# CloudCompare的简单的使用说明

#### **File**

open: 打开save: 保存

• Global Shift settings: 设置最大绝对坐标, 最大实体对角线

• Primitive Factory: 对点云进行原始加工, 改变原始点云的形状

• 3D mouse: 对3D鼠标 (如3Dconnexion) 的支持

• Close all: 关闭所有打开的实体

• Quit: 退出

#### Edit:

• Clone: 克隆选中的点云

- Merge: 合并两个或者多个实体。可以合并点云(原始云会被删除);可以合并网格(原始网不会修改,CC会创建一个新的网格结构)
- Subsample: 采集原始点云的子样本,可以用随机、立体、基于八叉树的方式采集,子样本会保持原始点云的标量、颜色、法线等性质。
- Apply Transformation: 可以对选中的实体做变换(4\*4矩阵、轴线角,欧拉角)
- Multiply / Scale: 让选中实体的坐标倍增。
- Translate / Rotate (Interactive Transformation Tool): 可以相对于另外一个实体或者坐标系移动选中的实体
- Segment (Interactive Segmentation Tool): 通过画2D多边形分隔选中的实体
- Crop: 分割一个或多个在3D-Box里面的点云。
- Edit global shift and scale: 进行全局变换和和比例缩放。
- Toggle (recursive): 用于控制键盘的快捷键。
- Delete: 删除选中的实体。
- Colors > Set Unique: 为所选实体设置唯一一个的颜色
- Colors > Colorize: 为所选实体着色,具体表现为分别用所选颜色乘以当前颜色的RGB而得到新的颜色
- Colors > Levels: 通过调整颜色的柱形图变色, 类似于Photoshop的Levels方法
- Colors > Height Ramp: 为所选实体设置颜色渐变(线形、梯形、环形)
- Colors > Convert to Scalar Field: 将当前的 RGB 颜色字段转换为一个或几个标量字段
- Colors > Interpolate from another entity: 在所选实体中插入另外一个实体的颜色
- Colors > Clear: 移除所选实体的颜色域
- Normals > Compute: 计算所选实体的法线
- Normals > Invert: 反转所选实体的法线
- Normals > Orient Normals > With Minimum Spanning Tree: 用同样的方法重新定位点云的全部法线(最小生成树)
- Normals > Orient Normals > With Fast Marching: 用同样的方法重新定位点云的全部法线(快速行进法)
- Normals > Convert to > HSV: 将云的法线转换到 HSV 颜色字段
- Normals > Convert to > Dip and Dip direction SFs: 转换点云的法线到两个标量域
- Normals > Clear: 为选定的实体移除法线
- Octree > Compute: 强制计算给定实体的八叉树
- Octree > Resample: 通过代替每个八叉树单元内的所有点来重新取样
- Mesh > Delaunay 2.5D (XY plane): 计算点云在xy平面上的2.5D三角剖分(Delaunay 2.5D triangulation, 德洛内2.5D三角算法)
- Mesh > Delaunay 2.5D (best fit plane): 计算点云在最佳平面的2.5D三角剖分(Delaunay 2.5D triangulation, 德洛内2.5D三角

#### 算法)

- Mesh > Convert texture/material to RGB: 将选定网格的网格材料和纹理信息转换为逐个点的 RGB 字段
- Mesh > Sample points: 在一个网格中随机取样
- Mesh > Smooth (Laplacian): 平滑一个网格(Laplacian smoothing, 拉普拉斯平滑算法)
- Mesh > Subdivide: 细分网格, 此算法递归细分网格三角形, 直到他们的表面细分到用户指定值之下。
- Mesh > Measure surface: 测量网格的总体表面积和每个三角形的平均表面积, 在控制台输出
- Mesh > Measure volume: 测量闭合网格的体积, 在控制台输出
- Mesh > Flag vertices: 检查网格的基本特性, 为每个网格样本做标志: 0 = normal, 1 = border, 2 = non-manifold
- Mesh > Scalar field > Smooth: 平滑网格顶点相关联的标量场。此方法与高斯滤波(Gaussian Filter)相反。运用qPCV插件后, 此方法特别有用
- Mesh > Scalar field > Enhance: 增强与网格顶点相关联的标量场。运用qPCV插件后, 此方法特别有用
- Sensors > Edit: 修改指定传感器内外在参数
- Sensors > Ground Based Lidar > Create: 创建'Ground Based Lidar' (= TLS)传感器实体,附加到所选的点云
- Sensors > Ground Based Lidar > Show Depth Buffer: 显示选中的Ground Based Lidar的深度
- Sensors > Ground Based Lidar > Export Depth Buffer:以ASCII文件的形式导出选中的Ground Based Lidar传感器的深度图
- Sensors > Camera Sensor > Create: 创建影像传感器
- Sensors > Camera Sensor > Project uncertainty:输出影像模块不确定的点云,输出不确定的x、y、z、3D信息
- Sensors > Camera Sensor > Compute points visibility (with octree): 统计选中影像传感器选中的点云。0=NOT VISIBLE, 1=VISIBLE
- Sensors > View from sensor: 更改当前的 3D 视图影像设置以匹配选定的传感器的设置 (用泡沫视图模式)
- Sensors > Compute ranges: 计算全部点(对于任何点云)相对于指定传感器的范围
- Sensors > Compute scattering angles: 计算全部点(对于任何有法线的云)相对于选中传感器分散的角度
- Scalar fields > Show histogram: 对当前选中的实体显示有效标量域的柱形图
- Scalar fields > Compute statistical parameters: 计算统计分布(高斯分布、威布尔分布)
- Scalar fields > Gradient: 计算标量域的梯度
- Scalar fields > Gaussian filter: 通过应用一个立体高斯滤镜, 平滑一个标量域
- Scalar fields > Bilateral filter: 用双边滤镜平滑一个标量域
- Scalar fields > Filter by Value: 用标量值筛选选定的云
- Scalar fields > Convert to RGB: 将有效的标量场转化为RGB颜色域
- Scalar fields > Convert to random RGB: 将有效的标量场转化为随机的RGB颜色域
- Scalar fields > Rename: 对选中实体重命名有效的标量域
- Scalar fields > Add constant SF: 用一个常数添加一个标量域
- Scalar fields > Add point indexes as SF: 用点索引的方式为所选点云创建一个新的标量域
- Scalar fields > Export coordinate(s) to SF(s): 导出坐标到标量域
- Scalar fields > Set SF as coordinate(s): 为选中的点云设置标量域的坐标
- Scalar fields > Arithmetic: 可以对在同一个点云的两个标量域进行标准运算(+, -, \*, /), 或者对单个标量域进行函数运算
- Scalar fields > Color Scales Manager: 色阶管理, 可以管理和创建新色域
- Scalar fields > Delete: 对选中的实体删除有效的标量域
- Scalar fields > Delete all (!): 对选中的实体删除全部的有效标量域

# Tools:

- Level: 可以选择三个点确定一个平面来操作
- Point picking:可以选择一个、两个、三个点来得到各种信息,如点的坐标、RGB、标量值、距离、角度等信息(尤其是两点间的距离)

- Point list picking:可以选择多个点创建一个点列表,可以输出为一个文件、一个新点云、一个折线
- Clean > Noise filter: 类似于qPCL插件的S.O.R.滤镜, 但又更多功能
- Projection > Unroll: 展开圆柱或圆锥体的点云成一个平面
- Projection > Rasterize: 栅格化点云(转化为2.5D网格),然后可以导出为一个新点云或者一个光栅图像
- Projection > Contour plot to mesh: 可以把一组折线转化为网格, 输出边缘轮廓线
- Projection > Export coordinate(s) to SF(s): 导出坐标到标量域
- Registration > Match bounding-box centers: 调整所有选中的实体, 让它们的中心在一个地方
- Registration > Match scales: 匹配所有选中实体的规模
- Registration > Align (point pairs picking): 在两个实体中挑选至少三个对应的点来对齐两个实体
- Registration > Fine registration (ICP): 自动精确地融合两个实体。前提是: ①两个云大体上相融; ②表现为同样的对象或者至少有同样的形状
- Distances > Cloud/Cloud dist. (cloud-to-cloud distance): 计算两个点云之间的距离
- Distances > Cloud/Mesh dist. (cloud-to-mesh distance): 计算点云和网格之间的距离
- Distances > Closest Point Set: 计算两个点云之间最近的点的集合
- Statistics > Local Statistical Test: 可以以标量域的局部统计为基础进行分割和过滤点云
- Statistics > Compute Stat. Params: 计算统计分布(高斯分布、威布尔分布)
- Segmentation > Label Connected Components:设置最小距离,把所选的云分割成更小的部分,每一部分相互连接
- Segmentation > Cross Section: 用户可以定义一个裁剪框,可调整框的范围和方向,来裁剪点云。可以用来:①在一个或多个维度 重复分割过程;②获取多边形的轮廓
- Segmentation > Extract Sections: 可以在一个点云的顶部画或者导入多边形来提取截面和轮廓
- Fit > Plane: 匹配点云中的一个平面和输出各种信息,如拟合 RMS、 垂直平面、地质的倾角、倾角方向值等
- Fit > Sphere: 适配点云中的一个球体
- Fit > 2D Polygon: 适配点云中的二维多边形
- Fit > Quadric: 适配点云中的2.5D曲面
- Other > Density: 估量一个点云的密度
- Other > Curvature: 估量一个点云的曲率
- Other > Roughness: 估量一个点云的粗糙程度
- Other > Remove duplicate points: 通过设置两点之间最小距离来删除重复的点

### Display:

- Full screen: 全屏
- Refresh: 刷新, 强制刷新有效的3D视图的内容(OpenGL图形重绘)
- Toggle Centered Perspective: 在正交视图和对象中心视图模式中切换
- Toggle Viewer Based Perspective: 在正交视图和透视图中切换
- Lock rotation about vert. axis: 锁定围绕Z轴的影像旋转
- Enter bubble-view mode: 进入泡沫视图模式
- Render to File: 可以渲染当前的3D视图成一个图像文件(支持多数标准文件格式),还可以缩放以适应更大分辨率的屏幕
- Display settings:对各种显示进行设置:颜色和材质、色阶、标签、其他
- Camera settings: 影像设置
- Save viewport as object: 保存当前3D视图的可视体的参数(影像位置和方、透视状态)为一个可视实体,这个实体自动地添加DB 树的根
- Adjust zoom: 调整缩放比例
- Test Frame Rate: 测试帧速率, 让有效的3D视图在一个较短时间旋转从而估量平均帧数, 结果在控制台显示
- Lights > Toggle Sun Light: 切换太阳光

- Lights > Toggle Custom Light: 切换自定义的光
- Shaders and Filters > Remove filter: 禁用任何活动的着色器或者OpenGL过滤器
- Active scalar field > Toggle color scale: 为所选活动的实体切换色阶
- Active scalar field > Show previous SF: 改变当前所选对象的标量域, 激活先前的标量域
- Active scalar field > Show next SF: 改变当前所选对象的标量域, 激活下一个的标量域
- Console: 控制台(显示/隐藏)
- Toolbars: 工具栏,包括主工具栏、标量域、视图、插件、GL滤镜
- Reset all GUI elements: 退出前自动存储当前GUI信息(位置和工具栏的可见性等),可以恢复原始配置

# Plugins:

#### Standard plugins:

- qHPR (Hidden Point Removal): 如果点云是闭合曲面,则可以过滤(删除)掉通过当前3D影像不能看到的云
- qPCL (Point Cloud Library Wrapper):有PCL库一些方法的接口,主要包括:①计算法线和曲率②异常点和噪声点的去除③平滑点云(移动最小二乘法)
- qPCV (ShadeVis / Ambient Occlusion): 计算点云的明亮度, 类似于光线来自于对象周围的半球或球体(可以自定义光线距离)
- qPoissonRecon (Poisson Surface Reconstruction): Poisson表面重建,用三角网络生成算法构建的简单的表面
- qRansacSD (RANSAC Shape Detection): 随机抽样一致形状检测, 运用自动形状检测算法的简单接口
- qSRA (Surface of Revolution Analysis): 计算一个点云和一个假定旋转平面之间的距离(旋转平面用2D轮廓定义),距离计算好后,用户可以创建一个偏差的2D图或者圆柱或圆锥的投影
- qCANUPO (Point Cloud Classification):可自动对点云进行分类,也可以手动分类
- qM3C2 (Robust C2C Distances Computation): 用独特的方法计算两个点云之间的有向(稳健)距离
- qCork (Boolean Operations on Meshes): 可以执行网格中的布尔操作(也称CSG = 构造实体几何), 它基于Cork库
- qAnimation: 动画渲染插件
- qFacets: 可以从点云中自动提取二维切面, 以它们的垂直距离分开
- qCSF (Cloth Simulation Filter): 基于布模拟滤波算法,能实现地面点与非地面点的分离,去除非地面点
- qCompass: 简单地实现点云中地质结构的它的轨迹的数字化
- qBroom (qVirtualBroom): 高效地扫描和清理
- qHoughNormals: 计算法法线
- qGMMREG: 对小型实体的非刚性云的匹配
- qLAS\_FWF: 这个插件可以读写标准雷达文件,可以在命令模式下打开LAS 1.3+文件
- qPoissonRecon: 可以让输入的点云颜色映射到成网格(快速直接地分配到颜色接近输入点颜色的网格顶点)

#### OpenGL 'shaders' plugins:

- qEDL (Eye Dome Lighting):实时底纹滤镜,用来在空白的点云或者网格中增强少量特质(除了几何信息外,它不依赖于其他信息)
- qSSAO (Screen Space Ambient Occlusion):实时底纹滤镜,与环境相似的遮挡
- qBlur: 一个简单的模糊处理滤镜, 主要用于开发人员的演示

#### Deprecated

• qKinect (Point Cloud Acquisition with a Kinect): 可以用Kinect设备获取(有色的)点云

# 3D Views:

New: 创建3D视图Close: 关闭3D视图

• Close All: 关闭所有3D视图

• Tile: 共享的所有 3D 视图之间的显示空间

• Cascade: 用串联的方式重新排列所有 3D 视图

• Next: 激活顺序创建的下一个3D视图

• Previous: 激活顺序创建的上一个3D视图

# Help:

• Help:

• About: CloudCompare版本信息

• About Plugins: 插件信息

# Thanks:

•

•

还有其他功能保存一下链接: