From: AAAI Technical Report FS-93-03. Compilation copyright © 1993, AAAI (www.aaai.org). All rights reserved.

MICE and the Science of Vacuuming

David J. Musliner*and David Kortenkamp†
Artificial Intelligence Laboratory
The University of Michigan
Ann Arbor, MI 48109
djm@engin.umich.edu
korten@engin.umich.edu

Abstract

Artificial intelligence and robotics are fields notorious for developing many different systems that are never compared with each other in a quantitative way. This paper suggests that the task of vacuum-cleaning could be used as a benchmark for comparing different robot systems. Such a comparison should take place within a simulation environment, which is outlined in the paper. This paper also suggests a possible standard robot and some benchmark environments, as well as a set of quantitative evaluation metrics that could be used to compare robot systems.

Introduction

In general, research in robotics and artificial intelligence has been notoriously bereft of the type of rigorous scientific analysis and testing that lead to well-understood advances in a field. Instead, research in these areas has been characterized by "problem-itis" and short-term research projects that are developed, tested on a very small set of problems, and then abandoned. It is our contention that, in order to make truly meaningful advances in robotics and artificial intelligence, researchers need to pursue a more structured approach, particularly in the development of a set of common problems (benchmarks) to which various approaches may be applied and thus directly compared.

Some subfields of AI have been progressing towards this goal. For example, the machine learning community has established data-sets with which many ML algorithms can be tested and compared (e.g., the congressional voting records, soybean diseases, etc.) and the computer vision community has established sets of images with which to compare vision algorithms (e.g., the coke can). In contrast, other subfields of AI, such as planning, have no established sets of tasks with which to compare different systems. Unfortunately, mobile robotics falls in the latter category; rarely are two mobile robot systems tested using the same task, robot, and environment. The recent AAAI robot competitions [Dean and Bonasso, 1993] are some of the first instances in which different robots have been tested and compared by placing them in the same domain, with the same goals. Even so, because the physical robots in the competitions had dramatically different capabilities, the competitions can hardly be viewed as comparing robot control systems directly.

The topic of this workshop, robotic vacuum-cleaning, presents another opportunity for researchers in AI and robotics to make head-to-head comparisons of different approaches to the task of mobile robot control. The goal of research in this area should not be to have every research group develop their own vacuum-cleaning robot design, which is then tested a few times in their environment to see if it works. Instead, the goal should be to delineate several dimensions of the vacuum-cleaning task that are deemed essential research issues, and design a series of benchmarks that will measure, quantitatively, the degree to which a particular approach succeeds in addressing these issues. Doing so will not only allow a direct comparison between competing approaches, but will also show the overall progress of robotic systems in general.

In this paper, we describe a simulation testbed which could provide a well-understood, common environment for measuring the capabilities of any robot control system with respect to the vacuum-cleaning task. We briefly review the advantages and disadvantages of simulated environments before introducing the basis for our proposed system, the Michigan Intelligent Coordination Experiment (MICE) testbed. In the context of this environment, we describe a possible set of standard robotic capabilities, and we then propose a set of metrics which could prove useful in evaluating the performance of different robotic systems. The goal of this paper, then, is to describe a testbed environment for automated vacuuming systems which will reveal significant performance differences between various approaches to the vacuuming-agent control problem, by allowing them to operate in a strictly-controlled, com-

*Now at the University of Maryland, Computer Science Department, College Park, MD 20742

†Now at The MITRE Corporation, Houston, TX 77058

mon environment.

Advantages of Simulation

While simulated robots and environments should not be the only means of evaluating robotic systems, they offer many benefits to the mobile robot community. Simulations allow researchers to conduct numerous experiments with very low cost. Simulations also allow complete control over environmental and robotic variables such as the noisiness of sensors or the dynamics of external objects. With this type of control, simulators can support repeatable experiments under identical conditions, and they offer the opportunity to modify specific, isolated variables between testing runs. With real robots it is impossible to control all variables, so each experimental run is unique and non-repeatable. Real-world robots and environments also make it difficult or impossible to isolate variables and control them independently.

One of the most compelling arguments for the use of simulations is their support for reproducible, shareable experimental environments. Most research in robotics and artificial intelligence is never continued because it is impossible to reproduce another researcher's results without having access to the same robot, the same sensors, and the same environment. This is clearly infeasible. However, it is feasible for researchers to have access to the same simulated robot, simulated sensors, and simulated environment, since the simulator can be made available via ftp, etc.

Of course, the benefits of simulators are offset by some serious disadvantages. Because simulators allow complete control over the environment and over the robot, it is possible (in fact almost certain) that they abstract away much of the complexity, dynamics, and uncertainty of the real world. Thus, simulators need to be used very carefully. They cannot replace real-world experimentation using real robots; they complement such experimentation. Simulators can be built to emulate a robotic subsystem that is known to work in the real world, allowing the system to be tested in a controlled way that is not possible in the real world. Such a simulated system can then be compared with other robot systems that were tested in the same way. Testing robot systems in both the real world and in a controlled fashion using simulation will allow robotics to emerge as an exciting, yet disciplined scientific field.

In the following section we introduce a simulator that could be used as a robot simulation testbed for the vacuum-cleaning task. With the capabilities of this simulation environment in mind, we can then be concrete in our discussion of a vacuum-cleaning testbed and useful performance metrics.

MICE

The MICE testbed is a simulation tool originally designed for experimenting with coordination between intelligent systems [Montgomery and Durfee, 1990]. MICE simulates a two-dimensional grid-world in which agents may move, communicate with each other, and affect their environment. MICE is essentially a discrete-event simulator that helps control the domain and a graphical representation, but provides relatively few constraints on the form of the domain and the agents' abilities. Users may specify the time consumed by various agent activities, the constraints on an agents' sensors, the configuration of the domain and its properties, etc. MICE has been used to simulate numerous environments, including predator-prey domains, fire-fighting scenarios, and air-traffic control.

To interact with the MICE world, an agent issues commands to the testbed, which then simulates the execution of those commands in the environment, changing the world model in appropriate ways and returning information when sensor commands are issued. The simulator maintains its own simulation time distinct from wall-clock time, allowing an agent to specify how long it has "thought" about a problem, and how long various simulated actions will take.

In the context of simulating an environment for vacuum-cleaning robots, MICE has a number of useful capabilities:

- Static environment features can be defined for each grid element. This function is useful for representing immovable obstacles like walls, as well as other relevant features of grid locations (e.g., carpeting, wood flooring, etc). This mechanism may also be used to give a primitive third dimension (height) to the grid elements, if desired.
- Grid elements may also have dynamic features that can change by themselves (at programmed, possibly random times) or can be changed by an agent's behavior. The system supports both direct effects, such as a door that stays open when pushed, and indirect side-effects, such as the nap of a rug that is changed by the robot's motion over it.
- The simulator supports multiple agents interacting in a common environment. Agents may communicate directly over bounded communication channels, so negotiation and cooperation would be possible with other household robots (e.g., if the butler sees a plant knocked over, it calls the vacuumer). Other moving agents may also pose dynamic collision-avoidance challenges to the vacuum-cleaning robot, or may interfere with its activities (e.g., by replacing a chair the robot has moved).
- Agents are given only constrained sensors: the environment-designer can specify arbitrarily complex limitations on the types of information an agent can acquire about the world, the accuracy of that information, and the time costs of using the corresponding sensors. The simulation environment will take care of figuring out, for example, which objects are visible to a short robot, or which dust-balls are too small for a particular robot's sensors to detect.

Proposed agent

In order to conduct comparisons between robotic control systems, everyone must be using the same robot with the same sensors. To that end, we now propose a possible robotic agent for the vacuum-cleaning task, which should be feasible both as a simulated agent and a real-world implementation. The robot should have at least the following capabilities:

- can switch on and off vacuum, power usage changes.
- can move at a few set speeds; uses more power at faster speeds.
- fixed recharging station in environment, and sensor telling status of batteries.
- short-range obstacle proximity detectors to 2 grids distance, giving robot time of 1 grid-dist to stop before collision.
- longer range obstacle detectors of sonar-ring form... i.e., distance open in several directions. error bounds variable here too.
- ability to open/close doors, move smaller furniture (e.g., chairs but not entertainment center).

Proposed environments

In addition to a common robot, direct system comparisons will also require common environments. Obviously, one environment is not enough to test the complete capabilities of a vacuum-cleaning robot, since any commercially-produced robot sold for residential use will have to adapt to a variety of environments with little or no user modification.

The ease of constructing multiple domains and testing different agents many times in each domain is one of the most important advantages of simulation testbeds. Thus we have developed specifications for three classes of suitable environments, each of which can be parameterized in many ways to yield a large assortment of test cases.

The major environment classes are defined by the type of structure the robot will operate in:

- Apartments: Single-floor domains with several cluttered rooms and no hallways.
- Houses: Larger residences characterized by more rooms, a few short hallways, and open stairways between floors.
- Office buildings: Very large buildings with many rooms, long hallways, no open stairs, and relatively un-cluttered spaces.

For each of these general classes, many specific test instances can be generated by varying the architectural configuration (e.g., rooms, doors, stairways) as well as more detailed features such as furniture placement, carpeting, and other floor coverings. Other dynamic, autonomous agents can also be placed in each environment to represent people, pets, and other robots.

Each robot system will be tested in each of the environments over several simulated "days" in which the level of clutter, the placement of furniture, the introduction of other agents, and the status of doors is controlled. Each robot system will face exactly the same environments and exactly the same dynamics, allowing for completely comparable results. The next section describes some proposed evaluation metrics.

Proposed Evaluation Metrics

The following evaluation metrics could be used to quantitatively compare robot system, both to each other and to an "ideal" vacuum cleaning agent. Some of these metrics involve trade-offs (i.e., no system could be expected to do well on *all* of the metrics), for example, speed may be traded off with safety.

- **Efficiency**
  Efficiency metrics have to do with how quickly the robot can do the task and how optimally it can do the task. Ideal systems will vacuum rapidly and efficiently. Some of the efficiency metrics are:
  - total time to cover a given area.
  - percent of the given area covered.
  - percent of the given area re-covered (i.e., traversed repeatedly).
  - total power consumption.
- **Robustness**
  Robustness metrics have to do with how the robot handles uncertainty in the environment and in its sensors and actions. Robustness should also measure how the robot handles changes in the environment or unexpected events in the environment. An ideal system should cope with all levels of sensor noise and environment dynamics, even if coping simply means stopping. Some of the robustness metrics are:
  - behavior in the face of varying levels of sensor noise.
  - behavior given dynamic environment features such as doors and other agents (possibly malicious) that move without our agent's volition
  - behavior given large-scale variations in the environments, e.g., large open rooms vs. narrow corridors and small, obstacle-cluttered rooms.
- **Safety**
  Safety metrics have to do with how "housebroken" the robot is. Some bumping or stumbling can be expected (after all, people stub their toes), but will be minimized in good systems. An ideal system will rarely hit things and when it does it will be moving slowly. An ideal system should never fall down stairs or pick up small objects. Some of the safety metrics are:
  - how often does the robot bump into things?
  - does it ever run over the cat (or other small obstacles).

– does it fall down stairs?

- **Usability**
  The usability metrics measure the ease with which a user (the customer) can begin using the robot. These are necessarily more qualitative, but are important nonetheless. The ideal system would be taken out of the box, thrown on the floor, and would work. An ideal user interface might be a speech recognition system. Some of the usability metrics are:
  - amount of environment-specific, prior knowledge required.
  - number of environmental modifications.
  - quality of user interface.

Conclusion

We believe that simulations play an important role in testing and evaluating robotic systems. A common simulator and a common set of tasks and environments with which to test robot systems can help the mobile robot community identify promising avenues of research and identify issues that need to be addressed. However, simulations are only one part of the process and do not replace real-world experiments with real-world robots. This paper demonstrates how a current simulator could be used to evaluate the performance of different robotic systems all performing the same vacuum-cleaning task.

References

[Dean and Bonasso, 1993] Thomas Dean and R. Peter Bonasso. 1992 AAAI robot exhibition and competition. *AI Magazine*, 14(1), 1993.

[Montgomery and Durfee, 1990] Thomas A. Montgomery and Edmund H. Durfee. Using mice to study intelligent dynamic coordination. In *IEEE Conference on Tools for AI*, 1990.