A Rapid Development Methodology for an Autonomous Warehouse Picking Robot

Kris Hauser
ECE & MEMS & CS Departments
Duke University

with Miles Aubert, Anne W. Draelos, Mark Draelos, Yihui Feng, Brenton Keller, Jianqiao Li, Branch Vincent, Shengbin Wu, Kevin Zhou
AMAZON PICKING CHALLENGE 2015

Team Duke, Intelligent Robotics class team
AMAZON PICKING CHALLENGE 2016

Team Duke, Intro to Robotics and Automation / Advanced Robot System Integration class team (3 undergrads, 1 grad)

Amazon Picking Challenge
Friday July 1st - Stow Task Finals

This competition challenges entrants to build original robot hardware and software to attempt the challenging task of picking a variety of different items from shelves (pick) and putting them back (store). The challenge takes place over four days.

<table>
<thead>
<tr>
<th>Time</th>
<th>Team</th>
<th>Time</th>
<th>Team</th>
</tr>
</thead>
<tbody>
<tr>
<td>09:30</td>
<td>C²M</td>
<td>13:50</td>
<td>AA-team</td>
</tr>
<tr>
<td>09:55</td>
<td>MIT</td>
<td>14:15</td>
<td>KTH</td>
</tr>
<tr>
<td>10:20</td>
<td>PFN</td>
<td>14:40</td>
<td>microRecycler</td>
</tr>
<tr>
<td>10:45</td>
<td>Dataspeed-Grizzly</td>
<td>15:05</td>
<td>IITK-TCS</td>
</tr>
<tr>
<td>11:10</td>
<td>Rutgers ARM</td>
<td>15:45</td>
<td>Applied Robotics</td>
</tr>
<tr>
<td>11:35</td>
<td>Duke</td>
<td>16:10</td>
<td>HARP</td>
</tr>
<tr>
<td>13:00</td>
<td>ACRV</td>
<td>16:35</td>
<td>Delft</td>
</tr>
<tr>
<td>13:25</td>
<td>Team K</td>
<td>17:00</td>
<td>NimbRo Picking</td>
</tr>
</tbody>
</table>
LESSONS LEARNED

Expertise vs determination?
Existing tech + many test cycles = “ok” performance
Be agile: rapid prototyping phase is never over!

2017 challenges:
Denser clutter, packing / unpacking, novel objects
Need to integrate research products into a complex system
Need to comprehensively test (and hack) research products offline on real scenarios
Developer-in-the-loop
SW INFRASTRUCTURE REQUIREMENTS

Many CPUs (sensor capture, vision processor w/ GPU, motor control Raspberry Pi, robot controllers, master / planning)

Frequent HW reconfiguration

Short design-debug-test cycle times

Visualization, state introspection, logging

Many, diverse team members (CS, EE, ME, BME, Physics) developing simultaneously
SOFTWARE INFRASTRUCTURE

Global key-value data store
• persistence across process lifetimes
• concurrent access from multiple processes in a networked environment
• partial and complete saving and loading in a human-readable format
• installation using only dependencies available via the Python package manager
• stateless and simple client API in Python, C/C++, and JavaScript

Integrated modeling and visualization

Finite state machine

User Interface
SOFTWARE INFRASTRUCTURE

Global key-value data store

Integrated modeling and visualization
- Klamp’t 0.7 ([http://klampt.org](http://klampt.org))
- cross-platform (Win, Mac, *nix)
- Python API used
- displays robot, CAD model, point cloud, frames, camera frustums, marked regions, trajectories
- kinematics, dynamics, planning, physics simulation, sensor simulation built-in

Finite state machine

User Interface
SOFTWARE INFRASTRUCTURE

Global key-value data store

Integrated modeling and visualization

Finite state machine

• 5 main phases containing internal states
• configurable from UI
• trigger calls to all algorithms
• error handling states

User Interface
SOFTWARE INFRASTRUCTURE

Global key-value data store

Integrated modeling and visualization

Finite state machine

User Interface

- autonomous run, step through, or manual actions
- toggle manual or autonomous actions
- toggle visual-debugging breakpoints
EARLY-STAGE DESIGN: END EFFECTOR MOUNTING

HYBRID VACUUM/MECHANICAL GRIPPER

Vacuum works poorly for heavy / porous / slippery packaging / tiny exposed surfaces / irregular geometry

Slim profile (7cm)

14cm maximum span

3D printing for prototypes, final machined aluminum

Swappable fingers
Many 3D sensors (RealSense, PhotoNeo)
Segment then identify

Tuned graph cut algorithm [Felzenszwalb and Huttenlocher, 2004]
LAB+D feature space
Hyperparameters optimized to match manual segmentations using genetic algorithm
Many 3D sensors (RealSense, PhotoNeo)

Segment then identify

Deep learning for identification

Training data
- Combined Amazon’s provided images and images taken from a rotation stage
- Augmented with random rotation, dilation, and shrinkage

Validation dataset
- 540 manually segmented/labeled images from the shelf and tote

CNN accuracy 61.2%; Top-3 81.4%

40-class confusion matrix
Over-generate grasps
Separate grasp scoring / object identification steps
Vacuum gripper: find planar regions
Mechanical gripper: find parallel sides around objects
GRASP PLANNING STRATEGY

Over-generate grasps
Separate grasp scoring / object identification steps
Vacuum gripper: find planar regions
Mechanical gripper: find parallel sides around objects

Geometrically-scored regions
GRASP PLANNING STRATEGY

Over-generate grasps

Separate grasp scoring / object identification steps

Vacuum gripper: find planar regions

Mechanical gripper: find parallel sides around objects

Parallel surface identification
PACKING STRATEGY

Online bin packing heuristics
- Height
- Corner preference
- Stack count

Object bounding box on heightfield
- From denoised, cropped point cloud
- Rotations considered
- Re-sense heightfield after drop

Background with some obstacles

Height-minimizing packing of three objects
ROBOT + SIM DEVELOPMENT IN PARALLEL
ONGOING WORK

Novel object identification
Gripper integration
“Fancy” manipulation primitives
Test, test, test!

Manipulation primitive: pushing aside to retrieve a blocked object
THANK YOU!

Past Team Members
- 2015: Mark Draelos, Brenton Keller, Andrew Hutchins, Miles Aubert, Yilun Zhou
- 2016: Bernie Amaldoss, Hayden Bader, Hyunsoo Kim, Yilun Zhou

Sponsors
- Lord Foundation of North Carolina

More info & software available at
http://motion.pratt.duke.edu
and
http://klampt.org