# Proposal for Interleague Mapping Challenge

Abstract

It is proposed that teams prove their mapping solutions installed in a virtual machine (SUN Virtual Box). At RoboCup 2010 the judges run the mapping software of the teams on their own computers (in a virtual machine) with logged sensor data (gyro and LRF) collected by the teams and gathered by NIST beforehand. The data comes from the Real Rescue Maze, Virtual Rescue Maps and from the @Home League and is gathered with robots and by hand (on a stick).

Version 2010.2 (18.06.2010) – updates are blue

See http://www.nist.gov/mel/isd/ms/mapping_challenge.cfm for the most recent version of this document.

## 1 Introduction

Providing mobile robots with algorithms to map their environment is extremely useful in many applications (USAR, Service Robotics, Military). Furthermore is mapping the corner stone for many autonomous capabilities, for example autonomous navigation.
Measuring the performance of mapping solutions is important to be able to compare different approaches and identify advances in this technology. However this is not easily done. Most often whole robotic systems are compared, as in the RoboCupRescue Robot League, where many factors have great impact on the mapping performance (sensors used, stability and speed of platform, path driven, computing capabilities on the robot, etc.). In the Virtual Rescue League mapping is even more prominent, since multiple robots autonomously have to explore an area in a coordinated way. Still the quality of the maps depends on the robots, their controllers and exploration strategies and also on their numbers. In the @Home League maps are not provided to the user. Still their quality influences the performance in some of the tasks performed.

Different applications have different requirements on the properties of maps. Sometimes coverage is the only important attribute (did you search everything?), sometimes topological correctness is used (guiding human rescue personnel through a maze) while in other cases global accuracy has to be achieved (coordinates of a victim in a collapsed building such that rescue workers can breach through the ceiling). Some of these attributes are affected mostly by the mapping algorithm (accuracy, topological consistency) while others depend more on the robots path (coverage). Still scores in different categories will be provided when scoring the maps generated.

The proposed solution to the problem of scoring the mapping algorithms is to run the mapping software on recorded data and don't allow any interaction with a human operator.

## 2 Motivation

We want a system that is:

- cheap – $ (does not involve the costs involved in having the hardware)
- cheap – man hours (applying the mapping algorithm should be very easy to do)
- repeatable (an algorithm can be run with the same data (usually) delivering a very similar result)
- independent of other robotics issues (radio, mobility, teleop or autonomy, path planning)
- independent of hardware (the only thing that is judged is the mapping algorithm)
- fair (all algorithms have the same environment)
- purely algorithmic (no human interacting with the software or improving the result)
- flexible (tests can be run anywhere)
- supports off-site participation
- can be extended to 3D

Options discussed included

- On-site time restricted mapping (give teams the data, they return the map withing x minutes/ hours)
- On-line „Web 2.0“ mapping (teams provide a mapping server where the judges upload the datasets)

## 3 Approach

The following approach is proposed: Teams are asked to provide their mapping solution to the judges. Quite often those programs rely on a multitude of external libraries and are compiled for different Linux distributions and/ or versions (or even Windows?). Therefore teams are asked to provide their software together with their OS as a Sun Virtual Box image. The judges will then start the provided systems on their (as defined in section 3) computer in a virtual machine. There they will upload the dataset, start the mapping software and gather the generated map. Those maps will then be judged using the map analysis toolkit.

There is no need to provide the source code for any software – closed source solutions are welcome to participate.

There is also no constraint regarding the programing language or environment used. All solutions are also welcome as long as they can be started from command line (see section 3.3). It is also ok for programs to open a GUI etc. – as long as they automatically close it when the mapping task is done.

Teams that already have the capability to run their mapping algorithm on the command line from recorded data should have no problem to implement the suggested solution – it should take one person not more than one day to setup the virtual machine, install the software and write a parser for (simple version of) the text format/ use the classes provided by Johannes.

## 3.1 OS/ image requirements

SUN VirtualBox will be used as visualization solution. The VM will have two gigabytes of memory and have access to both cores of a Intel Core2 Duo CPU T9600 @ 2.80GHz.

In order to make it feasible to a) download the images over the Internet and b) store them all on the computer a size restriction for those images has to be established. A maximum download size of 3GiB is suggested. Images can be created start very small initially and dynamically increase their size as needed (option "dynamic expanding store"). This option has to be used. The maximum size of the virtual hard disk should not be bigger than 8 GiB.

The OS booting in the virtual machine should:

- require no login (on text-only systems use "mappy" as the password (and username ;) )
- automatically start (without any user interaction) a console
- have the mapping program (called **generateMap** ) in its path (such that the mapping program can be started from any directory created by the judges)
- have a file called „mapping-info.txt" in the home folder OR on the desktop (if there is any) containing:
    - The team name
    - The name and version of the mapping software
    - The password of the root/ Admin user (just in case!?) (?)
    - If it is an uncommon OS, info on how to exchange data with the host system (?) OR
- the os and the user automatically logged in have to have the ability to mount/ connect an external USB hard drive (on wich the data files are and the maps will be saved)
- the os will have no connection to the internet
- use English as system language

## 3.2 Suggested OS

If you are using Linux a suggested option is to install a minimal Ubuntu. Create a new virtual hard drive (in the Virtual Media Manager) with dynamically expanding storage and a size of maybe 8 Gigabytes. Insert a Ubuntu 10.04 boot medium in the virtual CD drive and boot up. After selecting your language (English please) go to "Install Ubuntu in text mode", press F4 and choose "install console only system". When asked about partitioning do it manually. First a primary partition of 6 Gig mounted on / and then a swap partition of 10 MiB. Leave the rest of the virtual hard drive free.

When asked about the password for the user use "mappy". After the installation is complete you should have a running Linux whose virtual drive takes not more than 850 MiB.

Create you binaries on another system with the same distribution (and version). You could use a copy of the virtual machine for that. Of course you have to install all the necessary libraries and gcc, etc. on that other system. Now copy just your mapping binaries to the (other) virtual machine, for example in the folder ~/mapping . Also install any libraries that your mapping binary needs. Be sure to test the mapper!

In case your mapping software needs lots of memory you can, immediately before providing the image, delete the 10 MiB swap partition and create a 2 GiB swap partition instead.

An example image created after the suggestions in this section with a fake mapping algorithm is provided here: http://www.nist.gov/mel/isd/ms/mapping_challenge.cfm. You can use this image and

install your own mapping software in there. A ssh server is already running in this image such that one can connect using multiple sessions.

## 3.3 Running the mapping software

The following two scripts have to be installed somewhere in the path (e.g. ~/bin or /use/local/bin) and execution right has to be provided:

**mappingScript**

```
#!/bin/bash
time generateMap $1
```

**timingScript**

```
#!/bin/bash
FILENAME=$(echo $1 | sed 's/.*\//')
mappingScript $1 2> $FILENAME.timing.txt
```

The mapping software provided by the teams is available as **generateMap** and takes one command line argument – the file with the sensor data. The generated map is to be put in the folder from which the command was **executed** (not the folder of the sensor data!). It has to have the same name as the sensor data with .gtiff added. Example:

```
mkdir team1
cd team1
timingScript /media/usb-disk/sensorData/run1.log
ls
```

Output of ls:

```
run1.log.gtiff  run1.log.timing.txt
```

The time given to the software to create the map is limited. This is because a) time constrains for creating all the maps for the competition but more importantly b) because most mapping systems anyways should deliver their data close to real time. Anyways the time given for each run is double amount of time that it actually too to gather the data. So you can work with half real-time. You can use the timestamps to figure out if you have to skip scans.

Algorithms taking too much time will be allowed to finish (within reasonable time) and deducted points (%over-time subtracted  from score: eg. 50% over-time == 50% points; 100% over time 0% points)

It would be nice to have some kind of output on the console of how many scans have already been processed or a general progress (to verify that the program is not just stuck).

# *4 Data Format*

A binary file format storing all sensor data in one flexible, quickly accessible, time-stamped way is used in this competition. This is the format of the log files used at the University of Koblenz-Landau. developed by Frank Neuhaus, Denis Dillenberger and Johannes Pellenz from the University of Koblenz-Landau (Germany) have developed this sensor data log file format.
The format is provided "as-is" and is open source.

A separate document describing this format can be downloaded from the challenge web-page along with a C++ implementation and sample applications. See
http://www.nist.gov/mel/isd/ms/mapping_challenge.cfm.

The following restrictions apply to the data that will be given to the algorithms:

- The coordinate system for the measurements in IMUStateM is the same as in LaserRange2DConfigM. Meaning, that there is no rotation of the IMU with respect to the robot coordinate frame. See 3.4.4 in the log file format document for the definition of the robot frame.
- The LRF data will have at least one scan per degree of the opening angle.
- The LRF data will have an opening angle not smaller than 180 degrees.

## 4.1 Result Format

The resulting geoTIFF file should be formatted according to the standards of the real rescue league. It would be really nice if not only the team name but also the name of the sensor data file provided would be written in the top part of the tiff.

The world coordinate system always has its origin (0,0) in the robot start pose (first sensor data). The geoTIFFs specify the position of the top left corner of the image in this coordinate system in m. The scale is the resolution of your image in m – a 0.05 means that one pixel is 5cm wide. The orientation is relative to the start orientation looking north (up).

See http://www.remotesensing.org/geotiff/spec/geotiffhome.html (especially 2.5 and 2.6) for more info. Instead of embedding the information of the geo-reference in tags in the tiff-file you can also create a twf-file ( http://en.wikipedia.org/wiki/World_file ).

The map will be treated as having just three different colors: Free (white), Unknown (gray (127, 127, 127)) and Occupied (black). Maps containing other colors will be transformed accordingly by the judges (e.g. Green to white, robot path to white). Shades of gray will be mapped to one of the three allowed colors based on appropriate thresholds. For optimal results algorithms should thus return maps in just these three colors.

# *5 Preliminary Schedule*

- 10. June: Call for participation
- 11. June: Provide example data from the ASTM international standard test method for emergency response robots (random maze apparatus) and from a virtual map with added fiducials
- 14-17. June: Attempt to give everybody interested a test-score by downloading their image from their server and running one map
- 1st day of Competition (21. June):
  - Judges will download/ get the images from all teams
  - Judges will collect a (maximum) one-page-flyer shortly describing the mapping algorithm from each team
  - Collect data during the runs
  - Generate maps from collected data
  - Generate maps from (prepared) virtual runs
  - Score those maps and give this feedback to the teams
- 2nd day (22. June):
  - Collect data during the runs

- 3rd day (23. June):
  - Collect data during the runs
  - Generate maps from 2nd and 3rd day runs
  - Final Scoring for preliminaries
  - If possible gather data from @home – visit in the shop
- 4th day (23. June):
  - For those that made it to the finals download new version of mapping software (if available)
  - Generate maps from collected data (4th day, virtual, @Home)
  - Score
  - Done

## 6 Scoring of the Maps

The maps will be scored according to the fiducial approach (where fiducials are available) (50%) and using expert judges (50%).

The fiducial approach works by finding landmarks (parts of barrels) in the maps and using those to determine coverage, global accuracy, local consistency and other values which are used together to generate a score.

## 7 Further Use of Mapping Algorithms/ Maps

It is intended to use the virtual machines as well as the maps generated for research on map evaluation. Maps generated by the provided algorithms might be published in this context – especially also maps which are bad or broken maps.

Teams can request that their names may not be mentioned in published papers

Upon request team names may not be mentioned in those papers (specify in mapping-info.txt and/ or by sending E-Mail to s.schwertfeger@jacobs-university.de ).

Maps generated during the competition and the algorithms might go into in a repository for further use by the ASTM Committee.