RAWSEEDS
Robotics Advancement through Web-publishing
of Sensorial and Elaborated Extensive Data Sets

Raw seeds can be consumed as they are...
or be the start for the growth of new results.

RAWSEEDS
WorkPackage 1

Deliverable D1.1
Roadmap of Indoor Activity

author: Domenico G. Sorrenti (WP-1 Leader)
internal reviewers: C. Stachniss (ALU-FR), Neira J. and Tardós J. D. (UNIZAR)
external reviewer:

contributors:

Burgard Wolfram, ALU-FR
Caccia Claudio, UNIMIB
Fontana Giulio, POLIMI
Matteucci Matteo, POLIMI
Sorrenti Domenico G., UNIMIB
Stachniss Cyrill, ALU-FR
Tardós Juan Domingo, UNIZAR

Table of Contents

1. About this document

This document has the purpose of defining the framework for the technical work of the RAWSEEDS project, for what concerns the indoor scenarios (please see the following section for a definition of *scenario*). This document collects high-level information only, without going into the implementation details, unless required to clarify specific points.

1.1 RAWSEEDS terminology

The aim of the RAWSEEDS project is to produce and make easily available through the Internet a **benchmarking toolkit** (or simply **toolkit**). This includes all the elements needed to test algorithms designed for the problems of mapping, self-localization or SLAM (Simultaneous Localization And Mapping). RAWSEEDS is specifically oriented towards robotic systems, although its toolkit could be also useful in different contexts (e.g., surveillance); the use of the toolkit should greatly reduce the time and effort needed for the successful development of innovative algorithms or products, by eliminating the need to set up costly data acquisition campaigns.
The RAWSEEDS toolkit will be based on exclusively real-world data (i.e.,, there will not be simulated data), and will include instruments (called **Benchmark Problems** and **Benchmark Solutions**) to test, rate and compare different algorithms. Along with these, the toolkit will include readily usable examples of state-of-the-art algorithms, to be used as examples in the design of new algorithms.

The foundations for all the work of RAWSEEDS are the data sets, also called *datasets.* In the rest of this document, the term **dataset** will be used to identify the set of synchronized data streams obtained by recording the output of the sensors mounted on a robot when the robot explores an environment. A single instance of this exploration procedure will be called a (data-gathering) **session**. A session can be performed by splitting it into the exploration of different (but strictly related, e.g., adjacent in space or time) environments, thus generating multiple datasets; in this case the single explorations will be called sub sessions.
Alternative, but "real" datasets (i.e., composed of sensor measurements, not calculated or simulated data) can be obtained from a given one by discarding part of the data: for example by omitting the data generated by one or more sensors or by performing under-sampling of the data. This can be useful to test the performance algorithms which use different sensor sets or to simulate the effect of sensors with lower performance than the ones currently used. These datasets will be called **derived datasets**.
The complete set of conditions defining a single data-gathering session will be called **scenario**. A scenario will be then defined by information such as: hardware setup, physical location of the experiment, the presence or absence of people, lighting conditions, the kind of terrain, and so on. Please note that for the same location and hardware setup different scenarios can be defined.

Project RAWSEEDS will gather two types of datasets: *indoor* datasets and *outdoor* datasets. The former have the objective of covering the typical environments encountered by robots operating in locations where surrounding walls and roofs are present: e.g., homes, industrial plants, offices, warehouses. In this kind of environment artificial lighting is usually present, possibly with sunlight entering windows or other openings, and the terrain is generally even (though not necessarily flat everywhere: ramps or stairs are common). Currently (as in the past history of robotics) most research or commercial robots are designed to operate in indoor environments. Therefore, indoor datasets are the most important ones and also the most used ones. On the other hand, this means that several indoor datasets (albeit usually with a much lower quality compared to the ones that RAWSEEDS will make available) are already available to the community. On the other hand, the second type of datasets (i.e., outdoor datasets) is extremely rare to find: partly because outdoor robotic applications are still rare, and partly because setting up a session of outdoor data-gathering with mobile robots is time-consuming, difficult and costly. Thus, the datasets provided by RAWSEEDS will address a serious stumbling block to the development of outdoor robotic applications.

It is important to note at this point that the data with which the RAWSEEDS toolkit will be based will all be verified and *validated*, i.e., their quality and correspondence to requirements will be explicitly certified by the RAWSEEDS Consortium with reference to specific, published standards. Moreover, together with each of the datasets, RAWSEEDS will provide the associated **ground truth**. This is a set of information accurately describing the real environments explored by the robots and the trajectory followed by the robots. Ground truth is used as a reference against which the results obtained by applying algorithms (e.g., for mapping) to the datasets can be evaluated. None of the real-world datasets currently available to the robotic community have been validated as described above, nor do they provide a ground truth or a ground truth obtained with an independent device, some dataset uses as ground truth the output of the currently-best algorithm. This will be the very last resort for providing ground truth in RAWSEEDS datasets.

This document describes the activities related to the indoor datasets only: both for the generation of the datasets (which requires specific hardware and software architectures) and for the generation of the parts of the RAWSEEDS toolkit which are based on those datasets. Exploration of these topics for the outdoor datasets is left to Deliverable D1.2.

2. Project overview

The work of the RAWSEEDS project can be split into different *aspects*, each of which requires a specific design phase preceding the implementation phase. In the context of this section, the word "aspect" is used as a generalization of the concept of "task". Some of the above aspects, notably those requiring the acquisition of special equipment, need to be defined well in advance, to allow for the delivery time of all the parts to be used. This definition phase is part of the work of WorkPackage1 (WP-1).
Below is a table of all the aspects of RAWSEEDS' work concerning the indoor activity. To each aspect is associated a brief note, describing its advancement status. The status ranges from "open", for aspects where everything except a basic description is absent, to "closed", for aspects where every detail has been settled. Of course even "closed" aspects could be re-evaluated and possibly modified if such a need emerges from the subsequent activities of RAWSEEDS.
Please note that the following table includes two different categories of aspects: those that are an integral part of the work of WP-1 and those that lie *outside* of WP-1. The latter are of aspects which pertain to WorkPackages that at the moment are not completed or even not yet started, but the activities of which have to be defined and planned by WP-1. As a consequence, the meaning of the status column is different for the two categories of aspects: for the first category, it reflects the actual state of accomplishment of the set of tasks concerning an aspect; for the second, it describes the advancement of the planning of the aspect.
In the following sections of this document each aspect outlined in the table will be described in detail. Please note that each element of the following table is associated to an item in the Table of Contents of this document, to facilitate consultation.

Aspects of the indoor activity of RAWSEEDS	**Advancement status**
Hardware and software setup	
robot platform	closed
sensor systems	closed
Setup of the data-acquisition robot	almost closed
Indoor scenarios	
location	closed
scenarios	closed
data acquisition methods	closed
data-gathering sessions	almost closed (session schedule not definitive)

Data validation	
evaluation criteria	closed
acceptability thresholds	mostly open
evaluation instruments	closed
Ground truth	
ground truth for localization	closed
ground truth for mapping	closed
Benchmark Problems	
problems	closed
data representation and file formats	mostly open
evaluation methodologies for the solutions	closed
Benchmark Solutions	
solution algorithms	mostly closed
web-publishing policy for user-generated BSs	closed
Documentation and manuals	almost closed

3. Hardware and software setup

3.1 Robot platform

The choice of the robot platform to use for the indoor data-gathering sessions of RAWSEEDS was subject to strict constraints. These arose mainly from the consideration that the acquisition robot (i.e., the platform equipped with sensors, computers and associated equipment) would have to be capable of easy movement through narrow passages (such as doors or partially-obstructed corridors) and be able to manoeuvre safely in cramped environments, possibly in the presence of people. Two main courses of action were open: the use of a commercial platform or the choice of a robot built by the partners.
The available commercial platforms come basically in two sizes; the first is about the size of a very widespread research platform, the ActivMedia Pioneer 3DX (see Figure 1). The second is about the size of the larger Robuter by Robosoft (whose current version is called *Robulab 150*), also quite widespread in Europe (see Figure 2).

Figure 1. A Pioneer 3DX, from MobileRobots (USA). In this image, over the (black) upper platform of the robot are mounted a laser range scanner (blue) and a stereoscopic camera (white), which are not part of the robot.

Figure 2. A Robuter, from Robosoft (Fr).

Robots of an intermediate size are more difficult to find. The Pioneer 3DX-like platforms are not large enough to carry the conspicuous sensing suite required for the project; they are also not adequate under the point of view of the power supply that is required for running such set of devices for a reasonably long time. Conversely, platforms like the Robuter are too large for agile indoor motion; for instance entering a door means very reduced side margins. Moreover, as RAWSEEDS aims at producing real-world datasets, for indoor scenarios we considered incongruous the use of platforms that are not of realistic use in that type of environments.
Platforms that are appropriate in size for RAWSEEDS' indoor operations are less common on the market; an example is the circular evolution of the Robosoft Robuter. However, even if commercial robots of the required size were easily available, the choice of one of them would have brought in evidence a problem that (from past experiences) is common to all commercial platforms: commercial robots are “black boxes”, whose internal systems are not described in detail in the associated documentation. Moreover, it is nearly impossible to get usable working knowledge (beyond manuals) from the manufacturers, who are unwilling to divulge their know-how. The consequence of this is that “putting the hands in” a commercial robot is always difficult, because it necessarily involves a dose of reverse engineering. This would have created serious problems to RAWSEEDS: in order to integrate the disparate set of sensors required for data collection, knowledge of the internals of the platforms is necessary.

In the end, we decided for not using a commercial platform, even if some of them were already available at the partners' premises. The downside to that is the fact that the platform is not a standard one, introducing an element of non-replicability in the indoor data gathering sessions of RAWSEEDS. However this is a very minor problem, as the kind of robot platform used does not influence the datasets on which all of the RAWSEEDS toolkit will be constructed. (On the other hand, as will be explained in the following sections, great effort has been put in the choice of the *sensors* mounted on the platform. As the RAWSEEDS datasets are the output of these sensors, it is important for replicability that the sensors are, wherever possible, commercially available and widespread.)
Basing on shared knowledge of POLIMI and UNIMIB and the experience gained mainly in the RoboCup soccer (midsize) benchmarking competitions, we selected the robot called **Robocom**, built by UNIMIB for POLIMI; this assured complete knowledge about any aspect of the device. Below are some images of the platform (please note that the sensors shown in the following figures are not the ones mounted for the RAWSEEDS operations).

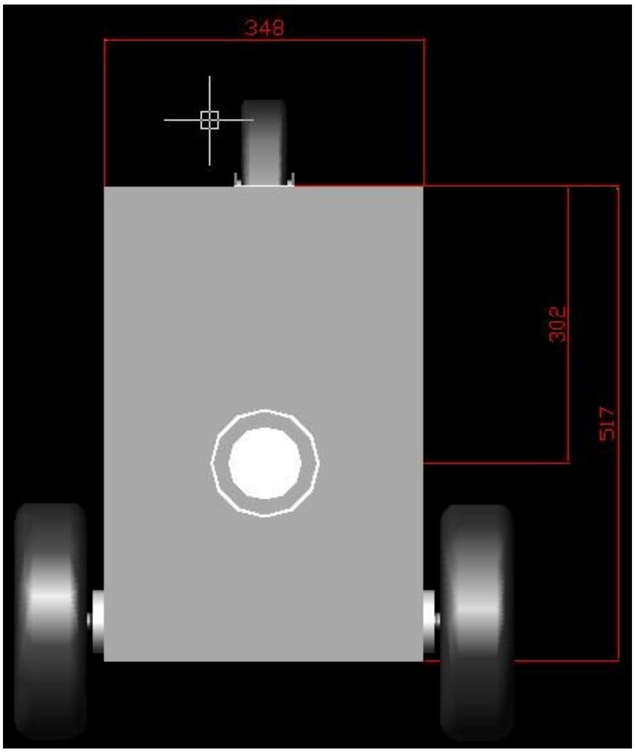


Figure 3. Robocom robot (rendering) with main dimensions in mm, view from the top.

Figure 4. Robocom robot. Mounted on the robot are the following sensors: omnidirectional catadioptric vision system, correlation-based stereo camera, normal camera, LRFs.

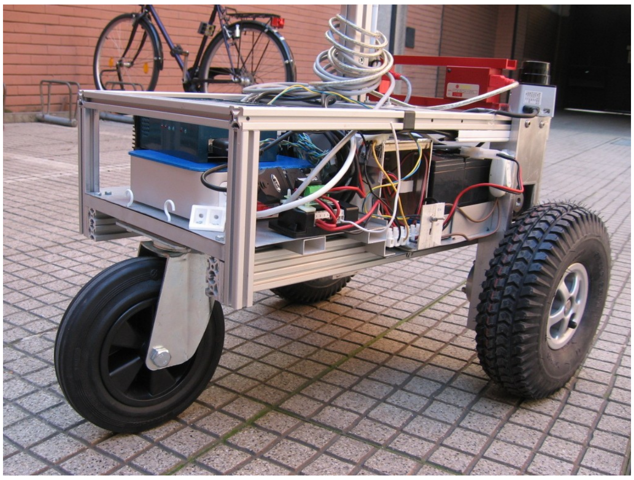


Figure 5. Robocom robot, view from the rear.

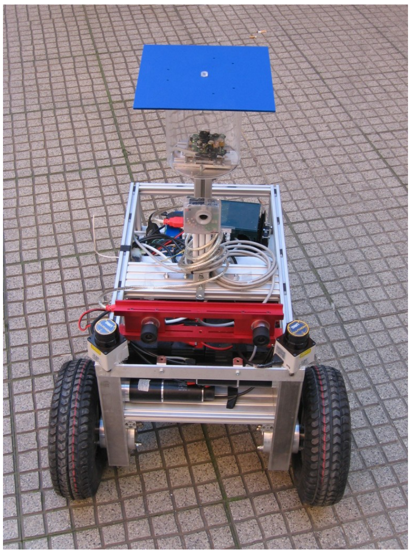


Figure 6. Robocom robot, view from the front.

Robocom is a differential drive platform, in line with the tenet of using realistic devices for indoor operation. The robot is governed by an Apple “Mac mini” computer. As we will explain later, additional computers are used to handle the datastreams produced by the sensor systems, leaving motor command and odometry tasks to the Mac mini. The motor controller is a two boards box (power and logic boards) which has been jointly developed by POLIMI and UNIMIB.
Robocom has been chosen, over alternative self-built platforms, for several reasons:

- relatively small footprint;
- high payload (>60kg);
- good maneuverability and smooth motion;
- easy mounting and dismounting of additional hardware, due to the use of modular aluminum profiles (made by Item) for the frame and the square and flat upper surface;
- remote control for safety stop (this is necessary to operate the robot remotely in the presence of people);

- complete knowledge about its internals;
- realistic, i.e., standard, kinematics (differential drive platform);
- readily available odometry: the wheel encoder data are sent to the Mac mini from the low-level controller and odometry is readily available in the Mac mini;
- limited capability for outdoor operation, in addition to indoor (this could be useful for “mixed” environments).

The software architecture of Robocom's Mac mini is based on the Linux operating system. The software for robot control has been custom-written by POLIMI and adapted to more efficient motor power consumption by UNIMIB. The printed circuit implementing the robot control has been developed by POLIMI and UNIMIB.

3.2 Sensor systems

3.2.1 Choice of the sensing suite

The **sensing suite**, i.e., the set of sensors mounted on the robot platform for indoor data acquisition, was determined according to the following requirements:

- comprehensiveness - all the main categories of (non-contact) sensors widely used in robotics had to be covered;
- *state-of-the-art* – the data produced by the sensors have to be of the highest quality available;
- low- and high-end – both cheap (i.e., likely to be found on consumer robots) and reasonably expensive (i.e., likely to be found on research or industrial robots) sensors had to be present. Very expensive devices (e.g., LIDAR) were disregarded.

When available, we always used commercial sensors. This assured that technical specifications, driver software, manuals and manufacturer support were available; but the main reason for this choice was the need to make RAWSEEDS' results as easily reproducible as possible. For the same reason, when (within a given sensor category) a specific make and model of sensor was regarded as “standard” by the robotic community, we chose to use that model.
Additional constraints on the choice of sensors were: the limited payload and size of the robot; the necessity to avoid interaction or interference between sensors (e.g., the field of view of the cameras has to be completely unobstructed); and finally the need to limit the overall mass and bulk of the sensor-equipped robot, which will have to move in narrow environments.

At the end of the selection phase, the RAWSEEDS sensing suite for indoor operations was composed of the following types of sensors:

1. robot odometry;
2. binocular and trinocular black-and-white (B/W) vision, as examples of feature- and correlation-based vision systems;

3. normal perspective, color and B/W cameras;
4. omnidirectional color vision with hyperbolic mirror, and possibly omnidirectional B/W vision with hyperbolic mirror;
5. short-range (<4m range, shorter at low reflectivity) cheap Laser Range Finders (LRF);
6. medium- and long- range (respectively <30m and <100m range, at 100% reflectivity) high performance LRFs;
7. sonar belt with multiple ultrasonic sensors;
8. Inertial Measurement Unit (IMU) providing 3-axis angular orientation, acceleration, rate-of-turn and Earth magnetic field data.

The binocular and b/w monocular systems are, in practice, realized with subsets of the trinocular system, to avoid adding unnecessary devices to an already very populated sensor suite.
Specific details on the sensors and their usage will be given in the following sections.

3.2.2 RAWSEEDS' *sensor frame*

The following is a list of the specific devices used to assemble the sensing suite (the numbering of each item is coherent with the list given in the previous section):

1. Conventional differential drive odometry, made available by the control board driving the motors of the Robocom robot.
2. Binocular vision system composed of a two-camera Videre Design STH-DCSG-VAR system (two FireWire, B/W, 640x480 pixel cameras mounted on a common mechanical frame that allows for an adjustable baseline). Trinocular vision system is realized combining the binocular STH-DCSG-VAR with an additional Videre Design DCSG camera (the same camera used by the STH-DCSG-VAR). Although CMOS, these cameras feature a global shutter, which is important for shooting moving scenes or from a moving observer (both things happen in our case). Web: http://www.videredesign.com/sthdcsgvar.htm, http://www.videredesign.com/Templates/dcsg.htm
3. Each of the three cameras of the trinocular system provides a B/W monocular data stream. Color monocular vision is covered by an Unibrain Fire-i 400 camera (FireWire, color, 640x480 pixel). Web: http://www.unibrain.com/Products/VisionImg/Fire_i_400_Industrial.htm
4. Omnidirectional color vision is obtained by using an Eizoh SOIOS-55CAM catadioptric camera with 55mm hyperbolic mirror (FireWire, color, 640x480 pixel). Web: http://www.eizoh.co.jp,
5. 2 Hokuyo URG-04LX LRFs, mounted on the front and the back of the robot. The LRFs will be tilted down, pointing at the floor about 2 to 3 meters away from the robot (precise orientation will be chosen later). One of the Hokuyos could be mounted on a tilting base, in order to perform 3D scans through a succession of planar scans taken at different inclinations, but this depends on the completion

time of the base (which is not a RAWSEEDS project).
Web: http://www.hokuyo-aut.jp/products/urg/urg.htm

6. Sick LMS291 and LMS200 LRFs, mounted on the front and the back of the robot. Web:
http://mysick.com/partnerPortal/eCat.aspx?c=1&go=FinderSearch&Cat=Row&At=Fa&Cult=English&Category=Produktfinder&FamilyID=267&Selections=8641,0,0,8775,0
http://mysick.com/partnerPortal/eCat.aspx?go=FinderSearch&Cat=Row&At=Fa&Cult=English&Category=Produktfinder&FamilyID=267&Selections=8644,0,0,8775,0
7. Sonar belt composed of 12 to 16 Polaroid 600-series sensors (positioned all around the robot) and associated control electronics built by POLIMI. The number of Polaroid sensors actually used depends on availability at the moment of the data acquisition.
8. Xsense MTi IMU (USB, 1,7g full scale acceleration, 150deg/s full scale rate of turn). Web:
http://www.xsens.com/index.php?mainmenu=products&submenu=machine_motion&subsubmenu=MTi
9. Color frontal camera, this device will likely be a FireWire color camera from Unibrain, the rugged version of their single-CCD product.
10. Upward looking color camera, as the camera above.

The above sensor suite is quite extensive and considerable effort has gone into its effective integration into a coherent, time-synchronized, multiple-computer sensor system. For this reason we intend to re-use most of it for the outdoor data collection activities of RAWSEEDS.
Such a comprehensive sensor suite cannot be mounted aboard a robot in a casual fashion, if conflicts and unwanted interactions or movements are to be avoided. Therefore we have designed and built a specific mechanical frame devoted to the data collection, on which all of the sensors have been mounted. This supporting frame is very rigid and is entirely composed of aluminum profiles, built by Item (http://www.item.info), thus making the mounting and un-mounting of sensors relatively easy. In the rest of this document this structure, complete with sensors and all associated devices, will be called **sensor frame**.
The sensor frame has been made a completely autonomous entity, capable of operation on different robots or even when not aboard a robot: in fact it includes its own multi-voltage battery-powered power supply and its own computer modules (up to 3 complete PCs). An additional vantage of this modular design is the fact that in this way the (not negligible) electrical power drain of the complete system "robot + sensor frame" is split between two battery packs, thus affording a longer operation time without recharge. A detailed analysis of the issues related to power consumption and battery life for RAWSEEDS' indoor data-gathering robot will be given in the following sections.
Figure 7 shows a rendering of the sensor frame.

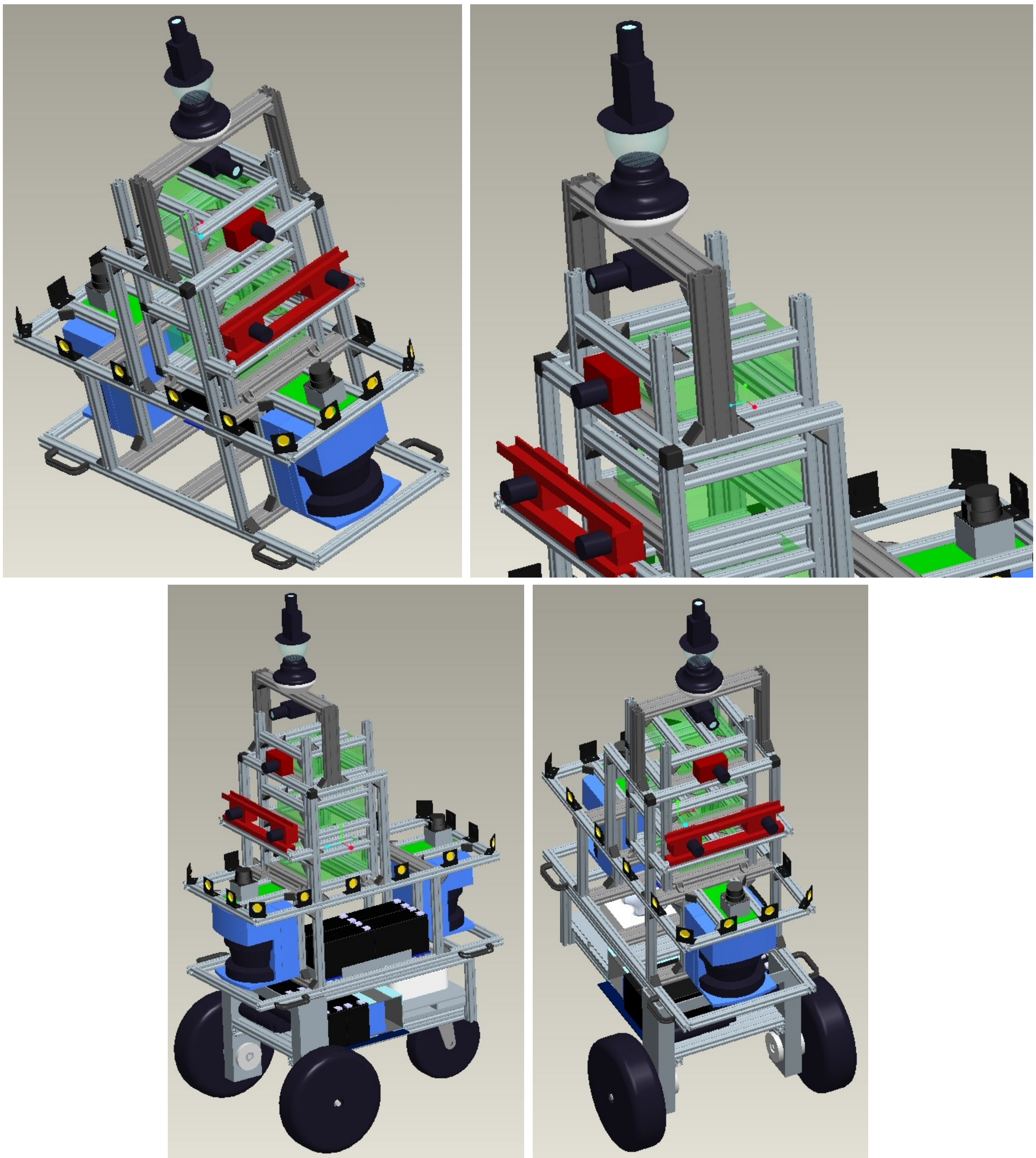


Figure 7. Rendering images of the sensor frame alone (top) an mounted on Robocom (bottom). Some of the sensors are shown: in particular Sicks (blue), Hokuyos (grey/black), sonar belt (yellow/black), front camera (black), omnidirectional camera (the topmost camera). The inner frame with semitransparent green interior is a stack of 3 PCBricks (please see the following sections for details). The black boxes are the batteries. The upward looking camera is not depicted and will be mounted on the top of the omnidirectional one.

3.3 Setup of the data-acquisition robot

In this document the term **data-acquisition robot** is used to define the union of the Robocom robotic platform with the sensor frame defined in previous sections. It is therefore the complete system that will be used to acquire the datasets on which the RAWSEEDS toolkit is based.
The following sections give a detailed description of how the data-acquisition robot has been designed and assembled, and of its expected performance.

3.3.1 Processing power

RAWSEEDS operations do not require any kind of elaboration or processing of the data coming from the sensors: therefore there is no need for powerful computers aboard the robot. However, such an extended sensor set requires considerable data-transport and storage resources just for capturing and storing the amount of data produced, most of which comes from the cameras. Many interfaces have to be accurately managed to exclude any data loss (which would cause all the dataset to be rejected during the validation phase); in addition to that, accurate synchronization between the single elements (e.g., a single frame or scan) of the data streams produced by the sensors is required by RAWSEEDS' specifications. As the sensor frame (including its processing facilities) is a battery-powered system, all these requirements had to be met while keeping power consumption as low as possible.
The Robocom robot includes a powerful Mac mini computer, equipped with a dual-core Intel CoreDuo CPU with a clock frequency of 1.83GHz. However, the design approach of RAWSEEDS' sensor frame requires that it is completely self-sufficient: therefore we chose not to use the Mac mini for sensor data acquisition, but only for robot navigation and odometry (this is a function that is necessarily platform-dependent, so it was pointless performing it in the sensor frame).

Data acquisition, synchronization and storage in the sensor frame are performed by multiple small-footprint PCs, called **PCBrick**s, expressly assembled by POLIMI for RAWSEEDS. The sensor frame can contain up to three PCBricks, but two computers should suffice (if preliminary acquisition tests should prove it necessary, the third can be easily added). The PCBricks aboard the robot are interconnected by a TCP/IP Ethernet network.

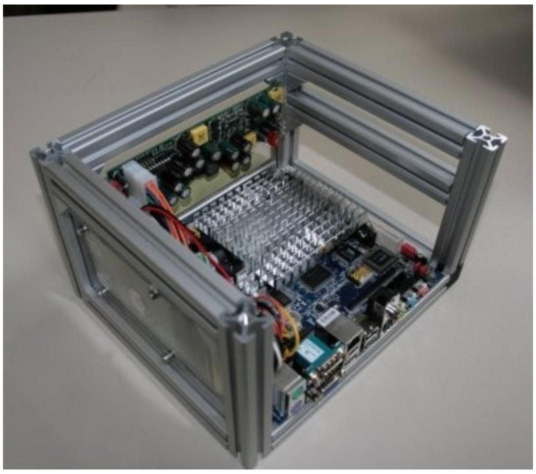


Figure 8. A PCBrick. External dimensions are 225mm (l) x 195mm (d) x 135mm. It can be used in horizontal or vertical position.

Each PCBrick is a complete x86-compatible PC, designed for maximum modularity in terms of mechanical construction and data interface provision. It has been designed explicitly for robotic applications, and for RAWSEEDS in particular. The main design goal has been to minimize power consumption while maintaining acceptable performance of the single PCBrick: if more computing power is required, multiple PCBricks can be used. In this way the use of powerful and power-hungry PCs can be confined to the applications where they are really necessary.

The PCBrick design is based on the use of motherboards with mini-ITX form factor (17cm x 17cm). For RAWSEEDS we chose the VIA EN15000 motherboard, fitted with a single-core VIA C7 processor with a clock frequency of 1.5GHz and with an extensive set of interfaces. An aluminum frame is used, made with the same Item profiles used in the construction of Robocom and of the sensor frame, for easy mechanical connections. Included in the frame of the PCBrick are an 80GB 2,5" fast (7200RPM) hard disk, 1GB of DDR2 RAM and a DC/DC power supply accepting 6V÷24V input. The frame includes the space necessary for a PCI card and fixtures for the mounting of additional hardware.
To give an impression of the performance/power ratio of the PCBrick architecture, let us consider the single Apple Mac mini computer (dual core processor, 1.83Ghz), which controls the Robocom robot, which has a maximum power consumption of 100W (already low, as this computer is based on laptop PC components). The same 100W can power *five* PCBricks: these not only have a superior computing power and a much more extended interface set compared to the Mac mini, but allow for a much increased flexibility in hardware and software system building.

The software architecture of RAWSEEDS' PCBricks is based on the Linux operating system; custom software, developed by POLIMI, is used for acquisition and synchronization of sensor data.

3.3.2 Networking and synchronization

The computers (Mac mini and PCBricks) on board the robot and the sensor frame will be connected by means of a 100Mbps wired LAN. Network traffic is expected to be low, as storage of sensor data will be executed by the computers to which the sensors are connected. The network is used only for the exchange of commands and synchronization data, thus minimizing latencies. Network architecture makes use of a D-Link DI-624 wireless router as its center: this device includes a 4-port switch and will also be used to establish wireless IEEE 802.11g connections with external clients in order to access the PCs mounted on the robot (e.g., via SSH protocol).

The main reason for using a LAN aboard the data-gathering robot is the necessity to maintain strict synchronization between the various data streams produced by different sensors, acquired by different computers. To be useful, a multisensorial dataset need in fact to guarantee that any subset of data can be precisely positioned in time with reference to any other. For instance, to use vision data for robot mapping it is absolutely necessary that each frame is linked to the odometry data pertaining to its acquisition; and the same is true for the data from the other sensors. In RAWSEEDS, this is obtained by associating each single piece of data in the datasets (e.g., a single frame acquired by a camera) to a *timestamp* indicating the time instant at which it has been generated; and this in turn means that a unified, robot-wide time-generation system is an absolute necessity. In RAWSEEDS' data-gathering robot the sensors that require the highest timing precision are the vision sensors: therefore the level of precision afforded by the time-generation system has to be significantly lower than the frame rate of the cameras. As the frame rate that we plan to use is not greater than 15Hz, corresponding to a period of 67ms, the maximum time error in the generation of timestamps ideally need to be not larger than a few milliseconds. It must be noted that RAWSEEDS' sensor frame actually includes a sensor which produces data at a rate greater than 15Hz: namely, the MTi Inertial Measurement Unit. However that data stream, its dynamics being strictly linked to the mechanical dynamics of the robot's motion, is less demanding in terms of time precision than the video streams.

Synchronization between the nodes of the Robocom LAN will take place by means of Network Time Protocol (NTP) server and clients. Network Time Protocol (http://www.ntp.org/) is a well-established software system that, for machines belonging to the same (wired) LAN, should assure synchronization within a few milliseconds. For the time being (i.e., before any acquisition test has been done) it is the chosen solution. If necessary, the Chrony software (http://chrony.sunsite.dk/) will be used to overcome the limitation that NTP could impose on a computer system which does not have a continuously active connection to an NTP Time Server, i.e., one of the physical machines distributing synchronization data.
We expect this level of time accuracy to be sufficient for our applications, although this will depend on the actual frame rates chosen for camera acquisition. However, in the context of autonomous robotic systems high frame rates are neither realistic nor easily attainable for technical reasons. In addition to that, the use of high frame rate vision data would pose significant problems for RAWSEEDS. In fact RAWSEEDS sensor data streams need not only to be acquired and stored, but most importantly *made available to users through the Internet*. The use of high frame rates would generate datasets so enormous that their transmission over the Internet would become unfeasible.
Currently the work on the synchronization systems of RAWSEEDS' data-gathering robot is still ongoing, so no definitive answer about the correctness of our approach to the problem can be given yet.

3.3.3 Robot remote control

Remote guidance of the robot will be performed through a Logitech Cordless RumblePad 2 wireless joy pad; the receiver will be connected to the Mac mini via USB.
A radio remote control switch is fitted on the robot for safety reasons. It disables robot motion, and its operation is utterly independent from the PCs and the wireless network.

3.3.4 Interconnections

The overall amount of data generated by the sensors of RAWSEEDS' sensor frame is rather high. This is mainly due to the presence of multiple cameras: the bit rate output by the other sensor systems is significantly lower. Therefore, two kind of problems could occur:

1. input saturation: the overall input data stream for a single data interface (e.g., FireWire) or for a single PCBrick is excessive;
2. storage saturation: as all sensor data need to be stored, the overall data volume could exceed the capabilities of the onboard mass-storage devices.

For example, the data rate produced by the Videre trinocular camera system at 15 frames per second amounts to 108Mbit/s (without the overhead due to the FireWire protocol): each one of the three cameras generates 15 frames/s, each composed of

640x480 pixels with a pixel depth of 8 bit/pixel.
These problems have been considered through system design. In particular:

1. the PCBricks have been fitted with fast hard disk drives;
2. multiple PCBricks have been used, thus splitting the data stream between them;
3. connections between sensors and PCBricks have been chosen to balance the data rates between the PCBricks and to avoid exceeding the usable data rate of each data interface.

Calculation of the data volume to be acquired and stored leads to the conclusion that two PCBricks are sufficient; if, for any reason, this should be proved false during the acquisition tests, it will be extremely easy to add a third PCBrick to the sensor frame.
It is important to note that the Mac mini is only used for tasks related to robot movement: the only sensor connected to it is the odometry system, which is an integral part of the robot. This design choice maximizes the independence of the sensor frame from the robotic platform, allowing the re-use of the frame on different platforms (or even in stand-alone mode). Figure 9 shows the interconnections (data and power) between the elements of Robocom and of the sensor frame.

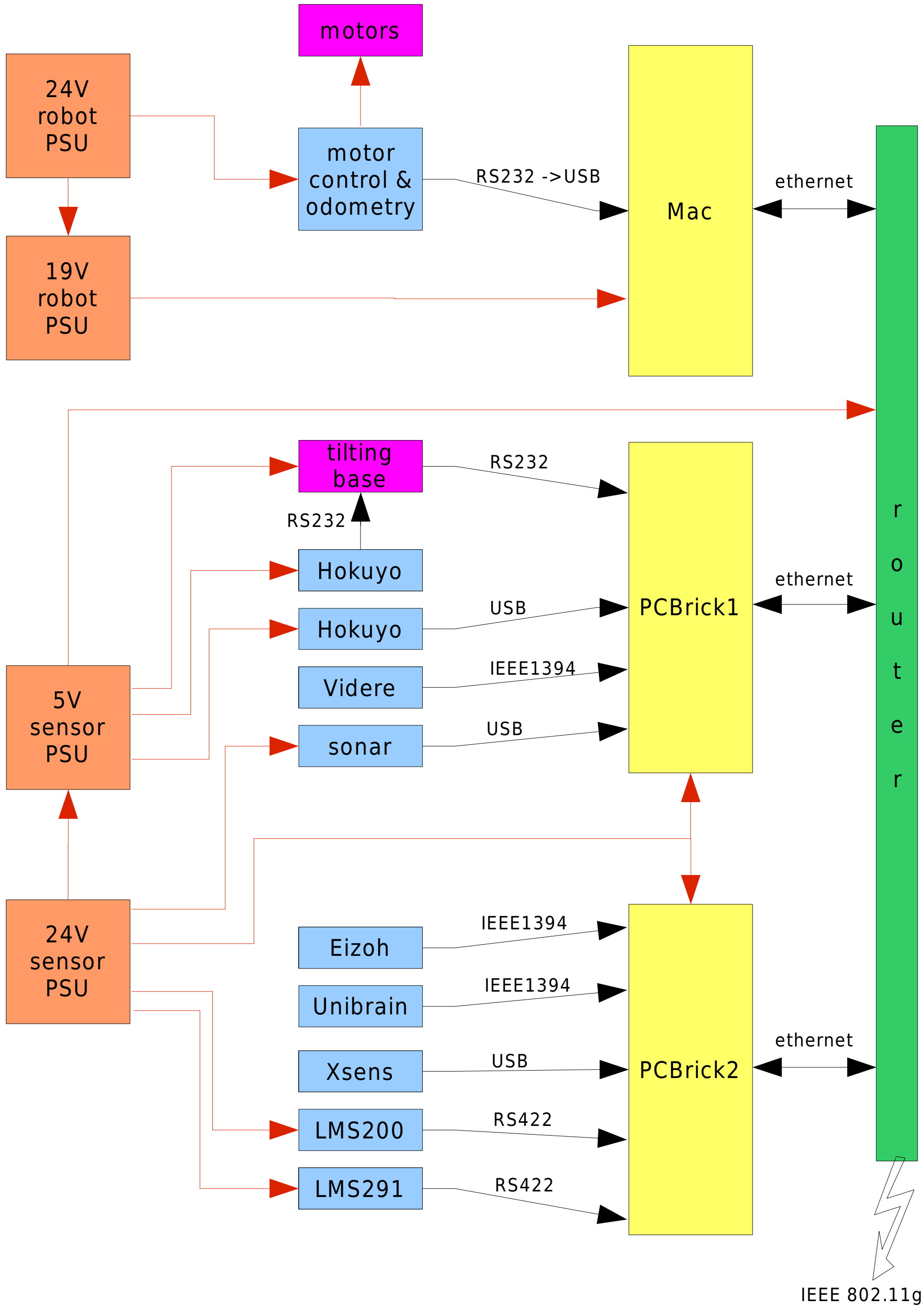


Figure 9. Diagram of interconnections for RAWSEEDS' indoor data-gathering setup.

3.3.5 Power usage and operating life expectations

Power consumption is a crucial matter in robot design, as it is directly related to the operating life of the robots (i.e., the working time after which the batteries need to be recharged). In typical situations, and with good robot design, current battery technology allows a few hours of operation: therefore power consumption of each subsystem of the robot must be kept as low as possible.
The following table summarizes the electrical power used by each subsystem of the

data-gathering robot that RAWSEEDS will use for indoor operations.

Device	**Power Supply Unit (PSU) voltage**	**Power**	**The device is powered by:** (R = Robocom, SF = sensor frame)
24V => 19V DC/DC converter module on Robocom (estimated conversion *efficiency 75%*)	24V	33W max	R batteries
Mac mini	19V	100W max	R batteries (through 24V=>19V DC/DC converter)
PWM control board for Robocom's motors and odometry (estimated power efficiency 90%)	24V	16W	R batteries
Robocom motors (2)	24V (PWM)	2 x 70W max	R batteries (through motor control board)
remote control receiver	24V	negligible	R batteries
PCBrick1	6 - 24V	20.5W max 15.5W idle	SF batteries
24V => 5V DC/DC converter module on sensor frame	24V	4W max	SF batteries
PCBrick2	6 - 24V	20.5W max 15.5W idle	SF batteries
Videre Design STH-DCSG-VAR + DCSG	FireWire specs	3 x 1W	SF batteries (through PCBrick1 via FireWire)
Eizoh SOIOS-55CAM	FireWire specs	1W (estim.)	SF batteries (through PCBrick2 via FireWire)
Unibrain Fire-i 400	FireWire specs	0.9W	SF batteries (through PCBrick2 via FireWire)
control board for sonar belt	24V	negligible	SF batteries
Sick LMS291	24V	20W	SF batteries
Sick LMS200	24V	20W	SF batteries
2 x Hokuyo URG-04LX	5V	2 x 2.5W	SF batteries (through 24V => 5V DC/DC

			converter)
tilting base for Hokuyo URG-04LX	5V	2W (estim.)	SF batteries (through 24V => 5V DC/DC converter)
Xsense MTi	5V	0,4W	SF batteries (through PCBrick2 via USB)
D-Link DI-624	5V	12W	SF batteries (through 24V => 5V DC/DC converter)

total power consumption on Robocom batteries (between brackets the corresponding DC current)	289W (11.4A@24V)	this is a peak value, with motors and Mac mini at full power; in typical operating conditions used power is much lower
total power consumption on sensor frame batteries (between brackets the corresponding DC current)	59W (2.5A@24V) 109W (4.5A@24V)	with PCBricks idle, Sick sensors off and everything else fully active with every subsystem fully active

Both Robocom and the sensor frame are fitted with a 24V DC battery power supply; each power supply includes two 24V battery packs connected in parallel; each pack is composed of two 12V, sealed lead-acid batteries connected in series; each battery has a capacity of 9.2Ah for Robocom and 7.2Ah for the sensor frame. Thus each battery power supply is equivalent to a single 24V battery; a 18.4AH one for the platform and a 14.4Ah one for the sensor frame. In practice, if the batteries are not to be damaged by excessive discharge, only about 60% of their charge capacity can be drained. Therefore each power supply has an usable charge capacity of about 11Ah and 8,6Ah respectively.
Under full load (a not realistic situation where motors and computers constantly draw full power) operating life time would then be about an hour for Robocom and 1 hour and a quarter for the robot frame. In a more realistic scenario, where the motors operate at a much reduced power, computing power is not fully exploited and occasional pauses are made (with robot idle and Sick sensors powered off), we expect to be able to execute 2-hours data acquisition sessions without battery discharge problems.

3.3.6 Calibration of sensors

We have a few issues here:

1. calibration of all sensors w.r.t. the same robot reference system
2. calibration of cameras
3. calibration of multi-camera sensors

Joint calibration of all sensors w.r.t. a single reference system might be an automatized task, if the ongoing work in this field will be successful. We will anyway make available manual measurements, w.r.t. the kinematic (odometry) reference system.

Calibration of cameras will be performed with well-known tools, widely mentioned in the literature and available in the Internet.

The calibration of the trinocular and binocular stereo head will be performed differently, as the binocular head is a commercial product, but it is not already calibrated, it comes with its own calibration procedure. It turned out that this procedure does exploit the relative position of the imagers, so that it does not work for pairs involving the 3^{rd} camera. The well-known tools, widely mentioned in the literature and available in the Internet, at the moment does not show a consistent performance. On the other hand, our old-fashioned tool, which has been in use since many years, is not appropriate for the cameras we are using, as their radial distortion is not negligible and the tool estimates just a linear projection. Like for the feature detection, camera calibration might be controversial, therefore, although we are still working on this issue (it is something that is part of our research), we are planning to make available images of checkerboards, as used by most current camera calibration tools, altogether with the results obtained with these publicly available tools.

4. Indoor scenarios

The following sections describe the scenarios for the (soon to be started) indoor data-gathering activity of RAWSEEDS. (Please remember that a scenario is defined as "*the complete set of conditions defining a single data-gathering session*".)
The indoor scenarios have been chosen to include a comprehensive set of the typical environments in which a mobile robot designed for indoor operation will likely need to operate. Examples of such environments include rooms, offices, halls, corridors. Typical features of them are flat and non-flat walls, doors and passages, windows and other surfaces which are transparent (to visible light), horizontal floors and ceilings, ramps, stairs, elevators. Objects which can be usually found in these environments include people (still or moving), tables, chairs, benches, armchairs and sofas, cabinets and other containers, bookcases, wastebaskets and bins. Lighting can be mixed (artificial light plus sunlight) or wholly artificial, but artificial lighting is always present.

Indoor environments are by far the most represented in the datasets already available to the robotics community (or used by actors belonging to this community, but not shared with others). This is due to three different reasons: first, neglecting some very specific tasks (such as lawn mowing) indoor environments are in general more likely to offer useful tasks for autonomous robotics both for commercial applications and for research; second, commercial robot platforms are generally built for indoor applications: therefore whoever uses them as a development tool is forced to concentrate on indoor operation; third, because the setup of a data-gathering experiment in the outdoors is generally much more complex and costly than doing the same indoor.
We expect that RAWSEEDS' indoor datasets will be regarded as an extremely useful tool by the robotics community, thanks to its composition of many high-quality, multi-sensor, real sensor, time-synchronized, and also rigorously validated collection of sensor streams. None of the currently existing datasets jointly possesses all the above qualities, and many datasets do not possess any. For this reason we expect RAWSEEDS' indoor datasets (and the associated Benchmark Problems) to become a standard yardstick to assess the performance of localization, mapping and SLAM algorithms, and to give a perceivable boost to research and applications in the robotics field. RAWSEEDS datasets will be especially valuable for companies currently evaluating the opportunity to enter the market of robotic products: these companies do not usually have the hardware and the know-how needed to acquire datasets, and therefore will not dare to invest money on the development of software for robotics which can only be tested by applying it to quality real-world datasets. The development of original algorithms and software, however, is the key to the development of commercially viable products. By eliminating the significant stumbling block created by the absence of usable datasets, we think that a number of small but high-technology companies could choose to enter the now dawning market of robotic solutions.

4.1 Location

The chosen indoor scenarios share the same location: the inside of a pair of the buildings belonging to the campus of UNIMIB (University of Milano – Bicocca; Bicocca is an area of the city). This choice allows for an easy setup and conduction of the indoor acquisition campaigns, as the location and the research groups of POLIMI and UNIMIB are located in the same city (Milano, Italy). The **Bicocca** location (as it will be called from now on), in addition to being a typical indoor environment possessing all of the characteristics listed above, can be put (under suitable circumstances, such as on weekends) under the control of the UNIMIB partner: this is necessary for some of the scenarios that RAWSEEDS will exploit, such as the ones where no people are present. Moreover, this location includes conventional office-like environments along with more unusual features, such as a fully windowed bridge between the two buildings or a library. RAWSEEDS datasets will include both.

Details on the data-gathering scenarios that have been defined in the Bicocca location will be given in the following section.

It is interesting to point out that we considered the possibility to use (also) domestic environments as indoor locations. However, this option has been discarded because we think that, in the contest of mapping, localization and SLAM, all the challenges to robotic systems presented by domestic scenarios are already covered by the chosen indoor scenarios.

4.2 Scenarios

Multiple indoor scenarios have been defined, each of which will lead to a separate dataset. All are set in the Bicocca location. In the following of this section we will describe these scenarios.

1. **Static with no light changes**
   In this scenario every feature of the explored environments perceivable by the robot is constant over time. If the robot returns to a previously visited place, everything will appear unchanged to its sensors (except for the effect of noise). This requires the complete absence of people, tight control over moving or easily moved objects (such as doors) and, most difficult of all, static lighting. The location has many windows that carry in natural light from the outside, which changes naturally with the passing of time in the day, passing of clouds, etc. Therefore the data collection for this scenario will be effected by night, with artificial lighting only.
2. **Static with light changes**
   In this scenario every feature of the explored environments perceivable by the robot is constant over time, *with the exception of lighting*. We will allow natural light through the windows to affect the dataset, by performing the data-gathering during the day. This will inevitably bring to variable lighting conditions over the course of the session, due to variation of the lighting over time and over space. The latter are due to the fact that windows are not uniformly distributed in the environment, and even similar spaces can have different lighting conditions because they depend on the opening state of the doors leading to windowed rooms, such as offices. The data of this scenario will be collected during the day; to avoid breaking the "static" hypothesis, data-gathering will probably occur during weekends.

3. **Dynamic with no light changes**
   In this scenario we will not allow natural light through the windows to affect the dataset, but we will collect the data during normal office hours, so to have moving people and objects captured by the robot's sensors. We plan to perform the data collection for this dataset in the winter, when late afternoon is not distinguishable from night.
4. **Dynamic with light changes**
   In this scenario we will allow natural light affect the dataset and will perform the collection during office hours, so to have moving people and objects captured by the robot's sensors.

4.3 Data acquisition methods

Two methods for the acquisition of sensor data are commonly used in mobile robotics: *stop-and-go acquisition* and *continuous acquisition*. In stop-and-go acquisition the activities of motion and sensor acquisition are separated in time: the robot moves, then stops, and then (with robot still) sensor data is acquired for a predefined time; after that time, the robot moves again, and so on. The overall trajectory of the mobile robot is therefore split into a succession of steps, and sensory exploration of the environment is done only between the steps. On the contrary, in continuous acquisition the sensors are always operated and robot motion is effected without pauses: therefore the sensory data streams are affected by the fact that the sensors are moving during acquisition. In particular, each acquisition event (e.g., the capture of a single frame by a camera) performed by a sensor might lead to a different output compared to the one that would have been obtained if the robot was still during the event; this difference increases with the duration of acquisition events and with robot speed.
It should also be mentioned that the presence of an Inertial Measurement Unit onboard the robot is an additional reason to prefer continuous acquisition over step-by-step operation. Minimizing the peak accelerations that the robot is subject to (and thus eliminating possible oscillations or abrupt changes of speed, which could be generated when the robot is starting or stopping to move) seems to be, in fact, a good way to minimize errors and drift in the data output from the IMU.
While the use of stop-and-go acquisition produces sensor data which are easier to interpret and elaborate, as they are essentially "still images" of the environment that the robot is moving through, we consider the continuous approach as much more realistic for present and future robotic applications. We believe that future applications will likely see continuously moving robots (although the use of a stop-and-go initial exploratory phase, just for the SLAM problem, is not to be excluded).
For the above reasons, continuous acquisition will be used for the RAWSEEDS indoor scenarios, unless preliminary acquisition tests show significant problems.

For many sensors, acquisition time intervals are long enough to make robot motion not negligible. This is true, for example, for cameras and Laser Range Scanners; the effect on the output of a camera is the well-known phenomenon of *motion blurring*, which can drastically reduce the possibility to extract useful environmental features from a video stream. The frame rate for the various cameras mounted on the RAWSEEDS sensor frame will be set during preliminary acquisition tests, considering the need to ensure

both good tracking of robot movements (whose speed will be subject to tuning in itself), and the need to remain within the given limits of computing power and total data volume. The latter is an essential requisite, as we want users to be able to download the RAWSEEDS datasets in a reasonable time. POLIMI has performed evaluations on this matter: preliminary results suggest that 100-200GBytes of uncompressed data overall are a good compromise.
The camera frame rate will be kept as low as possible, as dictated by robot speed (and above all its rotation speed, as even low rotation speeds can generate heavily blurred frames and difficulties into tracking features). If more than 15fps should prove necessary in field tests, the robot will be slowed down instead of increasing the frame rate. Exposure time for the cameras will be tuned too, as it could be critical: the presence of robot motion favors short exposure times, whilst the presence of dimly-lit zones in the explored location will ask for long exposure times.
The other sensors will be acquired at the maximum admissible speed.
A concluding remark concerns the real speed of the robot and the usage of the datasets: as the robot will move quite slowly during the collection of the datasets, a potential usage could encompass the sub-sampling of the dataset to simulate an higher speed.

4.4 Data-gathering sessions

4.4.1 Location Bicocca

Important note.
Obtaining from academic authorities the authorization to publicly distribute information about this location has not been easy. (Actually, the largest part of the delay in delivering this document is related to this.) The policy that has been agreed with the central offices of UNIMIB, for distributing photos, drawings, datasets, etc. of the Bicocca location to partners and to the general public is as follows.
UNIMIB location data will always be made available without any reference to the specific building and floor. Files, datasets and documents (including filenames) will not include any such reference. UNIMIB apologizes for the inconveniences that could result from such policy.

The physical places included in this location are the following (m, n, A and B are fictitious names):

- j^{th} floor of UNIMIB building A: includes corridors, doors, windows, halls, normal and automatic stairs, elevators; artificial and natural light; bridge going to building B, windowed from ceiling to floor;
- j^{th} floor of UNIMIB building B: includes halls, library (wooden walls, much more windows than in building A, large amount of tables and seats, etc.).
- $(j-1)^{th}$ floor of UNIMIB building B: includes corridors, doors, windows, halls, stairs, elevators; artificial and natural light;

The following part of this section gives a general impression of the environments included in the Bicocca location. It is not intended as a precise description of the actual trajectory of the robot during the data-gathering session, as the details of that will be

defined after the first acquisition tests; it is meant as a *reportage* on the chosen location.
Figures 10 to 13 are technical drawings showing the floor plans of the location. They include numbered dots that correspond to the places where photos have been taken. Figures 14 to 33 are these photos. In the drawings there is no indication about the direction the camera was pointing to during shooting. The red lines represent a possible path for the data collection. Photos were taken in the afternoon of a very sunny day in March 2007.

The reportage takes the start in building A, floor j^{th}, in a big hallway with uneven lighting and mixed features such as escalators and stairs (Photos 1, 2, 3, 4).
We then have the entrance into a corridor (Photos 5, 6).
We then turn right over a slight ramp (about 10% slope) to reach a glass-walled passageway between buildings A and B (Photos 7, 8).
Then we enter the passageway (Photos 9, 10), where we have the view of another similar passageway from the windows (Photo 11).
The passageway ends into a small hall of building B with elevator doors, separated from a library by glass doors (Photos 12, 13, 14, 15).
We then go down one floor to floor $(j\text{-}1)^{th}$, by taking the elevator (no pictures of that); then we have a passage through a wide corridor with features such as ramps (Photo 16), hallways (Photo 17), stairs (Photo 18). We then go to a central hallway of building B, still one floor down with reference to the library floor (Photo 19).
With one brief trip on the escalator (we can do that with the robot too, if needed) (Photo 20), we are back to the library floor (i.e., floor j^{th}) (Photo 21) then we move to the library entrance (Photos 22, 23).
The library features a very different environment: walls are usually covered with wood or books, and there are many tables and seats (Photos 24, 25, 26).
We then reach a hallway with stairs (Photos 27, 28), which ends in a library exit (Photo 36), which we do not use.
We then turn left to explore more of the library: more halls (Photos 28, 29, 30, 31) a corridor (32), the entrance to a small reading room (33), the room itself (34), the end of the corridor (35) that leads to a small, glass-walled room (36). This room is the same featured in Photos 12 to 15, now seen from the other side of the glass door. Here the loop closes and we can now exit through the glass doors and go back through the glass-walled passageway to building A, to return to the starting point.
Photos 37, 38 and 39 show some views of the open-air space of building B that was glimpsed through a window in floor $(j\text{-}1)^{th}$ (in Photo 16), which could be an interesting addition to the path and is easily accessible through ramps.

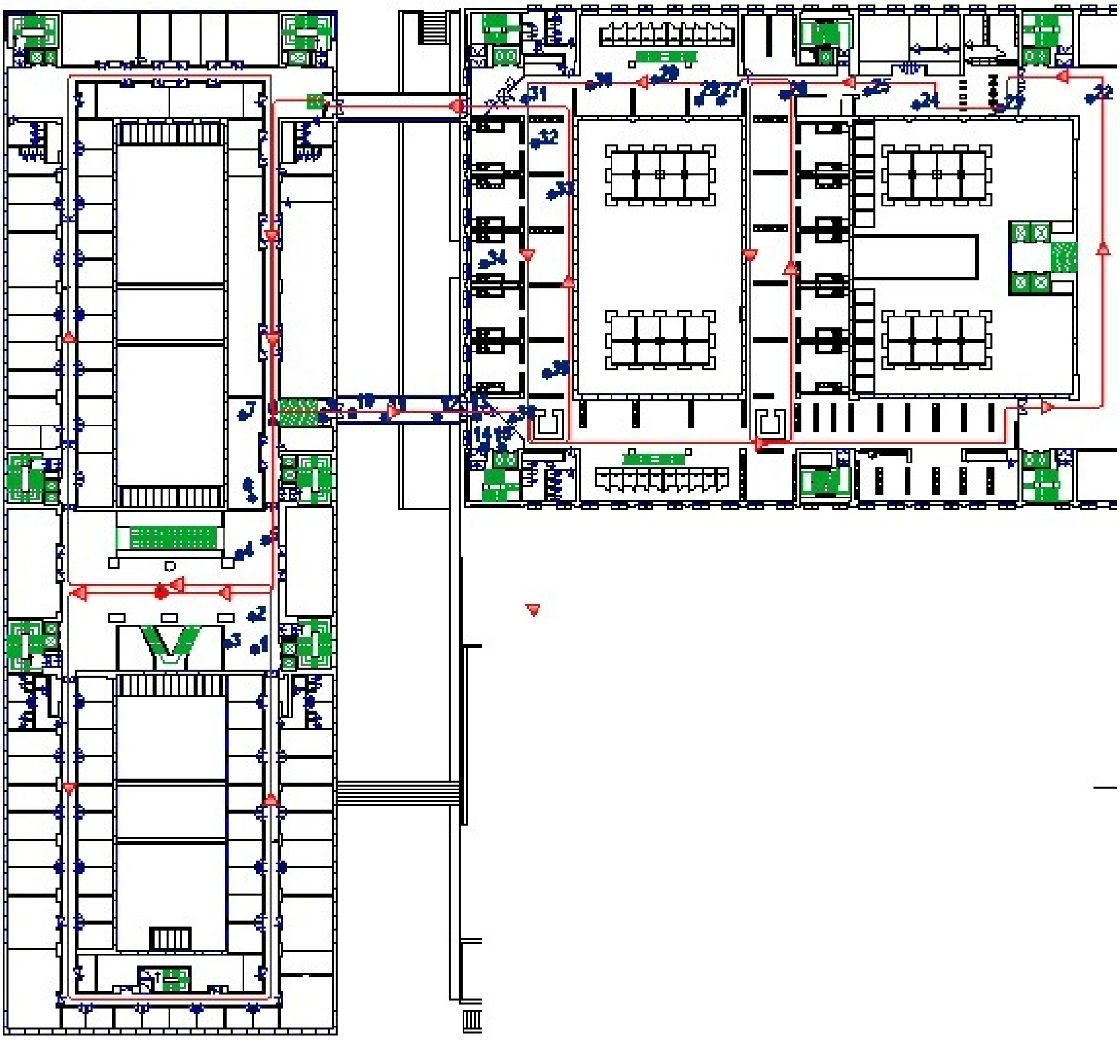


Figure 10. Bicocca building A and B, floor j^th^.

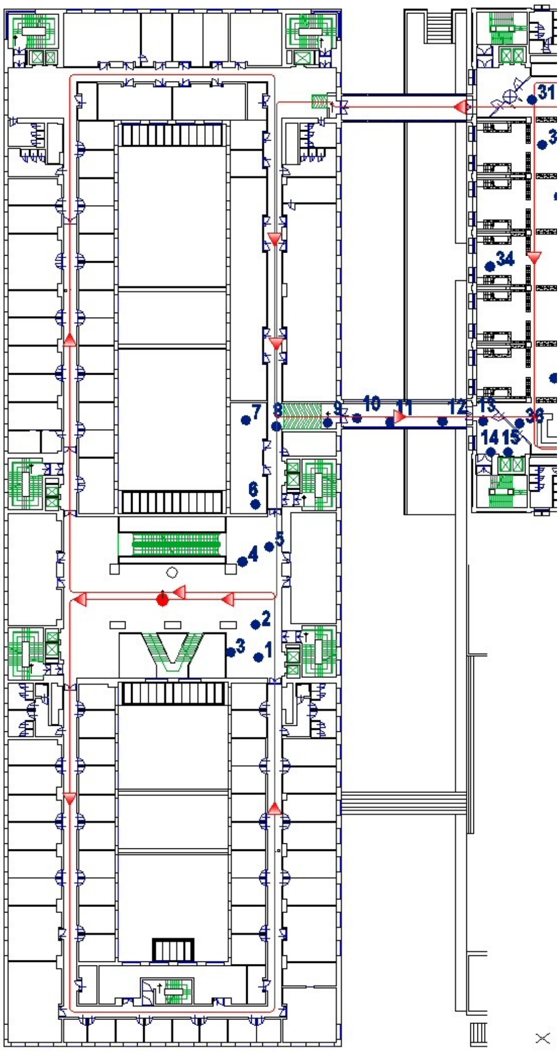


Figure 11. Bicocca building A, enlarged view of floor jth.

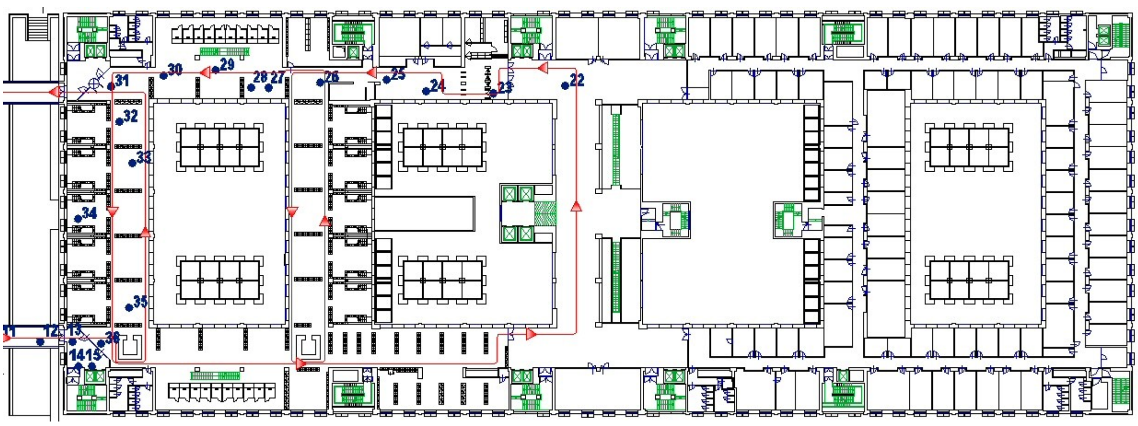


Figure 13. Bicocca building B, enlarged view of floor jth.

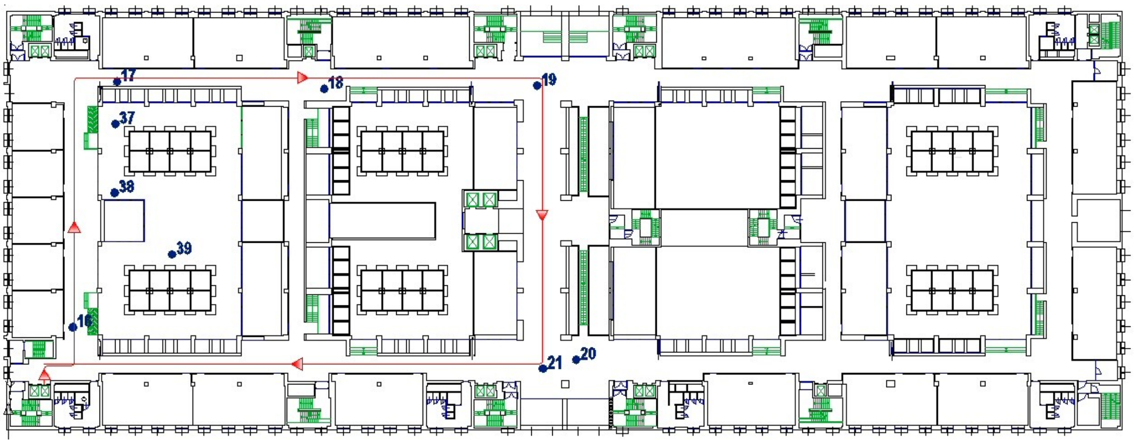


Figure 12. Bicocca building B, enlarged view of floor (j-1)th.

Figure 14. Photos 1 and 2.

Figure 15. Photos 3 and 4.

Figure 16. Photos 5 and 6.

Figure 17. Photos 7 and 8.

Figure 18. Photos 9 and 10.

Figure 19. Photos 11 and 12.

Figure 20. Photos 13 and 14.

Figure 21. Photos 15 and 16.

Figure 22. Photos 17 and 18.

Figure 23. Photos 19 and 20.

Figure 24. Photos 21 and 22.

Figure 25. Photos 23 and 24.

Figure 26. Photos 25 and 26.

Figure 27. Photos 27 and 28.

Figure 28. Photo 29.

Figure 29. Photos 30 and 31.

Figure 30. Photos 32 and 33.

Figure 31. Photos 34 and 35.

Figure 32. Photo 36.

Figure 33. Photos 37, 38 and 39.

4.4.2 Session schedule

The factors involved in the definition of a schedule for the RAWSEEDS indoor data-gathering session are too many to allow for a precise statement here. We believe that the acquisition will start during October 2007.

5. Data validation

One of the key differences between the RAWSEEDS toolkit and other datasets and tools available to the robotics community lies in the *validation* of the RAWSEEDS data. In this context, validation is a set of procedures applied to the sensor data acquired during the sessions, to ascertain if they meet some predefined quality requirements. Quality requirements for RAWSEEDS' datasets will be kept stringent, to ensure that the whole structure of the RAWSEEDS toolkit (BPs, BSs) has a layer of high quality data as its foundation.

The fact that RAWSEEDS data have been validated, and that the validation criteria are public and published along with the data, should ensure potential users about their usefulness. In fact the conduction of tests or experiments on software for robot mapping, localization or SLAM can lead to meaningful and reliable results only if the dataset used has been captured with real sensors in the real world *and* if the quality of the dataset is sufficiently high and constant over its extension. The use of low-quality of corrupted datasets can lead to totally misleading results. Unfortunately these problems are (excluding extreme cases) not easy to detect: thus the availability of RAWSEEDS' datasets will give a significant help to all those current or prospective actors in the field of robotics who do not have the hardware, the know-how or the financial resources necessary to build their own datasets.

5.1 Evaluation criteria

Each multisensorial dataset acquired by RAWSEEDS' robots will be validated with regard to the following criteria:

1. file format;
2. timing;
3. data overlap;
4. data density and quality.

The validation process will lead to an extended knowledge of the characteristics of each dataset. This knowledge will be used when defining the BPs, to optimally choose which dataset is the most suitable for each specific problem and to fine-tune the characteristics of the BPs.

In the following of this section we will describe in more detail each of the four criteria listed above.

5.1.1 File format

The dataset must conform to the file formats specified. The dataset must be accompanied by a "dataset description", which is a document describing the structure and contents of the dataset, the platform and the sensors used, the type of environment, the length and duration of the trajectory performed and the number of data acquisitions for each sensor. The description of the dataset should report the calibration techniques used and the values of the calibration parameters obtained.

All the files will be checked to be readable, consistent with the file format specification, and complete according to the dataset description.

5.1.2 Timing

All data acquired by each sensor comes in discrete acquisitions, such as an odometry reading, a laser scan or an image. Each sensor acquisition must carry the timestamp of the instant when it was acquired by the robot. Timestamps can be relative to the start of the experiment or to any other arbitrary time origin. The dataset description must specify the precision of the timestamps for the different elements in the platform. In each file, the timestamps will be checked to be monotonically non-decreasing.
The synchronization between all elements will be checked using the odometry timestamps as time base. Depending on the sensor, its synchronization will be checked using one of these procedures:

- use the sensor data to compute the speed of the platform (for example, using laser odometry or visual odometry techniques) and compare it with the speed obtained by odometry;
- use one sensor acquisition and the odometry readings to predict the next sensor acquisition, and compare it with the actual data obtained by the sensor.

The synchronization will be verified sparsely at different points of the trajectory in order to detect possible clock drifts and unexpected latencies in the sensor stream. The tests should concentrate in points with varying angular speeds, where synchronization errors are easier to detect.

5.1.3 Data overlap

To be able to track environment elements to perform SLAM, any pair of successive data acquisitions from the same sensor must have a significant overlap. Laser scans and images will be verified to have a minimum overlap, at any point of the dataset. As situations of overlap inferior to the minimum could happen during the realistic motion of a real mobile robot, they have to be detected, and "graded" for the amount of overlap, so that the users can know how difficult the dataset is on the average and also in its different subparts. If the condition of not enough overlap is true, it will be documented in the description of the dataset. This could also be the reason for separating the dataset, so to have one with a run where the condition holds.

5.1.4 Data density and quality

The density and quality of the data acquired must be adequate to perform SLAM. For example, images should have enough contrast, be in focus and not blurred due to the platform motion. This will be verified by applying classical feature extraction techniques on the dataset. Laser scans will be processed obtaining straight edges or performing scan matching. Image sequences will be processed by extracting features, such as Harris corners. In all cases, the number of features obtained will be documented in the description file; it will also be mentioned whether they are enough to perform SLAM with state of the art techniques.

5.2 Acceptability thresholds

For each dataset (or group of similar datasets), data quality validation will be performed by selecting a set of parameters to be evaluated and an *acceptability threshold* for each one of them. As sensor datasets differ greatly one from the other as environmental conditions vary, different datasets will be generally evaluated with different acceptability thresholds or even different sets of parameters. However, each BP belonging to RAWSEEDS' benchmarking toolkit will be accompanied by a detailed description of the parameters and thresholds used for its validation.

5.3 Evaluation instruments

Software developed from partners and also public domain software will be used for feature detection; in case of not being possible to find some detector in the consortium or publicly available, it will be developed by the involved partner.

6. Ground truth

One of the RAWSEEDS aims is to enable the evaluation of the performance of different algorithms; for this reason we have the need for a joint collection of the datasets altogether with the appropriate ground truth. Collection of the ground truth means collecting the real value for the variables to be determined by the algorithms that will be then evaluated. In the cases where such values change in time, the collection has to take place at the same time as the sensor data collection.
Please note that RAWSEEDS will not concentrate on the performance evaluation, but it will produce datasets such that the evaluation is enabled. The ground truth is considered as part of the set of data constituting a Benchmark Problem (BP), beside the data collected by the robot sensors.

RAWSEEDS deals with some mobile robotics problems, whose solutions are considered enabling technologies for mobile robotics:

- map building from the data collected by the robot sensors;
- robot pose estimation given a known map and sensor readings (self-localization problem);
- robot pose estimation and map building at the same time (Simultaneous Localization And Mapping problem, or SLAM problem).

The RAWSEEDS effort in collecting the ground truth is not intended for benchmarking sensing devices, which is the task of other initiatives: e.g., the "Stereo Vision Research Page" on benchmarking stereo vision algorithms, maintained by R. Szeliski and D. Scharstein at http://www.middlebury.edu/stereo. Ground truth collected for map building by means of manual environment measurement will not be used to benchmark accuracy and reliability of the sensing devices; it will be used to verify the effectiveness in dealing with map building as a stand alone activity, i.e., with robot pose provided, or as part of SLAM.

Of course, no device is available to measure "the real ground truth", i.e., real position with zero error; instead the best accurate ground truth estimate suitable for common robotics requirement will be provided. This estimate will be integrated with error bounds and/or confidence intervals to be properly compared with the accuracy of the proposed Benchmark Solutions (BSs). In the unfortunate case that the accuracy of the independent measuring device is (or in the time will become) not high enough, the ground truth will be built basing on the output of the best known algorithm. However, the use of the output of an algorithm as ground truth will be taken into consideration only when no alternatives exist, and is regarded as a "last resort" choice.

6.1 Ground truth for localization

Ground truth for the robot position and orientation is necessary for the evaluation of the solutions to both the self-localization and the SLAM problems. In the indoor scenarios considered in this document a potentially interesting devices, e.g., D-GPS, does not work properly and we therefore need to base on a different technology. Given that we are currently considering not to perform stop-and-go acquisition, the manual measurement approach is also not admissible, beside being error prone and very cumbersome.

We are considering two different approaches, the first is based on wireless localization technology, while the second on the usage of cameras and simple computer vision algorithms. Both allow for the collection of ground truth in reduced areas, w.r.t. the whole explored area. Each such area is called a cell.

Generally speaking wireless localization systems are characterized by a localization accuracy that is quite bad. The best of them, to our knowledge, is the one used by the FIFA football federation for checking the ball position in the soccer field; it is considered not usable for RAWSEEDS, for the large amount of cross-talk expected in indoor environments. The second best known system is the Ubisense system (http://www.ubisense.net), which is able of localizing its *tags* with good precision within limited zones of space. One RAWSEEDS partners (UNIMIB) has already acquired such system, and POLIMI will complement that system with accessory hardware in order to increase the accuracy and/or increase the number of zones of space where the ground truth can be measured. In order to clarify this aspect, we give a short introduction to the system usage. The Ubisense system is based on beacons and tags. The first are positioned in fixed locations, while the second are attached to the object to be localized. The system uses the radio band from 5.8GHz to 7.2GHz. The device uses two positioning methods, TDOA and AOA. Time Difference of Arrival (TDOA) is the most straightforward way to estimate the position. In this method the time difference of transmitted signal arrival at two base stations is measured. From time differences can be drawn hyperbola. In two-dimensional locations at least two pair of base stations are needed, and can be composed from three base stations. In three-dimensional locations four base stations are needed. The angle of arrival method (AOA) is also known as the direction of arrival. The idea is to position and measure the angle of arrival of transmitted signal. For capturing the angle of arrival an antenna array or directional antenna is needed. Each measurement composes a line between the tag and base station. The advantage of this method is that synchronization and accurate timing references are not required in the base stations. Still, a calibration in order to compensate antenna differences and the variations of humidity and temperature are required. The angle of arrival method can be executed using two base stations. Knowing the coordinates of them, coordinates of the tag can easily be calculated using basic geometric operations. A Ubisense cell can be setup using a certain number of base stations. The maximum range of the Ubisense cell is about 20m. Inside a cell a number of objects, carrying a tag, can be tracked real-time in 3D (position update at a rate of up to 20 times per second). In order to track the 6DoF robot pose, and also to track it as a 3DoF object in the plane, which is an easier task, we need more than one tag on the robot (2 tags for the 3DoF robot in the plane, 3 for the full 6DoF). Having such number of tags is not a problem, but the relative position of the tags could be critical, in order to have a good estimate of the orientation. Anyway, provided that the onboard tags could be put far enough to each other, a number of them (larger than the above mentioned minimum values) can be used for increasing the accuracy of the estimates. Time averaging can also be used for increasing accuracy, both in position and orientation, but it is still unclear in what measure the noise on the measurements can be considered White Gaussian; it could turn out that there are significant biases that are changing with the spatial position of the robot, therefore making impossible to estimate the bias.

The second considered approach bases on the usage of cameras and simple computer vision algorithms to locate the robot. The cameras are different from the robot ones, and positioned on the ceiling of the environment. By using simple software, typically already available in the Internet, some markers can be located in the scene, from their image. These data is then converted into a robot localization. Cameras need to be calibrated w.r.t a single reference frame and this turns into having some superimposition in the fields of view of neighboring cameras. Even with this constraint, we expect to be able to cover larger areas w.r.t. Ubisense cells. For what concerns the accuracy attainable, we expect about 40mm / pixel in the X-Y plane, that, considering the noise acting on the system (small changes of pose of the camera after calibration, errors in the localization of the image of the robot markers, inaccuracies in the calibration of projection parameters, etc.), is more or less at the same level of the accuracy attainable from Ubisense (100mm at the very best, 150mm is a realistic expectation). The camera-based system might allow much larger cell, i.e., a longer GT-ed trajectory. For the Z and/or the pose, we expect the accuracy to be also comparable to the one of the Ubisense system.

Concluding, both systems at the moment appear interesting although the setup of the Ubisense, as of today, looks more error prone and expensive, and the expected cell-size is smaller, given the budget.

We think that our approach, which can be in short described as "ground truth only in some(/one) area(s) along the path", is realistic and good enough for many years forward (of usage of RAWSEEDS datasets).

6.2 Ground truth for mapping

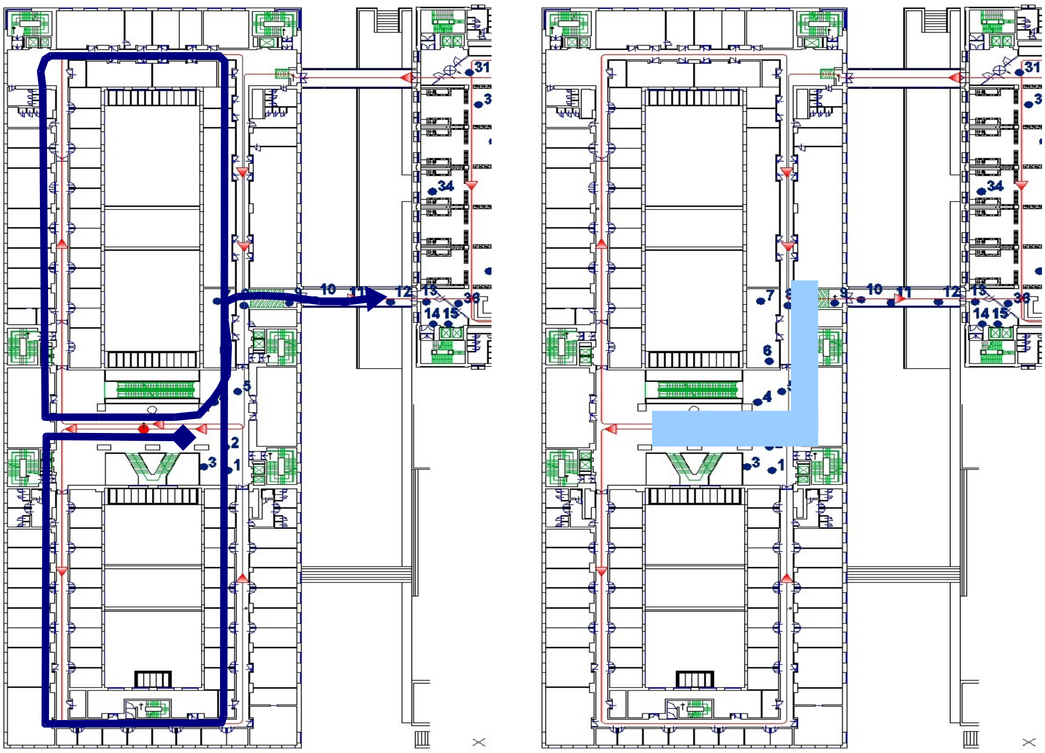
The ground truth for the mapping aspect requires accurate collection of data about the robot environment. For such purpose we have a quite diverse set of solutions.

- Executive drawings, integrated by measurements (made by hand or by a *total station*) for those items (e.g., furniture) which are not featured in the executive drawings.
- Maps collected independently from the robot dataset, e.g., by means of a total station or by some laser scanner, under the control of a human operator; on these maps the ground truth will be computed only on the relative position between pairs of well-defined environment landmarks, e.g., vertical edges. This will allow comparison of maps obtained by the algorithm under evaluation with reference to the ground truth.
- Maps collected by the robot, e.g., by its laser range finders, sticked together by the "GroundTruth-ed" robot pose. This approach requires a "good-enough" ground truth for the robot pose, so to get to a reasonably good ground truth for the map.
- Maps produced applying the best state-of-the-art mapping algorithms to the data output by the most accurate available sensor, e.g., laser range scanners.

Our current expectation is to base mainly on the executive drawings and to eventually integrate them with measurements for the item not featured in the drawings, if their presence is such that there are not enough other landmarks available whose relative can be considered as ground truth.

6.3 Area where the ground truth collection is planned

In blue on the right is depicted a potential path in one of the buildings of the Bicocca location, where we are planning to have the ground truth collection cell. In light blue on the left is depicted the quite large area that we are planning to cover for the collection of the ground truth. It is where loop closure will likely happen in the SLAM activity, i.e., where it is interesting to compare the robot perception and the reality.



7. Benchmark Problems

A Benchmark Problem, or BP, is defined as the union of: (i) the detailed and unambiguous description of a task; (ii) an extensive, detailed and *validated* collection of multisensorial data, gathered through experimental activity, to be used as the input for the execution of the task; (iii) a rating methodology for the evaluation of the results of the task execution. The application of the given rating methodology to the output of an algorithm or piece of software designed to solve a Benchmark Problem produces a set of scores that can be used to assess the performance of the algorithm or compare it with other algorithms.
[*From RAWSEEDS' Description Of Work (Annex I to the Contract between Partners and EU).*]

The creation of the BPs is the work of WorkPackage 4, and therefore it is not possible at present to give detailed information about which specific BPs will be created and included in the RAWSEEDS toolkit, or even about the number of BPs which will be defined. These are issues that it will be possible to settle only at the end of the validation process, when the characteristics and peculiarities of the available datasets will have been analyzed. The data validation process will also lead to identify which datasets, or parts of datasets, are more suitable for a specific type of problem, and which are more or less "difficult" to work on.
WP-1 had the task of choosing which *kinds* of Benchmark Problems RAWSEEDS will produce, i.e., of defining the general characteristics of those problems. These characteristics will be described by the following sections; which and how many specific problems will be published will become known as the work of WP-4 proceeds.

7.1 Problems

The BPs that we will create using the RAWSEEDS' indoor datasets belong to three categories, covering a wide range of applications in the field of mobile robotics. In fact, different types of problem can arise within the general context of letting an autonomous mobile robot move through an indoor environment.
The three categories of BPs are the following.

7.1.1 Mapping

This is the problem of developing an algorithm that, provided with:

1. a dataset composed of sensor streams recorded by a mobile robot exploring an environment,
2. the trajectory of the robot through the environment,

is capable of producing a *map* of the explored environment, i.e., a formalized description of its geometric characteristics. The trajectory supplied to the algorithm is extracted from the ground truth associated to the dataset.
Many different kind of maps can be conceived: line segment maps, occupancy grid maps, and so on. Therefore it will be necessary to specify, in each BP, which kind of map is required from the solution algorithm. Of course, it will be possible to define

multiple BPs differing only for the kind of map.

7.1.2 Localization within known map

This is the problem (simply called "localization" in the following of this document) of developing an algorithm that, provided with:

1. the map of an environment,
2. a dataset composed of sensor streams recorded by a mobile robot exploring the same environment,

is capable of reconstructing the trajectory of the robot through the environment, i.e., the set of successive positions taken by the robot in correspondence to a given set of time instants (possibly reducing to a single instant), in the reference time-frame of the dataset. The map supplied to the algorithm is extracted from the ground truth associated to the dataset.
The critical issue for this kind of problem is: which kind of map will be made available to the algorithm? We are still exploring this issue, but two possible approaches (not necessarily mutually exclusive) are:

- a technical drawing, e.g., a .dwg file, which describes an idealized (but accurate with respect to reality) version of the explored building;
- a "reconstructed" map generated by one of the mapping algorithms included in the RAWSEEDS toolkit as BSs, when applied to a dataset different from the one in the BP, but extracted from the same environment.

Of course the use of a reconstructed map poses the problem of choosing one type of map over the others.

7.1.3 SLAM (Simultaneous Localization And Mapping)

This is the problem of developing an algorithm that, provided with a dataset composed of sensor streams recorded by a mobile robot exploring an environment, is capable of producing a map of the explored environment *and* an estimate of the trajectory that the robot has followed.
SLAM problems require the same kind of data used in mapping problems plus odometry data, which are useful to calculate a rough estimation of the pose of the robot; pure mapping problems, on the contrary, do not need odometry data as robot trajectory is known.

In addition to the previous categories of BPs, and only if the datasets will be considered appropriate during the validation phase, BPs covering the *relocation problem* could be included into RAWSEEDS' toolkit. This kind of problem can be defined when two different datasets covering the same zone of space are available. The first dataset is used to generate a map of the environment, while a subset of the second is extracted and used as the input for an algorithm which has the task of comparing the subset with the map and determining the positions of the data-gathering robot in the map.

As previously said, one or more BP for each of these (indoor) categories will be included in the RAWSEEDS benchmarking toolkit: when a category will include multiple problems, they could be differentiated by the specific dataset they are based on, their difficulty level and/or the presence of *loop closures* (i.e., multiple passages of the robot from the same zone of space) in the dataset.

7.2 Data representation and file formats

Each of the streams generated by the sensors of the data-gathering robot needs to be recorded, stored and published within the RAWSEEDS toolkit. Therefore a data representation format, and an associated file format, must be selected for each kind of sensor. This choice is not easy as it must be a compromise among conflicting requirements, such as:

- *fidelity*: the data must not be altered or corrupted by the formatting process;
- *compactness*: the datasets must be as small (in terms of byte count) as possible, because they will mainly be distributed as downloads from RAWSEEDS' website;
- *ease of use*: it must be easy to extract information from the datasets;
- *compatibility*: if possible, standard formats should be used.

This is still a mostly open issue, and will require further research before final settling, as we will need to be certain to have made the best possible choice before committing definitively.

The main problem we are facing is the choice of video data representation. A camera outputs a huge quantity of data: for example a single color VGA camera with 30Hz frame rate and 24-bit color representation, e.g., a standard low-market webcam, produces a 210Mbit/s bit stream. RAWSEEDS' robot is fitted with multiple cameras, and the data-gathering sessions will have a duration of many minutes each, thus leading to very massive datasets. This is a problem, as we would like to make the duration of the download of RAWSEEDS' complete toolkit as short as possible (please note that we are considering a data volume of tens or hundreds of GB, so even with a very fast Internet connection downloads will take at least many hours). It is possible, of course, to split the dataset into smaller portions: and it will be done. However, minimizing the amount of data describing the video streams is highly desirable.
The usual solution to this problem would be the use of *compressed* data formats, as lossless compression gives low compression ratios. However, the use of lossy video coding in the context of RAWSEEDS is considered not applicable, not only because of the obvious degradation induced in the data, but also because the model of such degradation is *not* easy to be taken into account, when processing the data. Moreover, and worst of all, the wide-spread lossy video codecs are so-called *perceptual* codecs, i.e., designed to minimize the subjective degradation, as perceived by humans. The effect of these codecs on the performance of an algorithm applied to the compressed video stream is not only unknown, but most likely very variable with both the structure and the implementation of the algorithm. We are also considering to allow subsequent upload of an higher-level level version of the same data-set, i.e., the output of some processing like, e.g. sift or corner detectors. Although we fear that then arguments could start about the actual implementation and parameterization used, we think this could be an advantage for same research groups.

7.3 Evaluation methodologies for the solutions

The solution to a BP is an *algorithm*. When this algorithm is applied to the data of the BP, it produces an output (which, if the algorithm has to be put in the form of a Benchmark Solution, must have a strictly specified representation; BSs will be described in the following of this document). This output can successively be evaluated to analyze the performance of the algorithm: for this purpose, each BP includes the detailed description of an *evaluation methodology*. This methodology can be applied to the output of the algorithm manually or through the use of a purpose-built piece of software, possibly supplied by RAWSEEDS: in any case, the application of the evaluation methodology to the output of the algorithm generates a *rating* (or, more likely, a set of ratings). This rating is a quantification of the performance of the algorithm under test, and can be used to compare it to different algorithms, such as the ones proposed by different groups.
It is extremely important to note that such rating is *arbitrary*: it depends on the specific dataset included in the BP and on the choice of rating methodology. However, it gives a rough way to compare different algorithms, which is something that has always been very difficult in robotics. We would like to stress that the purpose of RAWSEEDS is not the one of compiling and publishing a "hit parade" of the more successful algorithms for mobile robotics (which we hope people will submit to RAWSEEDS for publication in the form of BSs), nor to certificate the performance of algorithms, as the published ratings will always be measured by the authors of the BSs themselves. On the contrary, RAWSEEDS wants to contribute to the progress of robotics by publishing a set of instruments - the benchmarking toolkit - useful to develop, evaluate and perfect algorithms for mobile robotics. The fact that the evaluation ratings can, with attention to all the pitfalls associated to such an operation, be used to compare the performance of different algorithms when applied to the same data is certainly useful for research and development but is not, in any measure, the focus of RAWSEEDS' activity.

Defining the specific evaluation methodologies to be included in each of the BPs is part of the work of WP-4; here we will describe the chosen methodologies for the three category of BPs described in the previous sections, as this choice was part of WP-1.

7.3.1 Evaluation methodologies for mapping BPs

The output of an algorithm that solves a mapping BP is a map of an environment. The evaluation methodology associated to the BP is the comparison of the reconstructed map with the "reference" map of the environment, which is included in the ground truth associated to the environment.
This comparison is usually made by comparing the position in the maps of specific *landmarks*, i.e., points that are both important for navigation through the environment and easy to identify, such as corners or borders of the walls. A (pre-defined) set of landmarks is chosen on the reference map, and then the same landmarks are searched in the reconstructed map: the ratings of the algorithm are then defined in terms of presence and correct positioning of the landmarks in the reconstructed map. Examples of such ratings are the percentage of landmarks that can actually be identified in the reconstructed map, or the mean error obtained when comparing the distances between couples of landmarks in the reconstructed map with the same distances evaluated in

the reference map.

In RAWSEEDS' BPs we will probably include such kind of evaluation methodologies, as they are (relatively) easy to apply and well established in literature. However, we would like to introduce an element of doubt about the fact that such methodologies are really the best for robotic applications. In fact such geometric methodologies measure the geometric quality of the map when compared to a reference (i.e., correct and complete) map: in other words, they measure the capability of the mapping algorithm to produce good maps. Our point is that this is *not* the purpose of a mapping algorithm for robotics. The real purpose of such an algorithm is, instead, the one of creating a map that, when used by a robot to navigate into the real environment, allows the best performance of the robot in terms of reaching its goals (e.g., going to a target position with maximum precision and minimal travel time). This requires the map to be good, for example, as an instrument for planning or for obstacle avoidance: but it is absolutely possible that, given a map with good geometric accuracy and one with bad geometric accuracy, the second will lead to much better performance of the robot in the real environment. For example, in planning it is more important to know which passages (such as doors) exist between the rooms of an office than to know the exact position of each passage: a map which has perfect geometry but shows a wall instead of one of the doors could well lead to disastrous results, up to the incapacity to perform a given task.
We therefore advocate the need to define new metrics for the evaluation of maps and of mapping algorithms, more closely tailored on the real usage objectives of those maps in robotic applications. We are currently working on this subject and will present our proposals in future occasions.

7.3.2 Evaluation methodologies for localization BPs

The output of an algorithm that solves a localization BP is the trajectory of the robot through an environment. A trajectory is defined as a succession of data, each having the form {pose of the robot, time}. The evaluation methodology associated to a localization BP is the comparison of the reconstructed trajectory with the real one, as included in the ground truth associated to the BP. That comparison can be made on the overall trajectory, i.e., making a geometric comparison of the shapes and positions of the real and reconstructed trajectories; or on a point-to-point basis, i.e., evaluating the distances between each pose of the reconstructed trajectory and the pose of the real trajectory associated to the same time instant, which would imply the definition of a distance between poses. Both categories of comparison can be applied to the same trajectory and used as a basis for the definition of performance ratings.

7.3.3 Evaluation methodologies for SLAM BPs

As the SLAM problem is the union of a localization problem and a mapping problem, the evaluation methodologies associated to a SLAM BP are the union of those associated both to the mapping BPs and to the localization BPs. In addition to that, it is possible to define SLAM-specific methodologies, such as the evaluation of *loop-closure error*. When the the dataset includes a *loop*, i.e., the trajectory of the robot returns to a previously visited point, a SLAM algorithm that generates and updates the map, as the robot proceeds on its trajectory, has a means for correcting the errors on the estimated

pose of the robot: in fact it possesses two different estimates of the robot's pose, in two different time instants, knowing that they are coincident. Forcing this coincidence gives additional constraints on the trajectory of the robot and greatly reduces the errors due to imperfect odometry. Moreover, as the features of the reconstructed map has an estimated position in space which has been calculated in reference to the estimated trajectory of the robot, when the trajectory is corrected by "closing the loop" the map is subject to correction too, and its precision rises. Loop-closure error is the error between the estimated pose of the robot (or the position of some feature of the environment) when reaching the end of a loop and the modified pose of the robot (or position of feature) after the correction due to the closure.
A notice here is about the collection of the real pose (ground truth): it might happen that the algorithm matches, i.e., closes the loop, before entering the ground-truth collection cell. This might interesting, for rating the algorithms

7.3.4 Measurement of running time

In robotics, when a promising algorithm for a data-elaboration task has been devised, it will typically be implemented into a computer program to be tested. Therefore, one of the most obvious evaluation methodologies that can be applied to an algorithm as applied to a specific set of data, is the measurement of the running time of the associated computer program. The idea is, of course, that - everything else being equal - a faster-running algorithm is better than a slower one.
In practice, the introduction of running time measurements among the evaluation methodologies of RAWSEEDS' Benchmark Problems introduces many difficulties and could even lead the RAWSEEDS community of users and generators of BSs (which we hope to establish) into a spurious drive towards optimizing the wrong aspects of their algorithms. Therefore we are oriented towards *not* including running time measurements into the evaluation methodologies of the BPs: in the following part of this section we will explain why.
As a matter of fact, RAWSEEDS users will be able to include notes, documents and additional material in the BSs that they will submit for publication. Therefore they will be able to add running time measurements and hardware descriptions to their BSs, and will even be encouraged to do so. But this is much different from the introduction of *performance ratings* based on running speed.

The main problem associated to the use of evaluation methodologies based on running time is the lack of uniformity in the software and hardware implementation of different algorithms that solve the same problem.
Differences in the performance, architecture and characteristics (e.g., available RAM) of the computer used to perform the running test can lead to large differences in running time; different running times can even be observed when the same machine is used, in dependence to the state of all the other running processes which compete with the one under test for the available resources.
Even more significant are the differences in running time that can be observed when using different programming languages to implement the same algorithm. For example, many researchers use the mathematical programming environment Matlab, which is based on an interpreted language, as a rapid development tool for testing

algorithms; in many cases this leads to a massive slow-down, in comparison with an implementation of the same algorithms based on a compiled language, such as C++.
Last but not least, the running time of a program implementing an algorithm depends heavily on the way this implementation has been coded (i.e., the way in which the source code algorithm has been designed and written). This dependence is in turn greatly influenced by the specific data that the program is called to elaborate and by the specific machine it is running on: for example, an implementation which uses few lines of code but a large memory space will run fast on machines with much RAM and/or on small datasets, but will become slower when data amount grows or available memory is scarce, to finally become unusable when the RAM is full and the operating system is forced to perform page swapping to the hard disk drive.

As it is impossible to include all of these elements into an evaluation methodology for algorithms, the only way to define (apparently) meaningful running time ratings of implemented algorithms would be to define a "standard computer", perfectly specified in every detail of its hardware and software, that each BS contributor is invited to run its program on. As forcing every RAWSEEDS user to buy such a machine would be absurd, RAWSEEDS would be forced to set up a "reference computer", accessible remotely (e.g., via SSH), on which contributors could perform their running tests. With, of course, associated precedence rules, waiting lists and so on. This is enough to exclude the feasibility of the use of running time as an evaluation methodology, because we do not want to use RAWSEEDS' resources to set up such a "benchmarking service" instead of working on a realizing and distributing a quality toolkit. All the more so because the ratings that such a service would produce would be arbitrary, because the choice of the machine is arbitrary. And not taking into account other factors, such as the fact that everyone not developing software on the operating system chosen for the reference computer would be *a priori* excluded from the possibility of publishing BSs.

All of that said, there is a category of algorithms for mobile robotics that focuses exactly on time-related performance: the so-called *on-line algorithms*. These algorithms (or, to be precise, the computer programs implementing them) are designed to use the data coming from the sensors as it is made available, without storing it and executing computations at the end of the exploration. RAWSEEDS' datasets can be used by such an algorithm, provided that they are "played" by a suitable software system that, by reading the time information associated to the data, supplies them to the algorithm at the exact rate that they were originally recorded.
We expect the importance of on-line algorithms to grow over time, as this is the only class of algorithms really suitable for use with autonomous robots that interact directly with human beings. However, anyone wanting to use RAWSEEDS' datasets to test a real time algorithm would be forced to design a specific *player software*, capable to read the datasets and perform the operation described above. It is possible that (as a project collateral to RAWSEEDS) we could be able to supply such a *player* along with the RAWSEEDS toolkit, and publish its source code on RAWSEEDS' website: but at the moment this is only a possibility.

One last idea, still related to runtime, is to perform many evaluations, implying a relative increase of the runtime, by using, for example, longer trajectories, more features, etc., in order to increase the workload of the algorithm. This would lead to an "experimental complexity analysis" (O(n), O(n2), etc.), which would be interesting, as the relative runtime, although still machine-dependent, will shed much more light on the solution under evaluation than the absolute.

8. Benchmark Solutions

A Benchmark Solution, or BS, is defined as the union of: (i) a BP; (ii) the detailed description of an algorithm for the solution of the BP (possibly including the source code of its implementation and/or executable code); (iii) the complete output of the algorithm applied to the BP; (iv) the rating of this output, calculated with the methodology specified in the BP.
[*From RAWSEEDS' Description Of Work (Annex I to the Contract between Partners and EU).*]

RAWSEEDS' BSs can be useful in two ways: as a benchmark, i.e., as something that another competing solution can be confronted with; or as a source for high-level data, because the data output by a BS algorithm (and included into the BS) can be used as the input of algorithms performing higher-level tasks. For example, a researcher or company that has developed a software for SLAM can apply it to one of RAWSEEDS' SLAM BPs and directly compare the ratings of its own software (calculated with the methodologies specified by the BP) to those obtained by the published BSs solving the same BP. On the other hand, a researcher or company that has developed a software which performs planning activities and uses the output of a SLAM system as its input, does not need to also develop the SLAM system to be able to test the planner: the output of such an algorithm - when applied to a known dataset - is already available from RAWSEEDS.

8.1 Solution algorithms

At the moment it is not possible to precisely define the characteristics of the BSs that will be included into RAWSEEDS' toolkit. Decisions about that, in fact, depend heavily on the nature of the specific BPs that the BSs will be used to solve, which in turn will be known only after the data-gathering, data validation and BP preparation tasks are completed (or almost completed). Therefore the following of this section is adapted from RAWSEEDS' Description Of Work and describes the kind of algorithms that will most probably be employed, and the expertise of RAWSEEDS' Partners in the definition and application of such algorithms.

POLIMI will have a secondary role in WP-5, mainly centered around the application of algorithms for the fusion of multiple sensory streams (coming from cameras, odometry, accelerometers, etc.) to the RAWSEEDS data sets and to the participation to the evaluation of algorithms for 3D-6DoF Hierarchical SLAM with trinocular vision in collaboration with UNIMIB. The activities of WP-5 will be mainly performed by UNIMIB, ALU-FR and UNIZAR.

UNIMIB will provide techniques for solving the Simultaneous Localization And Mapping (SLAM) problem based on Six Degree of Freedom Hierarchical SLAM on the RAWSEEDS data sets, and will provide one or more BSs based on 3D segment-based environment reconstructions (relying mainly on data coming from trinocular vision). UNIMIB will provide detailed descriptions of all the algorithms used to generate the BSs. The BSs generated by UNIMIB as part of the work of WP-5 and the associated 3D reconstruction of the environments will provide other researchers and enterprises with fundamental comparison material for the development of novel algorithms for SLAM, image processing and 3D reconstruction.

ALU-FR will contribute to WP-5 with its expertise in learning maps with mobile robots. In particular it will use the techniques for solving the SLAM problem based on Rao-Blackwellized particle filters on the RAWSEEDS data sets. Based on these algorithms ALU-FR will provide optimized trajectories that can be used by other researchers to evaluate their own algorithms. ALU-FR will also use offline techniques for solving the SLAM problem to generate trajectories for these types of applications. Additionally, ALU-FR will apply terrain classification techniques to the maps generated from the data sets: in this way, other groups can compare their results to novel techniques found in the literature. All the results obtained with the exploration techniques developed by ALU-FR will be transformed into BSs. The BSs will include, in particular, techniques for single-robot exploration as well as multi-robot exploration. In addition to the data and the results, ALU-FR will provide as part of the BSs the source code of the algorithms used to generate the results. This will allow researchers to more systematically evaluate the algorithms, understand their sensitivity with respect to design parameters, and also to easily integrate extensions to them.

UNIZAR will contribute to WP-5 its experience in robot localization and map building using different sensing modalities, which include laser range-finders, sonar and vision with one or more cameras. The group is well known internationally for having developed one of the first SLAM systems based on the Extended Kalman Filter as estimation technique. UNIZAR will evaluate the data sets obtained using different sensors by the RAWSEEDS project and will produce BSs using the EKF-SLAM approach. Each BS will include the complete trajectory computed for the robot and the map obtained of the environment. UNIZAR will also perform, and publish in the form of BSs, experiments of robot relocation (currently a very active research field): that is, given a previously built map and a new set of data obtained in the same environment, the computation of the possible robot localizations. This will provide a set of benchmarks allowing other researchers to compare with their own relocation algorithms.

8.2 Web-publishing and IPR policy for of user-generated BSs

The work of RAWSEEDS is open-ended, in the sense that we do not consider it concluded with the Internet publication of the benchmarking toolkit. In fact, the RAWSEEDS website is intended not only as a repository for the toolkit, but also as a meeting point for the whole robotics community and as a means for the exchange of knowledge. To this end, user participation will be encouraged: both by exchanging opinions in the RAWSEEDS forum and by submitting new content for publication. This content will be published along with that produced by the RAWSEEDS project, after having been evaluated by the website administrator.

As RAWSEEDS-published content is composed of BPs and BSs, in principle a user could contribute new items of both categories; in practice, though, the kind of new BPs which is more likely to be accepted for publication are (at least in an initial phase) ones that are based on RAWSEEDS' own datasets. The reason for that lies in the fact that the RAWSEEDS website will publish new datasets only if and when they are validated and certified with quality standards similar to those used for RAWSEEDS' own

datasets: so it is likely that only organizations with the financial resources and technical expertise necessary to set up a high-profile data-gathering and validation campaign will be able to propose BPs based on completely new datasets.
On the other hand, no such limitations exist for BSs: any user of the RAWSEEDS website will be able to submit new BSs with little effort, as they do not have to fulfill any *a priori* quality requirement. All that will be asked to new BSs is the compliance with the publishing policy that we will describe in this section.
By the way, RAWSEEDS will not publish only BSs based on novel algorithms: BSs using algorithms known in the literature are almost as useful, because being able to compare their performance with that of the newer ones is important.
Publishing content on the RAWSEEDS website will not mean losing intellectual property over it. On the contrary, RAWSEEDS have devised a very flexible system to manage IPR (Intellectual Property Rights) and licensing issues, with the aim of leaving each contributor free to define exactly which rights on the content she/he is publishing will be granted to the public. We feel that this a critical point in convincing private companies that it is in their own interest that they publish their results on RAWSEEDS' website, without regarding it only as a source for knowledge.
As these are important matters for the success (or lack of success) of RAWSEEDS, we did clarify our ideas well in advance of the start of the project. Consequently, a clear statement about these issues was already present in RAWSEEDS' Description Of Work, and the following part of this section is an adaptation of that.

As previously said, the material that will be published by the RAWSEEDS website can be divided into two broad categories:

1. material produced by the RAWSEEDS project itself;
2. material voluntarily submitted for publication by the users of the website.

Both will be subject to the same IPR regime, which we will now describe.
RAWSEEDS requires that any material published on the website complies with the following three requisites:

- **R1**: its creator chose to make that material publicly available;
- **R2**: the rights granted by the creator to the users of the material are clearly and explicitly defined;
- **R3**: the material must qualify as *useful* and *appropriate* for publication.

Requisite R1 is automatically satisfied by the fact that it is the free choice of each user of RAWSEEDS if he/she wants to publish any of his/her creations, and which ones (if any). The upload process will require that a the user has been registered, so that anyone who contributed to RAWSEEDS is identifiable, if needed. During the registration process the user will be warned that any material submitted for publication will, if approved, become publicly available.

To meet requisite R2, any contributor to RAWSEEDS will have to accompany the proposed material with a *license* stating which rights he/she reserves to himself/herself and which are instead granted to the public. So intellectual property of any material published by RAWSEEDS will remain to its creator, who (with the act of submitting the

material for publication) chooses to relinquish part of the rights on it to the public. Which rights are actually given to the public is defined by the chosen license.
To foster the sharing of knowledge and avoid limiting submissions by external partners, including SMEs and companies of any kind, RAWSEEDS will *not* force its contributors to choose a specific license. On the contrary, it will leave the choice of the license to the user, with the only constraint that a copy of the chosen license must be sent along with the material submitted for publication. In case of publication, the license will be published on the website, and it will be possible to download it for inspection before downloading the material it is associated to.

Nonetheless, one of the aims of RAWSEEDS is the promotion of a diffuse *sharing attitude* in the robotics field: we are convinced that the more "open" a contribution is (i.e., the more freedom is conceded to the user over its possible uses) and the more useful it will be - in the long run - for the progress of this sector, both from the scientific and the commercial points of view. For this reason on the RAWSEEDS website will be present a list of **RAWSEEDS Suggested Licenses** (RSLs), which will be proposed as "suggested choices" in the process of submitting any new material. Of course, this process will also include (with the same visual evidence) the option to choose a license other than the RSLs. Amongst all existing licensing schemes, the RSLs will be chosen with the following objectives:

- to maximize diffusion and usefulness of the published contributions, while at the same time granting effective protection of the intellectual property of their creators;
- to encourage the formation, among the users of the RAWSEEDS website, of a cultural climate favoring the *sharing* of results and tools;
- to constitute a sufficiently flexible set to fit most needs, but not so extended as to be confusing.

A complete list of the RSLs, with links to the associated web pages, will be maintained on the RAWSEEDS website. Anyone wanting to submit a contribution under one of these licenses will not have to manually upload the chosen license, as that will be made by the website itself; on the contrary, licenses not belonging to the RSL set will have to be manually uploaded. Initially the RSLs will include the following licenses:

- for any kind of material, all the licenses issued by Creative Commons (CC is a nonprofit organization that offers flexible copyright licenses for creative works): a description of those licenses can be found on the Internet at the address http://creativecommons.org/about/licenses/meet-the-licenses;
- in addition, and mainly for software in source code form, all the licenses certified by the Open Source Initiative (OSI is a non-profit corporation dedicated to managing and promoting the "Open Source" definition), whose list is available on the Internet at the address http://www.opensource.org/licenses.

To be included in the RSL set, a license will have to conjugate strong protection of IPR from the legal point of view with the possibility, for the users of the licensed material, to effectively build (and possibly publish) new results over the previously published ones, without violating the license. While OSI-certified licenses strongly limit the

amount of rights retained by the content creator, the CC licensing scheme includes a broad spectrum of possible licenses, with widely different characteristics. CC licenses can be applied to source code, so it will be possible (if the creator will choose to) to publish source code as RSL licensed material and nonetheless not as Open Source Software.
Any material produced within the RAWSEEDS project (i.e., by the partners of the RAWSEEDS Consortium) and published on the RAWSEEDS website will be subject to a license belonging to the RSL set. Intellectual property of this material will be of the specific partner which produced it, which will also choose the license.

Finally, to meet requisite R3 any contribution will have to be previously examined and approved prior to publication. To be published, any contribution will need to be *useful* and *appropriate* according with the following definition:

> *Any publishable material is considered useful and appropriate for publication on the RAWSEEDS website if* all *of the following are true:*
>
> - *it is related in some way to the field of robotics;*
> - *it can help, in some way, progress in the field of robotics;*
> - *it is sufficiently detailed to be usable (e.g., the description of an algorithm must be complete enough to allow a reader to implement that algorithm into a piece of software);*
> - *it is usable (e.g., executable code is usable only if it is actually working and accompanied by all the information needed to install and configure it);*
> - *it does not have commercial purposes only (e.g., a company selling robotic products could publish the description of a product, but that description will need to disclose enough data about the product to be considered a worthwhile contribution to the field rather than a marketing operation).*

The content administrators of the RAWSEEDS website will have the right to withdraw from the site any already published material which - at a successive re-evaluation - is judged to be non-compliant to the above requisites R1, R2 and/or R3. This includes also, but not only, the case of materials that are outdated due to technical advancement or availability of better equivalents, or materials that have been confuted totally or partially. Each contributor maintains the rights over her/his published material stated by the license chosen during the publication phase, but has not the right to force the removal of that material from the RAWSEEDS website, as with the act of publication it has become a (limited) public property. This limitation will prevent the creator of a contribution over which many others have worked and constructed afterwards, from having the power to lower the value of those successive works by removing his/her own work from RAWSEEDS. However, it will be possible to send motivated requests of content removal to the content administrators of the RAWSEEDS website, after which they will freely decide if and when to perform the requested removal.

8.3 Formatting requirements

Each Benchmark Solution (both the ones generated by RAWSEEDS and the ones submitted by the users of the website) has a structure comprising three categories of elements: loosely-specified elements, strictly-specified elements and extras.

The loosely-specified elements are the components that describe the solution algorithm. As the algorithm can be supplied in many different forms (e.g., detailed description, source code, executable code with installation support), there are no strict requirements on this description except that it must be *useful* and *accurate*, in the sense illustrated in the previous sections of this document.
The strictly-specified elements are:

1. The output of the algorithm, when applied to the dataset included in the BP that the BS is a solution of. This output must comply perfectly with the data structures and file formats defined by the BP.
2. The ratings of the output of the algorithm, as obtained by the application to it of the rating methodologies defined as part of the BP.

The reason for the rigidity over the output format is the fact that in this way an algorithm that uses the output of a certain BSs as its input will also be applicable without modifications to the output of any other BS associated to the same BP. This makes the testing of such an algorithm much easier.
The rigidity over the ratings is necessary to assure that the ratings obtained by different BSs (associated to the same BP) are directly comparable.
The extra elements are anything that the creator of a BS thinks will be useful to anyone interested to the BS. For example: suggestions on the efficient implementation of similar algorithms; description of possible modifications to the algorithm and of their effects; analysis of the characteristics of a dataset that make it more or less "good" for the algorithm; analysis of the parameters of the algorithm and of the sensitivity of the latter to their modifications; notes on the comparison of the BS with alternative BSs; running times for the software implementation of the BS when run on different types of computers; screen shots; and so on.

9. Documentation and manuals

The specifications and/or user manuals for all the devices involved into the indoor data acquisition activity of RAWSEEDS will be published on the project's website. In this way anyone wanting to know the (declared) performance and functioning modes of such devices, or contemplating the replication of our data-acquisition experiments, will be provided with all the necessary data.