Natural user interface for lighting control: case study on desktop lighting using modular robots

Mehmet Mutlu[1,2], Stéphane Bonardi[1], Massimo Vespignani[1], Simon Hauser[1], Alexandre Bernardino[2] and Auke Jan Ijspeert[1]

***Abstract*— Roombots (RB) are self-reconfigurable modular robots designed to explore physical structure change by robotic reconfiguration and adaptive locomotion on structured grid environments or unstructured environments. The primary goal of RB is to create adaptive furniture. In this study, we propose a novel and user-friendly interface to control position and intensity of a mobile desk light using RB modules. In the proposed method, the user interacts with the RB with only hand/arm gestures. The user's arm is tracked with a single Kinect having bird's eye view. We demonstrate the effectiveness of the proposed interface in real hardware setup and discuss contributions of it.**

I. Introduction and Motivation

The world of robotics has evolved dramatically over the last decade. Robots have seen their capabilities increasing, both in terms of mechanics and electronics but also in terms of control. A growing number of robots are no more limited to lab spaces and are being designed to be integrated in every day life environments. They should provide services, help, and support to a wide variety of end-users, ranging from young children to elderly, all of them having specific needs. These robots appear in many shapes and orders of complexity, from the very advanced humanoid robots, such as Asimo [1], able to walk, run, and manipulate objects, to the simpler vacuum cleaner robot Roomba [2], limited to a specific task. But the complexity of these robots is often linked to their cost, which confines the most advanced ones to lab's environments. They are also often specialized into carrying out a specific set of tasks, such as manipulating objects or exploring unknown environments. More and more robots are being developed to support humans, such as the Keepon [3] robot, used for example as an helper therapy for autistic children, or the RI-MAN robots [4], designed to carry patients from their bed to their wheel chair. But these robots suffer from their high level of specialization into a specific domain and are lacking the ability to adapt to the task to be performed.

As opposed to this rise in complexity trend, the domain of reconfigurable modular robots has emerged as a potential solution. Reconfigurable modular robots are simple interchangeable units able to assemble to form a more complex structure to solve various more complicated tasks. Among them, Self-Reconfigurable Modular Robots (SRMRs) are equipped with active connection mechanisms allowing them to dynamically change shape to adapt to the user needs or to the task to be performed. They are different from more classical bio-inspired and anthropomorphic robots since they do not necessarily exhibit traits that would allow for an intuitive way of interaction (such has a head with cameras or hands with embedded tactile sensors).

The SRMR Roombots (RB) developed at the Biorobotics laboratory (EPFL, Switzerland) have been designed to study three major challenges: (i) When being configured in chain or lattice structures we use RB modules as a rapid prototyping set to study distributed locomotion control in unknown terrains. (ii) The self-reconfiguration (SR) capabilities of RB support the exploration of algorithms for self-organization, self-optimization and collaboration between modules. (iii) The name "Roombots" refers to our goal of creating self-reconfigurable adaptive furniture, i.e. furniture that can move and change shape thanks to reconfiguration using dynamic connection mechanisms. RB are made for building reconfigurable living and working environments that adapt to the current needs of human beings. We aim at a smart assistive environment where the robotic furniture is at the service of the user and helps with important aspects of his/her daily life. The hardware technology will be modular using a novel universal connector such as to allow flexible use, easy plug-and-play, and gradual implementation in the house. Different research aspects linked to the Roombots project are illustrated in Fig. 1.

The needs for a natural way of interacting with such robots is growing, especially if we envision to deploy them in everyday life environments, as it is the case in the RB project. When considering interaction inside homes or public spaces, we have to keep in mind that the proposed interaction solution should be non-intrusive but also easy to handle for non-experts or people with disabilities. An example scenario can be seen in Fig. 2.

We have developed several interaction strategies to control our modular robots Roombots. The first one is a low level communication targeted towards expert users in which we send motion command sequences. A second interface based on a classical GUI offers to the user the ability to build pieces of furniture made of RB modules in a 3D environment. It requires the user to focus on a computer display using a

[1]M. Mutlu, M. Vespignani, S. Hauser, S. Bonardi and A. J. Ijspeert are with Biorobotics Laboratory, School of Engineering and Institute of Bioengineering, École Polytechnique Fédérale de Lausanne, CH-1015 Lausanne, Switzerland mehmet.mutlu@ieee.org, massimo.vespignani@epfl.ch, simon.hauser@epfl.ch, stephane.bonardi@epfl.ch, auke.ijspeert@epfl.ch

[2]M. Mutlu and A. Bernardino are with Electrical and Computer Engineering of the Faculty of Engineering at IST, Instituto Superior Téchnico, 1049-001 Lisbon, Portugal mehmet.mutlu@ieee.org, alex@isr.ist.utl.pt

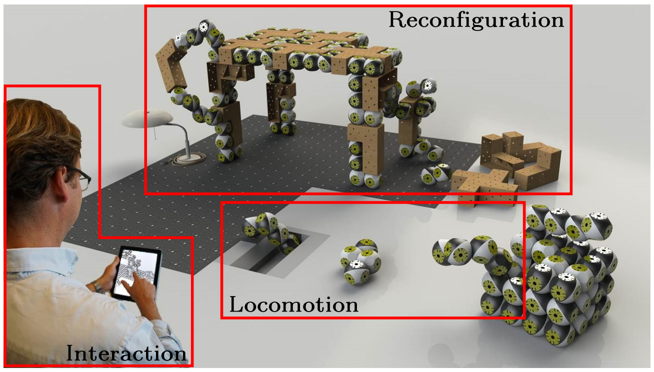


Fig. 1. Rendered picture representing the different aspects of the Roombots project. On this illustration, a table is being constructed out of active modules and passives elements (wooden color) evolving on a 2D grid (in dark grey). A set of modules is located out of the grid and metamodules separate from the main group to perform off-grid locomotion. They reattached to the grid using a passive alignment mechanism included in the ground. A user is controlling the process using a tablet device.

Fig. 2. Roombots are being designed to work in daily life environments. A rendered image that shows an example.

keyboard and a mouse for to remotely interact with robotic units situated elsewhere. To enhance the user experience, we designed an Augmented Reality application based on a tablet display in which the user could place robots into the environment and freely move them around [5]. The main limitation of this approach was the necessity for the user to carry an external device at all time. To alleviate the need for an extra device we developed a gesture based interface where the user is able to select individual RB modules and move them to target locations by pointing at the modules and at the targets, receiving visual feedbacks on LED setups on both the grid tiles composing the environment and the RB modules [6]. In this paper, we proposed a new way of interacting with these non-anthropomorphic platforms, in which the end-user was placed at the center of the interaction by abstracting away the complexity of the control techniques inherent to SRMRs.

Motivated by the existing natural control interface studies [7]–[11], we decided to complement our approach to take into account the versatility of our modular system and the re-usability of its components. We propose in this paper a novel tracking method allowing the user to interact with a set of modular robots using hand movements. We illustrate our approach with a main usage scenario using the modularity brought by the modular robotic platforms. We created a multi-directional spotlight made of two Roombots modules which is able to control light's direction and intensity based on the user arm/hand movement on a table top.

This paper is organized as follows. In a first section, we describe the hardware setup and software design of our interface. We then present our case studies and the related experiments and results we carried out. Finally, we summarize our study and give potential future extension plans in the last section.

II. System Design

A single RB module can perform locomotion on a structured grid environment. When multiple of RB come together, they can attach to each other and/or do more complex tasks by collaborating with other modules in the environment. For instance, a single module can be used as a 3 degree of freedom (DOF) manipulator when it is attached to the grid with its active connection mechanism. A second module can attach to the end of the first one. The resulting structure consisting of two RB modules in series is called metamodule. A metamodule can be used as a 6DOF manipulator on the grid or it can do controlled locomotion on or off the grid. A particular advantage of RB is their flexibility of use. They can potentially form structures such as furniture, carry existing objects or manipulate surrounding environment whenever possible. The focus of this work is introducing an easy to use human-robot interface for controlling Roombots to illuminate desired location at controlled light intensity.

Light is the essential part of almost all visual systems. Human and machine vision is only possible with existence of either radiated or reflected light. Hence, proper lighting systems are entangled with our daily lives. Although most of the lighting systems are stationary and only on/off controlled, there exists many variations such as light intensity and color controlled ones or mobile lights as commonly seen on theatre stages [12] to increase comfort, give focus to certain areas or just for entertainment. Another need for local lighting arises on desktops. Small and adjustable desktop lights are commonly used while studying or working on a table. User needs to manually adjust position of the light and the table lamp occupies small space on the table. There are also intelligent and robotics solutions in the literature. A lamp design that uses a higher level of cognitive architecture to achieve fluency in human robot cooperation is presented in [13]. Our proposition is using more generic-use RB modules instead of table lamps since RB are mainly designed for autonomously creating furniture. Manipulation capabilities of RB for self reconfiguration purposes [14] and some native interfaces for controlling robot locomotion [6] were studied in previous works. Therefore, we assume a robot can carry the light to the the desired location. In this paper, the focus is on human robot interaction once the RB are on the location where light is needed.

A. Overview of the Proposed Human Robot Interface

The proposed system consists of a depth camera, light equipped RB metamodule and a computer. The depth camera can be any camera giving a depth information of image pixels. It is also possible to use stereo cameras, but, in this study Kinect is used since it is widely available and relatively cheap. One of the key requirements is having a mobile and remotely controllable light. Even though in this work RB are used to actuate the light, a mechanically simpler pan and tilt system [15] [16] can also be used as light actuator. Finally, a computer is needed to implement our autonomous lighting control interface. Fig. 3 illustrates all components of the proposed system and Fig. 4 shows the real implementation of the experimental setup. The Kinect is mounted on the ceiling to have bird's eye view of the desktop. It captures both depth and color images and transfers them to PC over USB. Depth images are used to extract the desired light location by detecting the position of the user's arm. The user lifts the arm over certain height to activate the *light position control mode*. The end point of the hand is assumed to be the desired light location. Then, it is geometrically possible to calculate desired pose of the light to enlighten the desired spot, if the coordinates of Kinect and light are known. Once the desired pose of the light is known, robot is commanded to bring the light to the calculated pose. The communication between PC and RB is wireless by using Bluetooth. Finally the robot directs the light to the calculated orientation. If the control of the robot is rigid enough and extrinsic calibration of robot and Kinect is done accurately, there is no need for color images. However, color is used as a feedback mechanism to ensure the desired location is really illuminated.

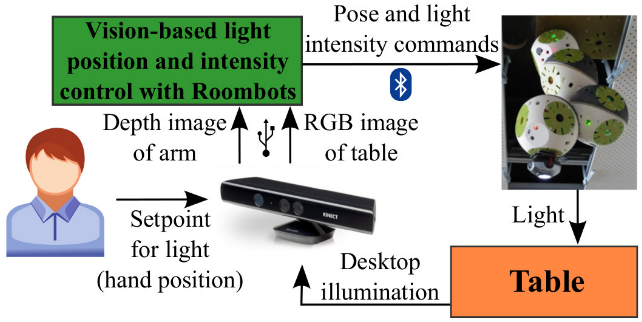
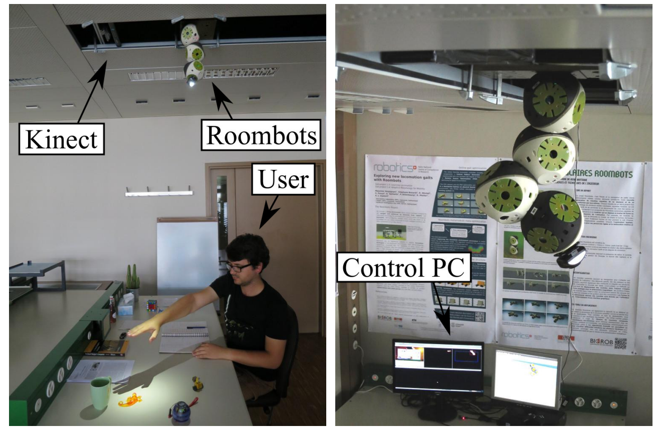


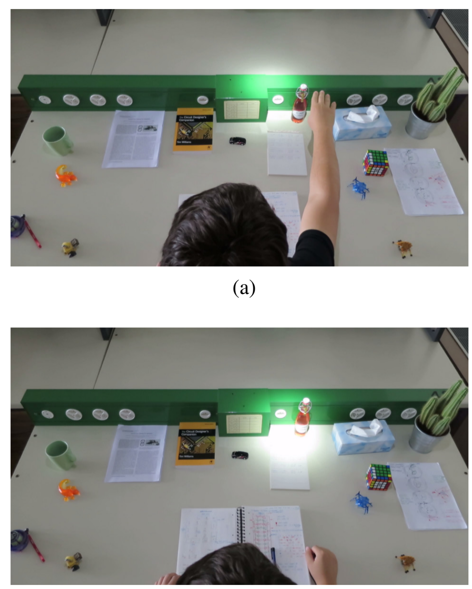
Fig. 3. Overview of proposed user interface. The key components and relations between them are shown.

Using gestures for controlling hardware [6] or software [17] can be one of the most intuitive control ways depending on the implementation. When a table environment is considered, a person would need to first reach the lamp and then direct the light towards a desired direction. However, with the proposed interface, simply showing the target spot is enough. The interface protocol is quite intuitive. The user needs to raise his/her arm a little upper than the usual working height and reach to the place where light is needed (Fig. 5a). RB will immediately respond to the request by directing the light to the set direction. The spotlight will follow the end of the arm as long as the arm stays up, in other words in the position set mode. System continues to light the final set point when the arm is lowered to usual working height as it can be seen in Fig. 5b.



(a) (b)

Fig. 4. Experimental robotic spotlight system consisting of a Roombots metamodule, Kinect (a) and computer(b).



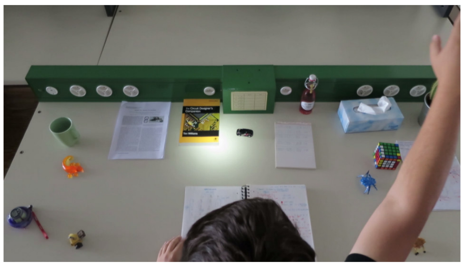
(a)

(b)

Fig. 5. User is pointing where the light is needed(a). Once the pose of Roombots spotlight is set, it remains on the desired location (b) as long as the user does not switch to the control mode.

The second mode of the system is adjusting the brightness of the light. It is similar to setting the position. First, the user needs to raise the arm further up. When the arm is high enough, the system switches into *light brightness adjustment mode*. In this mode, the robot stops position tracking mode and stays in the final set point. At this point the user controls a virtual slider that regulates the brightness. Moving the hand towards right results in dimmer and eventually turned off the light. Similarly, moving hand toward the left side means brighter illumination as illustrated in Fig. 6. Once the appropriate level of brightness is set, the user can lower hand vertically and switch back to the *light position setting mode*.

(a)



(b)

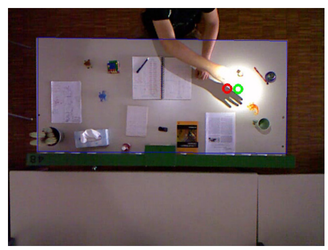
Fig. 6. Adjusting brightness of the light. Raising the arm further up puts the system on intensity setting mode. For higher or lower illumination hand should be on the left (a) or right (b) of the table respectively.

A computer is needed to process the Kinect data to get input from the user and to control the robot. However, the user does not need to interact with the PC which is hidden from the user. The illustrations given in Fig. 7 show how the automated control part works.

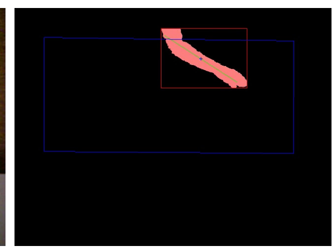
B. Detecting User's Commands

The proposed interface is designed to have minimal learning for the user before starting to use the system. The fundamental information required to accomplish the task is detecting the arm, finding its end point and measuring the horizontal position and height of the end point. This information can be extracted using a depth image of the table captured from above. The depth images captured by Kinect give the distance of each pixel to the camera in millimeters. Hence, the obtained depth image is a two dimensional distance matrix of pixels in the field of view of the camera. The first processing step consists in converting the depth image to binary image by thresholding within position-set-mode height. If there are no objects higher than approximately shoulder height, the obtained binary image would show only the user arm when it is raised to set the light position. The implemented case, shown in Fig. 4a, is an example of such scenario which simplifies the computer vision steps. If there are objects higher than the minimum range of light-position-set mode, those objects needs to be filtered out from the depth image. Using methods such as background subtraction [18]. The next step is obtaining meaningful regions in the binary image. This step requires connected component analysis on the image, also known as blob detection. Ideally, the binary image should have only a single blob which is the arm of the user. However, due to noise on the Kinect, many other small blobs can be observed on the camera. Moreover, noise can separate the arm blob into multiple blobs. In order to get rid of noise, closing operation (dilation followed by erosion) is applied on binary image and only the biggest blob is taken into account provided that it has minimum size. All the other noise generated blobs are discarded. In the end of the depth image processing, only the arm blob is obtained as seen in Fig. 7b. Once the arm is extracted, the tip point of it, the furthers point from user, is considered to be the target set-point for spotlight. The target point corresponding to Fig. 7b is marked as red circle on Fig. 7a.

Knowing the target position on image plane is sufficient to calculate the spotlight's desired orientation such that its light will illuminate the target point. If the intrinsic calibration of depth camera is done properly, x_{ti} and y_{ti} coordinates of the target on image plane can be mapped to actual x_{tw} and y_{tw} coordinates in the world coordinate frame. Also note that depth image gives the third coordinate, z_{tw}.

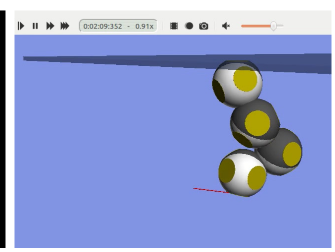


(a)



(b)

(c)



(d)

Fig. 7. The application GUI gives real time information about the detection and control. Although the user does not need to interact with the PC, this information is still useful to explain the system. The Operator can see (a) color images, (b) processed depth and (c) RGB images and (d) simulation view on the application GUI.

The detection of the brightness adjustment mode is almost the same as the position target setting mode. If the height of the target point is higher than the predetermined threshold, the system goes into the light intensity control mode. In this mode the lateral position of the arm tip point is mapped to the brightness of the light.

C. Controlling Roombots

Knowing the target position on the table is enough to calculate the desired orientation of the spotlight. But still Roombots need to have a low level controller for bringing the light to the desired orientation. As a design choice, the position of the end effector of the RB metamodule is kept constant at a certain point and the light orientation is changed around the pre-determined operating point. Thus, the resulting overall motion of the light is only rotation which also simplifies the control for the operator. Rotation around each axis is called pan, tilt, and roll motion. Ideally, roll has no observable result since the light going out of the source forms a conic shape. However, pan and tilt results in displacement of illuminated region on the desk. Even though only pan and tilt motions are required, achieving it is not trivial with RB metamodule.

Attaching two Roombots modules to each other in series results in a 6-DOF serial manipulator (Roombots metamodule). Once the Roombots metamodule fixes itself on a stationary surface, the forward kinematic model of the system can be written as follows,

$$ {}^{b}_{e}T = \left(\prod_{i=1}^{6} {}^{i-1}_{i}T \right) {}^{6}_{e}T \tag{1} $$

where ${}^{b}_{e}T$ denotes the transformation from base (${}^{0}T = {}^{b}T$) to the end effector, ${}^{i-1}_{i}T$ adds one rotation variable for each DOF, and ${}^{6}_{e}T$ is the constant transform to reach the end effectors center which is the center of light source in this study. For each desired light orientation (i.e. end effector orientation) the inverse kinematics of the metamodule are numerically solved. The solution of inverse kinematic gives the rotation angles, Q_i, that each joint should reach. Desired light orientation can be achieved after knowing all Q_i values, . However, inverse kinematics may not always have a solution or may have multiple solutions. In the multiple solution case, the solution which is closest to the previous states is used. Cases with no solution should be treated carefully or should be avoided. In order to simplify the control problem, continuous space is discretized with resolution of 0.5 degree rotation angles. When the light target orientation is calculated, it is assigned to the closest available discrete angle. Although discretization results in slight errors, the resolution is sufficient for human perception. Since the roll rotation is a free parameter, it provides some flexibility to find valid inverse kinematics solutions. When inverse kinematics could not be found in the first try, the roll angle is changed to find the solution for desired pan and tilt angles. In the end we were able to calculate inverse kinematics in real time for all possible pan and tilt angle combinations within our operation range and resolution. Each solution is tested in Webots simulation environment [19] to validate the results in real time during the operation. The Webots screenshot corresponding to time instance for Fig. 7 is shown in Fig. 7d.

D. Closing the Control Loop

When ideal conditions are assumed, open loop control of the RB manipulators is sufficient. However, there are many noise sources in the real world. Extrinsic calibration between RB and Kinect can be done but, slight structural misalignments are unavoidable. Furthermore, there are uncertainties arising from RB hardware such as gearbox backlash and body elasticities. Those uncertainties cause the open-loop-illuminated region to have stochastic position errors. One

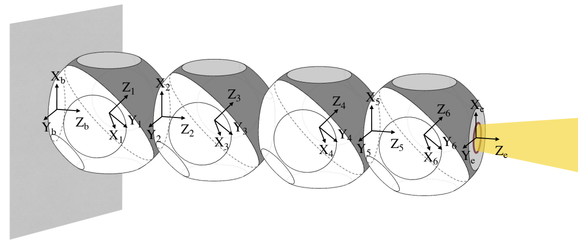


Fig. 8. Kinematic model of RB metamodule manipulator consisting of two RB modules. Each module (two spheres) has three DOF with continuous rotation capability.

solution to reduce the errors for different scenarios could be using simple, yet precise pan and tilt mechanism or using a rigid manipulator such as industrial ones. However, we would like to implement our proposed interface for RB and it is still possible to compensate most of the noise by implementing active control.

For the closed loop control, feedback from the environment is needed. It is obtained by processing the RGB images captured by Kinect. RGB images are first converted to greyscale and then thresholded to get the brightest region in the binary image. To avoid noise, a similar blob filtering technique to the one explained for depth images is implemented on binary image. An example of processed RGB image can be seen in Fig. 7c. The highlighted region shows the blob corresponding to the brightest region. We assume that the illuminated region appears to be the brightest part in the image. Finally, the center of that blob is found by calculating,

$$ S_x = \frac{1}{n} \sum_{n=1}^{n} x_i, \quad S_y = \frac{1}{n} \sum_{n=1}^{n} y_i \tag{2} $$

where S_x and S_y are x and y coordinates of the first moment of the area, n is the number of pixels in that region and x_i and y_i are image coordinates of blob pixels. The first moment is also known as center of mass of the area in an image. The enter of mass of that blob is assumed to be the center of the illuminated region.

Knowing the set point and the resulting light location, a conventional PID controller can be implemented to compensate errors. Only P control is implemented to have a simpler system, i.e.

$$ u = K_p e \tag{3} $$

where u is the input to the plant that is error compensation signal, K_p is the proportional control constant and e is the error. Both u and e are two dimensional vectors since the position control of the light is in the (x,y) plane. Thus, the error signal can be written as

$$ \begin{pmatrix} e_x \\ e_y \end{pmatrix} = \begin{pmatrix} x_t - x_l \\ y_t - y_l \end{pmatrix} \tag{4} $$

where subscripts t and l corresponds to target and actual light positions respectively. The control signal u tries to bring the actual light closer to the target point. In order to avoid overreactions to an unexpected situation u is passed through a saturation function to limit the correction signal's effect.

III. RESULTS AND DISCUSSION

Proposed system has been implemented in hardware and subjective tests are conducted to evaluate the system. A video demonstrating the performance of the whole system can be found among the media submission of this paper.

Another interesting aspect of the interface is the fact that light position gives a direct visual feedback to the user. In Fig. 3, we can observe one more implicit feedback loop. Since the user can directly see the location of the light, in case of an unexpected situation when the light does not reach the target position for some reason, the user can intervene the control process by giving a proxy target such that the light would end up the location the user wants. This situation actually means human in the loop control.

IV. CONCLUSIONS AND FUTURE WORK

In this study we presented an intuitive user interface to control a robotic spotlight which is actuated by a RB metamodule. We integrated a Kinect based input device to detect the arm of the user that is used to set the desired spot and brightness for illumination. The manipulation of the light is done in continuous space and in real time. Finally, we incorporated visual feedback, to improve tracking performance with closed loop control.

The main contributions of this work are twofold: (1) a natural and contactless interaction designed for autonomous and mobile lighting and (2) Roombots gained a new functionality and behaviour which is bringing the light to most needed spots. The proposed lighting interaction is designed to make light position and intensity control as intuitive as possible. The interface can potentially be useful for people who are doing a task during which they need proper local lighting and cannot manipulate a lamp to redirect the light, e.g. cooking, painting, surgery, etc. Although, light is actuated with a complex robot, the natural human robot interaction made the control of the lighting quite user friendly.

The light used in this work was a commercially available LED with a reflective cone. As a future work, we will design the light system from scratch to fit everything it inside RB module. Additionally, backlash in custom made gearboxes of RB has been identified as the biggest source of uncertainty for precise control of RB. We will revise the design and try to reduce the backlash as much as possible.

Another extension to this study will focus on implementing a similar interface for controlling an end effector of RB manipulator with teleoperation [20]. RB manipulator can be used to locate objects, even if the system is physically in a different place. For instance, one can help a disabled person who has to stay at home to reach and fetch items using Roombots from somewhere else in the world. In such extensions, improvements to the feedback loop will be needed.

V. ACKNOWLEDGMENTS

This work was supported by the Swiss National Science Foundation through the National Centre of Competence in Research (NCCR) Robotics (Project 153299) and Fundação para a Ciência e Tecnologia (FCT) agency of Ministry for Education and Science of Portugal under the contract number PD/BD/105781/2014 and project grant AHA-CMUP-ERI/HCI/0046/2013.

REFERENCES

[1] M. Hirose, “Development of humanoid robot asimo,” in *Proc. IEEE/RSJ Int. Conference on Intelligent Robots and Systems (Oct. 29, 2001)*, 2001.

[2] iRobot, “http://www.irobot.com/us/,” robots designer and builder. [Online]. Available: http://www.irobot.com/us/

[3] H. Kozima, M. P. Michalowski, and C. Nakagawa, “Keepon,” *International Journal of Social Robotics*, vol. 1, no. 1, pp. 3–18, 2009.

[4] T. Odashima, M. Onishi, K. Tahara, K. Takagi, F. Asano, Y. Kato, H. Nakashima, Y. Kobayashi, T. Mukai, Z. Luo *et al.*, “A soft human-interactive robot ri-man,” in *Intelligent Robots and Systems, 2006 IEEE/RSJ International Conference on*. IEEE, 2006, pp. 1–1.

[5] S. Bonardi, J. Blatter, J. Fink, R. Moeckel, P. Jermann, P. Dillenbourg, and A. J. Ijspeert, “Design and Evaluation of a Graphical iPad Application for Arranging Adaptive Furniture,” in *RO-MAN*, 2012, pp. 290–297.

[6] A. Özgür, S. Bonardi, M. Vespignani, R. Möckel, and A. J. Ijspeert, “Natural user interface for roombots,” in *RO-MAN, 2014 IEEE*. IEEE, 2014.

[7] M. Lichtenstern, M. Frassl, B. Perun, and M. Angermann, “A Prototyping Environment for Interaction Between a Human and a Robotic Multi-Agent System,” in *Human-Robot Interaction (HRI), 2012 7th ACM/IEEE International Conference on*, 2012, pp. 185–186.

[8] J. McLurkin, J. Smith, J. Frankel, D. Sotkowitz, D. Blau, and B. Schmidt, “Speaking Swarmish: Human-Robot Interface Design for Large Swarms of Autonomous Mobile Robots,” in *AAAI Spring Symposium: To Boldly Go Where No Human-Robot Team Has Gone Before*, 2006, pp. 72–75.

[9] C. Bellmore, R. Ptucha, and A. Savakis, “Interactive Display Using Depth and RGB Sensors for Face and Gesture Control,” in *Image Processing Workshop (WNYIPW), 2011 IEEE Western New York*. IEEE, 2011, pp. 1–4.

[10] F. Klompmaker, K. Nebe, and A. Fast, “dSensingNI: a Framework for Advanced Tangible Interaction Using a Depth Camera,” in *Tangible and Embedded Interaction 2012, 6th International Conference on*, 2012, pp. 217–224.

[11] S.-t. Noh, S. Hashimoto, D. Yamanaka, Y. Kamiyama, M. Inami, and T. Igarashi, “Design and enhancement of painting interface for room lights,” *The Visual Computer*, vol. 30, no. 5, pp. 467–478, 2014.

[12] W. Sugden, “Computer controlled stage lighting system,” Apr. 11 1995, uS Patent 5,406,176. [Online]. Available: https://www.google.com/patents/US5406176

[13] G. Hoffman and C. Breazeal, “Achieving Fluency Through Perceptual-symbol Practice in Human-robot Collaboration,” in *Proceedings of the 3rd ACM/IEEE International Conference on Human Robot Interaction*, ser. HRI ’08. New York, NY, USA: ACM, 2008, pp. 1–8.

[14] A. Spröwitz, S. Pouya, S. Bonardi, J. van den Kieboom, R. Möckel, A. Billard, P. Dillenbourg, and A. Ijspeert, “Roombots: Reconfigurable Robots for Adaptive Furniture,” *Computational Intelligence Magazine, IEEE*, vol. 5, no. 3, pp. 20–32, 2010.

[15] A. D. Wilson, H. Benko, S. Izadi, and O. Hilliges, “Steerable augmented reality with the beamatron,” in *In UIST12, ACM Press*, 2012, pp. 413–422.

[16] M. Winkler, K. Hover, A. Hadjakos, and M. Muhlhauser, “Automatic camera control for tracking a presenter during a talk,” in *Multimedia (ISM), 2012 IEEE International Symposium on*, Dec 2012, pp. 471–476.

[17] M. N. K. Boulos, B. J. Blanchard, C. Walker, J. Montero, A. Tripathy, R. Gutierrez-Osuna *et al.*, “Web gis in practice x: a microsoft kinect natural user interface for google earth navigation,” *International journal of health geographics*, vol. 10, no. 1, p. 45, 2011.

[18] M. Piccardi, “Background subtraction techniques: a review,” in *Systems, Man and Cybernetics, 2004 IEEE International Conference on*, vol. 4, Oct 2004, pp. 3099–3104 vol.4.

[19] Webots, “http://www.cyberbotics.com,” commercial Mobile Robot Simulation Software. [Online]. Available: http://www.cyberbotics.com

[20] J. Vertut, *Teleoperation and robotics: applications and technology*. Springer Science & Business Media, 2013, vol. 3.