University of Sarajevo
Faculty of Electrical Engineering
Department of automatic control and electronics

ThermalMapper 3D reconstruction - Technical Documentation

Dinko Osmanković

Sarajevo, September 2012.

Contents

1 About

This document represents the technical documentation that is accompanied by the source code of the 3D model reconstruction procedure of the ThermalMapper Project[1].

The source code is divided into folders, each representing distinct module that can be used separately for different purposes:

- modelbuilder - building 3D mesh from point cloud,
- scalarfieldmapping - mapping temperature scalar field onto reconstructed 3D mesh,
- falsescans - increasing the density of the point cloud by the removal of problematic points which increases overall precision of the model,
- detection - module for detecting heat sources and heat leaks in the 3D model,
- modelcombiner - module for combining several 3D models into one,
- viewer - Qt based application for viewing, inspecting and analysing the 3D model,
- exporter - exporting the 3D model into .vrml and .x3d files,
- DATA - Data folder with sample 3d scan and resulting 3d models.

This work is an extension of the work done by Jacobs University in 6D slam: `http://slam6d.sourceforge.net/`

[1]ThermalMapper is a SEE-ERA.NET project and has the project number ERA 14/01. SEE-ERA.NET PLUS has launched a joint call for European Research projects in September 2009 in order to enhance the integration of the Western Balkan Countries into the European Research Area. `http://www.faculty.jacobs-university.de/anuechter/thermalmapper.html`

2 Installation

The environment for building the code is chosen to be Qt and standard C++. This means that all code and GUI should compile and run across all desktop platforms (Linux, Windows, Mac).

The code was developed using Ubuntu Linux (versions 10.04 and 12.04). It is recommended to have Ubuntu Linux 12.04 LTS installed on your machine.

Also, VTK 5.8 or newer is needed, along with Qt4 (4.8). In Ubuntu, you should run these console commands:

```
$ sudo apt-get install libvtk5-dev qt4-dev-tools libqt4-dev libqt4-core libqt4-gui
```

After this, you can compile the code by going into each module and running these console commands:

```
$ qmake ModuleName.pro
$ make
```

The executables are now built. Same thing can be achieved in Windows and Mac but you need to configure the environment. It should be automatic with Qt SDK. Also, you need to install VTK and add the paths into .pro files in each module.

3 Model Builder

This module is used for building a 3D mesh from the 3D point cloud recorded by laser scanner (written in .3d file). Note that all scans need to be processed by slam6D to register the point cloud into global coordinate system. This is needed for creating a combined model using all of the model reconstructed from point clouds.

After compilation a ModelBuilder executable will appear. You can use it to build a 3D model in .vtp format.

You can run it with this command:

```
$ ./ModelBuilder input_file SampleDimensions Reduction ExponentFactor output_file
```

where arguments are:

- **input_file** - 3d point cloud where the scan recording are written,
- **SampleDimensions** - value of space subdivision, the more is better although takes more time to rebuild the model (default:50),
- **Reduction** - Parameter for intelligent model size reduction (default:0.0),
- **ExponentFactor** - Factor affecting model complexity (default:0.1).

More info about the methodologies can be found in [1] and in [2].

4 Scalar Field Mapping

This module is used to map scalar temperatures acquired by thermal imaging camera (also written in .3d file) onto a newly reconstructed 3d mesh.

After compilation a ScalarFieldMapping executable will appear. You can use it to build a 3D model in .vtp format, but with added temperature information.

You can run it with this command:

```
$ ./ScalarFieldMapping input_mesh input_pointcloud output_mesh
```

where arguments are:

- **input_mesh** - input file in .vtp format containing the reconstructed 3d model,
- **input_pointcloud** - input file in .3d format containing the original point cloud with temperatures,
- **output_mesh** - output file in .vtp format containing the reconstructed 3d model with scalar temperatures mapped onto it,

More info about the methodologies can be found in [1] and in [2].

5 False Scans

This module is used to remove points that represent falsely recorder points by laser scanner. This problem is described in [3] in details.

You can run it with this command:

```
$ ./FalseScans input_model output_model number_clusters
```

where arguments are:

- **input_model** - input file in .vtp format containing the 3d point cloud,
- **output_model** - output file in .vtp format containing the 3d point cloud,
- **number_clusters** - number of clusters.

Alternatively you can use the fuzzy C means clustering by:

```
$ ./FalseScans_FCM input_model output_model number_clusters expo max_iter min_impro
```

where arguments are:

- **input_model** - input file in .vtp format containing the 3d point cloud,
- **output_model** - output file in .vtp format containing the 3d point cloud,
- **number_clusters, expo, max_iter, min_impro** - parameters of fuzzy C means clustering.

6 Detection

This module is using clustering methodology to detect heat sources and leaks. You can use it with this command:

```
$ ./LeakDetection input output number_of_clusters det
```

where arguments are:

- **input** - input file in .vtp format containing the 3d point cloud,
- **output** - output file in .vtp format containing the 3d point cloud,
- **number_of_clusters** - number of clusters of temperature,
- **det** - det = 1 for leaks of det = 2 for heat sources.

More info can be found in [4].

7 Model Combiner

This module is simply used to combine several models in .vtp format into one model. You can use it with this command:

```
$ ./ModelCombiner output input1 input2 input3 ...
```

As you can see, you can add as many input files as you can and produce one output file.

8 Viewer

In this module the viewer application is contained. This is a simple Qt based application, and by using Qt it was designed specifically for desktop cross-platform running (Windows, Linux, MacOS) as well as embedded.

After compiling the application it can be started by running it either from desktop or console. Few screenshots are given in Fig. 1.

This application allows opening the model and simple inspection. By using mouse, trackball or joystick and keyboard you can get basic interactive inspection of the model - moving, rotating etc. You can also move by arrow buttons in the GUI.

Also, by using keyboard you can activate picker tool (key P) so you can get a temperature information for every point in the model.

If you want to see the model in different colourmaps you choose from 4 different colourmaps that are implemented in the applicaton. It is required to refresh the colourmap by pressing the button.

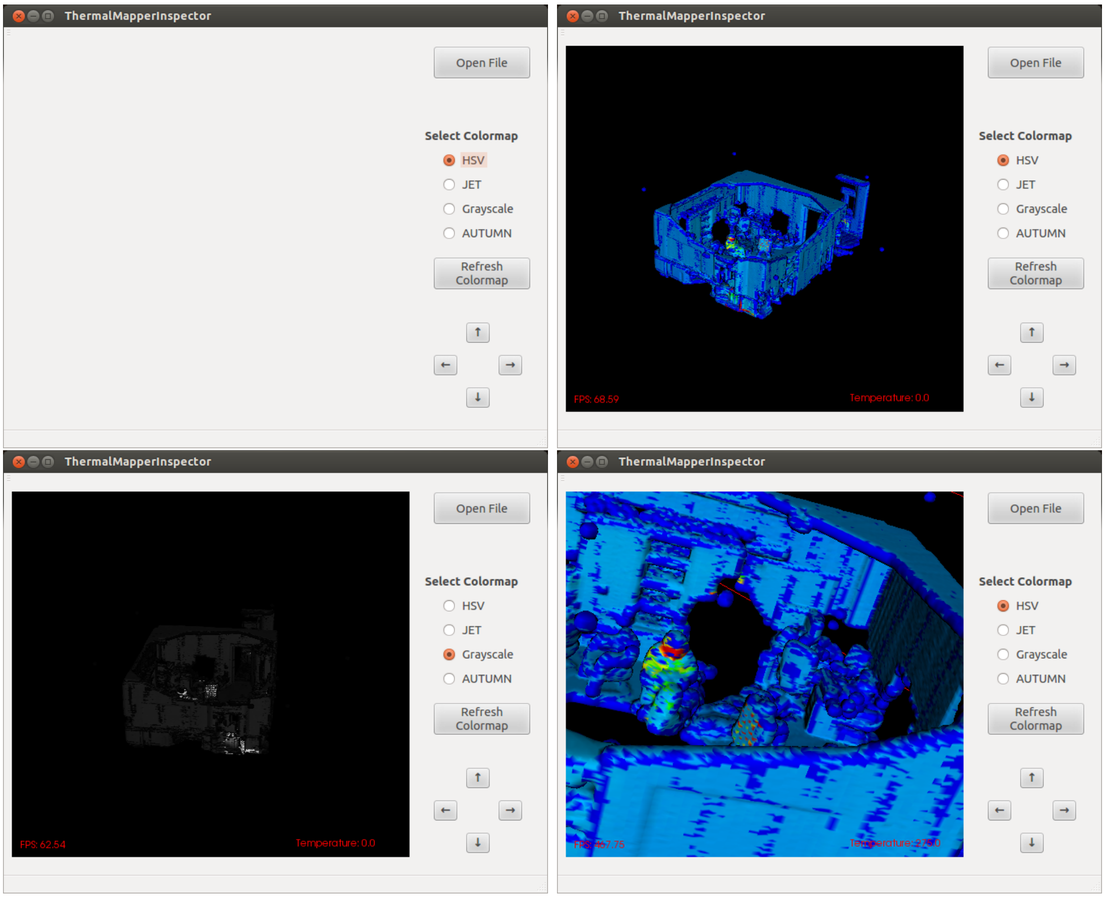

Figure 1: Screenshots of viewer application

9 Exporter

This module is used for exporting the model into VRML type formats. You can used it with this command:

```
$ ./ModelExporter input output type
```

where arguments are:

- **input** - input file in .vtp format containing the 3d point cloud,
- **output** - output file in vrml type format,
- **type** - type = x3d or vrml for selecting the output format.

Always remember the Arnold's Laws of Documentation:

1. If it should exist, it doesn't.
2. If it does exist, it's out of date.
3. Only documentation for useless programs transcends the first two laws.

Bibliography

[1] D. Osmankovic and J. Velagic. Reconstructing the 3d thermal model of indoor environment from unorganized data set acquired by 3d laser scans and thermal imaging camera.

[2] D. Borrmann, A. Nuechter, M. Djakulovic, I. Maurovic, I. Petrovic, D. Osmankovic, and J. Velagic. The project thermalmapper–thermal 3d mapping of indoor environments for saving energy.

[3] D. Osmankovic and J. Velagic. Increasing the precision of reconstructed 3d model of indoor robot environment by elimination of problematic points. In *Robot Control*, volume 10, pages 594–598, 2012.

[4] D. Osmankovic and J. Velagic. Detecting heat sources and heat leaks from 3d thermal model of indoor environment.