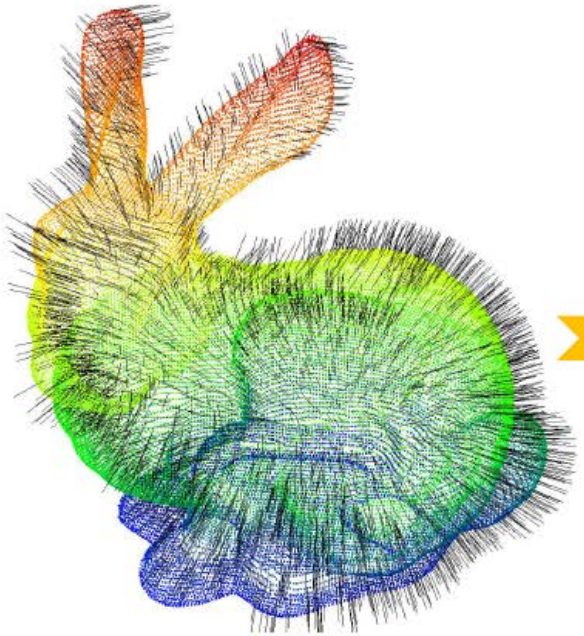


EXTENDED GAUSSIAN IMAGE DESCRIPTOR

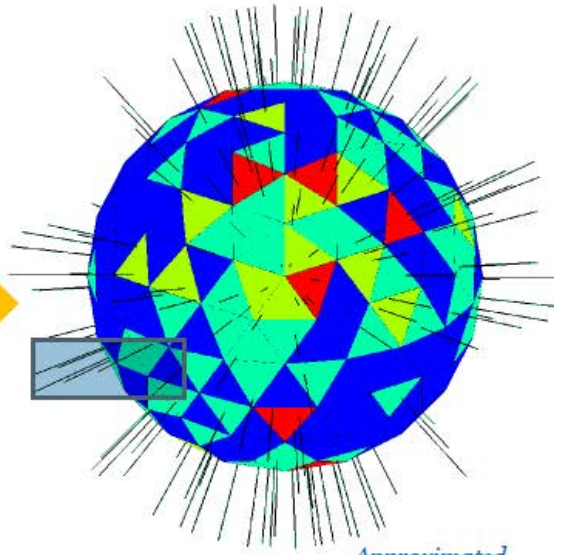
Collect local normals on unit sphere

Bunny

Points



Normals



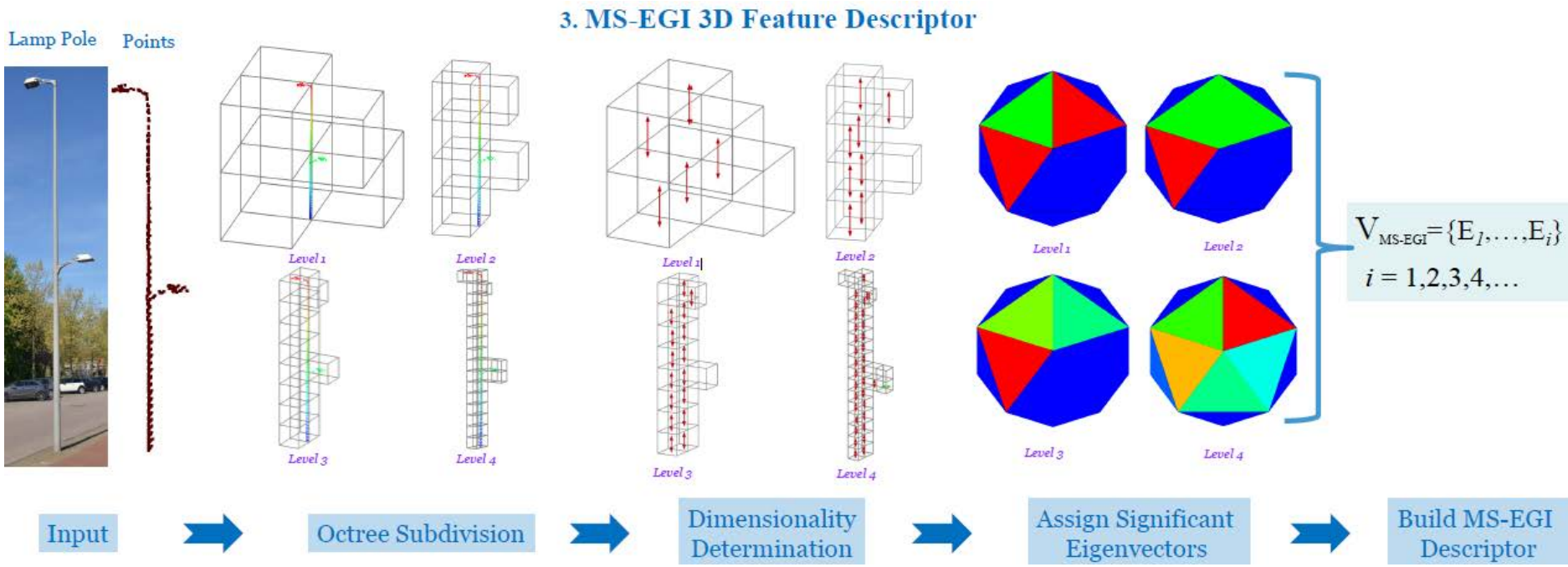
EGI

Approximated by an Icosahedron consisting of ***320*** *faces*

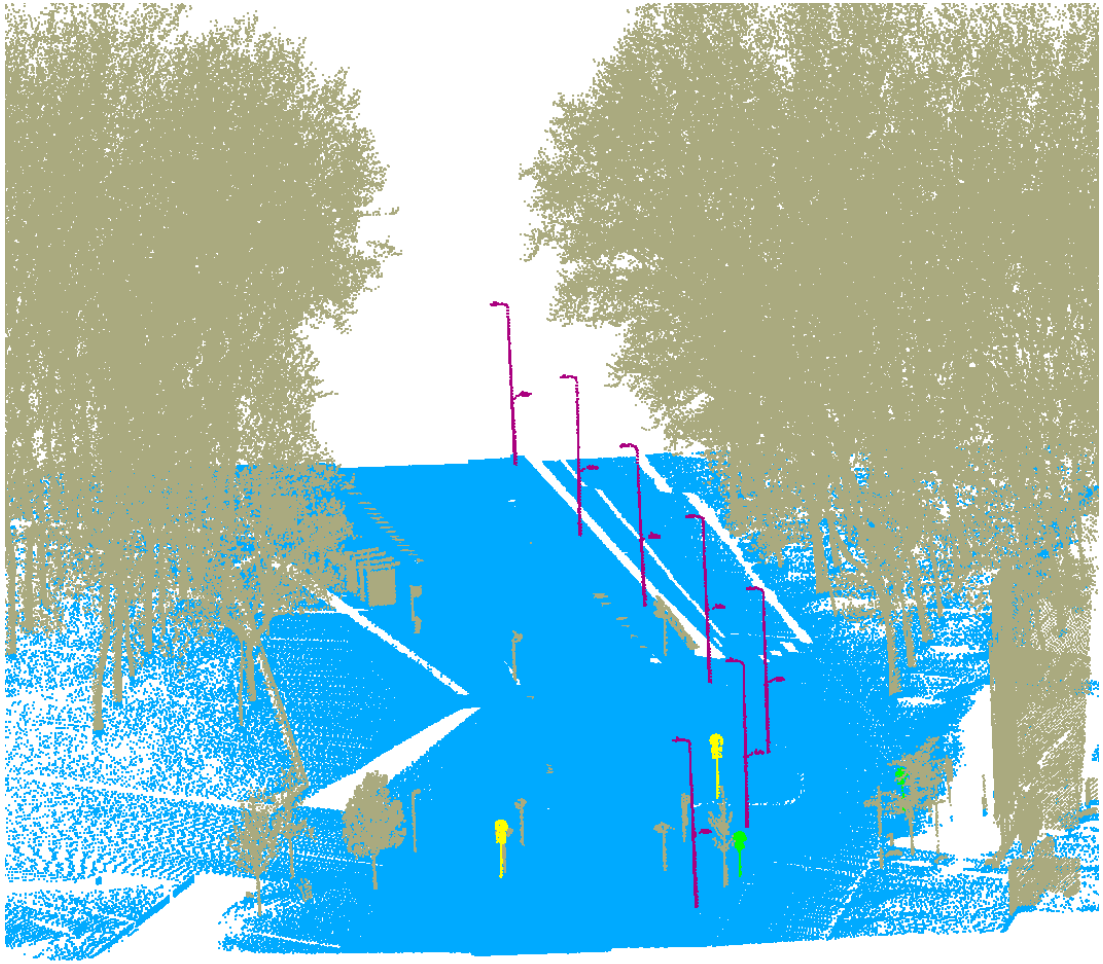
3D MS_EGI MULTISCALE SHAPE DESCRIPTOR

Construct 3D MS-EGI Descriptor:

- Organize object point cloud in voxels at, say, 4 different scales
- Collect significant PCA eigenvectors in icosahedrons for each (of 4) scales



Group objects with similar 3D MS_EGI descriptor



CONCLUSIONS + FURTHER WISHES

5 HOURS PROCESSING vs. 2 HOURS ACQUISITION

- We could solve the **processing bottleneck**!

Cloud computing

- `Simple` distribution
- Hadoop and Spark implementations
- Super computer is not needed

Flexibility should be further increased

- Tree extraction at different locations: **not trivial at all**
- Different wishes from the users: e.g. **street poles + street signs**
- Compatibility between formats: **Open Geospatial Consortium** (OGC) standards
- Contests to compare different strategies

QUESTIONS FOR US2?