# 2024年春季学期SLAM课程项目报告

## 1 项目描述

单目相机相比于立体相机系统或LiDAR等深度传感器，具有下面的优势：

- **低成本和广泛适用性：** 单目相机成本更低，重量更轻，适用于各种设备（如无人机、机器人和移动设备等），使得SLAM技术更具普遍适用性和经济可行性。
- **紧凑和灵活的系统设计：** 单目相机系统设计更为紧凑，易于集成到小型设备中。其灵活性使得SLAM技术可以应用在许多场景中，如狭窄空间或具有复杂几何结构的环境。
- **高分辨率和丰富信息：** 单目相机可以捕捉高分辨率的图像，提供丰富的视觉信息，有助于SLAM算法提取更多的特征点，进行更精确的地图构建和定位。

单目相机深度估计在SLAM中至关重要，因为它提供了一种低成本、高分辨率且紧凑的方法来实时估计相机的三维运动和场景深度信息，广泛应用于自动驾驶、机器人导航和增强现实等领域，极大地提升了环境感知和路径规划的精度与可靠性。

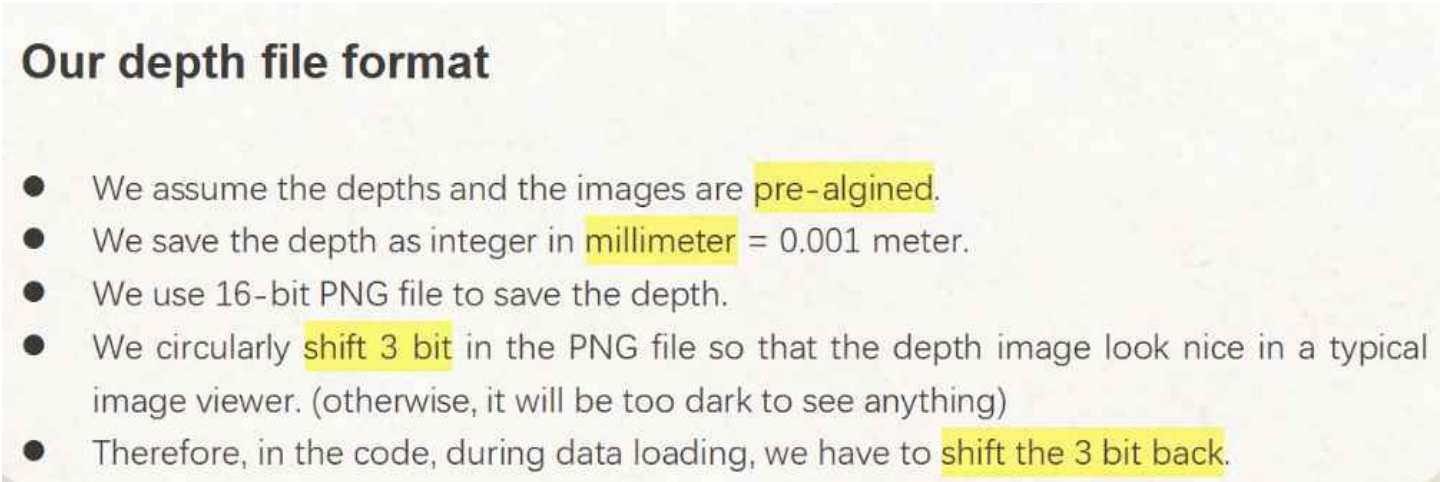
本项目必做部分利用ORB, SIFT以及SURF算子，估计Portland Hotel数据集中的图像深度，输出RMSE，ABS，RMSE log等实验指标。扩展部分复现DELTAS论文中的方法，验证论文中所实现的估计效果。

## 2 数据处理

### 2.1 数据集介绍

Portland Hotel数据集是SUN3D数据集的一部分，专注于三维重建和室内场景理解。SUN3D数据集由麻省理工学院发布，用于研究和开发室内环境中的三维重建、相机定位和场景理解等任务。Portland Hotel子集包含详细的三维数据和标注信息。

### 2.2 数据特征



- **预对齐的图像和深度：** 数据集中的图像和对应的深度数据已经过处理，使它们在空间上对齐。这意味着对于图像中的每个像素点，其在深度图像中有对应的深度值。
- **绝对的深度信息：** 深度值以实际物理单位mm表示，而不是相对或归一化的深度值。这种信息通常通过深度传感器或精确的几何计算获得。
- **移位和缩放操作：** 在处理深度图像时，Portland数据集把像素值左移了三位来让深度图像可见，所以在数据预处理过程中，我们需要将像素值首先右移三位，然后除以0.001，才能代表真实的以米为单位的深度。

```
# Example image and depth data loading
def load_images_and_depths(image_paths, depth_paths):
    images = [cv.imread(p) for p in image_paths]
    depths = [cv.imread(p, cv.IMREAD_UNCHANGED).astype(np.uint16) for p in depth_paths]
    depths = [(depth) * 0.001 for depth in depths]  # Adjust depth values
    return images, depths
```

## 3 对极约束方法

利用单目相机进行深度估计的常用方法如下：


算法流程
读取原始图像
读取深度图像
特征点识别与匹配
移位
对极约束求出基础矩阵F
缩放
计算本质矩阵E
SVD还原相机位姿
三角化方法求出深度信息
比较估计深度与真实深度
输出MAE，MSE，RMSE等特征

1. 读取原始图像

首先，从单目相机获取两张原始图像，这些图像是同一场景从不同角度拍摄的。通过这些图像，可以获取场景的视角信息。

2. **读取深度图像**

从数据集中读取与原始图像对应的深度图像，这些深度图像通常通过深度传感器或其他方法预先生成。这些深度图像为后续的深度估计提供了初始参考信息。

3. **特征点识别与匹配**

在两张图像中识别特征点，并进行匹配。常用的方法有SIFT、SURF等。这一步的目的是找到图像中相同位置的特征点，以便后续进行对极几何计算。

4. **计算基础矩阵和本质矩阵**

利用匹配的特征点，计算基础矩阵（Fundamental Matrix）和本质矩阵（Essential Matrix）。基础矩阵描述了两个视角之间的几何关系，而本质矩阵结合相机的内参进一步描述了相机的旋转和平移。

5. **SVD分解求解相机位姿**

通过对本质矩阵进行奇异值分解（SVD），求解出相机的相对位姿，即旋转矩阵（R）和平移向量（t）。这一步得到的结果是归一化的，不能直接反映实际距离尺度。

6. **三角化方法求出深度信息**

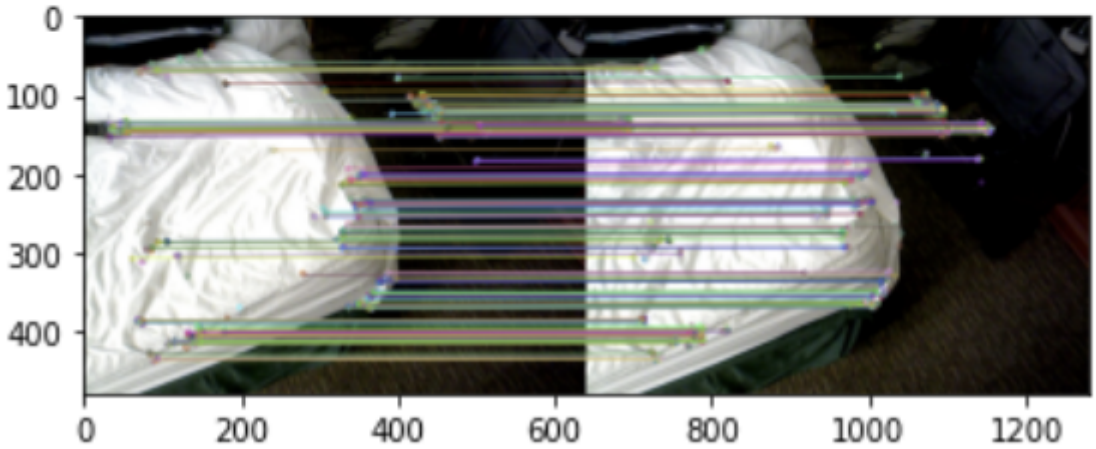
利用三角化方法，根据已知的相机位姿和匹配的特征点，计算出场景中点的深度信息。通过这一步，可以获得相对深度值，但由于缺乏绝对尺度，这些深度值仅是相对的。

7. **比较估计深度与真实深度**

将计算得到的深度值与真实深度图像中的深度值进行比较，评估深度估计的精度。常用的评估指标包括平均绝对误差（MAE）、均方误差（MSE）和均方根误差（RMSE）。

我尝试着用python代码构建了上述算法，实现了对Portland Hotel数据集的前两张图像进行深度估计，结果如下：

```
最小距离: 1.0
最大距离: 83.0
匹配数: 324
```




```
Rotation Matrix R:
[[ 9.99990867e-01  4.27376609e-03 -3.50000644e-05]
 [-4.27384282e-03  9.99987469e-01 -2.60703560e-03]
 [ 2.38577654e-05  2.60716138e-03  9.99996601e-01]]
Translation Vector t:
[[-0.16672418]
 [ 0.19679237]
 [ 0.96616552]]
```

```
MAE for image1 depth comparison: 381.54510165118535
MAE for image2 depth comparison: 381.5455446042055
```

可以看到，通过对极约束和三角化求出来的深度信息范围太大，并且正负值交替，结果根据特征点寻找的策略的不同而呈现很大的方差。正如前文第五步所说，单目相机的图像仅包含视角信息，而没有直接的深度信息。通过三角化方法，我们可以计算出视点之间的相对深度，但由于缺乏尺度参照，这

些深度值只是相对深度，通常被归一化，没有具体的单位。所以，直接用对极约束求出来的相机深度和真实深度进行比较，是根本没有意义的。

# 4 PnP方法

上述问题的根本原因是用了对极几何来估计相机位姿，联想到之前的作业2，如果已经有一部分的深度信息，那么就可以确定出尺度，计算得到的R&t误差也会更小。我们在这里不妨做出一个退步：假设第一张图片的深度信息是已知的，这下我们所求出来的R和t就是没有归一化之后的数据，能够继续通过三角测量的方法求出另一张图像的深度值了，于是，我想到了使用PnP的方法进行深度估计。

## 4.1 PnP方法原理

PnP方法是一种在计算机视觉中广泛应用的技术，用于求解相机位姿（位置和方向）。具体而言，PnP问题是在已知一组三维点在空间中的位置及其在图像平面上的投影位置的前提下，求解相机的外部参数，即旋转矩阵（R）和平移向量（t）。PnP方法的核心思想是通过最小化二维投影点与三维点的重投影误差，来精确估计相机的姿态。

在PnP方法中，我们需要有两组数据：一组是空间中的三维点，这些点的实际位置是已知的；另一组是这些三维点在图像平面上的二维投影点，通过图像处理算法如特征检测和匹配获得。这些点对提供了从图像坐标系到空间坐标系的映射关系。

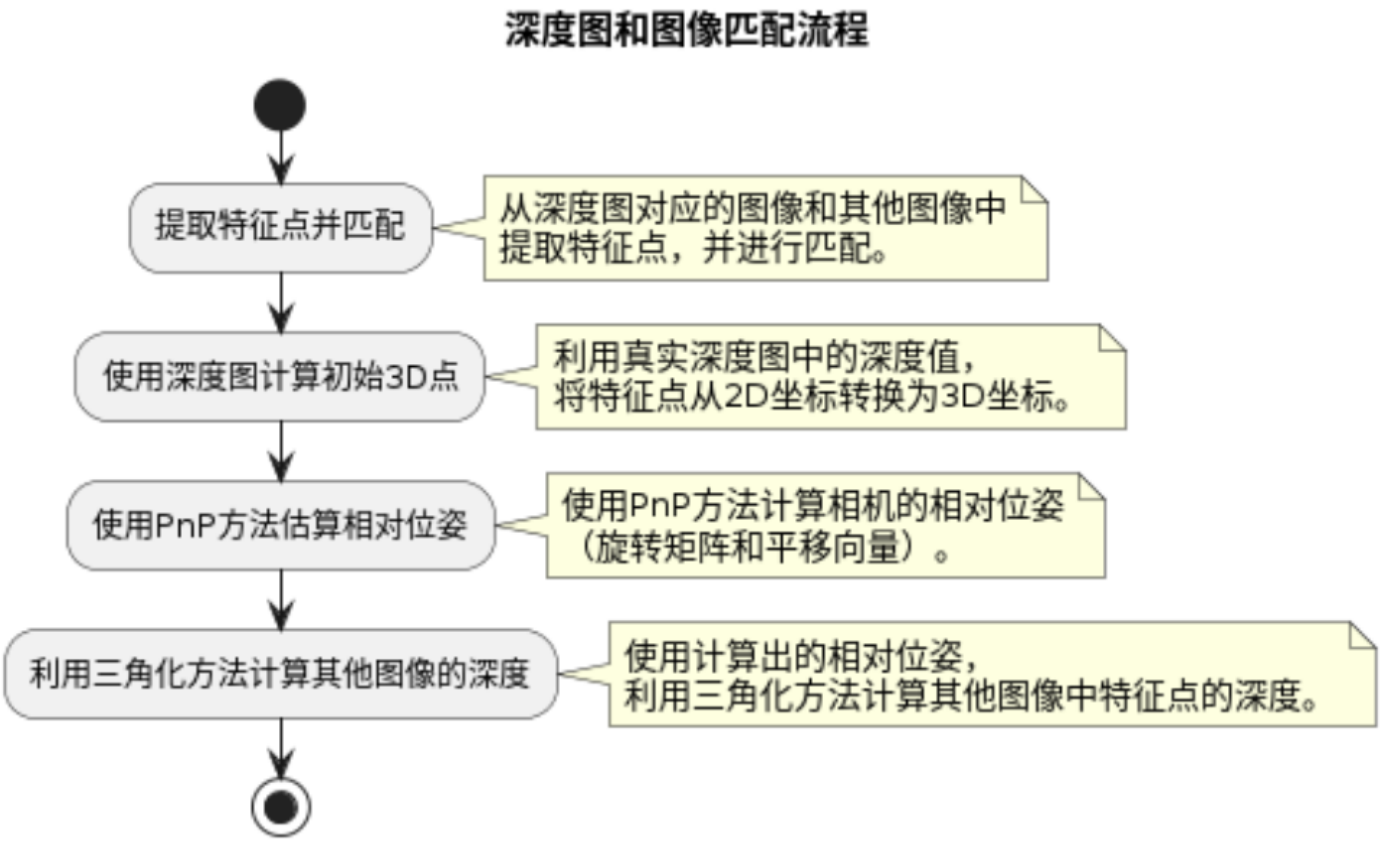
通过最小化误差函数，PnP方法利用这些已知的三维点和其在图像中的投影关系，来精确计算相机的位姿。这不仅解决了对极几何方法中深度值归一化的问题，还能提供绝对尺度，使得计算得到的深度信息更加准确和鲁棒。

## 4.2 PnP方法的优势

- **引入绝对尺度：** 假设第一张图像的深度信息已知，PnP方法利用这些已知深度信息来确定相机的位姿，从而引入绝对尺度。这解决了对极几何方法中深度值归一化的问题。
- **精确的位姿估计：** 通过最小化重投影误差，PnP方法能够更准确地估计相机的位姿（R和t）。这减少了由对极几何方法带来的误差。
- **稳定性和鲁棒性：** PnP方法在有部分已知深度信息的情况下，能够更加稳定和鲁棒地进行深度估计，避免了对极几何方法中因特征点选择不同导致的方差较大的问题。

# 5 必做部分实验结果

更新之后的算法框架如下图所示：




1. **提取特征点并匹配**

首先，从深度图对应的图像和其他图像中提取特征点，并进行匹配。这一步的目的是在不同图像之间找到对应的特征点，以便后续进行三维重建和位姿估计。

2. **使用深度图计算初始3D点**

利用真实深度图中的深度值，将特征点从二维坐标转换为三维坐标。这一步将二维图像中的特征点转换为三维空间中的点，为后续的PnP算法提供基础数据。

3. **使用PnP方法估算相对位姿**

利用PnP方法计算相机的相对位姿（旋转矩阵R和平移向量t）。PnP方法通过最小化三维点投影到二维图像的误差，精确估计出相机的外部参数。

4. **利用三角化方法计算其他图像的深度**

使用计算出的相对位姿，通过三角化方法计算其他图像中特征点的深度。通过已知的相机位姿和匹配特征点的位置，可以求解出这些特征点在三维空间中的深度信息。

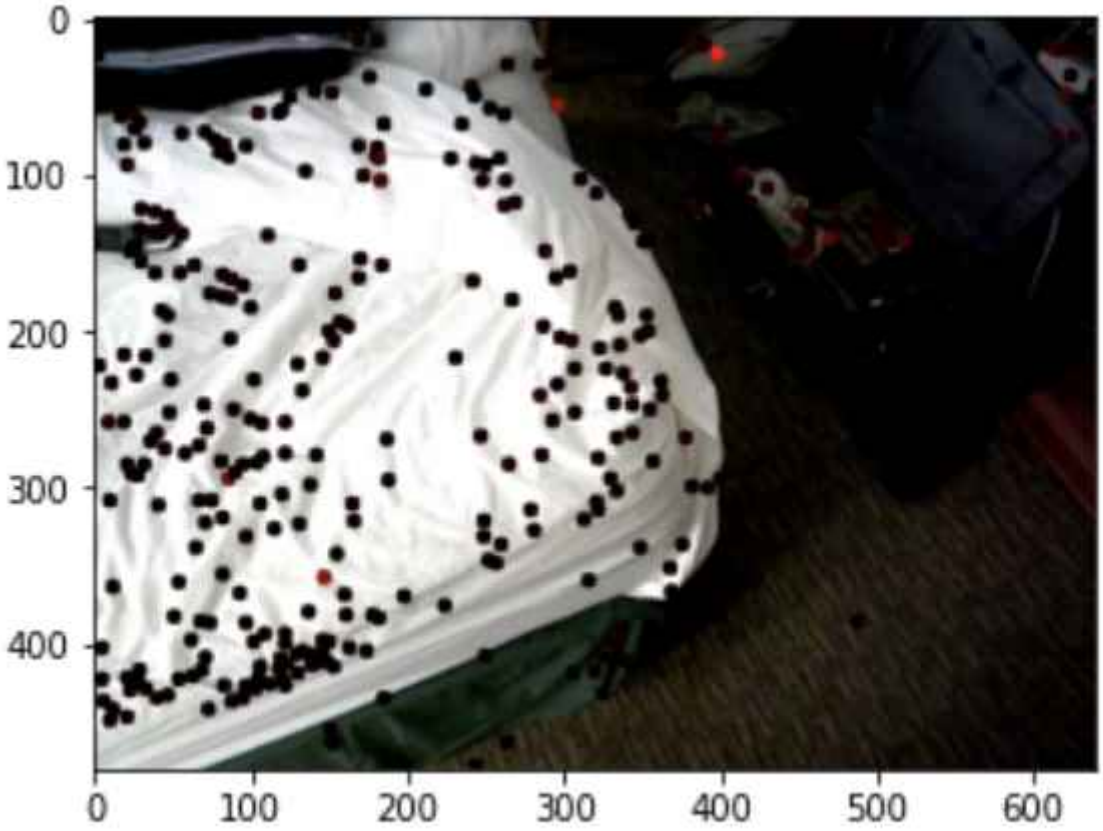


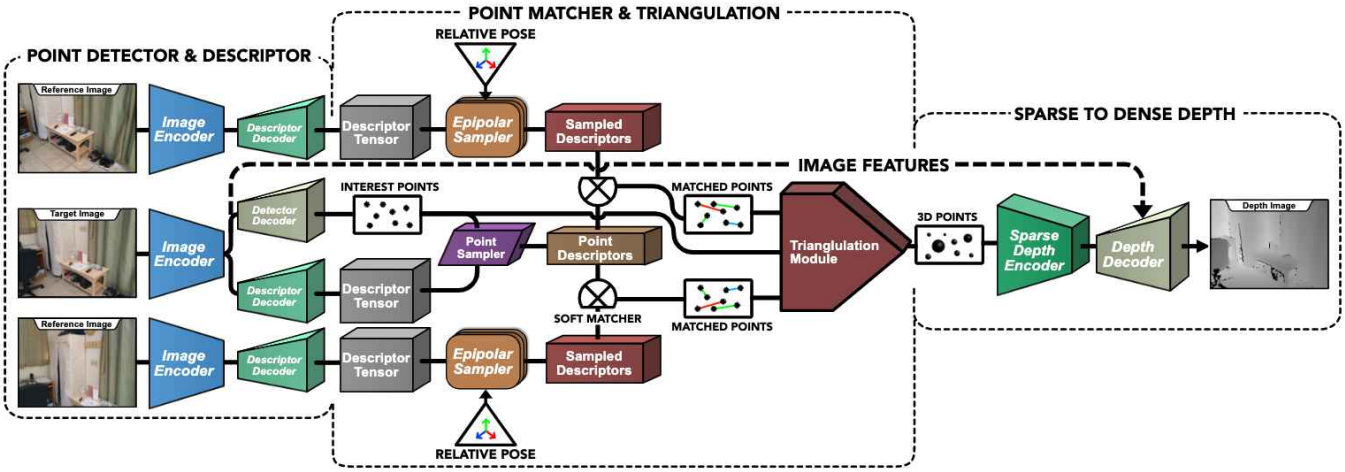

表 4: Comparison of SIFT, ORB, and SURF

| | SIFT | ORB | SURF |
|---|---|---|---|
| **Good Matches (FLANN)** | 358 | 313 | 374 |
| **Absolute Error (Abs)** | 0.5159 | 0.8000 | 0.3660 |
| **RMSE** | 1.4084 | 2.2932 | 0.7930 |
| **RMSE log** | 0.3021 | 0.4090 | 0.2369 |

上述三种方法都使用FLANN搭配Lowe's Ratio Test来匹配筛选特征点，SURF由于专利保护，降级使用python3.6 + OpenCV 3.4.2.16版本运行。

## 6 扩展部分概述

对于实验的扩展部分，我首先阅读了DELTAS: Depth Estimation by Learning Triangulation And densification of Sparse points这篇论文，对模型架构有了大概的认识。




**第一步：特征点检测和描述子生成**

首先，将目标图像和多视角图像输入到共享的RGB编码器和描述子解码器中，生成每张图像的描述子场。对于目标或锚点图像，还会检测兴趣点。这一步的输出包括每张图像的描述子张量和目标图像中的兴趣点。

**第二步：特征点匹配和三角化**

在第二步中，利用目标图像中的兴趣点和相对位姿确定参考或辅助图像中的搜索空间。在搜索空间中对描述子进行采样，并将这些描述子与兴趣点的描述子进行匹配。然后，使用奇异值分解（SVD）对匹配的关键点进行三角化，输出三维点，并创建稀疏深度图像。

**第三步：稀疏到稠密深度估计**

在最后一步中，将稀疏深度编码器的输出特征图和RGB编码器的中间特征图共同输入深度解码器，生成稠密深度图像。通过这一步，稀疏的三维点被密集化，从而生成完整的深度图。

通过以上三个步骤，算法在保持计算效率的同时，实现了高精度的深度估计。这种方法通过结合2D图像和3D几何监督，以及深度监督，形成了端到端的网络框架，提供了精确且鲁棒的深度估计结果。

## 7 扩展部分实验结果

接下来我找到了网上给出的对应github代码，虽然模型结构已给出，但是代码里并没有模型训练相关的代码，所以我就下载了预训练好的模型，在测试集上对模型进行测试，并更改为输出ABS，MAE，RMSE，RMSE log等指标。结果如下：

```
(base) root@autodl-container-a1fc46a083-a15c31f3:~/autodl-tmp/ML_Project-main/SLAM/DELTAS# python test_learnabledepth.py
=> fetching scenes in './assets/sample_data/scannet_sample'
3 samples found in 1 valid scenes
=> creating model
/root/miniconda3/lib/python3.8/site-packages/torch/functional.py:568: UserWarning: torch.meshgrid: in an upcoming release,
ent. (Triggered internally at  ../aten/src/ATen/native/TensorShape.cpp:2228.)
  return _VF.meshgrid(tensors, **kwargs)  # type: ignore[attr-defined]
TEST: Depth Error Abs Rel 0.0766 (0.1429), MAE 0.1429, RMSE 0.2113, RMSE log 0.1205
```

可以看出测试集上的结果与论文里给出的结果非常近似，由此得以验证。

但是现在我们只知道用DELTAS的实验结果很好，无法和我们上面所用的PnP的方法做比较，经过我的尝试，我基于ScanNet数据集，将上面的PnP方法也应用于深度估计，从而与DELTAS方法形成对比。实验结果如下：

表 4: Comparison of Feature Detectors

| | SIFT | ORB | SURF | DELTAS |
|---|---|---|---|---|
| **Good Matches (FLANN)** | 210 | 279 | 156 | / |
| **Absolute Error (Abs)** | 1.0528 | 0.6785 | 0.9342 | 0.1429 |
| **RMSE** | 1.9587 | 0.9481 | 3.4374 | 0.2113 |
| **RMSE log** | 0.4491 | 0.2505 | 0.3320 | 0.1205 |

可以看出DELTAS方法无论是哪一种指标，都要好于基于PnP的三种方法，由此结果得以对比验证。

## 8 心得体会

在本次项目中，我探索了单目相机深度估计的不同方法，经历了从对极约束到PnP方法的转变，并最终验证了DELTAS方法的优越性。通过这些方法的应用和比较，我对深度估计技术有了更深入的理解和认识。

在对极约束方法的应用中，我发现由于单目相机的图像缺乏直接的深度信息，导致计算出的深度值范围过大且存在正负值交替的问题。这使得估计结果不够稳定和精确。为了改进这一问题，我引入了PnP方法，通过已知的部分深度信息来确定相机的位姿，解决了深度值归一化的问题，显著提高了深度估计的精度和稳定性。

在扩展部分，我复现了DELTAS方法，并将其与PnP方法进行了对比。实验结果表明，DELTAS方法在各项指标上均优于基于PnP的三种方法。这一结果验证了DELTAS方法的优越性，也为我的研究提供了宝贵的对比数据。

通过这些探索和实验，我不仅学会了如何处理数据、实现算法，还提升了我的实验设计和结果分析能力。本次项目的经验将对我未来在SLAM技术和深度估计领域的研究和应用产生积极的影响。