МЕТОДЫ ПЛАНИРОВАНИЯ ПУТИ НА ОСНОВЕ ГРАФОВ

А.Д. Серенёв

Один из известных методов организации маршрутных сетей основан на применении графов видимости. В простейших случаях граф видимости строится на множестве вершин полигонов или полиэдров, являющихся геометрическими моделями препятствий сцены, и дополняется начальными и конечными точками маршрутов. Все вершины попарно соединяются линейными отрезками, которые принимаются в качестве ребер графа при условии попарной "видимости" инцидентных им вершин и отсутствия пересечения с препятствиями сцены.

$$
\begin{aligned}
&d[a] \leftarrow 0, p[a] \leftarrow 0 \\
&\forall u \in V, u \neq a \\
&while\ \exists v \notin U \\
&\quad d[u] \leftarrow \infty \\
&v \notin U \\
&v \rightarrow U \\
&\forall u \notin U, vu \in E \\
&if\ d[u] > d[v] + w[vu] \\
&\quad d[u] \leftarrow d[v] + w[vu] \\
&\quad p[u] \leftarrow (p[v], u)
\end{aligned}
$$

$V -$ множество вершин графа
$E -$ множество рёбер графа
$w[ij] -$ вес (длина) ребра ij
$a -$ вершина, расстояния от которой ищутся
$U -$ множество посещённых вершин
$d[u] -$ содержит длину пути из a в u
$p[u] -$ содержит кратчайший путь из a в u
$v -$ вершина, расстояния до которой ищется

Сложность построения графа видимости для сцены, представленной полигонами с общим числом вершин n, составляет $O(n^2)$, что является довольно затратным для применения в динамической среде.

Альтернативный метод определения маршрутов основан на использовании обобщенных диаграмм Вороного. Наиболее привлекательное свойство диаграммы Вороного заключается в том, что она является деформационным ретрактом свободного пространства. Тем самым, любой бесконфликтный путь может быть непрерывно отображен в диаграмму Вороного. Более того, метод обеспечивает наибольшее удаление от границ препятствий при условии, что движение осуществляется вдоль ребер диаграммы. Данное свойство позволяет определять наиболее "безопасные" маршруты перемещения в сцене.

```
Polygon voronoiRegion(Point &p, Point s[ ], int n, Polygon &box)
{
Edge edges = new Edge[n];
 for (int i = 0; i < n; i+= 1) {
  edges[i] = Edge(p, s[i]);
  edges[i].rot( );
 }
Polygon r = halfplaneIntersect(edges, n, box);
delete edges;
return r;
}
```

```
List < Polygon > voronoiDiagram(Point s[ ], int n, Polygon &box)
{
List < Polygon > regions = new List < Polygon >;
for (int i = 0; i < n; i+= 1) {
 Point p = s[i];
 s[i] = s[n − 1];
 regions → append(voronoiRegion(p, s, n − 1, box));
 s[i] = p;
 }
return regions;
}
```

Наивная реализация данного метода обладает высокой вычислительной сложностью O(n^2 log n), где n – общее количество вершин и граней препятствий.

Метод вероятностных маршрутных сетей (Probabilistic Roadmap Method) выращивает дерево с корнями в начальной конфигурации, используя случайные выборки из области поиска. При отрисовке каждого образца предпринимается попытка установить соединение между ним и ближайшим состоянием в дереве. Если соединение возможно (полностью проходит через свободное пространство и подчиняется любым ограничениям), это приводит к добавлению нового состояния в дерево.

```
G(V, E) ← ∅
step ← 0
while(step < N)
rand ← GenerateState(bounds)
if(rand ∈ free)
  G.V ← G.V ∪ rand
  U ← NearestNeighbours(rand, G)
  for neighbour ∈ U
    if(Path(rand, neighbour))
      G.E ← G.E ∪ (rand, neighbour)
step = step ← 1
```

Длина связи между деревом и новым состоянием часто ограничивается фактором роста. Если случайная выборка находится дальше от своего ближайшего состояния в дереве, чем позволяет этот предел, вместо самой случайной выборки используется новое состояние на максимальном расстоянии от дерева вдоль линии до случайной выборки. Затем случайные выборки можно рассматривать как управляющие направлением роста дерева, в то время как фактор роста определяет его скорость. Рост RRT может быть искажен путем увеличения вероятности выборки состояний из определенной области. В большинстве практических реализаций RRT это используется для направления поиска к целям задачи планирования. Это достигается за счет введения небольшой вероятности выборки цели в процедуру выборки состояния. Чем выше эта вероятность, тем с большей жадностью дерево растет к цели.

Таким образом методы на основе графов легко реализовать в компьютерном моделировании. Основные методы на основе графов предназначены для использования в статической окружающей среде, хотя некоторые из них могут быть преобразованы в вид, применимый для динамической среды. Поскольку граф связан с опорными точками, результат в этих методах оказывается ломаной линией. Кроме того, получаемое решение не является оптимальным: оно допустимое или относительно оптимальное. Для получившихся маршрутов необходимо учитывать размеры мобильного робота и его кинематические и динамические особенности при движении.

Для графа видимости перебор вершин является довольно затратным для применения, особенно в динамической среде. В результате получаем большое количество маршрутов, которые необходимо проанализировать. Постройка пути требует предварительного расширения границ объектов, т.к. граф строится через вершины препятствий.

Методы на основе диаграммы Вороного дают более безопасный путь с точки зрения столкновения с препятствиями, но общая длина пути увеличивается.

Метод быстро исследующих случайных деревьев каждый раз даёт разный результат после неопределённого количества итераций, что делает его затратным для динамической среды, а полученную ломаную необходимо скорректировать.

Литература

W. Liu, Path Planning Methods in an Environment with Obstacles (A Review) Bauman Moscow State Technical University, Moscow, Russia, 2018

Madhusmita Panda, Bikramaditya Das, Bidyadhar Subudhi, Bibhuti Bhusan Pati, A Comprehensive Review of Path Planning Algorithms for Autonomous Underwater Vehicles, International Journal of Automation and Computing, 2020

Alessandro Gasparetto, Paolo Boscariol, Albano Lanzutti, Renato Vidoni, Path Planning and Trajectory Planning Algorithms: a General Overview, 2015

Казаков К.А., Семенов В.А. Обзор современных методов планирования движения. Труды ИСП РАН, том 28, вып. 4, 2016