

This is a software for object part segmentation. To build the software, please go to the build directory. Then type `cmake ..`, and `make`. The executable "comp" will be built. Please adjust the values in the config file for the intended functionality.

mode: There are six modes for the program which are explained based on order below:

sample: for sampling data into training and test set.

- "input-dir": path for reading files.
- "sample-perc": percentage of training data, a value between 0 and 1, by default : 0.7
- output: stores test files in "default_path/evals-pcds/", train files for clustering in "default_path/train-pcds/" and labeled object parts in "default_path/model-parts-pcds"

load: load a pcd file

- "debug": if enabled, you can select the object parts from PCD viewer to label object parts.
- "input-pcd": path of the input pcd file
- In debug mode, press x, then you can select a region in the viewer and you can repeat this procedure multiple time. At the end, press "q", the program asks if you are done segmenting the object, if not, type "n", and select another part.
- output: stores object parts in "default_path/patches/file_id/" each labeled region is ended with "_patch_id.pcd"

estimate_region_models: Extract patch features for the manually labeled parts.

- "input-dir": Directory of the labeled object parts/regions, for example "default_path/patches/"
- output: stores the models in "default_path/prototypes"

estimate_patch_threshold: estimates minimum threshold for clustering patches

- The threshold is estimated from the trained object parts/regions.
- "prototypes": path of trained part/region models
- output: writes the threshold in "default_path/cls-patch-thresh.txt"

cluster: clusters patches from training data and makes patch codebook

- "input-dir": Directory of the input training data
- output: stored clusters and their centers and distances (for codebook) in "default_path/clusters"

collect_statistics: collect patch type co-occurrence matrices for part inference in novel objects

- "patch-cluster": Directory of stored clusters, only their centers and distances will be read
- "prototypes": path of trained part/region models
- output: stores "default_path/patch-cluster/patch-occ" for patch type co-occurrence that can form a part and "default_path/patch-n-occ/n-occ" for patch types co-occurrence which cannot form an object part

part_seg: segment novel objects into parts

- "patch-cluster": Directory for patch clusters
- "patch-cluster-mat": Path for patch type co-occurrence matrix for forming object parts
- "patch-n-mat": Path for patch type co-occurrence matrix which cannot form object parts

- "eval-path": Directory which contains test pcd files
- output: The segmented parts will be stored in "default_path/eval-parts" as "file_name_part_id". In addition, supervoxels, patches, merged patches/regions in each iteration of the algorithm are stored in "default_path/vis-parts/file_name/"
- The algorithm computes overlapping between segmented object parts and the groundtruth parts which are in "default_path/eval-ground-truth/" and stores the results in "overlap-file.txt" in the execution directory. To compute this measure for evaluation, you should save the labeled ground truth object parts based on "load" mode in "default_path/eval-ground-ruth"

There are some additional parameters which are explained below:

voxel-resolution: voxel resolution for supervoxel algorithm
 seed-resolution: seed resolution for supervoxel algorithm
 default-path: The path based on which files will be stored