Learning Object Deformation Models for Robot Motion Planning

Barbara Frank, Cyrill Stachniss, Rüdiger Schmedding, Matthias Teschner, Wolfram Burgard

European Workshop on Deformable Object Manipulation
March 20, 2014 — Lyon, France
Motivation

- Real environments contain deformable objects such as plants or curtains
- So far: robots ignore or avoid such obstacles
- This work: considers the deformation properties of obstacles when planning robot motions
Motivation – Example
Key Questions

How can a robot

- learn about the deformation properties of objects?
- efficiently consider object deformations during planning?
- successfully navigate among deformable objects?
Key Contributions

- Parameter estimation of deformable objects with a manipulation robot

- Efficient approximation of object deformation cost functions for planning

- Applications to different real robots
Planning with Deformation Costs

**offline**

- Learn object deformation model
- Generate training examples using simulations
- Train GP to predict cost
- Construct roadmap

**online**

- Roadmap cost by GP regression
- Efficient planner that trades off motion- and deformation costs
- Train GP to predict cost
Deformation Simulation

- Dynamic simulation
  - Collision handling
  - Time integration

- Finite element model
  - Assumption: linearly elastic, isotropic, homogeneous material
  - Hooke’s law: linear relation between stress and strain

\[ f = K(E, \nu)q \]

Young’s modulus  Poisson’s ratio
Geometric Models for Simulation

- 3D volumetric representation
  - Register point clouds from different view points into a consistent surface mesh
  - Compute a tetrahedral mesh from the surface mesh
Acquisition of Deformation Data

Measurement:
- Point cloud
- Applied force
- Contact point
Parameter Estimation

- Comparison of observed and simulated deformation
- Error function: distance between registered surfaces
  \[
  \text{MSE} = \frac{1}{n} \sum_{i=1}^{n} \left\| p_i - q_i^* \right\|^2,
  \]
  \[
  q_i^* = \arg\min_j \left\| p_i - q_j \right\|^2
  \]
- RPROP to optimize Young’s modulus $E$ and Poisson’s ratio $\nu$
Results: Learned Model - Teddy

- Estimated parameters: \( E = 31.4 \frac{N}{dm^2}, \nu = 0.4 \)
- Residual MSE: 19.4 mm\(^2\)
Evaluation of Learned Models

Force error (%)  Residual MSE (mm²)

Foam  Cube  Ball  Teddy
Our Approach to Efficient Motion Planning

- Sample a subset of possible motions and simulate deformations before planning.
- Estimate the deformation costs of new motions by **Gaussian process** regression.
- Planning framework: Probabilistic Roadmaps (PRM)
Gaussian Processes (GPs)

- GPs are a framework for **non-parametric** regression
- Model the data points (here deformation costs) as jointly Gaussian

\[
y_1, \ldots, y_n \mid x_1, \ldots, x_n \sim \mathcal{N}(\mu, \Sigma)
\]

- Predictive model for an input trajectory:

\[
p(y_* \mid x_*, x_{1:n}, y_{1:n}) \sim \mathcal{N}(\mu, \sigma^2)
\]

- Provides a **mean** and a **predictive variance**
- A covariance function \( k(x_p, x_q) \) models the influence of the data points on the query point
Gaussian Processes (GPs)

- Neural network covariance function
  \[ k(x_i, x_j) = \sigma_f^2 \arcsin \left( \frac{\beta + 2x_i^T \Sigma x_j}{\sqrt{(1 + \beta + 2x_i^T \Sigma x_i)(1 + \beta + 2x_j^T \Sigma x_j)}} \right) \]

- ... the covariance function requires hyperparameters
  \[ \theta = \{ \Sigma, \beta, \sigma_f, \sigma_n \} \]

- Learning the hyperparameters by maximizing the likelihood of the training data
  \[ \theta_* = \arg \max_{\theta} \log p(y_1, \ldots, y_n \mid x_1, \ldots, x_n, \theta), \]

- Popular: maximization via gradient methods
- Problem: significant cost of learning the GP from data
Problem Decomposition

- We need many samples to accurately approximate the deformation costs
- Problem: GP learning has cubic runtime complexity in the number of samples due to matrix inversion

Local Approximation

- Store all samples in a KD-tree for efficient nearest neighbor queries
- Select only trajectory samples that are “close” to build the GP
Results: Deformation Cost Prediction

- Comparison: GP-regression vs. baseline
Results: Statistical Evaluation

GPD-SE  GPD-NN  GPO-SE  GPO-NN  baseline

MAE  sMSE  sMSLL

Curtain-A  Curtain-R  Duck  Teddy  Foam
Planning for Manipulators in 3D

Experimental setup – deformable foam mat

3D-model for roadmap generation and planning
Example Planning Task

Consider obstacles as rigid: no path

Ignore deformable obstacles: shortest path

Our planner: minimize trade-off between motion and deformation costs
Robot Navigation in 2D

Experimental setup – Robot in a corridor with curtains

3D-deformation model for generation of samples

2D-gridmap + curtain position for roadmap generation
Example Planning Task

- Our planner optimizes the trade-off between travel costs and object deformations.

- During path execution: sensor-based collision avoidance for non-deformable objects.
Related Work

- Deformation models
  - FE for surgical simulations: Picinbono et al. (2001)

- Parameter estimation
  - With robots: Lang et al. (2002), Boonvisut et al. (2012)
Related Work

- Robot motion planning
  - Deformable robots: Kavraki et al. (1998), Bayazit et al. (2002)
  - Surgical tools: Gayle et al. (2005), Alterovitz et al. (2009)
  - Completely deformable environments: Rodriguez et al. (2006)
  - Pre-computations for deformable robots: Mahoney et al. (2010)

- Model-predictive control: Jain et al. (2013)
Conclusions

- Learning deformation models with a manipulation robot for simulation and path planning
- Motion planning system that considers object deformations
- Object deformation cost functions based on Gaussian process regression speed up planning
- Probabilistic approach to collision avoidance that distinguishes between deformable and non-deformable obstacles
Thanks for Your Attention!