

V1版软件代码思路与效果展示

一.代码思路

1. 图形用户界面（GUI）结构

该应用的图形界面是基于 Tkinter 实现的，它提供了多个控件来帮助用户与程序交互。



主要界面分为两个部分：

- **左侧框架：**用于图像加载、选择和管理图像。
 - 包含按钮 ("添加图像") 用于加载图像，`Listbox` 用于展示已加载的图像，`Scrollbar` 用于滚动查看图像列表。
 - 还包括选择图像 A 和图像 B 的两个 `Listbox`，这两个列表框帮助用户选择要处理的图像对。
- **右侧框架：**用于展示功能按钮和图像预览。
 - 功能按钮包括：
 - **计算并可视化双机相对位姿：**计算并绘制两台相机之间的相对位姿（旋转和平移）。
 - **双视几何：**显示几何概念图和绘制对极线。
 - **执行双目立体校正：**进行图像校正，使得图像中的对应点位于相同的水平线上。
 - 预览区域显示用户选择的图像 A 和图像 B 的缩略图。

2. 图像加载与处理

2.1 图像加载：

- 用户通过点击“添加图像”按钮加载图像文件，代码通过 `filedialog.askopenfilenames()` 打开文件选择对话框，用户可以选择一张或多张图像文件。
- 加载的图像会存储在 `self.image_paths`、`self.image_cv_originals`（原始图像）和 `self.image_cv_grays`（灰度图像）中。

2.2 计算相机内参矩阵：

每加载一张图像，代码会基于图像的尺寸（宽度和高度）估算一个内参矩阵 K ，这里使用的估算方法比较简化：

- **焦距**：通过图像的宽度和高度来简单估算，通常为图像尺寸的最大值。
- **光心**：假设图像的光心位于图像的中心。

内参矩阵形式为：

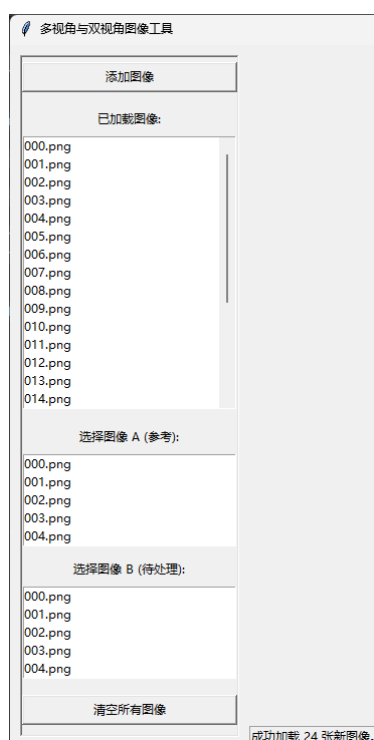
$$K = \begin{bmatrix} f & 0 & c_x \\ 0 & f & c_y \\ 0 & 0 & 1 \end{bmatrix}$$

其中， f 为焦距， c_x, c_y 为图像中心（即光心）的坐标。

2.3 特征检测：

每加载一张图像，程序都会调用 `_detect_features_for_image()` 方法，使用 **SIFT** 或 **ORB** 算法进行特征检测和描述符计算，提取图像中的关键点（keypoints）和描述符（descriptors）。这些特征点将用于后续的图像匹配和位姿计算。

加载完的图片如图所示：



3. 相机相对位姿可视化

3.1 计算相机相对位姿：

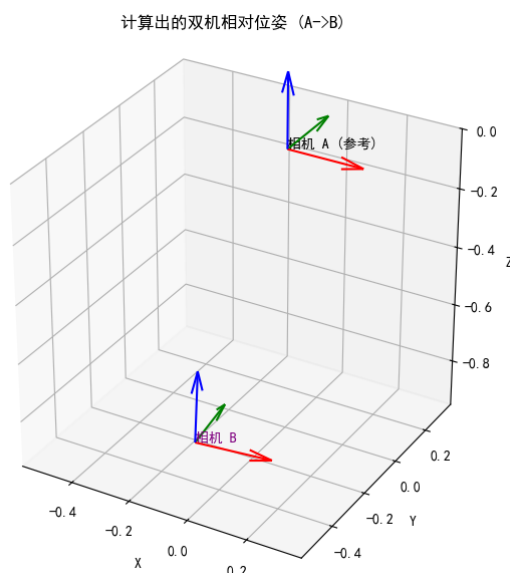
当用户选择两张图像时，程序会：

- 调用 `_get_common_inliers()` 方法，计算两张图像中的匹配点，进而计算出 **基础矩阵** FFF。
- 通过基础矩阵和相机内参矩阵，计算出 **本质矩阵** EEE。
- 使用 `cv2.recoverPose()` 从本质矩阵恢复出两台相机之间的 **旋转矩阵** RRR 和 **平移向量** ttt，这些信息表示了两台相机之间的相对位姿。

3.2 三维可视化：

然后，通过 **Matplotlib** 和 **3D 图形库 Axes3D**，将计算出的相机位姿绘制在三维空间中。两台相机的坐标系会在同一三维图形中显示，帮助用户理解它们之间的位置和方向关系。

实现效果如图：



4. 双视几何与对极线绘制

4.1 双视几何概念图：

双视几何描述了两台相机之间的几何关系，核心是 **对极线** 和 **极点**：

- **对极线**：给定一张图像中的点，另一个图像中对应点必定位于一条对极线上。
- **极点**：是两台相机的光心在另一台相机图像中的投影点。

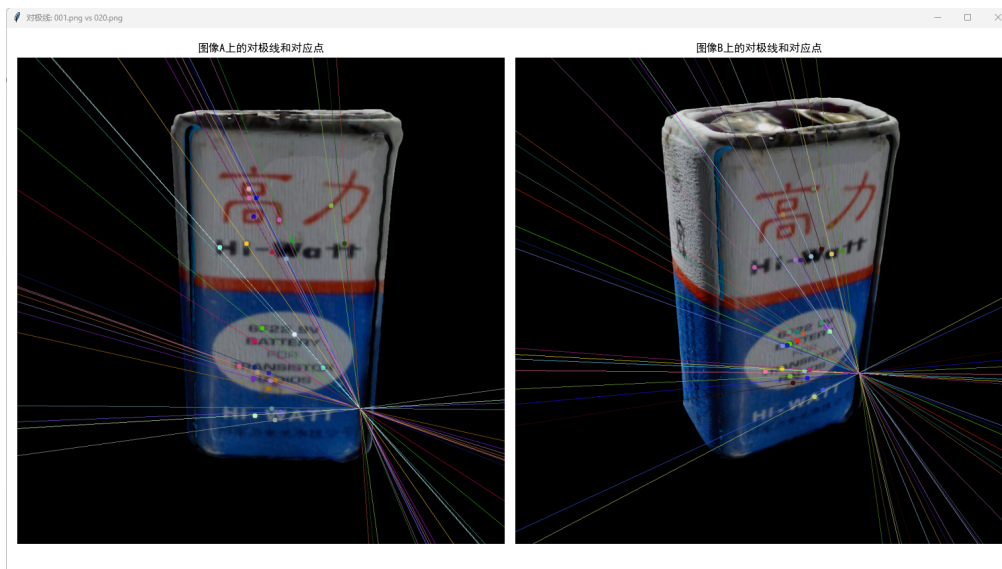
程序通过 `show_epipolar_geometry_concept()` 方法绘制简化版的双视几何示意图，包括相机中心、图像平面、3D点、极点、对极线等概念。

4.2 绘制图像对极线：

当用户选择了两张图像时，程序会计算出 **基础矩阵** F，并利用

`cv2.computeCorrespondEpilines()` 函数计算出对极线，然后在两张图像上绘制出这些对极线。

实现效果如图：



5. 双目立体校正

立体校正 (Stereo Rectification) 是使两张图像的视角对齐的过程。目标是将两幅图像的对应点调整到同一水平线上，从而简化后续的深度计算。

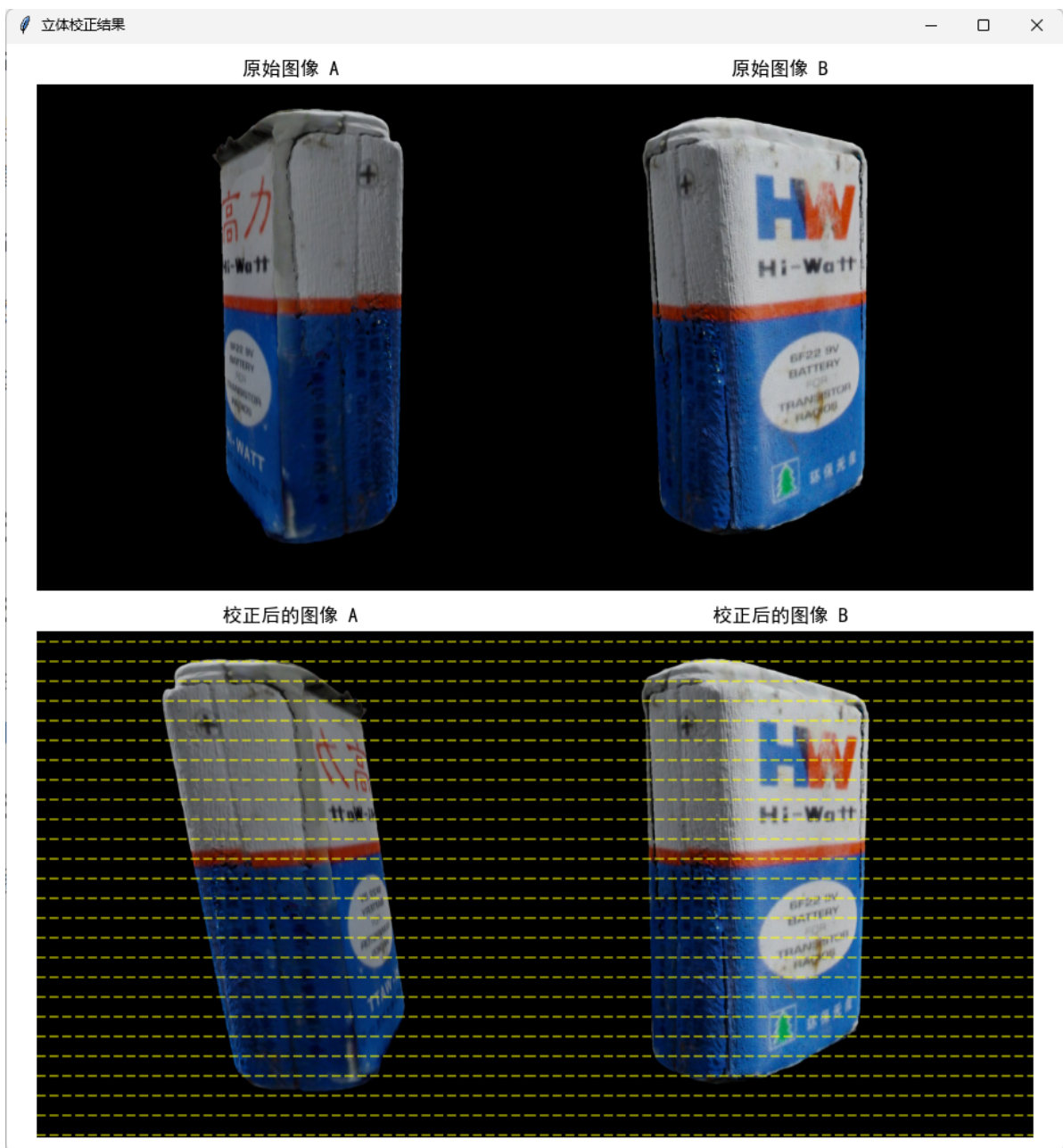
5.1 计算变换矩阵：

程序通过 `cv2.stereoRectifyUncalibrated()` 函数计算两张图像的 **校正矩阵** H_1 和 H_2 ，这些矩阵用于将图像 A 和图像 B 校正到同一平面上。

5.2 应用变换：

然后，程序通过 `cv2.warpPerspective()` 函数应用这些变换矩阵，将两张图像校正，使得对应点在两幅图像中对齐。

实现效果如图：



6. 图像预览与交互

6.1 图像预览：

用户可以在左侧选择已加载的图像，并在右侧预览这些图像的缩略图。每当用户选择一张图像时，预览区域会更新显示该图像的缩略图。

6.2 交互功能：

通过图形界面，用户可以：

- 加载、选择和清空图像。
- 执行图像处理操作，如计算相机相对位姿、绘制对极线、执行立体校正等。
- 查看图像处理结果的三维可视化和图像展示。