PROGRESSIVE MORPHOLOGICAL FILTER

EXPLANATION

INPUT

		VALUE TYPE	DEFAULT VALUES
1	max_window_size_	Int	max_window_size_ (33)
2	slope_	Float	slope_ (0.7f)
3	max_distance_	Float	max_distance_ (10.0f)
4	initial_distance_	Float	initial_distance_ (0.15f)
5	cell_size_	Float	cell_size_ (1.0f)
6	base_	Float	base_ (2.0f)
7	exponential_	Bool	exponential_ (true)
8	input_cloud	PCL::PointCloud	–

OUTPUT

- Indices of ground points

PREPARATION

STEP 1

STEP 1

- **PREPARATION = Compute the series of window sizes and height thresholds**
  - w_k: $window\ sizes$
  - ht_k: height threshold values
  - k: iteration
- **Initial values:**
  - $w_k = 0$
  - $ht_k = 0$
  - $k = 0$
- **while** $(w_k < max\_window\_size\_)$
  - $w_k = cell\_size\_ * (2 * base^k + 1)$
  - $$ht_k = \begin{cases} initial\_distance & if\ w_k \leq 3 (k = 0) \\ s(w_k - w_{k-1})c + initial\_distance & if\ w_k > 3 \\ \max\_distance & if\ ht_k > \max\_distance \end{cases}$$
  - $k++$

MORPHOLOGICAL FILTERING

STEP 2

STEP 2

- MORPHOLOGICAL FILTER = Apply an opening operation to the grid surface

```
---------------------------------
pcl::applyMorphologicalOperator<PointT> (cloud, window_sizes[i], MORPH_OPEN, *cloud_f);
---------------------------------
```

- PARAMETERS

[in]	**cloud_in**	the input point cloud dataset
[in]	**resolution**	the window size to be used for the morphological operation
[in]	**morphological_operator**	the morphological operator to apply (open, close, dilate, erode)
[out]	**cloud_out**	the resultant output point cloud dataset

DETECT GROUND POINTS

STEP 3

STEP 3

- Find indices of the points whose difference between the source and filtered points clouds is less than the current height threshold. (and push them in a vector)

```
---------------------------------
for (size_t p_idx = 0; p_idx < ground.size(); ++p_idx) {
  float diff = cloud->points[p_idx].z - cloud_f->points[p_idx].z;
  if (diff < height_thresholds[i]) pt_indices.push_back(ground[p_idx]);
}
---------------------------------
```

PROGRESSIVE MORPHOLOGICAL FILTER

STEP 4

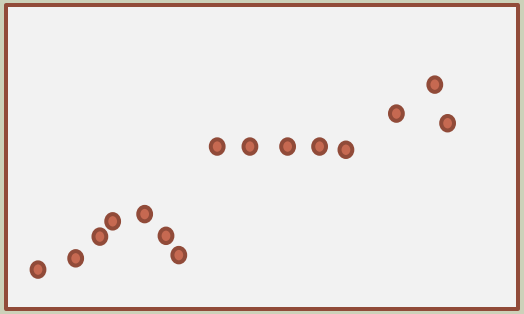
STEP 4

- PROGRESSIVE MORPHOLOGICAL FILTER = Increase the size of the filter window and apply the morphological filter to the points classified as ground from the previous iteration

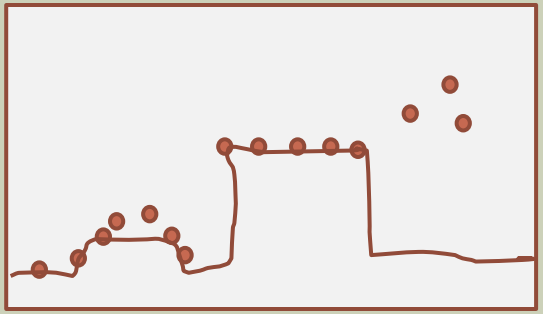
- For every window_size in w_k
  - Create a point cloud containing the ground points using the indices found in the previous step
  - Apply the morphological filtering from step 2
  - Detect the ground points using step 3

SIMPLE EXAMPLE

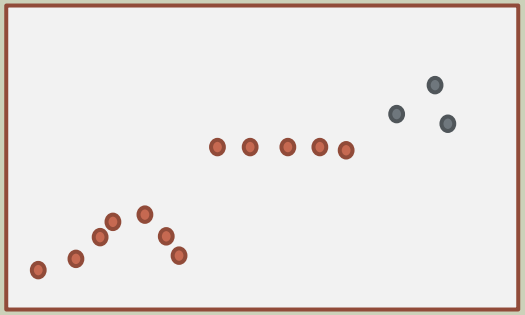
INPUT



1st MORPHOLOGICAL FILTER - OPENING

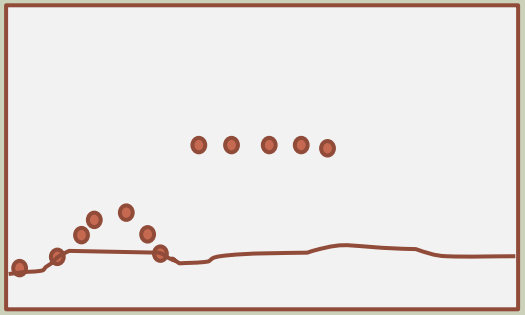


1st GROUND POINTS SELECTION

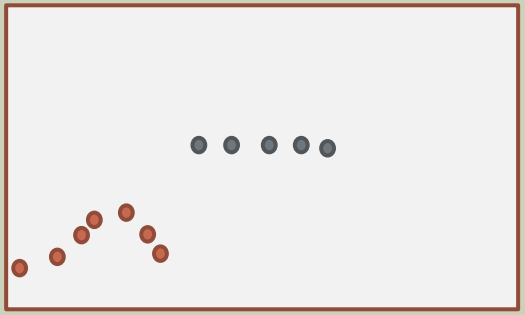


INPUT

1st MORPHOLOGICAL FILTER - OPENING

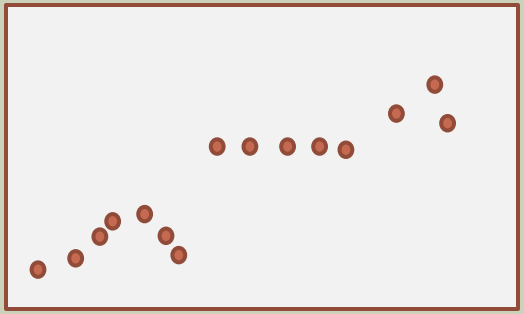


1st GROUND POINTS SELECTION

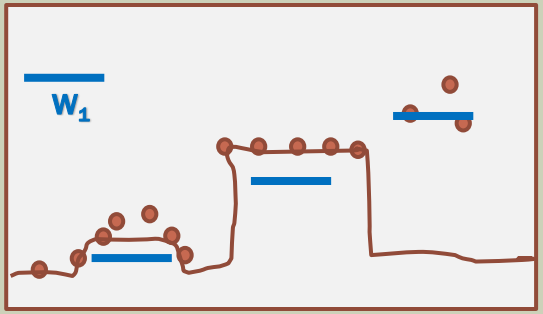


EXAMPLE – FILTER WINDOW SIZE

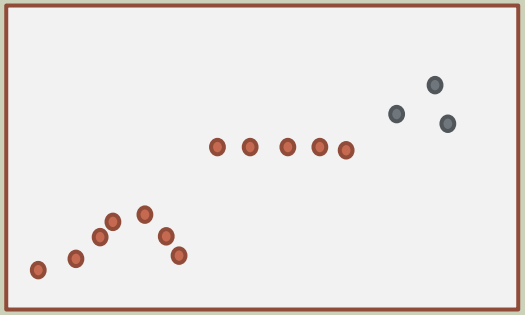
INPUT



1st MORPHOLOGICAL FILTER - OPENING

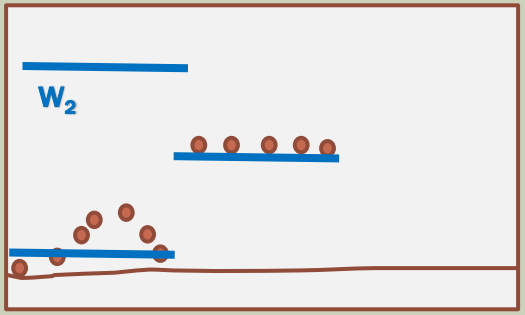


1st GROUND POINTS SELECTION

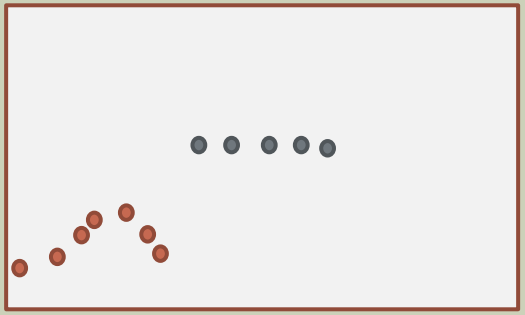


INPUT

1st MORPHOLOGICAL FILTER - OPENING

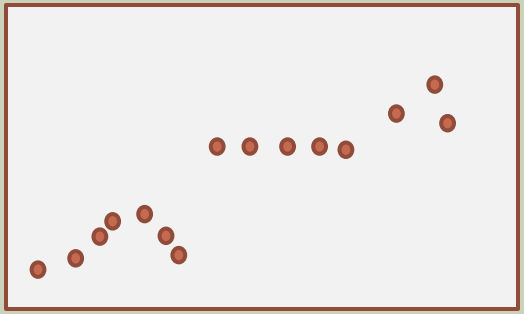


1st GROUND POINTS SELECTION

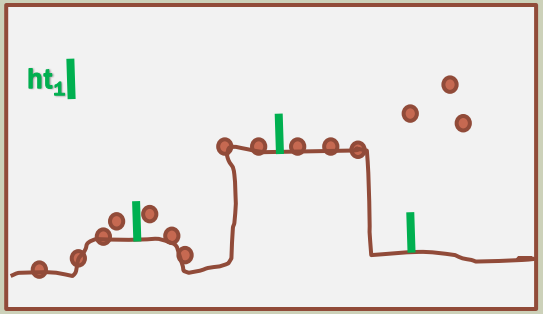


EXAMPLE – HEIGHT THRESHOLD

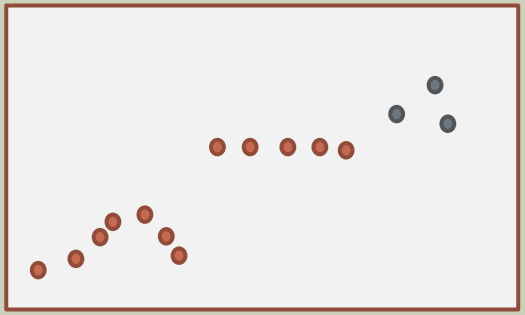
INPUT



1st MORPHOLOGICAL FILTER - OPENING

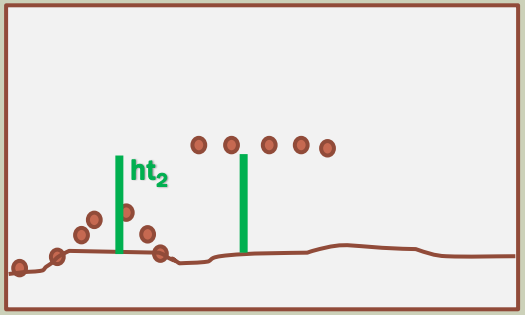


1st GROUND POINTS SELECTION

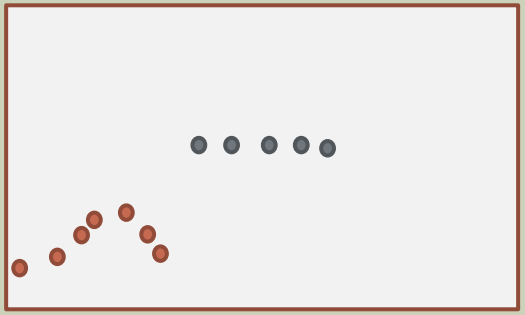


INPUT

1st MORPHOLOGICAL FILTER - OPENING

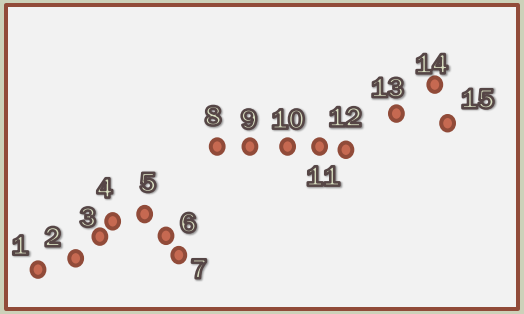


1st GROUND POINTS SELECTION

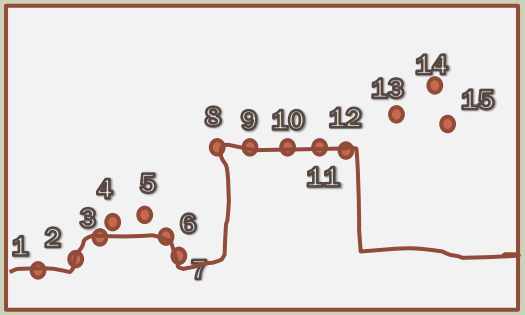


EXAMPLE - INDICES

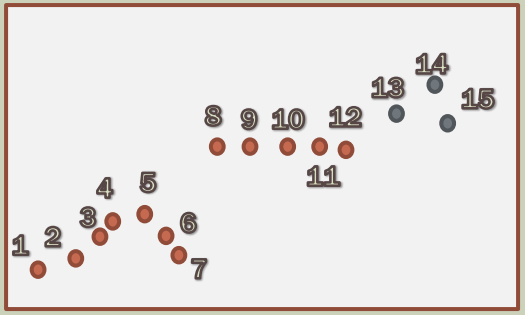
POINT CLOUD TO PROCESS



1st MORPHOLOGICAL FILTER - OPENING

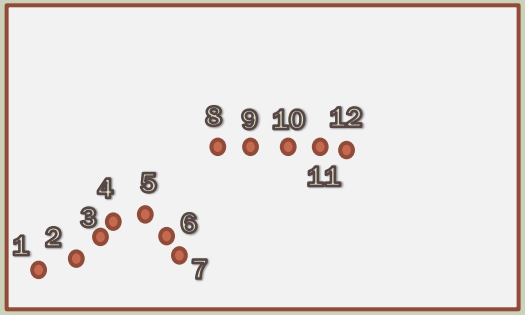


1st GROUND POINTS SELECTION

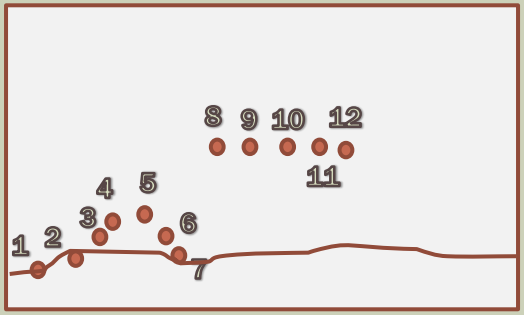


1 2 3 4 5 6 7 8 9 10 11 12

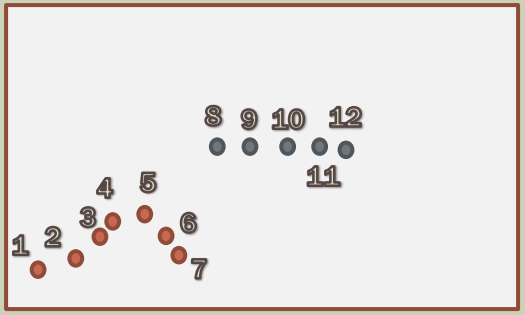
POINT CLOUD TO PROCESS



1st MORPHOLOGICAL FILTER - OPENING



1st GROUND POINTS SELECTION



1 2 3 4 5 6 7

OUTPUT