Object segmentation and registration of point clouds from different perspectives aided by a service robot

Nikolas Slottke
7slottke@informatik.uni-hamburg.de

Universität Hamburg
Fakultät für Mathematik, Informatik und Naturwissenschaften
Department Informatik

Technische Aspekte Multimodaler Systeme

15. April 2014
Table of Contents

Introduction
  Motivation
  Goal
State of the Art
Approach
  Tabletop Scan
  Gripper Scan
  Problems
Software
Conclusion
  Results
  Comparison
  Summary
Introduction - Motivation

Object segmentation and registration of point clouds from different perspectives aided by a service robot

Motivation

- Cost-efficient devices like Microsoft Kinect made depth sensors widely available
- It is possible to create a 3D model from point clouds of different perspectives of an object
- This task can be executed mostly autonomous by a service robot in indoor environments
- Those 3D models can be stored in a database and used for object recognition
The RACE Project started in 2011
- RACE (Robustness by Autonomous Competence Enhancement)
- Robots capable of storing experiences
- Methods for learning and generalising from experiences
- Robots demonstrating superior robustness and effectiveness in new situations

http://project-race.eu/
Goal

- Create an autonomous process for the PR2 service robot
- 3D models of unknown objects with point clouds from a depth sensor
- Manipulation of objects to create different perspectives
- Segmentation of object and background
State of the Art - PR2

- PR2 is made by Willow Garage
- PR (Personal Robot)
- Two arms with 7 DOF each, can lift objects up to 1.8 Kg
- It has a lot of sensors: a 5 MP camera, two stereo camera systems, two laser range finders, gyroscopes, etc.
- The base of the robot holds two computers with two Intel server CPUs and 24GB RAM each
- The battery consists of 16 laptop batteries
State of the Art - PR2
State of the Art - Kinect

- Microsofts Kinect is a sensor for motion control
- It has a RGB and a depth camera with VGA resolution and an IR projector
- It can provide 11-bit depth information up to VGA resolution at 30Hz
- Depth sensing works between 1.2m and 3.5m distance from the sensor
Approach - Tabletop Scan

- First approach is called the Tabletop Scan
- Goal
  - recognize an object on a table
  - create a point cloud and segment the object
  - manipulate the object to create new perspectives
  - merge the segmented point clouds into a complete model
Scenario and requirements

- The PR2 is in a room with a table and the head with the depth sensor points at the table
- On the table is an object that needs to be graspable by the PR2 and has to be within reach of the arms
- The object also needs to be graspable with a top grasp
Approach - Gripper Scan

- The second approach is the Gripper Scan
- Goal
  - recognize an object on a table
  - pick up the object
  - move the object in front of the depth sensor
  - create point clouds and rotate the gripper to create new perspectives
  - segment the points of the object from the arm and the rest of the scene
  - merge all point clouds into a single model
Scenario and requirements

Basically the same as the Tabletop Scan, except that the object does not necessarily need to be picked up with a top grasp.

- The PR2 is in a room with a table and the head with the depth sensor points at the table
- On the table is an object that needs to be graspable by the PR2 and has to be within reach of the arms
Problems

Problem: Registration with the ICP algorithm. Single segmented pointcloud from one perspective (coffee cup, ca. 3000 points):
Object segmentation and registration of point clouds from different perspectives aided by a service robot

Problems

Alignment via ICP of 24 perspectives created with the Tabletop Scan:
Alignment with the knowledge about the rotations of 24 perspectives created with the Tabletop Scan:
Problems

Problem: Calibration. Example for the registration of the coffee cup with uncalibrated coordinate systems (depth sensor):
Software - ROS

- ROS (Robot Operating System)
- ROS is a framework that is used to control robots
- Main tasks:
  - hardware abstraction
  - provide device drivers
  - communication between programs
  - package management

There are several distributions. This work relies completely on ROS Fuerte.
Software - Architecture

The software of this work is organized as a package in ROS.

- Big, complicated tasks are split into several simple tasks which are implemented as modules
- Mostly client-server architecture
- 12 modules (nodes)
- One action-client on top of the hierarchy
Software - Architecture

Hierarchy of the 12 nodes:

- **CompleteScanClient**
  - **CompleteScanAction**
    - **GripperScanAction**
      - **ScanObjectInGripperAction**
        - **SegmentObjectInGripperServer**
      - **RotateGripperAction**
      - **GetArmPoseServer**
    - **TabletopScanAction**
      - **TabletopGraspAction**
      - **TabletopPlaceAction**
      - **MoveArmToPoseAction**
      - **RetreatAndMoveAction**
Software - CompleteScanClient

The CompleteScanClient:

- Executable for the command line (can be called via rosrun)
- Interface between the user and the software
- Has seven parameters
  - path to a folder where the results will be stored
  - execute a Tabletop Scan
  - execute a Gripper Scan with a side grasp and/or a top grasp and/or an auto grasp
  - how many perspectives should be used to create a model
  - storing of single point clouds for each perspective
Software - TabletopScanAction

The TabletopScanAction:

- Implementation of the Tabletop Scan approach
- Uses several other modules to accomplish this task
- Uses TabletopSegmentation Service from ROS Fuerte
- Outputs the point clouds of the object
TabletopScanAction - Implementation (1)

The implementation consists of many steps:

1. Recognize the object on the table and grasp it with a top grasp (TabletopGraspAction)
2. Manipulate the object by rotating the gripper (RotateGripperAction) and save the successful grasp to ease the next grasp
3. Save the position of the gripper (center of rotation) for each perspective in a vector/array
4. Release the object and move the arm to the side (RetreatAndMoveAction)
5. Create a point cloud and segment the points of the object (TabletopSegmentation)
6. Repeat steps 1-5 until the object has been rotated by $360^\circ$.

7. Merge all segmented point clouds to a complete model by aligning them with the knowledge of the rotation.

The angle of rotation is determined by the number of perspectives that can be given by the user. It is equal to $\frac{360^\circ}{\text{number of perspectives}}$. 
TabletopScanAction - Implementation (3)

Alignment of the perspectives:

▶ The alignment starts by rotating all segmented point clouds back to the position of the first perspective

▶ For example: The second point cloud/perspective is rotated back to the first position with the second saved center of rotation

▶ The third point cloud is rotated back to the position of the second point cloud with the third center of rotation and then rotated again to the first position with the second center of rotation

▶ etc…
Software - GripperScanAction

The GripperScanAction:

- Implementation of the Gripper Scan approach
- Uses several other modules, especially the ScanObjectInGripperAction and SegmentObjectInGripperServer
- Outputs the point clouds of the object
GripperScanAction - Implementation (1)

The implementation consists of many steps

1. Recognize the object on the table and grasp it (TabletopGraspAction)
2. Move the object to the first predefined pose in front of the depth sensor (MoveArmToPoseAction)
3. Create a point cloud and segment the points of the object from the arm and background (SegmentObjectInGripperServer)
4. Transform the point cloud into the gripper coordinate system
5. Rotate the gripper to create a new perspective
6. Repeat steps 3-5 until the object has been rotated by 360°
GripperScanAction - Implementation (2)

7. Move the object to the second predefined pose in front of the depth sensor (MoveArmToPoseAction)

8. Repeat steps 3-5 until the object has been rotated by 360°

9. Place the object back on the table (TabletopPlaceAction)

10. Merge all segmented point clouds to a single model

Alignment of the point clouds is not needed since the object does not move in the gripper coordinate system
Software - ScanObjectInGripperAction

The ScanObjectInGripperAction:

- Scans an object in the gripper by rotating the gripper and creating point clouds with the depth sensor
- Uses the SegmentObjectInGripperServer to separate the points of the object from the arm and the background
- Merges the point clouds of the perspectives
- Makes the result available to the GripperScanAction
Software - SegmentObjectInGripperServer

The SegmentObjectInGripperServer:

- Service that segments the object in the gripper
- Uses markers (bounding boxes) to separate the arm and the object
- Those markers are attached to the different joints of the arm and the gripper
Software - SegmentObjectInGripperServer

Markers visualized in RViz:
Software - TabletopGraspAction & TabletopPlaceAction

The TabletopGraspAction:
- Server to grasp an object on the table with several options
  - right or left arm
  - top grasp, side grasp
  - lift distance
  - desired grasps (if the object has been grasped and you already know successful grasps)

The TabletopPlaceAction:
- Server to place an object back on the table
Software - GetArmPoseServer

The GetArmPoseServer:

- Service that computes the pose of the arm
- Pose of the arm is the pose of the wrist coordinate system ('r_wrist_roll_link' or 'l_wrist_roll_link')
- It can compute the pose for the right or left arm and can transform that pose into a desired coordinate frame
Software - MoveArmToPoseAction

The MoveArmToPoseAction:

- Server to move the arm to a predefined pose, e.g. the side
- It is possible to disable collision detection or point the head at the gripper after the movement is finished
Results - Tabletop Scan

Point cloud of the coffee cup taken with 4 perspectives:

Point cloud of the coffee cup taken with 8 perspectives:
Results - Tabletop Scan

Point cloud of the coffee cup taken with 16 perspectives:

Point cloud of the coffee cup taken with 24 perspectives:
## Results - Tabletop Scan: Amount of points (1)

Amount of points in relation to the number of perspectives:

<table>
<thead>
<tr>
<th>ID</th>
<th>Object</th>
<th>4</th>
<th>8</th>
<th>16</th>
<th>24</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>cup</td>
<td>11563</td>
<td>22503</td>
<td>47119</td>
<td>71917</td>
</tr>
<tr>
<td>2</td>
<td>video cassette</td>
<td>18165</td>
<td>41271</td>
<td>80805</td>
<td>-</td>
</tr>
<tr>
<td>3</td>
<td>rubik’s cube</td>
<td>5909</td>
<td>13090</td>
<td>25338</td>
<td>-</td>
</tr>
<tr>
<td>4</td>
<td>salt shaker</td>
<td>3295</td>
<td>6153</td>
<td>12617</td>
<td>-</td>
</tr>
<tr>
<td>5</td>
<td>big vase</td>
<td>17439</td>
<td>32873</td>
<td>67326</td>
<td>102012</td>
</tr>
</tbody>
</table>
## Results - Tabletop Scan: Amount of points (2)

Amount of points in relation to the number of perspectives:

<table>
<thead>
<tr>
<th>ID</th>
<th>Object</th>
<th>4</th>
<th>8</th>
<th>16</th>
<th>24</th>
</tr>
</thead>
<tbody>
<tr>
<td>6</td>
<td>small vase</td>
<td>8107</td>
<td>16816</td>
<td>33217</td>
<td>-</td>
</tr>
<tr>
<td>7</td>
<td>chess king</td>
<td>1755</td>
<td>5352</td>
<td>10695</td>
<td>-</td>
</tr>
</tbody>
</table>
Results - Gripper Scan

Point cloud of the coffee cup taken with 4 perspectives:

Point cloud of the coffee cup taken with 8 perspectives:
Results - Gripper Scan

Point cloud of the coffee cup taken with 16 perspectives:

Point cloud of the coffee cup taken with 24 perspectives:
### Results - Gripper Scan: Amount of points (1)

Amount of points (side grasp / top grasp) in relation to the number of perspectives:

<table>
<thead>
<tr>
<th>ID</th>
<th>Object</th>
<th>4</th>
<th>8</th>
<th>16</th>
<th>24</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>cup</td>
<td>-/32832</td>
<td>-/58272</td>
<td>-/118372</td>
<td>-/177348</td>
</tr>
<tr>
<td>2</td>
<td>video cassette</td>
<td>64370/59404</td>
<td>127686/111540</td>
<td>249542/229166</td>
<td>-</td>
</tr>
<tr>
<td>3</td>
<td>rubik’s cube</td>
<td>-/17930</td>
<td>-/31665</td>
<td>-/67399</td>
<td>-</td>
</tr>
<tr>
<td>4</td>
<td>salt shaker</td>
<td>-/8030</td>
<td>-/16318</td>
<td>-/34605</td>
<td>-</td>
</tr>
<tr>
<td>5</td>
<td>big vase</td>
<td>59314/52532</td>
<td>109016/87933</td>
<td>219069/205963</td>
<td>318674/288240</td>
</tr>
</tbody>
</table>
## Conclusion - Results

Object segmentation and registration of point clouds from different perspectives aided by a service robot.

### Results - Gripper Scan: Amount of points (2)

Amount of points (side grasp / top grasp) in relation to the number of perspectives:

<table>
<thead>
<tr>
<th>ID</th>
<th>Object</th>
<th>4</th>
<th>8</th>
<th>16</th>
<th>24</th>
</tr>
</thead>
<tbody>
<tr>
<td>6</td>
<td>small vase</td>
<td>-/ 26023</td>
<td>-/ 49181</td>
<td>-/ 103484</td>
<td>-</td>
</tr>
<tr>
<td>7</td>
<td>chess king</td>
<td>-/ 6558</td>
<td>-/ 12918</td>
<td>-/ 26542</td>
<td>-</td>
</tr>
<tr>
<td>8</td>
<td>fork</td>
<td>-</td>
<td>-/ 25248</td>
<td>-/ 48006</td>
<td>-</td>
</tr>
<tr>
<td>9</td>
<td>knife</td>
<td>-</td>
<td>-/ 32639</td>
<td>-/ 75296</td>
<td>-</td>
</tr>
<tr>
<td>10</td>
<td>spoon</td>
<td>-</td>
<td>-/ 28816</td>
<td>-/ 57653</td>
<td>-</td>
</tr>
</tbody>
</table>
## Results - Scan time

Scan times for the big vase in relation to the number of perspectives:

<table>
<thead>
<tr>
<th>Approach</th>
<th>4</th>
<th>8</th>
<th>16</th>
<th>24</th>
</tr>
</thead>
<tbody>
<tr>
<td>Complete</td>
<td>6:40min</td>
<td>10:57min</td>
<td>19:33min</td>
<td>28:52min</td>
</tr>
<tr>
<td>Tabletop</td>
<td>2:24min</td>
<td>4:43min</td>
<td>9:22min</td>
<td>14:46min</td>
</tr>
<tr>
<td>Gripper (side)</td>
<td>2:10min</td>
<td>3:09min</td>
<td>5:06min</td>
<td>6:57min</td>
</tr>
<tr>
<td>Gripper (top)</td>
<td>2:02min</td>
<td>3:02min</td>
<td>4:59min</td>
<td>7:01min</td>
</tr>
</tbody>
</table>
Breuckmann Professional 3D Scanner

Point cloud taken with a professional 3D scanner from Breuckmann (ca. 60,000 points):

Point cloud of a single perspective from the Tabletop Scan (ca. 3000 points):
Summary

▶ Software shows possibility of creating 3D models of objects with a depth sensor and a service robot
▶ Independent of shape of object, object only has to be graspable by the robot
▶ Alignment relies completely on the robots coordinate systems and transformation
▶ Quality of scans depends on the surface of objects
▶ The process is mostly autonomous, the user only has to move the robot in front of the table with the object
Conclusion - Summary

Object segmentation and registration of point clouds from different perspectives aided by a service robot

Outlook

- Use of Kinect 2.0 or other better depth sensors that can improve density and quality of scans
- This could also improve the grasping process
- Extend the Tabletop Scan for different grasps (currently top grasp only)
- Scanning of more than one object on the table
- Adding RGB information to the point clouds (currently limited by the Tabletop Segmentation)
Thank you for your attention!

Nikolas Slottke
7slottke@informatik.uni-hamburg.de