CSCI598A: Robot Intelligence

Feb. 12, 2015
Midterm Project
## Midterm Project

<table>
<thead>
<tr>
<th>Task</th>
<th>Points</th>
</tr>
</thead>
<tbody>
<tr>
<td>Moving a target item from a multi-item area into the order bin</td>
<td>+20 points</td>
</tr>
<tr>
<td>Moving a target item from a double-item area into the order bin</td>
<td>+15 points</td>
</tr>
<tr>
<td>Moving a target item from a single-item area into the order bin</td>
<td>+10 points</td>
</tr>
<tr>
<td>Moving a non-target item from the table top</td>
<td>-12 points</td>
</tr>
<tr>
<td>Damaging any item or packaging</td>
<td>-5 points</td>
</tr>
<tr>
<td>Dropping a target item from a height above 0.3 meters</td>
<td>-3 points</td>
</tr>
</tbody>
</table>
Midterm Project

Safety First

[Image of a warning sign with the text "Safety First" and an illustration of a robot and a person]

[Image of a robot and a person in an action pose]

Art. Nr. 00-104-232
Point Cloud Library

- Features
- Filters
- Keypoints
- Registration
- Segmentation
- Sample Consensus
- Surface
- Range Image
- I/O
- People
- Simulation
- Out-of-core
- Visualization
- Segmentation
- 2D
- ML
- Recognition
- Kdtree
- Octree
Point Cloud Library

- PCL can deal with both **organized** (e.g. range maps) and **unorganized** point clouds
  - if the underlying 2d structure is available, efficient schemes can be used (e.g. **integral images** instead of kd-tree for nearest neighbor search)
- Both are handled by the same data structure (**pcl::PointCloud**, templated thus highly customizable)
  - Points can be XYZ, XYZ+normals, XYZI, XYZRGB, ...
  - Support for **RGB-D data**
- **Voxelized** representations are implemented by **pcl::PointCloud + voxelization functions** (e.g. voxel sampling)
  - no specific types for voxelized maps
- Currently rather limited support for **3D meshes**
3D Object Recognition Using PCL

- **Definition (typical setting):**
  - a set of 3D models (often, in the form of views)
  - one scene (at a time) including one or more models, possibly (partially) occluded, + clutter.

- Models can be present in **multiple instances** in the same scene

- **Goal(s):**
  - determine which model is present in the current scene
  - (often) estimate the 6DoF pose of the model wrt. the scene

- Applications: industrial robotics, quality control, service robotics, autonomous navigation, ..
Representation for 3D Object Recognition

- Usually Object Recognition in clutter is done on 2.5 data (model views against scene views)
- Can be done also 3D vs 3D, although scenes are usually 2.5D (and 3D vs. 2.5D does not work good)
- When models are 3D, we can render 2.5D views simulating input from a depth sensor:

```cpp
pcl::apps::RenderViewsTesselatedSphere render_views;
render_views.setResolution (resolution_);
render_views.setTessellationLevel (1); //80 views
render_views.addModelFromPolyData (model); //vtk model
render_views.generateViews ();
std::vector<pcl::PointCloud<pcl::PointXYZ>::Ptr > views;
std::vector < Eigen::Matrix4f > poses;
render_views.getViews (views);
render_views.getPoses (poses);
```
Pipeline: 3D Object Recognition

LOCAL PIPELINE
- Keypoint Extraction
- Description
- Matching
- Correspondence Grouping
- Absolute Orientation

GLOBAL PIPELINE
- Segmentation
- Description
- Matching
- Alignment

ICP refinement

Hypothesis Verification
Descriptor Matching

- Typical paradigm for finding similarities between two point clouds
  1. Extract compact and descriptive representations (3D descriptors) on each cloud (possibly over a subset of salient points)
  2. Match these representations to yield (point-to-point) correspondences

- Applications: 3D Object recognition, cloud registration, 3D SLAM, object retrieval, ..
Keypoints or Points of Interest

- **3D keypoints** are
  - Distinctive, i.e. suitable for effective description and matching *(globally definable)*
  - Repeatable with respect to point-of-view variations, noise, etc... *(locally definable)*
- The **pcl::keypoint** module includes:
  - A set of detectors specifically proposed for 3D point clouds and range maps
    - Intrinsic Shape Signatures (ISS) [Zhong 09]
    - NARF [Steder 11]
    - (Uniform Sampling, i.e. voxelization)
  - Several detectors «derived» from 2D interest point detectors
    - Harris (2D, 3D, 6D) [Harris 88]
    - SIFT [Lowe 04]
    - SUSAN [Smith 95]
    - AGAST [Mair 10]
    - ...
Global vs Local Descriptors

- **Pcl::Features: compact** representations aimed at detecting similarities between surfaces (*surface matching*)
- Based on the support size
  - **Pointwise descriptors**
    - Simple, efficient, but not robust to noise, often not descriptive enough (e.g. normals, curvatures, ..)
  - **Local/Regional descriptors**
    - Well suited to handle clutter and occlusions
    - Can be vector quantized in codebooks
    - Segmentation, registration, recognition in clutter, 3D SLAM
  - **Global descriptors**
    - Complete information concerning the surface is needed (no occlusions and clutter, unless pre-processing)
    - Higher invariance, well suited for **retrieval and categorization**
    - More descriptive on objects with poor geometric structure (household objects..)