Project Rawseeds

General concepts and definitions for BPs

This document defines the main concepts and terms associated with Rawseeds' Benchmark Problems. It is meant as a guide to the use of BPs, and especially as a help to users who are developing their own Benchmark Solutions. More information about the concepts illustrated here can be found in the Deliverables of project Rawseeds.

Table of Contents

1. Operative definition of Benchmark Problem (BP)

A Benchmark Problem is the union of:

1. *the detailed and unambiguous description of a task;*
2. *a set of rating methodologies, to be used for the evaluation of the data produced by the execution of such task.*

Each BP acts as a template for a category of problems: i.e., it is not a specific, solvable problem. Such a specific problem is called a **BP instance**, and it is obtained by complementing a BP with the following element:

3. *a collection of sensor data, gathered through experimental activity, to be used as the input for the execution of the task.*

2. Operative definition of Benchmark Solution (BS)

A Benchmark Solution is the union of:

1. *a BP instance (or a reference to it);*
2. *the description of an algorithm for the solution of such BP instance;*
3. *the output of the algorithm, when applied to the data of the BP instance;*
4. *the results obtained by applying to such output the evaluation methodologies specified by the BP instance.*

The above results can be used to assess the performance of the algorithm, or to compare it with other algorithms.

To be considered as a valid, a Benchmark Solution must include a given set of elements, described below in a specific section of this document. Valid BSs can be submitted to Rawseeds for publication on the project's website, along with those of all other proposers (including the project itself). This can be an effective way of publicizing one's own achievements and to spread knowledge about them.

3. Ground truth (GT)

In the context of Rawseeds, the term *ground truth* (often shortened to “GT”) describes any source of precise information about the environment explored by the robot and about the movements of the robot that does not derive from the data collected by the sensors on board of the robot. A set of data can be considered as ground truth only if their precision is comparable (or better) with the precision obtainable by extracting - by processing with state-of-the-art algorithms - the same information from robot-collected data. GT is used as a reference for the application of the performance evaluation metrics included by the Benchmark Problems.

In practice, there are two types of GT that can be associated to Rawseeds' dataset:

- ground truth for mapping, called **mappingGT**;
- ground truth for pose, called **poseGT**.

The mappingGT is composed of executive drawings of the locations explored by the robot. By its nature, each piece of mappingGT is associated to a location, but not to a specific dataset. The mappingGT is used to evaluate the performance of a Benchmark Solution generating a map of the environment: a comparison is made between the reconstructed map and the mappingGT. This comparison is based on the concept of *feature*: a definition of this term in the context of Rawseeds' Benchmark Problems will be given in the following of this document.

The poseGT, on the contrary, is associated to a specific dataset. It is composed of portions of the trajectory of the robot, recorded while it was acquiring the data for the dataset. Such trajectory portions have been reconstructed with the use of systems that are independent from the sensors on board of the robot.

4. Features are corners

Some of the performance metrics associated to a BP require, for their computation, a comparison between a map of the environment generated by the algorithm used to solve the BP and a reference. This reference is the *mappingGT* described previously in this document; in these cases, then, a comparison between two different - and possibly heterogeneous - maps is required.

For practical reasons, the above comparison is done for a limited set of suitable *features* of the environment. The features that have been selected for this purpose are the **corners**, i.e., the intersections between two non-parallel segments in the birds-eye view of the environment. Therefore, all performance metrics associated to a BP that are based on the use of the mappingGT are computed by comparing specific corners in the mappingGT with the corresponding corners in the map of the environment reconstructed by the algorithm used to solve the BP. To facilitate the use of such metrics, Rawseeds' Benchmark Problems include a list of the positions of all the relevant corners for the specific location that the BP is associated to.

5. Use of corner-based performance metrics

Any algorithm (applied to the solution of a BP) that produces a reconstructed map of the environment will define such map using the type of features and geometric primitives that are more appropriate for its inner functioning; usually these will not be corners. However, if the creators of such an algorithm desire to present it as a Benchmark Solution and are required (from the specifications of the BP) to compute one of the mappingGT-based performance measurements, they need to extract corners from the map that their own algorithm produced. Precisely, they need to extract from their own map a set of corner positions to be compared with those coming from the mappingGT.

The above process of “corner extraction” has to be done in accordance to the specific structure of the map generated by the algorithms: the choice of geometric primitives done by the developers, in particular, is a key factor. On one hand, we have the mappingGT, which will essentially be a point-based 2D map of the world, and its associated list of corner positions. On the other hand, we have the reconstructed map that, depending on the specific algorithm, might be based on 2D or 3D features, on segments or points, and so on. In the following, we propose some guidelines - necessarily limited to a general level - to perform the “corner extraction” operation.

The 2D-3D mismatch issue can be solved with a suitable projection: precisely, *all reconstructed features must be projected on the mappingGT floor plane* (please note that in RAWSEEDS' datasets, floors are mostly horizontal).

To overcome any mismatch between different geometric primitives, the developer of the algorithm to be presented as a BS must extract "reconstructed corners" from the map generated by her algorithm. A short list of suggested steps for handling a mismatching of geometric primitive follows (please notice that very simple software helpers can largely speed up this kind of work).

For 3D segment maps, like those produced by 3D segment-based SLAM approaches:

1. project the 3D extrema of the segments on the floor;
2. verify (human intervention) that the 2D segments (i.e., the reconstructed walls) are longer than a given threshold;
3. reconstruct the corners by intersection of adjacent segments.

For 3D point maps, like those produced by monocular SLAM approaches:

1. project the 3D points on the floor;
2. cluster (human intervention) the 2D points into 2D segments;
3. verify (human intervention) that the 2D segments (i.e., the reconstructed walls) are reliable;
4. compute the support line of the segments;
5. reconstruct the corners by intersection of adjacent segments;

For grid-based 2D maps, like those produced by most approaches based on sonars and/or laser scanners:

1. cluster (human intervention) the grid elements into segments,
2. verify (human intervention) that the 2D segments (i.e., the reconstructed walls) are reliable;
3. compute the support line of the segments, taking into account the PDF on the grid;
4. reconstruct the corners by intersection of adjacent segments.

6. Uploading new BSs

6.1 Introduction

A Benchmark Solution is not simply an algorithm that solves a Benchmark Problem instance. To be considered as a valid BS, such algorithm must be part of a more comprehensive set of elements. The presence of all the required elements is a necessary condition for a Benchmark Solution to be considered valid by Rawseeds, and therefore it is required when submitting a BS for publication on Rawseeds' website. Please note that submitted BSs must pass a review process (performed by Rawseeds' staff) before they are published.

Any new BS must be submitted using a specific upload form, described in the following. All information that is not directly input in the fields of the upload form has to be uploaded in the form of “packages”. A package is a collection of related files, collected into a single file (e.g., using *tar*, and possibly a compression algorithm such as *zip*). Please note that to upload a BS you need to log in: therefore, you have to be a registered user. This is done by requesting a user account for the Rawseeds Forum (if you haven't one yet); the account will be also usable to submit new BSs. To perform the submission of a new BS you have to open the webpage of the BP that the BS is dedicated to, and then use the login form that is present in that page.

6.2 Components of a BS submitted for publication

Below is a description of the components of a BS, as listed in the upload form. **Whenever a specific element of one of these components is mandatory for publication, this is clearly specified in the following list.** The presence of the other elements is not necessary for publication, although it is appreciated.

BS component #1: information that is directly input in the BS upload form.

- Title of the BS **(mandatory)**. Suggested choice for this is the name of the algorithm that the BS is based on.
- Short description of the above algorithm. This is not mandatory, but its presence is strongly recommended.
- Uploader name **(mandatory)** (generally the username is used here)**.**
- Name of the company or institution that the submitter belongs to (if any).
- Logo of the above company or institution.
- List of the results obtained from the application of the performance evaluation metrics to the output of the BS **(mandatory)**. The metrics are specified by the chosen BP. Usually, only some of them are mandatory (i.e., their value is required for the publication of the BS): application of the other metrics specified by the BP is recommended but not required.

The outcome of the application of a performance metric to the output of a BS is a collection of one or more *results*. Each result is a numerical value, the physical dimensions of which are specified by the metric. In the input form, the section dedicated to a specific performance measurement can be used to input all of these results, one per line. By default, only a single line is present, but additional lines can be added as needed. Each of the lines is composed of two parts: one for the numerical value of the result, and the other for a text comment associated to it. The latter is generally used to specify the name of the specific result within the set pertaining to the same metric (e.g., “mean value”), so that the reviewer is able to identify it correctly.

If the outcome of the application of a given performance metric to the output of the BS to be submitted cannot be described by a set of numerical results (e.g., if a graph is considered necessary by the uploader), only the comment part of the associated line must be filled, specifying the name of a file where the result of the application of the performance measurement is described. Such file, of course, will need to be included into the *solution package* described in the following.

BS component #2: solution package.

1. Description of the algorithm **(mandatory)**, generally in the form of one or more documents (e.g., suitable papers). Such description must:
   - be sufficiently detailed to be usable[1];
   - not have commercial purposes *only*[2].
2. Full output of the algorithm **(mandatory)** when applied to the data of the chosen BP instance.
3. A document describing the data formats used for the output of the algorithm **(mandatory)**. This description must be full, i.e., sufficiently detailed to allow the use of any part of such output for further processing.
4. If necessary, a document with comments on the application of the performance metrics defined by the BP instance to the output of the algorithm. Please note that the above application is performed by the author(s) of the BS, and the results obtained are considered valid on good faith.
5. All additional information (if any) specified by the description of the performance metrics applied to the BS **(mandatory)**.

BS component #3: license.

1. A file specifying the license under which the BS -if accepted- will be pubilshed **(mandatory)**. This file can include the full text of the license or a reference to an existing license available through the internet (e.g., one of the Creative Commons licenses). The license will be published on Rawseeds' website alongside the BS.

BS component #4: source code package.

1. Source code of the algorithm, in any programming or scripting language (e.g., Matlab code), along with any accompanying documentation, usage notes, tutorials and/or examples. The submission of BSs complete with source code is encouraged.

[1] The description of the algorithm must be complete enough to allow a reader to write a piece of software implementing it in basic form. Details about data representation, efficient implementation, management of special cases, and so on are appreciated but not mandatory.

[2] BSs coming from companies marketing robotic products are welcome; however, a BS is required to be a worthwhile scientific contribution in itself, rather than a technically-enhanced press release. For instance, a company could submit a BS based on one of their proprietary algorithms, without giving away the implementation details that could make the copying of their products possible; but at the same time the scientific usefulness to the robotic community of such a submission must be ensured.

BS component #5: virtual machine package.

1. A file or set of files describing a virtual machine (VM), including an operative system (*guest OS*) and an installed and working implementation of the algorithm[3]. The VM must be provided with all the documentation needed to use it and to run the embedded algorithm implementation.

If its author so prefers, the algorithm in the VM can be supplied in executable form only, and/or be functionally limited in such a way that it works only on the data files of the specific BP instance considered. All reasonably widespread virtual machine formats are accepted, but those for which a free and multi-platform *VM player* is available are preferred. Please note that VM files are usually very large: so hosting and publication arrangements must be taken between the author(s) and the Rawseeds project.

[3] Please be aware that licensing schemes for proprietary OSs could impose limitations on the distribution of virtual machines including such software as the guest OS. These limitations can be absent for non-proprietary OSs (e.g., Linux).