深蓝学院
shenlanxueyuan.com

激光SLAM理论与实践第七章作业分享

主讲人 李聪平

目录

深蓝学院
shenlanxueyuan.com

1.1 覆盖栅格建图方法

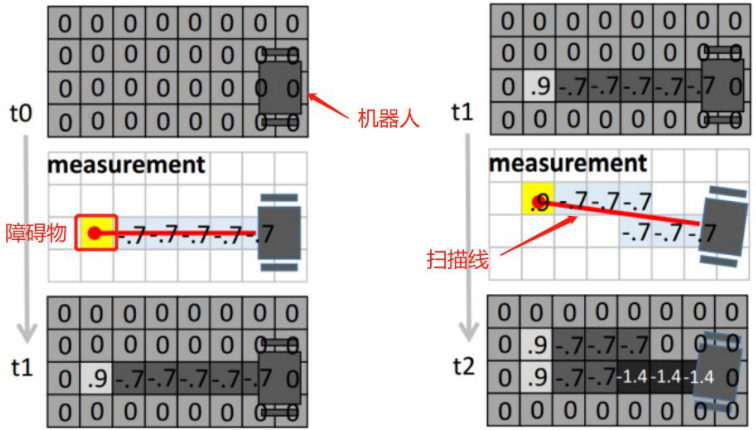
深蓝学院 shenlanxueyuan.com

- 栅格值符合加法：

$$S^{+} = S^{-} + lofree$$

或

$$S^{+} = S^{-} + looccu$$

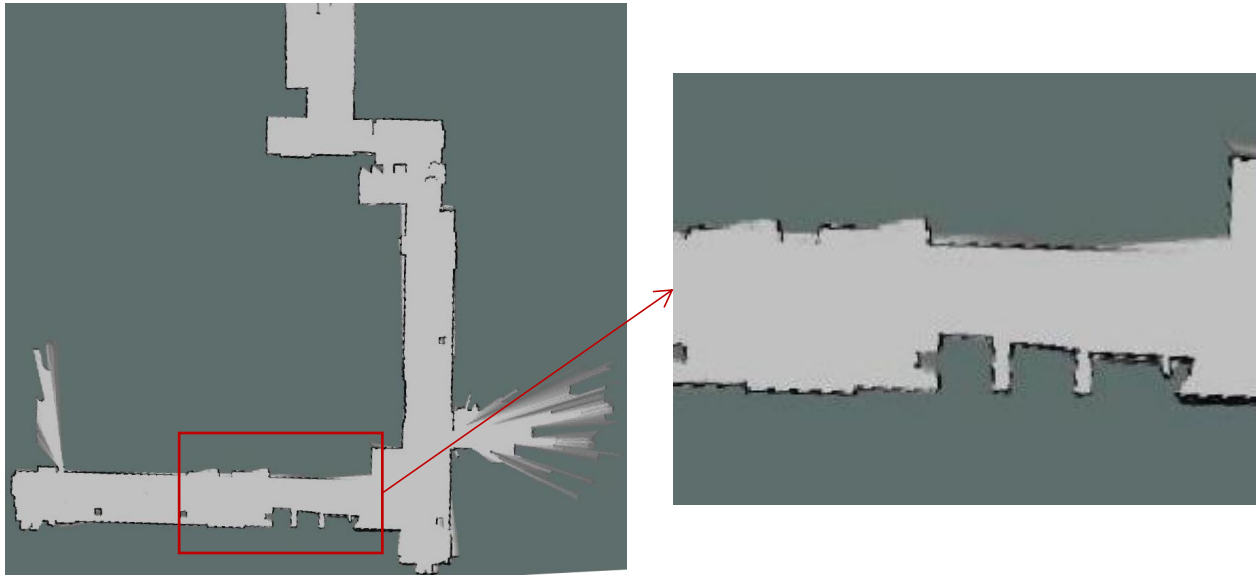



```
//世界坐标转栅格坐标
GridIndex M_index = ConvertWorld2GridIndex(world_x,world_y);
//查找激光束上free点
std::vector<GridIndex> free_index = TraceLine(robotIndex.x,
                                    robotIndex.y,M_index.x,M_index.y);
//遍历一条线的每一个free点，进行更新点值
for(int k= 0;k<free_index.size();k++)
{
    //二维栅格转一维，序号化
    int line_index = GridIndexToLinearIndex(free_index[k]);
    // pMap[line_index] 是unsinged ，不能直接减
    int value  = pMap[line_index];
    value += mapParams.log_free;        //无障碍 减一
    if( value < 0) pMap[line_index] = 0;//栅格值最小为0
    else pMap[line_index] =value;
}
//更新被占用点
    int occ_index = GridIndexToLinearIndex( M_index);
    pMap[occ_index] += mapParams.log_occ;
    //栅格值最大为100
    if(pMap[occ_index] > 100) pMap[occ_index] = 100;
```

1.1 覆盖栅格建图方法

深蓝学院
shenlanxueyuan.com

- 实验效果：
- 边界线条不均匀，有的深色有的模糊



1.2 计数建图方法

深蓝学院 shenlanxueyuan.com

- 栅格值： $m_j = \frac{a_j}{a_j+b_j} * 100$

  其中，a_j表示$hits(j)$，

  b_j表示$misses(j)$

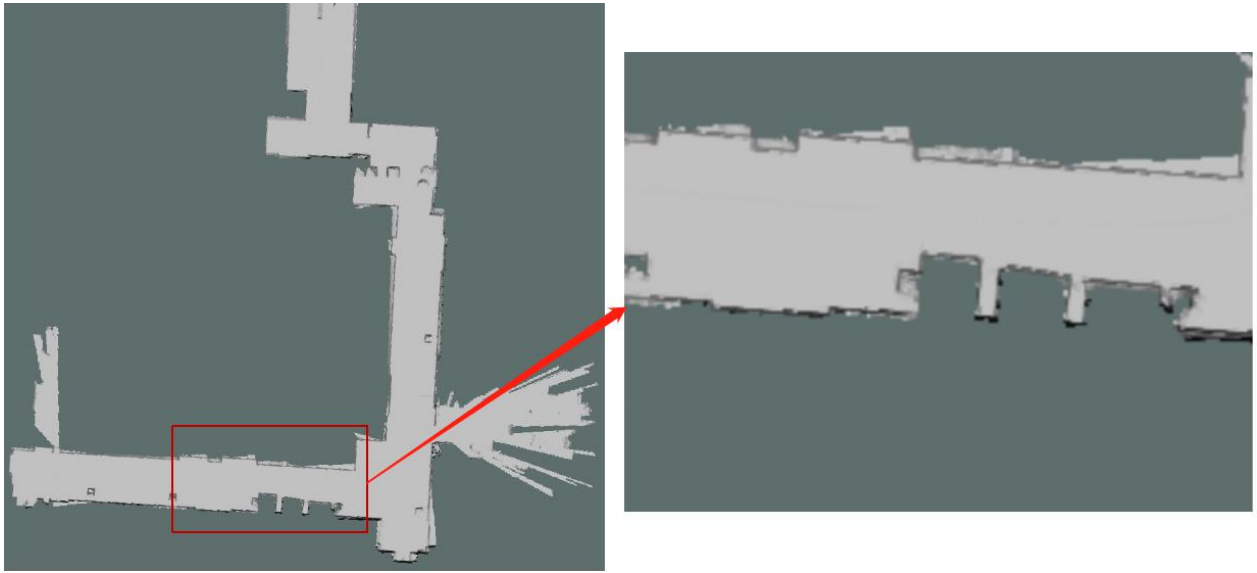
```
//世界坐标转栅格坐标
GridIndex M_index = ConvertWorld2GridIndex(world_x,world_y);
//查找激光束上free点
std::vector<GridIndex> free_index = TraceLine(robotIndex.x,
                                    robotIndex.y,M_index.x,M_index.y);
for(int j= 0;j<free_index.size();j++)//遍历一条线的每一个空点
{
    //二维转一维
    GridIndex tmp_free =free_index[j];
    int line_index1 = GridIndexToLinearIndex(tmp_free );
     ++pMapMisses[line_index1];
} //更新被占用点
    int occ_index1 = GridIndexToLinearIndex( M_index);
    ++pMapHits[occ_index1];
```

```
for(int i = 0 ;i< mapParams.height*mapParams.width;i++)
    if(pMapHits[i] || pMapMisses[i])  {
        int m = 100 * pMapHits[i] / (pMapMisses[i]+pMapHits[i]);
    if (m > 35) pMap[i] = 100 ;
    else pMap[i] = m;
    }
```

1.2 计数建图方法

深蓝学院
shenlanxueyuan.com

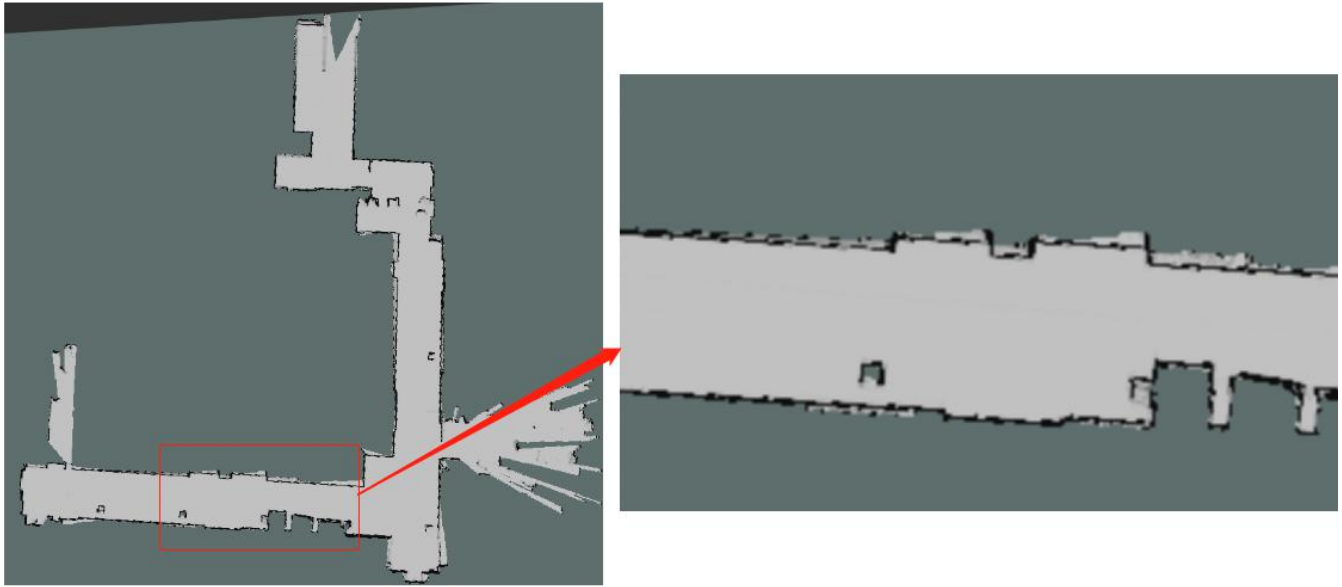
- 实验效果（未加入大于35%比例约束）：
- 边界模糊



1.2 计数建图方法

深蓝学院
shenlanxueyuan.com

- 实验效果（加入大于35%比例约束）：
- 轮廓相对清晰，但边界依线条然粗细不均匀和存在一些噪点



1.3 TSDF建图方法

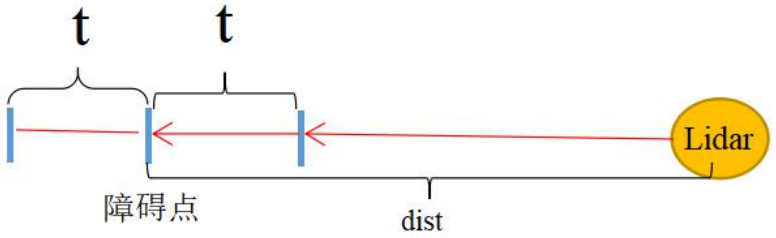
深蓝学院 shenlanxueyuan.com

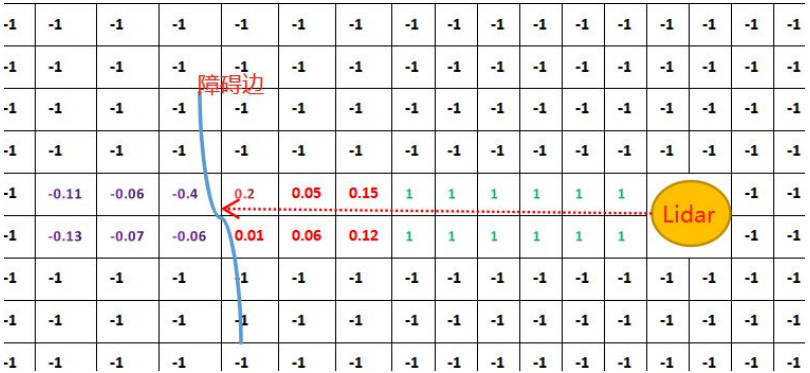
- 1、对每一个激光束建立TDSF函数
- 2、栅格分辨率为0.05，令截断距离t=0.15，

则计算单激光束附近六个TSDF栅格值

- 3、多次扫描加权计算每个栅格的tsdf值
- 4、查找发生符号变化的两个值

  并通过线性插值求得tsdf值为0的栅格
- 5、令该栅格的值为100





1.3 TSDF建图方法

深蓝学院 shenlanxueyuan.com

- 将栅格坐标转世界坐标
  - $sdf(x)$的定义：

$$sdf_i(x) = laser_i(x) - dist_i(x)$$

$laser_i(x)$表示激光测量距离

$dist_i(x)$表示栅格离传感器原点的距离

- tsdf栅格值计算

$$tsdf_i(\mathbf{x}) = \max(-1, \min(1, \frac{sdf_i(\mathbf{x})}{t}))$$

```
double TSDF,tsdf,x,y,delta_x,delta_y,b;
for(int i = 0;i< free_index.size();i++)
{  //栅格点转世界坐标
   x= (free_index[i].x-mapParams.offset_x)*mapParams.resolution +mapParams.origin_x;
   y= (free_index[i].y-mapParams.offset_y)*mapParams.resolution +mapParams.origin_y;
   //计算栅格点点到机器人的实际距离
   delta_x =x-robotPose(0);
   delta_y =y-robotPose(1);
   b = std::pow(std::pow( delta_x,2)+std::pow( delta_y,2),0.5);
   //sdfi(x)
    TSDF = std::max(-1.0,std::min(1.0,(dist-b)/t));
    //求出二维栅格坐标转一维序号
    int pose_num = GridIndexToLinearIndex(free_index[i]);
    //TSDFi(x)
    pMapTSDF[pose_num] = (pMapW[pose_num]*pMapTSDF[pose_num]+TSDF)/(pMapW[pose_num]+1);
    //Wi(x)
    ++pMapW[pose_num];
}
```

多次观测的融合更新方法：

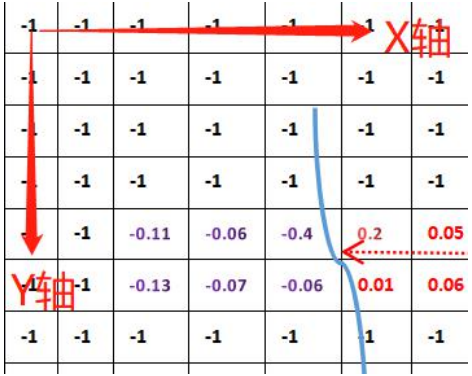
$$TSDF_i(\mathbf{x}) = \frac{W_{i-1}(\mathbf{x})TSDF_{i-1}(\mathbf{x}) + w_i(\mathbf{x})tsdf_i(\mathbf{x})}{W_{i-1}(\mathbf{x}) + w_i(\mathbf{x})}$$

$$W_i(\mathbf{x}) = W_{i-1}(\mathbf{x}) + w_i(\mathbf{x})$$

1.3 TSDF建图方法

深蓝学院 shenlanxueyuan.com

- 同时对X和Y轴两个方向进行寻找相邻异常值



插值思路：

A（x=12.4）　　X（？）　　B（x=11.6）

-----|----------|----------|----------→x 轴

a（-0.08）　　0　　b（0.07）

可得

$$\frac{A-X}{a}=\frac{X-B}{b}$$

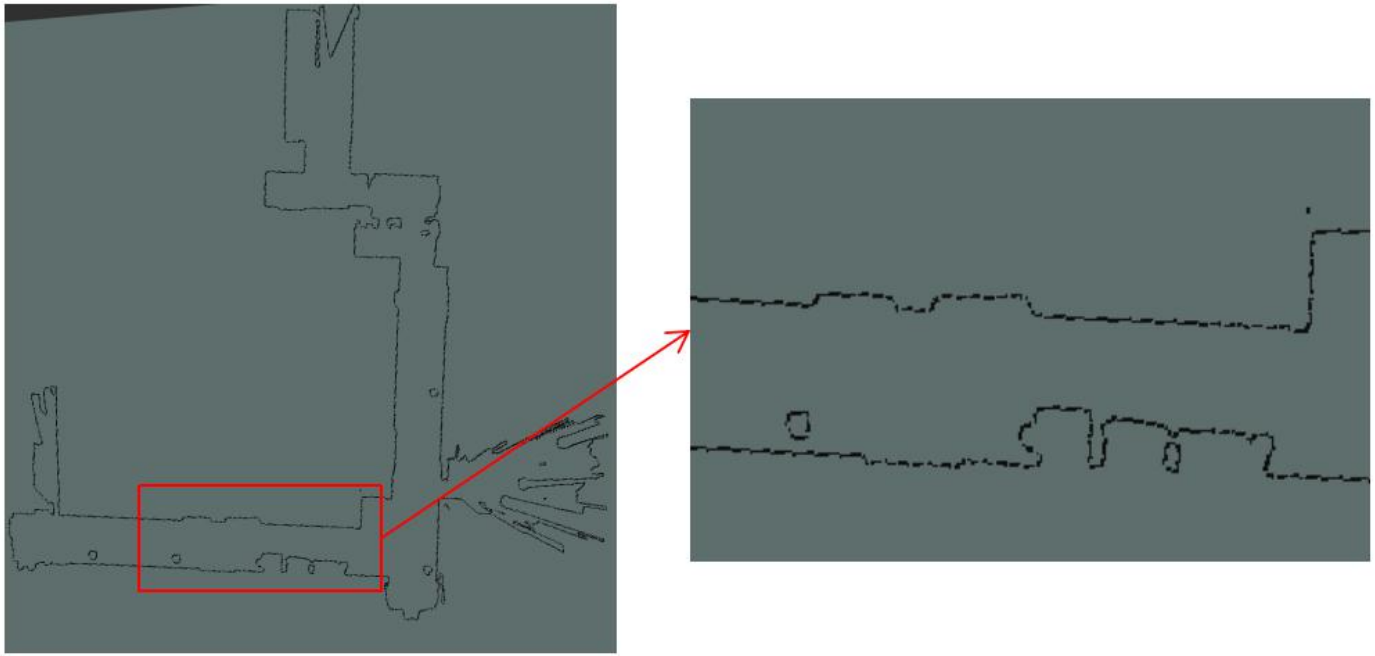
化简得　　X = （A * b - B * a）/（b - a）

```
for(int i= 0;i<mapParams.height-2;i++)  //设i 为y轴
{
    for(int j = 0;j<mapParams.width-2;j++)  //设j为x轴
    {
        int line_value = j + i*mapParams.height;
        int line_x = line_value +1;//往x轴移动一格
        int line_y = line_value +mapParams.height;//往y轴方向移动一格
        //转世界坐标
        double A_x = GridIndex2ConvertWorld(j);
        double A_y = GridIndex2ConvertWorld(i);
        double B_x = GridIndex2ConvertWorld(j+1);
        double B_y = GridIndex2ConvertWorld(i+1);
        double a,b,b1,x,y;
        a = pMapTSDF[line_value];  b= pMapTSDF[line_x];  b1 = pMapTSDF[line_y];
        if( a*b < 0){ //x方向
                x = A_x;
                y = interpolation(A_y,B_y,a,b);//插值
            }
        else if( a*b1 < 0){ //y方向
                x = interpolation(A_x,B_x,a,b1);
                y = A_y;
            }
                pMap[GridIndexToLinearIndex(ConvertWorld2GridIndex(x,y))] = 100;
    }
}
```

```
double interpolation(double A,double B,double a,double b)
{
    double value = (b*A-a*B) / (b-a);
    return value = a==b? A : value;
}
```

1.3 TSDF建图方法

- 实验效果：边界轮廓清晰，厚度一致，无噪点，但运算量大费时长



2、三种建图方法比较

深蓝学院 shenlanxueyuan.com

特点\建图方法	覆盖栅格	计数栅格	TSDF
原理	直接加法运算	hits 和 Miss 加法计算，计算比值	通过加权线性最小二乘，对障碍点进行多帧融合
效率	每次扫描加法运算，实时性最高	时效性高	程序复杂计算量大，全部地图最后生成，不能实时
建图效果	都容易受噪声影响，有噪点，边界粗细出现不均匀		对噪声不敏感，地图边界清晰

在线问答

深蓝学院
shenlanxueyuan.com

Q&A

深蓝学院
shenlanxueyuan.com

感谢各位聆听！

Thanks for Listening