I. DATABASE STRUCTURE AND FILE FORMATS

The SOC2012 database is arranged in three main directories:

- ./object
- ./scenes
- ./overview

where:

- ./object directory contains data for single objects:
  - from 14 object categories with 10 object instances per category what gives in total 140 object instances.
    - categoryName = \{ball, bottle, box, can, car, citrus, cup, mammal, mobile, sd, tissue, toipap, tube, vegroot\}
    - categoryLabel = \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14\}
  - that were collected from 16 views uniformly spaced around the object (every 22.5 degree).
  - NOTE: Within each category objects are approximately aligned with respect to the pose. However, the pose labels are not exact, since objects within one category differ in shape.
  - angelsLabels = \{000, 022, 045, 067, 090, 112, 135, 157, 180, 202, 225, 247, 270, 292, 315, 337\}.

- ./data-raw directory contains raw (not preprocessed) data as they were collected by the ARMAR-III robotic head.
  - For each object instance, for example ./ball01 we provide for each object view an image with a segmentation mask, a segmented 3D point clouds and camera parameters:
    - leftFinal_*.ppm - RGB image captured by the left foveal camera
    - mask_*.ppm - corresponding segmentation mask for the leftFinal_*.ppm
    - filterSegment_*.crd - segmented 3D color pointcloud in the robot arm coordinate system. The outlier points are removed. Additionally, each 3D point is annotated with corresponding image coordinates (pixels values).
    - The .crd file format is described in Section ??.
    - Software to load a .crd file to the PCL and Matlab is provided in ./overview/software.
    - camOrig_*.txt - coordinates of a camera orgin in the robot arm coordinate system
    - fov2Arm_*.txt - transformation matrix from a camera to the arm coordinate system
  - NOTE: ARMAR-III robotic head is equiped with an active stereo camera system. Thus, positions of the camera for each collected view differ.

- ./data-pointcloud we provide for each object view a segmented and filtred 3D pointcloud. The point density is equilized using the grid filtering implemented in the PCL library. We used these 3D data in the experimental evaluation in [1], [2], [3], [4].

REFERENCES