



Graphical User Interface for the URUS project

Eduard Trulls Fortuny



Abstract

The aim of this report is twofold. First, to present the Graphical User Interface (GUI) developed at IRI for monitoring robotic experiments in the context of the EU project URUS, its capabilities and intended use, and its place within the communications workflow, based around middleware YARP. Second, to introduce the GIS map, extended by us to full 3D, of the experimental area, known as Barcelona Robot Lab, a $10.000m^2$ section of the Campus Nord at UPC, Barcelona, Spain; this 3D map is extensively used in the GUI but has been found useful out of its scope.

Institut de Robòtica i Informàtica Industrial (IRI)

Consejo Superior de Investigaciones Científicas (CSIC)

Universitat Politècnica de Catalunya (UPC)

Llorens i Artigas 4-6, 08028, Barcelona, Spain

Tel (fax): +34 93 401 5750 (5751)

<http://www.iri.upc.edu>

Corresponding author:

E Trulls

tel: +34 93 401 5780

etrulls@iri.upc.edu

<http://www.iri.upc.edu/people/etrulls>

1 Introduction

This work is part of the EU project Ubiquitous Robotics in Urban Settings (URUS, IST-045062) [3]. This research project is aimed at the improvement of quality of life in urban areas through the deployment of network robot systems which should cooperate with people in tasks such as guidance or assistance, transportation of goods and/or people, and surveillance. To this end a large section of the Technical University of Catalonia's (UPC) Campus Nord was outfitted as an experimental area for robotics research. This installation covers over $10.000m^2$ and is equipped with wireless coverage and a network of over 20 video cameras, and will be referred to from now on as Barcelona Robot Lab.

Given the number of partners involved in the URUS project a common communication protocol was required in order to ease software integration. The resulting specification [4] was based on YARP [6], a free, open-source set of libraries, protocols and tools aimed at decoupling the transmission of information from the particulars of devices and processes in robotic systems. The GUI acts as one of these software components, requesting information from other modules, be it from the servers at the central station or the mobile robots via wi-fi, as necessary. The reader may find a more thorough description of the architecture in the experiments' progress report files [7, 5].

One of the main challenges to be addressed in robotics in urban settings is the non-ideal nature of these settings. The experimental area at the Barcelona Robot Lab, placed in a hilly area on the northwesternmost side of Barcelona, very close to the mountain range of Collserola, serves as a good example and testing ground. Featuring many slopes and staircases, the difference in elevation can reach figures as high as ten meters within the area of interest. The uneven nature of the ground is then a critical element and a 2D map, sufficient to address many problems in robotics, may not be enough in our case. The interface was designed in 3D, and a 3D map of the experimental area was built accordingly, as part of the interface itself. This map has been found useful outside of this scope, as for instance the reference for 3D localization algorithms (work is ongoing on this front as of the time of this writing). This map will be presented in further detail in section 2.

Given the complexity of the underlying structure (the number of processes running at any given time during a major experiment and the resulting communications workflow), this software's primary purpose is for general monitoring during experiments and rehearsals. It should present a current overview of the task to carry out, the number of robots involved and their respective status, and their position according to localization routines, as well as the result of algorithms for the tracking of people involved in the experiment. It should allow for easy access to live video feeds from the camera network, and from the mobile robots' cameras, if any, and depending on the wireless network's load. As such, it can be of use to URUS engineers as well, but it does not attempt to centralize all information and, most notably, does not allow for direct control of mobile robots, which remains a low level task. The GUI will be described in further detail in section 3.

Lastly, this is a work in progress, and a number of features regarding interaction with URUS services are yet to be implemented or in an embryonic state.

2 3D Map

A basic requirement for the development of this interface was the availability of a GIS map of the experimental area. The campus, originally built during the 1990s, has since undergone a number of transformations, and the original architectural drafts, some of which could be obtained by IRI, were not of much use for our purposes. A GIS map was developed by URUS partner Agència d'Ecologia Urbana de Barcelona, based on an exhaustive set of in-site measurements obtained by measuring tape, and later refined by members of the same agency and IRI with measuring tape and a laser distance meter, with the aim to better define certain areas of the map (most notably the FIB square), include elements such as trashcans and other objects which may pose no problems to humans but are critical for robot navigation, and to incorporate full 3D information. Said measurements were obviously all relative and as such carried a large degree of ambiguity, and were later patched up by hand to obtain a coherent 3D model. A 2D version derived from the original GIS map has been used extensively for localization algorithms and monitoring purposes, and is shown in figure 1. This map's specification can be found in [2].

The 3D model was built using Sketchup, a 3D design tool by Last Software acquired in 2006 by Google, available for OSX and Windows, for which a light version without exporting capabilities can be downloaded and used free of charge. The Pro version, featuring the exporting capabilities necessary to later convert the 3D model into a non-proprietary format readable by the GUI, is offered in the form of a full version currently limited to a very generous 8 hours. This choice proved critical, as most 3D design tools feature a very steep learning curve and do not lend themselves to this kind of work. Sketchup, originally meant for architectural design and with a very streamlined and intuitive interface, simplified considerably the task of building a 3D map from a large number of disjointed measurements. The map can be supplied in a proprietary Sketchup `skp` format, *de facto* exchange standards `3ds` and `obj`, and OpenSceneGraph proprietary format `osg`.

The final version of the map features the following information:

- Buildings.
- Slopes.
- Stairs, when appropriate. Only those which may be seen as ascending from within the experimental area have been considered, as such are what a robot's sensors may see.
- Stone and wooden benches.
- Large jardinières, elevated 20 to 40 cm from the floor, prevalent throughout the campus, many of which feature trees at roughly regular intervals.
- The FIB square's four tree bases, cast in stone and raised from the floor a variable height of about 15 to 40 cm.
- Miscellaneous objects such as lightposts, bicycle stands, large plant pots, trashcans, water fountains, large bulletin boards, and plaques bearing the names of buildings.

Particular attention was paid to elements which may not be traversable by robots such as those available within the project's context, and to features which may provide a good point of reference for laser range finders, as this map should eventually be used by 3D localization algorithms. Glass has been modeled as solid, as it has been observed that it may be transparent

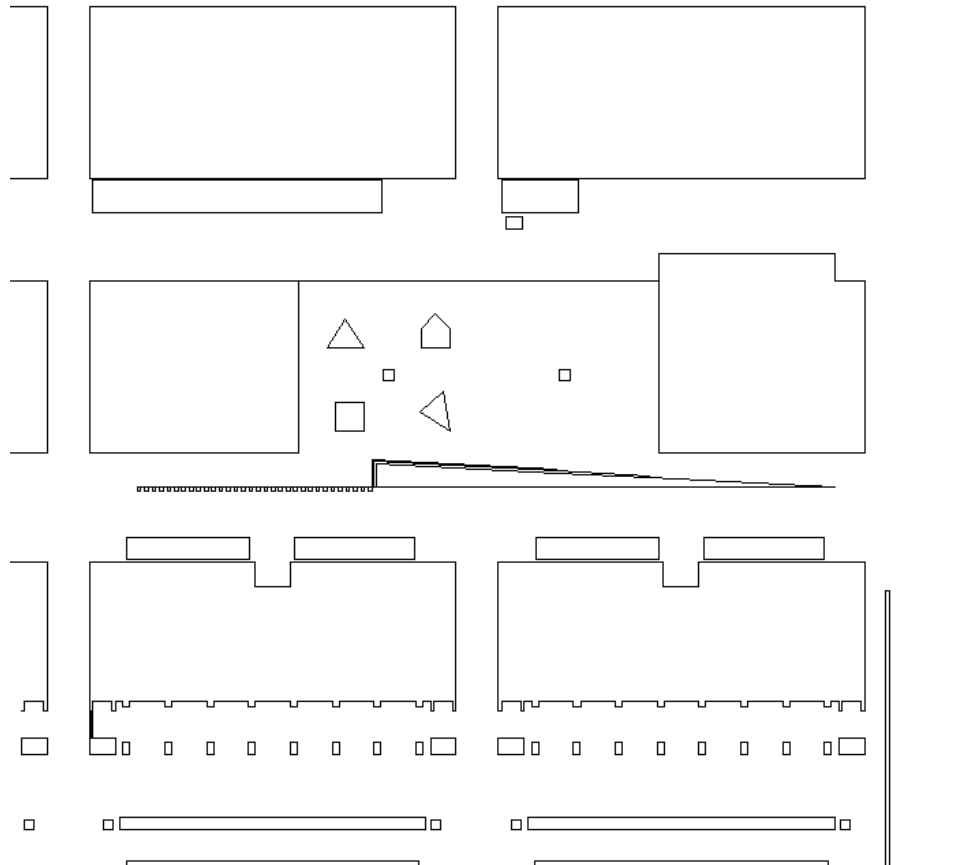


Figure 1: The experimental area covers only the southeasternmost part of the campus. Buildings are placed over a four-row, six-column grid and labeled according to row (letters from A to D, bottom to top) and column (numbers from 1 to 6, left to right). The buildings on the picture correspond to A5, A6, B5, B6, C5 and C6. The other main features found on this section of the campus are the terrace at the bottom of the map, the FIB (Computer Faculty) square, and a promenade with another terrace above it, between rows B and C.

or opaque for different laser range finder models. The southernmost terrace, the FIB square, the promenade above the square, and the easternmost passageway were defined in greater detail than other zones, as these are of the foremost importance for the experiments.

Most of the seemingly flat surfaces found on the campus are not actually flat, but feature slight, uneven slopes. Accurate registration of these variations would require the work of professionals. These were considered only when necessary, as for instance for the FIB square or in some of the platforms along the easternmost passageway.

All objects considered dynamic or liable to be moved, such as garbage containers or trashcans not nailed to a wall or to the ground, were not included in the final design. Neither were trees, as most feature highly irregular, curved trunks, hard to characterize. This may be addressed in the future, according to need.

Figures 2 and 3 display a view of the general model, along with actual photographs of the campus.

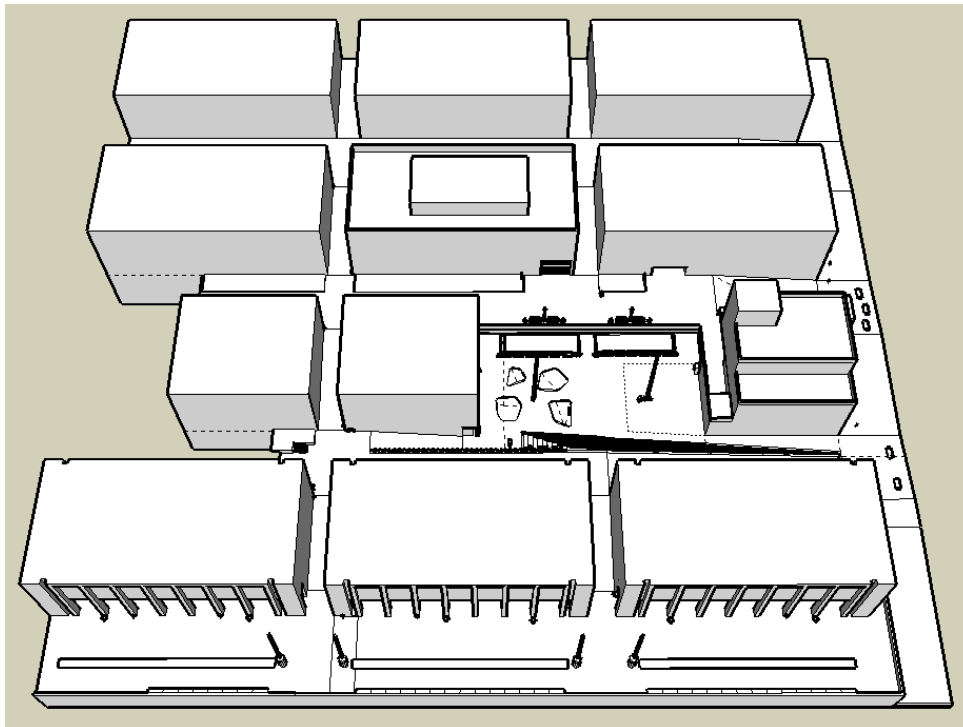


Figure 2

2.1 Other maps of the experimental area

There are two other maps or sets of maps of the experimental area currently in use or in active development. The 2D version of the GIS map, for use mostly internally at IRI, has been presented already. Another set of maps, based on properly-registered high-resolution laser range finder measurements, is currently in development as part of URUS WP4, which should provide a number of maps for different purposes (points, lines, planes, traversability, etc.), as referenced in [8].



Figure 3

Preliminary results show a remarkably good correspondence between the 3D map and the laser-based map, as of now available as a cloud of points. These clouds can be displayed offline in the GUI, as explained in the next section. Early, qualitative results are shown in figure 4.

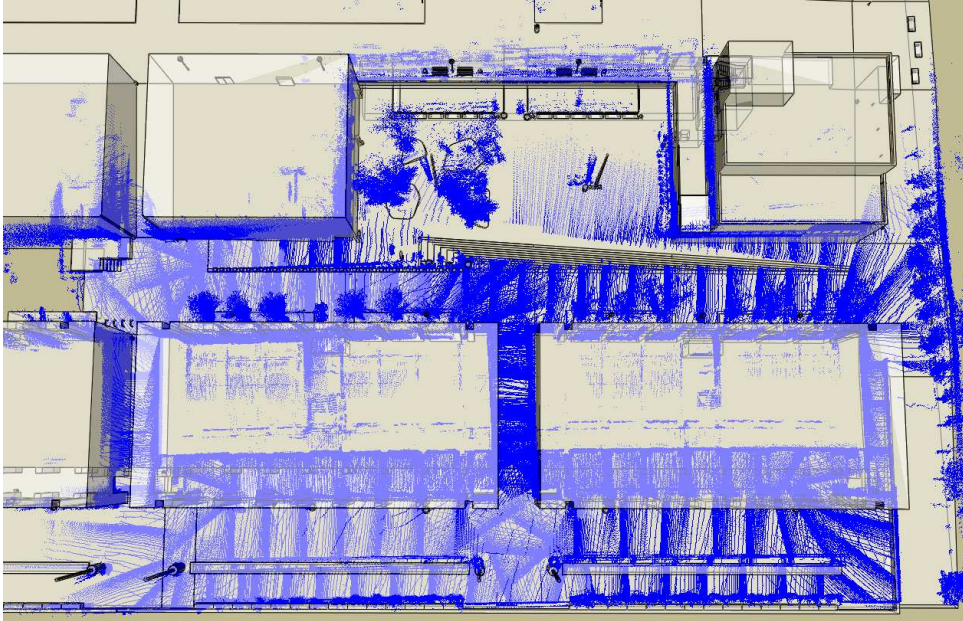


Figure 4: 3D point map crudely registered over the 3D map. Early analysis suggests the accuracy is within the range of one to two meters.

2.2 Places of interest

Of particular interest for the experiments is the definition of symbolic names for places of interest in Barcelona Robot Lab. These areas are defined as a position and two radii, one for stationing the robot and another marking the limit for interaction, both robot-dependent. This feature is yet to be implemented in the GUI, as the final list, with precise coordinates, is to be defined sometime during 2009. It will be available as part of the WP4 specification [8]. As of the time of this writing, places of interest are defined as in figure 5.

3 The Graphical User Interface

The GUI has been built using the Qt framework, free of charge for development of open source software. 3D development is based around the OpenSceneGraph (OSG) graphics toolkit. Both libraries are written in C++ and provide many useful tools; OSG routines can be embedded into Qt windows as widgets. Both libraries are cross-platform as well, and it should be possible, with some effort, to compile this software on Linux, Windows and OSX; all development and testing has been carried out on Linux.

Figure 6 shows a screenshot of the interface in action. The scene can be panned, rotated and zoomed in or out. Interaction is mostly carried out through the widgets on the left, which may be docked in either side of the application, floated as independent windows, or hidden. The camera may be attached to moving robots to follow their trajectory. Most elements can be

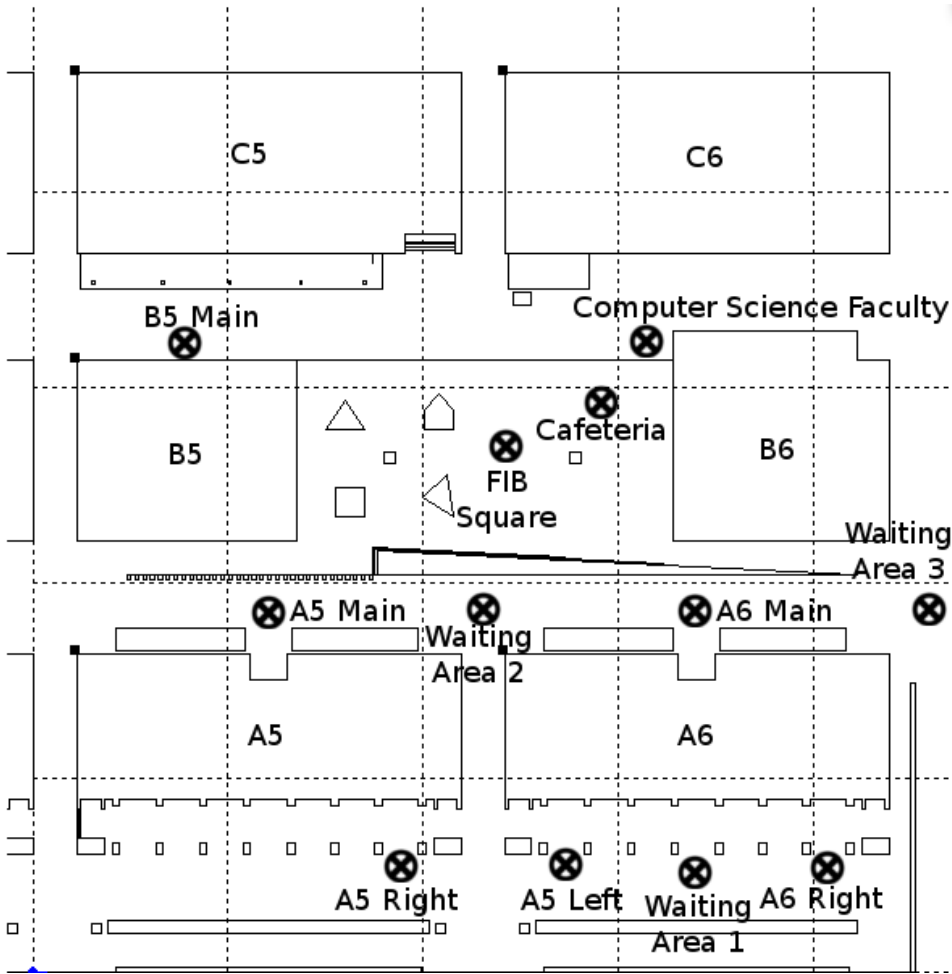


Figure 5: Preliminary version of the semantic map.

hidden to not clutter the scene, and it is possible to try to reestablish YARP connections from the interface itself in case of need. The map will render itself transparent if any robot enters a building due to inaccurate localization; transparency can be forced with the appropriately labeled button. Controls are self-explaining.

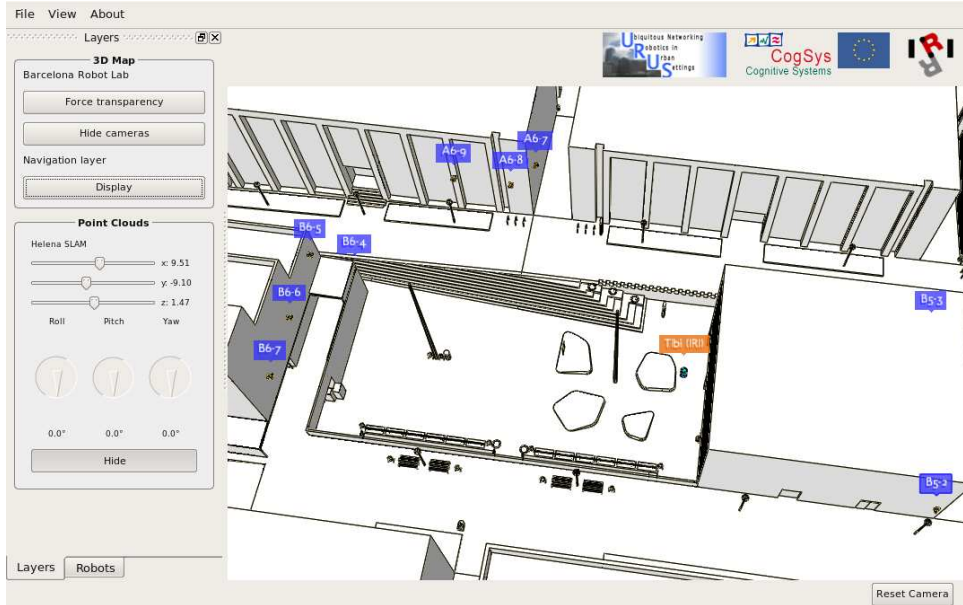


Figure 6

The interface can display live video streams from the static cameras, as seen in figure 7. These cameras employ a proprietary compression format [1], so the stream needs to be decoded beforehand. The solution currently in use involves decoding the stream on the servers for transmission to the interface in raw RGB. This may change in the future due to bandwidth constraints; a possible solution involving JPEG compression is outlined in [9]. It should be trivial to extend this feature to retrieve images from mobile robots, but this service is not defined for the time being.

Upon start-up, a number of configuration files (see 3.3) are parsed and the map is built according to specifications. Likewise, the software creates the appropriate instances for cameras and mobile robots. URUS localization algorithms are currently 2D-based, so robots may be able to localize themselves in the XY plane but know not of height; their position is defined unambiguously since no region in the experimental area is traversable at different heights. More complex situations may require another approach. The interface calculates the closest vertical intersection against a one-layer version of the map including only traversable regions to place robots correctly over the 3D map.

Point clouds such as those coming from laser range scanners can be displayed over the 3D map as an offline process, as seen in section 2.1. This may be extended in the future to cover WP4 maps in other formats. Interaction with URUS services is further described in the next section.

3.1 Integration with URUS framework

This module has ties with WP2 (localization and navigation), WP3 (cooperative perception), WP4 (mapping) and mission monitoring (WP6: task allocation, and the robot supervisor). It

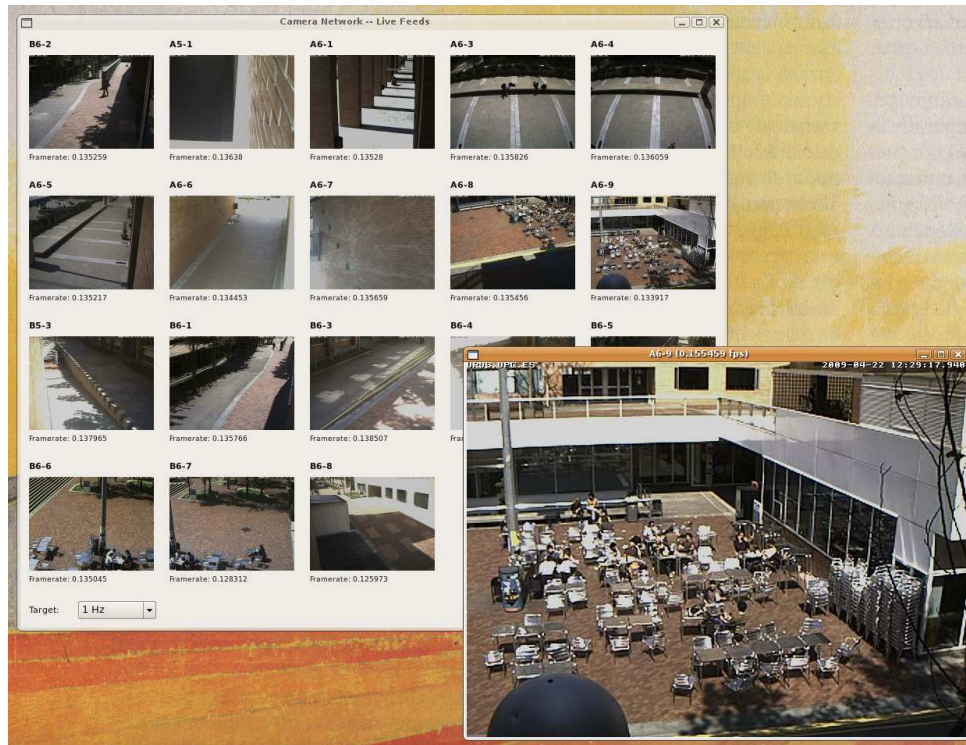


Figure 7

should be able to retrieve and display the position of robots and people as determined by WP2 (localization) and WP3 (detection and tracking), and a number of maps, physical and semantic, from WP4. At the time of this writing, the localization service has been integrated and tested; robots are automatically placed as per the algorithm's instructions and localization uncertainty is displayed as an ellipse over the robot's position. People detection and tracking remains an open issue, and current discussion is taking place regarding which WP3 interfaces can be open for external services such as the GUI. Mapping is integrated as far as offline retrieval of 3D point cloud maps. The robot supervisor's interface is yet to be integrated. It is expected that interfaces will change as the result of integration meetings throughout 2009, and other needs may be identified in the future as experiments are rehearsed and redefined.

3.2 Requirements and compilation

Compilation takes advantage of the `qmake` tool for `Makefile` generation; Qt configuration as well as pertinent files and libraries are specified in two `.pro` files.

The main dependencies are Qt4, including development packages, and OpenSceneGraph 2. For recommended library versions as well as sub-dependencies please refer to the latest build's instructions. The program has been developed and tested on several versions of Ubuntu up to 8.04.

3.3 Usage

Experiment specifics are described through GUI-specific XML files. The look-up file `settings.xml`, at the root folder, determines further XML files for the configuration of the map files required by the interface, the robots involved in the experiment, and the static cameras, as well as a boolean parameter which serves to indicate whether the software is being executed inside or outside the URUS network. This is required as for instance the cameras' IPs are in different domains in each case.

The XML file for map configuration is as follows. Faces and edges are split for convenience. An additional model indicating traversability (reasonably flat surfaces) is used for a number of internal operations. Only one layer may be specified (successive ones will be ignored), which may be associated to an undetermined number of point cloud files.

```
<!DOCTYPE maps_spec>
<map_layers>
  <layer shortLabel="BCN Robot Lab" longLabel="Barcelona Robot Lab"
    displayableFaces="models/bcn_robot_lab_200901_faces.osg"
    displayableEdges="models/bcn_robot_lab_200901_edges.osg"
    traversable="models/bcn_robot_lab_200901_trav.osg"
    x="0" y="0" z="0" yaw="" pitch="" roll="">
    <pointCloud label="Helena SLAM" source="data/upc_helena_slam.dat"
      x="9.97" y="-9.64" z="2.52" yaw="-0.94" pitch="1.72" roll="0"
      pointsR="255" pointsG="0" pointsB="0" pointsSize="1" pointsSmoothing="0"
      posesR="0" posesG="0" posesB="255" showPoses="0" posesSize="10"
      show="1" />
  </layer>
</map_layers>
```

The XML file for robot configuration is as follows. The parameter `testDevice` instantiates a robot which can be moved around by the user, for testing purposes. Note that there is no indexed list for URUS partners or robots as of the time of this writing.

```
<!DOCTYPE robots_spec>
<robots>
  <!-- startZ necessary only where conflict (multiple z-layers) -->
  <robot id="r0" partner="0"
    shortLabel="Tibi (IRI)" longLabel="Segway Tibi (IRI)" model="models/segway_2_we.osg"
    width="" length="" height="1.5" radius=""
    startX="0" startY="0" startZ="" startHeading="0"
    testDevice="0" />
  <robot id="rtest" partner="0"
    shortLabel="Test Device (IRI)" longLabel="Test 1 (IRI)" model="models/segway_2_we.osg"
    width="" length="" height="1.5" radius=""
    startX="0" startY="-3" startZ="" startHeading="90"
    testDevice="1" />
</robots>
```

The XML file for camera network configuration is as follows. Several parameters are of use for drawing the cameras over the 3D model only; note that these coordinates are crude approximations. The boolean parameter `operative` determines whether the interface should attempt to connect to the camera streams in addition to displaying their 3D models over the map.

```
<!DOCTYPE camera_network_spec>
<cameras>
  <camera shortLabel="B6-2" longLabel="B6-2"
    externalIp="*.*.*.*" internalIp="*.*.*.*"
    x="58.59" y="-4.3" z="6.42" headingSupport="90" headingHead="-75" pitch=""
    yarpServerPort="/mx_server/b6_2" yarpClientPort="/mx_client/b6_2"
    operative="1" />
  <camera shortLabel="A5-1" longLabel="A5-1"
    externalIp="*.*.*.*" internalIp="*.*.*.*"
    x="-24.8" y="-36.1" z="6.42" headingSupport="-90" headingHead="-75" pitch=""
    yarpServerPort="/mx_server/a5_1" yarpClientPort="/mx_client/a5_1"
    operative="1" />
</cameras>
```

Command line options are as follow:

```
urusGui [settings.xml]
```

A YARP server must be reachable by the interface. As more than one instance of the software may be running at the same time on the same network, a random, disambiguation key is attached to the GUI's label and so to the YARP ports open by every instance of the software. This is possible since any communication process involving the interface is initiated by the interface itself.

3.4 Known issues

See latest build's documentation.

4 Future work

As final results from WP4 are available it will be possible to evaluate the quality of the 3D map. This map may be better suited to certain kinds of algorithms than laser-range based maps, as its description is very compact and all dynamic and movable elements have already been discarded. For instance, it is intended to extend current 2D localization algorithms to full 3D using this map as reference. As such, it may be interesting to refine it according to WP4 results.

It has been already stated that many URUS services will be defined and/or implemented during 2009, when the project's experiments are scheduled to take place. Some services yet to be integrated in the GUI have already been listed in detail, and others may be identified in the future.

A First Appendix

The source code should be available to URUS partners on SVN. It has been tested on 32-bit Ubuntu Linux 8.04. It remains a beta version, and in development.

References

- [1] http://developer.mobotix.com/mobotix_sdk_1.0.1/docs/why.html.
- [2] J.M. Mirats-Tur A. Corominas-Murtra. Map Format for Mobile Robot Map-Based Autonomous Navigation. In *Internal Report, URUS*, 2007.
- [3] J. Andrade-Cetto A. Sanfeliu. Ubiquitous networking robotics in urban settings. In *Proceedings of the IEEE/RSJ IROS Workshop on Network Robot Systems*, pages 14–18, 2006.
- [4] M. Barbosa. URUS System Description Files. In *Internal Report, URUS*, 2008.
- [5] Joao Sequeira (editor). Experiment 2: Review. In *URUS Deliverable D9.1*, 2008.
- [6] L. Natale P. Fitzpatrick, G. Metta. Towards Long-Lived Robot Genes. In *Robotics and Autonomous Systems*, 56, pages 29–45, 2008.
- [7] Contributing partners. Experiment 1: Guiding and Transportation Review Report. In *URUS Deliverable D8.1*, 2008.
- [8] E. Teniente. WP4 Service Definition. In *Internal Report, URUS*, 2008.
- [9] E. Trulls. Image compression for efficient transmission within the URUS communication protocol. In *Technical Report, IRI-TR-09-02*, 2009.

Acknowledgements

This work was partially supported by IST-045062 project of the European Community Union, CICYT project DPI2007-614452, and Universitat Politècnica de Catalunya (UPC).

The source code features contributions from several IRI members regarding integration of YARP software modules. The elaboration of the 3D map was a joint effort with URUS partner Agència d'Ecologia Urbana de Barcelona.

IRI reports

This report is in the series of IRI technical reports.

All IRI technical reports are available for download at the IRI website

<http://www.iri.upc.edu>.