基于采样的路径规划

纲要

深蓝学院
shenlanxueyuan.com

- 第一部分：RRT*
- 第二部分：Informed RRT*

RRT*

深蓝学院 shenlanxueyuan.com

●算法流程

1. 初始化KD树并插入起点start_node
2. 在配置空间中随机采样生成点x_rand
3. 在KD树中寻找x_rand的最邻近点x_nearst
4. 在x_rand和x_nearst连线上以一定步长生成新节点x_new
5. 判断x_nearst与x_new的连线是否无障碍，若有则返回第二步，无则设置x_nearst为x_new的父节点
6. 以x_new为中心在搜索半径search_radius_内生成k个最邻近点，为x_new重新选择父节点

   6.1 分别计算x_new选取邻近点作为父节点时距离起点的代价，选取代价最小的邻近点,若与x_new的连线无障碍,则作为x_new的父节点
7. 对k个最邻近点进行重新布线

   7.1 k个最邻近点依次选择x_new作为父节点，生成的新路径大小如果小于原路径大小,且碰撞检测通过，则对该邻近点进行父节点的更新，否则切换为下一个邻近点。

RRT*

深蓝学院 shenlanxueyuan.com

- 代码实现：重新选择父节点

```
for (auto &curr_node : neighbour_nodes)
{
  double dist2cur = calDist(curr_node->x, x_new);          //计算潜在父节点到新节点的距离
  double dist_from_start(curr_node->cost_from_start+dist2cur); //计算新节点通过潜在父节点时离起点的距离
  if(dist_from_start<min_dist_from_start)                    //判断是否选择潜在父节点作为新节点的父节点
  {
    if(map_ptr_->isSegmentValid(curr_node->x,x_new))         //检测潜在父节点到新节点之间有无碰撞
    {
      min_node = curr_node;                                  //存储潜在父节点作为新节点的父节点用于后续更新
      min_dist_from_start = dist_from_start;
      cost_from_p = dist2cur;
    }
  }
}
```

RRT*

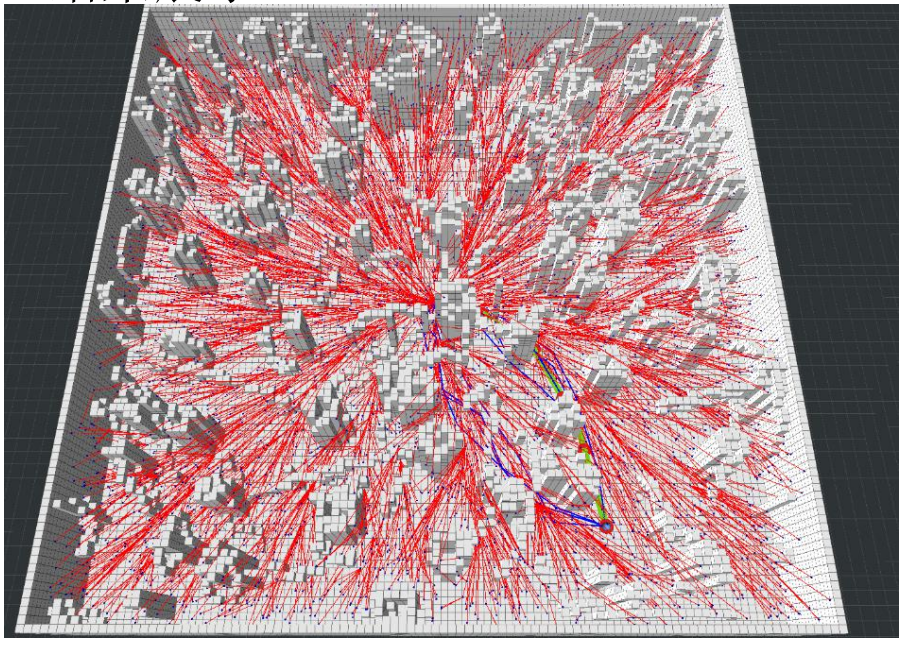
- 代码实现：重新布线rewire

```
for (auto &curr_node : neighbour_nodes)
{
  double best_cost_before_rewire = goal_node_->cost_from_start;
  // ! ---------------------------------
  double dist_from_new = calDist(new_node->x,curr_node->x);     //计算当前节点到新节点的距离
  if(curr_node->cost_from_start>new_node->cost_from_start+dist_from_new)  //如果当前节点的代价大于当前节点选取新节点作为父节点时的代价，则重新布线
  {
    if(map_ptr_->isSegmentValid(new_node->x,curr_node->x))
    {
      changeNodeParent(curr_node,new_node,dist_from_new);
    }
  }
  // ! ---------------------------------
  if (best_cost_before_rewire > goal_node_->cost_from_start)
  {
    vector<Eigen::Vector3d> curr_best_path;
    fillPath(goal_node_, curr_best_path);
    path_list_.emplace_back(curr_best_path);
    solution_cost_time_pair_list_.emplace_back(goal_node_->cost_from_start, (ros::Time::now() - rrt_start_time).toSec());
    best_path_cost = goal_node_->cost_from_start;     //记录最优的路径代价，用于informed rrt*的采样
  }
}
```

RRT*

深蓝学院 shenlanxueyuan.com

●结果展示



```
goal rcved at 18 12  2
[ INFO] [1659062209.671165883]: [RRT*]: RRT starts planning a path
[ INFO] [1659062209.753381024]: [RRT*]: first path length: 25.8938, use_time: 0.000523377
[ INFO] [1659062209.753445568]: [RRT*] final path len: 23.6043
```

可以看出与RRT相比RRT*算法搜索出的树结构并不是杂乱无章的，每个节点到起点的路径基本都是最优的。随着迭代次数的增加，以及搜索时间足够长，该算法是可以得到一条最优路径的。

纲要

深蓝学院
shenlanxueyuan.com

- 第一部分：RRT*
- **第二部分：Informed RRT***

Informed RRT*

深蓝学院 shenlanxueyuan.com

- 算法流程

Algorithm 2: Sample $(\mathbf{x}_{\text{start}}, \mathbf{x}_{\text{goal}}, c_{\max})$

1 **if** $c_{\max} < \infty$ **then**
2 $\quad c_{\min} \leftarrow \left|\left|\mathbf{x}_{\text{goal}} - \mathbf{x}_{\text{start}}\right|\right|_2$;
3 $\quad \mathbf{x}_{\text{centre}} \leftarrow \left(\mathbf{x}_{\text{start}} + \mathbf{x}_{\text{goal}}\right)/2$;
4 $\quad \mathbf{C} \leftarrow$ RotationToWorldFrame $\left(\mathbf{x}_{\text{start}}, \mathbf{x}_{\text{goal}}\right)$;
5 $\quad r_1 \leftarrow c_{\max}/2$;
6 $\quad \{r_i\}_{i=2,\ldots,n} \leftarrow \left(\sqrt{c_{\max}^2 - c_{\min}^2}\right)/2$;
7 $\quad \mathbf{L} \leftarrow \text{diag}\{r_1, r_2, \ldots, r_n\}$;
8 $\quad \mathbf{x}_{\text{ball}} \leftarrow$ SampleUnitNBall;
9 $\quad \mathbf{x}_{\text{rand}} \leftarrow (\mathbf{C}\mathbf{L}\mathbf{x}_{\text{ball}} + \mathbf{x}_{\text{centre}}) \cap X$;
10 **else**
11 $\quad \mathbf{x}_{\text{rand}} \sim \mathcal{U}(X)$;
12 **return** $\mathbf{x}_{\text{rand}}$;

Informed RRT* 与RRT*的主要区别在于采样方式，当算法搜索到第一个解后，算法就会在以起点和终点为焦点的椭圆内进行直接采样，从而提高优化效率。

Quote: Gammell J D , Srinivasa S S , Barfoot T D . *Informed RRT*: Optimal Sampling-based Path Planning Focused via Direct Sampling of an Admissible Ellipsoidal Heuristic. 2014.*

Informed RRT*

深蓝学院 shenlanxueyuan.com

●算法流程

Algorithm 2: Sample $(\mathbf{x}_{\text{start}}, \mathbf{x}_{\text{goal}}, c_{\max})$

1 **if** $c_{\max} < \infty$ **then**
2 $\quad c_{\min} \leftarrow \left|\left|\mathbf{x}_{\text{goal}} - \mathbf{x}_{\text{start}}\right|\right|_2$;
3 $\quad \mathbf{x}_{\text{centre}} \leftarrow \left(\mathbf{x}_{\text{start}} + \mathbf{x}_{\text{goal}}\right)/2$;
4 $\quad \mathbf{C} \leftarrow$ RotationToWorldFrame $\left(\mathbf{x}_{\text{start}}, \mathbf{x}_{\text{goal}}\right)$;
5 $\quad r_1 \leftarrow c_{\max}/2$;
6 $\quad \{r_i\}_{i=2,\ldots,n} \leftarrow \left(\sqrt{c_{\max}^2 - c_{\min}^2}\right)/2$;
7 $\quad \mathbf{L} \leftarrow \text{diag}\{r_1, r_2, \ldots, r_n\}$;
8 $\quad \mathbf{x}_{\text{ball}} \leftarrow$ SampleUnitNBall;
9 $\quad \mathbf{x}_{\text{rand}} \leftarrow \left(\mathbf{C}\mathbf{L}\mathbf{x}_{\text{ball}} + \mathbf{x}_{\text{centre}}\right) \cap X$;
10 **else**
11 $\quad \mathbf{x}_{\text{rand}} \sim \mathcal{U}(X)$;
12 **return** $\mathbf{x}_{\text{rand}}$;

对于旋转矩阵C的计算

$$\mathbf{C} = \mathbf{U}\,\text{diag}\{1, \ldots, 1, \det(\mathbf{U})\det(\mathbf{V})\}\,\mathbf{V}^T$$

其中矩阵U和V可以通过矩阵M的SVD分解得到

$$\mathbf{M} = \mathbf{a}_1 \mathbf{1}_1^T$$

$$\mathbf{a}_1 = \left(\mathbf{x}_{\text{goal}} - \mathbf{x}_{\text{start}}\right) / \left|\left|\mathbf{x}_{\text{goal}} - \mathbf{x}_{\text{start}}\right|\right|_2$$

$\mathbf{1}_1^T$是单位矩阵的第一列的转置

Quote: Gammell J D , Srinivasa S S , Barfoot T D . *Informed RRT*: Optimal Sampling-based Path Planning Focused via Direct Sampling of an Admissible Ellipsoidal Heuristic. 2014.*

Informed RRT*

- 算法流程

Algorithm 2: Sample $(\mathbf{x}_{\text{start}}, \mathbf{x}_{\text{goal}}, c_{\max})$

1 **if** $c_{\max} < \infty$ **then**
2 $\quad c_{\min} \leftarrow \left|\left|\mathbf{x}_{\text{goal}} - \mathbf{x}_{\text{start}}\right|\right|_2$;
3 $\quad \mathbf{x}_{\text{centre}} \leftarrow \left(\mathbf{x}_{\text{start}} + \mathbf{x}_{\text{goal}}\right)/2$;
4 $\quad \mathbf{C} \leftarrow$ RotationToWorldFrame $\left(\mathbf{x}_{\text{start}}, \mathbf{x}_{\text{goal}}\right)$;
5 $\quad r_1 \leftarrow c_{\max}/2$;
6 $\quad \{r_i\}_{i=2,\ldots,n} \leftarrow \left(\sqrt{c_{\max}^2 - c_{\min}^2}\right)/2$;
7 $\quad \mathbf{L} \leftarrow \text{diag}\{r_1, r_2, \ldots, r_n\}$;
8 $\quad \mathbf{x}_{\text{ball}} \leftarrow$ SampleUnitNBall;
9 $\quad \mathbf{x}_{\text{rand}} \leftarrow \left(\mathbf{C}\mathbf{L}\mathbf{x}_{\text{ball}} + \mathbf{x}_{\text{centre}}\right) \cap X$;
10 **else**
11 $\quad \mathbf{x}_{\text{rand}} \sim \mathcal{U}(X)$;
12 **return** $\mathbf{x}_{\text{rand}}$;

矩阵L的计算相对简单
这里的C_max为已经搜索出的最优路径大小
然后构造一个n维的对角矩阵，n就是xstart的维数

Quote: Gammell J D , Srinivasa S S , Barfoot T D . *Informed RRT*: Optimal Sampling-based Path Planning Focused via Direct Sampling of an Admissible Ellipsoidal Heuristic. 2014.*

Informed RRT*

深蓝学院 shenlanxueyuan.com

- 算法流程

Algorithm 2: Sample $(\mathbf{x}_{\text{start}}, \mathbf{x}_{\text{goal}}, c_{\max})$

1 **if** $c_{\max} < \infty$ **then**
2 $\quad c_{\min} \leftarrow \left|\left|\mathbf{x}_{\text{goal}} - \mathbf{x}_{\text{start}}\right|\right|_2$;
3 $\quad \mathbf{x}_{\text{centre}} \leftarrow \left(\mathbf{x}_{\text{start}} + \mathbf{x}_{\text{goal}}\right)/2$;
4 $\quad \mathbf{C} \leftarrow$ RotationToWorldFrame $\left(\mathbf{x}_{\text{start}}, \mathbf{x}_{\text{goal}}\right)$;
5 $\quad r_1 \leftarrow c_{\max}/2$;
6 $\quad \{r_i\}_{i=2,\ldots,n} \leftarrow \left(\sqrt{c_{\max}^2 - c_{\min}^2}\right)/2$;
7 $\quad \mathbf{L} \leftarrow \text{diag}\{r_1, r_2, \ldots, r_n\}$;
8 $\quad \mathbf{x}_{\text{ball}} \leftarrow$ SampleUnitNBall;
9 $\quad \mathbf{x}_{\text{rand}} \leftarrow \left(\mathbf{C}\mathbf{L}\mathbf{x}_{\text{ball}} + \mathbf{x}_{\text{centre}}\right) \cap X$;
10 **else**
11 $\quad \mathbf{x}_{\text{rand}} \sim \mathcal{U}(X)$;
12 **return** $\mathbf{x}_{\text{rand}}$;

单位球的采样：
考虑到配置空间只有3维，因此可以直接通过每个维度的均匀采样实现。分别对x,y,z在区别-1到1之间进行采样，然后判断采样点距离原点的距离是否小于半径1，如果大于则丢弃，重新采样

!但要注意此方法不适用于 N 太大的情况，因为几乎不可能采样到在单位球中的点。

Quote: Gammell J D , Srinivasa S S , Barfoot T D . *Informed RRT*: Optimal Sampling-based Path Planning Focused via Direct Sampling of an Admissible Ellipsoidal Heuristic. 2014.*

Informed RRT*

代码实现：初始化一些必要参数

一些固定的参数只需计算一次：椭圆焦点x_f1,x_f2，旋转矩阵C

```
void setEpllise(const Eigen::Vector3d& start, const Eigen::Vector3d& goal)
{
  x_f1 = start;
  x_f2 = goal;
  x_center = (x_f1+x_f2)/2;
  c_min = (x_f2-x_f1).norm();
  a1 = (x_f2-x_f1)/c_min;
  id_1t<<1,0,0;
  Eigen::JacobiSVD<Eigen::MatrixXd> svd(a1*id_1t.transpose(),Eigen::ComputeThinU|Eigen::ComputeThinV);
  Eigen::Matrix3d V = svd.matrixV();
  Eigen::Matrix3d U = svd.matrixU();
  Eigen::DiagonalMatrix<double,3> d;
  d.diagonal()<<1,1,V.determinant()*U.determinant();
  C = U*d*V.transpose();
}
```

Informed RRT*

深蓝学院 shenlanxueyuan.com

代码实现：单位球的采样

```
uniform_rand_2_ = std::uniform_real_distribution<double>(-1.0, 1.0);
```

```
bool sampleUnitNBall(Eigen::Vector3d& sample)
{
  sample[0] = uniform_rand_2_(gen_);
  sample[1] = uniform_rand_2_(gen_);
  sample[2] = uniform_rand_2_(gen_);
  if(sample.norm()<=1)
  {
    return true;
  }
  else
  {
    return false;
  }
}
```

1. 初始化一个-1到1之间的均匀分布生成器
2. 对各个维度进行均匀采样
3. 判断是否在单位球内

Informed RRT*

深蓝学院 shenlanxueyuan.com

代码实现：informed采样

```
void informedSampling(Eigen::Vector3d &sample, double c_max)
{
  Eigen::DiagonalMatrix<double,3> L;
  double r1 = c_max/2;
  double r2 = sqrt(c_max*c_max-c_min*c_min)/2;
  double r3 = r2;
  L.diagonal()<<r1,r2,r3;
  Eigen::Vector3d x_ball;
  while(!sampleUnitNBall(x_ball)){}
  sample = C*L*x_ball+x_center;

}
```

1. 旋转矩阵C已经在setEpllise函数中已经获取
2. 计算对角矩阵L
3. 单位球采样
4. 根据公式计算椭圆内直接采样

Informed RRT*

深蓝学院 shenlanxueyuan.com

代码实现：这里只需修改下RRT*中采样部分即可完成Informed RRT*算法

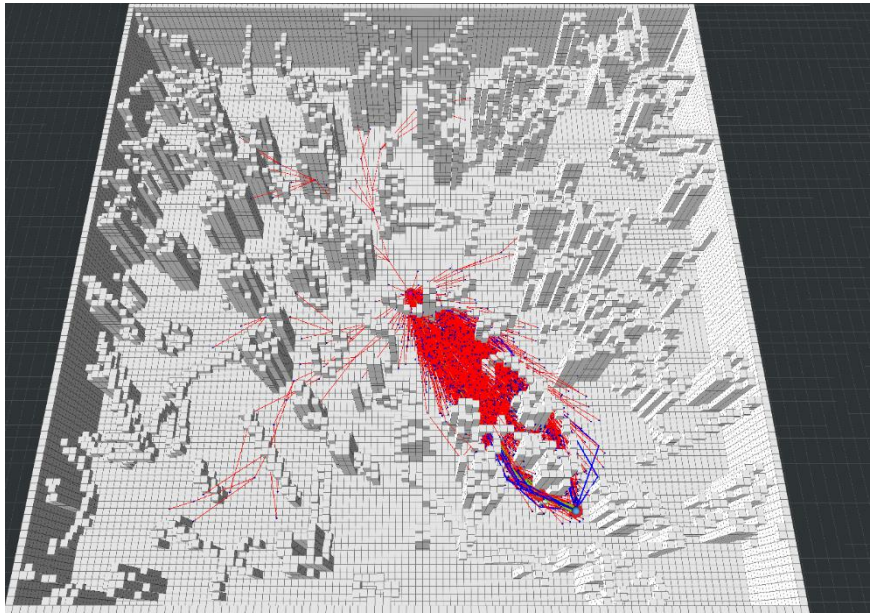
1. 首先在进入主循环前对informed采样的一些参数通过setEpllise()进行一次初始化。
2. 设置informed RRT*的采样条件：如果算法还没找到一个解，即变量goal_found为false时，继续采用随机采样，一旦找到解了即goal_found为true，则采用informed采样。

```
/* main loop */
int idx = 0;
double best_path_cost = 0;
sampler_.setEpllise(s,g);
for (idx = 0; (ros::Time::now() - rrt_start_time).toSec() < search_time_ && valid_tree_node_nums_ < max_tree_node_nums_; ++idx)
{
  /* biased random sampling */
  Eigen::Vector3d x_rand;
  if(!goal_found)
  {
    sampler_.samplingOnce(x_rand);
  }
  else
  {
  //这里添加informed RRT*的采样
    sampler_.informedSampling(x_rand,best_path_cost);
  }
```

Informed RRT*

深蓝学院 shenlanxueyuan.com

结果展示



```
goal rcved at 18 12  2
[ INFO] [1659061910.344977071]: [RRT*]: RRT starts planning a path
[ INFO] [1659061910.545792222]: [RRT*]: first path length: 26.3116, use_time: 0.000214585
[ INFO] [1659061910.546068914]: [RRT*] final path len: 22.4401
```

可以看出Informed RRT*的采样更具方向性，相比RRT*明显提高了路径优化效率，减少了不必要点的采样。

深蓝学院
shenlanxueyuan.com

感谢各位聆听！

Thanks for Listening