

大量的使用了 pytorch 版的 sfmlerner 和 monodepth 的代码。
训练素材都是我用手机拍摄的

需要用同一手机拍摄多段视频：

- 1, calibration.mp4 这个视频用于分解生成标定板的图片，标定手机相机
- 2, 很多段手机拍摄的视频，比如 1.mp4、2.mp4、3.mp4 ... 这些就取名叫训练视频吧。

calibration.mp4 大概拍个一两分钟就可以，拍标定板，大概分解出二三十张图片出来用于标定。

训练视频，就是你拿着手机到室外拍的视频，**因为不处理动态物体，所以你拍的时候画面中不要有动态物体！不要有人和车走动！我没有把这部分的 tensorflow 代码转成 pytorch，要的话你们自己写吧。**

然后把上面拍的视频全部分解成图片写到文件夹中。

比如 calibration.avi 可以每隔 20 帧取一帧写入文件夹。

训练视频每隔 5 帧取一帧写入文件夹。看你拍视频时走的快慢，走的慢的话每隔 10 帧写一张也可以，根据实际情况确定。

注意：拍摄训练视频的时候最好走快一点，因为参考的 sfmlerner 里头是这样做的。它把 kitti 中汽车停车等红灯时的静态图片都剔除了，我就照搬过来。

其实分解训练视频成图片的原则是：两张图片之间有一定的运动，但是运动又不是太大。如果有的时候走的太慢，可以手动删除一些图片，让前后两帧图片之间的运动大一点。看 kitti 的训练图片就知道什么样的效果最合适。

最好绕着建筑物拍个回环吧，因为如果有回环的话，你还可以用 orbslam 之类的算法看看 posenet 生成的 pose 准确度如何。

dataset 文件夹的结构就是下面有 video1, video2, ... 这样的文件夹，然后这些文件夹下面是对应的视频分解出来的图片。
里面留了几张图片演示一下怎么用。

手机相机没什么畸变，所以只是标定了下内参，没做畸变校正。

所以阅读顺序是先看

split.py (需要手动分解各个视频到文件夹中)

calib.py

data_loader.py (顺带看下 custom_transforms.py, data_loader.py 里头的内参 intrinsic 要手动改为 calib.py 中标定出的内参)

然后看

DispNetS.py

PoseExpNet.py

然后就是

main.py

loss_func.py

util.py

这 3 个一起看

params.py 给各个文件提供参数

很多地方参数我都写死了，没有提供在 params.py 里头

还有很多必要的部分没写，兴致来了我再补上。

给了一个之前我自己写的 monodepth 的主程序做参考。