Program Details

1. Start MIAS-LCEC Toolbox

The program consists of two independent pacakges:

No.	Package Name	Function
1	zvision	A C++ package , which is the main UI of the project
2	c3m	A Python package, the main function is to receive command and data from zvision package and complete the segmentation function and cross-modal mask matching function, and then send the matched points pair back to zvision package for further PNP solution.

The zvsion package and c3m package communicate with each other through ros2 topic publishing and subscribing mechanism,there will be some problem if you run both zvision package and c3m package in two computers in a local network since there will be two pubulishing node and subscribe node with the same name in the network. If you need to run zvision and c3m on two or more computers at the same time, please disconnect the computers with the network.

To start MIAS-LCEC Toolbox, you only need to run:

```
cd ~/MIAS-LCEC/
sh mias_lcec.sh
```

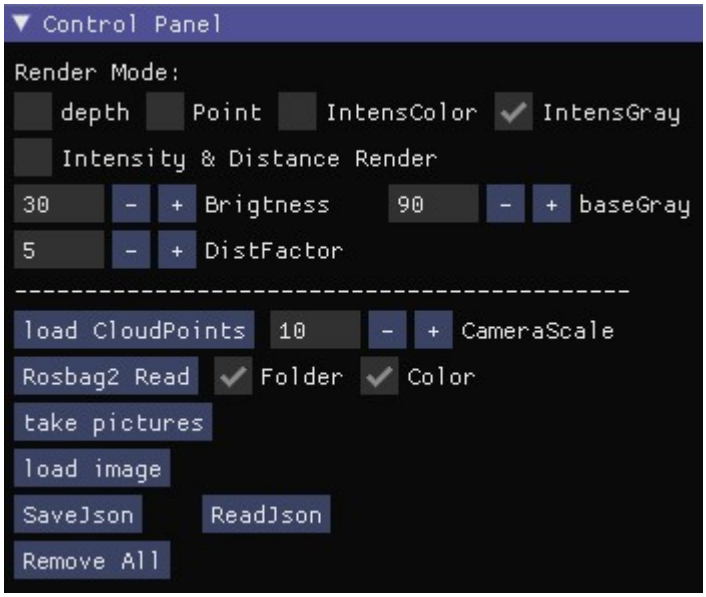
2. UI Introduction

2.1 control panel

This panel includes checkboxs, inputbox and buttons for general cloud and picture control.

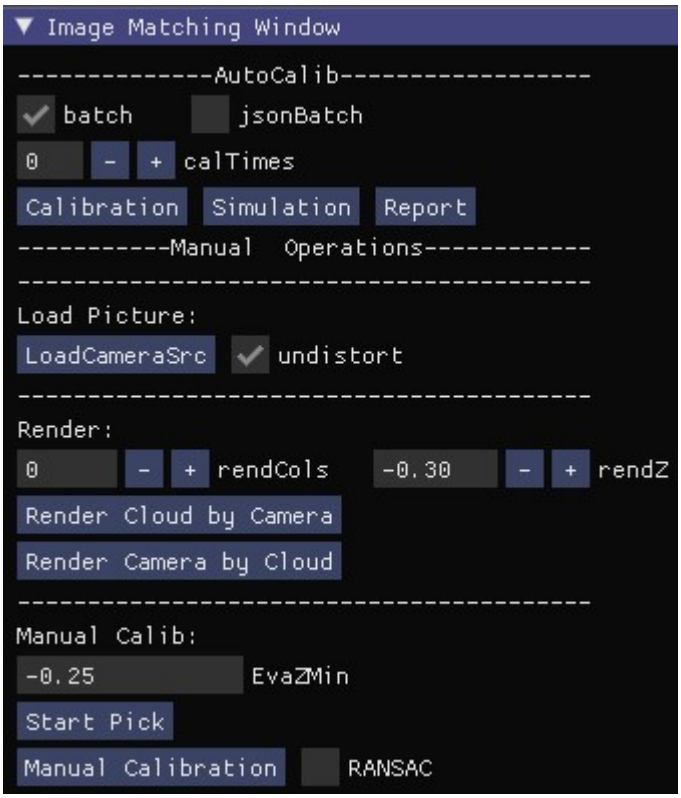
Controls	Function
render Mode checkboxes and inputboxes	choose the cloud rendering mode and set the rendering params.
load CloudPoints Button	load a cloud file and display. then use left/right/mid/mid press/ of the mouse to rotate and zoom the cloud

Controls	Function
Rosbag2 Read Button	read the pcd and image in a rosbag2 file, if the checkbox Folder is selected, it can handle all the bag files in a folder, if the checkbox color is selected, all image files will be in RGB mode.
take picture Button	take a cloud picture by the cloud camera and show it in the cloud image window.
load image Button	load an image from a file and show it in the cloud image window
ReadJson Button	read a json file including pics rendering condition and intrinsic/extrinsic parameters, and then set the "cloud camera window" and set the intrinsic/extrinsic parameters of the cloud camera.
SaveJson Button	save the intrinsic/extrinsic parameters of the cloud camera, and the pic rendering conditions to a json file.
Remove All Button	remove all loaded pictures and clouds.



2.2 image matching panel

Controls	Function
load picture & render controls	load a picture and render the picture by cloud or render the cloud by the picture
autocalibration controls	automatic calibration and generate calibration report
manual calibration controls	picking points from cloud and image, and then complete the manual calibration



2.3 cloud camera panel

Controls	**Function**
camera coordination controls	adjust the x,y,z and rotation angles around axis for the cloud camera
pics rendering condition controls	adjust the params for rendering pictures genrated by the cloud camera
points filter controls	adjust the filter params for the cloud camera to filt the points
speedup conditions controls	adjust params for speedup the cloudcamera projection process

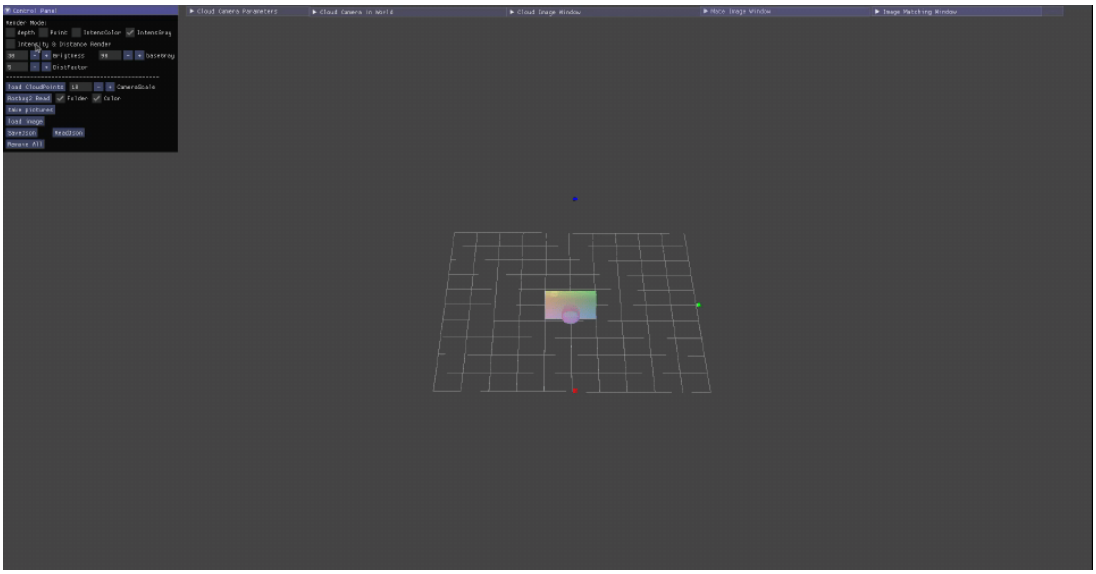
2.4 cloud camera parameters

The inputbox in the panel is to show the distcoeffs, intrinsic parameters and extrinsic parameters of the cloud camera.

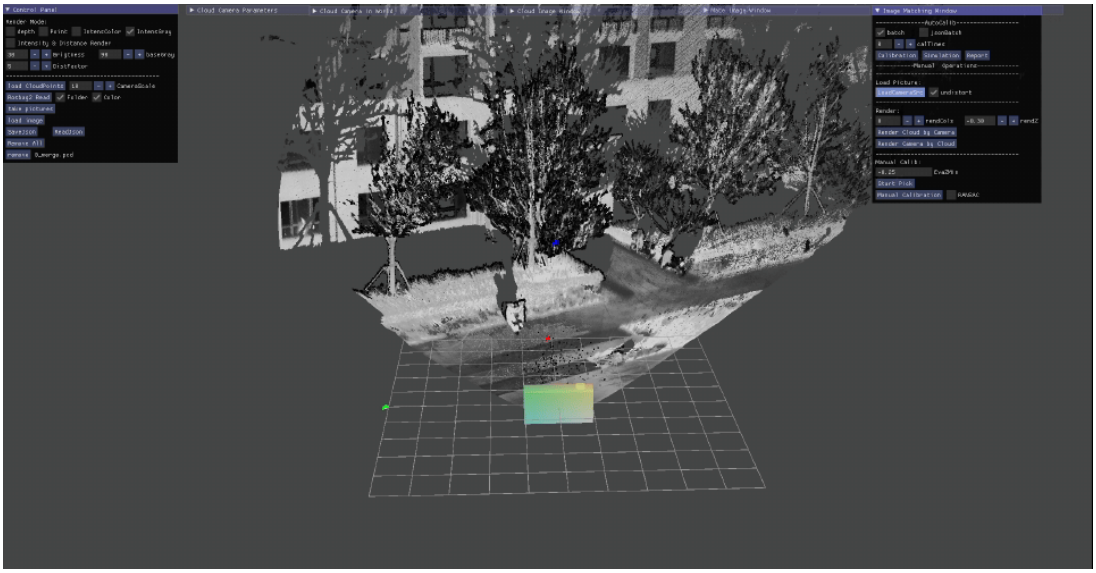
3. Browse LiDAR Point Clouds and Camera Images

In MIAS-LCEC Toolbox, it is convenient to browse point clouds and camera images.

- ***Load and browse point cloud***: Click button [load CloudPoints] in control panel, you can easily load pcd/ply files and browse them in the toolbox interface. It allows 3D perspective transformations through simple mouse operations, enabling the observation of point clouds from various view points.



- ***Load and browse images***: Click button [LoadCameraSrc] in image matching panel, you can read images (bmp/png/jpg) in the internal window [cloud image window]. You can also drag the internal window in the interface to view images wherever you want. The window can be adjusted to different sizes.



4. Automatic Calibration

4.1 Single Calibration

Step	Operation
1	deselect checkbox "**batch**" and checkbox "**jsonbatch**"
2	click Button "**calibration**"
3	select a cloud file according to the promption
4	select a image file according to the promption
5	select a json file of intrinsic,config and extrinsic according to the promption

Step	Operation
6	the result file will be found at directory "**cal**" in the same directory with the cloud file

4.2 Batch Calibration

When you have a group of clouds and images with same intrinsic and extrinsic parameters, batch calibration function would be a better choice.

4.2.1 File Naming Rules

To ensure the cloud and its matching image be recognized correclty, file name shall comply with below rules:

- cloud file and its matching image file shall be put in the same folder.
- the essential name of the cloud and the image shall be the same.
- pcd name shall be : essential name+"_merge.pcd"
- image name shall be : essential name+"_1.png"
- example: test_merge.pcd, test_1.png, will be regarded as a pair and calibrate.

4.2.2 Operation Steps

Step	Operation
1	select checkbox "batch" and deselect checkbox "jsonbatch"
2	click Button "calibration"
3	select a json file of intrinsic,config and extrinsic according to the promption
4	select a folder including cloud and image files
6	the result file will be found at directory "intelibatch" in the same directory with the cloud file

4.3 JsonBatch Calibration

When dealing with multiple point clouds and images with varying intrinsic and extrinsic parameters, you can organize the files by grouping those with the same intrinsic and extrinsic parameters into separate folders. Then, by editing a batch testing JSON file, you can instruct the program to automatically batch calibrate all the files.

For batchtesting json file , please read the example file "batchtestingexample.json":

```
{
    "targetFolder":"DatasetsReleaseTest",
    "batchType":1,
    "itrCounts":6,
    "testFileFolder":"/media/D/zvoutput/DatasetsRelease/TF70Data",
    "trueJsonFolder":"/media/D/zvoutput/DatasetsRelease/TF70TrueJson",
    "RemarksA":"this is the example Json file of bath Test for TF70 Dataset",
    "RemarksB":""
}
```

4.3.1 Operation Steps

Step	Operation
1	deselect checkbox "**batch**" and select checkbox "**jsonbatch**"
2	click Button "**calibration**"
3	select the batch testing json file according to the promption
4	the result file will be found at directory you assigned in the batch testing json file

5. Manual Calibration

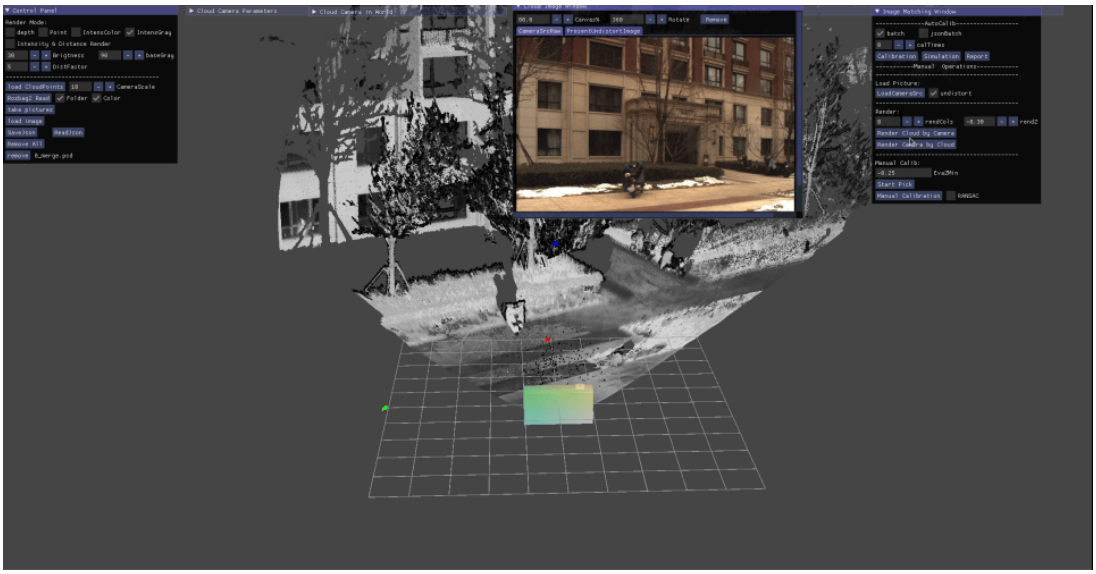
5.1 Manual Calibration

Step	Operation
1	press "**load cloudPoints**" button in "**Control Panel**" to load a cloud
2	press "**Read Json**" Button in"**Control Panel**" to load a json file including intrinsic/distcoeffs/picRenderChoice
3	Press "**load CameraSrc**" Button to load a image, make sure the "undistort" checkbox is selected
4	Press "**Picking**" Button, and press "**set pick**" button in cloud image window to set the "undistort picture" as picking image, then using right-click mouse to pick points in cloud, and using left-click 0.2S to pick points in picture.
5	Press "**Manual Calibration**" , then a evaluation table will appear in the cloud image window, and a testreport will be generated in the "**man**" folder at the same location of the cloud file
6	then you can use "**Render Camera by Cloud**" and "**Render Cloud by Camera**" to further observe the fusion effect

5.2 Rendering Function

Rendering function is designed to observe the fusion effect of cloud and camera images.it can be used for further adjust the manual calibration results and get a better extrinsic calibration.

Step	Operation
1	press "**load cloudPoints**" button to load a cloud
2	press "**Read Json**" Button to load a json file including intrinsic/distcoeffs/picRenderChoice
3	Press "**load CameraSrc**" Button to load a image, make sure the "**undistort**" checkbox is selected
4	Press "**Render Camera by Cloud**", a rendered picture will appeared in the cloud image window
5	Press "**Render Cloud by Camera**", the cloud will be rendered by the image



6. Calibration Configuration

Calibration config is critial for geting a correct calibration result.

6.1 Objects in the Config Json File

In this program, we integrate all the configs in a json file.

- object "**picRenderChoice**": setting the parameters for how to take pictures of the cloud by the cloud camera
- object "**cloudCamera.Intrinsic**": setting the intrinsic of the cloud camera,including K matrix and DistCoeffs vector
- object "**cloudCamera.Extrinsic**": setting true extrinsic (R and T vec) of the real camera, the extrinsic parameters do not affect the calibration result, they are only used to evalute the calibration result.
- object "**cloudCamera.EulerR"**: the corresponding Euler angle of R, which is calculated by the program throught the R and T vec.
- object "**cloudCamera.WorldT**" it is used to depict the point translation, and is calculated by the program using -(R*)^(-1)t*.
- object "**EulerType**": default value is 5, means "ZYX".

The program will automatically calculate the EulerR and WorldT anytime we reading or write a json file, or using the json file for calibration. so the calibration will keep correct even if the EulerR and WorldT is wrong input manually.

6.2 Definitions of [picRenderChoice]

the params in the picRenderChoice is to define how to take cloud pictures by the virtual camera, it's very important for calibration, please read this section carefully before starting calibration.

No.	Params	definistion	Default Value

No.	Params	definistion	Default Value
1	fourNeighbor	define if the up/down/left/right four neighbor points need to be rendered in the projected image,default is false	false
2	eightNeighbor	define if the up/down/left/right eight neighbor points need to be rendered in the projected image,default is false	true
3	baseGray	define the base gray value when calculate the gray value of the projected points	0
4	contrast	define the contrast of the projected points	2
5	lowIntensity	define the min intensity of the points to be projected	0
6	zMin	only the cloud points with Z value greater than zMin will be projected	-30
7	zMax	only the cloud points with Z value less than zMax will be projected	30
8	xMin	only the cloud points with x value greater than xMin will be projected	-40
9	xMax	only the cloud points with x value less than xMax will be projected	800
10	picWidth	the width of the virtual picture ,** it should be equal to the real camera picture width **	1200
11	picHeight	the height of the virtual picture, ** it shoud be equal to the real camera picture height **	800
12	samplingStep	defines the interval of sampling from the cloud points	1
13	dMin	only the cloud points with the Zc value greater than dMin will be projected	-10
14	dMax	only the cloud points with the Zc value less than dMax will be projected	800
15	color	defines take colorfull image or gray image for the virtual camera	false
16	OMP	defines if the OMP function will be used in virtual camera	true
17	OMPthreads	defines the qty of threads used in the function	3
18	selfRotation	defines virtual camera rotation method	true
19	saveornot	defines if the picture will be saved or not(only effectively in manual mode)	false

7. How to Read the Calibration Report

The calibration report is written in json format, it includes below objects:

- **trueCamera**: recording the Intrinsic and Extrinsic set by the calibration config

- **calCamera**: recording the calibration output Intrinsice and Extrinsic
- **Evaluation**: recording all the evaluation indicators.
- **cloudCamera**: recording the Intrinsic and Extrinsic of the cloud camera used to take cloud pictures in the calibration
- **matePRC**: recording the setting of picRenderChoice used in the calibration, it's the same with the input config before calibration.

main evaluation indicators

- **Er**: the Errors of EulerR of the calCamera against trueCamera
- **Et**: the Errors of WorldT of the calCamera against trueCamera