A Comparison of Dimensionality Reduction Techniques for Robotic Motion Planning

Brandon Araki Alex Wallar

May 10, 2016

Dimensionality reduction is a key component of making robotic motion planning fast and efficient. A rigid body has six degrees of freedom (DoF), so a multi-link robot such as a humanoid or snake robot can have dozens or hundreds of DoF. The rotational and translational transformations of a rigid body can be described with the 3-dimensional Special Euclidean group (known as SE(3)). SE(3) is homeomorphic to the topological space R3xRP3 (where RP3 is the 3-dimensional real projective plane). Therefore it is easy to imagine a multi-link robot with an arbitrarily complex state space. The picture is further complicated by the addition of obstacles into a robot's world. The state space for motion planning, which must take into account both the robot and obstacles, is called the configuration space.

Motion planning essentially consists of a search over the configuration space from a start configuration to a goal configuration. A huge number of methods for searching through the configuration space have been developed, most of which can be divided into two classes, sampling-based motion planning and combinatorial motion planning. In sampling-based motion planning, obstacles in the C-space are defined implicitly so that the path is constructed by randomly or pseudorandomly sampling points from the C-space and doing collision checking. In combinatorial motion planning, obstacles are defined explicitly and a complete search of the C-space is made. Because of the difficulty of explicitly defining all of the geometry in a potentially complex world, sampling-based motion planning techniques such as the Rapidly-exploring Random Tree (RRT) and Probabilistic RoadMap (PRM) planning are the most popular motion planning techniques.

[RRT* algorithm] [PRM* algorithm]

We used RRT* and PRM* in our project. The * indicates that these are the asymptotically optimal versions of RRT and PRM (they are guaranteed to find the optimal path as time goes to infinity). [Description of how RRT* and PRM* work]

Despite the efficiency of sampling-based search, the high dimensionality of the configuration space of complex robots means that motion planning for many robots is slow or infeasible. This is because of the ”curse of dimensionality”, in which the size of the search space increases exponentially with the dimension. One way to speed up motion planning is to reduce the dimensionality of the C-space and do search in this reduced dimensionality space. For this project, we tested the performance of a number of dimensionality reduction techniques to see which one was the most effective in speeding up motion planning while providing reliable obstacle avoidance.

We used FeatureAgglomeration, TruncatedSVD, PCA, KernelPCA, RandomizedPCA, and our own ”Trained Johnson-Lindestrauss” to perform dimensionality reduction.

PCA: Principle Component Analysis, or PCA, transforms a set of variables in their ”principle components”, or linearly uncorrelated variables.

ncollisions, duration, length, num failed. Run each one 100 times

Algorithm 1 PRM* [1]

```
$V \leftarrow \{x_{init}\} \cup \{SampleFree_i\}_{i=1,\ldots,n}$
$E \leftarrow \emptyset$
for all $v \in V$ do
    $U \leftarrow$ Near$(G = (V, E), v, \gamma \cdot \mathrm{PRM}(\log(n)/n)^{1/d}) \backslash \{v\}$
    for all $u \in U$ do
        if CollisionFree$(v, u)$ then
            $E \leftarrow E \cup \{(v, u), (u, v)\}$
return $G = (V, E)$
```

References

[1] Sertac Karaman and Emilio Frazzoli. Sampling-based algorithms for optimal motion planning. *The International Journal of Robotics Research*, 30(7):846–894, 2011.

Algorithm 2 RRT* [1]

```
$V \leftarrow \{x_{init}\}$
$E \leftarrow \emptyset$
**for** $i = 1$ **to** $n$ **do**
    $x_{rand} \leftarrow \mathit{SampleFree}_i$
    $x_{nearest} \leftarrow \textsc{Nearest}(G = (V, E), x_{rand})$
    $x_{new} \leftarrow \textsc{Steer}(x_{nearest}, x_{rand})$
    **if** $\textsc{ObstacleFree}(x_{nearest}, x_{new})$ **then**
        $x_{near} \leftarrow \textsc{Near}(G = (V, E), x_{new}, \min\{\gamma \cdot \mathrm{RRT}(\log(|V|)/|V|)^{1/d}, \eta\})$
        $V \leftarrow V \cup \{x_{new}\}$
        $x_{min} \leftarrow x_{nearest}$
        $c_{min} \leftarrow \textsc{Cost}(x_{near}) + c(\textsc{Line}(x_{nearest}, x_{new}))$
        **for all** $x_{near} \in X_{near}$ **do**
            $C \leftarrow \textsc{Cost}(x_{near}) + c(\textsc{Line}(x_{near}, x_{new}))$
            **if** $\textsc{CollisionFree}(x_{near}, x_{new}) \wedge C < c_{min}$ **then**
                $x_{min} \leftarrow x_{near}$
                $c_{min} \leftarrow C$
            $E \leftarrow E \cup \{(x_{min}, x_{new})\}$
**return** $G = (V, E)$
```