ROBOT LEARNING BY VISUAL OBSERVATION

ALEKSANDAR VAKANSKI
FARROKH JANABI-SHARIFI

WILEY

Robot Learning by Visual Observation

Aleksandar Vakanski

Farrokh Janabi-Sharifi

WILEY

This edition first published 2017

Registered Offices
John Wiley & Sons, Inc., 111 River Street, Hoboken, NJ 07030, USA

Editorial Office
111 River Street, Hoboken, NJ 07030, USA

For details of our global editorial offices, customer services, and more information about Wiley products visit us at www.wiley.com.

Wiley also publishes its books in a variety of electronic formats and by print-on-demand. Some content that appears in standard print versions of this book may not be available in other formats.

Library of Congress Cataloguing-in-Publication Data

Names: Vakanski, Aleksandar, author. | Janabi-Sharifi, Farrokh, 1959– author.
Title: Robot learning by visual observation / Aleksandar Vakanski, Farrokh Janabi-Sharifi.
Description: Hoboken, NJ, USA : John Wiley & Sons, Inc., [2017] | Includes bibliographical references and index.
Identifiers: LCCN 2016041712| ISBN 9781119091806 (cloth) | ISBN 9781119091783 | ISBN 9781119091998 (epub) | ISBN 9781119091783 (epdf)
Subjects: LCSH: Robot vision. | Machine learning. | Robots–Control systems. | Automatic programming (Computer science)
Classification: LCC TJ211.3 .V35 2017 | DDC 629.8/92631–dc23
LC record available at https://lccn.loc.gov/2016041712

Cover image: fandijki/Gettyimages

Cover design by Wiley

Set in 10/12pt Warnock by SPi Global, Pondicherry, India

10 9 8 7 6 5 4 3 2 1

Robot Learning by Visual Observation

To our families

Preface

The ability to transfer knowledge has played a quintessential role in the advancement of our species. Several evolutionary innovations have significantly leveraged the knowledge transfer. One example is rewiring of the neuronal networks in primates' brains to form the so-called mirror neuron systems, so that when we observe tasks performed by others, a section of the brain that is responsible for observation and a section that is responsible for motor control are concurrently active. Through this, when observing actions, the brain is attempting at the same time to learn how to reproduce these actions. The mirror neuron system represents an especially important learning mechanism among toddlers and young kids, stimulating them to acquire skills by imitating the actions of adults around them. However, the evolutionary processes and modifications are very slow and prodigal, and as we further developed, we tended to rely on employing our creativity in innovating novel means for transferring knowledge. By inventing writing and alphabets as language complements, we were able to record, share, and communicate knowledge at an accelerated rate. Other innovations that followed, such as the printing press, typing machine, television, personal computers, and World Wide Web, each have revolutionized our ability to share knowledge and redefined the foundations for our current level of technological advancement.

As our tools and machines have grown more advanced and sophisticated, society recognized a need to transfer knowledge to the tools in order to improve efficiency and productivity, or to reduce efforts or costs. For instance, in the manufacturing industry, robotic technology has emerged as a principal means in addressing the increased demand for accuracy, speed, and repeatability. However, despite the continuous growth of the number of robotic applications across various domains, the lack of interfaces for quick transfer of knowledge in combination with the lack of intelligence and reasoning abilities has practically limited operations of robots to preprogrammed repetitive tasks performed in structured environments. Robot programming by demonstration (PbD) is a promising form for transferring new skills to robots from observation of skill examples performed by a demonstrator. Borrowed from the observational imitation learning mechanisms among humans, PbD has a potential to reduce the costs for the development of robotic applications in the industry. The intuitive programming style of PbD can allow robot programming by end-users who are experts in performing an industrial task but may not necessarily have programming or technical skills. From a broader perspective, another important motivation for the development of robot PbD systems is the old dream of humankind about robotic assistance in performing everyday domestic tasks. Future advancements in PbD would allow the general population to program domestic and service robots in a natural way by demonstrating the required task in front of a robot learner.

Arguably, robot PbD is currently facing various challenges, and its progress is dependent on the advancements in several other research disciplines. On the other hand, the strong demand for new robotic applications across a wide range of domains, combined with the reduced cost of actuators, sensors, and processing memory, is amounting for unprecedented progress in the field of robotics. Consequently, a major motivation for writing this book is our hope that the next advancements in PbD can further increase the number of robotic applications in the industry and can speed up the advent of robots into our homes and offices for assistance in performing daily tasks.

The book attempts to summarize the recent progress in the robot PbD field. The emphasis is on the approaches for probabilistic learning of tasks at a trajectory level of abstraction. The probabilistic representation of human motions provides a basis for encapsulating relevant information from multiple demonstrated examples of a task. The book presents examples of learning industrial tasks of painting and shot peening by employing hidden Markov models (HMMs) and conditional random fields (CRFs) to probabilistically encode the tasks. Another aspect of robot PbD covered in depth is the integration of vision-based control in PbD systems. The presented methodology for visual learning performs all the steps of a PbD process in the image space of a vision camera. The advantage of such learning approach is the enhanced robustness to modeling and measurement errors.

The book is written at a level that requires a background in robotics and artificial intelligence. Targeted audience consists of researchers and educators in the field, graduate students, undergraduate students with technical knowledge, companies that develop robotic applications, and enthusiasts interested in expanding their knowledge on the topic of robot learning. The reader can benefit from the book by grasping the fundamentals of vision-based learning for robot programming and use the ideas in research and development or educational activities related to robotic technology.

We would like to acknowledge the help of several collaborators and researchers who made the publication of the book possible. We would like to thank Dr. Iraj Mantegh from National Research Council (NRC)—Aerospace Manufacturing Technology Centre (AMTC) in Montréal, Canada, for his valuable contributions toward the presented approaches for robotic learning of industrial tasks using HMMs and CRFs. We are also thankful to Andrew Irish for his collaboration on the aforementioned projects conducted at NRC-Canada. We acknowledge the support from Ryerson University for access to pertinent resources and facilities, and Natural Sciences and Engineering Research Council of Canada (NSERC) for supporting the research presented in the book. We also thank the members of the Robotics, Mechatronics and Automation Laboratory at Ryerson University for their help and support. Particular thanks go to both Dr. Abdul Afram and Dr. Shahir Hasanzadeh who provided useful comments for improving the readability of the book. Last, we would like to express our gratitude to our families for their love, motivation, and encouragement in preparing the book.

List of Abbreviations

CRF	conditional random field
DMPs	dynamic motion primitives
DoFs	degrees of freedom
DTW	dynamic time warping
GMM	Gaussian mixture model
GMR	Gaussian mixture regression
GPR	Gaussian process regression
HMM	hidden Markov model
IBVS	image-based visual servoing
LBG	Linde–Buzo–Gray (algorithm)
PbD	programming by demonstration
PBVS	position-based visual servoing
RMS	root mean square
SE	special Euclidean group

Contents

1
Introduction

Robot programming is the specification of the desired motions of the robot such that it may perform sequences of prestored motions or motions computed as functions of sensory input (Lozano-Pérez, 1983).

In today's competitive global economy, shortened life cycles and diversification of the products have pushed the manufacturing industry to adopt more flexible approaches. In the meanwhile, advances in automated flexible manufacturing have made robotic technology an intriguing prospect for small- and medium-sized enterprises (SMEs). However, the complexity of robot programming remains one of the major barriers in adopting robotic technology for SMEs. Moreover, due to the strong competition in the global robot market, historically each of the main robot manufacturers has developed their own proprietary robot software, which further aggravates the matter. As a result, the cost of robotic tasks integration could be many folds of the cost of robot purchase. On the other hand, the applications of robots have gone well beyond the manufacturing to the domains such as household services, where a robot programmer's intervention would be scarce or even impossible. Interaction with robots is increasingly becoming a part of humans' daily activities. Therefore, there is an urgent need for new programming paradigms enabling novice users to program and interact with robots. Among the variety of robot programming approaches, *programming by demonstration* (PbD) holds a great potential to overcome complexities of many programming methods.

This introductory chapter reviews programming approaches and illustrates the position of PbD in the spectrum of robot programming techniques. The PbD architecture is explained next. The chapter continues with applications of PbD and concludes with an outline of the open research problems in PbD.

Robot Learning by Visual Observation, First Edition. Aleksandar Vakanski and Farrokh Janabi-Sharifi.

1.1 Robot Programming Methods

A categorization of the robot programming modes based on the taxonomy reported by Biggs and MacDonald (2003) is illustrated in Figure 1.1. The conventional methods for robot programming are classified into manual and automatic, both of which rely heavily on expensive programming expertise for encoding desired robot motions into executable programs.

The *manual programming systems* involve text-based programming and graphical interfaces. In text-based programming, a user develops a program code using either a controller-specific programming language or extensions of a high-level multipurpose language, for example, C++ or Java (Kanayama and Wu, 2000; Hopler and Otter, 2001; Thamma *et al.*, 2004). In both cases, developing the program code is time-consuming and tedious. It requires a robot programming expert and an equipped programming facility, and the outcomes rely on programmer's abilities to successfully encode the required robot performance. Moreover, since robot manufacturers have developed proprietary programming languages, in industrial environments with robots from different manufacturers, programming robots would be even more expensive. The graphical programming systems employ graphs, flowcharts, or diagrams as a medium for creating a program code (Dai and Kampker, 2000; Bischoff *et al.*, 2002). In these systems, low-level robot actions are represented by blocks or icons in a graphical interface. The user creates programs by composing sequences of elementary operations through combination of the graphical units. A subclass of the graphical programming systems is the robotic simulators, which create a virtual model of the robot and the working environment, whereby the virtual robot is employed for emulating the motions of the actual robot (Rooks, 1997). Since the actual robot

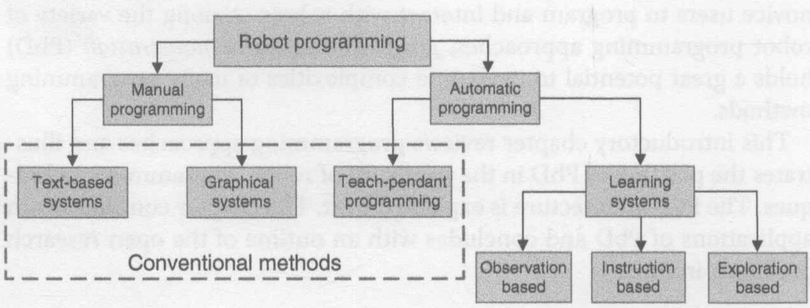

Figure 1.1 Classification of robot programming methods. (Data from Biggs and MacDonald (2003).)

is not utilized during the program development phase, this programming method is referred to as off-line programming (OLP).

The conventional *automatic programming systems* employ a teach-pendant or a panel for guiding the robot links through a set of states to achieve desired goals. The robot's joint positions recorded during the teaching phase are used to create a program code for task execution. Although programming by teach-pendants or panel decreases the level of required expertise, when compared to the text-based programming systems, it still requires trained operators with high technical skills. Other important limitations of the guided programming systems include the difficulties in programming tasks with high accuracy requirements, absence of means for tasks generalizations or for transfer of the generated programs to different robots, etc.

The stated limitations of the conventional programming methods inspired the emergence of a separate class of automatic programming systems, referred to as *learning systems*. The underlying idea of robot learning systems originates from the way we humans acquire new skills and knowledge. Biggs and MacDonald (2003) classified these systems based on the corresponding forms of learning and solving problems in cognitive psychology: exploration, instruction, and observation. In *exploration-based systems*, a robot learns a task with gradually improving the performance by autonomous exploration. These systems are often based on reinforcement learning techniques, which optimize a function of the robot states and actions through assigning rewards for the undertaken actions (Rosenstein and Barto, 2004; Thomaz and Breazeal, 2006; Luger, 2008). *Instructive systems* utilize a sequence of high-level instructions by a human operator for executing preprogrammed robot actions. Gesture-based (Voyles and Khosla, 1999), language-based (Lauria *et al.*, 2002), and multimodal communication (McGuire *et al.*, 2002) approaches have been implemented for programming robots using libraries of primitive robot actions. *Observation-based systems* learn from observation of another agent while executing the task. The PbD paradigm is associated with the observation-based learning systems (Billard *et al.*, 2008).

1.2 Programming by Demonstration

Robot PbD is an important topic in robotics with roots in the way human beings ultimately expect to interact with a robotic system. Robot PbD refers to automatic programming of robots by demonstrating sample tasks and can be viewed as an intuitive way of transferring skill and tasks knowledge to a robot. The term is often used interchangeably with learning by demonstration (LbD) and learning from demonstration (LfD) (Argall *et al.*, 2009; Konidaris *et al.*, 2012). PbD has evolved as an interdisciplinary field of robotics, human–robot interaction (HRI), sensor fusion, machine learning, machine vision, haptics, and motor control. A few surveys of robot PbD are available in the literature (e.g., Argall *et al.*, 2009). PbD can be perceived as a class of supervised learning problems because the robot learner is presented with a set of labeled training data, and it is required to infer an output function with the capability of generalizing the function to new contexts. In the taxonomy of programming approaches shown in Figure 1.1, PbD is a superior learning-based approach. Compared to the exploration-based learning systems (as an unsupervised learning problem), PbD systems reduce the search space for solutions to a particular task, by relying on the task demonstrations. The learning is also faster because the trial and errors associated with the reinforcement methods are eliminated.

In summary, the main purpose in PbD is to overcome the major obstacles for natural and intuitive way of programming robots, namely lack of programming skills and scarcity of task knowledge. In industrial settings, this translates to reduced time and cost of programming robots by eliminating the involvement of a robot programmer. In interactive robotic platforms, PbD systems can help to better understand the mechanisms of HRI, which is central to social robotics challenges. Moreover, PbD creates a collaborative environment in which humans and robots participate in a teaching/learning process. Hence, PbD can help in developing methods for robot control which integrate safe operation and awareness of the human presence in human–robot collaborative tasks.

1.3 Historical Overview of Robot PbD

Approaches for automatic programming of robots emerged in the 1980s. One of the earlier works was the research by Dufay and Latombe (1984) who implemented inductive learning for the robot assembly tasks of mating two parts. The assembly tasks in this work were described by the geometric models of the parts, and their initial and final relations. Synthesis of program codes in the robotic language was obtained from training and inductive (planning) phases for sets of demonstrated trajectories. In this pioneering work on learning from observation, the sequences of states and actions were represented by flowcharts, where the states described the relations between the mating parts and the sensory conditions.

Another early work on a similar topic is the assembly-plan-from-observation (APO) method (Ikeuchi and Suehiro, 1993). The authors presented a method for learning assembly operations of polyhedral objects. The APO paradigm comprises the following six main steps: temporal segmentation of the observed process into meaningful subtasks, scene objects recognition, recognition of performed assembly task, grasp recognition of the manipulated objects, recognition of the global path of manipulated objects for collision avoidance, and task instantiation for reproducing the observed actions. The contact relations among the manipulated objects and environmental objects were used as a basis for constraining the relative objects movements. Abstract task models were represented by sequences of elementary operations accompanied by sets of relevant parameters (i.e., initial configurations of objects, grasp points, and goal configurations).

Munch *et al.* (1994) elaborated on the role of the teacher as a key element for successful task reproduction. The learning was accomplished through recognition of elementary operations for the observed tasks. The demonstrator supervised and guided the robot's knowledge acquisition by (i) taking into considerations the structure of robot's perceptibility sensors when providing examples, (ii) taking part in preprocessing and segmentation of the demonstrations, and (iii) evaluating the proposed task solution.

Ogawara *et al.* (2002a) proposed to generate a task model from observations of multiple demonstrations of the same task, by extracting particular relationships between the scene objects that are maintained throughout all demonstrations. Each demonstration was represented as a sequence of interactions among the user's hand, a grasped object and the environmental objects. The interactions that were observed in all demonstrations were called essential interactions, whereas the variable parts of the demonstrations called nonessential interactions were ignored in the task planning step. Generalization across multiple demonstrations was carried out by calculating the mean and variance of all trajectories for the essential interactions. A robot program was generated from the mean trajectory, and mapped to robot joints' motors using an inverse kinematics controller.

The advancements in the fields of machine learning and artificial intelligence in the past two decades produced an abundance of new methods and approaches. This trend was reflected by the implementation of approaches in robot PbD based on neural networks (Liu and Asada, 1993; Billard and Hayes, 1999), fuzzy logic (Dillmann *et al.*, 1995), statistical models (Yang *et al.*, 1994; Tso and Liu, 1997; Calinon, 2009), regression techniques (Atkeson *et al.*, 1997; Vijayakumar and Schaal, 2000), etc.