**MaixSense立体标定标准作业流程 (SOP)**

**1. 目的**

本标准作业流程 (SOP) 旨在规范使用 MaixSense 深度摄像头进行 RGB 和 IR 传感器双目立体标定的操作过程，以获取精确的相机内参、畸变系数以及两个传感器之间的相对位置关系（外参），为后续的三维重建、深度数据映射、机器人视觉等应用提供准确的基础数据。

**2. 范围**

本 SOP 适用于 MaixSense 系列深度摄像头，使用棋盘格标定板和配套的 Python 脚本 (stereo\_calibrate\_interactive.py, analyze\_stereo\_error.py, recalibrate\_stereo.py) 进行立体标定。

**3. 职责**

* **操作员/工程师:** 负责执行本 SOP 中的所有步骤，确保标定过程的准确性和结果的有效性。

**4. 定义与缩写**

* **内参 (Intrinsics / M1, M2):** 相机内部参数，包括焦距 (fx, fy) 和主点 (cx, cy)，描述了 3D 点如何投影到相机 2D 图像平面。通常表示为 3x3 矩阵。
* **畸变系数 (Distortion Coefficients / d1, d2):** 描述相机镜头光学畸变的参数，包括径向畸变 (k1, k2, k3, ...) 和切向畸变 (p1, p2)。通常表示为 1xN 向量。
* **外参 (Extrinsics / R, T):** 描述两个相机坐标系之间相对姿态的参数。
  + **旋转矩阵 (R):** 3x3 矩阵，描述从 IR 坐标系到 RGB 坐标系的旋转。
  + **平移向量 (T):** 3x1 向量，描述 IR 坐标系原点相对于 RGB 坐标系原点的位置（在 RGB 坐标系中表示），单位通常为毫米 (mm)。
* **RMS 重投影误差 (RMS Reprojection Error):** 衡量标定精度的指标，表示根据标定参数将 3D 点投影回 2D 图像后，投影点与实际检测点之间的平均距离（像素）。值越低越好，理想情况在 1.0 像素左右。
* **棋盘格 (Chessboard):** 用于标定的黑白相间网格图案。
* **立体标定 (Stereo Calibration):** 同时确定两个或多个摄像头的内部参数和它们之间的相对位置关系的过程。
* **图像对 (Image Pair):** 同一时刻或尽可能接近时刻捕获的 RGB 图像和 IR 图像。
* **SOP:** 标准作业流程 (Standard Operating Procedure)。

**5. 所需材料与设备**

* MaixSense 深度摄像头及配套电源、网络连接线。
* 已安装 Python 环境 (推荐 3.6+) 及所需库 (opencv-python, numpy, requests) 的计算机。
* 标定脚本：
  + stereo\_calibrate\_interactive.py (用于捕获图像和执行主标定)
  + analyze\_stereo\_error.py (用于分析误差，可选但推荐)
  + recalibrate\_stereo.py (用于重新标定，可选但推荐)
* **高品质棋盘格标定板:**
  + **要求:** 必须是**刚性、绝对平整**的。**强烈不推荐**直接使用 A4 纸。建议将高质量黑白打印的棋盘格完美粘贴在厚玻璃板、厚亚克力板或其他已知平整的刚性基板上。
  + **已知参数:** 精确测量并记录棋盘格内部角点的列数 (cols)、行数 (rows) 以及每个方块的边长 (size, 单位 mm)。
* **稳定的相机支架:** 如三脚架，用于在捕获图像时固定相机，避免抖动。
* **稳定的标定板支架/放置表面:** 用于在不同姿态下稳定放置标定板。
* **(推荐) 均匀可控的光源:** 如可调节角度和亮度的台灯，用于优化照明条件，特别是改善 IR 图像质量。避免产生眩光和反射。

**7. 操作流程**

**7.1 准备工作**

1. **环境设置:** 选择一个光照均匀、背景相对简洁的区域。使用台灯等辅助光源优化棋盘格照明，确保在 RGB 和 IR 预览中均清晰可见，无明显反光、阴影或过曝/欠曝。
2. **软件准备:** 确保计算机已安装 Python 和所需的库 (pip install opencv-python numpy requests)。将标定脚本放置在工作目录中。
3. **标定板准备:** 确认标定板绝对平整、干净。精确测量并记录其参数 (cols, rows, size)。
4. **相机连接:** 连接 MaixSense 相机电源和网络，确保相机在网络中可访问，并记录其 IP 地址。
5. **创建目录:** 在计算机上创建一个用于保存标定图像和结果的文件夹 (例如 my\_stereo\_calib)。

**7.2 图像对捕获 (使用 stereo\_calibrate\_image.py)**

1. **启动脚本:** 打开终端或命令提示符，导航到脚本所在目录，运行以下命令 (替换 <...> 为实际参数):
2. python stereo\_calibrate\_interactive.py --host <相机IP> --cols <内角点列数> --rows <内角点行数> --size <方块尺寸mm> --num\_pairs <目标图像对数> --outdir <输出目录> --ir\_bit <0或1>

* + --ir\_bit 1 通常是推荐的 8 位模式。
  + --num\_pairs 建议设置在 25-40 之间。

1. **检查恢复:** 脚本会检查输出目录中是否已有图像，并提示是否从上次中断处继续。
2. **配置相机:** 脚本会自动配置相机以同时输出 RGB 和 IR 图像。
3. **捕获循环:**
   * 两个预览窗口 (RGB Feed, IR Feed) 会打开，显示实时图像和捕获进度。
   * 将标定板放置在两个摄像头都能看到的区域。
   * **姿态多样性:** 变换标定板的位置和角度，确保：
     + 覆盖视野的中心、边缘和角落。
     + 包含不同的距离（远、中、近）。
     + 包含不同的倾斜角度（前后、左右显著倾斜）。
   * **按 'c' 捕获:** 当棋盘格在两个预览中都清晰可见且姿态满意时，按下键盘上的 'c' 键。
   * **角点检测:** 脚本会尝试在 RGB 和 IR 图像中同时检测角点。
   * **交互确认/删除:**
     + 如果**成功**在两个图像中找到角点，脚本会暂停并在图像上绘制角点，同时显示提示 "Accept: [c] or [Enter] | Delete: [d]"。
     + **仔细检查**两个预览窗口中的图像质量和角点绘制情况。如果满意，按下 **'c'** 或 **Enter** 键接受。图像对将被保存，计数器增加。
     + 如果对当前这对图像不满意（例如模糊、光照差、角点检测看似有误），按下 **'d'** 键删除，然后重新调整姿态并按 'c' 再次尝试捕获**当前编号**的图像对。
     + 如果**未能**在两个图像中都找到角点，脚本会提示并在短暂延迟后继续，你需要调整姿态后再次按 'c'。
   * **按 'q' 退出:** 在捕获循环中可随时按 'q' 键中止捕获。
4. **完成捕获:** 当成功捕获并接受了指定数量 (--num\_pairs) 的图像对后，捕获循环结束，窗口关闭。

**7.3 执行立体标定计算**

1. **自动执行:** 图像捕获完成后，stereo\_calibrate\_interactive.py 脚本会自动使用收集到的有效图像对数据执行立体标定计算。
2. **计算过程:** 包括可选的初始单目标定和核心的 cv2.stereoCalibrate 调用。
3. **输出结果:**
   * 计算过程和结果（RMS 误差、内外参、畸变系数、R、T）会打印在终端上。
   * 最终的标定结果会保存到一个 .npz 文件中 (例如 <输出目录>/stereo\_calibration\_data.npz)。该文件包含了所有重要的标定参数。

**7.4 (可选但推荐) 误差分析与筛选 (使用 analyze\_stereo\_error.py)**

1. **目的:** 检查标定结果的质量，识别可能导致误差过高的不良图像对。
2. **运行脚本:** （前提是你的主标定脚本保存了 rvecs, tvecs）
3. python analyze\_stereo\_error.py --calib\_file <标定结果.npz文件路径> --img\_dir <图像保存目录> --cols <内角点列数> --rows <内角点行数> --size <方块尺寸mm>

1. **分析输出:**
   * 脚本会列出每对图像的角点查找情况和（如果可能）计算出的重投影误差。
   * 注意观察误差显著高于平均值的图像对编号。
   * **手动检查:** 打开这些高误差图像对对应的 pair\_XX\_rgb.png 和 pair\_XX\_ir.png 文件，仔细查找质量问题（模糊、光照、平面性等）。
   * 记录需要排除的图像对编号。

**7.5 (可选) 重新标定 (使用 recalibrate\_stereo.py)**

1. **目的:** 基于误差分析的结果，排除不良图像对后重新计算标定参数，以期获得更低的 RMS 误差和更准确的结果。
2. **运行脚本:**
3. python recalibrate\_stereo.py --img\_dir <图像保存目录> --cols <内角点列数> --rows <内角点行数> --size <方块尺寸mm> --ir\_bit <0或1> --exclude <编号1> <编号2> ... --out\_file <新结果文件名.npz>

* + 使用 --exclude 参数指定要排除的图像对编号。
  + 使用 --out\_file 指定一个新的文件名保存重新标定的结果。

1. **评估新结果:** 比较新的 RMS 误差和参数。如果改进显著且满意，则使用新的结果文件。如果改进不大或仍然不满意，可能需要重复步骤 7.4 和 7.5，或者回到步骤 7.1/7.2 重新捕获更高质量的图像。

**8. 结果解释与使用**

标定成功后，最重要的结果保存在 .npz 文件中：

* mtx1, dist1: RGB 相机的内参矩阵和畸变系数。
* mtx2, dist2: IR 相机的内参矩阵和畸变系数。
* R, T: 从 IR 相机坐标系到 RGB 相机坐标系的旋转矩阵和平移向量。
* error: 最终的 RMS 重投影误差。
* rgb\_size, ir\_size: 标定时使用的图像尺寸。
* (可选) rvecs, tvecs: 每次拍摄时棋盘格相对于 RGB 相机的姿态。

**使用:**

* **图像去畸变:** 使用 cv2.undistort() 配合对应相机的 mtx 和 dist 可以校正图像畸变。
* **立体校正 (Stereo Rectification):** 使用 cv2.stereoRectify() 结合所有参数 (M1, d1, M2, d2, R, T, imageSize) 可以计算校正变换矩阵，用于将左右图像对准，使得对应点位于同一水平线上，便于深度计算。
* **点坐标变换:** 使用 R 和 T 可以将在一个相机坐标系下测得的点转换到另一个相机坐标系。
* **深度图与彩色图对齐:** 结合内参、外参和畸变系数，可以将深度图中的点精确地投影到去畸变后的 RGB 图像上。

**9. 故障排除**

* **RMS 误差过高 (>2.0):**
  + 检查图像质量：是否存在模糊、光照不均、反光？IR 图像对比度是否足够低？
  + 检查标定板：是否绝对平整？尺寸参数是否准确？
  + 检查图像多样性：是否包含足够多的不同角度、距离和位置的图像？
  + 执行误差分析和筛选步骤 (7.4, 7.5)。
* **无法检测到角点:**
  + 检查光照和对比度。
  + 确保棋盘格完整且在视野内。
  + 检查棋盘格图案是否清晰。
  + 对于 IR 图像，检查红外散斑是否严重干扰。尝试使用可见光辅助照明。
* **脚本报错:**
  + 检查 Python 环境和库是否正确安装。
  + 检查命令行参数是否正确提供。
  + 检查相机网络连接是否正常。