



深志科技
Perseverance Technology

Moildev Documentation

Release 4.1.0

Haryanto

July 25, 2023

CONTENTS:

1	Release Note	1
1.1	Moildev-4.1.0	1
1.2	Release Highlight	1
1.3	Copyright	1
2	Getting Started	2
2.1	How to install	2
2.2	Import Library	2
2.3	Utilization	2
3	Architecture	6
3.1	Overview	6
3.2	Supported Platform	7
4	Sample Application	8
4.1	1. Car Application	8
4.2	2. Internal Thread Inspection	8
4.3	3. 3D measurement	8
4.4	4. Visual odometry	8
5	API Reference	9
5.1	Moildev Module	9
5.2	Testing code	15
	Python Module Index	16
	Index	17

RELEASE NOTE

1.1 Moildev-4.1.0

Moildev-4.1.0 is continued development of moildev SDK. Its has a new updated for make it better result in processing fisheye image and more easy in utilization

1.2 Release Highlight

- Panorama car and Panorama Tube can horizontally flip by passing the boolean parameter on this function

1.3 Copyright

© 2019-2023 Perseverance Technology. All rights reserved.

GETTING STARTED

Moildev Library is a collection of functions support python to developments fisheye image applications.

2.1 How to install

There are two ways to use moildev. The first way is by visiting the moildev library, now available in the pypi distribution. The second way is cloning from GitHub repository.

```
$ pip install Moildev
```

or

```
$ git clone https://github.com/McutOIL/moildev.git
```

2.2 Import Library

import moildev library:

```
from moildev import Moildev
```

2.3 Utilization

Create moildev object

To create the object from Moildev, you have to provide the parameter. The camera parameter is the result from calibration camera by MOIL laboratory that will store on **.json** file.

```
moildev = Moildev("Camera_Parameter_Path")
```

Parameter:

Camera_Parameter_Path : The path of the *.json* file that stored the camera parameter from calibration result.

Example:

```
moildev = Moildev("camera_parameter_path.json")
```

Load Image

```
image = cv2.imread("Image_Path in your computer")
```

2.3.1 Create anypoint maps mode 1

Purpose:

Generate a pair of X-Y Maps for the specified alpha, beta and zoom parameters, and then utilize the resulting X-Y Maps to remap the original fisheye image to the target angle image. This function has 2 mode to generate maps anypoint, mode 1 is for tube application and mode 2 usually for car application.

```
map_X, map_Y = moildev.maps_anypoint_mode1(alpha, beta, zoom)
```

Example:

```
map_X, map_Y = moildev.maps_anypoint_mode1(90, 180, 2)
anypoint_maps_m1 = cv2.remap(image, map_X, map_Y, cv2.INTER_CUBIC)
anypoint_maps_m1 = cv2.resize(anypoint_maps_m1, (400, 300))
cv2.imshow("anypoint maps mode 1", anypoint_maps_m1)
```

2.3.2 Create anypoint mode 1

Purpose:

Generate anypoint view image. for mode 1, the result rotation is betaOffset degree rotation around the Z-axis(roll) after alphaOffset degree rotation around the X-axis(pitch). for mode 2, The result rotation is thetaY degree rotation around the Y-axis(yaw) after thetaX degree rotation around the X-axis(pitch).

```
anypoint_m1 = moildev.anypoint_mode1(image, alpha, beta, zoom)
```

Example:

```
anypoint_m1 = moildev.anypoint_mode1(image, 90, 180, 2)
anypoint_m1 = cv2.resize(anypoint_m1, (400, 300))
cv2.imshow("anypoint made 1", anypoint_m1)
```

2.3.3 Create anypoint maps mode 2

Purpose:

Generate a pair of X-Y Maps for the specified pitch, yaw, and roll also zoom parameters, and then utilize the resulting X-Y Maps to remap the original fisheye image to the target image.

```
map_X, map_Y = moildev.maps_anypoint_mode2(pitch, yaw, roll, zoom)
```

Example:

```
map_X, map_Y = moildev.maps_anypoint_mode2(-90, 0, 0, 2)
anypoint_maps_m2 = cv2.remap(image, map_X, map_Y, cv2.INTER_CUBIC)
anypoint_maps_m2 = cv2.resize(anypoint, (400, 300))
cv2.imshow("anypoint maps mode 2" anypoint_maps_m2)
```

2.3.4 Create anypoint mode 2

Purpose:

Generate anypoint view image. for mode 1, the result rotation is betaOffset degree rotation around the Z-axis(roll) after alphaOffset degree rotation around the X-axis(pitch). for mode 2, The result rotation is thetaY degree rotation around the Y-axis(yaw) after thetaX degree rotation around the X-axis(pitch).

```
anypoint_m2 = moildev.anypoint_mode2(image, pitch, yaw, roll, zoom)
```

Example:

```
anypoint_m2 = moildev.anypoint_mode2(image, -90, 0, 0, 2)
anypoint_m2 = cv2.resize(anypoint_m2, (400, 300))
cv2.imshow("anypoint mode 2", anypoint_m2)
```

2.3.5 Create panorama tube

Purpose:

To create an image with a panoramic view

```
panorama_tube = moildev.panorama_tube(image, alpha_min, alpha_max)
```

Example:

```
panorama_tube = moildev.panorama_tube(image, 10, 110)
panorama_tube = cv2.resize(panorama_tube, (400, 300))
cv2.imshow("panorama tube", panorama_tube)
```

2.3.6 Create panorama car

Purpose:

The function that generate a moil dash panorama image from fisheye camera. the image can control by alpha to change the pitch direction and beta for yaw direction. in order to select the roi, we can control by the parameter such as left, right, top, and bottom.

```
panorama_car = moildev.panorama_car(image, alpha_max, alpha, beta, left, right, top, ↵
↵bottom)
```

Example:

```
panorama_car = moildev.panorama_car(image, 110, 80, 0, 0.25, 0.75, 0, 1)
panorama_car = cv2.resize(panorama_car, (400, 300))
cv2.imshow("panorama car", panorama_car)
```

2.3.7 Create recenter

Purpose:

Function to change the optical point of the fisheye image.

```
recenter = moildev(image, alpha_max, IC_alpha_degree, IC_beta_degree)
```

Example:

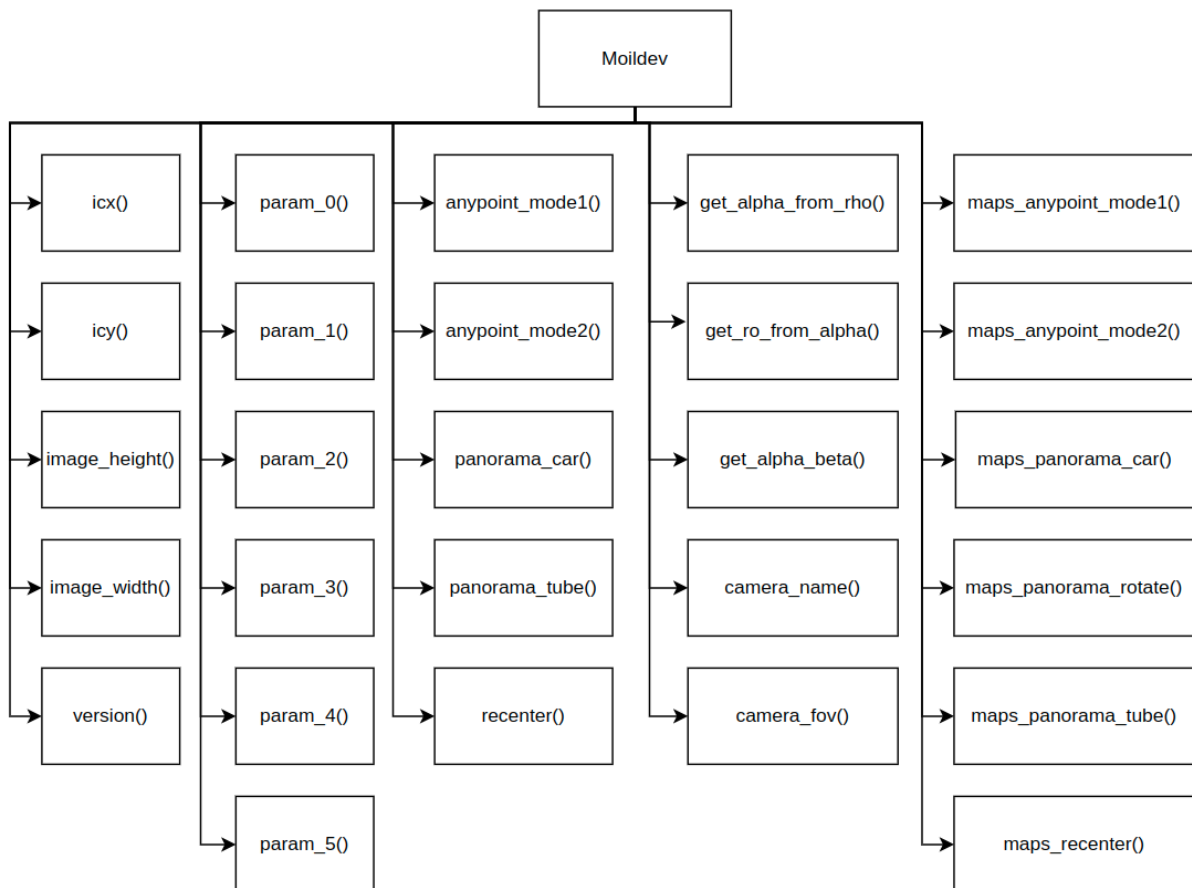
```
recenter = moildev.recenter(image, 110, 25, 10)
recenter = cv2.resize(recenter, (400, 300))
cv2.imshow("show recenter", recenter)
```

Ming-Chi Omni-directional, Surveillance, and Imaging laboratory (MOIL-Lab), Ming Chi University of Technology, Taiwan

ARCHITECTURE

3.1 Overview

Moildev Library is a collection of functions to developments fisheye image applications. Initially, this library is written for C++ programing language. In the recent decade, python is a programming language that is very popular for developers. That is the reason why we are developing the Moildev library for python.



3.2 Supported Platform

SAMPLE APPLICATION

4.1 1. Car Application

4.2 2. Internal Thread Inspection

4.3 3. 3D measurement

4.4 4. Visual odometry

API REFERENCE

5.1 Moildev Module

class `Moildev`(*file_camera_parameter=None, camera_type=None, **kwarg*)

Bases: object

Before the ensuing functions can function properly, the camera parameter must be configured at the start of the program. The camera parameter is the outcome of the MOIL laboratory's calibration camera.

Parameters

- **file_camera_parameter** – *.json file
- **camera_type** – the name of the camera type used (use if you pass the parameter using *.json file)
- **cameraName** – the name of the camera used
- **cameraFov** – camera field of view (FOV)
- **sensor_width** – size of sensor width
- **sensor_height** – size of sensor height
- **Icx** – center image in x-axis
- **Icy** – center image in y-axis
- **ratio** – the value of the ratio image
- **imageWidth** – the size of width image
- **imageHeight** – the size of height image
- **calibrationRatio** – the value of calibration ratio
- **parameter5** (*parameter0* ..) – intrinsic fisheye camera parameter get from calibration

for more detail, please reference <https://github.com/perseverance-tech-tw/moildev>

classmethod `version()`

Showing the information of the version moildev library.

Returns

Moildev version information

property `camera_name`

Get camera name used.

Returns

Camera name (string)

property camera_fov

Get Field of View (FoV) from camera used.

Returns

FoV camera (int)

property icx

Get center image x-axis from camera used.

Returns

Image center X (int)

property icy

Get center image y-axis from camera used.

Returns

Image center Y(int)

property image_width

Get the width of the image used.

Returns

image width(int)

property image_height

Get the height of the image used.

Returns

image height(int)

property param_0

Get the value of calibration parameter_0 from camera used.

Returns

Parameter_0 (float)

property param_1

Get the value of calibration parameter_1 from camera used.

Returns

Parameter_1 (float)

property param_2

Get the value of calibration parameter_2 from camera used.

Returns

Parameter_2 (float)

property param_3

Get the value of calibration parameter_3 from camera used.

Returns

Parameter_3 (float)

property param_4

Get the value of calibration parameter_4 from camera used.

Returns

Parameter_4 (float)

property param_5

Get the value of calibration parameter_5 from camera used.

Returns

Parameter_5 (float)

maps_anypoint_mode1(*alpha, beta, zoom*)

Generate a pair of X-Y Maps for the specified alpha, beta and zoom parameters, and then utilize the resulting X-Y Maps to remap the original fisheye image to the target angle image. This function has 2 mode to generate maps anypoint, mode 1 is for tube application and mode 2 usually for car application

Parameters

- **alpha** – value of zenith distance(float).
- **beta** – value of azimuthal distance based on cartography system(float)
- **zoom** – value of zoom(float)

Returns

the mapping matrices X mapY: the mapping matrices Y

Return type

mapX

please reference: [`https://github.com/perseverance-tech-tw/moildev`](https://github.com/perseverance-tech-tw/moildev)

maps_anypoint_mode2(*pitch, yaw, roll, zoom*)

Generate a pair of X-Y Maps for the specified pitch, yaw, and roll also zoom parameters, and then utilize the resulting X-Y Maps to remap the original fisheye image to the target image.

Parameters

- **pitch** – pitch rotation (from -110 to 110 degree)
- **yaw** – yaw rotation (from -110 to 110 degree)
- **roll** – roll rotation (from -110 to 110 degree)
- **zoom** – zoom scale (1 - 20)

Returns

the mapping matrices X mapY: the mapping matrices Y

Return type

mapX

please reference: [`https://github.com/perseverance-tech-tw/moildev`](https://github.com/perseverance-tech-tw/moildev)

maps_panorama_tube(*alpha_min, alpha_max, flip_h=False*)

To generate a pair of X-Y Maps for alpha within 0 ... alpha_max degree, the result X-Y Maps can be used later to generate a panorama image from the original fisheye image.

Parameters

- **alpha_min** – the minimum alpha degree given
- **alpha_max** – the maximum alpha degree given. The recommended value is half of camera FOV. For example, use 90 for a 180 degree fisheye images and use 110 for a 220 degree fisheye images.
- **flip_h** – Flip horizontal axis (boolean True or False)

Returns

the mapping matrices X mapY: the mapping matrices Y

Return type

mapX

please reference: [`https://github.com/perseverance-tech-tw/moildev`](https://github.com/perseverance-tech-tw/moildev)

maps_panorama_car(*alpha_max*, *iC_alpha_degree*, *iC_beta_degree*, *flip_h=False*)

To generate a pair of X-Y Maps for alpha within 0 *alpha_max* degree, the result X-Y Maps can be used later to generate a panorama image from the original fisheye image. The panorama image centered at the 3D direction with alpha = *iC_alpha_degree* and beta = *iC_beta_degree*.

Parameters

- **alpha_max** – max of alpha. The recommended value is half of camera FOV. For example, use 90 for a 180 degree fisheye images and use 110 for a 220 degree fisheye images.
- **iC_alpha_degree** – alpha angle of panorama center.
- **iC_beta_degree** – beta angle of panorama center.
- **flip_h** – Flip horizontal axis (boolean True or False)

Returns

mapX mapY

please reference: [`https://github.com/perseverance-tech-tw/moildev`](https://github.com/perseverance-tech-tw/moildev)

maps_panorama_rt(*alpha_max*, *iC_alpha_degree*, *iC_beta_degree*)

To generate a pair of X-Y Maps for alpha within 0..*alpha_max* degree, the result X-Y Maps can be used later to generate a panorama image from the original fisheye image. The panorama image centered at the 3D direction with alpha = *iC_alpha_degree* and beta = *iC_beta_degree*.

Parameters

- **alpha_max** – max of alpha. The recommended value is half of camera FOV. For example, use 90 for a 180 degree fisheye images and use 110 for a 220 degree fisheye images.
- **iC_alpha_degree** – alpha angle of panorama center.
- **iC_beta_degree** – beta angle of panorama center.

Returns

mapX mapY

please reference: [`https://github.com/perseverance-tech-tw/moildev`](https://github.com/perseverance-tech-tw/moildev)

maps_recenter(*alpha_max*, *beta_degree*)

Create maps for reverse image. this can work using input panorama rotation image

Parameters

- **alpha_max** – max of alpha. The recommended value is half of camera FOV. For example, use 90 for a 180 degree fisheye images and use 110 for a 220 degree fisheye images.
- **beta_degree** – beta angle.

Returns

maps_x_reverse, maps_y_reverse

please reference: [`https://github.com/perseverance-tech-tw/moildev`](https://github.com/perseverance-tech-tw/moildev)

anypoint_model1(*image*, *alpha*, *beta*, *zoom*)

Generate anypoint view image. for mode 1, the result rotation is betaOffset degree rotation around the Z-axis(roll) after alphaOffset degree rotation around the X-axis(pitch). for mode 2, The result rotation is thetaY degree rotation around the Y-axis(yaw) after thetaX degree rotation around the X-axis(pitch).

Parameters

- **image** – source image given

- **alpha** – the alpha offset that correspondent to the pitch rotation
- **beta** – the beta offset that correspondent to the yaw rotation
- **zoom** – decimal zoom factor, normally 1..12

Returns

anypoint image

please reference: [`https://github.com/perseverance-tech-tw/moildev`](https://github.com/perseverance-tech-tw/moildev)

anypoint_mode2(*image, pitch, yaw, roll, zoom*)

Generate anypoint view image. for mode 1, the result rotation is betaOffset degree rotation around the Z-axis(roll) after alphaOffset degree rotation around the X-axis(pitch). for mode 2, The result rotation is thetaY degree rotation around the Y-axis(yaw) after thetaX degree rotation around the X-axis(pitch).

Parameters

- **image** – source image given
- **pitch** – the alpha offset that correspondent to the pitch rotation
- **yaw** – the beta offset that correspondent to the yaw rotation
- **roll** – the beta offset that correspondent to the yaw rotation
- **zoom** – decimal zoom factor, normally 1..12

Returns

anypoint image

please reference: [`https://github.com/perseverance-tech-tw/moildev`](https://github.com/perseverance-tech-tw/moildev)

panorama_tube(*image, alpha_min, alpha_max, flip_h=False*)

The panorama image

Parameters

- **image** – image source given
- **alpha_min** –
- **alpha_max** –
- **flip_h** – Flip horizontal axis (boolean True or False)

Returns

Panorama view image

please reference: [`https://github.com/perseverance-tech-tw/moildev`](https://github.com/perseverance-tech-tw/moildev)

panorama_car(*image, alpha_max, alpha, beta, left, right, top, bottom, flip_h=False*)

The function that generate a moil dash panorama image from fisheye camera. the image can control by alpha to change the pitch direction and beta for yaw direction. in order to select the roi, we can control by the parameter such as left, right, top, and bottom.

Parameters

- **image** – input fisheye image
- **alpha_max** –
- **alpha** – change the pitch direction(0 ~ 180)
- **beta** – change the yaw direction(-90 ~ 90)
- **left** – crop the left image by scale(0 ~ 1)
- **right** – crop the right image by scale(0 ~ 1)

- **top** – crop the top image by scale(0 ~ 1)
- **bottom** – crop the bottom image by scale(0 ~ 1)
- **flip_h** – Flip horizontal axis (boolean True or False)

Returns

Panorama image

please reference: [`https://github.com/perseverance-tech-tw/moildev`](https://github.com/perseverance-tech-tw/moildev)

recenter(*image*, *alpha_max*, *iC_alpha_degree*, *iC_beta_degree*)

Change the optical point of fisheye image.

Parameters

- **image** – input image
- **alpha_max** – max of alpha. The recommended value is half of camera FOV. For example, use 90 for a 180 degree fisheye images and use 110 for a 220 degree fisheye images.
- **iC_alpha_degree** – alpha angle of panorama center
- **iC_beta_degree** – beta angle of panorama center

Returns

reverse image

please reference: [`https://github.com/perseverance-tech-tw/moildev`](https://github.com/perseverance-tech-tw/moildev)

get_alpha_from_rho(*rho*)

Get the alpha from rho image.

Parameters

rho – the value of rho given

Returns

alpha

get_rho_from_alpha(*alpha*)

Get rho image from alpha given.

Parameters

alpha – the value of alpha given

Returns

rho image

get_alpha_beta(*coordinateX*, *coordinateY*, *mode=1*)

Get the alpha beta from specific coordinate image.

Parameters

- **coordinateX** –
- **coordinateY** –
- **mode** –

Returns

alpha, beta (if you get none, the coordinate is out of range that can cover)

please reference: [`https://github.com/perseverance-tech-tw/moildev`](https://github.com/perseverance-tech-tw/moildev)

5.2 Testing code

You can open the testing code on *unittest* folder.

PYTHON MODULE INDEX

m

`mmm.Moildev`, [9](#)

INDEX

A

`anypoint_model()` (*Moildev method*), 12

`anypoint_mode2()` (*Moildev method*), 13

C

`camera_fov` (*Moildev property*), 9

`camera_name` (*Moildev property*), 9

G

`get_alpha_beta()` (*Moildev method*), 14

`get_alpha_from_rho()` (*Moildev method*), 14

`get_rho_from_alpha()` (*Moildev method*), 14

I

`icx` (*Moildev property*), 10

`icy` (*Moildev property*), 10

`image_height` (*Moildev property*), 10

`image_width` (*Moildev property*), 10

M

`maps_anypoint_model()` (*Moildev method*), 10

`maps_anypoint_mode2()` (*Moildev method*), 11

`maps_panorama_car()` (*Moildev method*), 11

`maps_panorama_rt()` (*Moildev method*), 12

`maps_panorama_tube()` (*Moildev method*), 11

`maps_recenter()` (*Moildev method*), 12

`mmm.Moildev`

module, 9

module

`mmm.Moildev`, 9

`Moildev` (*class in mmm.Moildev*), 9

P

`panorama_car()` (*Moildev method*), 13

`panorama_tube()` (*Moildev method*), 13

`param_0` (*Moildev property*), 10

`param_1` (*Moildev property*), 10

`param_2` (*Moildev property*), 10

`param_3` (*Moildev property*), 10

`param_4` (*Moildev property*), 10

`param_5` (*Moildev property*), 10

R

`recenter()` (*Moildev method*), 14

V

`version()` (*Moildev class method*), 9