Using Compliant Robots as Projective Interfaces in Dynamic Environments

Davide De Tommaso, Sylvain Calinon, and Darwin G. Caldwell

Department of Advanced Robotics, Istituto Italiano di Tecnologia,
via Morego, 30, 16163 Genova
{davide.detommaso,sylvain.calinon,darwin.caldwell}@iit.it

Abstract. We present a human-robot interface for projecting information on arbitrary planar surfaces by sharing a visual understanding of the workspace. A compliant 7-DOF arm robot endowed with a pico-projector and a depth sensor has been used for the experiment. The perceptual capabilities allows the system to detect geometry features of the environment which are used for superimposing undistorted projection on planar surfaces. The proposed scenario consists of a first phase in which the user physically interacts with the gravity compensated robot for choosing the place where the projection will appear. After, in the second phase, the robotic arm is able to autonomously superimpose visual information in the selected area and actively adapt to perturbations. We also present a proof-of-concept for managing occlusions and tracking the position of the projection whenever obstacles enter in the projection field.

1 Introduction

Portable projecting devices open a new trend in Human-Robot Interaction (HRI) by offering a new communication medium, that complements the use of natural language processing and haptic feedback, as tool for robots to provide information to humans. While gathering inputs from the surrounding environment is a central topic of robotic research since decades, allowing robots to exploit this environment to communicate with the user is still a largely unexplored issue in the HRI field. Humanoids able of emulating facial emotions or body gestures are just few examples of making machines anthropomorphically communicative towards humans. Non-verbal communication can improve the efficiency and robustness of the interaction between humans and robots involved in collaboration tasks by ameliorating the overall task performance, see e.g. [1]. Virtual and Augmented Reality interfaces have been developed in this direction to let the robot communicate information to the user without requiring to rely on standard computer screen interface, see e.g. [2]. Robots with projection capabilities represent an important area of interest for sharing information about the environment or about a task to achieve with the physical objects and tools surrounding the robot. The use of light beams superimposed on physical objects might allow the user to focus his/her gaze on specific areas of the work space. Moreover, colors, position marks, trajectories and texts may provide additional information about task and

S.S. Ge et al. (Eds.): ICSR 2012, LNAI 7621, pp. 338–347, 2012.

relevant locations. While different application domains have been explored by using projective technology in Human-Computer Interaction (HCI), such as entertainment [3], training [4,5], elderly assistance [6] and surgery [7], the use of such devices in the HRI community remains largely unexplored so far.

In this paper, we present a robotic interface to assist users in dynamic environments, allowing them to visualize digital information in any planar surface. Working spaces like factories, classrooms, homes, training rooms, hospitals are possible target locations for tasks focusing more on the visual and haptics feedbacks of the users. For example users involved in manual tasks have not the possibility to handle any computing devices for accessing digital information. In such cases robot interfaces might help at providing human assistance services. For tasks requiring a worker to move and change posture, it remains difficult to predefine a generic positioning of the sensing and projecting device that would work well for all the tasks and situations. The use of the robotic arm allows a flexible repositioning of the sensing and projection devices that automatically adapts to the situation and task constraints.

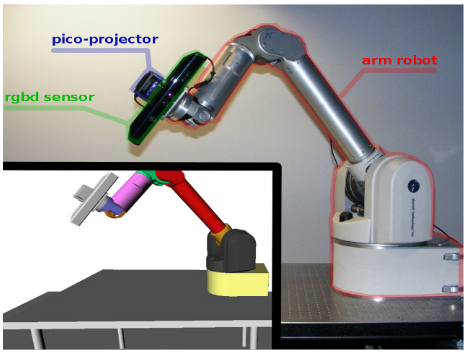


Fig. 1. The projection system using the compliant Barrett WAM 7-DOF robot endowed with a Kinect device and a pico-projector

The development of the proposed interface was mainly driven by two goals. First, to provide the user with the possibility of choosing and modifying the position of the projection, by intuitively moving the robot by hand when it is gravity compensated. Second, to make the interface accessible to the user, by maintaining its geometric projection features fixed, even in presence of perturbations caused by the users or by the robot changing its projection posture. We present a prototype following these two guidelines, by using a 7-DOF compliant arm robot equipped with a pico-projector and a Microsoft Kinect device (see Fig. 1).

2 Related Work

The recent advances of projector technology have grown interest for enabling new user friendly communication interface. Here we present two previous works, from which we draw inspiration.

Vogel *et al* in [8] propose a spatial augmented reality interface able to establish a physical safety area in a shared workspace between users and robots, by using a camera and projector fixed to the ceiling. The projective device gives feedback to the user about the safe working area, by projecting virtual barriers directly aligned with the real portion of space. The perception device helps the system to actively monitor the physical state of the user and the robot within the safety area, by changing position, shape and orientation of the projected image dynamically.

In the field of wearable computing, Harrison *et al* suggest in [9] an innovative way to access digital information everywhere. The developed *OmniTouch* system is a wearable device that enables the user to interact with a GUI projected on any physical surface by using gestures. By exploiting the perception capabilities of a depth sensor, the system is able to detect suitable surfaces in which to project the GUI, by using a pico-projector, and to interact with fingers like a mouse pointer.

The approach we discuss in the paper does not require a fixed setup, a structured environment, or tag-based surfaces for finding the place of projection. In fact the use of the Kinect, rigidly attached to the robot, allows the system to detect the geometric properties of planar surfaces in the surrounding environment. In such a way, by using a fully calibrated system, undistorted images can be projected in planes chosen by the user and at the same time the space around the projection can be monitored for detecting obstacles and occlusions.

3 System Overview

3.1 Experimental Setup

To the best of our knowledge, the proposed interface represents the first attempt of developing an interactive projective interface using an actively compliant robot.

We exploit here such manipulator as an interface that can move, perceive and project in its environment. These features and its light weight well fit with the requirements of human-robot collaboration scenarios, in which the physical contact between the user and the manipulator represents an important modality of interaction.

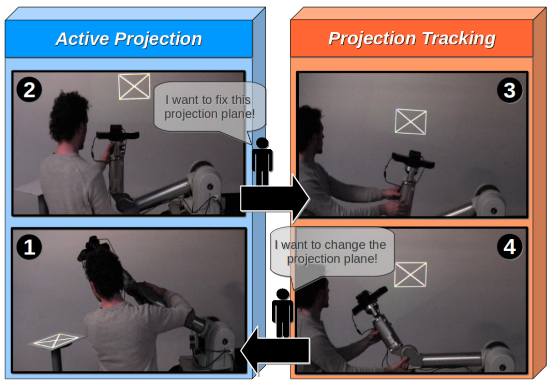


Fig. 2. The two operating modes of the system

Our experimental setup consists of a compliant Barrett WAM 7-DOF arm robot endowed with a plastic support mounted at the end-effector holding rigidly a Microsoft Kinect and an AXAA laser pico-projector. The Kinect has been extensively exploited in different fields of research as depth sensor, introducing an affordable option for point cloud tracking and detection [10].

For the projection capability, we selected a pico-projector because it is small enough to be mounted on top of the Barrett WAM and its laser technology allows us to project at any distance without requiring to adjust focus. In the experiment the robot and the user share the same working space. Adopting such a mobile configuration, instead of a fixed setup, leads two key advantages: 1) an extended field of view due to the different viewpoints reachable by the robotic arm and 2) the possibility to actively handle occlusions and facilitate tracking of task-relevant features. For selecting appropriate projecting surfaces, we decided to exploit both the control capabilities of the robot and the perception capability of the depth sensor. Instead of using structured environments, the system actively projects distortion-free images on the basis of the geometry of the projection surface, detected by the Kinect. Accordingly, whenever the user wants to select the position and orientation of the projected display, she/he just needs to manually move the robotic arm in an appropriate position, while the robot compensates for the weight of its arm and friction in its joints. For continuously tracking the projection, the system autonomously reacts to the changes of the robot arm configuration by 1) modifying the orientation of the end-effector, and 2) recomputing the perspective of the projected image. During this projection phase, the system is also able to simultaneously perceive three dimensional information about the area between the end-effector and the surface of projection. Indeed, the Kinect's field of view is larger than the pico-projector so that the entire frustum of projection can be monitored to detect obstacles causing occlusions.

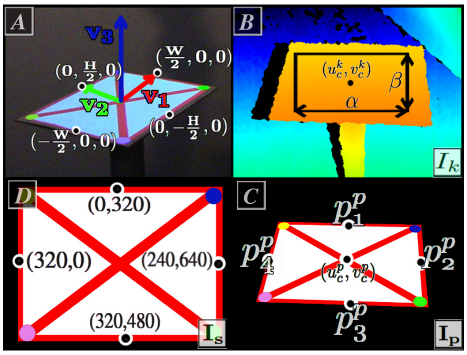


Fig. 3. (A) The frame of reference of the projection is defined by the orthogonal vectors $\mathbf{v_1}$, $\mathbf{v_2}$, $\mathbf{v_3}$, whose origins correspond to the center of the projected image. (B) The Kinect's depthmap is used for estimating the geometric equation of the plane chosen for the projection. *Principal Component Analysis* is applied to points samples from a rectangular area of $\mathbf{I_k}$. (C) The input image of the pico-projector is the 640×480 RGB matrix $\mathbf{I_p}$ representing the result of the perspective transformation for fitting the image in the projection plane. (D) The 640×480 RGB matrix $\mathbf{I_s}$ represents the source image to project.

4 Developed Prototype

The system involves two mutually exclusive operating modes: 1) *Active Projection* and 2) *Projection Tracking* (see Fig. 2).

4.1 Active Projection

For allowing the user to select the projection plane, the system starts with the *Active Projection* mode. In this phase the process of warping the projection is carried out by using jointly the Kinect and the pico-projector, while the robot is only controlled by compensating for the gravity. This enables the user to easily change the joints configuration of the robot by looking for the pose of the end-effector allowing the projection in the desired plane. According to the end-effector pose, the source image $\mathbf{I_s}$, in Fig. 3-D, is warped in the image to project $\mathbf{I_p}$ in Fig. 3-C by using the perspective transformation

$$\underbrace{\begin{bmatrix} wp_x \\ wp_y \\ w \end{bmatrix}}_{\mathbf{I}_p(x,y)} = \underbrace{\begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix}}_{\mathbf{H}_\pi} \underbrace{\begin{bmatrix} s_x \\ s_y \\ 1 \end{bmatrix}}_{\mathbf{I}_s(x,y)} . \quad (1)$$

Since the 3D points of the projection surface lie on the same plane, the views of the Kinect and the pico-projector are related by an homography. For estimating the homography matrix $\mathbf{H}_\pi$, two sets of four 2D points are required, four points in the source image $\mathbf{I_s}$ (e.g the matrix elements (0,320), (240,640), (320,480), (320,0) see Fig. 3-D) and four points $p_1^p, ..p_4^p$ in the destination image $\mathbf{I_p}$ (see Fig. 3-C). Such points in $\mathbf{I_p}$ correspond to the 3D points $P_1^k, ..P_4^k$ in the Kinect's frame, and can be found by changing the coordinate system from the Kinect to the pico-projector. The four corners of the projection are automatically detected by following the geometry of the planar surface and can be found by

$$\begin{bmatrix} P_1^k \\ P_2^k \\ P_3^k \\ P_4^k \end{bmatrix} = \mathbf{T} \begin{bmatrix} 0 & -\frac{W}{2} & 0 & \frac{W}{2} \\ \frac{H}{2} & 0 & -\frac{H}{2} & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 \end{bmatrix} , \quad (2)$$

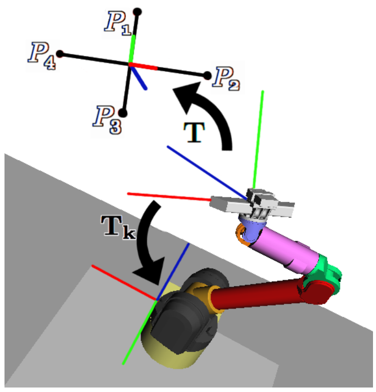


Fig. 4. A preliminary calibration process between the frames of reference of the robot, the pico-projector and the Kinect enables the system to find the corresponding transformation matrices

see Fig. 3-A. The 4×4 matrix $\mathbf{T}$ is the transformation between the Kinect and the projection frames (see Fig. 4), which can be written as

$$\mathbf{T} = \begin{bmatrix} \mathbf{R} & C^k \\ 0 & 1 \end{bmatrix}, \mathbf{R} = \begin{bmatrix} \mathbf{v_1} & \mathbf{v_2} & \mathbf{v_3}, \end{bmatrix} \quad (3)$$

where the equation of the projecting plane is

$$\pi : \quad \mathbf{r} = \mathbf{r_0} + s\mathbf{v_1} + t\mathbf{v_2}. \quad (4)$$

The geometry of the plane, namely the vectors $\mathbf{v_1}$ and $\mathbf{v_2}$ (see Fig. 3-A), can be estimated by *Principal Component Analysis (PCA)* applied to a set of 3D points in the Kinect's frame. As shown in Fig. 3-B, such point cloud is extracted from a rectangular area of the depthmap $\mathbf{I_k}$, defined by the center (u_c^k, v_c^k), the height β and the width α. The 2D point (u_c^k, v_c^k) corresponds to the pico-projector's principal point (u_c^p, v_c^p) in $\mathbf{I_p}$, while C^k represents the projection of (u_c^k, v_c^k) in the Kinect's frame.

Algorithm 1. Active Projection

function $\mathrm{AP}(\mathbf{I_k}, \mathbf{I_s})$
  $\mathbf{T} = paramsPlaneEstimation(\mathbf{u_c^k}, \mathbf{v_c^k}, \alpha, \beta)$
  $\mathbf{P_1^k}, ..\mathbf{P_4^k} = selectingFour3DPoints(\mathbf{T}, W, H)$
  $\mathbf{p_1^p}, ..\mathbf{p_4^p} = fromKinect2Projector(\mathbf{P_1^k}, ..\mathbf{P_4^k})$
  $\mathbf{H}_\pi = homography(\mathbf{p_1^p}, ..\mathbf{p_4^p}, (0, 320), ..(320, 0))$
  $\mathbf{I_p} = warpImage(\mathbf{I_s}, \mathbf{H}_\pi)$
  return $\mathbf{I_p}$
end function

The *Active Projection* mode is summarized in Algorithm 1, in which the procedure *AP()* takes as input the source image $\mathbf{I_s}$ and the depthmap of the Kinect $\mathbf{I_k}$ and retrieves as output the image to project $\mathbf{I_p}$.

4.2 Projection Tracking

Once the user has selected a suitable projecting plane, the system switches from the *Active Projection* to the *Projection Tracking* mode. In such process, the robot is still maintained in gravity compensation. The *Projection Tracking* mode consists of an iterative process involving two main functionalities.

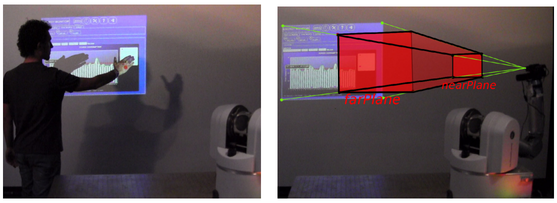
The first one computes the rotational forces to apply to the end-effector for pointing towards the center of the projection plane, which is represented as a 3D point in the robot's frame. The robot actively reacts to perturbation of its joints configuration by keeping constant the size and the perspective of the projected image. The perturbations to the gravity compensated arm, can come from the user physically moving the robot or from the robot actively changing its posture to avoid occlusions.

The second functionality computes an updated perspective transformation, based on the actual end-effector pose, for enabling the resulting image to be undistorted despite the geometric parameters of the plane in the pico-projector's frame may change. Moreover, whenever an occlusion occurs, the system reacts by finding another end-effector pose for making the projection again visible.

According to these two features, the system iteratively computes 1) the end-effector orientation to send to the robot $\mathbf{R_{end}}$ and 2) the warped projected image $\mathbf{I_p}$. Four 3D points $\mathbf{P_1}, ..\mathbf{P_4}$ in the robot's frame are obtained from the points

Algorithm 2. Projection Tracking

1: $\mathbf{P_1}, ..\mathbf{P_4} = fromKinect2Robot(\mathbf{P^k_1}, ..\mathbf{P^k_4})$
2: $\mathbf{C} = findProjectionCenter(\mathbf{P_1}, ..\mathbf{P_4})$
3: **while** true **do**
4: $\quad\mathbf{R_{end}} = LookAt(\mathbf{C})$
5: $\quad sendOrientationToRobot(\mathbf{R_{end}})$
6: $\quad\mathbf{p^P_1}, ..\mathbf{p^P_4} = fromRobot2ProjectorImage(\mathbf{P_1}, ..\mathbf{P_4})$
7: $\quad\mathbf{H}_\pi = homography(\mathbf{p^P_1}, ..\mathbf{p^P_4}, (0, 320), ..(320, 0))$
8: $\quad\mathbf{I_p} = warpImage(\mathbf{I_s}, \mathbf{H}\pi)$
9: $\quad\mathbf{P_{occl}}{=}DO(\mathbf{I_k}, mindistPI, maxdistPI)$
10: $\quad$ **if** $\mathbf{P_{occl}}$ *is not null* **then**
11: $\qquad\mathbf{P_{new}} = findingNewpose(\mathbf{P_{occl}})$
12: $\qquad sendingToRobot(\mathbf{P_{new}})$
13: $\quad$ **end if**
14: **end while**



(a) (b)

Fig. 5. (a) A projected red spot indicates the center of the detected occlusion. (b) The volume in red is the frustum of the projection continuously monitored by the *DO()* function.

$\mathbf{P^k_1}, ..\mathbf{P^k_4}$ in the Kinect's frame which defines the selected projection plane π. In the line 1 of Algorithm 2 the geometric transformation

$$\left[P_1 \;\; P_2 \;\; P_3 \;\; P_4\right]^\top = \mathbf{T_k} \left[P^k_1 \;\; P^k_2 \;\; P^k_3 \;\; P^k_4\right]^\top \tag{5}$$

is computed by using the transformation matrix $\mathbf{T_k}$ between the Kinect and the robot frames (see Fig. 4). Therefore, the target towards which the end-effector has to point, namely the projection's center, is computed in line 2. Then, in the main loop, the end-effector orientation and the warped image are continuously computed based on the actual robot's configuration. The *LookAt()* function, in line 4, provides the end-effector with an orientation matrix $\mathbf{R_{end}}$ for looking towards the point $\mathbf{C}$, while the homography/warp operations on lines 7 and 8 enable the projected image $\mathbf{I_p}$ to appear undistorted. Whether an obstacle occludes the projection (line 10), a new position of the end-effector is found and sent to the robot (lines 11 and 12).

4.3 Detecting and Handling Occlusions

In dynamic environments where humans and robots share the same space, one of the key requirement is to adapt the machine's behaviour to human's habits.

In the particular scenario we are proposing, it may happen that the user occludes the projection with some parts of the body. We propose a geometric method for enabling the robot to adaptively change its configuration to get around obstacles and keep the projection visible. Thus, the user can freely move in the surrounding environment without worrying about occluding the projected interface. While the *Projection Tracking* mode is activated, the system is able to adaptively change the robot configuration in order to manage occlusions that may occur in the frustum of the projection. The *Detecting Occlusion* function *DO()*, in line 9 of Algorithm 2, returns a 3D point in the robot's frame representing the mean occlusion point. Such point is found inside a portion of the projection's frustum by using the Kinect device. The frustum is the region of space, defined geometrically as a regular rectangular pyramid with the apex cut by a parallel plane (*nearPlane*) above the base (*farPlane*). In our case the *farPlane* is the plane parallel to π with perpendicular distance $mindistPI$ from it, while the *nearPlane* is far $maxdistPI$ (see Fig. 5b). In such a way, all the 3D points inside the frustum are collected by computing the depthmap $\mathbf{I_k}$ and a mean value $\mathbf{P_{occl}}$ in the robot's frame is extracted from these samples (see Fig. 5). Indeed, whenever an obstacle occludes the projection, a new end-effector position $\mathbf{P_{new}}$ is computed and reached by the robot, for avoiding the occlusion and making the projection again visible. We propose a geometric approach implemented in the function *findingNewpose()* (line 11 of Algorithm 2).

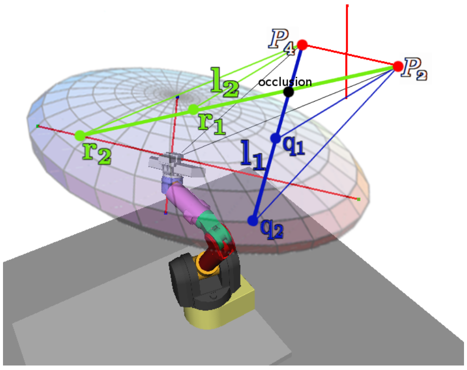
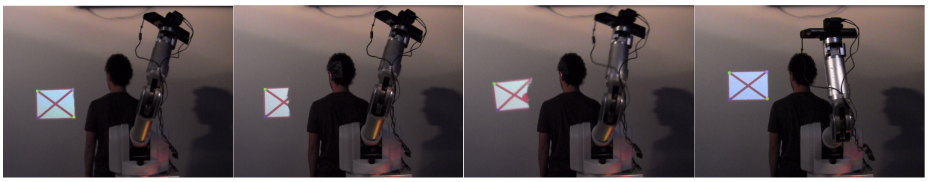


Fig. 6. For each of the two lines $\mathbf{l_1}$ and $\mathbf{l_2}$, two different solutions can be found by the intersection with the ellipsoid $\mathcal{E}$. The selected solution represents the new end-effector position to reach by the robot.



(a) (b) (c) (d)

Fig. 7. The snapshot (a) shows the user visualizing a projected interface before an occlusion occurs. Once the user changes his position and an occlusion is detected, the robot moves by reaching the new end-effector pose for avoiding the occlusion, as shown in (b), (c) and (d).

Let $\mathcal{E}$ be the ellipsoid defined by the equation

$$\mathcal{E} : (\mathbf{x} - \mathbf{P_{end}})^{\top} \begin{bmatrix} a^2 & 0 & 0 \\ 0 & b^2 & 0 \\ 0 & 0 & c^2 \end{bmatrix} (\mathbf{x} - \mathbf{P_{end}}) = 1, \tag{6}$$

where its center $\mathbf{P_{end}}$ is the actual end-effector position and a, b, c define the vertical distances for each axis from $\mathbf{P_{end}}$ to the ellipsoid surface (see Fig. 6). All the points on the surface of the ellipsoid represent candidate solutions to avoid the occlusion. Namely, the ellipsoid defines the maximum volume in which the end-effector can move around to manage each occlusion. Let $\mathbf{l_1}$ be a line between the points $\mathbf{P_{occl}}$ (**occlusion** in Fig. 6) and $\mathbf{P_4}$, and $\mathbf{l_2}$ a line between the points $\mathbf{P_{occl}}$ and $\mathbf{P_2}$, according to the equations

$$\mathbf{l_1} : \mathbf{P_{occl}} + t_1(\mathbf{P_4} - \mathbf{P_{occl}}) \ , \qquad \mathbf{l_2} : \mathbf{P_{occl}} + t_2(\mathbf{P_2} - \mathbf{P_{occl}}). \tag{7}$$

The intersection between the ellipsoid surface $\mathcal{E}$ and the line $\mathbf{l_1}$ defines two solutions, then two other solutions can be computed for the line $\mathbf{l_2}$. Thus, the return value $\mathbf{P_{new}}$ is the closest 3D point between the four solutions found.

5 Conclusions and Future Work

We presented a novel robotic interface for projecting visual information on planar surfaces. In our experimental setup, we considered to extend the capabilities of the compliant 7-DOF Barrett WAM robot by rigidly mounting a pico-projector and a Microsoft Kinect sensor on its end-effector. The perceptual features of the Kinect has been exploited to detect three dimensional information about the geometry of planar surfaces used for superimposing undistorted projections and handling possible occlusions. A human-robot collaboration scenario involving the task of finding suitable projecting surfaces and managing perturbations has been presented. We conducted an experiment showing that the user can manually interact with the gravity-compensated robot for finding a suitable end-effector pose which enables the projection to be superimposed on a desired surface. Therefore, once the position of the projection has been selected, we showed how the robot arm can actively adapt the projection when faced with changes of its joints configuration. Although the experimental results showed that the size and the perspective of the projection are kept quasi-constant, the position tracking will be improved in the future work by refining the calibration methods between the robot and the sensors. We also presented a prototype as a proof-of-concept for managing occlusions based on a geometrical approach.

The proposed experiment opens new research perspectives that will be explored in our future work. In order to measure the quality of the user interactions, evaluation studies will first be conducted. We also plan to improve the current setup by providing the system with an awareness mechanism of the human presence by managing collision avoidance in a predictable manner and by regulating the robot stiffness with respect to the user's distance. Finally, a further aspect to be considered will be the interaction between the user and the projected information by capturing human inputs from hands gestures on the projecting surface employed as an ubiquitous interface.

References

1. Breazeal, C., Kidd, C.D., Thomaz, A.L., Hoffman, G., Berlin, M.: Effects of nonverbal communication on efficiency and robustness in human-robot teamwork. In: Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems IROS, pp. 708–713 (2005)
2. Mollet, N., Chellali, R., Brayda, L.: Virtual and Augmented Reality Tools for Teleoperation: Improving Distant Immersion and Perception. In: Pan, Z., Cheok, A.D., Müller, W., Rhalibi, A.E. (eds.) Transactions on Edutainment II. LNCS, vol. 5660, pp. 135–159. Springer, Heidelberg (2009)
3. Bimber, O., Emmerling, A., Klemmer, T.: Embedded entertainment with smart projectors. IEEE Computer 38(1), 48–55 (2005)
4. Li, L., Zhang, M., Xu, F., Liu, S.: ERT-VR: an immersive virtual reality system for emergency rescue training. Virtual Reality 8(3), 194–197 (2005)
5. Brunnett, G., Rusdorf, S., Lorenz, M.: V-pong: an immersive table tennis simulation. IEEE Computer Graphics and Applications 26(4), 10–13 (2006)
6. Ikeda, S., Asghar, Z., Hyry, J., Pulli, P., Pitkanen, A., Kato, H.: Remote assistance using visual prompts for demented elderly in cooking. In: Proceedings of the 4th International Symposium on Applied Sciences in Biomedical and Communication Technologies, ISABEL 2011, pp. 1–5. ACM, New York (2011)
7. Tardif, J.P., Roy, S., Meunier, J.: Projector-based augmented reality in surgery without calibration. In: Proceedings of the 25th Annual International Conference of the IEEE Engineering in Medicine and Biology Society, vol. 1, pp. 548–551 (2003)
8. Vogel, C., Poggendorf, M., Walter, C., Elkmann, N.: Towards safe physical human-robot collaboration: A projection-based safety system. In: Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems IROS
9. Harrison, C., Benko, H., Wilson, A.D.: Omnitouch: wearable multitouch interaction everywhere. In: Proceedings of the 24th Annual ACM Symposium on User Interface Software and Technology, UIST, pp. 441–450. ACM, New York (2011)
10. Rusu, R.B., Cousins, S.: 3D is here: Point cloud library PCL. In: Proceedings of IEEE International Conference on Robotics and Automation (ICRA), Shanghai, China, pp. 1–4 (May 2011)