RoboJockey: Real-time, Simultaneous, and Continuous Creation of Robot Actions for Everyone

Takumi Shirokura[1&2], Daisuke Sakamoto[1], Yuta Sugiura[1&3],
Tetsuo Ono[2], Masahiko Inami[1&3] and Takeo Igarashi[1&4]

[1]JST, ERATO, IGARASHI Design UI Project [2]Hokkaido University [3]Keio University [4]The University of Tokyo

{sakamoto | sugiura | inami}@designinterface.jp, {shirokura | tono}@complex.eng.hokudai.ac.jp, takeo@acm.org

ABSTRACT

We developed a *RoboJockey* (Robot Jockey) interface for coordinating robot actions, such as dancing - similar to “Disc jockey” and “Video jockey.” The system enables a user to choreograph a dance for a robot to perform by using a simple visual language. Users can coordinate humanoid robot actions with a combination of arm and leg movements. Every action is automatically performed to background music and beat. The *RoboJockey* will give a new entertainment experience with robots to the end-users.

ACM Classification: H5.2 [Information interfaces and presentation]: User Interfaces. - Graphical user interfaces.

General terms: Design, Human Factors.

Keywords: Creation of robot action, robot jockey interface, multi-touch interface, visual language.

INTRODUCTION

Robots have been entertaining people in various ways, such as in amusement and theme parks, museums, shopping malls, and homes. Watching a robot dance is fun, but coordinating them is difficult. It takes a long time for robotics engineers to make robots that can entertain.

Although robots are becoming popular in daily life, it is difficult for the end-user to control their robots’ actions and motions. Robot action-coordination interfaces have been developed by robotics engineers recently. These interfaces were developed for robotics experts who can construct robots on their own, or have knowledge of robotics. Furthermore, it is difficult to coordinate dances for robots even for robotics experts. Unfortunately, robots are still difficult to use for end-users.

We developed an interface for easily controlling robots and providing new entertainment experiences to the end-user, who is not only controlling the robot but also watching it dance. The interface has a simple visual language, which allows the user to control a robot without any prior experience and knowledge about robots. User-created action is performed on the robots immediately. The *RoboJockey* interface will provide a feeling of “Robot Jockey,” who creates and play robot dances for an audience, to the end-users.

UIST’10, October 3–6, 2010, New York, New York, USA.
ACM 978-1-4503-0271-5/10/10..

RELATED WORK

This work is inspired by "Turning the tables" by Taylor et al. [5]. They proposed an intuitive "Visual jockey" interface on a tabletop. We extended their idea to support physical robots. We created a simple visual language to coordinated robot actions. There are visual languages and programming interfaces for robots [2][3][4]; however, programming and execution are separate in these systems. Further these interfaces were not easy-to-use for the end users. *RoboJockey* provides both experts and non-experts the chance to enjoy themselves by enabling them to coordinate robot actions while watching the robot dance. The tangible interfaces will be a good idea for creating entertainment systems, like TurTan [1]. In this paper, we just use a tabletop interface for showing our concept of RoboJockey.

VISUAL LANGUAGE

Users coordinate robot actions using a simple visual language on the tabletop interface. The visual language interface has two types of objects, robot and action. The user can coordinate a robot's actions by connecting the action object to the robot one. All objects are represented as an icon, which express the function of the object (Figure 2).

Basic Idea

A robot object has one seek ball circulating around it (Figure 1). When the seek ball crosses a line connected to an action object, the robot performs the action. That action continues until the seek ball crosses another line. This mechanism enables the robots to continuously and simultaneously performs various motions with combinations discussed in the next sub-section. Connected objects can be disconnected and reconnected to the robot object.

Primitive Objects

Robot object

A robot object represents a physical robot. An association between the robot object and the physical robot is shown

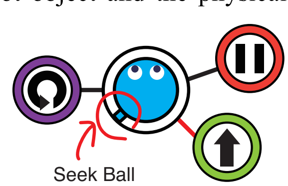

Figure 1: Example of visual program; the robot moves “forward”. It will continue until the seek ball crosses another action object, in this case, “Rotate-left”.

by the color of the robot object (Figure 2). The user can understand what robot is currently being controlled.

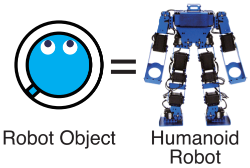

Figure 2: Robot Object

Action object

Action objects represent actual actions of the robot (Figure 3). Each set of action objects has a NULL object, which shows a fundamental condition of robots, which represents a standing pose of the humanoid robot. The NULL object has special mean of the visual language. It will be described in following section.

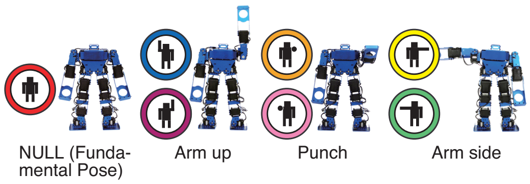

Figure 3: Set of action objects for humanoid robot

Connection semantics

Multiple action objects can be connected to a robot object. Connections follow certain connection rules.

Serial connection

All serially connected objects in one line are performed at the same time (Figure 4, left). For example, the combination of "right arm up" and "left arm up" object results in a "both arms up" motion, and "right up side" and "left arm side" results in both arms stretched out to the side.

Parallel connection

Parallel connection results in actions occurring in turn. For example, the top right image in Figure 4 shows the robot poses in turn. This motion consists of one right arm side object, punching with left arm object, and one NULL object. In this case, the program has two lines, one is a "NULL plus right arm side" action and the other is a "NULL plus punch with left arm" action. The NULL object plus another motion is useful for programming complex parallel connections.

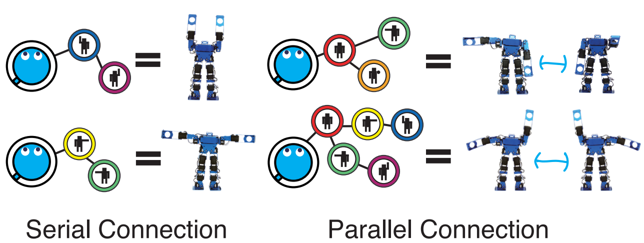

Figure 4: Connection semantics

Synchronization of robots

Just by connecting robot objects together, the robots' actions will synchronize. Figure 5 shows Robot B's action synchronizes with Robot A. When a robot synchronizes with another robot, the ring color of the connecting robot object will change to the color of the original's face.

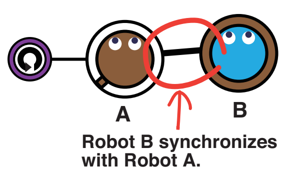

Figure 5: Example of Synchronization of Robots

Restrictions

There are some prohibited connections due to the hardware limitations of the robots and impossible semantic actions. When the user tries to make the prohibited connection, it will be rejected by the system. For example, action objects for the same arm, or the same actions cannot be connected on the same line, and the number of connection in one line is limited to three objects.

SYSTEM

The system consists of a multi-touch interface, robots, and a computer. The system produces music and beats and controls robots in rhythm to the music. The system uses a commercially available multi-touch interface and display, LCD-AD221FB-T by IO Data, for multi-user collaboration.

Robots

The system is designed for a small humanoid robot, which has a human-like body and can perform human-like actions[1]. The humanoid robot has 16 DOFs, which is enough to express realistic human-like behaviors.

Music and beat

One of the main features of *RoboJockey* is that robots automatically perform to music. The system has a clock for beat and music. The beat is currently set as 100 bpm, which means that the SeekBall moves, and actions are executed and switched 100 times per minute. The humanoid robot expresses the beat by moving its knees.

CONCLUSION AND FUTURE WORK

We developed *RoboJockey*, an interface for creating entertaining robot actions. The interface has a visual language that is simple but powerful enough for creating complex robot actions. We believe that *RoboJockey* could give the end-users, who are both experts and non-experts of robotics, a new entertainment experience with robots, and the feeling as "Robot Jockey."

REFERENCES

1. Gallardo, D., Julia, C.F. and Jordà, S. 2008. TurTan: A tangible programming language for creative exploration. In *Proc. of TABLETOP 2008*. 89-92.
2. Lego, Lego mindstorms NXT, http://mindstorms.lego.com/
3. Microsoft, Microsoft Robotics Developer Studio, http://msdn.microsoft.com/en-us/robotics/default.aspx
4. Takano, Y., Yamashita, N., Fujita, Y. 2007. Robot-Type Integrated User Interface Platform. *NEC Technical Journal*, Vol. 2. No.2. 20-23.
5. Taylor, S., Izadi, S., Kirk, D., Harper, R., and Garcia-Mendoza, A. 2009. Turning the tables: an interactive surface for vjing. In *Proc. of CHI '09*, ACM, 1251-1254.

[1] http://www.machinedesign.co.jp/tinywave.htm (Japanese)