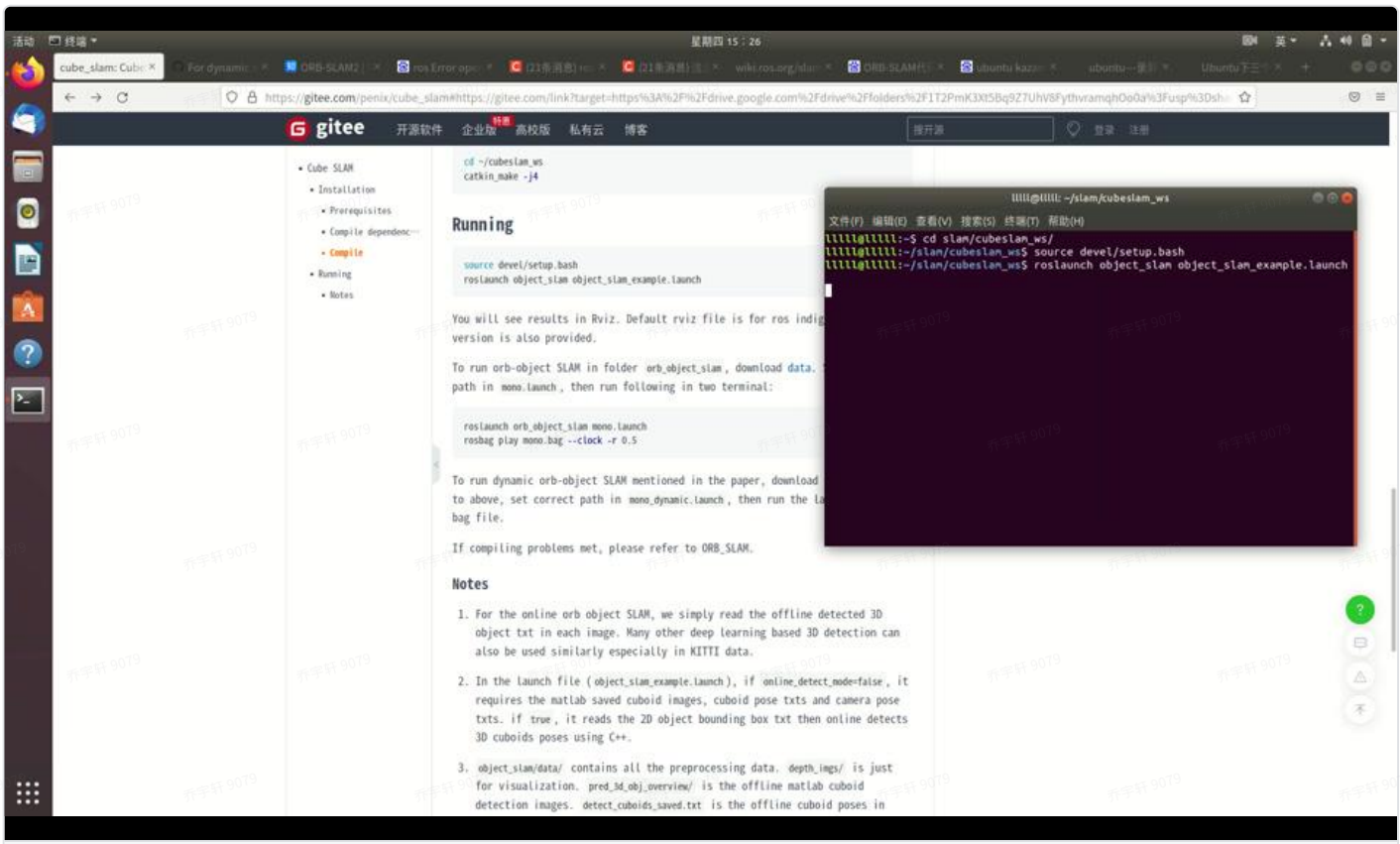
CubeSLAM示例编译运行

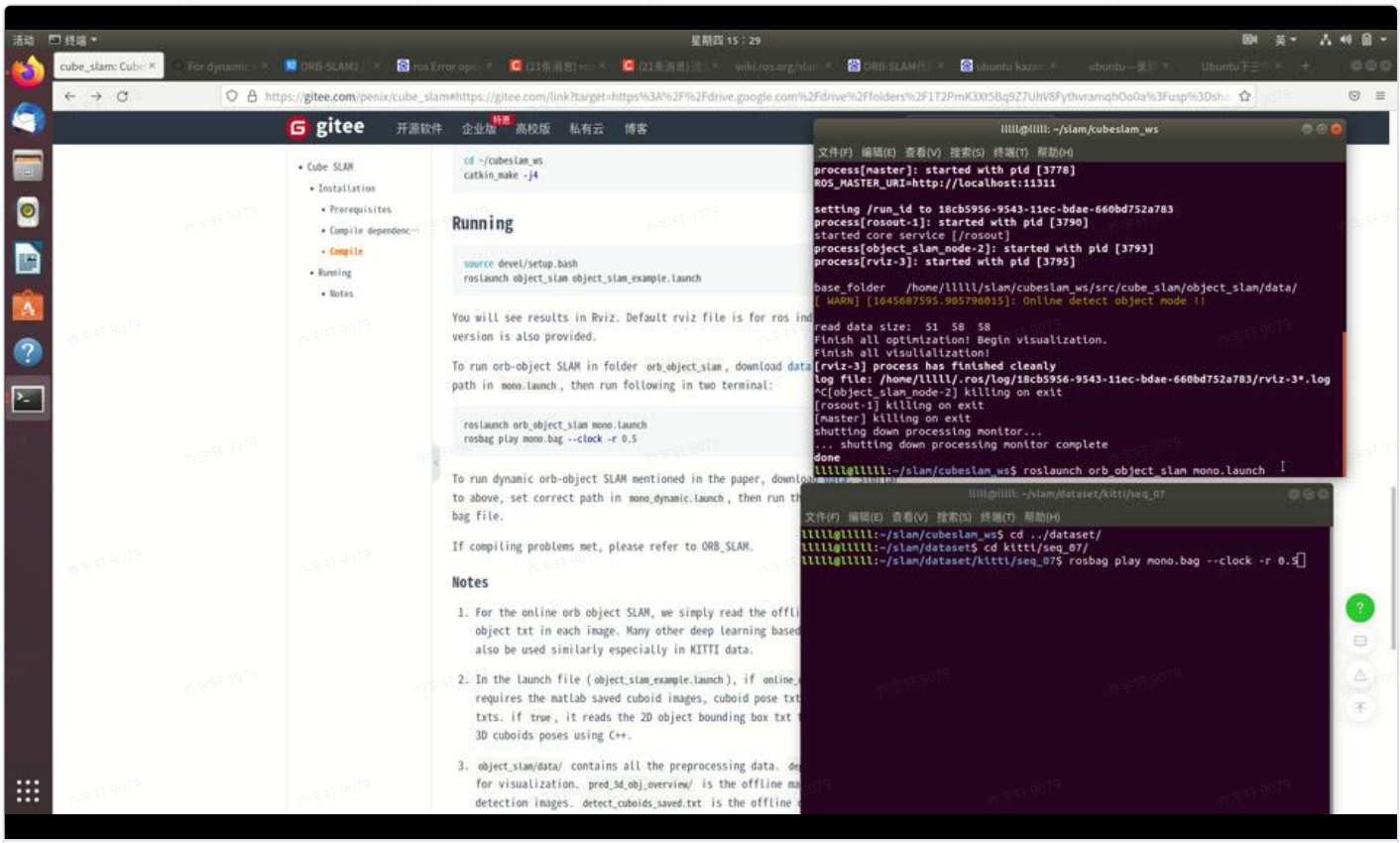
效果

orbject_slam



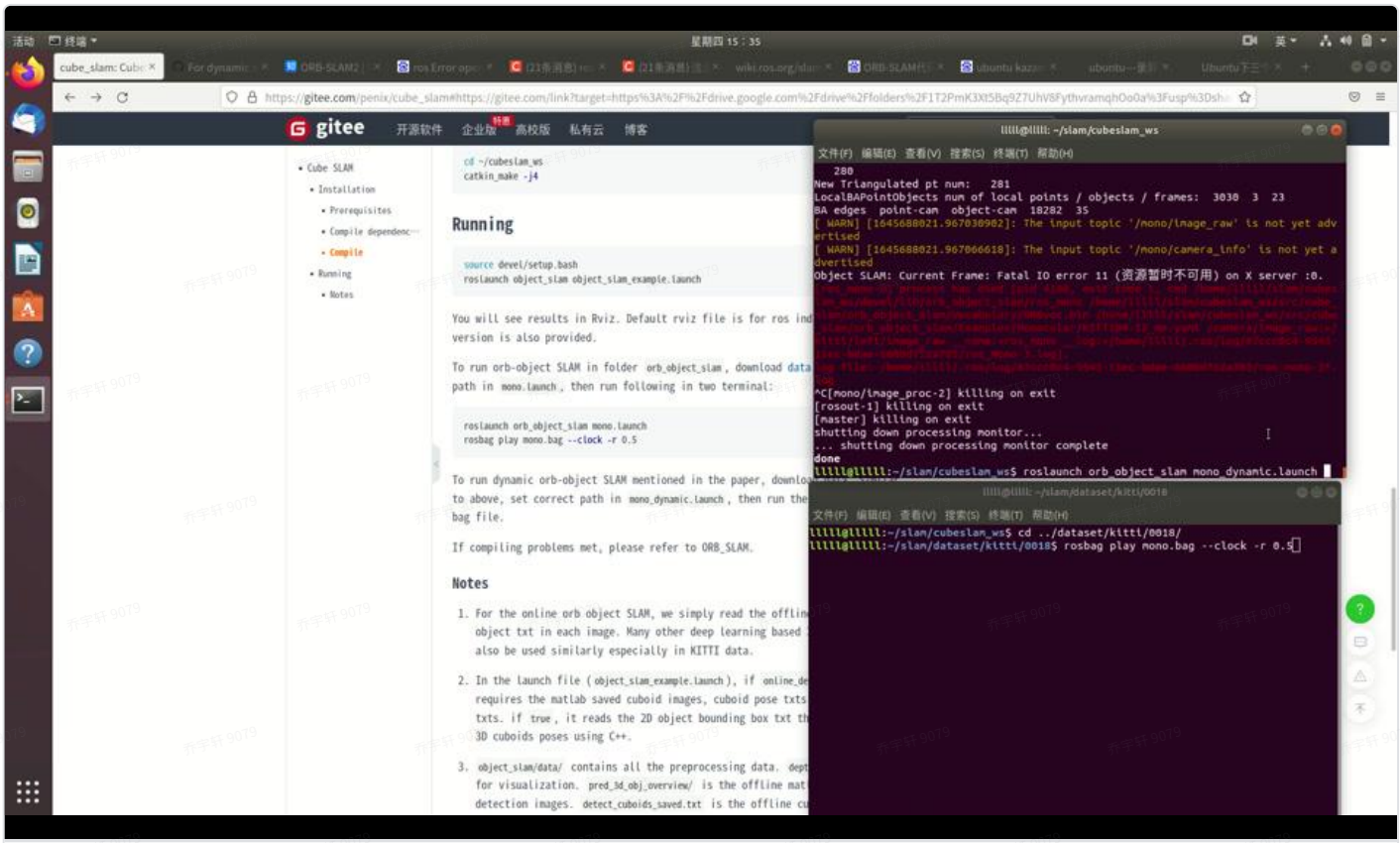
orbject.mp4

orb_object_slam mono



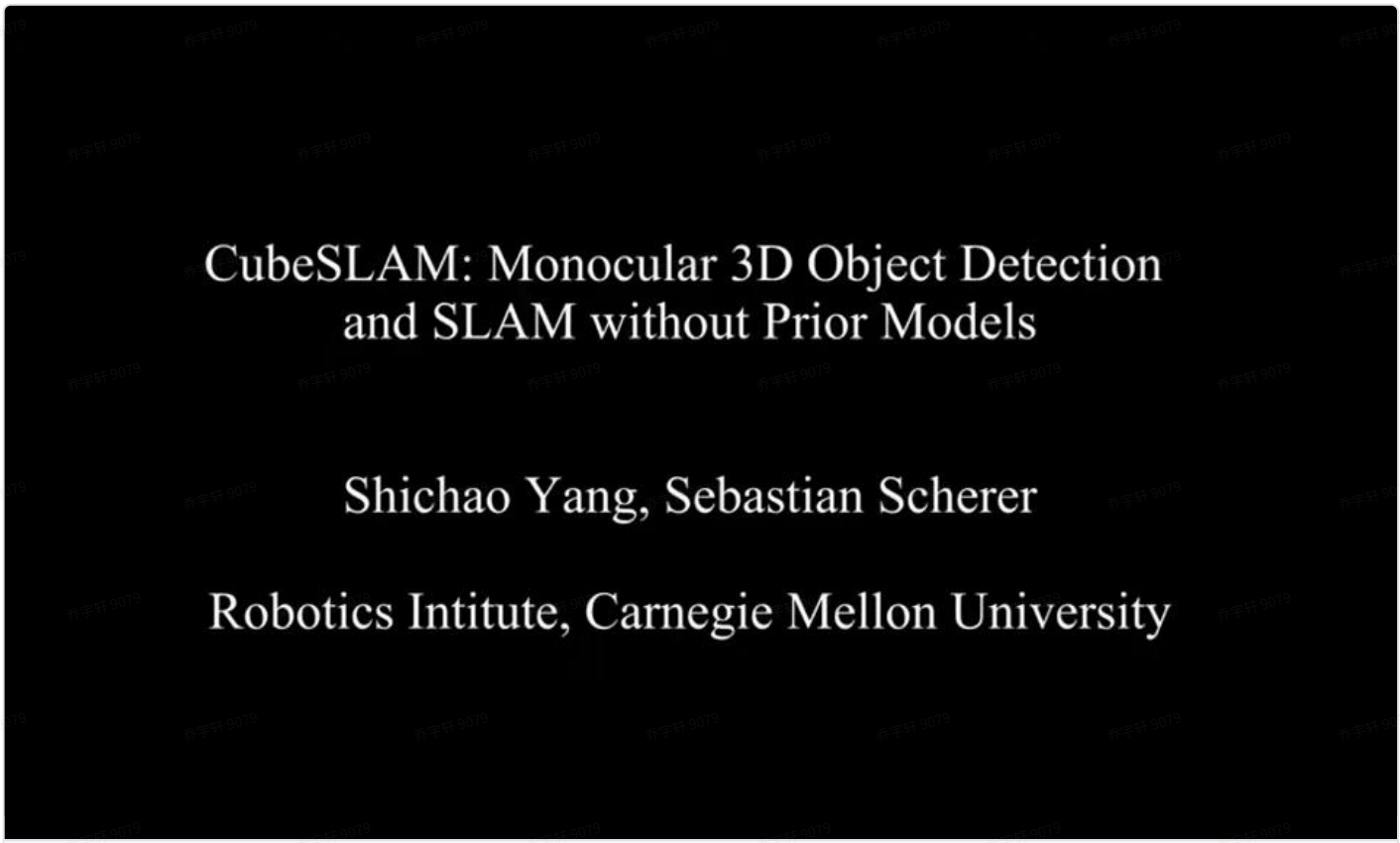
mono.mp4

orb_object_slam mono dynamic



mono_dynamic.mp4

这里给出一个作者在youtube上的效果视频，在源码基础上做了他自己的一些修改，效果比示例跑出来要好一些



demo.mp4

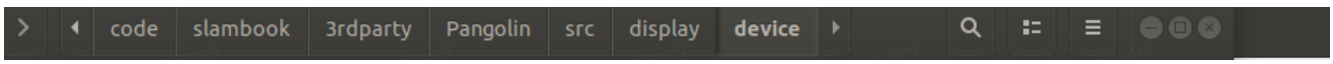
记录

源码：GitHub - shichaoy/cube_slam: CubeSLAM: Monocular 3D Object Detection and SLAM

编译环境：Ubuntu 18.04 + ros melodic

CubeSLAM依赖：Eigen.Pangolin.Opencv.PCL

1. 源码编译会在"iota","std:vector","usleep"等处报错，需要在报错处对应添加#include <numeric>,#include <vector>，#include <unistd.h>。此处有已修改好的源码：GitHub - zhezhou1993/cube_slam: CubeSLAM: Monocular 3D Object Detection and SLAM
2. 采用Eigen 3.3.7版本可以编译成功，但在跑object slam时出现double free or corruption的问题，github上提到是Eigen对齐问题。换成ubuntu源自带的libeigen3-dev可以成功，eigen版本为3.2.92。
3. 如果使用高翔老师《视觉SLAM十四讲》第三方库中的Pagolin，有可能报错Pagolin X11，按路径注释掉图中两行重新编译即可。



```
打开(O)  display_x11.cpp  ~/code/slambook/3rdparty/Pangolin/src/display/device  保存(S)
mono.launch | display_x11.cpp
) {
    display = XOpenDisplay(NULL);

    if (!display) {
        throw std::runtime_error("Pangolin X11: Failed to open X display");
    }

    // Desired attributes
    static int visual_attribs[] =
    {
        GLX_X_RENDERABLE    , True,
        GLX_DRAWABLE_TYPE   , GLX_WINDOW_BIT,
        GLX_RENDER_TYPE     , GLX_RGBA_BIT,
        GLX_X_VISUAL_TYPE   , GLX_TRUE_COLOR,
        GLX_RED_SIZE        , 8,
        GLX_GREEN_SIZE      , 8,
        GLX_BLUE_SIZE       , 8,
        GLX_ALPHA_SIZE      , 8,
        GLX_DEPTH_SIZE      , 24,
        GLX_STENCIL_SIZE    , 8,
        GLX_DOUBLEBUFFER    , glx_doublebuffer ? True : False,
        //GLX_SAMPLE_BUFFERS  , glx_sample_buffers,
        //GLX_SAMPLES         , glx_sample_buffers > 0 ? glx_samples : 0,
        None
    };

    int glx_major, glx_minor;
    if ( !glXQueryVersion( display, &glx_major, &glx_minor ) ||
         ( ( glx_major == 1 ) && ( glx_minor < 3 ) ) || ( glx_major < 1 ) )
    {
        // FBConfigs were added in GLX version 1.3.
        throw std::runtime_error("Pangolin X11: Invalid GLX version. Require GLX >= 1.3");
    }
C++  制表符宽度: 8  第 126 行，第 7 列  插入
```

4. orb_object_slam会在Vocabulary文件夹下报错。是因为论文作者应该后期优化过代码，词袋部分使用了ORBvoc.bin，直接下载的代码并没有，将ORBvoc.bin拷贝到Vocabulary文件夹下。

ORBvoc.bin.zip
31.48MB

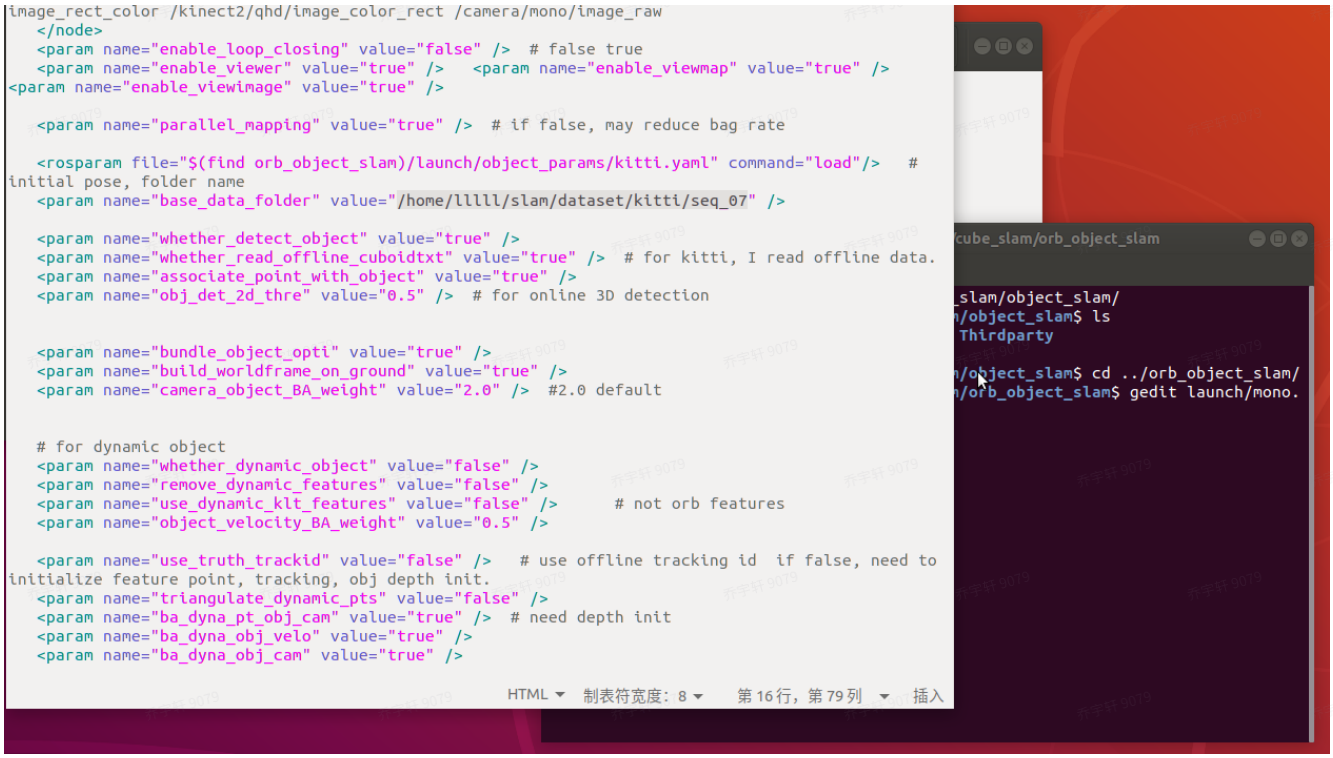
5. 数据集来自作者github上所给链接，seq_07为mono的数据，0018为mono_dynamic的数据

seq_07-20220223T022837Z-001.zip
328.20MB

0018-20220223T030930Z-001.zip
114.95MB

数据路径在orb_object_slam/launch/mono.launch中修改，使用的是seq_07数据

```
# /kitti/left/image_raw   /camera/rgb/image_raw /camera/rgb/image_color  /mono/
```



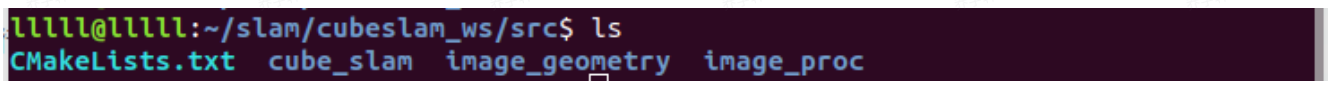
类似的修改mono_dynamic.launch中的路径，使用0018数据

以mono为例运行，在两个终端分别执行如下命令，运行mono_dynamic改为mono_dynamic.launch

```
roslaunch orb_object_slam mono.launch
rosbag play mono.bag --clock -r 0.5
```

第一行在cubeslam_ws下执行，第二行在数据集路径下执行，注意作者所给seq_07中的bag文件名不是mono，需要自己重命名以下，否则打不开mono.bag。

6. 如果Opencv版本合适，基本上就跑通示例了，对于部分Opencv版本会报gtk 2.0与gtk 3.0不能同时存在的问题。这是由于ros melodic自带了opencv 3.2.0与自己安装的opencv版本会不兼容。解决办法为：卸载ros自带的opencv `sudo apt-get remove libopencv-core3.2`，然后补齐ros所需的功能包，地址：https://github.com/ros-perception/vision_opencv， 包括cv_bridge,image_ geometry,image_proc等，放入ros工作站(cubeslam_ws)的src文件夹下，参考专栏:https://zhuanlan.zhihu.com/p/400316912，我在Config文件中链接好了cv_bridge，所以src中只有两个功能包。



重新编译，如果出现cv_bridge编译报错，有可能是与python版本冲突的原因，解决方法参考：https://gitee.com/bingobinlw/cv_bridge/blob/master/README.md

以上就是CubeSLAM编译运行遇到的一些问题，最终Eigen使用ubuntu源自带的libeigen3-dev，版本为3.2.92，pagolin为高翔slambook的第三方库，opencv版本为3.4.0(会出现gtk报错)，PCL来自github上的最新版本，gitee上有镜像文件。