Building Mapping Drone

Release 1.0.0

BMD team

Dec 03, 2023

CONTENTS:

CHAPTER

ONE

BDM

1.1 default package

1.1.1 Module contents

Here is stored default point cloud to display for user. If there is no new one created.

1.2 main module

This is the main function of the whole project. Here you can find MainWindow class which purpose is to process interaction with the user and running of other scripts. There are specific params connected with animations, display options and communication.

class main.**MainWindow**(*parent=None*)

Bases: QMainWindow

Purpose of this class is to:

- process interactions between user and main_ui or options_ui;
- run the main algorithms and update the UI status as they execute;
- create loading screen animation and text messages to user;
- display effects of algorithms in interactive windows.

See functions descriptions for more detailed information.

animate_drone()

This function updates move and rotation of a drone's picture for purpose of loading screen.

apply_options()

Here is changed currently shown window from options_ui to main_ui with applying of chosen settings (if they are).

check_script_status()

Purpose of this function is to check every 1 second if algorithm ended. It is done via connection with QTimer. When process ends a few things takes place:

- turning off animation;
- checking back background to previous one;
- enabling functional buttons again.

close_app()

Save close of whole application

close_options_window()

Here is changed currently shown window from options_ui to main_u without applying of chosen settings (if they are).

cloud_display()

Here is called point_cloud_visualizer from src package to create display window for effects of OpenMVG.

loading_screen()

Target of this function is to:

- change the background image of UI;
- call set_main_window_status to hide core UI functionalities;
- create animation and start loading screen.

mesh_display()

This function creates display for mesh objects which represents the final effect.

mouseMoveEvent(*event*)

Overwritten mouseMoveEvent function from QMainWindow. Done for purpose of enabling moving whole application window.

mousePressEvent(*event*)

Overwritten mousePressEvent function from QMainWindow. Done for purpose of enabling moving whole application window.

mouseReleaseEvent(*event*)

Overwritten mouseReleaseEvent function from QMainWindow. Done for purpose of enabling moving whole application window.

options_dialog()

Here is changed displayed window from main_ui to options_ui.

run_openmvg_script(*use_option_a: bool = False*, *test_windows: bool = False*)

Here is called installation and execution of OpenMVG on chosen data or test_win.py script depends on flags.

Args:

- use_option_a (bool): Indicates forcing agreement to install of required packages for OpenMVG
- test_windows (bool): Indicates to call test_win.py from src package instead of starting installation and execution of OpenMVG

select_input_directory()

Display of QFileDialog.Options window to enable changing input directory plus checking if it is valid via QFileDialog.getExistingDirectory.

select_output_directory()

Display of QFileDialog.Options window to enable changing output directory plus checking if it is valid via QFileDialog.getExistingDirectory.

set_main_window_status(*flag: bool*)

Function which blocs and hides interface for purpose of processing algorithm and loading screen.

Args:

- flag (bool): Indicates visibility of the core UI functionalities.

start_process()

This function starts processing of input data via OpenMVG. It also checks if user set proper input and output paths for the process.

staticMetaObject = <PySide2.QtCore.QMetaObject object>

1.3 src package

1.3.1 Submodules

1.3.2 src.meshLib module

TBD SZYKRY

src.meshLib.**BPA**(*point_cloud*)

src.meshLib.**visualize**(*mesh*)

1.3.3 src.point_cloud_visualizer module

class src.point_cloud_visualizer.**Viewer3D**(*test_mode: bool*, *out_dir: str*)

Bases: object

Class to visualize point cloud via Open3D.

Args:

- test_mode (bool): rewritten parameter from run_cloud_gui function equals to its test_mode_on
- out_dir (str): rewritten parameter from run_cloud_gui function equals to its output_directory

run_one_tick()

This function contains what should be done in one iteration of visualization.

setup_o3d_scene()

Here is prepared scene for main point cloud

setup_point_clouds()

Here is created and set up object which represents chosen via test_mode_on flag point cloud.

update_point_clouds()

Purpose of this function is to prepare new point cloud position before next application tick.

src.point_cloud_visualizer.**run_cloud_gui**(*test_mode_on: bool*, *output_directory: str*)

This function runs instance of Viewer3D class in infinite loop to visualize point cloud created via OpenMVG.

Args:

- test_mode_on (bool): Indicates usage of previously prepared test point cloud from default file or user's one
- output_directory (str): String which contains path to user's point cloud used if test_mode_on = False

1.3.4 src.test_win module

The purpose of this module is to simulate script execution by waiting 5 seconds. This is done to faster test user interface features.

1.3.5 Module contents

This package contains modules important for purpose of processing mesh files, visualization and testing.

1.4 srcUI package

1.4.1 Module contents

This package is dedicated only for UI purpose. You can find here .ui files for main and options window. Additionally, here are images folder with raw and compressed images for UI. Compressed object was translated to main_ui_bit.py via PySide2 following formula from main_ui_res.qrc.

1.5 tools package

1.5.1 Module contents

In this package is placed bash script to run OpenMVG on target operating system.

CHAPTER

TWO

INDICES AND TABLES

- genindex
- modindex
- search

PYTHON MODULE INDEX