Action-Related Places for Mobile Manipulation

Andreas Fedrizzi, Freek Stulp, Michael Beetz
Intelligent Autonomous Systems Group

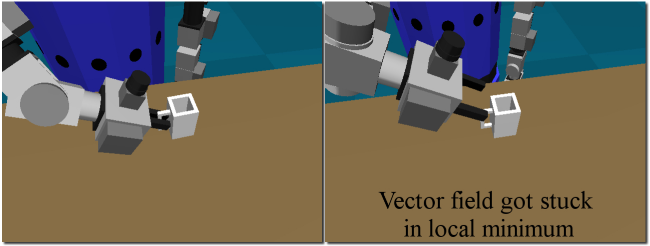
07.12.2009, at Humanoids 2009

Motivation

Mobile Manipulation

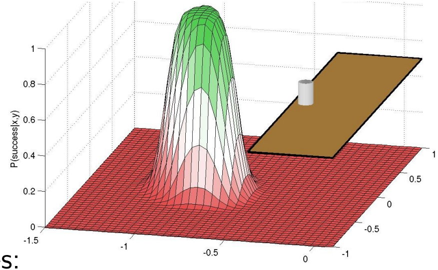
go to a place *l* in order to pick up and place an object *o*

- Problem: manipulation often fails, even if objects are in reach
- Possible Reasons: singularities, local minima, perceptual problems, . . .



Action-Related Places (ARPLACEs)

- Representation:
  - Discrete grid of grasp success probabilities
  - Map base locations to grasp success probabilities



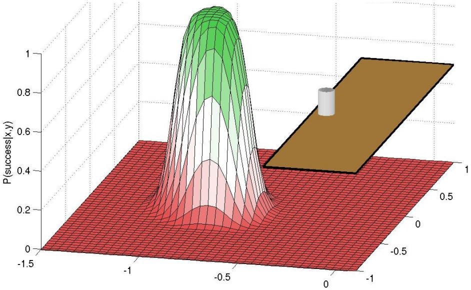
- Advantages:
  - least-commitment representation (obstacles)
  - represents "expected utility"
  - takes state estimation uncertainties into account
  - are learned from and are grounded in observed experience

Freek Stulp, Andreas Fedrizzi, and Michael Beetz. Learning and Performing Place-based Mobile Manipulation. In *Proceedings of the 8th International Conference on Development and Learning (ICDL), 2009.*

Features of ARPLACE

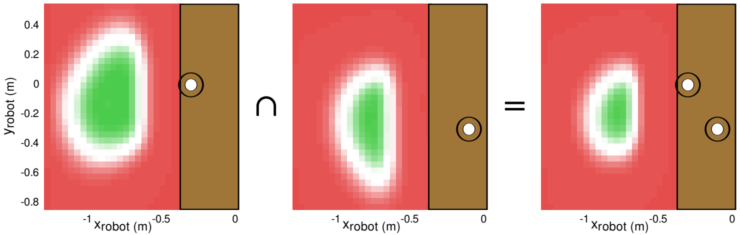
1. Update ARPLACE distribution as you go
   - More information: more accurate ARPLACE
   - Do not unnecessarily commit to a specific location
   - Delay decision about exact position

   ⇒ Robustness in real world applications with uncertainties and noisy sensor data



Features of ARPLACE

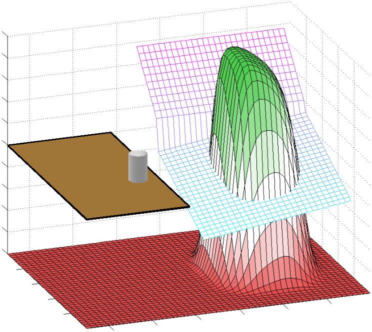
1. Update ARPLACE distribution as you go
2. Merge ARPLACEs for joint tasks
   - Piecewise multiplication of corresponding probabilities
   - Integrated in a transformational planner



Andreas Fedrizzi, Lorenz Moesenlechner, Freek Stulp, Michael Beetz. Transformational Planning for Mobile Manipulation based on Action-related Places. In *Proceedings of the International Conference on Advanced Robotics (ICAR), 2009.*

Features of ARPLACE

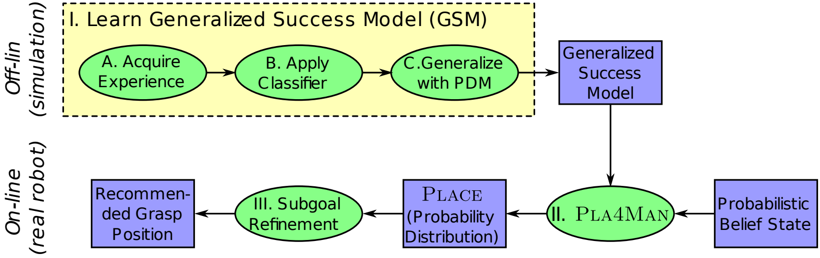
1. Update ARPLACE distribution as you go
2. Merge ARPLACEs for joint tasks
   - Piecewise multiplication of corresponding probabilities
   - Integrated in a transformational planner
3. Optimization of secondary criteria
   - Such as execution duration



Freek Stulp, Michael Beetz. Refining the execution of abstract actions with learned action models. In *Journal of Artificial Intelligence Research (JAIR), 2008.*

Acquisition of ARPlace (System Overview)

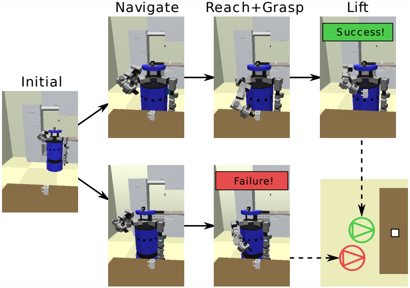
- ARPlace is difficult to determine analytically
  - Interaction of many hard- and software modules
  - Many skills and skill paramerizations involved
  - Many state estimation uncertainties to take into account
- Instead: learned from observed experience

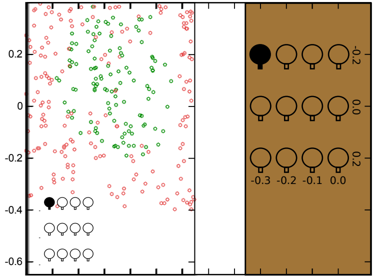


Michael Beetz et. al. Generality and Legibility in Mobile Manipulation. In *Autonomous Robots Journal (Special Issue on Mobile Manipulation), 2009.*

Gather Training Data

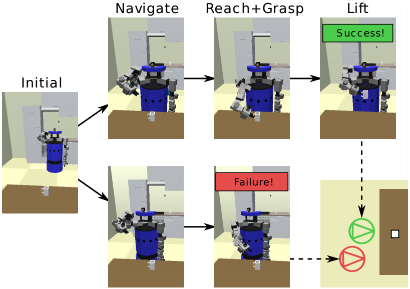
- Accurate model of our robot in simulation
- Execute navigate-reach-grasp-lift sequence and record result

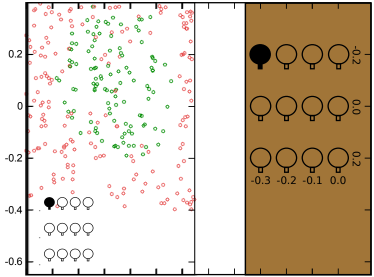




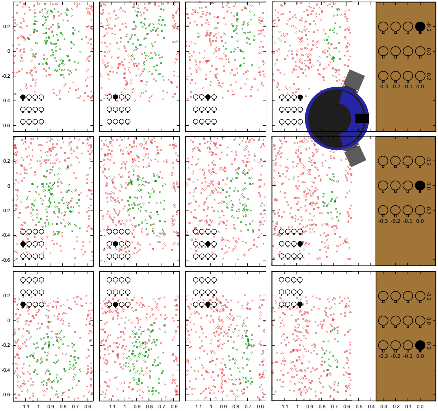
Gather Training Data

- Accurate model of our robot in simulation
- Execute navigate-reach-grasp-lift sequence and record result
- Vary task-relevant parameters, i.e. 12 object positions

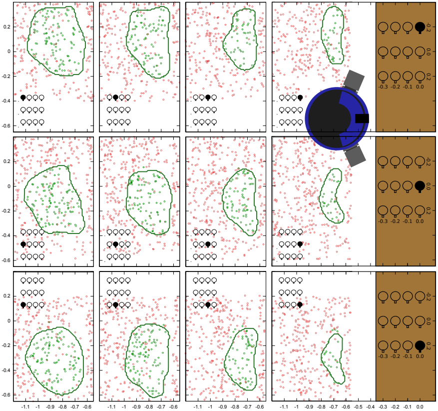




Learn Classification Boundaries

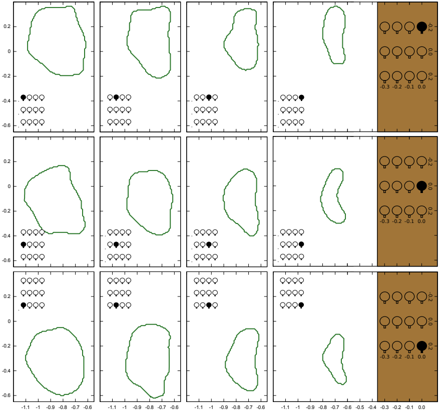


Learn Classification Boundaries



- Use Support Vector Machines to learn classification boundary

Learn Classification Boundaries



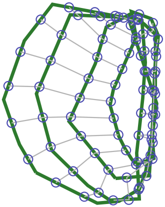
- Use Support Vector Machines to learn classification boundary
- Already very compact model of hardware, kinematics, skills
- Next step: generalize over these classification boundaries

Compute Generalized Success Model

- Goal: Compile *n* classification boundaries into one model
- Use a Point Distribution Model (PDM)

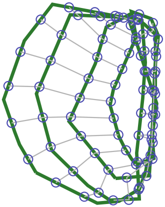
Compute Generalized Success Model

- Goal: Compile *n* classification boundaries into one model
- Use a Point Distribution Model (PDM)
  1. align points on classification boundaries



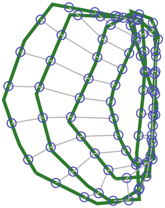
Compute Generalized Success Model

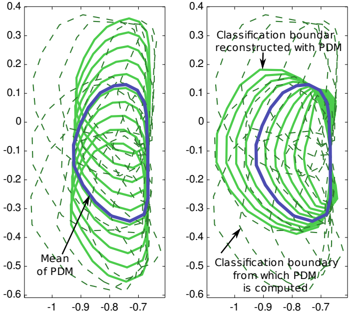
- Goal: Compile *n* classification boundaries into one model
- Use a Point Distribution Model (PDM)
  1. align points on classification boundaries
  2. perform PCA on points



Compute Generalized Success Model

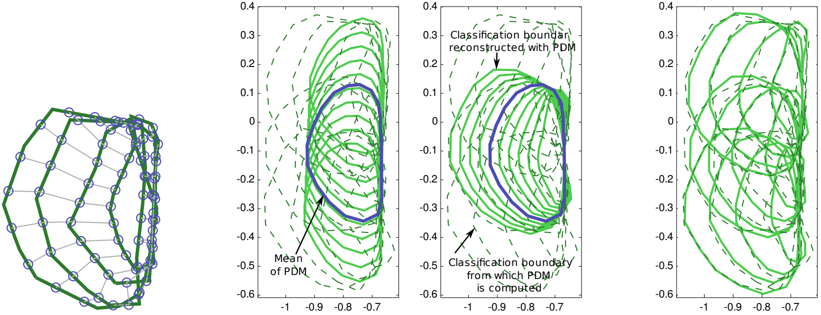
- Goal: Compile *n* classification boundaries into one model
- Use a Point Distribution Model (PDM)
  1. align points on classification boundaries
  2. perform PCA on points
  3. compute deformation modes





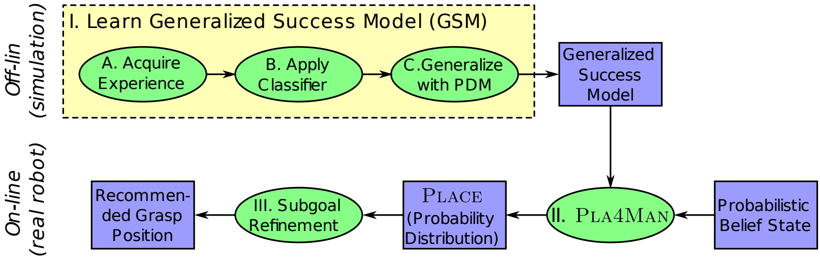
Compute Generalized Success Model

- Goal: Compile *n* classification boundaries into one model
- Use a Point Distribution Model (PDM)
  1. align points on classification boundaries
  2. perform PCA on points
  3. compute deformation modes
- Relate deformation modes to task-relevant parameters



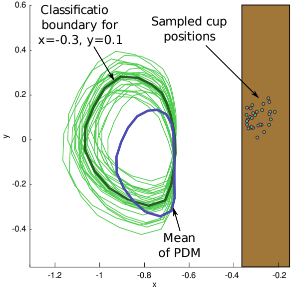
Compute ARPlaces

- Next step
  - Use Generalized Success Model to compute ARPlaces on-line



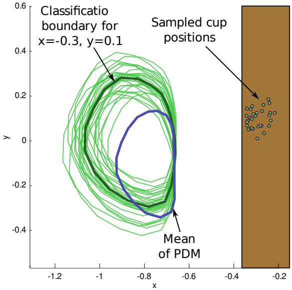
Computing ARPLACEs online (Pla4Man)

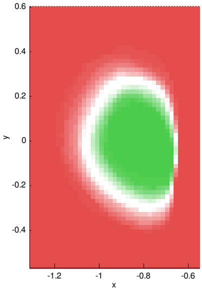
- Monte Carlo simulation
  1. Sample from target object distribution
  2. Compute corresponding classification boundaries



Computing ARPLACEs online (Pla4Man)

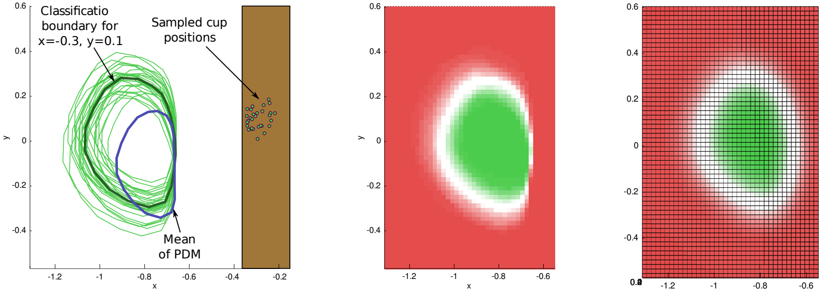
- Monte Carlo simulation
  1. Sample from target object distribution
  2. Compute corresponding classification boundaries
  3. Relative sum over classification boundaries





Computing ARPLACEs online (Pla4Man)

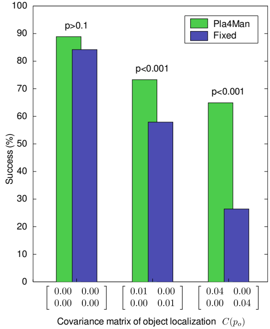
- Monte Carlo simulation
  1. Sample from target object distribution
  2. Compute corresponding classification boundaries
  3. Relative sum over classification boundaries
  4. Convolution with robot distribution



Freek Stulp, Andreas Fedrizzi, Michael Beetz. Action-Related Place-Based Mobile Manipulation. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2009.*

Evaluation

- Pla4Man performed significantly better than the best hardcoded strategy called Fixed



- Merging ARPlaces reduced the average execution time from 48s to 32s

Summary

Off-line Generalized Success Model (GSM) learned from observed experience

- Takes interactions between hard-/software modules and robot skills into account

On-line Monte-carlo simulation with GSM to account for state estimation uncertainties: ARPLACE

- Very fast, so ARPLACE can be updated as knowledge about the current state is updated
- Optimization of secondary constraints like merging ARPLACEs can save time
- Probabilistic representation enables least-commitment approach

End

Thank you for your attention