Lidar-based Detection of UIC-Hook using Point Cloud Library in ROS

Alvaro Santamaria

Institute for Railway Vehicles and Transportation Systems (IFS)
RWTH Aachen University

October 2022
Aachen

Agenda

- The shunting operation in the field of railways is currently carried out mainly by remote-controlled robots.
- The recognition of items through computer vision permits the automation of a multitude of processes imitating the sense of sight.
- In particular, in the train shunting operation, train wagons are made up of elements that are mainly standardized.

[1]

- **RECOGNITION OF THE UIC-HOOK STATE**

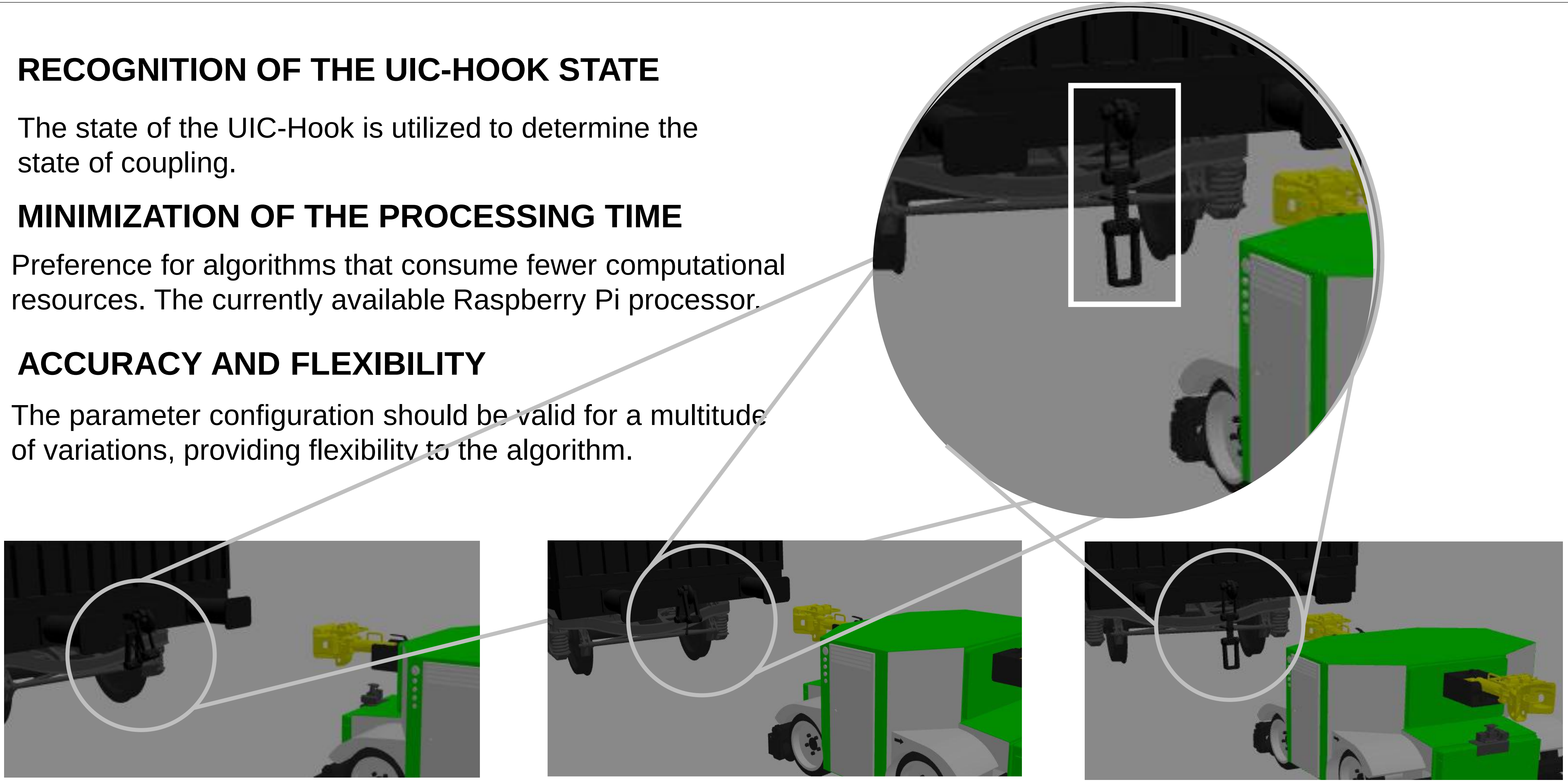
  The state of the UIC-Hook is utilized to determine the state of coupling.

- **MINIMIZATION OF THE PROCESSING TIME**

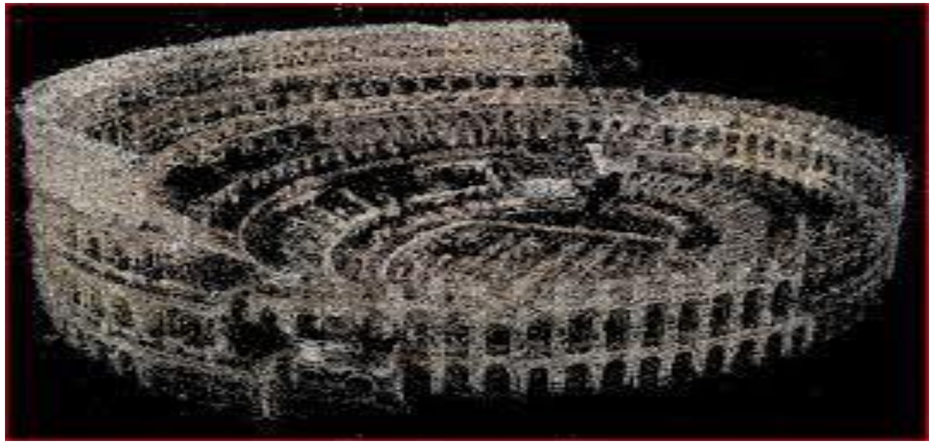
  Preference for algorithms that consume fewer computational resources. The currently available Raspberry Pi processor.

- **ACCURACY AND FLEXIBILITY**

  The parameter configuration should be valid for a multitude of variations, providing flexibility to the algorithm.

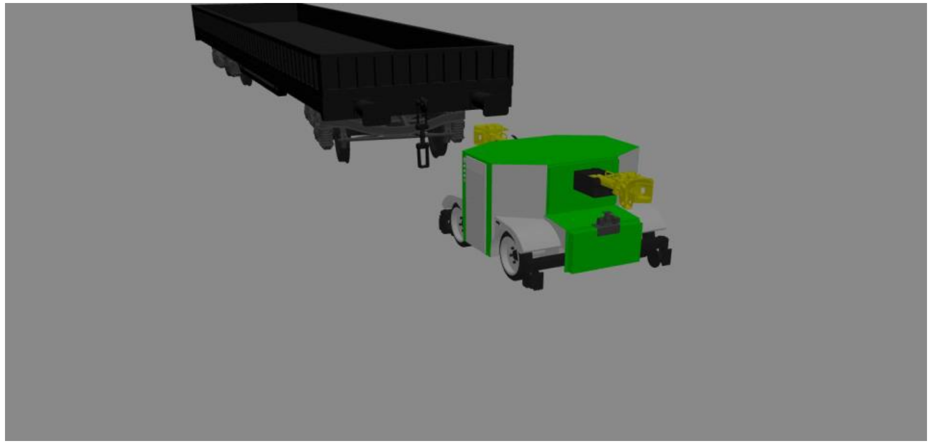


- Point Cloud based computer vision.
  - Objects are described from a multitude of points that denote depth from the sensor, creating a 3D descriptive cloud.



[2]

- Why process simulation?
  - It allows the operation of the process to be checked without the need to interact with the real robot.
  - Quickness in making changes in the program.
  - Added precision due to the simulation of the process using a Digital Twin.

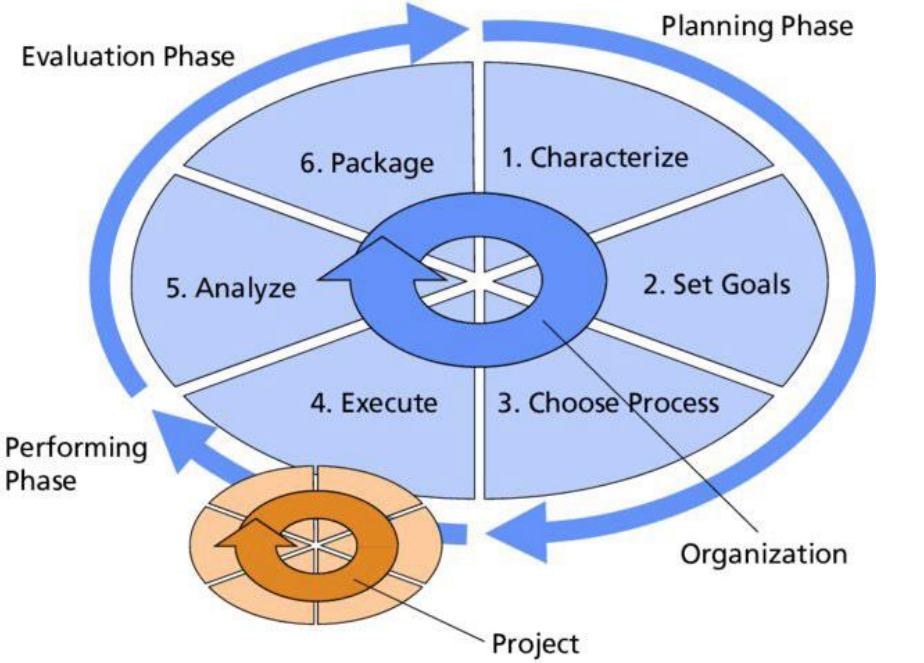


4. Approach

- The algorithms need a specific parametrization for each type of point cloud.

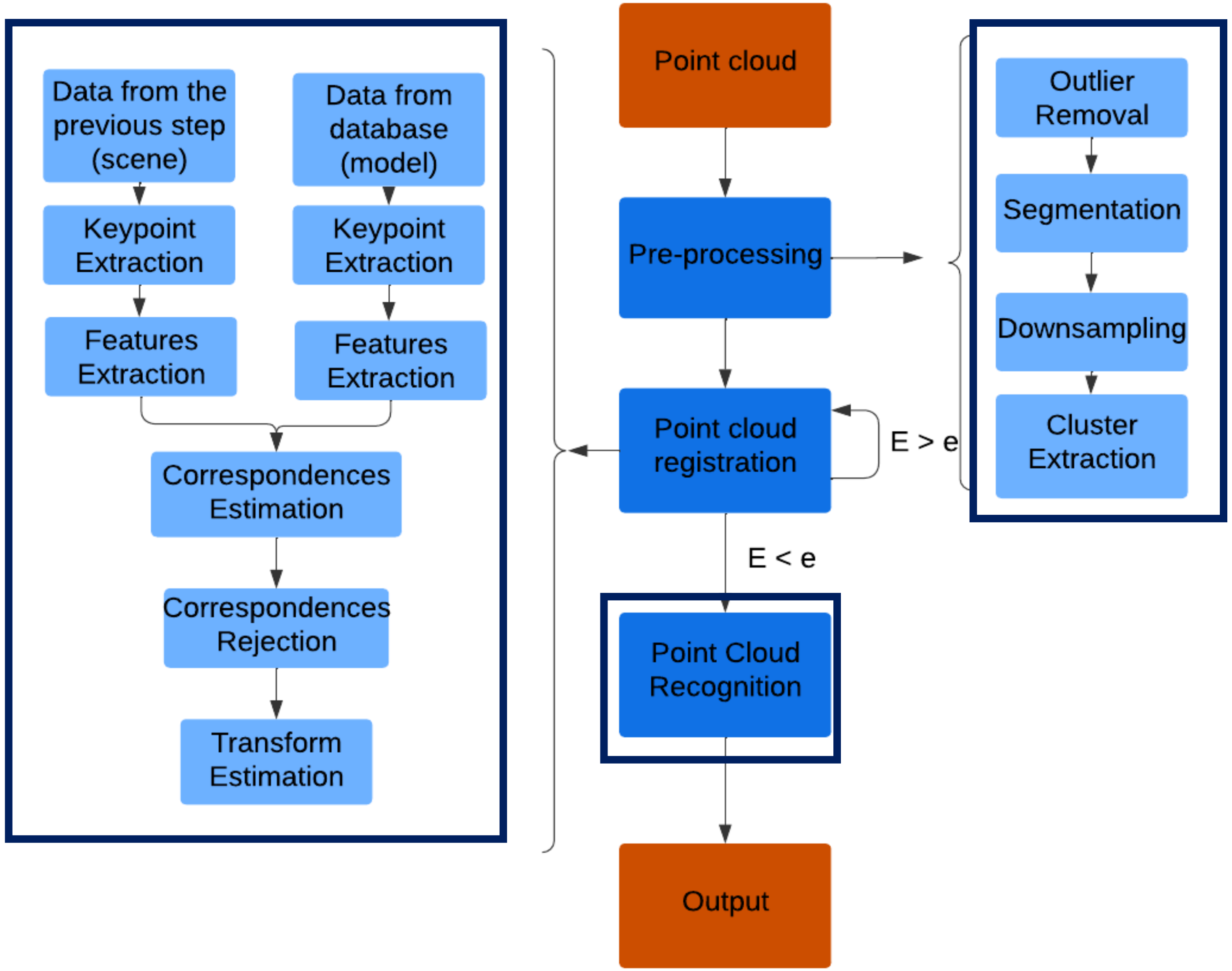
```
//Model
float min_scale_mod (0.01f); //Standard deviation of the smallest scale
int number_oct_mod (10); //Number of groups in the Gaussian pyramid
int number_scales_octave_mod (16); //Number of scales per group
float min_contrast_mod (0.00001f); //Threshold for Keypoint detection
```

- Quality Improvement Paradigm (QIP) method.

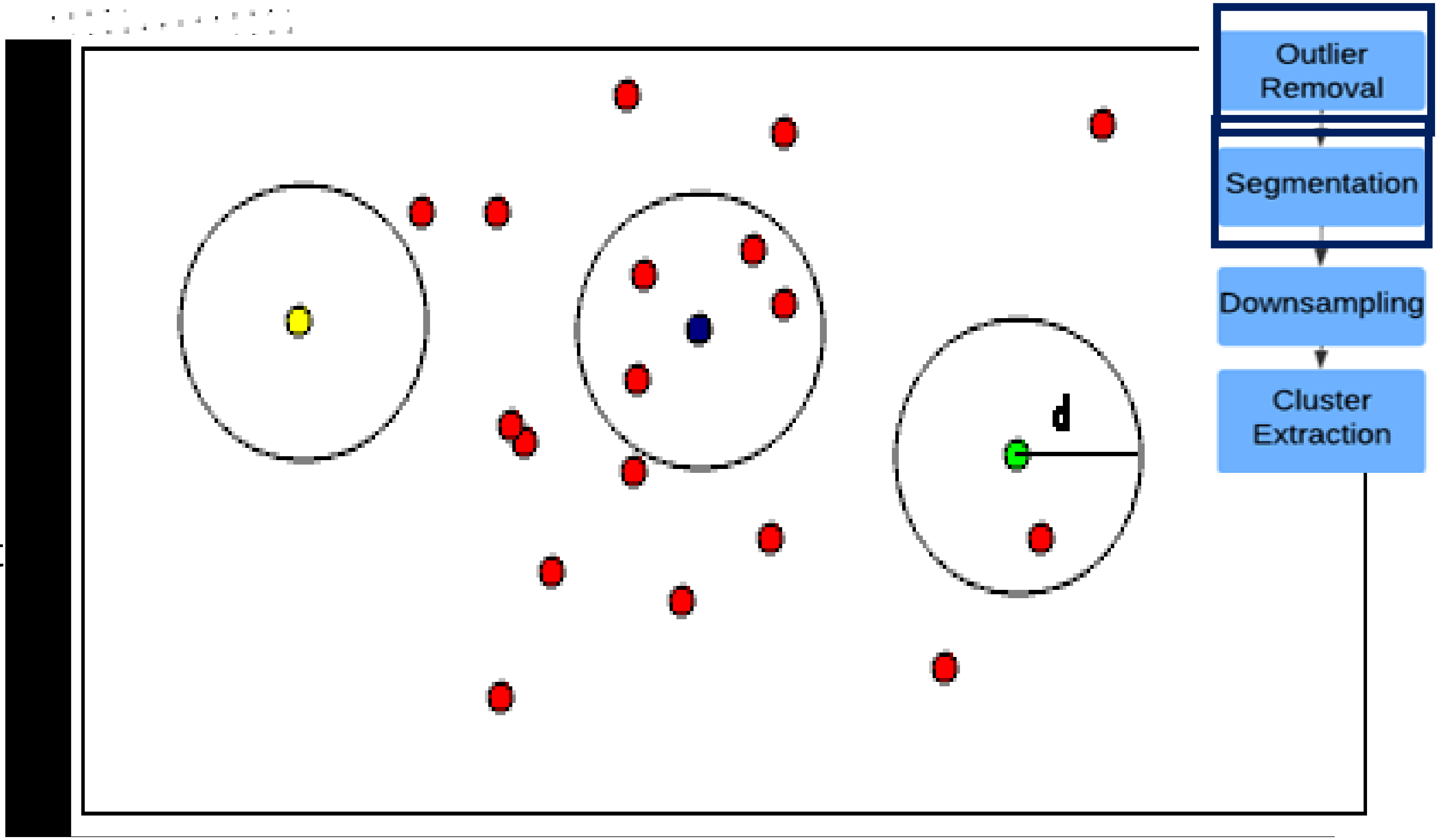


[9]

1. Pre-Processing
   - Its purpose is both to eliminate noisy points and to reduce the total number of points in the point cloud to increase accuracy and reduce processing time in subsequent operations.
2. Registration
   - Creation of the descriptors and correspondences for the definition of the objects that make up the point clouds, allowing the application of the recognition algorithm and the obtaining of the homogeneous transform.
3. Recognition
   - Recognition of objects in the scene from models

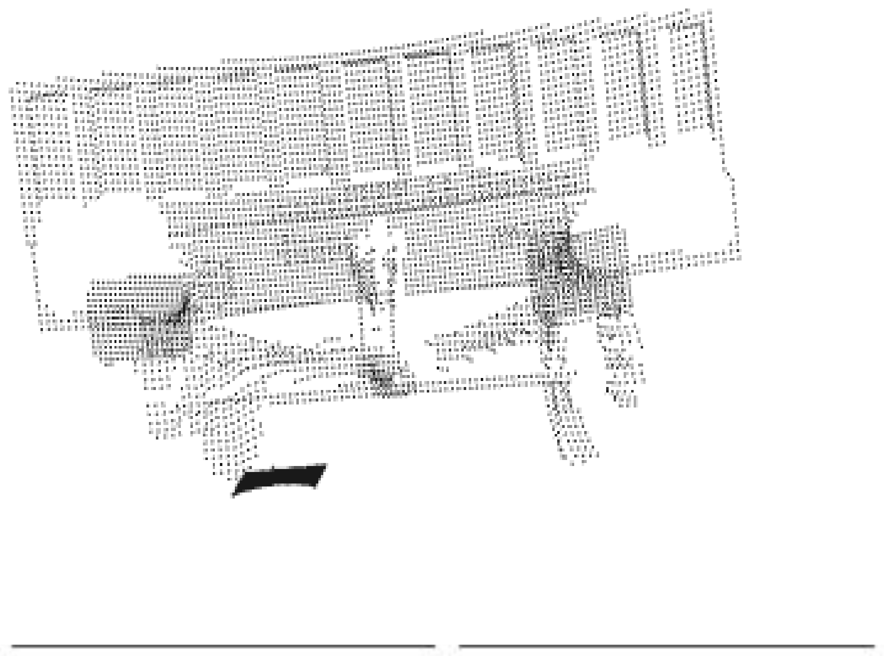


1. **Outlier removal**: Elimination of noisy points.
   - Radius outlier removal
     - MinNeighboursIn
     - Radius_Search
   - Statistical outlier removal
     - SetMeanK
     - StdDesvMul
2. **Segmentation**: Separation of planes in order t eliminate unimportant ones.
   - RANSAC

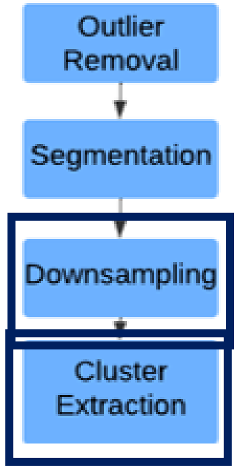


[4] [3]

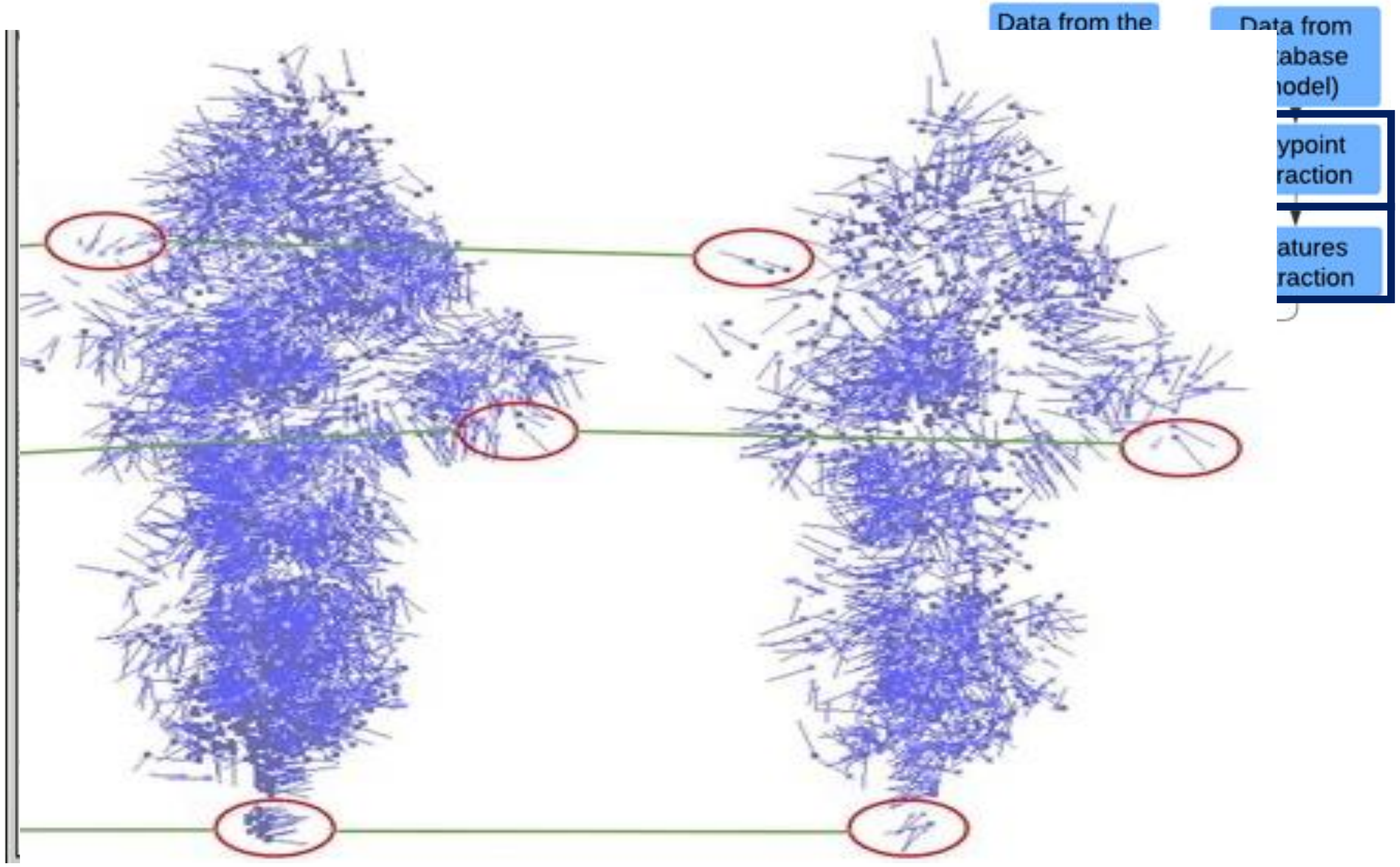
3. **Downsampling**: Point cloud density reduction.
    - Voxel Grid downsampling
        - LeafSize
4. **Clustering**: Extraction of the models from the scenes.
    - Euclidean Clustering
        - ClusterTolerance



[5]

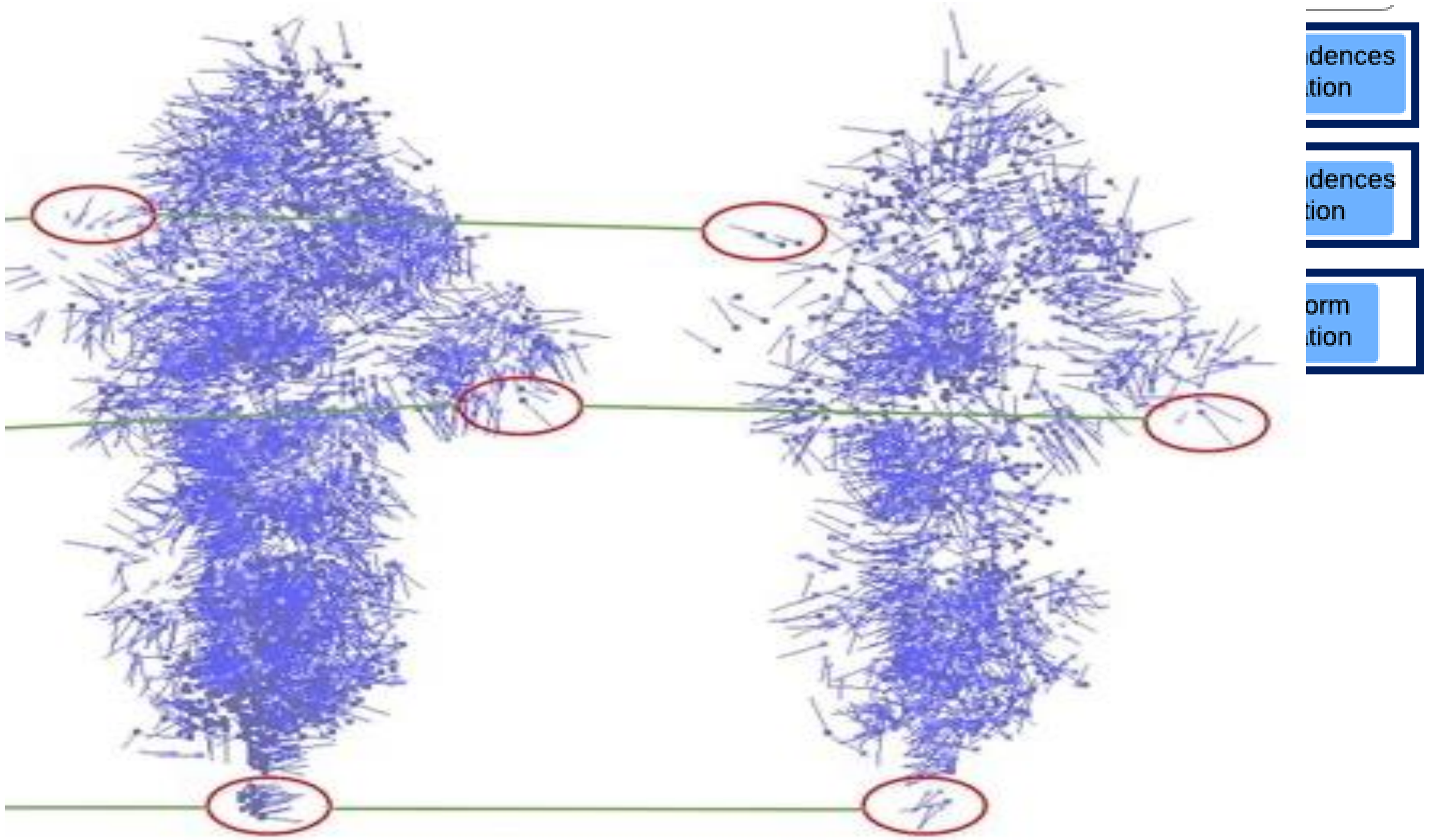


1. **Keypoint Extraction**: Extraction of the points which describe the main characteristics of the point cloud.
   - Uniform Sampling
     - RadiusSearch
   - SIFT
     - Min_Scale
     - Num_oct
     - Num_Scales
     - MinimumContrast
   - ISS
2. **Feature Extraction**: Obtaining the information surrounding the keypoints.
   - SHOT
     - RadiusSearch



[7] [6]

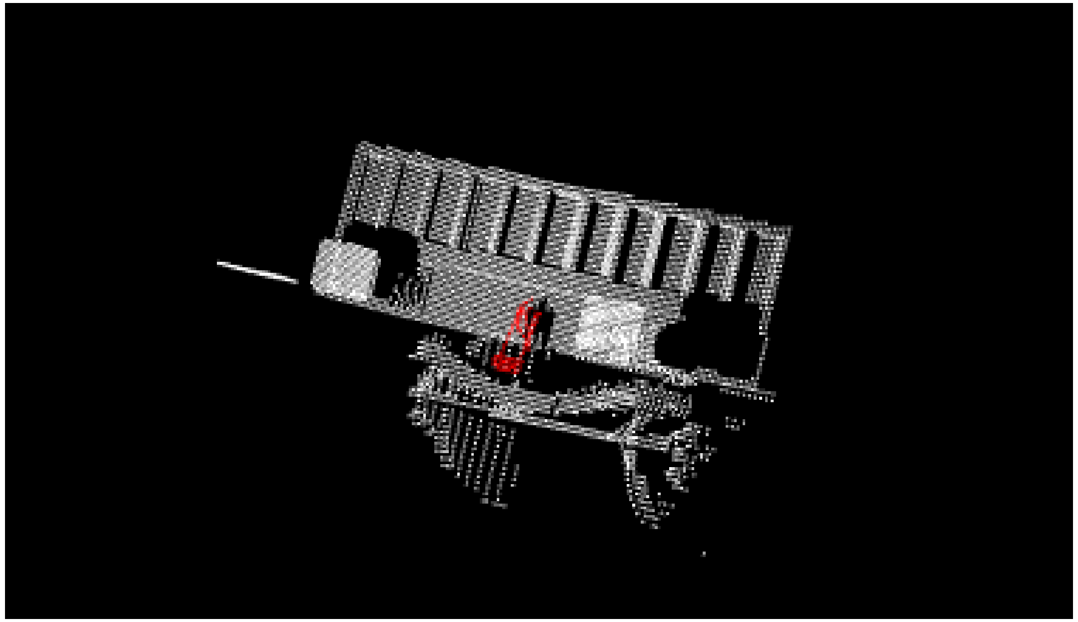
1. **Correspondence Estimation**: Assignment of correspondences between feature descriptors.
   - ICP
     - Kd-Tree - RadiusSearch
2. **Correspondence Rejection**: Elimination of inadequate correspondences.
   - SHOT-Based rejection
3. **Homogeneous Transform Estimation**: Transformation to overlap both clouds.



[7] [8]

- In recognition, two point clouds are considered.
  - Model
  - Scene

- The algorithm used for recognition is Hough3D. It is based on the voting of similarities between the features that have been related by correspondence.



- Division of the functionalities into independent programs.

- Programming language is C++, making use of the libraries present in ROS and PCL.

```
#include <ros/ros.h>
#include <pcl/point_cloud.h>
#include <pcl_conversions/pcl_conversions.h>
#include <sensor_msgs/PointCloud2.h>
#include <pcl/filters/statistical_outlier_removal.h>

class cloudHandler {

public:

 //Creation of the node, Subscriber and Publisher
 ros::NodeHandle nh;
 ros::Subscriber sub;
 ros::Publisher pub;
```

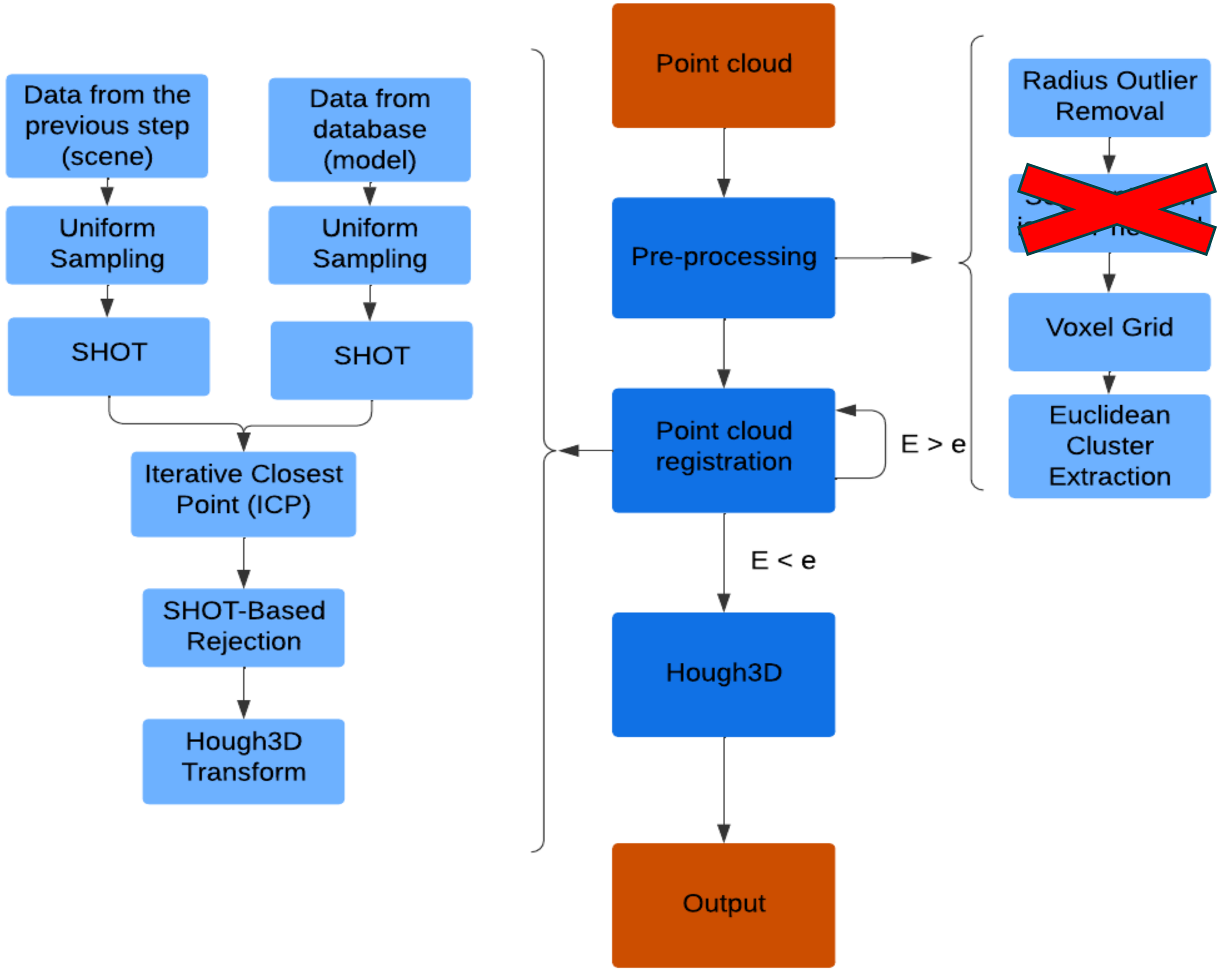
- For the interaction between programs and the simulation environment, .pcd files are used.

```
# .PCD v0.7 - Point Cloud Data file format
VERSION 0.7
FIELDS x y z
SIZE 4 4 4
TYPE F F F
COUNT 1 1 1
WIDTH 13985
HEIGHT 1
VIEWPOINT 0 0 0 1 0 0 0
POINTS 13985
DATA ascii
1.4804899 -3.5304341 -0.65766329
1.5113562 -3.5173316 -0.6576634
```

7. Results

Selected methods for each process

- Pre-Processing
  - Radius vs. Statistical Outlier Removal
  - No need for segmentation
  - Voxel Grid Downsampling
  - Euclidean Cluster Extraction
- Registration and Recognition
  - Uniform Sampling vs. SIFT
  - SHOT
  - Iterative Closest Point (ICP)
  - Hough3D Transform



Parametrization of selected algorithms

- Pre-Processing

Radius Removal	MinNeighborsIn	RadiusSearch
Models 0 & 1	6	0,1
Model 2	9	0.06

Voxel Grid	LeafSize
All models	(0.01f, 0.01f, 0.01f)

Euclidean Cluster Extraction	ClusterTolerance
All models	0.07

- Registration and Recognition

Uniform Sampling	RadiusSearch
All models	0.01
Scene	0.02

SHOT	RadiusSearch
All models	0.15f

Hough3D	descr.RadiusSearch	rf.RadiusSearch
All models	0.15	0.035

- The objectives in terms of object recognition, processing time and accuracy have been achieved.
- Simulation has allowed the entire process to be designed and tested in a considerably reduced time.
- The proposed method has been verified and validated through simulation.
- Validation of the process with experimental data in the real-time.
- Revision of the parametrization.
- Optimization of the code for the real situation.

References

[1] Zasiadko, Mykola (2020): Digital coupling reduces human exposure to risks. In RailTech.com, 7/24/2020. Available online at https://www.railtech.com/digitalisation/2020/07/24/digital-coupling-reduces-human-exposure-to-risks/

[2] Dense Modelling (2009). Available online at http://grail.cs.washington.edu/rome/dense.html, updated on 7/28/2009

[3] Unknown (2022): Removing outliers using a Conditional or RadiusOutlier removal - Point Cloud Library 0.0 documentation. Point Cloud Library. Available online at https://pcl.readthedocs.io/projects/tutorials/en/master/remove_outliers.html#remove-outliers, updated on 6/12/2022.

[4] Unknown (2022): Plane model segmentation - Tutorial. Point Cloud Library. Available online at https://pcl.readthedocs.io/en/latest/planar_segmentation.html#planar-segmentation

[5] Zygmunt, Maria (2013): The testing of PCL: an open-source library for point cloud processing. In GLL 3, pp. 105-115. DOI: 10.15576/GLL/2013.3.105

[6] David G. Lowe (2004): Distinctive Image Features from Scale-Invariant Keypoints. Available online at https://www.cs.ubc.ca/~lowe/papers/ijcv04.pdf

[7] Lv, Yanxing; Zhang, Yida; Dong, Suying; Yang, Long; Zhang, Zhiyi; Li, Zhengrong; Hu, Shaojun (2022): A Convex Hull-Based Feature Descriptor for Learning Tree Species Classification From ALS Point Clouds. In IEEE Geosci. Remote Sensing Lett. 19, pp. 1–5. DOI: 10.1109/LGRS.2021.3055773.

[8] Prof. Dr. Cyrill Stachniss (2022): ICP & Point Cloud Registration - Part 2: Unknown Data Association. Universität Bonn - Photogrammetry & Robotics Lab, 8/22/2022. Available online at https://www.ipb.uni-bonn.de/html/teaching/msr2-2020/sse2-03-icp.pdf

[9] Education Services Australia; Australian Mathematical Sciences Institute (2019): Normal distribution. Available online at https://amsi.org.au/ESA_Senior_Years/SeniorTopic4/4f/4f_2content_3.html, updated on 8/2/2019

Thank you for your attention