LAST WEEK

主要读了GSW的一些论文。

分类

1. classfication of MLS point clouds from height feature

point-based 的分类

主要用到了三个特征：两个高度成分、一个反射强度

特征描述：高度差异（圆柱内hmax-hmin）、圆柱内点的个数、反射强度

分类方法：LIBSVM（软件）[具体的方法可以使支持向量、随机森林、传统的神经网络]

只是可行性测试，效果并不是特别理想。

2.a two-step classfication approach to distinguishing similar objects in MLS point clouds

Feature-based的分类

其实研究的主要还是交通标志、杆状物体的分类。

主要分为两步：第一步现将交通标志提取。第二歩在将其细分为路灯、标志牌等。

本文研究识别相似目标，是基于分割的分类，用到了局部特征point feature histogram（bag of features）

这个bag of features见到很多回了，但是具体情况还不清楚，准备看（visual categorization with bags of keypoints）

流程：去除地面及建筑立面的点->剩余点被分组（手动的）->建立局部特征feature histogram->建立高维的特征bag-of-feature->两步识别

具体：

1.去除地面及建筑立面的点：TIN densification算法

2.分割：本文手动，但是实际可以通过算法（通过约束点间最大距离，一个segment中的最小个数）

3.建立特征：提取，每个点都建立了特征->将局部点特征与segment合并行程 bag of features 编码：建立vocabulary用k-means clustring算法->特征向量

4.先粗后细的分类

剩余点的分组是手动的，感觉效率不高得改进，但是用算法分组的准确性可能带来很大影响。

每个点都建立特征会不会很费时

Related:物体的识别主要依赖于几何特征：又包括全局特征（高度、大小等），局部特征（关键点的特征）。

1. using multi-scale features for 3D semantic labelling of ALS data

point-based feature

提供了一个关于特征描述的方法。

本文创新点：1.用了不同尺度、不同类型的邻域2.用到了normalized height feature

related往常尺度的选择有两类方法：dimensionality-based 和 eigenentropy-based

具体实现：

1.邻域的定义：4个圆柱体邻域（垂直分布、不同半径、竖直方向无线）、5个球体邻域（已解决柱体方向可能属于不同物体，四个规定半径、一个k邻域）、二维邻域（只用高度）

2.特征提取：covariance feature（特征向量）、geometric 3D properties（点闽都、垂直性、最大偏差）、shape distribution（shape value的histogram[随机选点，计算特性]）、normalized height feature（根据地势）

共产生62个固定半径+1个 eigenentropy-based+1个height

3.feature normalization:线性映射到[0,1]

这个看起来描述特征描述的很仔细。但是会不会过多？

展望就是从point-based到object-based

4.a curvature-based adaptive neighborhood for individual poind clouds classification

提供一个获得理想邻域的方法

具体实现：

1.curvature：某点在一定半径范围内的协方差矩阵，计算出特征值，设置参数C.C越大代表植被等不规则区域，需要比较小的邻域；反之则代表比较规则的区域，需要比较大的区域。

计算一个阈值cur，将分散和规则区域分开单独处理。分散区域用k-based方法；规则区域用r-based方法

2.提取特征：只用到了集合特征（verticality，最大高度差，特征向量）

3.分类：随机森林

关于汽车的识别

Vehicle recognition in ALS point clouds based on dynamic time wrapping

主要分为两步：1.分割：用汽车大小做约束进行区域增长2.识别：根据用DTW动态时间规划，基于profile shape进行识别。

创新点：1.被遮挡的可以被识别2.定位、定向、大小都可以被确定

实现要点：1.分割出尽可能多的segment2.判断segment是否为车

具体过程：

1.提取：按长宽高进行约束size limitation；多辆车的情况，根据水平间隔设置阈值

2.轮廓提取：segment中间顶部曲线的点。高度、到（起始点）的距离

3.比较相似性：DTW,不同长度同一类型的车的比较

第一步做不好会遗漏很多啊,对准确率影响很大

第二歩如果提取不完整或者多提取了附属物的点的话，中线上点的提取不就不准确了吗

此方法是否能运用到MLS的点云处理，如果是MLS的点云的话，只有汽车的一侧，提取不倒中线上的点了

related关于汽车的recognition两种方法：1.完整的汽车检测：遮挡或停靠的过近都有可能影像2.开宇汽车的attributes：就是相当于classification

分割

Fast edge detection and segmentation of TLS through normal variation analysis

基于detection和 region growing

主要步骤：计算projected incidence angle->进行normal variation analysis->轮廓边缘、面间交线被提取->region growing

创新点：避免计算每个点的法向

本文得益于grid data structure

具体实现：

通过grid结构寻找八邻域，根据余弦计算入射角度->进行法向变化的分析：八邻域八条边计算法相梯度，最大的与阈值进行比较判断其是否为边。

对于MLS扫描的数据结构、过程还不是很清楚，需要师兄讲一下

这个算法看上去对平面的分割很有效，但不清楚他对复杂的景观以及具有圆滑变化的车这种物体效果如何

Related 分割方法：1.point cloud-based两种：区域生长（法向统计、PCA特征、考虑点密度、表面粗糙程度、去了吧等等）；cluster（几何特性：feature vector）

2.image-based（投到二维平面再回到3D），优势在于有大量成熟算法，且平面处理效率快

去除地面点

CSF based non-ground points extraction from LiDAR data

只是提供了一个区域生长种子点选取的方法

区域增长的两个问题：1.种子点的选择2.生长的约束条件

传统步骤就是计算发现、曲率，选取曲率最小的点作为种子点，本文直接选取顶端。

就是CSF的反向过程

其他问题

如果加入高度描述特征，不就排除了一部分关键点（比如树什么的）

另外可不可以将每个segment 中的关键点也形成一个集合 就是segment与segment进行match（其实还是基于分割）

Matching过程用不用体现汽车的定位还是只提取就好了

THIS WEEK

1. 研究代码
2. 把不懂的问题在弄明白