**EVO使用总结**

**视觉SLAM基础：算法精度评价指标（ATE、RPE）**

* 当我们需要评估一个SLAM/VO算法的表现时，可以从时耗、复杂度、精度多个角度切入，其中对精度的评价是我们最关注的，这个过程中不可避免会遇到两个精度指标**ATE**和**RPE**。这两个evaluation metrics最早是在TUM数据集benchmark中定义的，应用非常广泛。
* 介绍前，先对定义一下公式标注——算法估计位姿：
* 真实位姿： 下标代表时间 （或帧）,这里我们假设估计位姿和真实位姿各帧时间已对齐，总帧数相同。 表示相隔时间。

1. **RPE： relative pose error 相对位姿误差**

相对位姿误差主要描述的是相隔固定时间差两帧位姿差的精度（相比真实位姿），相当于直接测量里程计的误差。 因此第i帧的RPE定义如下：

已知总数n与间隔的情况下，可以得到 个RPE，然后我们可以用均方根误差RMSE统计这个误差，得到一个总体值：

其中, 代表取相对位姿误差中的平移部分translation。当然也有人不用RMSE，直接使用平均值、甚至中位数来描述相对误差情况。需要注意的是，除了平移误差，RPE也包含旋转误差，但通常使用平移误差进行评价已经足够，如果需要，旋转角的误差也可以使用相同的方法进行统计。到这一步，我们基本可以从RMSE值的大小来评价算法的表现，然而实际情况中，我们发现对的选取有多种选择，为了能综合衡量算法表现，我们可以计算遍历所有的RMSE的平均值：

但这样新的问题又出现了，这样的计算复杂度非常高，很耗时间，因此TUM在自己给定的工具中，通过计算固定数量的RPE样本计算一个估计值作为最终结果。

2. **ATE： absolute trajectory error 绝对轨迹误差**

绝对轨迹误差是估计位姿和真实位姿的直接差值，可以非常直观地反应算法精度和轨迹全局一致性。需要注意的是，估计位姿和groundtruth通常不在同一坐标系中，因此我们需要先将两者对其：对于双目SLAM和RGB-D SLAM,尺度统一，因此我们需要通过最小二乘法计算一个从估计位姿到真实位姿的转换矩阵 ；对于单目相机，具有尺度不确定性，我们需要计算一个从估计位姿到真实位姿的相似转换矩阵。 因此第i帧的ATE定义如下：

与RPE相似，建议使用RMSE统计ATE

当然，使用平均值、中位数等来反应ATE亦可，现在很多evaluation工具会将RMSE、Mean、Median都给出。旋转误差可以通过相同的方式计算，目前的一些开源评测工具都提供了对应的选项。

综上，RPE与ATE具有强烈的相关性，但含义不尽相同，需结合实际，选择合适的指标进行算法评价。

**EVO:里程计测评工具**

**EVO是一款用于视觉里程计和slam问题的轨迹评估工具。核心功能是能够绘制相机的轨迹，或评估估计轨迹与真值的误差。支持多种数据集的轨迹格式（TUM、KITTI、EuRoC MAV、ROS的bag），同时支持这些数据格式之间进行相互转换。**

**EVO安装**

evo是个python工具，安装就一行代码：

|  |
| --- |
| Bash pip install evo --upgrade --no-binary evo |

**EVO支持的数据格式**

evo共支持kitti、tum、euroc这三个公开数据集格式。需要注意待比较的数据文件必须格式一致，都为kitti、tum或者euroc！

**TUM、KITTI、EuRoC数据格式**

**TUM数据格式:**每行有8个元素，结尾没有空格，时间以秒为单位

|  |
| --- |
| Plain Text timestamp tx ty tz qx qy qz qw |

**EuRoC数据格式:**每行有17个元素，逗号隔开，时间以纳秒为单位

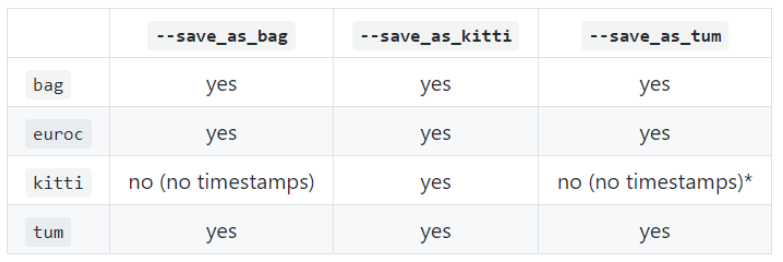
|  |
| --- |
| Plain Text timestamp,px,py,pz,qw,qx,qy,qz,vx,vy,vz,bwx,bwy,bwz,bax,bay,baz |

**KITTI数据格式:**储存变换矩阵的前三行，每行有12个元素，空格隔开没有时间

|  |
| --- |
| Plain Text r11 r12 r13 tx r21 r22 r23 ty r31 r32 r33 tz |

**各种数据格式之间的转换**

由于数据格式包含数据的关系，三种数据格式并不是均可以相互转换，下图说明了可以进行转换的数据格式。



将EuRoc的数据格式转化成TUM数据格式

|  |
| --- |
| C++ evo\_traj euroc data.csv --save\_as\_tum |

将TUM数据转换成KITTI格式

|  |
| --- |
| C++ evo\_traj tum traj\_1.txt 　traj\_2.txt 　traj\_3.txt --save\_as\_kitti |

将数据格式转化成rosbag

|  |
| --- |
| C++ evo\_traj tum traj\_1.txt traj\_2.txt traj\_3.txt --save\_as\_bag |

**EVO命令**

本节采用数据为VINS-Fusion采用双目IMU运行MH\_01\_easy.bag数据，采用TUM数据的格式记录的运行轨迹和回环轨迹。

evo工具主要有如下六个常用命令：

* evo\_traj这个主要是用来画轨迹、输出轨迹文件、转换数据格式等功能；
* evo\_ape用于评估绝对位姿误差；
* evo\_rpe用于评估相对位姿误差；
* evo\_res比较来自evo\_ape或evo\_rpe生成的一个或多个结果文件的工具；
* evo\_fig（实验）工具，用于重新打开序列化图（使用–serialize\_plot保存）；
* evo\_config 这个主要用于evo工具全局设置和配置文件操作。

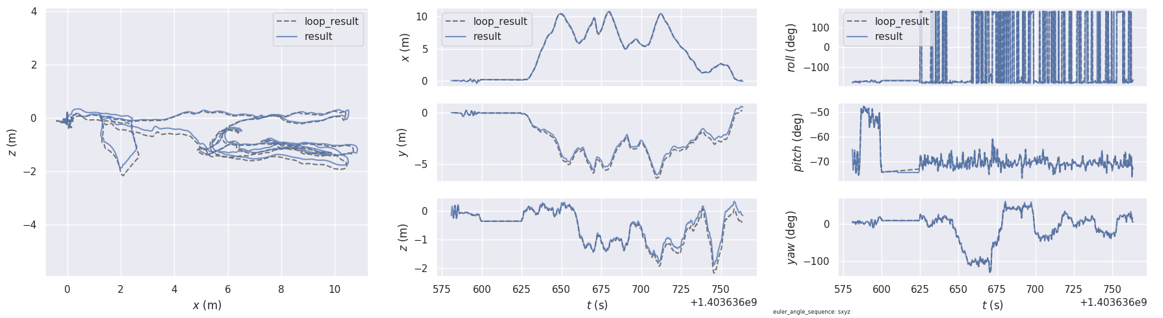
**一、实用命令之——evo\_traj**

这是一个分析，绘制或者导出一个或多个轨迹的命令。用法如下：

|  |
| --- |
| C++ evo\_traj tum result.csv --ref=loop\_result.csv -p --plot\_mode=xz |

tum为数据集类型；result.csv和loop\_result.csv为所使用的数据；--ref指定参考的数据；-p输出图片，如果没有这个参数，那么只在终端输出相关信息，不绘制图像；--plot\_mode指定绘制的坐标轴，上面命令表示轨迹按照x轴和z轴进行绘制。

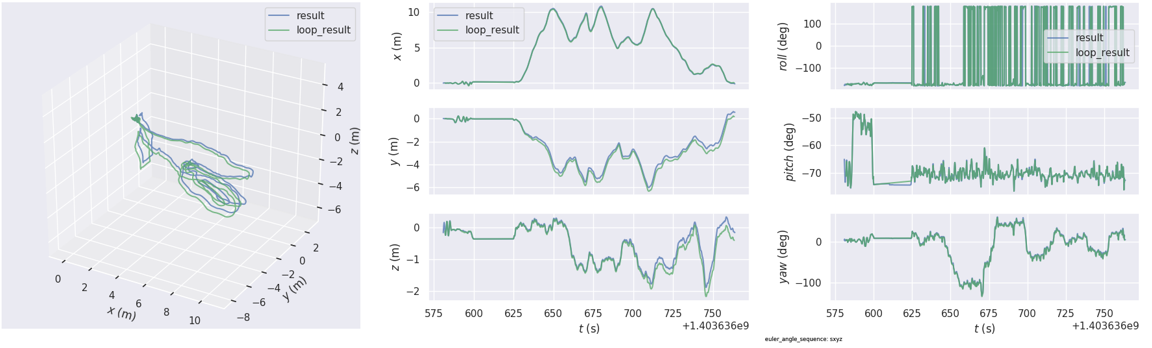
总共显示三张图片分别为轨迹、三个坐标轴的位置和三个坐标轴的姿态：



如果需要将估计轨迹与真实轨迹同时绘制，可采用指令：

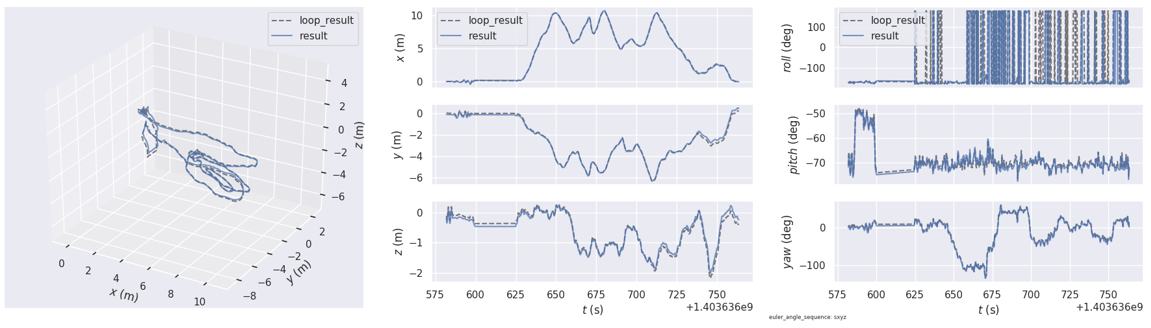
|  |
| --- |
| C++ evo\_traj tum result.csv loop\_result.csv -p |

然而存储时轨迹多为相对位置变化，所以绘制出的轨迹在初始位置上存在一定的位置和角度偏移。



这时我们采用对齐指令**将两条轨迹进行对齐**。为此需要通过**--ref参数**指定参考轨迹，并增加参数-a（或--align）进行对齐（旋转与平移）。

|  |
| --- |
| C++ evo\_traj tum result.csv --ref loop\_result.csv -p -a |



**二、实用命令之——evo\_ape**

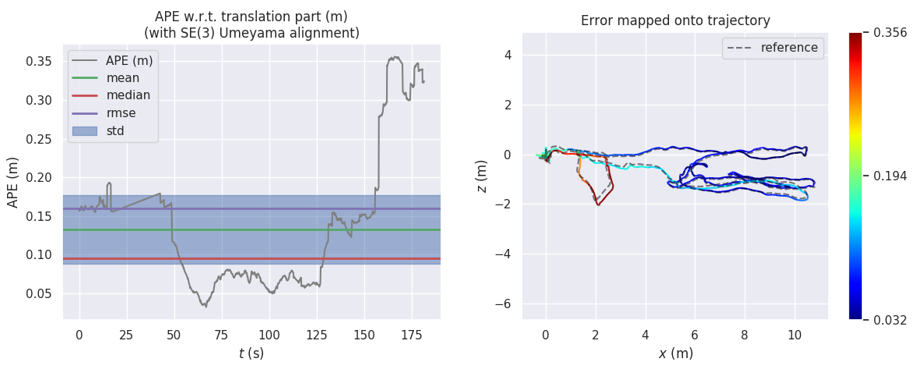
这是一个**评测轨迹绝对位姿误差**的工具，使用该命令比较tum数据集格式的result.csv和loop\_result.csv文件的绝对误差精度：

|  |
| --- |
| C++ evo\_ape tum result.csv loop\_result.csv -va --plot --plot\_mode xz --save\_results results/SPTAM.zip |

当使用上面的命令之后，会在终端中产生如下类型的结果：

max：最大误差；mean：平均误差；median：误差中位数；min：最小误差；rmse：均方根误差；sse：和方差、误差平方和；std：标准差。

并绘制下面的效果图：



当我们获得单目SLAM生成的轨迹文件时，此时我们需要和真实轨迹进行对比，但是我们知道**单目SLAM是没有尺度的，所以两个轨迹一定不会一样大**，所以我们需要对轨迹进行sim(3)相似变换，那么evo就会自动的对轨迹的位移、旋转、尺度进行对齐，完成这个任务的代码如下：

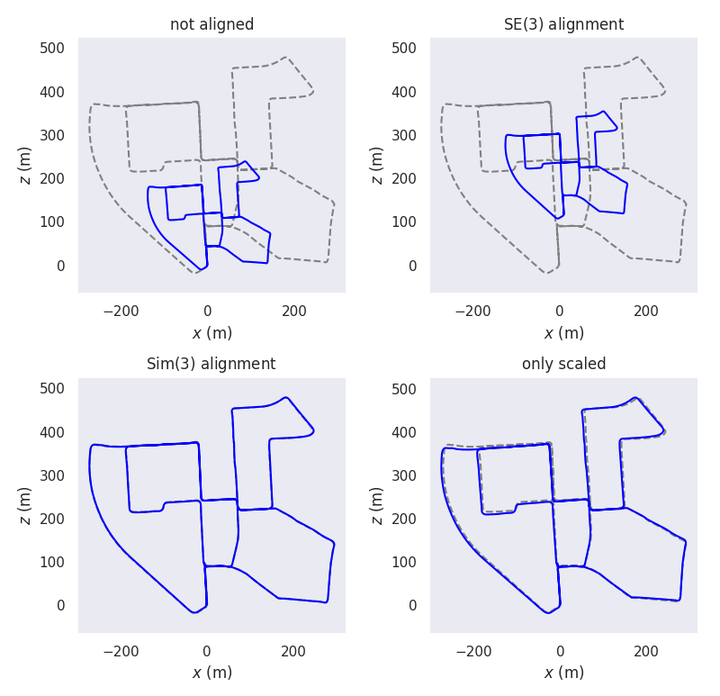
|  |
| --- |
| C++ evo\_ape tum data.tum CameraTrajectory.txt -a -p -s |

解释：参数-a、-s、-p即为选择的参数。-a、-s分别表示进行轨迹的匹配和尺度修正，以及-p表示画图误差的图和两个轨迹的图。

另外evo有一个比较好的功能，就是能在**单目SLAM或者单目视觉里程计中自动校准尺度**。TUM提供的工具需要手动输入。在命令后加--align --correct\_scale

|  |
| --- |
| C++ evo\_ape kitti KITTI\_00\_gt.txt KITTI\_00\_SPTAM.txt -va --plot --plot\_mode xz --save\_results results/SPTAM.zip --align --correct\_scale |

经过对齐和尺度校正与原始数据的对比如下图。



**三、实用命令之——evo\_res**

该命令用于比较两个轨迹之间的相对轨迹误差，也就是两个时间戳相同时刻的轨迹轨迹误差，更具体来说，两组轨迹持续时间相同，但是采样频率不同，那么之间就会有一些少量的相同时间戳，那么这些相同的时间戳就可以进行一对一比较误差。同样的这些误差也会有平均误差、最大、最小误差等等结果。命令使用方法如下：

|  |
| --- |
| C++ evo\_rpe xxx a.txt b.txt |

evo\_rpe命令依然可以添加一些可选参数项，你同样可以使用如下命令查看可选参数的具体介绍和用法：

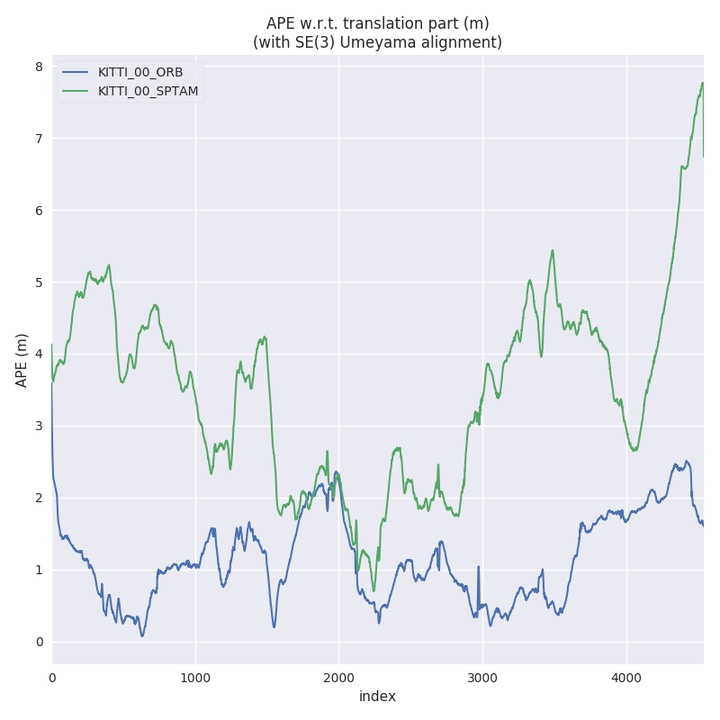
|  |
| --- |
| C++ evo\_rpe xxx -h |

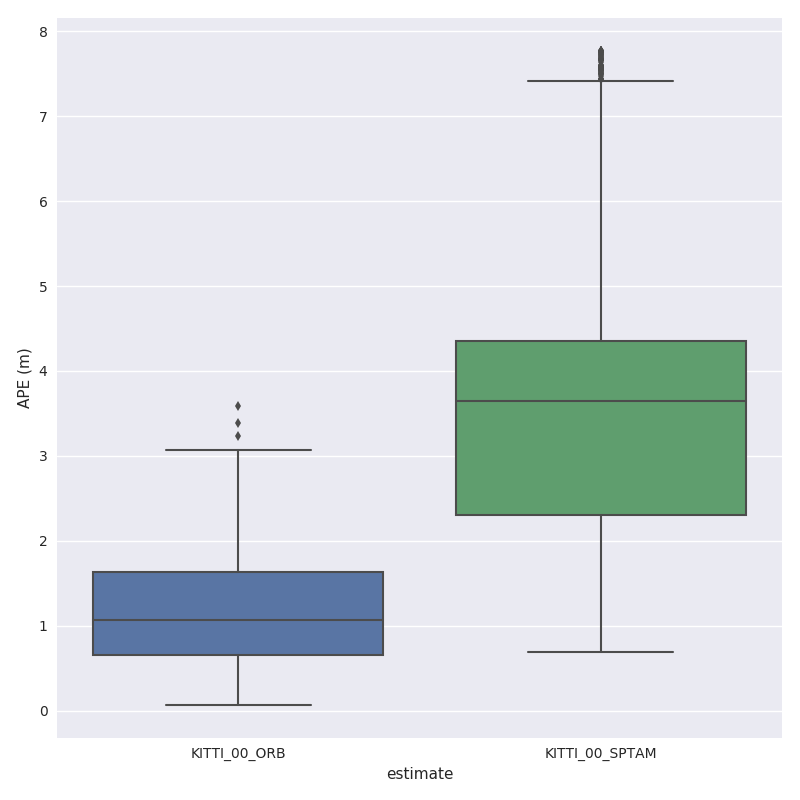
解释：其中xxx为tum、kitti、euroc、rosbag中的一种.

这个命令可以用来**比较多个结果文件**，打印统计信息，绘图，保存结果到表格等。命令用法举例如下：

|  |
| --- |
| C++ evo\_res results/\*.zip -p --save\_table results/table.csv |

命令效果如下图：





**四、实用命令之——evo\_config**

evo具有的参数和对应参数值

|  |
| --- |
| C++ {  "console\_logging\_format": "%(message)s",   "euler\_angle\_sequence": "sxyz",   "global\_logfile\_enabled": false,   "plot\_axis\_marker\_scale": 0.0,   "plot\_backend": "Qt5Agg",   "plot\_export\_format": "pdf",   "plot\_figsize": [  6,   6  ],   "plot\_fontfamily": "sans-serif",   "plot\_fontscale": 1.0,   "plot\_invert\_xaxis": false,   "plot\_invert\_yaxis": false,   "plot\_linewidth": 1.5,   "plot\_mode\_default": "xyz",   "plot\_multi\_cmap": "none",   "plot\_pose\_correspondences": false,   "plot\_pose\_correspondences\_linestyle": "dotted",   "plot\_reference\_alpha": 0.5,   "plot\_reference\_color": "black",   "plot\_reference\_linestyle": "--",   "plot\_seaborn\_palette": "deep6",   "plot\_seaborn\_style": "darkgrid",   "plot\_split": false,   "plot\_statistics": [  "rmse",   "median",   "mean",   "std",   "min",   "max"  ],   "plot\_texsystem": "pdflatex",   "plot\_trajectory\_alpha": 0.75,   "plot\_trajectory\_cmap": "jet",   "plot\_trajectory\_linestyle": "-",   "plot\_usetex": false,   "plot\_xyz\_realistic": true,   "ros\_map\_alpha\_value": 1.0,   "ros\_map\_unknown\_cell\_value": 205,   "save\_traj\_in\_zip": false,   "table\_export\_data": "stats",   "table\_export\_format": "csv",   "table\_export\_transpose": true,   "tf\_cache\_lookup\_frequency": 10,   "tf\_cache\_max\_time": 10000.0 } |

如果你想要对某一项参数进行修改，比如你想修改输出的图像格式，你可以使用如下命令：

|  |
| --- |
| C++ evo\_config set plot\_export\_format png |

**设置绘图的网格和背景：**默认情节设置有一个黑色的背景，为了方便印刷可以通过改变 Seaborn 的样式参数来改变背景为白色背景的网格

|  |
| --- |
| C++ evo\_config set plot\_seaborn\_style whitegrid |

**设置字体类型和比例**：图形中文本标签的相对大小也可以增加，以提高可读性。

|  |
| --- |
| C++ evo\_config set plot\_fontfamily serif plot\_fontscale 1.2 |

**参考轨迹的线条样式：**将画图所使用的线型改为 -

|  |
| --- |
| C++ evo\_config set plot\_reference\_linestyle - |

**设置默认图形大小：**将所画图的图像大小调整为10 9（宽 高）

|  |
| --- |
| C++ evo\_config set plot\_figsize 10 9 |

**坐标系标注：**坐标轴标记可以通过将地图坐标轴标记比例设置为非零值来激活。不需要设置为0即可。例如:

|  |
| --- |
| C++ evo\_config set plot\_axis\_marker\_scale 0.1 |

**恢复默认设置：**当你想要将配置还原为默认时，只需要使用命令：

|  |
| --- |
| C++ evo\_config reset |

**改变行宽：**为了匹配较小的字体，我们还缩小了行宽

|  |
| --- |
| C++ evo\_config set plot\_linewidth 1.0 |

**使用 LaTeX 渲染器：**使用 LaTeX 来写论文，所以使用 LaTeX 来渲染图中的字体:

|  |
| --- |
| C++ evo\_config set plot\_usetex |