



Best Practice in Robotics (BRICS)

Grant Agreement Number: 231940

01.03.2009 - 28.02.2013

Instrument: Collaborative Project (IP)

Benchmarks for Mobile Manipulation and Robust Obstacle Avoidance and Navigation

Walter Nowak, Alexey Zakharov, Sebastian Blumenthal, Erwin
Prassler

Deliverable: D3.1

Lead contractor for this deliverable:	GPS Gesellschaft für Produktionssysteme GmbH
Due date of deliverable:	March 1, 2010
Actual submission date:	April 9, 2010
Dissemination level:	Public
Revision:	1.0

Executive Summary

In this deliverable we present first year's results from Work Package 3 (Best Practice Robot Algorithms) of the European research project BRICS. Therein we aim to develop a methodology for identifying and providing best practice algorithms in a range of robotics fields. The methodology shall found on objective performance comparison measures and finally result in a number of open source best practice libraries.

More specifically this deliverable covers the first half of Task 3.1 "Benchmarking and harmonisation of robot algorithms". The first goal of the task is the definition of the operational context for the considered robotic domains. The two selected domains of reactive obstacle avoidance for mobile robots and sample-based motion planning for mobile manipulators will be outlined in this report. The main part of Task 3.1 deals with the definition of objective performance measures, test conditions and procedures for the involved algorithms, for which we present an overview. Finally these investigations shall lead to the implementation of useful benchmarks during the second year of the project.

The definition of benchmarks is closely related to a thorough understanding of the problems and algorithms in the respective domain. Therefore we analyzed motion planning algorithms concerning common sub-structures and investigated how their interfaces may be harmonized. This idea has been partly published before in [30] and [31] and is summarized in this document with respect to the relation to benchmarking activities.

The work presented here forms the basis for Task 3.2 "Framework for identification and refactoring of best practice algorithms in robotics", which has started with refactoring existing motion planning algorithms towards a component-based software library. A prototype for such a framework has been developed, the framework itself and its implementation will be the topic of the next deliverable D3.2.

Contents

Executive Summary	ii
1 Introduction	1
2 Overview on Benchmarking	2
2.1 Definition of Benchmarks	2
2.2 Classification of Benchmarks	2
2.3 Benchmarks in Robotics	4
2.4 Benchmarking activities	4
3 Metrics and Experimental Setup	8
3.1 Metrics	8
3.2 Experimental Setup	9
3.3 Datasets	11
4 Mobile Manipulation and Obstacle Avoidance	13
4.1 Mobile Manipulation	13
4.2 Robust Navigation and Obstacle Avoidance	14
4.3 Existing Benchmarks and Activities	15
4.4 Components of Motion Planning	17
5 Conclusion	24
Bibliography	25

1 Introduction

A lot of efforts and activities have been initiated in the last years with the goal to improve the availability and application of benchmarks in robotics. Although public awareness has raised significantly, still several authors come to the conclusion that many aspects of robotics are not yet covered adequately and a substantial change in the attitude of researchers is required. In this document we investigate the state of benchmarking in the domain of robot motion planning.

The European research project BRICS (Best Practice in Robotics) aims at identifying and applying best practice in a wide range of robotics areas, ranging from algorithms over hardware interfaces to control architectures. The answer to the question of what is "best" is thereof of central interest. The investigation of what is used in current systems or counting the number of citations of papers is important, but can give only some indications. Those aspects may be influenced by many undesired factors and do not provide reliable performance measures.

We believe that benchmarking should be the methodology to give objective comparisons and be one major tool to validate what is best practice. Therefore in Work Package 3 in BRICS we look systematically into performance comparison of algorithms. Our major interest is not to define new benchmarks or measures. We rather seek to exercise benchmarks in order to collect performance comparisons and evaluate existing software. This idea has been partly published before in [30] and [31] and is the focus in this document.

After a general introduction of benchmarking for robotics in Chapter 2 we overview existing measures and experimental setups in Chapter 3. In Chapter 4 we analyze the concrete application to the domain of motion planning with the two focuses of mobile manipulation and reactive obstacle avoidance.

2 Overview on Benchmarking

In this chapter we take a general look at benchmarks and present the most important contributions for the domain of robotics. In any scientific discipline, benchmark databases and comparison methodologies are important instruments, and the way of conducting experiments may be part of the self-conception of the discipline itself.

In domains such as processors and graphics cards, benchmarks are not only very common, but every new product is evaluated and estimated towards a whole set of benchmarks. Several organizations exist that explicitly deal with defining and applying benchmarks, such as SPEC, BAPCo, EEMBC, and many more. A major issue is the problem that companies may want to influence benchmarks in order to gain an advantage for their own products. Several incidents have been reported as well where products such as processors have been explicitly prepared to detect standard benchmarks and in that case execute a specially optimized behavior, pretending a high performance that is generally not available.

Benchmarks can play an important role in various aspects. Some see it related to roadmaps [17] where benchmarks may guide developments and make progress in the right directions measurable. Another important aspect is the relation to standards. Standards can provide the framework for benchmarks, and benchmarks on the other hand may be used to foster standards in a field.

2.1 Definition of Benchmarks

Several definitions of benchmarks have been proposed. In general it is related to comparing systems and deriving a relative performance measure.

- Wikipedia: "Benchmark (computing), the result of running a computer program, or a set of programs, in order to assess the relative performance of an object by running a number of standard tests and trials against it"
- [18] defines benchmarks as "precisely defined, standardized tasks" with three main aspects: task, standard, and precise definition.
- [14] adds numerical evaluation of results (performance metrics) as key element. The main aspects are repeatability, independency, and unambiguity.
- [15] defines a benchmark as "a specified set of rules (agreed by the community) for assessing specific features and measures of a system", including definitions of four things: the considered system in its environment, the benchmark context, the benchmark measures, and everything needed for experimentation.
- At the MLP workshop on Regional Benchmarking in November 2005 [17] benchmarks were defined as "process of identifying the highest standards of excellence for products, services and processes and then making the improvements necessary to reach those standards". Four steps were identified, that are (i) gathering information, (ii) comparing and understanding, (iii) analysing the information and (iv) implementation.

As we do not see serious contradictions between all the above definitions that do mainly focus on different aspects, for the context of our work all of them may be applied.

2.2 Classification of Benchmarks

Benchmarks themselves can be classified according to various criteria. An essential one is the complexity of the benchmarked system. This may be a single functionality, or an aggregation of functionalities up to a complete system. A similar differentiation between component and system level is mentioned in [14]. In [17] the interaction with humans is pointed out as one even higher level. In [39] benchmarking is divided into top-down or bottom-up. In top-down,

2.2. CLASSIFICATION OF BENCHMARKS

the robot is investigated primarily as a whole system. In bottom-up, each single subtask is individually evaluated with the aim to assess the overall performance as some kind of synergy of those subtasks. A related categorization is technology versus application evaluation [24].

Single functionalities or analytical approaches usually treat more technical aspects. Here specific problems arise from the definitions of those functionalities, which require common interfaces specifications that are sometimes not available [11].

On the other hand, system resp. integrated benchmarks test a robot as black box on more application-oriented aspects. This may allow to estimate whether a whole system can achieve a task. The disadvantages are usually higher efforts required to execute benchmarks and much more detailed test protocols needed to ensure reproducibility. It is remarked in [17] that system level benchmark for robots only exist in form of competitions so far.

Other dimensions for classification include:

- Precision of the benchmark [17]. A benchmark may be a clearly defined synthetic one, of comparative nature such as contests, or the subjective evaluation of a robot.
- Theory vs simulation vs real world benchmarks. Theoretical results for most robotics problems are often not available or can only give bounds that are of limited practical use. The distinction between simulation and real-world is an essential aspect for robotics, which may refer to the system itself, parts of it, input data, or the provision of the robot's environment.
- Real data vs artificially generated data. Provision of test data is in particular a problem when the system under investigation uses some active control and influences its environment. An example is active vision, where the perceived sensor data depends on the vision algorithm itself.
- Functional vs analytical [14]. While functional benchmark evaluate the performance of fulfilling specified problems and are the standard in robotics now, analytical benchmarks try to assess performances by mathematical means.

Various properties of benchmarks are summarized in [23], those are representativeness, repeatability, portability, non-intrusiveness, cost-effectiveness, and scalability.

The purpose of experiments is investigated in [22], which may be either to show that something works (proof of concept), to show that it works better than something else (horse race), get insight on behavior or limits, or a mixture of the above.

In communication and networking systems, Quality of Service (QoS) denotes the ability to obtain or guarantee certain quality attributes in a network. The term QoS is used as well in the context of component-based and service-oriented architectures, often in relation to non-functional requirements. Thus QoS may be seen as a criterion for benchmarks for ensuring a real application value that is not impeded by irrelevant optimizations of single properties.

A special kind of benchmarks are contests. Contests usually have the special property of some unpredictability, so that in spite of fixed rules and conditions, the exact test data during the contests shall not be known in advance. In contrast, other benchmarks aim explicitly at the exact definition of input values. [5] argued that the "RoboCup competitions do provide an objective performance measure", given the fact that some teams manage to win multiple times in a row.

A problem of contests is that they may produce some kind of standard solution which works best for the one contest, but does not offer universal usability [24]. Such solutions may hamper research towards alternatives which do not provide an immediate benefit in the contest. Similar issues may arise with other non-contest benchmarks as well. If standard performance measures exist, developers will usually have an inherent interest that their products perform

well, even if application in other contexts, where the performance is less observable, may suffer.

Another aspect of contests is that teams may refrain from making their developments public, in order to increase own chances to win. As an example how to counter this, some RoboCup leagues motivated or even required to make the software used public.

2.3 Benchmarks in Robotics

The situation in robotics concerning benchmarks and systematic experimentation is rather different compared to the domains mentioned at the beginning of this chapter. In robotics, a standard methodology for benchmarking is mainly missing, which makes it difficult to compare different approaches, in particular in different scenarios and environments [11]. But the situation seems to be changing, many initiatives are being set up and researchers are paying more and more attention to benchmarking [7]. In addition it should be noted that there is no coherent picture within the whole robotics domain. For industrial robots with highly specialized tasks or on the border of automotive industry and embedded systems there is often a much more evolved methodology of performance assessment available than for general purpose service robots.

The problems with tuning products towards benchmarks is by far not as escalated in robotics as elsewhere. It is more current practice that researchers present own developments or algorithms in a context that just fits well, partly because no alternatives exist.

Several aspects of the robotics domain in its current state make benchmarking difficult. Implementations of algorithms or robot functionalities are often difficult to compare due to different hardware or software setups [8]. Results that are presented in studies or scientific papers can be difficult to reproduce. In most papers tuning of parameters are not presented detailed enough [3] or parameters are not made explicit at all, with the effect that the exact setups of the robot systems can only be guessed. Missing comparisons between algorithms and the notion that a problem is "solved" once published, even when working in only one specific setup that potentially cannot be reproduced, becomes a hurdle for robotics research in general.

In our view one major characteristic is often neglected, that is the relationship between benchmarks and the robot development process. In many development processes such as V-Model or agile methods testing is a key element. As we are not aware of any generally accepted development process for robot applications, also the definition of tests as well as the interrelation between design and implementation of functionalities and their validation suffers from no general standards.

One possible reason for a weak role of experiments in the development of robots is the problem that extensive tests are very time consuming, in particular when real hardware is involved [3]. Solutions involving simulation and hardware-in-the-loop are either limited or not easily available. Another reason comes from the fact that the design of experiments and application of statistical methods has not such a long tradition in robotics as in other, natural sciences.

Finally many areas of robotics are still rather new and undergoing tremendous changes. As an example, a study in 2004 came to the conclusion that the field of human robot interaction is not mature enough yet for defining benchmarks [10].

2.4 Benchmarking activities

In the following sections we provide a short overview on existing activities related to benchmarks in robotics, comprising events at scientific conference, public data sets, benchmarking frameworks, research efforts and competitions.

2.4. BENCHMARKING ACTIVITIES

2.4.1 Workshops and conference tracks

- Workshops on benchmarks at international events (IROS'06, IROS'07, IROS'08, IROS'09, RSS'08, RSS'09)
- PerMIS (Performance Metrics for Intelligent Systems) is an annual workshop hosted by the National Institute of Standards and Technology (NIST) in USA.
http://www.isd.mel.nist.gov/PerMIS_2009/
- Workshops at EURON (European Robotics Research Network) Annual Meetings: 2006, 2007, 2008, 2009
<http://www.euron.org/activities/agms/index>
- EURON GEM SIG meetings - Good Experimental Methodologies and Benchmarking, Special Interest Group
<http://www.heeronrobots.com/EuronGEMSig/GEMSIGEvents.html>
- CogGEMBench'08 Workshop on Good Experimental Methodology and Benchmarks in Cognitive Robotics.
<http://www.heeronrobots.com/EuronGEMSig/GEMSIGCOGGEMBenchProgram.html>
- Workshop on Benchmarking at IEEE-RAS 7th International Conference on Humanoid Robots.
<http://wwwiaim.ira.uka.de/users/asfour/Humanoids2007.htm>

2.4.2 Public data sets

- Radish: Robotic Data Set Repository for SLAM
<http://radish.sourceforge.net>
- Rawseeds originated from an European project and defined data sets as well as benchmarking methodologies for SLAM
<http://rawseeds.elet.polimi.it/home>
- UCI Machine Learning Repository is a collection of databases and data generators for the empirical analysis of machine learning algorithms
<http://mlearn.ics.uci.edu/MLRepository.html>
- PASCAL is a set of ground-truth data and tools for visual object recognition
<http://www.pascal-network.org/challenges/VOC/databases.html>
- Motion planning benchmarks developed by Parasol Lab at A&M University of Texas is a list of well known benchmark problems, such as alpha puzzle, box folding problem etc
<http://parasol-www.cs.tamu.edu/groups/amatogroup/benchmarks/mp/>.
- MOVIE project includes tools for benchmarking of motion planning in virtual environments
<http://www.give.nl/movie/> (URL does not work anymore)
- FERET and XM2VTS are datasets for face recognition
http://www.itl.nist.gov/iad/humanid/feret/feret_master.html
<http://www.ee.surrey.ac.uk/CVSSP/xm2vtsdb/>
- COIL is a dataset for object recognition
<http://www.cs.columbia.edu/CAVE/software/softlib/coil-100.php>

2.4.3 Performance metrics and benchmarking frameworks

- Automatic Control Telelab is a general framework for designing remote experiments. It supports real-time configuration and observation of experiments, as well as playback access to acquired data
<http://act.dii.unisi.it/home.php>.

- JaViSS (Java-based Visual Servo Simulator) provides a number of tools for modeling and benchmarking visual servoing applications.
<http://www.robot.uji.es/research/projects/javiss>
- NIST (National Institute of Standards and Technology) promotes development of measurement and interoperability standards for manufacturing robotics, automation equipment and underlying industrial control systems. Various information on performance metrics for mobile robots in realistic environments is available at
<http://www.nist.gov/mel/isd/metrology.cfm>
- Automatic Evaluation Framework for obstacle avoidance algorithms was developed at University of Zaragoza
<http://webdiis.unizar.es/~jminguez/publications.html>

2.4.4 Research coordination efforts

- The most comprehensive survey of "Current Efforts in Comparative Robotics Research" is available at
<http://www.robot.uji.es/EURON/en/index.htm>.
- RoSta (Robot Standards and Reference Architectures) was a European research project. One of the goals of the project was to create a survey of existing benchmarks for service robotics.
<http://www.robot-standards.eu/>
- EURON performs various activities on benchmarking. Information on the current initiatives is available at
<http://www.euron.org/activities/benchmarks/index.html>.
- ALFUS (Autonomy Levels For Unmanned Systems) is a collection of descriptions of standards, metrics, specifications and tools for benchmarking the autonomy of unmanned systems
http://www.isd.mel.nist.gov/projects/autonomy_levels/

2.4.5 Competitions and challenges

- A comprehensive list of robot competitions is maintained at
<http://robots.net/rcfaq.html>.
- RoboCup is an international robotics competition consisting of several leagues. By now it includes soccer with mobile and humanoid robots, simulation leagues, rescue robotics, service robotics in Robocup@Home and multiple junior leagues for students
<http://www.robocup.org/>
- DARPA Grand Challenge and Urban Challenge is a series of competitions for autonomous vehicles in the US that gained significant public awareness also due to its high prizes in the order of a million dollars <http://www.darpa.mil/grandchallenge/index.asp>.
- ELROB (European Land-Robot Trial) is a European competition for autonomous vehicles
<http://www.elrob.org/>
- Eurobot is an international amateur mobile robotics competition. <http://www.eurobot.org/>
- FIRA (Federation of International Robot-soccer Association) is similar to RoboCup, but focuses on soccer games only.
<http://www.fira.net/>
- Cleaning robot contest took place jointly with IROS 2002 in Lausanne, Switzerland.
<http://robotika.cz/competitions/cleaning2002/en>, <http://www.robot.uji.es/EURON/en/cleaning.htm>

2.4. BENCHMARKING ACTIVITIES

- NIST Face Recognition Vendor Tests are independent benchmarks of available face recognition technologies

<http://www.frvt.org>.

3 Metrics and Experimental Setup

In the following chapter we give an overview on how benchmarks may be defined and executed. Central to benchmarks is the idea of measuring something, therefore metrics play a crucial role, which we are going to investigate first. In addition, benchmarks are characterized by a profound and thoroughly specified test setup and execution policy. Thus we will show how such policies have been implemented in previous works.

3.1 Metrics

Metrics here are meant in the broad sense of measuring something, not restricted to the mathematical functions acting on topological spaces. Metrics in general can be classified according to various properties, some of them are

- qualitative vs quantitative
- absolute vs relative
- directly measured vs computed or composed

In the context of benchmarks, most metrics fall into one of three categories:

- cost (where lower is better)
- utility
- reliability

In the industrial context, many metrics are well-known and applied often related to the production process. Examples include work-cycle time, throughput, energy consumption, mean time between failures, etc.

Many metrics in the domain of automotive and embedded systems are mentioned in [15], comprising

- execution time: worst case execution time (WCET), worst observed execution time (WOET), average observed execution time (AOET), component initialization duration
- memory consumption
- dependability

For several robotics fields it is not clear how useful metrics may be defined. One prominent example is mapping, where the quality of generated maps shall be compared. A difficulty comes from the fact that the map is not created for its own purpose, but it is being used by a robot system. Therefore small deviations in a map may result in dramatic failures if the topological structure of the environment is not captured well.

As the domain of robotics is developing, also new or refined metrics are being implemented. That phenomenon can be seen most clearly in robotic contests such as RoboCup or DARPA Grand Challenge, where the rules and thus implemented metrics are subject to a continuous change in order to accommodate for progress in the field.

An issue which can occur for robotic systems is the problem of plan deduction [4]. While the overall behavior of the robot system may be observed, the evaluation of the internally generated plans would be of higher interest, without unavoidable influences such as by low-level controllers or impacts from the environment. That may make some metrics such as time bad measures.

3.1.1 Specific metrics for Mobile Manipulation and Obstacle Avoidance

For the concrete fields of mobile manipulation and obstacle avoidance, a variety of designated metrics have been applied. For an overview we recommend [6] and [11]. But nearly every

3.2. EXPERIMENTAL SETUP

research paper in these domains implements some kind of metrics to evaluate the results. A list of relevant metrics is given below.

- Cost metrics:
 - computation time, execution time
 - reaction time, cycle length of update step
 - time to collision, distance to collision
 - number of parameters to tune
 - variance in results
- Utility metrics:
 - path length, distance traveled
 - clearance, i.e. minimum distance to obstacles
 - smoothness of trajectory
 - precision at target
 - distance/error from given trajectory, orientation error
 - scaling to 3D, in contrast to 2D
 - scaling concerning degrees of freedom
- Reliability metrics:
 - mission success
 - number of collisions
 - robustness in narrow spaces, number of narrow passages traversed, or narrowness of passages
 - dependency on parameters (small changes in parameters should not have big influences on result)
 - avoiding traps or cyclic behavior
 - occurrence of oscillations [32]
 - control loop stability

The classification of a metric might also depend on its context and concrete definition. Metrics are often combined in order to assess higher-level properties such as optimality (distance of path and clearance), safety or robustness [21]. Several mathematical equations for measuring trajectories are presented in [21].

More concrete contributions include [34], where Morales explicitly looks at dedicated metrics for sample-based motion planners. [35] defines dedicated performance metrics for navigation of mobile robots. [16] highlights measures in context of collision detection, including degree of separability (how much overlap between objects) and frame coherence (reusability of previous computations) to avoid unnecessary new complex collision checks. [21] states that non-kinematic measures, such as time optimality, depend on the dynamics of the robot and therefore cannot be measured in the own framework.

3.2 Experimental Setup

3.2.1 General

Very sophisticated theories exist concerning the Design of Experiments (cf. Wikipedia, [33]). The valid setup and analysis of experiments is a central element of any scientific discipline, including

- Specification of test setup
- Definition of test criteria and measures
- Proper execution and evaluation of tests

An important procedure for natural sciences is to setup a hypothesis and then create an experiment in order to proof or contradict that hypothesis. This can be done in particular to build and confirm a model of a complex or unknown system.

The identification of dependent and independent variables is an essential part of any test policy. These variables need to be controlled or varied in a way to ensure a sensible coverage and distribution. Statistical methods are relevant tools for the setup and analysis. They can provide crucial hints concerning the number of experiments, depending on the number of variables and variations.

An important part of experiments is the documentation of the process and all involved entities. This is one of the major elements in order to enable reproducibility of the tests. It is also one of the weaknesses for current robotics. The outcome of experiments and results of metrics depend strongly on implementation details, which are very difficult to describe in a comprehensive way, if no underlying standards are available. This also makes some metrics doubtful, such as execution times. Even if computer or CPU used are commonly mentioned in research papers, it can only provide a very rough indication.

[3] and [22] describe different ways how to compare:

- 1 Use same code.
- 2 Reimplement code by explanations in papers.
- 3 Compare the results with those listed in previous publications:
 - Try to explain why sth. performs better
 - Address all occurred anomalies (be honest and reveal new issues).
 - Repeatability: verify the experiment independently.
 - Number of trials needed: avoid effects that happen by chance.
 - Explain results.

Using the same code, or respectively using the same hardware, is one of the major concerns in robotics. One possible conclusion is given in [16], which describes the motivation for developing a new motion planning library as having one single system available where components can be exchanged and varied in an objective manner. A similar approach has been undertaken by us in the BRICS project [30] where an open source library of refactored motion planning algorithms is developed. The crucial element here is that those activities only can reach their desired impact if they do not remain isolated and take enough into account the exchangeability with other commonly used software frameworks.

Ongoing initiatives to harmonize interfaces or create common functional libraries are therefore important steps to improve the current situation. An example how this works out is given in [12], where multiple experiments are conducted using the OOPSMP and OMPL libraries. Those experiments can be reproduced by loading configuration files which the authors provide for download on the web.

A very related aspect is the definition of robots and environments in common file formats, where some versions such as COLLADA or XML variants seem to be emerging, but no widely standard yet exists.

The distinction between real hardware/robots and experiments in simulation plays a crucial role in robotics, as in most other similar domains. In simulation the setup of experiments can be much better controlled and may allow repeatability [11]. On the other hand, only tests with real hardware can measure the performance on a system level towards application value. Simulation has to abstract in various ways, introducing the so-called "reality gap". Several authors have tried to analyze the relationship between simulated and real system. Simulation in the design phase of a robot system also allows to explore and evaluate alternatives.

Existing simulation platforms include:

- Player/Gazebo
- USARSim
- Microsoft Robotics Studio

3.3. DATASETS

- Blender
- Webots, and others.

Another important element concerning the setup of experiments is the availability of environment information to the robot. An assumption that the robot has an a priori map relieves many problems. On the other hand there may be big variations of sensors types with very different noise properties. Several benchmark datasets therefore propose to provide multi-sensory data in order to enable some kind of flexibility of testable robot systems. A theoretical investigation has been done in [36], where robots are compared concerning their "information spaces", specifically applied to the field of localization.

3.2.2 Specific experimental setup for Mobile Manipulation and Obstacle Avoidance

Several testing frameworks have been developed particularly for the domains of robot motion planning. These include MoVeMa [11], Framework for Obstacle Avoidance [21] and SAM-PLA [16]. A major point which is mostly neglected in these frameworks is the integration of existing motion planning software libraries. Those libraries include MSL[27], MPK_{Kit}[25], MPK_{Kernel}[25], OOPSMP[38], OpenRave[13] and OMPL[42], see [12] and [30] for short descriptions and comparison. As much of current research is being related or implemented in one of those libraries, we believe they may be the means to providing common interfaces and exactly specifying implementation details as needed for benchmarking robot functionalities.

In the context of pole climbing robots, [43] describes the setup of benchmarks by defining three related entities: working environment, robot mission (split up into tasks) and robot itself. For manipulation and navigation tasks, the kind and representation of the environment plays a major role. Several metrics have been proposed to assess a quantitative evaluation and to capture the essential properties which mostly influence motion planning. [6] lists density, complexity, clutteriness, confinement, structure, and dynamism. In [21] mathematical equations are provided for density, clearness, confinement, uniformity, cluttering and structure. [21] also includes a scenario generator to create scenarios with given characteristics (in simulation). [1] defines goodness and expansiveness of environments.

Commonly used artificial environments include the alpha puzzle, narrow passages or holes, and variants of clutters. A list of such environments can be found at <http://parasol.tamu.edu/dsmft/benchmarks/mp/>. The objects in the environment, including robots, may be modeled for example in different resolutions, or with different number of triangles. The effects on motion planning algorithms have been analyzed in [40].

3.3 Datasets

Several benchmarking datasets have been developed in the robotics domains of vision and mapping, including Radish, Rawseeds, Victoria Park Set, etc. In comparison to other domains, we did not find much for mobile manipulation and obstacle avoidance. Datasets might be generated by the software frameworks mentioned above, but none of them seems to have generated such a dataset that would be of relevant significance so far.

In general a policy must be defined on how the available data is used. To give an example, for tests in the context of learning algorithms it is common practice to split the available test data into training, test and validation sets. A more complex example are face-recognition benchmarks. Here often two (or more) pictures of one person are made, where one is used for learning and the other for recognition.

A special problem for datasets in SLAM and mapping is to provide ground truth information. The measurement of the "true" location of the data gathering robot might be affected as well by sensor noise. Very expensive measuring tools might be used, but in robotics there is a trend

to build on cheaper solutions using sophisticated algorithms, for example by combining complementary sensor systems via sensor fusion.

4 Mobile Manipulation and Obstacle Avoidance

In this chapter we investigate benchmarks for robot motion planning. More specifically, we focus mainly on two tasks:

- Path planning for mobile manipulators assuming a static environment
- Reactive obstacle avoidance for a moving robot

The choice of those two tasks has multiple reasons. First they represent central and mature research fields in robotics that have been under thorough investigation for many years. Thus various different approaches have been developed, with more or less standard solutions available in theory and in software. Many properties and problems of those tasks are well understood by now, including the separation into single functionalities. Still research continues very actively, as the economically efficient implementation can be a major hurdle for real robot systems. In addition, it is remarkable that as [12] points out no good set of benchmark problems for motion planning exists.

One more reason for choosing those domains is that their level of complexity appears to fit well for benchmarking. The tasks are in general too complex to find computationally efficient theoretical solutions. It is not evident how to analyse the performance of algorithms, in particular when the algorithm as a whole shows many complex interrelations with its single components, which may be very hard to determine. On the other hand, it is harder to find or define benchmarks for even more complex tasks and applications, in particular when involving manipulation capabilities.

In the following chapters we first give a more precise definition of the domains under investigation. We summarize the activities and research initiatives that are mostly relevant for benchmarking those domains. Finally we take a close look at the subcomponents involved and point out what their role is in benchmarking robot motion planning.

4.1 Mobile Manipulation

The term mobile manipulation refers to algorithms for manipulators, e.g. robot arms, attached on a mobile base; the whole system will be referred to as robot in the following. The specific challenges here come from two aspects mainly. First, there can be a high number of degrees of freedom (DOF) that must be controlled. In addition, those may cover a large space, in particular the dimensions related to the mobile base. Second, the immediate environment of the robot is not known a priori and must be recognized via sensors, introducing uncertainties and possibly requiring a reactive behavior of the overall system.

More aspects in this domain comprise:

- Planning a path assuming static environment
- Updating a path based on new information about environment
- Execution of the path, or trajectory following
- Grasping of objects with end-effectors
- Visual servoing, i.e. guidance and verification of robot's position based on visual perception
- Manual teaching of trajectories

In this chapter we concentrate on the first three items. A more thorough analysis of the involved functionalities is given in chapter 4.4. Grasping, visual servoing and obstacle avoidance for manipulators is only shortly mentioned. For a comprehensive overview on the topic of motion planning we recommend [28].

Most planning algorithms work in the configuration space (C-space) of the robot. Each dimension of the C-space corresponds to one degree of freedom of the robot. While this approach enables efficient planning involving all joints of the robot, the representation of objects from the environment in the C-space may be very complex and prohibitively costly to compute.

A class of probabilistic, sample-based planners have emerged that can cope well with a high dimensional C-space. Most prominent are variants of Probabilistic Roadmaps (PRM) and Rapidly Exploring Random Trees (RRT).

The problem of finding paths has been for a long time analyzed also from a theoretical point of view [16], where some theoretical bounds could be determined. The general path planning problem is exponential in the number of DOFs, which is often experienced in practice when suffering from the "curse of dimensionality".

Planning algorithms can be classified according to several criteria:

- Static path planning in contrast to dynamic planning/updating and motion execution
- Completeness, resolution completeness, and probabilistic completeness: Does the algorithm always find a valid path, if such exists? Does it find with probability one, just need to run long enough? Does it find if resolution for planning is deep enough?
- Exact vs. approximate: Is the free or occupied C-space modeled exactly, or just approximated?
- Type of algorithm: cell decomposition, potential field, and roadmap methods [26]
- Single shot vs multiple shot: Are paths planned in one step (or multiple independent ones), or are there for example preprocessing steps that can be reused, such as building a roadmap.
- Centralized/single robot or distributed/multiple robot planning.

The path planning problem as regarded here has a big advantage concerning benchmarking, compared to other robotics areas. It is rather decoupled from the robot hardware, the planning process does not depend directly on the actions of the robot. Therefore simulated as well as real datasets can be provided, and many test runs can be easily executed in a short period of time.

The planning problems are either taken from a real robot, for example in a work cell or an building. Or it is an artificial problem such as the alpha puzzle, that usually shall emphasize some specific difficulties such as narrow passages.

4.2 Robust Navigation and Obstacle Avoidance

Motion planning as regarded in the last section requires a static environment model in order to compute a trajectory towards a goal state. The planned trajectory is usually represented as a set of way-points. Given that, the obstacle avoidance task is to safely execute motion from the current position of the robot along the way-points, taking into account the dynamics of the environment.

Thus obstacle avoidance here is regarded as a reactive behavior to let the robot adapt its motion according to immediate sensor measurements of the environment. The type of sensor, its update rate, accuracy and noise level have a strong influence on the possible algorithms. For service robots, laser scanners from Sick or Hokuyo seem to be the dominant sensor modalities. Such laser scanners retrieve quite reliable distance measurements, with the Sick sensors originally targeted at safety-critical systems. In contrast, cheaper sensors such as ultrasonic or cameras can suffer from much higher noise and be more sensitive to the environment's condition. In such cases the obstacle avoidance algorithm may have more of a sensor data processing task.

4.3. EXISTING BENCHMARKS AND ACTIVITIES

Algorithm	Included in framework
VFH+	Player, ORCA
Nearness Diagram	Player, MRPT
Potential Field	MRPT
Elastic Band	Sunflower Library
Dynamic Window	ROS, Sunflower Library

Table 4.1: Non-commercial robotics frameworks with obstacle avoidance algorithms

A critical factor of obstacle avoidance is its reactive component, i.e. generating a reaction fast enough to prevent collisions of the robot. Therefore the algorithms usually are required to run with a cycle time in the order of milliseconds, in contrast to static path planning problems which may take orders of higher magnitude.

Related to navigation and the execution of motion are the problems of localization, mapping and SLAM (Simultaneous Localization and Mapping). We do not include those domains in this version of the report.

A number of obstacle avoidance algorithms has been developed in the past years. An overview of important ones, available in open source libraries, is given in Table 4.1 These algorithms can be divided into directional and velocity space approaches. Directional approaches such as VFH+, Nearness Diagram, Elastic Band and Potential Field calculate a steering direction for a robot to head in. Usually they operate in Cartesian or Configuration space. On the other hand, velocity approaches such as Dynamic Window use a Velocity Space representation. Velocity approaches tend to be faster and more accurate dynamic-wise, as they operate directly in a Velocity Space.

Most of the implementations in Table 4.1 consider navigation of a mobile base as a 2D problem. However navigating in an everyday environment robot often has to deal with situations where the 2D approach would fail. In most of the frameworks navigation of a mobile base is separated from the navigation of the mobile manipulator. Robots often either navigate towards the goal with the fixed manipulator or performing a grasp with a fixed mobile platform. None of these software libraries has a generalized method suitable for 3D navigation of both mobile base and its manipulator. One example of such an approach is Elastic Strip [9]. The framework provides a method for performing local adjustments to a plan in dynamic environments while respecting original goals of the plan for multi-DOFs robots. However we are not aware of generally available software implementations.

4.3 Existing Benchmarks and Activities

In 2007, it was stated in [17]: "There is a lack of benchmarks in Action Planning, Manipulation and Grasping, Teaching by Demonstration and Learning, and Safety". Nevertheless, a lot of work has been done in this direction in the domain of mobile manipulation. Initiatives and activities that try to systematically compare or test are presented in this section.

Various competitions exist, on national as well as international level, that include motion planning aspects in some form, with highly varying degrees of autonomy, mechanical design, etc. Some of the most important and biggest ones include:

- RoboCup Rescue
- RoboCup@Home
- ELROB
- Eurobot
- Mobile Manipulation Challenge at ICRA 2010, organized by Willow Garage

An explicit focus on mobile manipulation is done in the RoboCup@Home league. In this league several so-called tests around service robots are executed, including navigation, manipulation and human-robot interaction. One of those tests is Fetch-and-Carry, where an object in the environment must be found by the robot, grasped, and carried to another location. As the whole league adjusts its rules every year, also this particular test has undergone significant modifications. As an example, in the second year of RoboCup@Home at Atlanta 2007, one team could achieve the task with a mechanical device controlled in an open-loop manner without any sensory feedback. As this does certainly not scale to general applications of service robots, the rules have been adapted to require some kind of object localization and obstacle avoidance. In other tests teams receive additional points if the robot does not collide with the environment.

In all other competitions, as we are aware of, the mobile manipulation aspect is not regarded for its own. The robot is evaluated as one complete system, without specific measures for the manipulation part. In some contests, for example the yearly changing Eurobot, manipulation of objects is required, such as placing objects on top of each other. Here usually the manipulation part seems to be solved mainly "mechanically" without general purpose grippers or path planning algorithms. This is reasonable to allow participants to come up with suitable solutions, but in many cases is not easy to scale up for general purpose service robots.

Independent from contests, several researchers tried to enable systematic comparisons of motion planning algorithms, without limiting to evaluate only own developments. One of the earlier works [4] proposes some ideas for benchmarks, but which were not further pursued to our knowledge. [20] proposes a methodology for benchmarks with the application to pick and place tasks, which also had been discontinued.

During his PhD work, Geraerts [16] has developed a software framework called SAMPLE (System for Advanced Motion Planning Experiments) to conduct experiments in a systematically manner. The focus in its work is on probabilistic planners, in particular PRM, where some "Results are surprising and differ from claims of developers". In order to execute objective experiments, the author developed one system (SAMPLE) as a C++ class library under Windows.

Components that can be varied in that framework include environment (geometry of environment and robot, degrees of freedom), roadmap construction (sampling, neighbor selection, distance metric, local planner, termination criterion), queries, and optional optimization techniques. Geraerts observes huge differences in execution time of the probabilistic algorithms. Measured are performance (computation time) and path clearance. In addition the author looks at the variances of these measures, taken from 100 runs.

For obstacle avoidance there are only few examples of explicit benchmarks of algorithms, as a stand-alone component. In [21] it is stated: "At present, there is only one experimental comparison" for obstacle avoidance, referring to a paper from 1991. A dedicated software framework is "Automatic Evaluation Framework for obstacle avoidance algorithms" developed at University of Zaragoza [39]. The framework can automatically build and qualify random environments, simulate and evaluate a robot motion. Environments are described by parameters such as density, clearness, confinement, uniformity, etc. The simulated trajectory of a robot motion is qualified by safety, optimality and success.

The problem of both frameworks mentioned before, similar to several other initiatives, is not so much the software itself, but the limited reachout and adaption by the community. As, to our knowledge, the initiatives have not been continued thoroughly it remains somewhat questionable that current state-of-the-art algorithms will be included to enable the objectiveness that was aimed for by integration into one framework.

EURON's Special Interest Group on experimental methodology specifies a set of guidelines for obstacle avoidance benchmarks [7]. During experiments hypotheses such as type of the robot,

4.4. COMPONENTS OF MOTION PLANNING

type of the sensors, goal location, or nature of the scenario should be taken into account. Suggested performance criteria are:

- Mission success
- Path length
- Time taken
- Number of collisions
- Obstacle clearance
- Robustness in narrow spaces

Research papers which will be mentioned in more detail in the next section include [1] (comparing distance metrics and local planners, for PRMs), [40] (comparing collision detectors) and [30].

4.4 Components of Motion Planning

Different functionalities that contribute to the complex task of motion planning have been identified for example in [41] or [30] and will be presented in the following. Figure 4.1 shows an architectural overview how functional elements are typically connected within a robot system. It should be noted that there may be a multitude of possible variants.

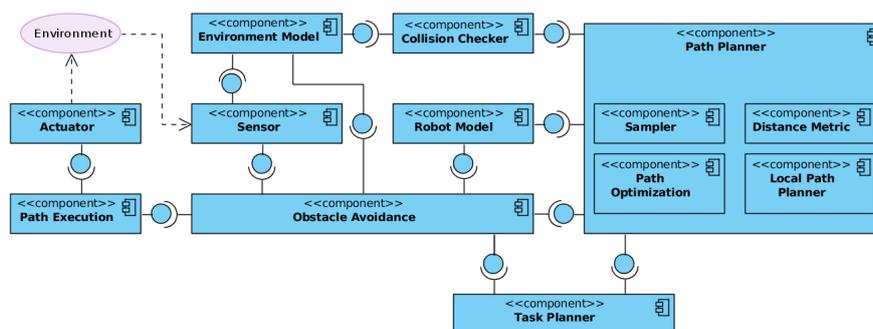


Figure 4.1: Components involved in robot motion planning

4.4.1 Collision Detection

Most of the time in probabilistic motion planning is typically spent for collision detection, which seems reasonable as the main goal is to find a collision-free path. Collision detection and object modeling is for itself a complex problem in the field of computational geometry. In the context of robotics, there are two facets: checking in an environment with known, modeled geometric objects, and checking for mostly unstructured sensor data. Here we will only regard the first case. The second case takes place when the environment cannot be modeled adequately, for example because it is only perceived via robot sensors, and will be described later in the context of obstacle avoidance.

In the first case, collision detection can be distinguished concerning the type of modeling of objects. These are either

- Polygon-based, either in a polygon soup, or in a structured way representing convex or non-convex objects.
- Solid objects, defined via Constructive Solid Geometry (CSG) or by their surfaces. Simple objects where intersections in between can be easily computed are combined into more complex ones.

In the context of path planning, mostly "external" software libraries have been used, which had their origins often in simulation or game engines, but were not developed or tuned towards path planning. An overview of the most important libraries that are available as open source is given in Table 4.2.

4.4. COMPONENTS OF MOTION PLANNING

Library	Based on	Comments
V-CLIP	polygons	feature-based algorithm on Voronoi regions, using Lin-Canny algorithm
SOLID	polygons	(axis-aligned bounding boxes (AABB))
PQP	polygons	hierarchical Rectangle Swept Spheres (RSS)
RAPID	polygons	hierarchical oriented bounding boxes (OBB)
V-COLLIDE	polygons	integration of RAPID and I-Collide, a library based on the Lin-Canny algorithm
Bullet	polygons	used in various products
ODE	solid objects	part of physics engine

Table 4.2: Non-commercial collision detection libraries

These libraries are mostly used in a black-box manner. That way they could be easier interchangeable. On the other hand it remains open whether any specific properties of the collision checking algorithms could be used for the path planning task. Some algorithms have been developed that try to avoid or delay collision checks, for example lazy PRM.

Two kinds of collision checks are commonly used. Either for a single point in C-space, which is the typical case for sampling-based planners. Or a whole path in C-space (or subspace, sweep volume) could be checked. Due to the required transformation between C-space and work space the latter option is not that common. Instead a given path is discretized, which is done via a local planner (see Section 4.4.4).

The collision check itself may be done via two methods. The first one is a boolean query that returns true or false, depending whether two objects intersect. Second one is to determine the minimal distance between any two objects. While theoretically both results could be computed from each other, in practice efficiency requires to choose one option, with the boolean version the preferred one for most path planning tasks.

Concerning benchmarking it is important to note that the choice of a collision checker, apart from some degenerate cases such as rounding errors, does typically only influence the computation time needed for path planning. The generated paths are not affected. Therefore the dominant metrics for comparing collision engines relate to the cost aspects, in particular computation time and memory consumption.

A comparison of collision detectors in the context of path planning (V-Clip, V-Collide, SOLID, RAPID, PQP) is given in [40]. Four artificial environments are used therein. These are a grid in two resolutions, a narrow passage problem and the alpha puzzle. The grid scenario uses up to around 8000 polygons. The authors come to the conclusion that the best choice depends strongly on the given problem, i.e. on the modeling of obstacles and the distribution of objects in the environment with the effect of varying ratios of checks with and without collision. The authors conclude that V-Clip performs best for a class of environments that is characterized by objects that are well-behaved, of moderate size and not too non-convex. But the performance derogates drastically in other environments. PQP is marked as having the worst computation times in general.

[31] compares PQP, YAOBI, V-Clip and Bullet, in two scenarios. [30] compares Bullet, YAOBI and PQP, coming to the conclusion that the performance depends strongly on the test case. For the given tests Bullet was in average clearly slowest and PQP best.

Two kinds of experiments in [16] vary the resolution of models. The outcome is that for the given scenarios the resolution has only a weak influence on the performance. From the avail-

4.4. COMPONENTS OF MOTION PLANNING

able results presented above it still remains somewhat open which collision detection library may be the optimal choice under which circumstances.

4.4.2 Distance Metrics

Distance metrics measure the distance of points in C-space. In order to avoid confusion with the broader sense of metrics in the rest of this report, we will use here the term "distance metric". As distance metrics determine which points in C-space can be connected or not, they play an important role in particular for sample-based path planning algorithm. Badly chosen metrics may make it difficult for the planner to connect regions in C-space, resp. generate relative movements of arm joints, thus enlarging the computation time. Still the used metric or its parameters is often not made explicit when comparing planning algorithms.

The distance metric does not have to be a metric in the mathematical sense. In particular the symmetry condition or triangle inequality may be violated for good reasons, leading to so-called pseudometrics.

A common choice for distance metrics are L_p norms, such as the Euclidean norm, which naturally extend to any dimension. Special consideration should be taken for dimensions resulting from rotational joints wrapping around e.g. in the interval $[0, 2\pi]$. Here other functions may better capture the specific topology of the C-space. For the case of rotations modelled by quaternions, dedicated distance metrics can be defined that allow for fast computation and yield convenient results.

Distance metrics can be defined either in C-space alone or in Workspace. Workspace metrics may take into account the displacement of the robot resp. its parts, or measure the volume swept by the robot when moving between two configurations. These Workspace approaches usually have the problem of an expensive computation. Approximations such as considering only vertices of the geometric robot model or fixed points on the robot joints may be feasible depending on the robot structure.

A common approach is to apply weighted metrics, where each dimension of the C-space is taken into account with a different weight. Intuitively joints near the base, which will produce a large motion of the whole robot should correspond to a higher weight, while joints near the end-effector, having a smaller effect, are weighted less. For mobile manipulation, the relation between the dimensions of the mobile base and arm are of particular interest. Weighted metrics may as well deal with the problem of different scaling of joints resp. dimensions, such as having intervals $[0, 1]$ or $[0, 2\pi]$. A problem arises with the determination of the weights, for which several solutions have been proposed.

The distance metric has multiple impacts on the path planning problem. First of all distance metrics contribute to the planning time by the time needed to compute the metric's value itself. For most simple metrics this may be negligible, but is a major point for variants defined in the workspace. Secondly distance metrics may reduce the number of calls to the local planner in order to determine whether two points in C-space can be connected by a collision-free path, [1] highlights this relationship explicitly. Finally distance metrics can have very complex influences on the generated paths themselves, due to points being tried to be connected or not.

[1] compares seven metrics in the context of PRM planning, in order to find out good combinations of distance metrics and local planners. By implementing dedicated test procedures, the authors come to the conclusion that the scaled Euclidean metric achieves the best performance in general. The more cluttered the environment is, the higher the weights for the position coordinates should be. Apart from a few cases, including measuring via bounding boxes in scenarios with highly-overlapping objects, there were no huge differences in the metrics' performances.

Algorithm	Type	Description
random	uniform	random number for each DOF
grid	uniform	take samples on grid with increasing resolution
cell-based	uniform	take samples within cells of decreasing size
Halton	uniform	based on Halton sequence
Hammersley	uniform	based on Hammersley sequence, with predefined finite length
Gaussian	non-uniform	add samples based on Gaussian distribution of distances to obstacles
obstacle-based	non-uniform	move colliding samples with increasing steps into free space
bridget-test	non-uniform	choose mainly samples that lie in between obstacles
medial axis	non-uniform	choose samples on medial axis, i.e. with equal minimum distance to two obstacles
nearest contact	non-uniform	move samples close to obstacles

Table 4.3: Samplers

In [31], three distance metrics (WeightedManhattan, WeightedEuclidean, WeightedLInfinity) are compared within a path planning problem. The result is that the metric has a noticeable effect on the planning time as well as the path lengths, while the reasons for different influences and answers to which metric to choose remain largely open.

4.4.3 Sampler

Samplers have the function of generating random points (samples) in the C-space. Sample-based probabilistic planners such as PRM and RRT rely heavily on the appropriate creation of samples, as those define the possible basic elements upon which the planners may act [19].

The distribution of generated samples is the major distinction between different samplers. These may be classified roughly into two groups:

- Uniform samplers spread samples uniformly in C-space, covering all areas in an equal manner.
- Non-uniform sampler may take into account for example obstacles or difficult regions.

Non-uniform samplers are typically more costly to compute, and may result in multiple collision queries for the generation of one single sample. In return, the generated samples are supposed to provide a beneficial distribution of the C-space for the path planning process by paying more attention at difficult local regions. That is why these approaches are also called importance sampling.

As the distribution of samples plays a major roles, various way how to measure it have been proposed. Those include:

- Dispersion is informally the largest ball inside C-space not containing any sample. In general a low dispersion is desired, meaning no big region remains uncovered.
- Discrepancy measures the irregularity of the distribution. Samples that are all equidistant or aligned on an axis may result in undesired behaviors and therefore can be avoided by assuring a high discrepancy value.

Rotational joints of robots introduce problems for sampling those dimensions in the C-space correctly. The simple solution of sampling each dimension separately does not give optimal

4.4. COMPONENTS OF MOTION PLANNING

Algorithm	Description
incremental	step from starts towards goal
binary	recurse via binary search
rotate-at-s	translate up to fraction s of total distance, then rotate, then translate remaining
A*-planner	use A* with some time limit

Table 4.4: Local planners

results concerning the C-space in total. Therefore better approaches involving quaternions and others methods have been proposed for those cases.

A list of commonly used samplers is given in Table 4.3. The randomness of samplers and the reproducibility of generated sequences may be affected by initiating the underlying random number generator with a specific seed. In particular Halton and Hammersley can produce deterministic sequences.

[30] compared uniform (random), Halton and Gaussian samplers in a path planning scenario concerning computation time and length of the generated path. Only small differences in performance were measured, without any clear indications. In particular the most simple random sampler does not show any significant disadvantage compared to the other ones.

[16] does a comparison of five uniform and six non-uniform samplers in six different environments concerning the computation time for path planning. The uniform samplers do only show negligible differences in performance. But the non-uniform samplers are in five of six environments nearly always slower than the Halton sample given for reference; sometimes by orders of magnitudes.

The situation changes drastically with the occurrence of narrow passages. In the one environment representing such a problem, all non-uniform samplers outperform the Halton sampler by far. This behavior should not be surprising as those samplers have been explicitly developed for use in such scenarios that typically give problems to path planners. This leads to the conclusion that the sampler depends strongly on the type of environment, and hybrid variants may provide the best performance in general. It should be noted that still some samplers may have clear benefits concerning other metrics, such as the medial axis sampler, which is very expensive to compute, but assures a high clearance of the resulting path.

As [19] points out sampling can be splitted up into the probability measure of the sampler, and the sampling source, consisting of random or deterministic numbers, for example in a unit cube that is then mapped to C-space. The source is shown to have only a minor effect on PRM planners, while the measure and distribution play a crucial role. The authors argue that the concepts of visibility and expansiveness in the C-space help to explain the good performance of PRM planners and may guide the definition of better sampling strategies.

A more general problem related to the task of sampling is the generation of random numbers on a computer. Classical algorithms include Mersenne twister or the linear congruential pseudo-random number generator, but we will not adress this issue here further.

4.4.4 Local Planner

Local path planners are used to check if two points in C-space can be connected in a collision-free way. They are different to global planners in the way that local planners are supposed to give quick answers, which might be that no connection was found. They usually work by discretizing the path between the two end points, as it is difficult to compute the required subspace in C-space from the robot's kinematics.

The connection between start and goal can be tried on a straight line, such as the first two methods in Table 4.4 do, or in more general ways. In addition to a boolean decision whether a connection exists or not, a path planning algorithm may use the information of how close the goal point can be approached. While rotate-at-s involves the choice of one parameter s , various properties and variants may be related to A* planners. The class of A* planners does already lead to the fuzzy boundary between local and global planners.

Local planners effect the overall planning task by their own runtime, adding up in particular by the involved collision checks. Therefore one major aim is to minimize the number of collision checks required. In addition the ability to find connections not lying only on a straight line can strongly influence the generated paths themselves.

[1] includes a number of comparisons of local planners in combination with different distance metrics, using some dedicated experimentation setup. As depending on the environment different planners manage to connect different sets of points, the authors also investigate the use of multiple local planners in one planning problem. The authors recommend the rotate-at- $\frac{1}{2}$ planner showing in general the best performance.

In [16] multiple variants are compared according to the computation time. The authors come to the conclusion that in contrast to previous claims, rotate-at-s is much slower than the other variants.

4.4.5 Other functionalities

One important post-processing step in particular for probabilistic planners is the optimization of the generated paths. This may include shortening the path, or increase some quality attribute such as straightness or clearance. Due to the sampling approach, probabilistic planners may generate very jerky motions that can only be used reasonably with such a post-processing step. In the context of performance evaluation this is in particular a relevant step, as it explicitly tries to decouple some criteria of the planning process. While the main planner can be optimized towards computation time or completeness, many other aspects can be mostly ignored first and be tackled directly in the optimization algorithm afterwards.

An important issue for most sample-based path planners is to determine which samples are closest to one point in C-space, a problem known as nearest neighbor search. For this dedicated task several algorithms as well as software frameworks exist, though we are not aware of a systematic comparison in the path planning context so far. Related to this problem is the actual choice of the next sample to process. As this is a rather intrinsic problem for each different type of planner, we do not go into more details here.

Related to the robot model, the computation of forward and inverse kinematics is a deeply studied mathematical field. Forward kinematics is often simple to compute by applying matrix transformations. In contrast, inverse kinematics is a complex problem where solutions may not exist or be redundant. Some kind of kinematic or dynamics constraints, such as a non-holonomic base, are typically encountered in mobile manipulation tasks. Special problems arise with closed kinematic chains.

Perception, vision, sensor data processing and environment modeling are all fields of their own. For robotics in particular, many different kinds of sensors are in use. Those are mostly

- distance-based, Time-of-Flight (TOF)
- vision
- proprioceptive

The use of specific sensors and algorithms can have an essential impact on the mobile manipulation task as well as form the basis for the obstacle avoidance behavior.

4.4. COMPONENTS OF MOTION PLANNING

Grasping and the design of sophisticated robot hands became a big research topic in the last years. In industrial contexts, usually the objects to grasp and their relative positions to the arm are exactly determined. For service robotics, there is the desire to get rid of those requirements and be able to grasp "arbitrary" objects. One special problem often occurring here is to plan with the robot gripper close to obstacles.

Finally, high-level functions involving some kind of task planning are usually more oriented towards real applications. Examples for those are pick-and-place, manipulate something, open a door, etc. This is the complexity level which is addressed by most competitions.

5 Conclusion

In this report we investigated the state of the art in benchmarking for motion planning, with a focus on mobile manipulation and obstacle avoidance. It was shown that several activities have taken place that aim at measuring the whole task as well as well-defined sub-components. Nevertheless the field seems to be rather at the beginning. Concerning an encompassing recommendation on how benchmarks may be applied in a more systematic way we agree and refer to the outcome of the RoSta project [37] and the resulting guidelines of the EURON GEM SIG [6].

We put a focus on the single functionalities that comprise the overall planning task. The thorough investigation of those elements seems to be crucial for a bottom-up approach on benchmarking. Looking from a system level view, one might ask if for example "Passing-through-a-door" is a good benchmark for mobile manipulation. While it appears very appealing on first sight, the task may be already too complex in order to ensure real insights on the given solutions and to enable reproducible experiments, with many functionalities involved and many properties very hard to specify. That said, the application within contests is possible, but objective benchmarking remains a problem.

There is a strong need for common software interfaces and hardware platforms, potentially such as ROS, BRICS, or various planning and motion libraries may offer. Otherwise it turned out to be nearly impossible for researchers and robot builders to describe all required details. Just one example that corroborates that statement are RoboCup leagues with fixed hardware (Aibo, Nao), where it appeared significantly easier to integrate and compare solutions.

There is also a need for common robot and environment descriptions. Candidates of file formats may be VMRL for environments and COLLADA for robot modeling. Actually several formats do already exist by now, but they are not commonly agreed or consequently used so far. Otherwise, as it was shown in many papers referenced here, the "same" algorithm can differ significantly with different implementation or parameterization of data structures or sub-components.

As an indication of current progress in this field, [2], [29] and [3] show that a stronger focus on elaborated methodologies for experiments is being adopted in subfields such as the SLAM domain. Taking the advances therein, the introduction of generally accepted benchmark data sets and methodologies also for motion planning tasks may be just a matter of time if similar efforts are undertaken.

Bibliography

- [1] Nancy M. Amato, O. Burchan, Bayazit Lucia, K. Dale Christopher, and Jones Daniel Vallejo. Choosing Good Distance Metrics and Local Planners for Probabilistic Roadmap Methods. Technical Report Technical Report 98-010, Department of Computer Science. Texas A&M University, May 1998.
- [2] F. Amigoni, S. Gasparini, and M. Gini. Good Experimental Methodologies for Robotic Mapping: A Proposal. In *Proc. of the IEEE International Conference on Robotics and Automation (ICRA 2007)*, 2007.
- [3] F. Amigoni, M. Reggiani, and V. Schiaffonati. An Insightful Comparison Between Experiments in Mobile Robotics and in Science. *Autonomous Robots*, pages 1–13, 2009.
- [4] J. Baltes. A Benchmark Suite for Mobile Robots. In *Proceedings IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2000)*, 2000.
- [5] Sven Behnke. Robot Competitions - Ideal Benchmarks for Robotics Research. In *In Proceedings of IROS-2006 Workshop on Benchmarks in Robotics Research*, 2006.
- [6] Fabio Bonsignorio, John Hallam, and Angel P. del Pobil. Special Interest Group Good Experimental Methodology - GEM Guidelines. Technical report, EURON SIG on Good Experimental Methodology, 2008.
- [7] Fabio P. Bonsignorio, John Hallam, and Angel P. del Pobil. Good Experimental Methodologies in Robotics: State of the Art and Perspectives. In *Proc. of the Workshop on Performance Evaluation and Benchmarking for Intelligent Robots and Systems, IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2007)*, 2007.
- [8] F.P. Bonsignorio, J. Hallam, and A.P. del Pobil. Defining the Requisites of a Replicable Robotics Experiment. In *Workshop on Good Experimental Methodology, at RSS*, 2009.
- [9] O Brock and O Khatib. Elastic Strips: A Framework for Motion Generation in Human Environments. *International Journal of Robotics Research*, 21, 2002.
- [10] J.L. Burke, R.R. Murphy, E. Rogers, V.J. Lumelsky, and J. Scholtz. Final Report for the DARPA/NSF Interdisciplinary Study on Human-robot Interaction. *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, 34:103–112, 2004.
- [11] D. Calisi, L. Iocchi, and D. Nardi. A Unified Benchmark Framework for Autonomous Mobile Robots and Vehicles Motion Algorithms (MoVeMA Benchmarks). In *Workshop on Experimental Methodology and Benchmarking in Robotics Research, at Robotics: Science and Systems (RSS)*, 2008.
- [12] Ioan A. Şucan and Lydia E. Kavraki. On the Implementation of Single-Query Sampling-Based Motion Planners. *IEEE International Conference on Robotics and Automation (ICRA)*, 2010.
- [13] Rosen Diankov and James Kuffner. OpenRAVE: A Planning Architecture for Autonomous Robotics. Technical report, Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, 2008.
- [14] Rüdiger Dillmann. Benchmarks for Robotics Research. Technical report, European Robotics Network, 2004.
- [15] Wolfgang Forster, Christof Kutschera, Dietmar Schreiner, and Karl M. Göschka. A Unified Benchmarking Process for Components in Automotive Embedded Systems Software. In *Proceedings of the 10th IEEE International Symposium on Object and Component-Oriented Real-Time Distributed Computing (ISORC'07)*, 2007.

-
- [16] Roland Geraerts. *Sampling-based Motion Planning: Analysis and Path Quality*. PhD thesis, Utrecht University, 2006.
- [17] Martin Hägele and Kai Pfeiffer. RoSta Deliverable D 4.1: Report on State of the Art on Benchmarks for Mobile Manipulation and Service Robots. Technical report, Fraunhofer IPA, 2007.
- [18] S. Hanks, M. Pollack, and P. Cohen. Benchmarks, Testbeds, Controlled Experimentation, and the Design of Agent Architectures. *AI Magazine*, 1993.
- [19] David Hsu, Jean-Claude Latombe, and Hanna Kurniawati. On the Probabilistic Foundations of Probabilistic Roadmap Planning. *International Journal of Robotics Research*, 25:627–643, 2006.
- [20] Ioannis Iossifidis, Gisbert Lawitzky, Stephan Knoop, and Raoul Zöllner. *Towards Benchmarking of Domestic Robotic Assistants*, volume 14/2004, pages 403–414. Springer Press, 2004.
- [21] J.L. Jiménez, I. Ranó, and J. Minguez. Advances in the Framework for Automatic Evaluation of Obstacle Avoidance Algorithms. In *Workshop of Benchmarking in Robotics, in the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2007.
- [22] D.S. Johnson. A Theoretician’s Guide to the Experimental Analysis of Algorithms. *American Mathematical Society*, 220(5-6):215–250, 2002.
- [23] Karama Kanoun and Lisa Spainhower. *Dependability Benchmarking for Computer Systems*. Wiley-IEEE Computer Society Press, 2008.
- [24] D. Kragic, V. Kyrki, P. Jensfelt, and F. Lingelbach. Drawing a Parallel: Benchmarking in Computer Vision and Robotics. In *Workshop on Benchmarking, at IEEE International Conference on Intelligent Robot and Systems (IROS’06)*, 2006.
- [25] J.-C. Latombe, F. Schwarzer, and Saha M. MPK - Motion Planning Kit. <http://robotics.stanford.edu/mitul/mpk/>.
- [26] Jean-Claude Latombe. *Robot Motion Planning*. Springer, Berlin, 2006.
- [27] S. Cheng LaValle, P. Kuffner, J. Lindemann, S. Manohar, A. Tovar, L. B. Yang, and Yershova A. MSL - Motion Strategy Library. <http://msl.cs.uiuc.edu/msl/>.
- [28] Steven M. LaValle. *Planning Algorithms*. Cambridge University Press, 2006.
- [29] M. Magnusson, A. Nüchter, C. Lörken, A.J. Lilienthal, and J. Hertzberg. Evaluation of 3D Registration Reliability and Speed-A Comparison of ICP and NDT. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), Kobe, Japan*, pages 3907–3912, 2009.
- [30] E. Milke, S. Christen, E. Prassler, and W. Nowak. Towards Harmonization and Refactoring of Mobile Manipulation Algorithms. In *International Conference on Advanced Robotics (ICAR 2009)*, 2009.
- [31] Edmund Milke and Stefan Christen. Best Practice in Mobile Manipulation - Motion Planning and Control. Master’s thesis, University of Applied Sciences Bonn-Rhein-Sieg, Department of Computer Science, Master of Science in Autonomous Systems, 2009.
- [32] Javier Minguez. *Robot Shape, Kinematics, and Dynamics in Sensor-Based Motion Planning*. PhD thesis, Universidad de Zaragoza, Spain, 2002.
- [33] Douglas C. Montgomery. *Design and Analysis of Experiments*. John Wiley & Sons, Inc., 1997.
- [34] Marco Morales. *Metrics for Sampling-Based Motion Planning*. PhD thesis, Department of Computer Science, Texas A&M University, 2007.
- [35] N. Munoz, J. Valencia, and N. Londoño. Evaluation of Navigation of an Autonomous Mobile Robot. In *Proc. of Int. Workshop on Performance Metrics for Intelligent Systems Work-*

BIBLIOGRAPHY

- shop (PerMIS)*, pages 15–21, 2007.
- [36] Jason M. O’Kane. *A Theory for Comparing Robot Systems*. PhD thesis, Department of Computer Science, University of Illinois, 2007.
 - [37] Kai Pfeiffer, Alexander Bubeck, Nikolaus Blümlein, and Martin Hägele. RoSta Deliverable D 4.3: Action Plan and Recommendation on Benchmarks for Mobile Manipulation and Service Robots. Technical report, Fraunhofer IPA, 2009.
 - [38] E Plaku, K. E. Bekris, and Kavraki L. E. OOPS for Motion Planning: An Online, Open-source, Programming System. In *Proc. of the IEEE International Conference on Robotics and Automation (ICRA)*, 2007.
 - [39] I. Ranó and J. Minguez. Steps Towards the Automatic Evaluation of Robot Obstacle Avoidance Algorithms. In *Workshop of Benchmarking in Robotics, in the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, 2006.
 - [40] M. Reggiani, M. Mazzoli, and S. Caselli. An Experimental Evaluation of Collision Detection Packages for Robot Motion Planning. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2002.
 - [41] Morten Strandberg. *Robot Path Planning: An Object-Oriented Approach*. PhD thesis, Department of Signals, Sensors and Systems. Royal Institute of Technology, Sweden, 2004.
 - [42] Ioan Sucan. OMPL - Open Motion Planning Library. <http://www.ros.org/wiki/ompl>.
 - [43] Mahmoud Tavakoli, Lino Marques, and Anibal T. de Almeida. Propose of a Benchmark for Pole Climbing Robots. In *European Robotics Symposium 2008*. Springer Berlin/Heidelberg, 2008.