第三章作业分享

主讲人 Franklin

纲要

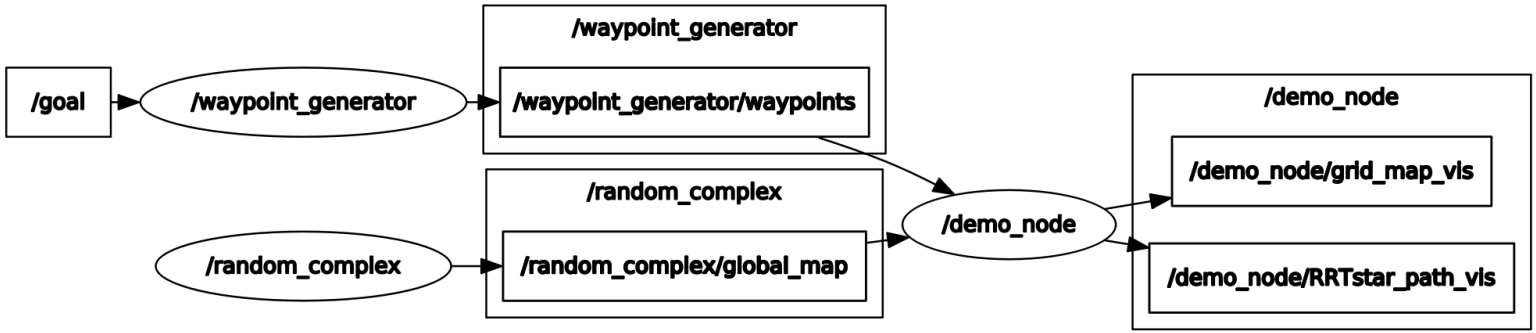
深蓝学院
shenlanxueyuan.com

- 第一部分：ROS作业
- 第二部分：MATLAB作业
- 第三部分：问题与解答

ROS作业

深蓝学院
shenlanxueyuan.com

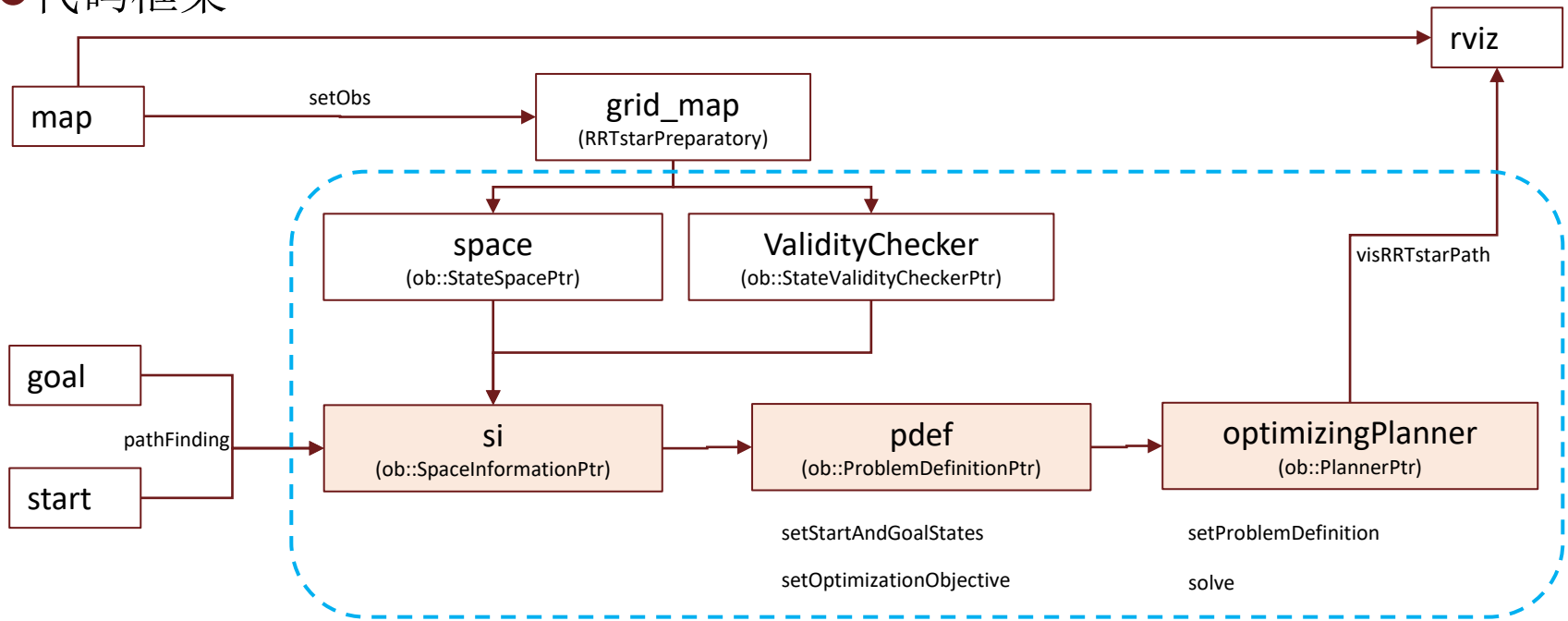
- 代码框架



ROS作业

深蓝学院 shenlanxueyuan.com

●代码框架



ROS作业

深蓝学院
shenlanxueyuan.com

●STEP 1

const ob::RealVectorStateSpace::StateType* state3

```
/** \brief The definition of a state in R<sup>n</sup> */
class StateType : public State
{
public:
    StateType() : State() {...}

    /**...*/
    double operator[](unsigned int i) const {...}

    /**...*/
    double& operator[](unsigned int i) {...}

    /** \brief The value of the actual vector in R<sup>n</sup> */
    double *values;
};
```

```
/**
    *
    *
    STEP 1: Extract the robot's (x,y,z) position from its state
    *
    *
    */
//    double x = state3D->as<ob::RealVectorStateSpace::StateType>()->values[0];
//    double y = state3D->as<ob::RealVectorStateSpace::StateType>()->values[1];
//    double z = state3D->as<ob::RealVectorStateSpace::StateType>()->values[2];
    double x = state3D->values[0];
    double y = state3D->values[1];
    double z = state3D->values[2];
```

ROS作业

深蓝学院 shenlanxueyuan.com

- STEP 2

ob::ScopedState<> start(space);

```
private:

    StateSpacePtr                   space_;
    StateSamplerPtr                 sampler_;
    StateType                      *state_;
```

```
/**
*
*
STEP 2: Finish the initialization of start state
*
*
*/
start()->as<ob::RealVectorStateSpace::StateType>()->values[0] = start_pt( index: 0);
start()->as<ob::RealVectorStateSpace::StateType>()->values[1] = start_pt( index: 1);
start()->as<ob::RealVectorStateSpace::StateType>()->values[2] = start_pt( index: 2);
```

```
/** \brief De-references to the contained state */
StateType& operator*() {...}

/** \brief De-references to the contained state */
const StateType& operator*() const {...}

/** \brief Returns a pointer to the contained state */
StateType* operator->() {...}

/** \brief Returns a pointer to the contained state */
const StateType* operator->() const {...}

/** \brief Returns a pointer to the contained state */
StateType* get() {...}

/** \brief Returns a pointer to the contained state */
const StateType* get() const {...}

/** \brief Returns a pointer to the contained state (used
StateType* operator()() const {...}
```

```
/** \brief Cast this instance to a desired type. */
<T>
T* as()
{
    /** \brief Make sure the type we are allocating is indeed a state */
    BOOST_CONCEPT_ASSERT( ModelInParens: (boost::Convertible<T*, State*>));

    return static_cast<T*>(this);
}
```

ROS作业

深蓝学院 shenlanxueyuan.com

- STEP 3

ob::ScopedState<> goal(space);

```
private:

    StateSpacePtr                   space_;
    StateSamplerPtr                 sampler_;
    StateType                      *state_;
```

```
/**
*
*
STEP 3: Finish the initialization of goal state
*
*
*/
goal->as<ob::RealVectorStateSpace::StateType>()->values[0] = target_pt( index: 0);
goal->as<ob::RealVectorStateSpace::StateType>()->values[1] = target_pt( index: 1);
goal->as<ob::RealVectorStateSpace::StateType>()->values[2] = target_pt( index: 2);
```

```
/** \brief De-references to the contained state */
StateType& operator*() {...}

/** \brief De-references to the contained state */
const StateType& operator*() const {...}

/** \brief Returns a pointer to the contained state */
StateType* operator->() {...}

/** \brief Returns a pointer to the contained state */
const StateType* operator->() const {...}

/** \brief Returns a pointer to the contained state */
StateType* get() {...}

/** \brief Returns a pointer to the contained state */
const StateType* get() const {...}

/** \brief Returns a pointer to the contained state (used
StateType* operator()() const {...}
```

```
/** \brief Cast this instance to a desired type. */
<T>
T* as()
{
    /** \brief Make sure the type we are allocating is indeed a state */
    BOOST_CONCEPT_ASSERT( ModelInParens: (boost::Convertible<T*, State*>));

    return static_cast<T*>(this);
}
```

ROS作业

深蓝学院
shenlanxueyuan.com

- STEP 6

```
// Construct our optimizing planner using the RRTstar algorithm.
/**
 *
 *
 STEP 6: Construct our optimizing planner using the RRTstar algorithm,
 please define varible as optimizingPlanner
 *
 *
 */
 ob::PlannerPtr optimizingPlanner( p: new og::RRTstar(si));
//  ob::PlannerPtr optimizingPlanner(new og::RRT(si));
//  ob::PlannerPtr optimizingPlanner(new og::InformedRRTstar(si));
//  ob::PlannerPtr optimizingPlanner(new og::LazyRRT(si));
```

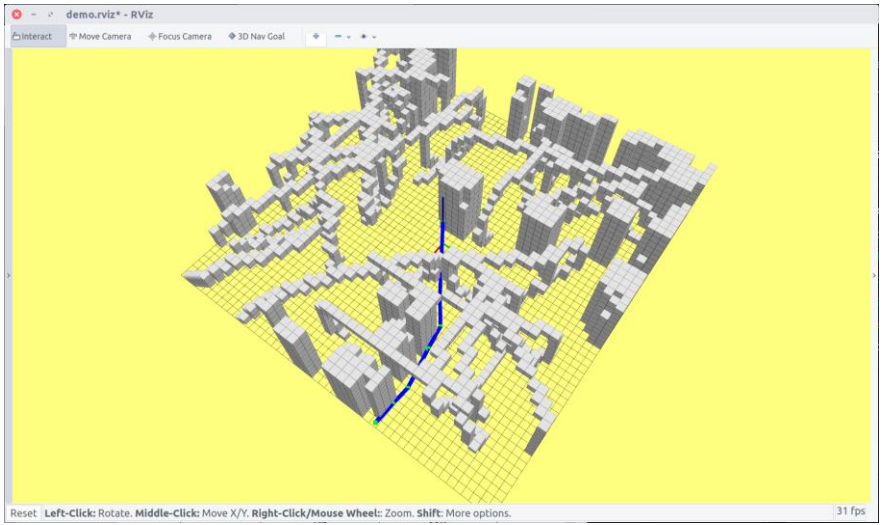
```
// attempt to solve the planning problem within one second of
// planning time
ob::PlannerStatus solved = optimizingPlanner->solve( solveTime: 0.2);
```

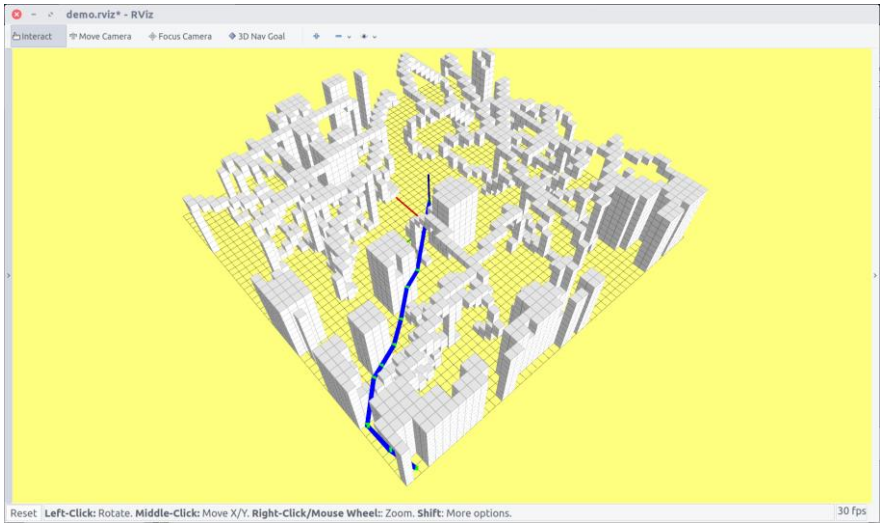
Franklin

ROS作业

深蓝学院
shenlanxueyuan.com

- 效果展示





ROS作业

深蓝学院 shenlanxueyuan.com

●效果对比

200ms对比	Created new states	Final solution cost
RRT*	6922	6.540
	6927	6.405
	7306	5.799
	7795	5.533
avg	7237	6.069
Informed RRT*	5462	5.499
	5298	4.448
	4938	5.518
	5315	5.552
avg	5253	5.254

	Created new states	Using time
RRT	30	1.30
	35	0.73
	41	0.79
	27	0.59
avg	33	0.85
Lazy RRT	144	5.47
	172	5.72
	178	2.82
	259	3.44
avg	188	4.35

纲要

深蓝学院
shenlanxueyuan.com

- 第一部分：ROS作业
- **第二部分：MATLAB作业**
- 第三部分：问题与解答

MATLAB作业

深蓝学院
shenlanxueyuan.com

●STEP1-2

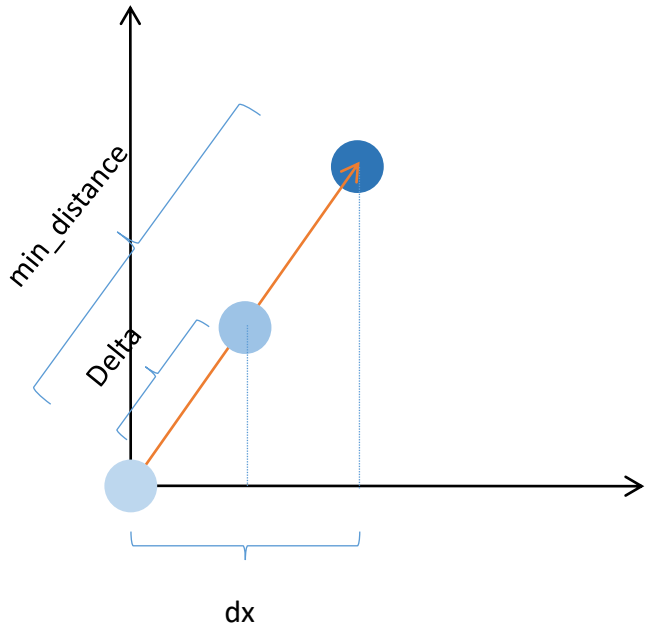
```
%Step 1: 在地图中随机采样一个点x_rand
%提示：用（x_rand(1),x_rand(2)）表示环境中采样点的坐标
x_rand(1) = rand(1)*xL;
x_rand(2) = rand(1)*yL;

x_near=[];
%Step 2: 遍历树，从树中找到最近邻近点x_near
%提示：x_near已经在树T里
  node_distance = [];
for i = 1:length(T.v)
    temp_node(1) = T.v(i).x;
    temp_node(2) = T.v(i).y;
    node_distance(i) = sqrt((temp_node(1) - x_rand(1))^2 + (temp_node(2) - x_rand(2))^2);
end

nearestNodeID = find(node_distance == min(node_distance));
min_distance = min(node_distance);
x_near(1) = T.v(nearestNodeID).x;
x_near(2) = T.v(nearestNodeID).y;
```

MATLAB作业

深蓝学院 shenlanxueyuan.com

●STEP3-4

```
x_new=[];
%Step 3: 扩展得到x_new节点
%提示：注意使用扩展步长Delta
dx = x_rand(1) - x_near(1);
dy = x_rand(2) - x_near(2);
x_new(1) = x_near(1) + (dx / min_distance) * Delta;
x_new(2) = x_near(2) + (dy / min_distance) * Delta;


%检查节点是否是collision-free
if ~collisionChecking(x_near,x_new,Imp)
    continue;
end
count=count+1;

%Step 4: 将x_new插入树T
%提示：新节点x_new的父节点是x_near
tlength = length(T.v);

T.v(tlength+1).x = x_new(1);
T.v(tlength+1).y = x_new(2);
T.v(tlength+1).xPrev = x_near(1);
T.v(tlength+1).yPrev = x_near(2);
T.v(tlength+1).dist=Delta;
T.v(tlength+1).indPrev = nearestNodeID;
```



MATLAB作业

深蓝学院 shenlanxueyuan.com

●STEP5-6

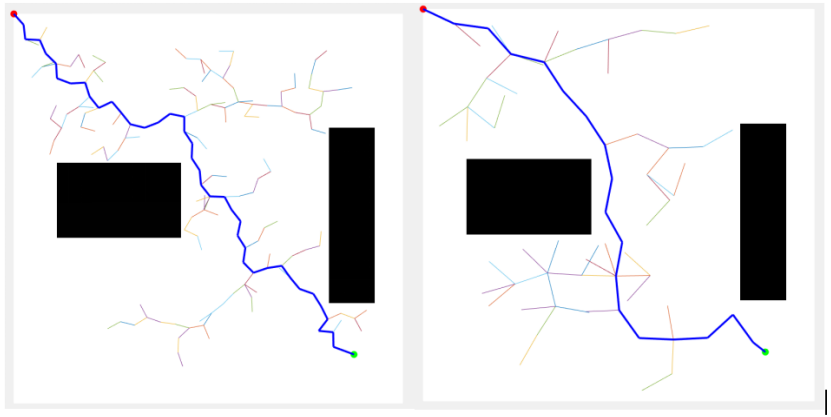
```
%Step 5:检查是否到达目标点附近
%提示：注意使用目标点阈值Thr，若当前节点和终点的欧式距离小于Thr，则跳出当前for循环
dist_to_goal = sqrt((x_new(1) - x_G)^2 + (x_new(2) - y_G)^2);
goal = [];
goal(1) = x_G;
goal(2) = y_G;
if dist_to_goal < Thr
    if collisionChecking(goal,x_new,Imp)
        bFind = true;
        break;
    end

end

%Step 6:将x_near和x_new之间的路径画出来
%提示 1：使用plot绘制，因为要多次在同一张图上绘制线段，所以每次使用plot后需要接上hold on命令
%提示 2：在判断终点条件弹出for循环前，记得把x_near和x_new之间的路径画出来

figure(1);
plot([x_near(1) x_new(1)], [x_near(2) x_new(2)]);
hold on;

pause(0.01); %暂停一会，使得RRT扩展过程容易观察
```

碰撞检测

MATLAB作业

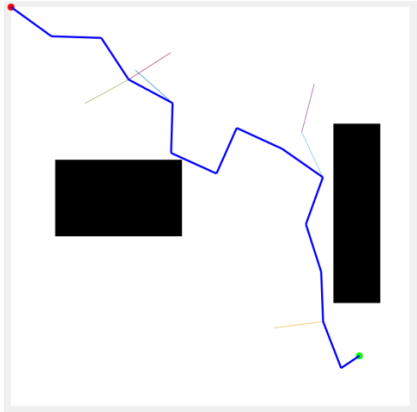
深蓝学院
shenlanxueyuan.com

- 效果对比

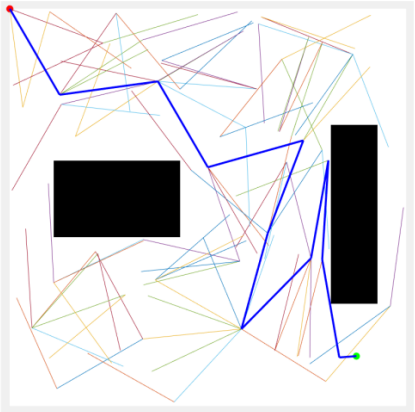


Delta = 30

Delta = 70



Delta = 100



Delta = 200

在线问答

深蓝学院
shenlanxueyuan.com

Q&A

Franklin

深蓝学院
shenlanxueyuan.com

感谢各位聆听！

Thanks for Listening