Learned Critical Probabilistic Roadmaps for Robotic Motion Planning

Brian Ichter, Aleksandra Faust @ Robotics at Google
{ichter, faust}@google.com

Introduction
- Sampling-based motion planning uses uniform probing samples to build an arbitrarily accurate, implicit representation of the state space
- Often only a few critical states are necessary to parameterize solution trajectories
- **Key Idea:** Learn to recognize the critical states and use them to construct a hierarchical PRM, leveraging these critical states more heavily

Contributions
- Compute criticality via betweenness centrality, a graph-theoretical importance to shortest paths
- **Offline:** Learn to predict criticality from local features with a Convolutional Neural Network
- **Online:** construct a Critical Probabilistic Roadmap
  - Connect non-critical points locally
  - Connect critical points globally
- **Shows 100x improvement in success, 10x in cost**

Critical States Identification
- Graph theoretic approaches allow state space critical states. Compared:
  - label propagation- unstable and poorly defined with less clear bottlenecks
  - K-cuts- requires fixed number of cuts, which heavily influences states
  - many graph centrality indicators
- **Selected betweenness centrality:**
  - computes the number of all-pairs shortest paths that pass through a node
  - can be computed approximately, very quickly
  - added an additional smoothing step to discount skippable free-space states
- Learn a network, $h_\theta(x, y)$, to predict sample $x$’s criticality from problem features $y$

Offline: Identify and Learn Samples
- Build PRMs
- Solve several one-to-all problems with smoothing
- Compute criticality
- Learn to predict, $h_\theta(x, y)$

Online: Critical Probabilistic Roadmaps
- Construct a Critical PRM by sampling:
  - $(\lambda \log n)$ critical, globally connected states
  - $(n - \lambda \log n)$ uniform, locally connected states
- Preserves the theoretical guarantees of SBMP, asymptotic optimality and $O(n \log n)$ complexity

Results
- On narrow passage problem:
  - Identifies narrow passages
  - 100x speedup in success rate
  - 10x speedup in solution cost
- On indoor dataset:
  - Trained on local image
  - Identifies doorways, hallways
  - Ignores open area
  - Success and cost to come