UVRSABI

Release 0.1

Graziella

Sep 22, 2022

CONTENTS

The world in which we currently live is well equipped with a variety of conventional methods for analysing and assessing the geometry and state of civil structures. Buildings that don't conform to seismic structural requirements run the risk of imposing serious damages. The vulnerability of a building to earthquake can be assessed through inspection that takes into account the expected damage progression of the associated component and the component's contribution to structural system performance. Currently, buildings and other civil structures inspections are done manually in accordance with different set of norms. Such inspections may be expensive, risky, labor-intensive and resource-intensive, and time-consuming. Hence, this project named, **UAV-based Visual Remote Sensing for Automated Building Inspection (UVRSABI)** proposes different methods that automate these inspections through UAV-based image data collection and a software library for post-processing that helps in estimating structural parameters.

Check out the *Data Collection* section for further information on how to collect the data, followed by the *Installation* and the *Instructions* pages.

Note: This project is under active development.

CHAPTER

ONE

CONTENTS

1.1 Data Collection

This page describes the data collection procedure for all the modules. We used DJI Mavic Mini for conducting the research experiments. The specifications of the UAV can be found at the official website .

1.1.1 Distance Between Adjacent Buildings

For estimating the distance between adjacent structures, the images are collected in 3 different modes: Frontal Mode, In-Between Mode and Roof Mode.

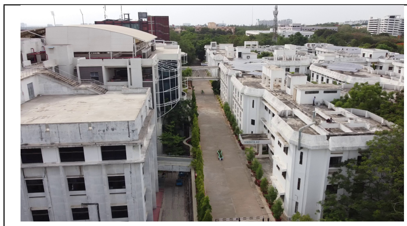


Fig. 1.1: Frontal Mode

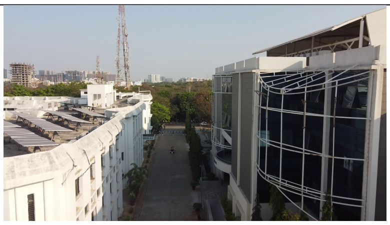


Fig. 1.2: In-Between Mode

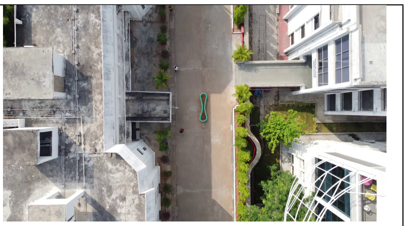


Fig. 1.3: Roof Mode

Frontal Mode

Fig. 1.1 Shows the frontal face of two adjacent buildings for which data was collected. In this mode, we focused on estimating the distance between the two buildings by analyzing only their frontal faces while flying a UAV with a forward-facing camera. This view is particularly helpful when there are impediments between the subject buildings and flying a UAV between them is challenging.

In-Between Mode

Fig. 1.2 Shows that the UAV's point-of-view for the In-between Mode. The UAV is navigated in between the subject buildings along a path parallel to the facade with a forward-facing camera. This mode enables the operators to calculate distances when the subject buildings have irregular shapes.

Roof Mode

Fig. 1.3 Shows the UAV's point-of-view for the Roof Mode. The UAV was navigated at a fixed altitude with a downward-facing camera so as to capture the rooftops of the subject buildings. The roof mode helps in tackling occlusions due to vegetation and other physical structures.

1.1.2 Plan Shape and Roof Area Estimation

For estimating the plan shape and roof area, the UAV was flown with a downward-facing camera over the roof at constant height. Our algorithm accounts for both orthoginal and non-orthogonal views of the roof.

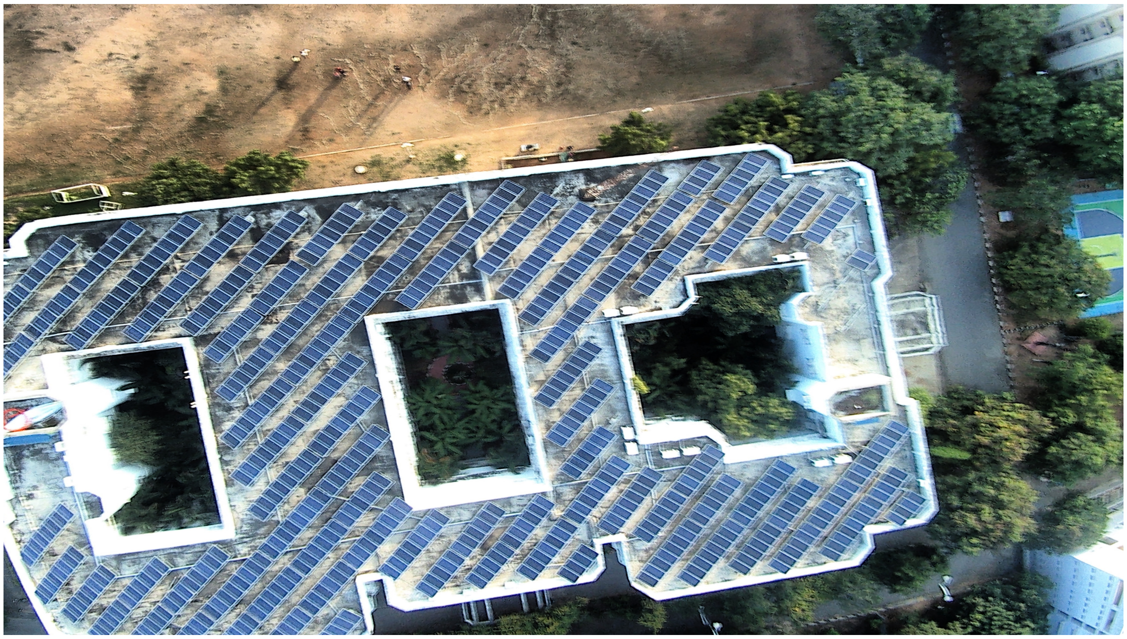


Fig. 1.4: Data Collection for Plan Shape and Roof Area Estimation

1.1.3 Roof Layout Estimation

For estimating the roof layout, the UAV was flown with a downward-facing camera over the roof at constant height.

1.2 Installation

1.2.1 Installing Conda

We are providing a conda env yaml file for easy installation of all the dependencies. Therefore, Miniconda installation is recommended for using the UVRSABI package.

1.2.2 Installing Colmap

Another dependency for the package is colmap which needs to be installed as follows: Colmap Installation

1.2.3 Installing the UVRSABI package

Clone the Github repository by running the following commands:

```
$ git clone --recurse-submodules https://github.com/UVRSABI/UVRSABI-Code.git
```

Once all the above dependencies have been installed, create a conda environment by running the following commands:

```
$ cd UVRSABI-Code/
$ conda env create -f uvrsabi.yaml
$ conda activate uvrsabi
```

We also require pre-trained weights to segment rooftops and detect objects. These can be downloaded by running the following command:

```
$ cd UVRSABI-Code/
$ ./weights.sh
```

The GUI can be launched by running the following command:

```
$ cd UVRSABI-Code/
$ python gui.py
```

The *Instructions* page can be referred to for more details on how to use the GUI.

1.3 Instructions

This page covers the instructions as to how to use the software. It is assumed that the user has followed the instructions mentioned on the *previous page* and has successfully installed the software. We make use of a GUI Application to run the software. Fig. 1.5 shows the layout of the GUI.

The user needs to select the video file and the log file to be processed by clicking on the *Video Path* and *Log File* options and then navigating to the respective files. In case of Roof Area Estimation, user needs to select the directory that contains all the test images by using the *Images path* option.

Note: The video should be collected as per the instructions mentioned in the *Data Collection* section. The log file formats have been described in the individual module sections.

Once the video file is selected, the user needs to select the module to be used for processing the video. We have made provisions to display significant intermediate results of the implemented algorithms (refer Fig. 1.6). After all the intermediate results have been displayed, the user can click on *Final Results* option to see the final results. Parallely, logs will be displayed to keep the user aware of the progress.

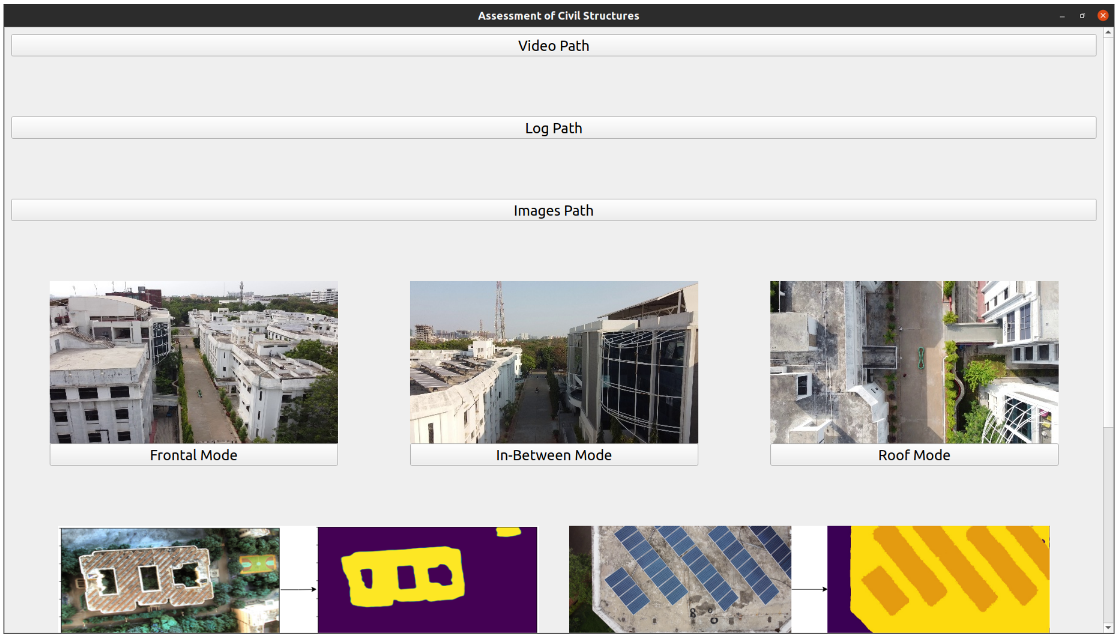
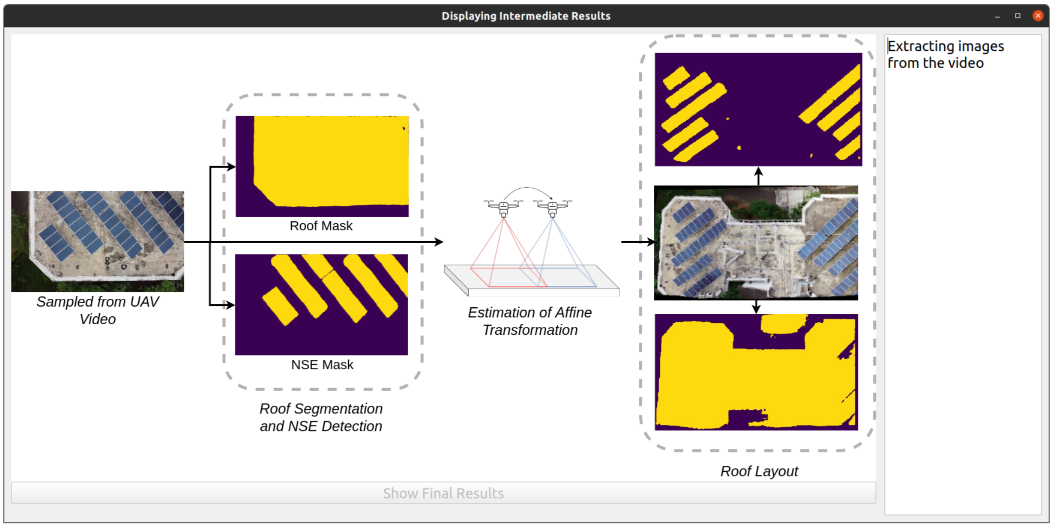


Fig. 1.5: GUI of the software


Displaying Intermediate Results
Sampled from UAV Video
Roof Mask
NSE Mask
Roof Segmentation and NSE Detection
Estimation of Affine Transformation
Roof Layout
Show Final Results

Extracting images from the video

Fig. 1.6: Displaying the architecture of the Roof Layout Estimation module. The display will be refreshed with the latest intermediate result.

1.3.1 Video and Log File Formats

- **Distance between Adajacent Buildings**: The log file should be of DJI log format. A sample log can be found here. The corresponding video should be collected as per the instructions mentioned in *Data Collection*.
- **Roof Area Estimation**: A sample log file has been provided here. The corresponding images should be collected as per the instructions mentioned in *Data Collection*.
- **Roof Layout Estimation**: This module does not require any log file. The corresponding video should be collected as per the instructions mentioned in *Data Collection*.

CHAPTER

TWO

CONTRIBUTORS

- Kushagra Srivastava
- Dhruv Patel
- Aditya Kumar Jha
- Mohhit Kumar Jha
- Jaskirat Singh
- Ravi Kiran Sarvadevabhatla
- Harikumar Kandath
- Pradeep Kumar Ramancharla
- K. Madhava Krishna