Embodied Visual Programming for Robot Control

Stasinos Konstantopoulos
Institute of Informatics and Telecommunications
NCSR 'Demokritos'
Athens, Greece
konstant@iit.demokritos.gr

Andreas Lydakis
Institute of Informatics and Telecommunications
NCSR 'Demokritos'
Athens, Greece
lydakis@iit.demokritos.gr

Antonios-Emmanouil Gkikakis
Institute of Informatics and Telecommunications
NCSR 'Demokritos'
and Dept. of Informatics
Athens University of Economics and Business
p3090041@dias.aueb.gr

ABSTRACT

In this paper we motivate and present our *embodied visual programming* research plan, aiming at allowing end-users without any technical expertise to define complex robot behaviours. The core idea is to combine *visual programming* with the sensing and actuation capabilities of robots and with teleoperated demonstrations of what needs to be achieved.

Categories and Subject Descriptors

I.2.9 [**Robotics**]: Operator interfaces; D.1.7 [**Visual Programming**]: Visual Programming for HRI

General Terms

Design, Theory

1. INTRODUCTION

As robots increasingly leave the controlled and structured confines of research labs and factory floors, research into robots that can be commoditized is becoming relevant. One of the key factors is that people without any specialized expertise are able to control and instruct robots.

We envisage HRI technology that is far more versatile and dynamic than selecting among a predefined menu but at the same time is restricted and unambiguous (that natural language is not) and more natural and intuitive than developing software. However, where 'natural' and 'intuitive' in HRI are conventionally understood to equate to spoken interaction, we argue that people have been exposed and accustomed to a variety of different interfaces with machines: from cars' steering wheels, shift sticks and pedals, to gamepads and TV remotes there is a huge selection of ways to unambiguously and yet naturally and intuitively 'explain' to a machine what needs to be done without uttering a single word. Furthermore, we envisage taking full advantage of the opportunities offered by the robot's embodiment and interaction with physical space in order to ground human instructions into concrete robot actions.

We build upon *visual programming*, a programming paradigm based on graphically manipulating program elements. Arguably the best-known example in robotics is *NXT-G*® [3] for programming the educational and hobbyist *LEGO Mindstorms*® kit. NXT-G® is simple, but does not scale well as there is no support for intermediate levels of sensor analysis so that the programmer can refer to qualitative perception results. This makes diagrams too large and complex for most non-trivial tasks. The *STEM Toolkit* and its *12blocks* language [1] and *Microsoft RDS*® [4] also offer visual programming for a variety of platforms, including Mindstorms®, iRobot®, and other educational, hobbyist, and production systems, both physical and simulated. However, their visual programming languages suffer from similar limitations as NXT-G® and they revert to textual programming languages for more complex applications.

Emphasising embodiment and interaction with the robot's environment, the user interface of the Roomba® robotic vacuum cleaner comes closer to what we are trying to achieve: its autonomous mapping, localization, and space coverage strategies are complemented by a simple control interface based on beacons and virtual 'walls' to direct its actions [2]. This interface, however, is more of a configuration interface for pre-programmed functionalities rather than a programming interface, and lacks the flexibility that would allow different robot behaviours to be programmed.

2. RESEARCH PLAN

Our goal is to create an interface that will combine visual programming with the simplicity of pointing out specific locations in physical space, requires very little technical expertise, or none at all, and is based on *Robot Operating System (ROS)*, so that it can be easily deployed on many different platforms. In fact, we perceive robot actions as a matching of items from an inventory of *robot capabilities* against the *affordances* offered at different locations in the robot's environment, and robot control as sequences of such actions conditioned by perception. Our plan is to integrate:

- a library of generic, abstract capabilities (such as moving to a location on a map, grasping, turning a knob, etc.) which can be composed into complex strategies and adapted into concrete actions depending on the current situation and environment

HRI'14, March 3–6, 2014, Bielefeld, Germany.
ACM 978-1-4503-2658-2/14/03.
http://dx.doi.org/10.1145/2559636.2563719.

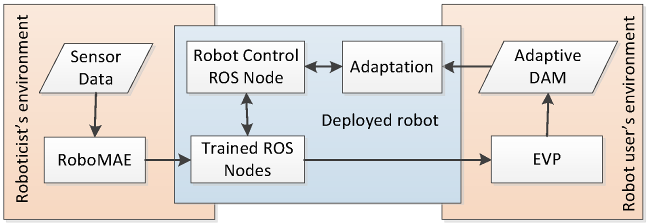

Figure 1: A graphic depiction of the interplay between development tools and end-users' tools.

- perception components for receiving external stimuli and for recognizing affordances where the different capabilities can be applied (a map of accessible locations, objects that can be gripped and lifted, knobs, etc.)
- a robot control environment following the visual programming paradigm for authoring visual representations of a system of actions and reactions to external stimuli; and that allows for new capabilities to be demonstrated and added to the library and new affordances to be pointed out to the robot via a combination of teleoperation and HRI

This new *embodied visual programming* paradigm augments visual programming with opportunities offered by the robot's embodiment and interaction with physical space: the ability to be instructed by being teleoperated and the ability to adapt by receiving external reinforcement stimuli. We put forward the development of *embodied visual programming* as a decisive step towards *consumer robotics*, along with long-term autonomy and the ability to self-test and recognize sub-system failures.

Figure 1 visualizes the interplay between robotic technology developers, platforms, and consumers: *RoboMAE*, the roboticist's workbench,[1] is used to develop components; the *embodied visual programming (EVP)* environment is used by the consumer to compose them into the desired behaviour, resulting in an *adaptive dialogue and action manager (DAM)*. Adaptivity, based on reinforcement learning or similar methods, allows for this specification to operate at a level of abstraction suitable for humans and be refined into the level of detail needed by software components.

3. CURRENT STATUS AND FUTURE STEPS

Besides the RoboMAE environment, we have also starting prototyping the interface at the end-user's side. At the current state of development, we have developed an Android app for controlling robot movements using both autonomous navigation (Figure 2) and teleoperation. Users can seamlessly switch between the two modes, issuing goals for the robot to reach and cancelling at any time to take over control. Users get real-time feedback on the mobile device's screen about the current location of the robot.

In order to facilitate communication between the android application and the robot, we have implemented a *ROS bridge* that passes ROS messages over a bluetooth or IP connection. The bridge is implemented as a ROS node on the robot side and as a library to be used by the user interface app on the Android side.

[1]Please see `http://roboskel.iit.demokritos.gr/Download/RoboMAE` for more details

Figure 2: The navigation panel, displaying the current (blue mark) and goal positions (red mark).

As future steps aim at both extending the movement control capabilities and extending the range of robot actions that can be controlled. With respect to the former, we are planning to implement the ability to record moves made using the teleoperation and to make these recording accessible for composing plans and behaviours that combine autonomous navigation with pre-recorded movements. This will be used in order to, for example, demonstrate to a robot how to navigate a narrow passage or how to best sweep the complete area of a room, and then define composite plans such as 'move into that room and sweep'.

With respect to the range of robot actions that can be controlled we are planning to extend the user interface so that the user can point out to the robot affordances in the environment, by identifying objects on the map or in visual perception and matching them against the robot's actuation capabilities. These will, again, be used as the building blocks for more complex composite plans, defined in a non-technical user interface following the *visual programming* paradigm.

4. ACKNOWLEDGEMENTS

The research described in this paper has been carried out at the Software and Knowledge Engineering Lab, in the context of *Roboskel*, the robotics activity of the Institute of Informatics and Telecommunications, NCSR "Demokritos". Please see `http://roboskel.iit.demokritos.gr` for more details.

5. REFERENCES

[1] Cogmation Robotics. *Everything you need to know about 12Blocks*, 2012. URL `http://virtualrobotgames.com/pages/12blocks`.

[2] iRobot. *Roomba 800 Users Manual*. URL `http://www.irobot.com/~/media/Files/Support/Home/Roomba/800/iRobot-Roomba-800-Manual.pdf`.

[3] J. F. Kelly. *LEGO MINDSTORMS NXT-G Programming Guide*. Apress Media LLC, New York, 2nd edition, 2010.

[4] Microsoft Corporation. *Microsoft Robotics Developer Studio 4*, 2012. URL `http://msdn.microsoft.com/en-us/library/dd939239.aspx`.