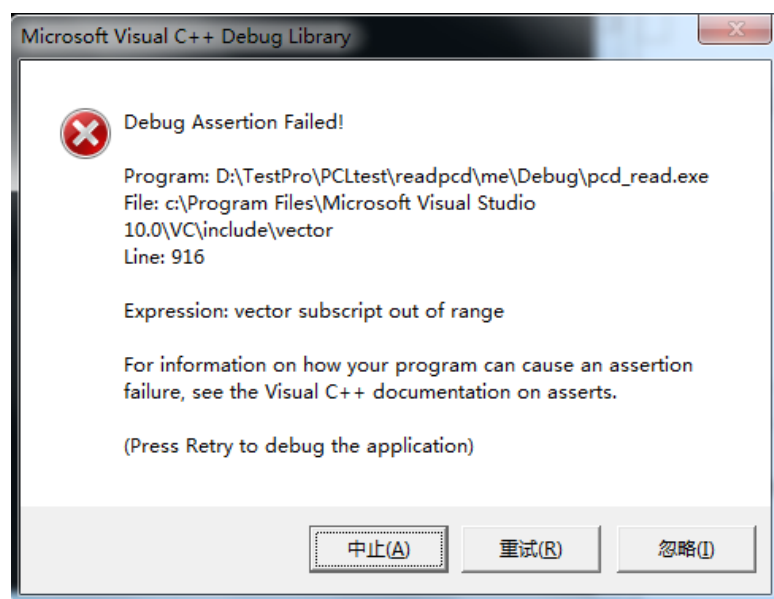


PCL 中下采样:

```
272 void vg_fliter(float voxelsize, pcl::PointCloud<PointRGBA>::Ptr cloud_filtered)
273 {
274     pcl::VoxelGrid<PointRGBA> vg;
275     vg.setInputCloud (cloud_filtered);
276     vg.setLeafSize (voxelsize, voxelsize, voxelsize);
277     vg.filter (*cloud_filtered);
278 };
```

离线的程序 pcd_read 中运行到断点处便出错。在线的程序 openni_grabber 却不会出错。

出错信息:



将 InputCloud 与 Output 改成不同的 Pointcloud 便可以运行。

由于 InputCloud 传入的指针类型，所以输入输出两个点云类共用同一地址空间，故猜想此处有问题。

根据 PCL 官网上的指示:

Can I still debug my PCL based project and step into PCL code even if I use PCL all-in-one installer?
Yes!

- Download the PDB archive that corresponds to the used PCL all-in-one installer;
- Unzip it inside the bin subfolder of your PCL installation folder, so that the pdb files lie near to the dlls;
- Once the debugger needs to step into PCL code, you will be asked for PCL sources folder. You can download the sources archive from [here](#). Unzip it somewhere on your disk, and give the debugger the path.

下载了对应版本的 pdb 文件。

源码之前已经下载过。

Debug 进入 voxel_grid.hpp 函数，找到问题处:

```
324     {
325         // Fill r/g/b data, assuming that the order is BGRA
326         pcl::RGB rgb;
327         memcpy (&rgb, reinterpret_cast<const char*> (&input_>points[index_vector[cp].cloud_point_index]) + rgba,
328             centroid[centroid_size-3] = rgb.r;
329             centroid[centroid_size-2] = rgb.g;
330             centroid[centroid_size-1] = rgb.b;
331     }
```

此时 input_>points.size()变为 4269，但 capacity 及 width 等其他属性仍为 11258。

又进了几次，从头开始排查，发现 size 的变化是从这个位置开始的：

```
279 // Fourth pass: compute centroid
280 output.points.resize(total);
281 if (save_leaf_layout_)
282 {
283     try
284     {
```

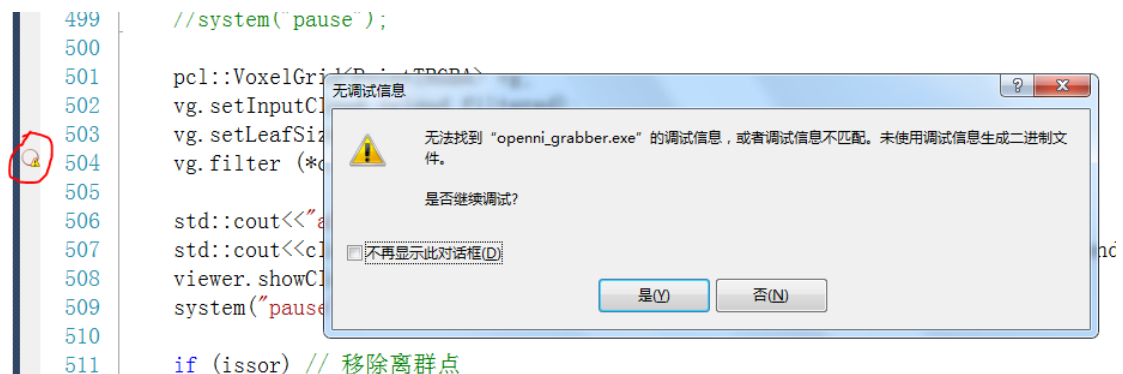
因为 input_ 与 output 共同地址空间，所以在对 output.points 进行 resize 的时候，input_>points.size()也随之改变了。

那么问题又来了。

为什么在线程序 openni_grabber 可以运行？

继续设断点，调试。

但又出现了新的问题：

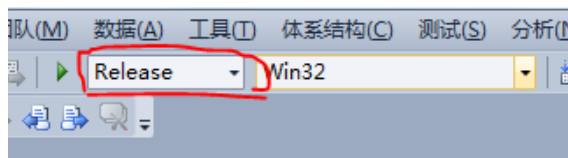


无法进入调试，而且断点也变成了红圈圈里那个样子。

原以为是因为在回调函数里面所以无法设断点，经过一番百度，无果。

后尝试在 main 函数里设断点，再进调试，仍然出现这个问题。

然后在与离线工程对比的时候发现：



于是将其改为 debug，调试成功。

并且出现了与离线程序相同的问题。

于是再把离线工程改为 release，运行成功.....

.....

我说之前怎么感觉离线的程序跑起来格外的慢，怪不得师兄曾和我反复强调要用 release 模式.....

可！是！谁 TM 能想到两种模式不止是快慢的问题啊！！

由于 release 模式下无法进调试，所以暂时还无法确定在这两种模式下函数内部执行的区别。