Part I

Part Title

1

传感器标定

LiDAR 和相机联合标定是确定车载 LiDAR 和相机间的刚体变换关系，对齐 3D 点云和 2D 像素，为多传感器数据融合提供基础。标定的关键是寻找在 LiDAR 和相机视野内的匹配模式。考虑到标定的可操作性和精确度，以及后续标定误差的矫正，我们首先采用传统的基于平面靶的标定方法，快速获取 6DOF 外参；然后基于边缘匹配在线更新外参，获取更高精度的外参。

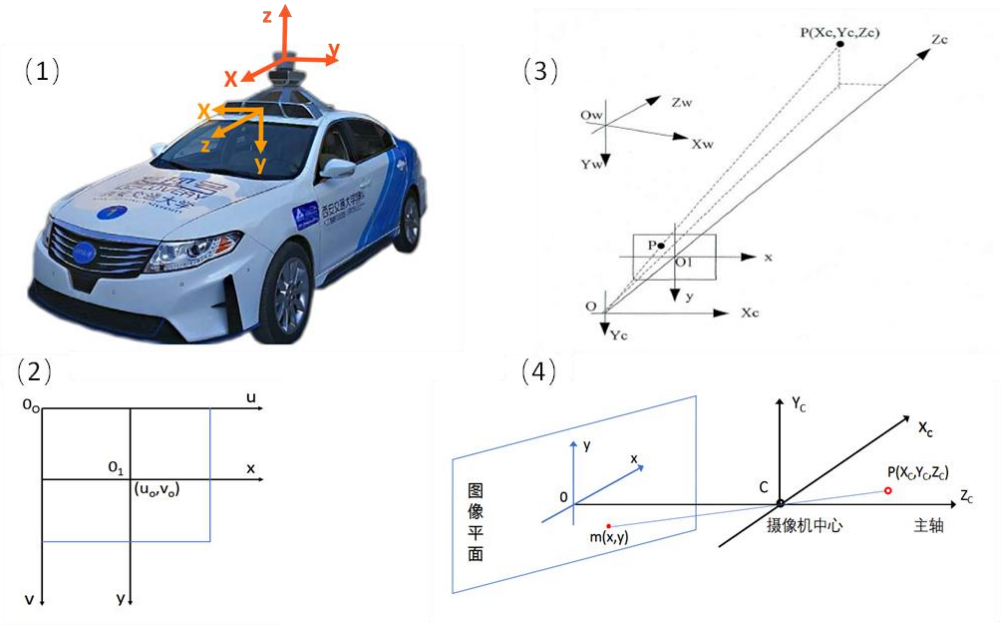


图 1.1. (1)LiDAR 和相机坐标系 (2) 像平面坐标系 (3) 成像模型 (4) 图像坐标系与相机坐标系

1.1 LiADR-Camera 联合粗标定

为描述车载传感器的相对位置关系，建立如图?? (1) 坐标系（所有坐标系均为右手系），并引入中间坐标系 (世界坐标系) 来建立相机内标定和 LiDAR-Camera 系统外标定间的联系。其中 Velodyne

64HDL LiDAR 坐标系定义 X_l 轴指向车身正前方，Z_l 轴垂直向上；定义相机坐标系 Z_c 轴指向车身正前方，Y_c 轴垂直向下。

1.1.1 从空间点到像平面坐标系

如图**??** (2) 所示，图像坐标系 (像素坐标系)，以图像左上角的顶点 O_0 为原点, (u, v) 为像素坐标。像平面坐标系 (物理坐标系)，以摄像机光轴与图像平面的交点 O_1 为原点，(x, y) 为像平面坐标。相机理想的成像过程如 图**??** (3)(4) 所示，世界坐标系下点 $\mathrm{P}(X_w, Y_w, Z_w)$ 经坐标变换到相机坐标系下 $P^{'}(X_c, Y_c, Z_c)$ 后，透视变换投影到图像平面，变换过程模型如下：

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} f_x & 0 & u_0 & 0 \\ 0 & f_y & v_0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} \mathbf{R_c} & \mathbf{t_c} \\ \mathbf{0}^T & 1 \end{bmatrix} \begin{bmatrix} X_w \\ Y_w \\ Z_w \end{bmatrix} \tag{1.1}$$

式中，f_x 和 f_y 表示沿图像坐标系的 x 轴和 y 轴方向上单位长度的像素值，与像平面主点 u_0 和 v_0 一起构成相机内参；$\mathbf{R_c}$ 和 $\mathbf{t_c}$ 分别为世界坐标系到相机坐标系的旋转矩阵和平移矩阵。

采集标定板数据 (图**??**)，利用张正友标定法标定相机内参时，可以得到标定板在相机坐标系下的位姿，因此定义世界坐标系 Z_w 轴与相机光轴重合 (垂直于标定板)，$O - X_w Y_w$ 平面与标定板重合，对于 n 帧标定数据，就可以得到 n 对 $\mathbf{R_{ci}}$ 和 $\mathbf{t_{ci}}$ 矩阵；将正交旋转矩阵用 “角-轴” 的方式表示就能得到旋转角 $\theta_{c,i}$, 并由平移量得到标定板相对于相机原点的距离 $\alpha_{c,i}$。

图 1.2. 标定板数据

1.1.2 从 LiDAR 坐标系到相机坐标系

已知标定板 (世界坐标系) 在相机坐标系下的位姿，要求解 LiDAR 坐标系和相机坐标系间的刚体变换关系，首先需得到 LiDAR 坐标系和标定板间的刚体变换关系，因此，我们手动提取标定板的点云

数据，并用最小二乘法进行拟合，得到该平面在 LiDAR 坐标系下位姿估计值的 “角-轴” 值，即 $\theta_{l,i}$ 和 $\alpha_{l,i}$。然后，以 LiDAR 坐标系和相机坐标系下各标定平面到相机坐标系原点的距离差的最小值为目标函数，优化该目标函数得到最优平移量，最后以 LiDAR 坐标系和相机坐标系下各标定平面到相机坐标系原点的法向量夹角的最小值为目标函数，优化该目标函数得到最优旋转量。

相机坐标系原点平移 t(LiDAR 与相机间的距离) 后，相机距标定平面的距离为 $\alpha_{c,i} - \theta_{c,i}^T t$，所以与平移量有关的目标函数如下：

$$\begin{aligned} &\min_t \sum_{i=1}^{n} (\alpha_{l,i} - (\alpha_{c,i} - \theta_{c,i}^T t))^2 \\ \Longrightarrow &t_1 = \arg\min_t ||(\theta_c^T t - (\alpha_l - \alpha_c))||_F^2 \end{aligned} \tag{1.2}$$

解 (**??**), 即可得到平移量 $t_1 = (\theta_c \theta_c^T)^{-1} \theta_c (\alpha_c - \alpha_l)$。

法向量夹角的最小值即夹角余弦的最大值，用正交旋转矩阵表示，则与旋转量有关的目标函数如下：

$$R_1 = \arg\max_R \sum_{i=1}^{n} \theta_{c,i}^T (R\theta_{l,i}) \tag{1.3}$$

其中，R 是正交旋转矩阵，满足 $R^T R = \mathbf{I}_3$，$det(R) = 1$，对公式 (**??**)变形，将该问题转换成典型的正交普鲁克问题 (Orthogonal Procrustes Problem)，如式下所示。

$$R_1 = \max_R trace(\theta_c^T R \theta_l) = \max_R trace(R\theta_l \theta_c^T) \tag{1.4}$$

用奇异值分解（Singular Value Decomposition，SVD）的方法解(**??**)，即可以得到旋转矩阵的解：

$$R_1 = VU^T, st. \theta_l \theta_c^T = USV^T \tag{1.5}$$

以上的求解过程，我们仅使用点云数据估计了标定板平面参数值，为了更有效地利用点云数据并提高算法的鲁棒性，我们使用点云平面拟合得到的内点进一步对上述过程中得到的 LiDAR-Camera 系统外参进行优化。

用 $x_{l,i}$ 表示第 i 帧点云中拟合的标定板平面的内点，即 $x_{l,i} = [x_{l,i}^{(1)}, x_{l,i}^{(2)}, ..., x_{l,i}^{(m)}]$，$x_{l,i} \in R^{3x1}$，$m = m(i)$ 是第 i 帧点云中拟合平面得到的内点的个数, 则建立的统一旋转和平移量的目标函数如(**??**)所示，以上述 R_1 和 t_1 为初值，优化该问题，得到粗标定过程的最终外参矩阵。

$$\arg\max_{R,t} \sum_{i=1}^{n} \frac{1}{m(i)} \sum_{j=1}^{m(i)} (\theta_{c,i}^T (R x_{l,i}^{(j)} + t) - \alpha_{c,i})^2 \tag{1.6}$$

1.2 LiADR-Camera 联合细标定

将相机内参标定和 LiDAR-Camera 联合标定过程统一，均使用棋盘格为靶标，方便采集实验数据，并降低了实验要求，简化了整个标定流程。这种基于黑白棋盘格的离线标定方法虽然简单易实现，但是仍存在一些问题：

(1) 标定精度难以保证；标定受光照和人为因素影响大，很多情况下的标定结果都不能满足后续点云和图像高精度融合的需要。

(2) 标定误差无法更新和校正；这种基于特定靶标的离线标定方法，一旦标定结果发生偏移，都需要重新实验，过程繁琐，无法满足无人驾驶系统在线感知算法的需求。

综合考虑以上问题，我们用真实道路场景中的目标 (人，树，路标，车等) 作为 LiDAR-Camera 系统外标定的匹配模式，基于点云和图像目标边缘对齐的融合结果，在线优化和更新该传感器系统外参。

1.2.1 数据处理

3D 点云和 2D 图像数据模态存在巨大差异，在进行边缘提取前，需先对点云减采样，取与相机视野重合扇区内的点云；并对点云进行简单滤波，除去地面点 (基于高度的滤波或者 RANSAC) 和非目标点 (基于距离滤波或者检测算法)。然后，剪裁 RGB 图像，除去 LiDAR 激光器扫描到的最高高度以外的图像区域，并根据图像剪裁的尺寸更新相机内参。最后处理得到如 图**??**所示的图像和点云边缘。

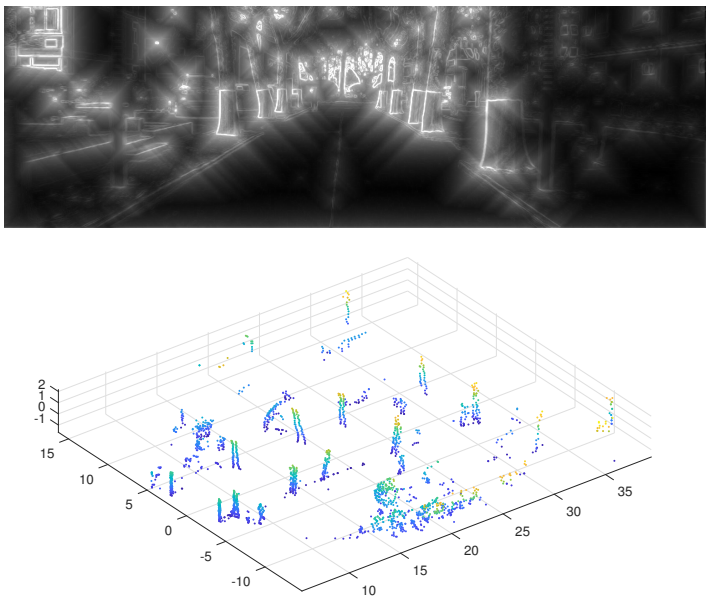


图 1.3. 图像和点云边缘提取结果

点云边缘提取

3D 点云数据离散而稀疏，边缘不连续性明显，因此利用点云数据获取的边缘存在误差 (与点云的分辨率有关)。我们单独分析每帧点云中的每个激光器的返回数据，对于场景中的目标，由于视差和遮挡，则具有固定垂直扫射角的激光器的返回数据中，与左右邻域点相比，越远的点越不可能是目标边缘点。基于该事实，建立式 (**??**)的点云边缘滤波指标。

$$X_i = \max(P_{i-1}^r - P_i^r, P_i^r - P_{i+1}^r, 0)^\gamma \tag{1.7}$$

其中，P_i^r 是返回的第 i 个点的深度值，$\gamma \in (0,1)$ 是为提高近处点云权重而设置的参数，一般取经验值 0.5。

对于 Velodyne 64HDL LiDRA 点云数据，我们以 $X_i \geq 0.3m$ 为基准滤波提取点云数据中目标的边缘。

图像边缘提取

与点云边缘检测算法相比，图像边缘检测算法成熟，检测精度高，但仍然存在一些影响边缘匹配的问题，诸如由光照和阴影导致的假边缘、离散孤立边，这些边难以滤波去除，因此除了人为选择标定场景 (无强光和阴影的阴天或者傍晚等) 外，我们对得到的边缘图进行距离逆变换 (inverse distance transform, IDT)，提高真实边缘的比重并弥补点云边缘不连续性导致的边缘误差。我们首先使用 Sobel 算子提取图像边缘，得到边缘图 E，然后对 E 做 IDT 变换，变换过程如下：

$$D_{i,j} = \alpha \cdot E_{i,j} + (1-\alpha) \cdot \max_{x,y} E_{x,y} \cdot \mu^{\max(|x-i|,|y-j|)} \tag{1.8}$$

其中，i, j, x, y 为图像边缘 E 的像素索引，α 和 μ 分别取经验值 0.333 和 0.98。

1.2.2 基于边缘匹配更新外参

将粗标定过程中得到的旋转外参矩阵转换为欧拉角表示可以得到 LiDAR-Camera 系统的 6DOF 外参，$C = (t_x, t_y, t_z, \theta_x, \theta_y, \theta_z)$，以此为初值，将点云投影到图像平面可以建立与 6DOF 外参有关的目标函数，如下：

$$F_T = \sum_{f=1}^{n} \sum_{p=1}^{X^f} X_p^f \cdot D_{i,j}^f \tag{1.9}$$

其中，n 是选择用于实验的点云图像对的数量，X^f 表示预处理后第 f 帧点云数据中点的个数，(i, j) 表示第 p 个点投影到图像平面的位置索引。

我们使用邻域搜索算法求解问题(**??**)，得到更精确的外参。

1.3 标定流程

1.3.1 数据采集与预处理

数据采集

标定操作在 MATLAB 环境下进行，在 Calibration 文件夹下，包含 CoarseCalibration、FineSearch、fToolBox、LCCT1、TOOLBOX_calib 五个文件夹。CoarseCalibration 文件夹下主要包含了 img_veloPreprocess.m 和 show.m 两个文件，img_veloPreprcoess 根据筛选出的的图片来产生图片和点云对，show 根据得到的标定矩阵来显示标定的结果。FineSearch 文件夹下主要包含了 dataPreprocess.m 和 Search.m 两个文件，dataPreprocess 用来生成细标定搜索使用的数据，Search 用来根据粗标定得到的结果和已有数据进一步优化标定矩阵，使之更为准确。fToolBox 文件夹主要包含了 timeAlign.m，timeAlign 用来对采集到的数据进行时间对齐并按时间顺序标号。LCCT1 文件夹为外参粗标定使用的工具箱文件夹。

TOOLBOX_calib 文件夹为相机内参标定使用的工具箱的文件夹。

时间同步

采集数据后，使用 MATLAB 对数据进行预处理。首先进行时间同步：

- ⋆ 使用文件: fToolBox\timeAlign.m；
- ⋆ 部分代码片段如图**??**所示；
- ⋆ 改其中的路径 Path 为存储标定数据的根文件夹；
- ⋆ 该程序会自动在根文件下创建 dataSet 文件夹，其中 dataSet 文件夹又包含 img、velo、target、img_select 四个子文件夹，分别用来存储时间对齐后的图像、时间对齐后的点云、筛选出用来标定的图片和点云对、筛选出的图片；
- ⋆ 点击 “Run” 运行，即可完成时间对齐，图像和点云分别以 img_xxxxxx.jpg 和 velo_xxxxxx，(xxxxxx 为时间戳顺序的编号) 存储在 img 和 velo 文件夹下。

```
Path = 'G:\20180527\data-search\';
img_Path = strcat(Path,'img-cal\');
imgDir = dir(sprintf('%s/*.jpg', img_Path)); % imgDir  = dir([img_Path '*.jpg']);
img_time=[];
for imgID = 1:1:length(imgDir)
    Str = imgDir(imgID).name;
    imgName(imgID).fullName=Str;
    hour = str2double(Str(12:13));
    minute = str2double(Str(15:16));
    second = str2double(Str(18:19));
    millisecond = str2double(Str(21:23));
```

图 1.4. 时间同步代码片段

选取图片

- ⋆ 在 img 文件夹下，人工选择各个位置下的光照良好标定板无亮斑且边缘整齐的图片约 40 张放入 img_select 文件夹内；
- ⋆ 使用 MATLAB 相机标定工具箱 Camera Calibrator，图标与界面如图**??**所示。

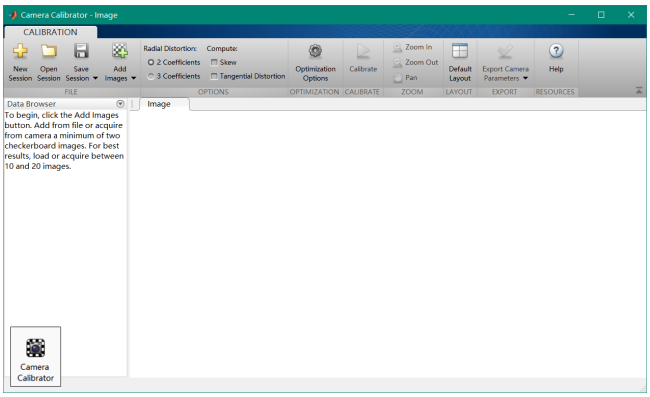


图 1.5. Camera Calibrator 图标与界面

具体操作流程如下：

1) Add Images 添加 img_select 文件夹下的图片，点击“打开”；
2) 弹出窗口如图??所示，Size of checkerbored square 为标定板格子的边长，设定为 100mm，点击“确定”；

图 1.6. 设定标定板格子边长

3) 点击 Calibrate 进行标定运算;
4) 选择右侧 Reprojection Errors 中值较高的图片，即为我们可以筛掉的图片。每次选择 Reprojection Errors 最高的一张图片，在 img_select 文件夹中将其删除，并在界面中右键选择对应照片，点击“Remove and Recalibrate”重新进行运算，如图??所示;

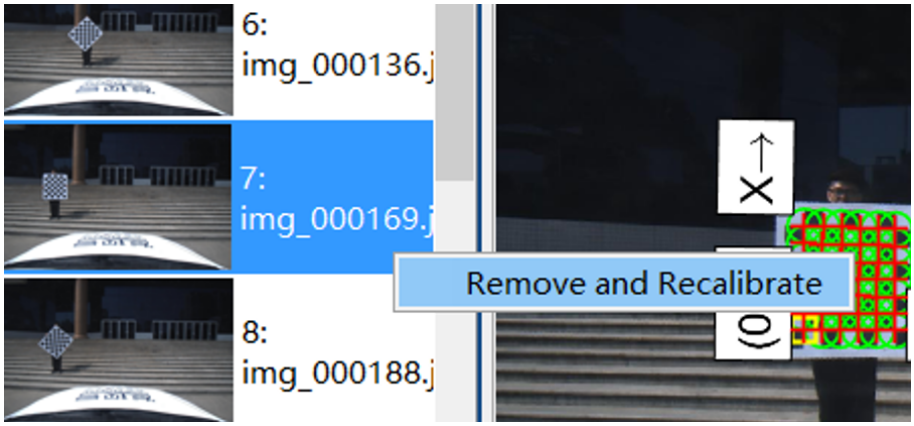


图 1.7. 删除图片并重新运算

5) 最后保留 28-35 张图片;
5) 点击“Export Camera Parameters”保存工具箱计算得到的相机内参，为后续标定作参考。

图片筛选完成后，进行如下操作来生成最终标定使用的图片点云对：

⋆ 使用文件: CoarseCalibration\img_veloPreprocess.m，部分代码片段如图??所示；
⋆ 修改其中的路径 Path 为标定数据文件夹下的 dataSet 文件夹；
⋆ 点击“Run”运行，即可找到选出的图片的对应激光点云，并保存在 target 文件夹里，名称为 laser_targetX.xyz 和 img_targetX.jpg。

```
Path = 'G:\20180522\data-cal\dataSet\';
save_file = strcat(Path,'target');
file_dir = strcat(Path,'velo\');
veloDir = dir(sprintf('%s\\*.bin', file_dir));
img_dir = strcat(Path,'img_select\');
imgDir = dir(sprintf('%s\\*.jpg', img_dir));
for i=1:1:length(imgDir)
%     i
    Str = imgDir(i).name;
    imgF=[img_dir, Str];
    img = imread(imgF);
```

图 1.8. img_veloPreprocess 部分代码片段

1.3.2 相机内参标定

使用标定工具箱 Matlab Calibration Toolbox:

⋆ 右键选择 TOOLBOX_calib 文件夹，添加到路径 → 选定的文件夹和子文件夹;
⋆ 将 MATLAB 当前文件夹设为 target 文件夹;
⋆ 命令行输入 calib_gui，出现界面如图??所示，选择 “Standard”;

图 1.9. 初始界面

⋆ 出现界面如图??所示，选择 “Image Names”;

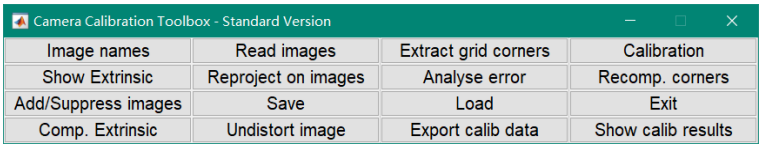
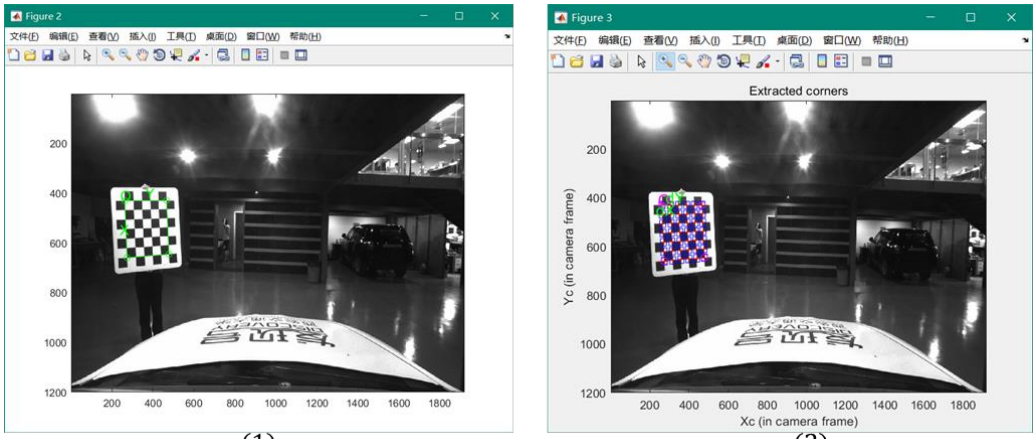


图 1.10. 工具箱界面

⋆ 命令行出现 “Basename camera calibration images (without number nor suffix):”，此时需要输入图像基础名称，所以在命令行输入 “img_target”，回车;
⋆ 命令行出现 “Image format: ([]=’r’=’ras’, ’b’=’bmp’, ’t’=’tif’, ’p’=’pgm’, ’j’=’jpg’, ’m’=’ppm’)”，此时需要选择图像格式，所以输入 “j”，回车;
⋆ 图片加载完成后，选择 “Extract Grid Corners”，命令行出现 “Number(s) of image(s) to process ([] = all images) =”，回车默认选择全部图片;

⋆ 命令行出现 “wintx ([] = 15) =” 和 “winty ([] = 15) =”，即为选择窗口大小，两次回车即可;
⋆ 命令行出现 “Do you want to use the automatic square counting mechanism (0=[]=default)or do you always want to enter the number of squares manually (1,other)?”，回车即可;
⋆ 弹出窗口中，接下来需要选择标定板的四个角点，选定其中一角作为原点，每次按照该原点按顺时针围一个框，如图**??**(1) 所示，选好后回车确认，命令行出现 “Size dX of each square along the X direction ([]=100mm) = Size dY of each square along the Y direction ([]=100mm) =”，回车确认，此时出现角点检测后的结果，如图**??**(2) 所示



(1) (2)

图 1.11. (1) 选择标定板角点 (2) 角点检测结果

⋆ 命令行出现 “Need of an initial guess for distortion? ([]=no, other=yes)”，再次回车出现下一张图，重复上述操作直到所有图片检测完毕;
⋆ 选择 “Calibration” 进行相机内参标定运算;
⋆ 选择 “Save”，保存结果，会在 target 文件夹下生成名称为 Calib_Result.mat 的文件。

1.3.3 LiDRA-Camera 联合粗标定

使用 LCCT 工具箱标定 LiDRA-Camera 系统初始外参:

⋆ 右键选择 LCCT1 下的 src 文件夹，添加到路径 → 选定的文件夹和子文件夹;
⋆ 将 MATLAB 当前文件夹设为 target 文件夹;
⋆ 命令行输入 lasercamcalib，出现界面如图**??**所示;
⋆ 选择 “Select a camera calib file”，选择 Calib_Result.mat;
⋆ 选择 “Search scan files”，命令行出现 “Enter base name of scans (without numbers or suffix):”，即选择点云文件基础名称，输入 laser_target，回车;
⋆ 命令行出现 “Enter suffix of scan files ([] = ”xyz”):”，即选择点云文件扩展名，直接回车即可;
⋆ 选择 “List active” ;

⋆ 选择 “Select planes”，命令行出现 “Enter indices of range images to preview ([]=all active):”，直接回车即可；

⋆ 命令行出现 “use range image (r) or 3D point cloud (p) to select checkboard plane? (default:p):”，此时需要选择选定标定板的方法，我们选择 rang image 的方式，故输入 r，回车；

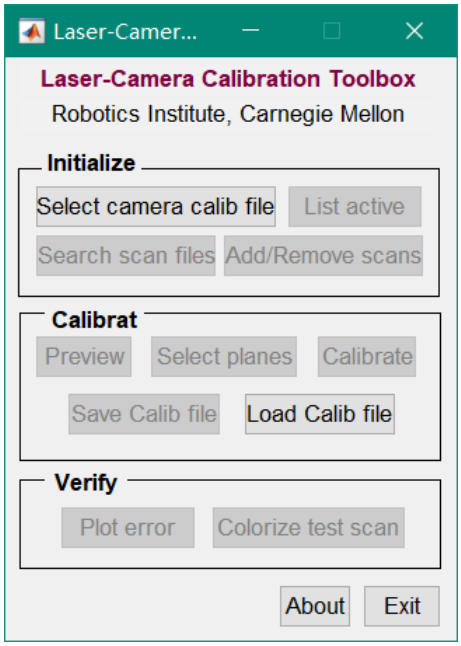


图 1.12. LCCT 工具箱界面

⋆ 出现图片窗口如图**??**所示，选择 Select plane, 左键选择标定板角点，ESC 可以取消之前的选定，选好后右键确认；

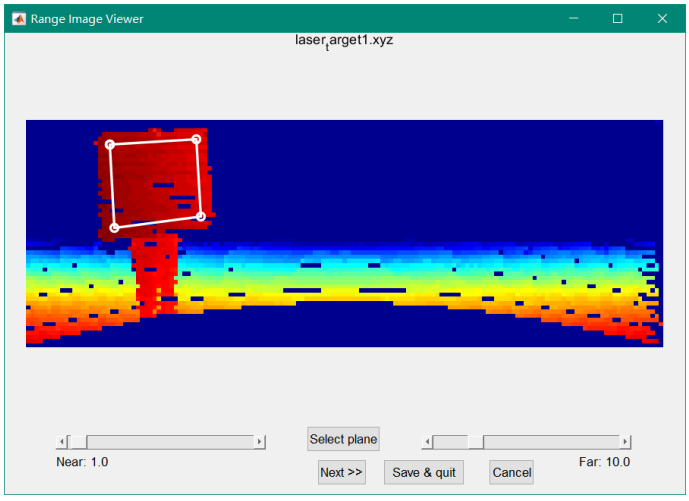


图 1.13. 选择标定板平面区域

⋆ 选择“Next”，然后重复操作，直至处理完所有图片；
⋆ 选择“Save&Quit”；
⋆ 在工具箱界面选择“Calibrate”进行标定运算；
⋆ 选择“Save”，输入要保存的名字，回车即可保存。

1.3.4 LiDRA-Camera 联合细标定

执行搜索算法前，先处理用于搜索的数据：

⋆ 预选选取 5-15 对时间同步后边缘明显的点云-图像数据，保存在 FineSearch\Data\ORGData 目录中；
⋆ 运行 FineSearch 目录下数据处理文件 dataPreprocess.m，调用 IDT.m 函数处理图像，调用 PC_filter.m 函数处理点云，将得到的图像边缘和点云边缘保存于结构体数据 Data.mat 中，存于 FineSearch\Data 目录下；
⋆ 调用自定义工具箱 fToolBox 中 Rt2C.m 将粗标定过程中得到的外参矩阵用 6DOF 外参 C 表示；

得到图像和点云边云后，使用搜索算法精确标定传感器系统外参：

⋆ 初始化 FineSearch 目录下 Search.m 函数中的内参 K_int 和外参 C；
⋆ 更新搜索步长 delta_t 和 delta_R；
⋆ 执行 Search.m 文件；
⋆ 搜索的关键代码片段如图??所示。

```
while(Is_True(C_max,C_optm0)==0)
    C_optm0 = C_max;
    CC = C_729(C_optm0, delta_t, delta_R);
    JC=zeros(729,1);
    disp('start')
    for CC_num=1:1:729
        JC_f = 0;
        C_f = CC(CC_num,:);
        for f = 1:1:4
            D = double(Data(f).D);
            X = Data(f).X;
            uv = xyz2uv(X, C_f, K_int); % uv=[u,v,Xpi]
            ithx = find(uv(:,1)<img_w & uv(:,1)>1 & uv(:,2)<img_h & uv(:,2)>1);
            uv_f = uv(ithx,:);
            JC0 = obj_func(uv_f,D);
            JC_f = JC_f+JC0;
        end
        JC(CC_num,1) = JC_f;
    end
    [maxJC,I] = max(JC(:,1));
    C_max = CC(I,:);
    iter_count = iter_count+1;
end
```

图 1.14. Search.m 代码片段

1.4 标定注意事项

1) 制作棋盘格靶标时应特别注意，黑色方格与白色方格尺寸需要相同，而且所有方格的尺寸必须严格一致。靶标的方格数量不宜太小，行数和列数以大于 7 为宜。方格的尺寸不宜太大或太小，采集的整幅靶标图像中方格的边长尺寸不小于 20 像素。
2) 采集靶标图像时应特别注意，需要在不同的角度不同的位置采集靶标的多幅图像。采集到的图像必须清晰，靶标图像尺寸以占整幅图像尺寸的 1/3-3/4 为宜。靶标图像最好在整幅图像的不同位置都有分布，不宜过于集中于同一区域。靶标放置位置与摄像机之间的距离最好为视觉系统的主要工作距离。靶标相对于摄像机的角度应有较大范围的变化，应包含绕三个轴较大角度的旋转，最好不小于 30 度。采集的靶标图像数量不应太少，建议以 20-35 幅靶标图像为宜。
3) 内参标定，提取角点时，在图形窗口利用鼠标点击设定棋盘格靶标的选定区域。点击的第一个角点作为靶标坐标系的原点，顺序点击 4 个角点形成四边形。相邻两次点击的角点应在同一条网格线上，使得所形成的四边形的边应与棋盘格靶标的网格线基本平行。为提高点击的角点的精度，建议将显示靶标图像的图像窗口放大到最大，利用鼠标的十字标线尽可能准确的点击 4 个角点。
4) 外参标定，提取角点时，在图形窗口利用鼠标点击的第一个角点作为靶标坐标系的原点，得到的外参数是靶标坐标系在摄像机坐标系中的位姿。
5) 采集细标定数据时，避免强光和阴影，建议傍晚和阴天采集数据。