

CubeSLAM跑kitti数据

CubeSLAM模式选择上可以选择是否读取offline 3D立方体数据与是否检测动态物体,并且可由ros在launch文件中修改,参考源码orb_object_slam模块中的mono.launch与mono_dynamic.launch

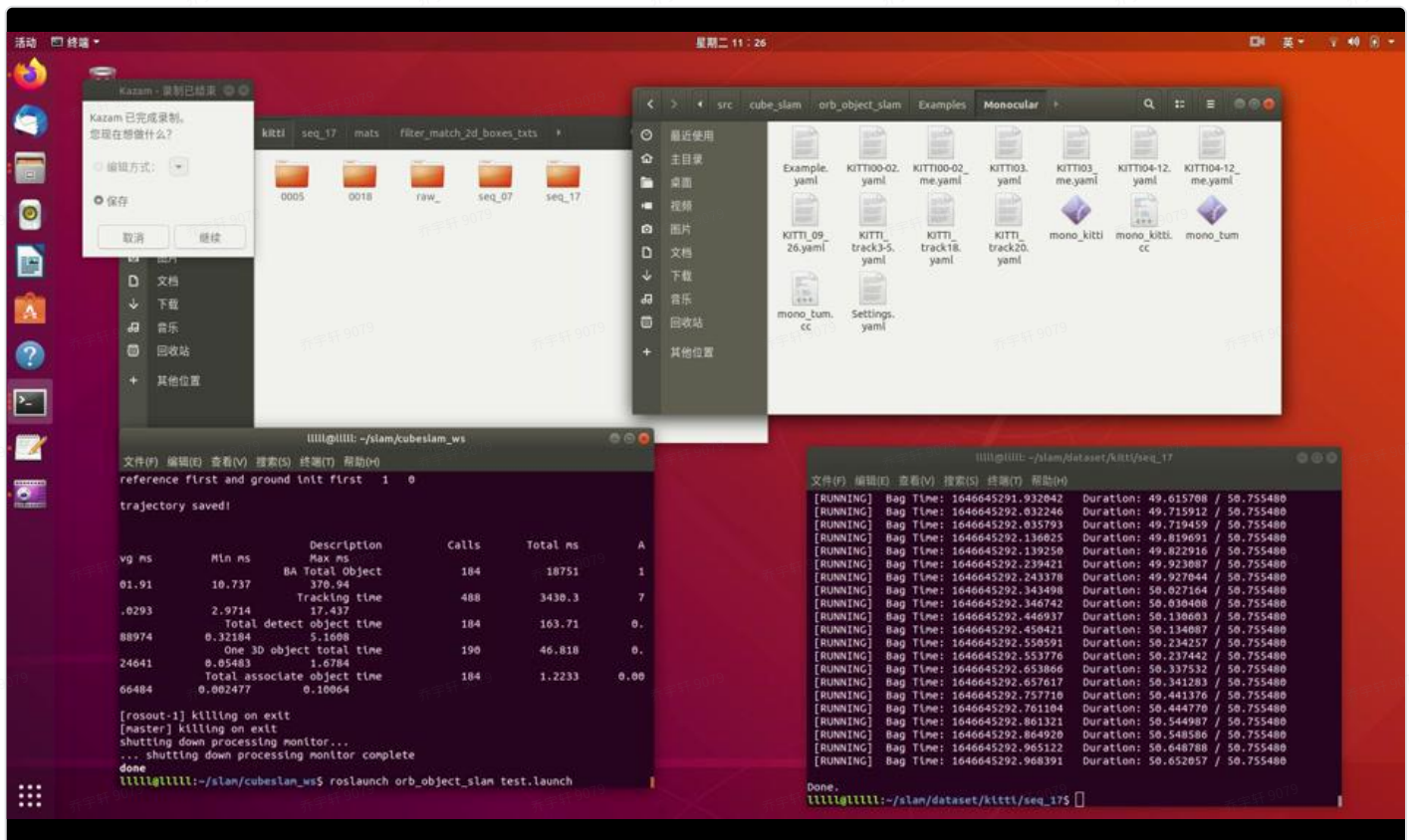
```
<param name="whether_detect_object" value="true" />
<param name="whether_read_offline_cuboidtxt" value="true" /> # for kitti, I read offline data.
<param name="associate_point_with_object" value="true" />
<param name="obj_det_2d_thre" value="0.5" /> # for online 3D detection

# for dynamic object
<param name="whether_dynamic_object" value="false" />
<param name="remove_dynamic_features" value="false" />
<param name="use_dynamic_klt_features" value="false" /> # not orb features
<param name="object_velocity_BA_weight" value="0.5" />
```

对应修改的参数为whether_read_offline_cuboidtxt与whether_dynamic_object和remove_dynamic_features。

效果与分析

我们选取了kitti odometry数据的seq_17, 设定whether_read_offline_cuboidtxt=false与whether_dynamic_object=false, 结合YOLO目标检测与作者的线检测源码给出了如下demo:



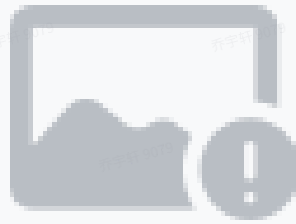
mono.mp4

效果相对于作者给出的demo有一些差距。作者demo中所给的是read_offline_cuboidtxt，我们此处是根据2d bbox与lbd线检测结果在线生成cuboid，在demo中我们可以看到对于object的cuboid生成在距离较近且速度较慢时比较稳定。对于远距离快速运动的物体，生成的cuboid有所偏差，并且demo的36s处在无遮挡的情况下都存在漏检。核对了YOLO的bbox数据后发现，cuboid没有生成时，对应的2d检测都是没有问题的。cuboid生成不出来的原因，一是源码中的detect_3d_cuboid模块于论文中思路有所出入，作者后来提到源码中的detect_3d效果不好，转而给出了一个新的生成cuboid的matlab源码，matlab版本的detect_3d与论文中的思路更接近一些，后续会尝试利用该源码生成cuboidtxt进行离线读取，二是远距离的object的纹理在快速的线检测算法中很难被检测到，lbd在这种情况下更多的检测到的是背景上的线条而不是object的线条，从而生成cuboid有所困难，参考下图：



记录

1. cubeslam读取图片序列是通过rosbag读取的。对于kitti数据，我们需要制作相应的rosbag，可以采用kitti2bag的脚本<https://github.com/tomas789/kitti2bag/>因为版本问题源码会出现许多bug，这里给出一份debug过的脚本文件，依赖pykitti



📎 kitti2bag.py

如果我们要制作2011_10_26的rosbag，在2011_10_26的文件夹下调用脚本，格式如下：

kitti2bag -t 2011_10_03 -r 0042 raw_synced [-t后加日期][-r后加序列][raw_synced为kitti数据类型]

raw_data数据需要下载对应的标定文件进行制作，对于odometry数据也可类似的转换为bag文件

转换好rosbag后，注意看bag文件中的topic信息

```
lllll@lllll:~$ cd slam/dataset/kitti/seq_17/
lllll@lllll:~/slam/dataset/kitti/seq_17$ rosbag info 17.bag
path:          17.bag
version:       2.0
duration:      50.8s
start:         Mar 07 2022 17:27:22.32 (1646645242.32)
end:           Mar 07 2022 17:28:13.07 (1646645293.07)
size:          437.6 MB
messages:      1964
compression:   none [492/492 chunks]
types:         sensor_msgs/CameraInfo [c9a58c1b0b154e0e6da7578cb991d214]
               sensor_msgs/Image      [060021388200f6f0f447d0fcd9c64743]
topics:        /kitti/camera_gray_left/camera_info  491 msgs    : sensor_msgs/Ca
               /kitti/camera_gray_left/image_rect   491 msgs    : sensor_msgs/Im
```

```
image /kitti/camera_gray_right/camera_info 491 msgs : sensor_msgs/CameraInfo
image /kitti/camera_gray_right/image_rect 491 msgs : sensor_msgs/Image
```

我们在ros_mono中remap的topic是/kitti/camera_gray_left/image_rect //单目只用一个topic

2. 我们通过YOLO对图片序列进行bbox检测，但YOLO源码中没有批量输出的选项。我们通过对源码进行一些修改来实现批量检测，主要修改darknet下的image.c,detector.c,darknet.c三个文件



detector.c



image.c



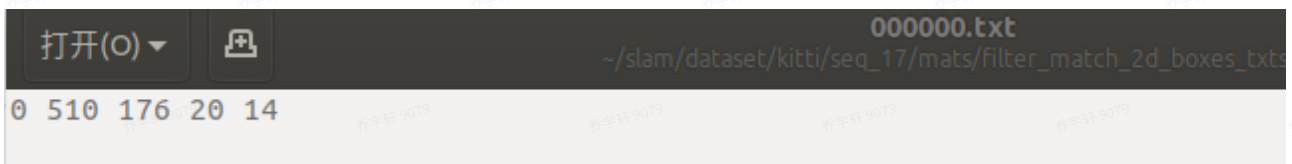
darknet.c

在src文件夹中替换了三个文件并对darknet编译后，可以使用shell脚本批处理(路径需要修改)



batch.sh

输出的txt文件格式已经经过修改与cubeslam读取的格式对齐



0代表标注，后四个数字分别是boundingbox的左上角坐标x,y 以及boundingbox的宽和高

3. 下面开始调用cubeslam中的源码。kitti数据中图片序列以6位左0填充的方式作为命名(%06d)，作者在源码中的数据格式为%04d_raw_data,对于相应的部分我们要修改为%06d。首先使用cubeslam源码中的线检测line_lbd模块实现图片序列的线检测，目标是批量的获得线检测的txt数据

在line_lbd文件夹下新建image文件夹，将图片序列放入其中，并新建saved_edges文件夹来存放批量的线检测txt数据，使用launch文件夹下的launch文件实现线检测

HTML

```
1  <!-- -->
2  <launch>
3
4      <node pkg="line_lbd" type="line_lbd_detect_node" name="line_lbd_detect_node"
        output="screen"
5          args="$(find line_lbd)/data/image/000000.png">
6      </node>
7
8      <param name="save_folder" value="$(find line_lbd)/data/saved_edges/" />
9      <param name="use_LSD_algorithm" value="true" /> # true, false   LSD or
        Edline detector.
10     <param name="save_to_imgs" value="false" />
11     <param name="save_to_txts" value="true" />
12
13 </launch>
```

launch文件如上修改，主要是设定save_to_txts这一参数与路径，修改后可以使用shell脚本批处理



📄 batch_for_lbd.sh

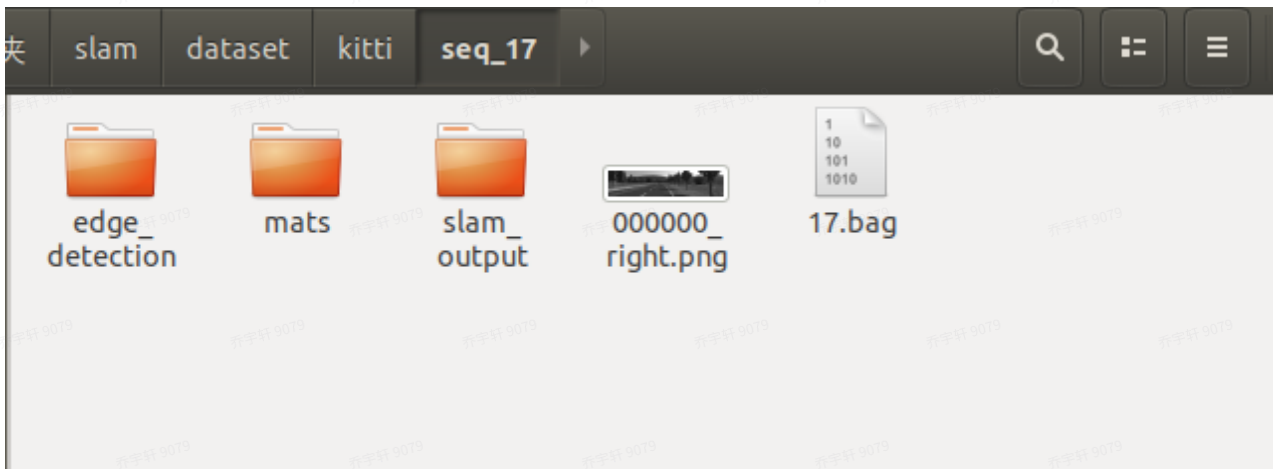
除了要修改路径以外，运行脚本会进入ros进程等待退出，可以新建一个脚本每隔3s杀死ros进程

Bash

```
1 #!/bin/bash
2 while :
3 do
4 sleep 3
5 killall -9 roscore
6 killall -9 rosmaster
7 killall -9 ros
8 done
```

以上两个脚本需要在两个终端运行，线检测数据会以图片序号.txt为名保存在saved_edges文件夹下

4. 有了2d bbox数据与线检测数据，我们可以制作数据集来跑cubeslam的在线模式，以seq_17为例



edge_detection中存放批量线检测得到的txt数据，mats文件夹下新建filter_match_2d_boxes_txts文件夹来存放2d bbox数据，格式先前已经提过，000000_right.png为图片序列中第一张右相机的图像(kitti数据集两组序列image_0和image_1分别是右相机与左相机)，17.bag为使用kitti2bag脚本对序列17生成的rosv文件

5. 制作好数据集后，可以跑read_offline_cuboidtxt=false时的orb_object_slam

在orb_object_slam模块src的Tracking.cc中我们需要修改数据名到%06d，这里给出一份修改过的



Tracking.cc

cub slam源码修改后都需要重新编译。我们在launch文件夹下新建launch文件，可参考mono.launch的写法，注意设定whether_read_offline_cuboidtxt=false，remap的topic改为/kitti/camera_gray_left/image_rect，并且注意修改base_folder，这里给出一份launch文件供参考

PowerShell

```
1 <!-- -->
2 <launch>
3   # ROS_NAMESPACE=mono rosrn image_proc image_proc  bayer to bgr8
4   <node pkg="image_proc" type="image_proc" name="image_proc" ns="mono"
args="bayer to bgr8" />
5   <node pkg="orb_object_slam" type="ros_mono" name="ros_mono"
output="screen" args="$(find orb_object_slam)/Vocabulary/ORBvoc.bin
6   $(find orb_object_slam)/Examples/Monocular/KITTI04-12.yaml"> #
KITTI_09_26 KITTI04-12_me kinect_mev2 TUM_mono tamu_corridor
7   <remap from="/camera/image_raw" to
="/kitti/camera_gray_left/image_rect"/>
8   # /kitti/left/image_raw /camera/rgb/image_raw
/camera/rgb/image_color /mono/image_rect_color /kinect2/qhd/image_color_rect
/camera/mono/image_raw
9   </node>
10  <param name="enable_loop_closing" value="false" /> # false true
11  <param name="enable_viewer" value="true" /> <param name="enable_viewmap"
value="true" /> <param name="enable_viewimage" value="true" />
```

```

12
13     <param name="parallel_mapping" value="true" /> # if false, may reduce bag
rate
14
15     <rosparam file="$(find orb_object_slam)/launch/object_params/kitti.yaml"
command="load"/> # initial pose, folder name
16     <param name="base_data_folder"
value="/home/lllll/slam/dataset/kitti/seq_17" />
17
18     <param name="whether_detect_object" value="true" />
19     <param name="whether_read_offline_cuboidtxt" value="false" /> # for kitti,
I read offline data.
20     <param name="associate_point_with_object" value="true" />
21     <param name="obj_det_2d_thre" value="0.5" /> # for online 3D detection
22
23
24     <param name="bundle_object_opti" value="true" />
25     <param name="build_worldframe_on_ground" value="true" />
26     <param name="camera_object_BA_weight" value="2.0" /> #2.0 default
27
28
29     # for dynamic object
30     <param name="whether_dynamic_object" value="false" /> #
31     <param name="remove_dynamic_features" value="false" /> #
32     <param name="use_dynamic_klt_features" value="false" /> # not orb
features
33     <param name="object_velocity_BA_weight" value="0.5" />
34
35     <param name="use_truth_trackid" value="false" /> # use offline tracking
id if false, need to initialize feature point, tracking, obj depth init. #
36     <param name="triangulate_dynamic_pts" value="false" />
37     <param name="ba_dyna_pt_obj_cam" value="true" /> # need depth init #
38     <param name="ba_dyna_obj_velo" value="true" />
39     <param name="ba_dyna_obj_cam" value="true" />
40
41     # for depth initialization
42     <param name="mono_firstframe_truth_depth_init" value="false" /> #
43     <param name="mono_firstframe_Obj_depth_init" value="false" />
44     <param name="mono_allframe_Obj_depth_init" value="false" /> # may not
need for kitti
45
46     # for ground height scaling
47     <param name="enable_ground_height_scale" value="false" /> # for kitti
48     <param name="ground_everyKFs" value="10" />
49     <param name="ground_roi_middle" value="3.0" /> # 3(1/3) or 4(1/2)
50     <param name="ground_roi_lower" value="3.0" /> # 2 or 3
51     <param name="ground_inlier_pts" value="20" />
52     <param name="ground_dist_ratio" value="0.08" />

```

```

53
54 # save result
55 <param name="whether_save_online_detected_cuboids" value="false" />
56 <param name="whether_save_final_optimized_cuboids" value="false" />
57
58 # gui drawing parameters usually set to true, for paper, set to false
59 <param name="draw_map_truth_paths" value="true" />
60 <param name="draw_nonlocal_mappoint" value="true" />
61
62
63 </launch>

```

6. 参考 [CubeSLAM示例编译运行](#) 示例的运行，将launch文件改为test.launch(根据自己launch文件的命名)，在新建终端在数据集文件夹下roslaunch test.launch，即可得到demo中的结果

如果在过程中出现tracking lost的情况，在launch文件中对应的.yaml中(代码块中的第6行，KITTI04-12.yaml)找到ORB feature参数并适当设定高一些，一般设置为3000就不会出现tracking lost的问题，这也说明cubeslam也是比较依赖特征点的方法



