



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	2
2	Installation	3
3	Model Builder	3
4	Scalar Field Mapping	3
5	False Scans	3
6	Detection	3
7	Model Combiner	3
8	Viewer	3
9	Exporter	3

1 About

This document represents the technical documentation that is accompanied by the source code of the 3D model reconstruction procedure of the ThermalMapper Project¹.

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.

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

¹ThermalMapper 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.

3 Model Builder

4 Scalar Field Mapping

5 False Scans

6 Detection

7 Model Combiner

8 Viewer

9 Exporter