Chapter 1

Introduction

There was a time when any book about robotics would need to start with a definition of what a robot is. This was, in some part, due to the fact that the robots of fiction had far outdistanced actual robots. Few people had seen an actual robot and authors needed to be mindful of the distractions, perhaps disappointments, which this could pose for their readers. It would be premature to say we are completely beyond this situation; however, robots are now much more common in the media and appear, and occasionally appear to behave, more like their fictional counterparts.

For example, the robot on the left in Figure 1-1 is the Robonaut 2, a humanoid robot designed to assist astronauts work and explore in space. The robot on the right is the HRP-4C humanoid developed by AIST in Japan. Both of these look and, to a limited extend, act more like the humanoid robots of fiction than like the industrial robots that have been with us since the late 1950s when George Devol and Joseph Engelberger founded the first robotics company, Unimation. Figure 1-2 shows a Puma® 560 robot arm, a robot built by Unimation for industrial applications such as welding, assembly, packing pallets and so forth.

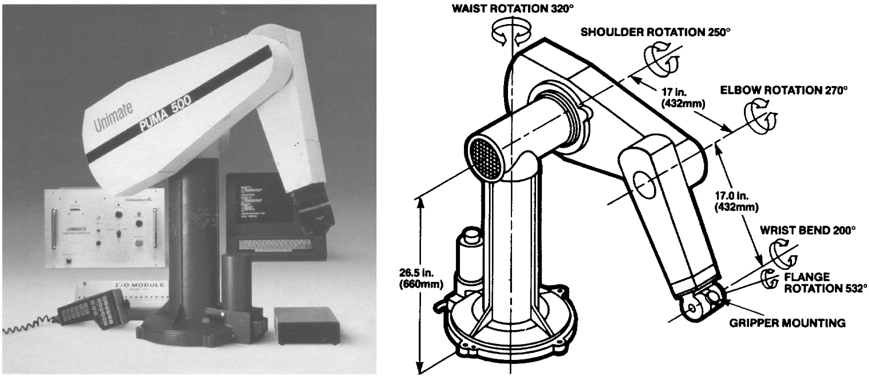
1.1. Robots

Robotic technology has been limited for much of its history to these kinds of industrial applications. The surrounding environment in which the industrial robot operates, that is to say the robot's "world," can be tightly controlled. The objects that the robot needs to manipulate or interact with

http://www.worldscibooks.com/compsci/7080.html

(a) (b)

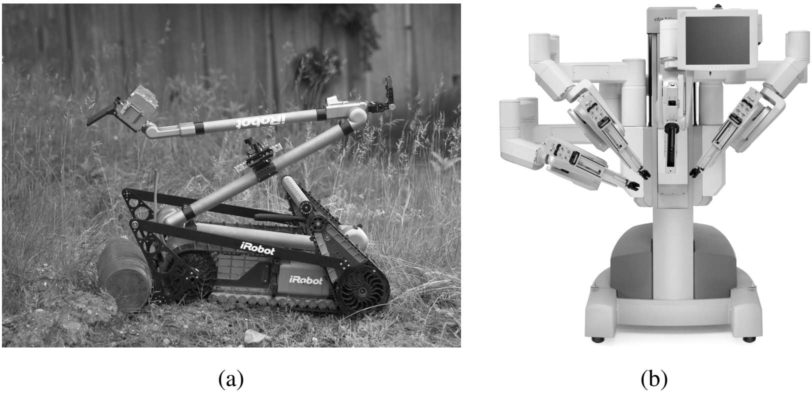
Figure 1-1: (a) Robonaut 2 (R2) (reprinted with permission from NASA); (b) Humanoid robot HRP-4C (reprinted with permission from AIST, Japan).



(a) (b)

Figure 1-2: Unimation Puma® 560 industrial robot.

can be precisely defined, and their timing and locations prearranged with a high degree of precision. In this way, the environment can be restricted to situations that the robot sensors, actuators and controller can handle quickly and successfully.



(a) (b)

Figure 1-3: (a) iRobot 510 PackBot® (reprinted by permission of iRobot); (b) da Vinci® Surgical System (reprinted by permission of Intuitive Surgical Inc).

Robotic technology is now spreading quickly into other, new application areas, including:

- **Military robots** such as the Reaper, Predator and Gnat unmanned aerial vehicles (UAV's) from General Atomics, the Talon® unmanned ground vehicles (UGV) from Foster Miller and the PackbBot® from iRobot.
- **Professional service robots** such as RobotWatch's OFRO® security robots, Pipe and tunnel cleaning robots from RedZone Robotics, the da Vinci® surgical robot, and robot guides in hospitals and museums;
- **Robots for the home consumer market**, such as the Sony AIBO® and QRIO®, iRobot Roomba® and Scooba®; and,
- **Robots for the educational market** such as Lego Mindstorm®, Stiquetto™, the HandyBoard, etc.

In many of these new applications areas, the environment in which the robot operates cannot be as tightly controlled as it can be in manufacturing applications. Therefore the robot needs to be capable of sensing its environment and exercising some degree of *autonomous action* if it is to deal effectively with the unknowns it may face. A security robot, for example, cannot predict where, when and what objects or people it will come across. It needs to invest sensor and processing time into monitoring its environment in a way that manufacturing robots typically don't. It needs to select appropriate actions to ensure it achieves its task objectives without damaging itself or others.

The expansion of robotic technology into this wide panoply of new and demanding application areas is fueled by improved battery, motor, sensor and computation technologies, providing the necessary higher performance for a lower cost.

1.2. Cluster Computing

Cluster computing is an approach to achieving high performance, high reliability or high throughput computing by using a collection of interconnected computer systems. The first Beowulf cluster was built by Donald Becker and Thomas Sterling at NASA's Center for Excellence in Space Data and Information Sciences in 1994. Their goal was to build Commodity Off-The-Shelf (COTS) based systems to satisfy specific computational requirements. They called their cluster of 16 off-the-shelf DX4 processors "Beowulf," and that name has now come to denote the entire class of COTS based cluster machines. The goal of cluster

Figure 1-4: 128-Processor Beowulf cluster built by Donald Becker while he was at NASA/GSFC (reprinted with permission Michigan Technological University).

CLUSTER COMPUTING FOR ROBOTICS AND COMPUTER VISION

technology — supercomputing performance at off-the-shelf prices — is one that is directly in tune with the current needs of robotics and computer vision. Inexpensive, high performance computing is one crucial factor in building robots that can sense and respond effectively to events in this new range of unstructured environments. The additional computing power can be used to process sonar and visual information more quickly and/or more thoroughly; it can be used to recognize and track additional features of the robot's environment; and it can be used to pick better action strategies, even considering those based on past performance and observations.

The additional computing performance comes at the cost of developing robotic computing algorithms that can exploit the parallel computing resources available in a cluster. To understand why this is not trivial consider the physical task of digging a hole in your back garden. If it takes you a certain time to dig a $1\,m$ deep hole of $1\,m$ in circumference with a shovel, you can theorize that having a friend help you dig would cut the time in half. This speed-up will only be realized if you and your friend do not get in each other's way while digging! Just as a four-man rowing crew needs to coordinate its strokes to avoid oar collisions, two people digging a hole may need to coordinate digging strokes if the best speed-up is to be obtained. Similarly, in designing algorithms for cluster computation, significant effort may need to be put into the division of subtasks between cluster processors so as to get an effective speed-up.

Recently, multi-core systems have been adopted by processor manufacturers as a solution to improving processor performance. A multi-core Central Processing Unit (CPU) has several processor cores built into a single chip. Each core in a multi-core processor can function as a separate cluster processor. A Graphical Processing Unit (GPU) is a processor for handling graphical operations such as texture mapping, shading, image processing and so forth. A CPU can offload graphics tasks to one of these auxiliary processors. Multi-core technology has also been applied to constructing more powerful GPUs and these can also be incorporated into cluster computing.

In this book, we will look at how cluster technology can be leveraged to build better robots. More specifically, we will concern ourselves with the computer algorithms that control robot behavior. Algorithms and approaches in key areas of robotics and computer vision will be introduced.

These will broadly include areas such as map building, localization, target tracking, action selection and learning. Each will be introduced and cluster implementations for algorithms in these areas presented. It would be difficult to provide a completely comprehensive collection of robot and computer vision algorithms. The ones chosen for this text were selected for their accessibility to new robotics or cluster computing practitioners and as broad examples of classes of algorithms or approaches.

1.3. Overview of the Book

Chapter 2 introduces the basic concepts of parallel processing and cluster computing. The focus is on understanding why cluster computing has become one of the dominant forms of parallel computing and how this relates to robotics. In addition to some more practical information about constructing clusters, this section lays the ground work for the mathematical models of computing and communication introduced in the next chapter.

Chapter 3 introduces the basic concepts in parallel programming, focusing specifically on message passing using the MPI standard. In addition to more practical information about compiling and running programs with MPI, this chapter also introduces several mathematical models used in later chapters to analyze the performance of parallel algorithms. In particular logarithmic time models are defined for the collective communication operations in MPI.

With Chapters 2 and 3 providing a cluster computing foundation, Chapter 4 begins the study of robot algorithms. It introduces the basic terminology for the motion of a wheeled mobile robot. The design process used in the remainder of the book is introduced here for the problem of *dead-reckoning*, that is, calculating the location of the robot based only on the motion commands transmitted to the robot. The design process consists of understanding how the data and operations can be *partitioned* on the cluster, carrying out the *program design* based on this partition, and *analyzing* the performance of the result. The MPI collective communication operations for *scattering*, *gathering* and *reducing* data are introduced here.

Chapter 5 looks at processing point data from sonar and laser sensors. In particular the Hough Transform algorithm is used to identify prevalent straight lines in a collection of points, a case that clearly shows the importance of balancing computation load on a cluster.

The important mobile robotics activities of localization (determining where the robot is with respect to a map) and mapping (building a map of the robot's surroundings from sensory data) form the basis of the material in Chapter 6. Map representation on a cluster is discussed and a parallel implementation of Monte Carlo localization is developed.

Computer vision algorithms can be one of the most computationally expensive algorithms that a robot needs to carry out. In Chapter 7, cluster implementations for iconic, or pixel-level computer vision operations are presented, as well as implementations for multi-scale versions of these operations. The important field of visual tracking is introduced and a parallel implementation of the Condensation algorithm using spatial histograms is presented.

Landmark selection and recognition can play a key role in allowing a robot to build a large-scale map. Chapter 8 continues the computer vision theme looking at issues involved in learning visual landmarks. Two unsupervised learning algorithms, K-Means and Expectation-Maximization, are presented.

Behavior-based approaches have had a profound impact on the field of robotics. Chapter 9 introduces the concept of behavior-based robotics and robot architectures. An MPI implementation of a typical, static behavior-based architecture is developed. The advantages of adding dynamic process creation and destruction to behavior-based robot architectures are then discussed. The new process control features of MPI 2.2 are used to build a dynamic behavior-based architecture.

CLUSTER COMPUTING FOR ROBOTICS AND COMPUTER VISION