Showing Robots How to Follow People using a Broomstick Interface

James E. Young[1,3], Kentaro Ishii[2], Takeo Igarashi[2,3], Ehud Sharlin[1]
jim.young@ucalgary.ca, kenta@designinterface.jp, takeo@acm.org, ehud@cpsc.ucalgary.ca

[1]University of Calgary
Calgary, Canada

[2]Design Interface Project, JST Erato
Tokyo, Japan

[3]The University of Tokyo
Tokyo, Japan

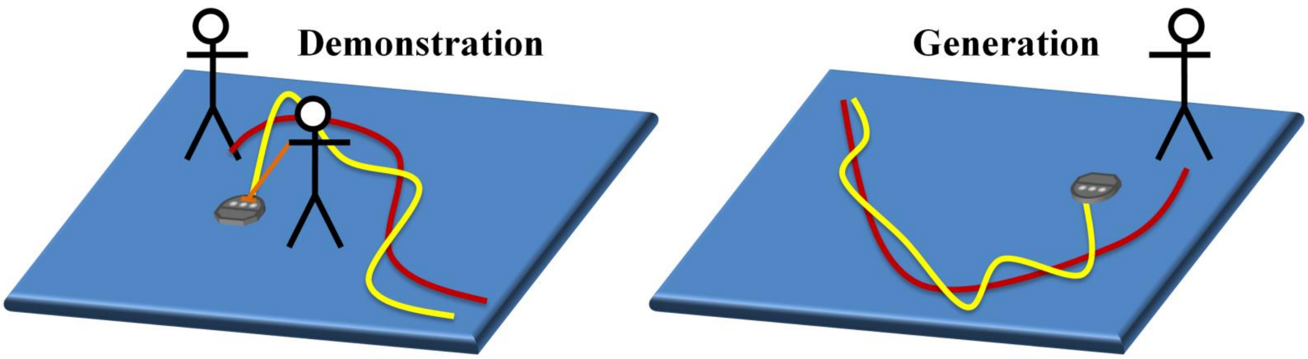

Figure 1 –During demonstration the primary person walks as they normally would in the given circumstance and the designer pushes a robot (iRobot Roomba) on a broomstick to demonstrate the style in which the robot should follow the primary person. Following, during generation the primary person walks normally and the interactive robot path and style is automatically generated in real-time to react to the primary person in a way that matches the demonstration style.

Abstract—**Robots are poised to enter our everyday environments such as our homes and offices, contexts that present unique questions such as the style of the robot's actions. Style-oriented characteristics are difficult to define programmatically, a problem that is particularly prominent for a robot's interactive behaviors, those that must react accordingly to dynamic actions of people. In this paper, we present a technique for programming the style of how a robot should follow a person by demonstration, such that non-technical designers and users can directly create the style of following using their existing skill sets. We envision that simple physical interfaces like ours can be used by non-technical people to design the style of a wide range of robotic behaviors.**

I. Introduction

The design of robots that enter our everyday environments must be considered beyond their utility and functionality, as people generally care a great deal about the style of objects and technologies that they possess [2]. Style and fashion are often important factors of technology adoption, and people will want attractive and pleasing robots the same as they want an attractive table, wristwatch, or car – design is directly related to user experience and satisfaction [1, 2].

Advanced programming is generally required to get robots to do even simple tasks in the real world, and the style of the resulting goal- and efficiency-oriented actions often tends to be very mechanical. It is difficult to programmatically embed richer style into robot actions, e.g., to program robots that pick up objects, shake your hand, or follow you, to do these actions "in style", for example, following an aggressive, timid, or careful fashion. These problems are exacerbated for non-scripted interactive robot behaviors, where the robot must interpret a changing environment or person's actions, and react appropriately, in real time.

We propose to use programming-by-demonstration to directly design the *style* of robot interaction – we focus on the *style* of how a robot should interact with a person, rather than any particular goal-oriented interaction with the environment.

Our current exploration focuses on enabling people to demonstrate to a robot how it should follow a person, a practical yet stylistically flexible and diverse interactive robotic task. Our implementation is a robotic extension to Puppet Master [3], a programming-by-demonstration system for the locomotion-based interactive styles of animated avatars. We choose this as it has been shown to capture style and emotion of interactive movements from an example demonstration.

II. Style-by-Demonstration

We see our approach as a kind of style-by-demonstration, where, through demonstration, the robot is programmed to act in a particular style in reaction to the real-time actions of a person. During the demonstration phase, example motions are provided both for the person (primary) and for the robot (reactionary), showing by acting how a robot should respond to the person's movements. Both paths are given simultaneously in real time (Figure 1).

After demonstration, the generation phase takes the style of the reactionary characteristics shown in the demonstration and incorporates it into real-time interaction. Here, the person moves freely while new, appropriately styled, robot actions are generated on the fly to respond to the person's movements. Our approach generally allows demonstration to be relatively short, from 30s to 2 minutes, and real-time generation occurs automatically and immediately, with no need for pre-processing.

III. Broomstick Interaction

The primary goal in developing an interface to enable a person to demonstrate stylistic motion to a robot was to make it easy to use, and to enable a person to focus more on the motion style and less on the mechanics of moving the robot. One note is that, for demonstrating the robot input, we ruled out the possibility of directly tracking a person's movements: the resulting demonstration would be too expressive and would contain motions and nuances not reproducible by the robot.

We developed a broomstick interface (Figure 2), where a regular aluminum broomstick is attached to an iRobot Roomba – using a real Roomba for demonstration allows us to encapsulate the robot's model of movement, expressive capabilities and limitations. The broomstick is attached to the robot via a two-axis swivel, allowing it to be freely moved forward, backward, left and right. The Roomba itself can be moved forward and backward by pushing and pulling, and twisting the broomstick turns the Roomba.

The result is a natural and familiar mechanism (and situation) for demonstrating following style to the robot. In our implementation (Figure 2), one person walks naturally while another person uses the broomstick to demonstrate a following style to the robot (reactionary entity). When demonstration is finished, a real (non-broomstick) Roomba enters the interaction space and follows the person in the style that was demonstrated.

Figure 2 – demonstrating how a robot should follow a person (left) and generation, where the robot automatically mimics the demonstration to follow the person (right)

IV. Implementation

Our implementation was realized through the extension of the Puppet Master [2] animation system. The primary challenge of adopting the animation-only Puppet Master system to real robots was to be flexible to the hard, real-world limitations of robots, as robots cannot be as easily moved, manipulated, and their movements cannot be smoothed or filtered in the same flexible way that on-screen animated characters can. Our early resulting robot output did not resemble the training – it was jittery, movements appeared incoherent, and the robot was not able to maintain localization (its position in respect to the other entity, the person).

Our solution surrounds the observation that the *texture*, or high-frequency detail movements of the Puppet Master output, are very important to the quality of the resulting movement. However, equally important is the robot's position in relation to the person, where the fine-detailed motions are intended to happen. For example, a robot motion would have different meanings if it were performed away from the person, in front of the person, or behind the person.

We filtered the Puppet Master output through a model of the robot's movement capabilities to obtain robot commands that would best reproduce the desired texture. Then, we modified these texture commands such that the robot *tends* toward the target location. The result is what we believe to be a very good representation of the Puppet Master output, such that the robot output matches well the demonstration given.

V. Discussion and Future Work

We are currently in the process of conducting and analyzing a multi-part user study surrounding our implementation, including comparisons to other systems as well as more exploratory and qualitative evaluations.

We believe that programming robotic style by demonstration is a viable approach to designing robotic interaction. Our initial research through our implementation presented here suggests that not only is programming style by demonstration possible, but that it is a natural and intuitive way for non-technical people to be involved in programming robots. We intend to investigate these ideas further through extended implementations and controlled user studies.

References

[1] Donald A. Norman. *Emotional Design: Why We Love (or Hate) Everyday Things*. Basic Books, NY, 2004.

[2] James E. Young, Richard Hawkins, Ehud Sharlin, and Takeo Igarashi. Toward acceptable domestic robots: Applying insights from social psychology. *Inagural Issue of the Int. J. Social Robotics. January 2009*, 1(1), 2009.

[3] James E. Young, Takeo Igarashi, and Ehud Sharlin. Puppet master: Designing reactive character behavior by demonstration. In *ACM SIGGRAPH / EG SCA '08.*, pages 183–191, Germany, 2008. EG Press, EG Association.