Sampling

1 Sampling Uniformly on Medial Axis

[2] proposed an algorithm for sampling uniformly on medial axis. The key idea is: Randomly put a lot of sticks(line segments) with random orientations in the free-space, whatever dimensions it has. Since these sticks are uniformly distributed in the space, any surface will uniformly intersect with these sticks. Medial axis will also intersect with these sticks uniformly.

Algorithm is as below:

function UNIFORMSAMPLEMA(pntset)
    pntset ← randomly generated points in free space.
    maSamples ← Empty Set.
    l ← stick length.
    for *point* ∈ *pntset* **do**
        dir ← random direction
        stick ← (point, l, dir) ▷ start point+length+direction=a stick
        if *stick* intersects with the medial axis **then**
            sample ← stick.search()
            radius ← clearance(sample) ▷ get the clearance at this point
            $maSamples.append(ball(sample, radius))$
        end if
    end for
     return *maSamples*;
end function

2 Sampling Balls Outside MA Samples

The idea of my algorithm is when we have a lot of random points in free space and some big discs on the medial axis, we don't want to waste points inside existing discs, instead, we "push" them to the boundary of these discs, and get new samples.

Testing if a point is inside some discs seems to be a $O(n^2)$ time problem. But it can actually be done by choosing k-nearest neighbors of the centers of discs. Suppose in a 2D free space, we have some uniformly distributed points. They are uniformly distributed so, in a unit area, the expectation of the number of points is fixed. Assume a disc has radius r, its area is monotonic to r^2. Therefore the number of points inside the disc is monotonic to r^2, meaning that if we choose k-nearest points to the center of the disc, (k is a function of r^2), they are very likely to be inside the disc. There is no harm if some points are not actually inside the disc or some inside points are left, we only want to use these points to generate some points on the boundary of the disc by extending them in the direction from center to each of them. In fact, we want to choose a k that is larger than the actual number of points inside.

Since all discs touch obstacles, by choosing a larger k, we are virtually extending the radius of our discs, which results in a smaller density in arcs that are very close to obstacles than those far from obstacles.

Now we have dense points on the boundary of existing discs. Using these points to sample new small discs to cover undiscovered area has some benefits compare with points not covered by any discs:

1. They are more likely to get discs with larger radius.
2. They are less likely to left some holes between discs.
3. Experiments shows, initially, we need much less random points in the free space.

2.1 Algorithm

function SampleDiscs(n)
  $pntset \leftarrow$ n points uniformly distributed in free space
  $maSamples \leftarrow$ UniformSampleMA($pntset$) ▷ Get discs on MA
  $bndSamples \leftarrow EmptySet$ ▷ Set of discs not on MA
  $newDisc \leftarrow$ True ▷ If we can get new discs
  $lastRoundDiscs \leftarrow maSamples$

```
    while newDisc do
        newDisc ← False
        bndpnts ← EmptySet                          ▷ A set of disc boundary points
        for disc ∈ lastRoundDiscs do
            disc.KNearest(pntset); ▷ Use k-nearest to get points inside it.
            bndpnts.append(disc.getBoundaryPoints())
        end for
        lastRoundDiscs ← EmptySet
        for point ∈ bndpnts do
            if point not inside any existing discs then        ▷ O(n^2) time
                disc ← getNewDisc(point)                        ▷ get a new disc
                if disc.radius ≥ minRadius then
                    newDisc ← True
                    lastRoundDiscs.append(disc)
                    bndSamples.append(disc);
                end if
            end if
        end for
    end while
    return maSamples, bndSamples
end function
```

3 Build Topology PRM using MA Samples

In last section, our algorithm returns two sets of discs: 1. discs centered on Medial Axis, 2. discs centered on somewhere else. What is useful for representing the topology of the space using the centers of Medial Axis samples to construct a graph.

Problem is, can we simplify the graph even more? Because the simpler our graph is, the faster we can find a path, and the less memory we need to represent a very large environment.

3.1 Some thoughts

"Discs" mention in this subsection are discs centered on the Medial Axis.

1. Some discs are more important than others. For example, in a 2d word, a disc connects with only two other discs is less important than a disc

that connects with 3 or 4 or more discs, because by connecting more discs, a disc can provide more routes.

2. I guess those "more important" centers are very likely to form clusters. By clustering them, i think we can do clustering and get the center of each cluster.

3. Choose these "more important" centers as initial vertices of our simpler graph. For every pair of vertices, find the shortest chain of discs that connects them, using A* search. If the chain doesn't contain any discs centered at other important vertices, keep it.

4. Now we need to reduce some disc centers in each chain. It could be done by binary search for points that connect the start and end vertices of the chain. In the best case, all points except for the start and end vertices are reduced. In the worst case, all points are kept.

References

[1] Steven A. Wilmarth, Nancy M. Amato, Peter F. Stiller. "MAPRM: A Probabilistic Roadmap Planner with Sampling on the Medial Axis of the Free Space", In Proc. IEEE Int. Conf. Robot. Autom. (ICRA), pp. 1024-1031, Detroit, MI, May 1999. Also, Technical Report, TR98-0022, Department of Computer Science and Engineering, Texas A & M University, Nov 1998.

[2] Uniform Medial Axis Sampling.