SEPO: Selecting by Pointing as an Intuitive Human-Robot Command Interface

Camilo Perez Quintero, Romeo Tatsambon Fomena, Azad Shademan, Nina Wolleb, Travis Dick and Martin Jagersand*

Abstract— **Pointing to indicate direction or position is one of the intuitive communication mechanisms used by humans in all life stages. Our aim is to develop a natural human-robot command interface using pointing gestures for human-robot interaction (HRI). We propose an interface based on the Kinect sensor for selecting by pointing (SEPO) in a 3D real-world situation, where the user points to a target object or location and the interface returns the 3D position coordinates of the target. Through our interface we perform three experiments to study precision and accuracy of human pointing in typical household scenarios: pointing to a "wall", pointing to a "table", and pointing to a "floor". Our results prove that the proposed SEPO interface enables users to point and select objects with an average 3D position accuracy of 9.6 cm in household situations.**

I. Introduction

In recent years, robots have started migrating from industrial to home assistive scenarios. One of the biggest challenges in this transition is finding natural communication mechanisms that allow humans to effortlessly interact with a robot. Pointing is one of the communication mechanisms frequently used by humans in all life stages. Some researchers believe that pointing is an important stage linking preverbal communication and spoken language [1].

When a human instructs another to perform a task, non-verbal communication cues in the form of pointing and gesturing play important roles. By contrast, robots and machines are usually instructed either by text-based programming, or direct control of motions, *e.g.,* using a teach pendant or joystick interface.

Pointing gestures provide only coarse spatial information of a target location. Adding verbal communication can dramatically improve inferences about the spatial location of the target. As humans, we interpret pointing in the rich context of prior information and other cues, but rarely reflect on the actual accuracy of pointing. Encoding verbal clues into location information is nontrivial and often entails increased complexity at the software level. Non-verbal pointing communication seems to be a low-cost alternative in the Human-Robot Interaction (HRI) context. A challenge in integrating pointing gestures is that the precise pointing direction cannot be easily inferred by a third person or a robot. Our aim is to make human pointing gestures accurate enough such that the robot can detect the pointing direction and find the desired target via the interface.

In this work we focus on visual pointing. We build a system that reads human pointing direction, and if the target is in the field of view, the system returns the 3D target position. Interactions are by making intuitive gestures without the need for an external pointing device. We evaluate the accuracy on these both metrically and in terms of success rate when selecting one among many ordinary household objects.

Human machine interaction researchers have focused on building interfaces capable of detecting pointing gestures and estimating the 2D pointing direction. Kahn and Swain [2] introduced a pointing gesture detection through their Perseus architecture. Jojic *et al.* [3] developed a system for detecting pointing gestures and found an estimation of the gesture direction using stereo cameras and dense disparity maps in real-time. Nickel and Stiefelhagen [4] used face and hand tracking then used Hidden Markov Models to classify the 3D trajectories. Kehl and Van Gool [5] present a multi-view approach that measures 3D directions of one or both arms in 3D. All of the above approaches are limited to give the direction of the pointing gesture. By contrast, our approach is akin to ray casting in graphics and virtual reality which computes the 3D position of the first object surface intersected by the pointing ray [6]. A challenge is to make this idea applicable and accurate in a real-world context.

The point-and-click method proposed by Kemp *et al.* [7] enables humans to select a 3D world location using a laser pointer. The robot is then able to detect the laser spot and estimate its position with respect to the robot. Our approach is different, because we want to avoid using external devices for pointing. Our proposed interface also returns both the pointing direction and the target position (x, y, z). Figure 1 shows the proposed system, where the Microsoft Kinect$^{\text{TM}}$ sensor is utilized as a non-verbal interface to select a target object by a pointing gesture. This system has potential relevance in HRI, *e.g.,* commanding a robot arm to bring a selected object.

Our target scenario addresses pointing and selection of household objects, even though this can be applied to any similar unstructured task. The main contributions are:

- A system for point-and-select gesture recognition. Using both the RGB camera and the depth sensor from a Kinect$^{\text{TM}}$, we compute the pointing ray, the 3D intersection between the ray and an object, and detect a selection gesture (Section II).

*This work is supported by NSERC and the Canadian Space Agency (CSA).

Authors are with the Department of Computing Science, University of Alberta, Edmonton, AB, T6G2E8, Canada. caperez@cs.ualberta.ca

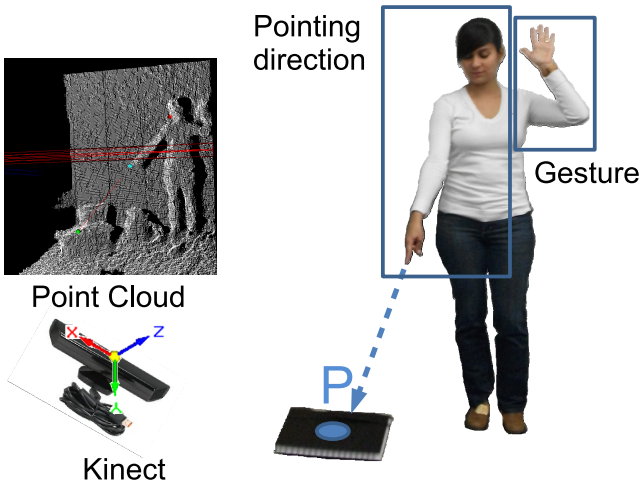

Fig. 1: Using a depth camera, the system returned the (x,y,z) coordinate of point P

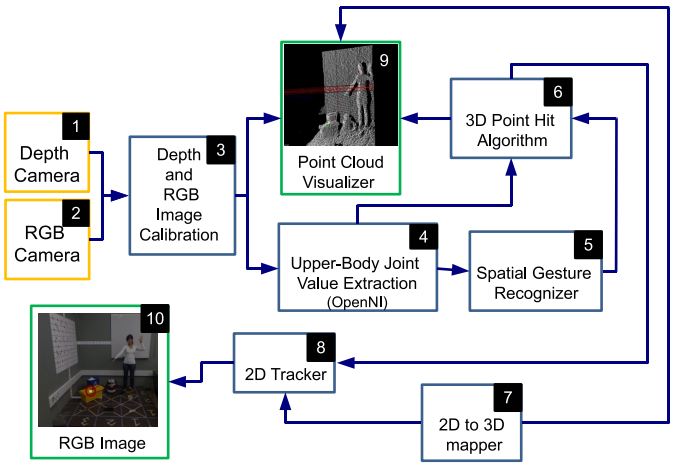

Fig. 2: SEPO system Diagram

- A study with nine subjects, where both the metric accuracy of pointing as well as the success ratio of selecting some common household objects is reported. Three common configurations are evaluated, namely pointing to a wall, a table and the floor (Section III).

II. Interface Description

Our proposed interface allows the user to point to an object and through a gesture inform the system that the target direction is defined. Having defined the direction, the system first calculates the location of the target object with respect to the camera location. It then launches a tracking algorithm in the neighborhood of the target position, which allows the system to track the target object and return the object's 3D position (see Figure 1). There are two motivations behind the use of tracking after selecting the object. The first is to correct some of the inaccuracy of the pointing gesture. The second is to deal with dynamic settings where objects may be moved around.

Figure 2 shows our complete system block diagram which can be simplified in three principal stages: (a) Gesture and Pointing Identification Algorithm (blocks 1-5 in Figure 2); (b) 3D Point Hit Algorithm (block 6 in Figure 2); and (c) Object Tracking (blocks 7 and 8 in Figure 2).

A. Gesture and Pointing Identification Algorithm

We used the Microsoft Kinect sensor for acquiring depth and color image data (blocks 1 and 2 in Figure 2). The first step is to do a color and depth camera calibration (block 3) [8]. Then using the free cross-platform Kinect driver OpenNI and the NITE skeletal tracking library [9], we identify the upper-torso joints (block 4). These joint locations are the input to the spatial gesture recognizer module (block 5), which is implemented as a finite-state machine. In our particular pointing interaction scheme, we use three states as shown in Figure 3. In the first state the user stands in front of the Kinect and makes a neutral calibration pose. This allows the NITE skeletal tracking library to initialize by doing background substraction, human silhouette identification, and skeleton joint-value extraction. The transition from first state (calibration/initialization) to the second state (target selection) is done when the user points to the target with her dominant hand and raises her non-dominant hand over her shoulder to confirm the pointing direction. If the user lowers her non-dominant hand below her shoulder, the state shifts to the third state, in which the object is being tracked. When the user points to other targets and gives a target selection gesture (raising her non-dominant hand over her shoulder again), the state is shifted to "selection".

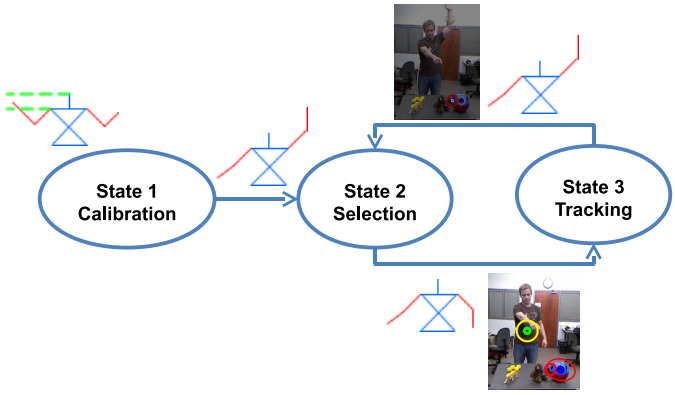

Fig. 3: Object selection state diagram

B. 3D Point Hit Algorithm

For building the hit algorithm, we used two pointing direction configurations, the line between head and dominant hand (LHH) and line between dominant elbow and dominant hand (LEH). Having the pointing direction, we need a 3D world representation for finding the 3D hit point. We used the PCL library [10] for extracting and manipulating the 3D point cloud from the depth camera sensor. And then, we add

in real-time the location of the user torso joint values inside the point cloud which allows us to find the pointing location looking at the LHH or LEH configuration.

After acquiring the pointing direction by having either LHH or LEH configuration we can define a parameterized line equation:

$$\vec{\mathbf{l}} = \vec{\mathbf{J}}_{upper} + t\,(\vec{\mathbf{J}}_{hand} - \vec{\mathbf{J}}_{upper}), \qquad (1)$$

where $\vec{\mathbf{l}}$ is the parameterized line vector, $t \in \mathbb{R}^+$ is a non-negative real-valued step of the parameterized line, $\vec{\mathbf{J}}_{hand}$ is the 3-vector of the 3D hand coordinates, and $\vec{\mathbf{J}}_{upper}$ is the 3-vector of the 3D elbow or head coordinates, depending on the chosen configuration. For our ray to point cloud searching algorithm, we can simply replace the parameterized line inside the sphere equation, returning a discrete set of spheres located across the line:

$$\| \vec{\mathbf{x}} - (\vec{\mathbf{J}}_{upper} + t(\vec{\mathbf{J}}_{hand} - \vec{\mathbf{J}}_{upper})) \|^2 = r^2, \qquad (2)$$

The point cloud generated by the depth sensor is structured in an octree and the algorithm iterates over the multiple spheres defined in (2) for a possible hit point.

Fig 4 shows a user inside the point cloud using the LHH direction configuration where the red sphere is the tracked position of her head, the blue sphere her hand and the green sphere the target location. Notice that a virtual ray (red line) is shown in real time indicating the pointing direction.

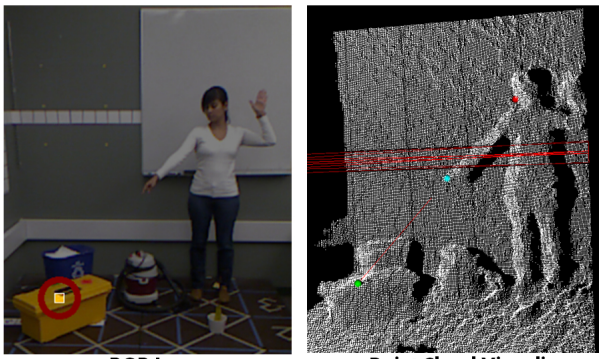

Fig. 4: System Visual Output

C. Object Tracking

The last stage of our interaction consists in tracking the selected object. We take advantage of the direct correspondence between the 2D image pixels and the 3D points from the RGB camera and depth camera calibration. After getting the 3D target location from the point cloud, we mapped this location to the 2D image and ran a 2D tracker in the neighborhood region for tracking the hit object. In our system we used a 2D Camshift tracker [11], it outputs the centroid location of the 2D tracking region. After the object changes its position the inverse mapping procedure is done using the centroid pixel coordinate as input and returning as output the 3D location of the object tracked (see Figure 4). Besides dealing with object location changes, and assuming that the hit object has a uniform color, the Camshift algorithm finds the centroid of the object and therefore corrects the hit point position into the centroid of the object.

Although in our experiments we didn't perform this stage, a typical use for SEPO in HRI context could be to initialize a tracker on an object as explained before and then input this to a control routine, *e.g.,* visual servoing, to command a robot to reach for the desired object.

III. Experiments

We are particularly interested in quantifying three aspects:

- Human accuracy in LHH and LEH configuration.
- Precision in LHH and LEH configuration.
- Hit accuracy for typical household objects.

Where accuracy is understood as the distance from the measure value to the reference value i.e. Euclidean distance between the hit point and the target location, and precision as the average spreading distance around the measure location.

A. Experiment setup

To avoid the influence of fatigue in our trials we divided the experiments in three main sections: "table", "floor" and "wall". Each section had a duration of approximately 20 minutes per participant. The experiments were carried out on three days in a row. We had nine subjects participating. Seven male, two female, all with normal or corrected to normal vision and right-handed. The age varied between 18 and 34. All setups consisted of a Microsoft Kinect running simultanously the RGB and depth camera, a regular Linux machine with our SEPO interface software.

All three main sections were divided like this.
For the "table" section (left Figure 7) a tabletop (height 67 cm) was divided to a field of 5 by 6 squares with 10 cm side length each. In the middle of these squares numbered targets were attached (red targets Figure 5). For pointing on the table the subject stood with a distance of approximately 0.5 m on one side of the table.

For the "floor" section (middle Figure 7) a plateau was divided in 3 by 3 squares of 50 cm and marked with numbers for rows and columns(Figure 6). The subject was asked to stand at different locations for pointing a particular intersection or objects.

For the "wall" section (right Figure 7) a poster of 14 by 10 squares of 10 cm side length each was attached and equipped with numbered targets (Figure 10). Additionally to the poster there were a horizontal (right) and a vertical row (up and down) of squares attached to the poster to measure the accuracy over a big area. The subjects had a distance of 80 cm to the wall centered in front of the poster (right Figure 7).

B. Experiment description

Before the pointing process could start each subject had to go through the calibration process of standing facing to the

Kinect, putting up both hands, waving them back and forth and maybe taking a step towards or away from the Kinect untill the person was tracked. After that the participant was asked to go to the pointing position. The person was allowed to orient herself freely and suitably for the pointing direction.

Sections “table” and “wall” were subdivided into 11 short tasks, and section “floor” in 10. For all the sections, in the first two tasks the subject was asked to point at the called number (bottom Figure 5) or intersection (bottom Figure 6) and to select the pointing position by raising the left arm.

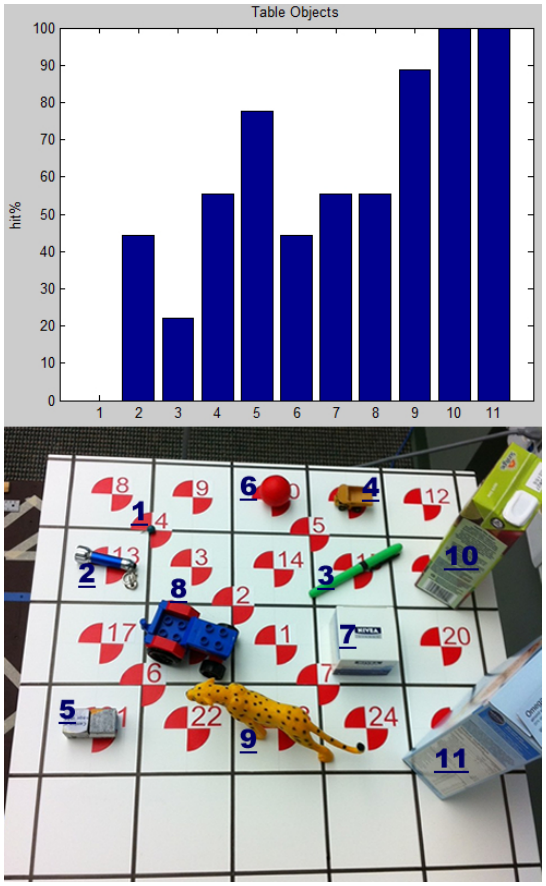

Fig. 5: Hit%, ”table” scenario, numbers in blue in the bottom image corresponds to numbers in bargraph

No further instructions were given. The subject was supposed to point as natural as possible. While carrying out the first task the LHH configuration was applied and in the second task the LEH configuration. In the third and fourth task the subject was told that the system is running with the LHH strategy now. In the fifth and sixth task the participant was asked to point with the LEH strategy. The accuracy of those tasks with the knowledge of the strategies were compared and the following tasks were carried out with the more accurate strategy. In the last task (eleventh for “table”and “wall”, tenth for the “floor”) real objects were added to the scene and the subject was asked to point at them.

After the last experiment the subjects were asked to complete a questionnaire. The evaluation showed that the

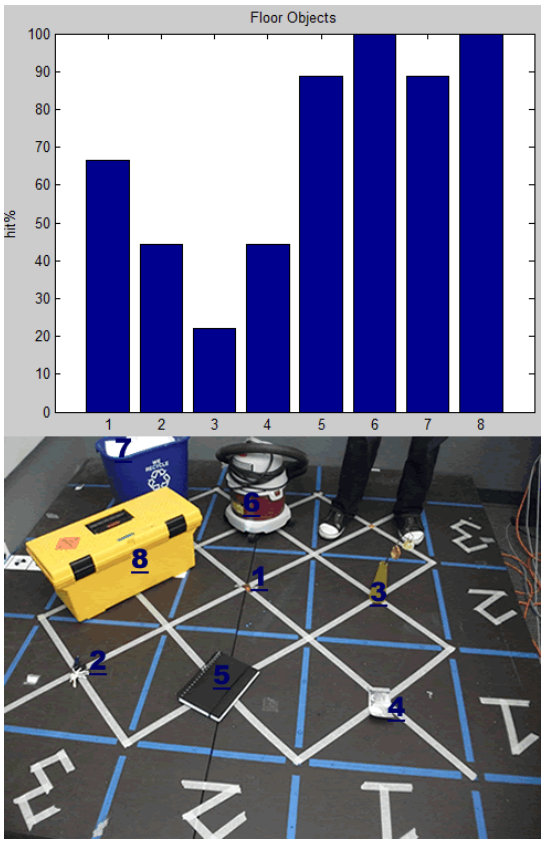

Fig. 6: Hit%, ”floor” scenario, numbers in blue in the bottom image corresponds to numbers in bargraph

Fig. 7: Experiments, Left:table, Middle:floor, Right:wall

subjects did not think the accuracy of their pointing changed in the three scenarios. The self-reported overall difficulty was rated in the middle of the spectrum (variance 5.61) and overall fatigue slightly under medium (3 of 7 with 7 as very high, variance 4.44). As the variance in both of these ratings shows people had a different sense of how difficult and tiring the pointing and selecting was. The head and hand alignment strategy was preferred, only two participants liked pointing with the forearm better. We also asked how annoying the gesture of raising the hand for finally selecting the object was. The participants rated the annoyance in the middle of the spectrum with a variance of 2.78.

C. SEPO Accuracy and Precision

In this section, we evaluate the accuracy and precision of the SEPO interface in three scenarios: “table”, “floor”, and “wall” and compare the two pointing configurations:

LHH and LEH. The results are summarized in Figure 8. Our

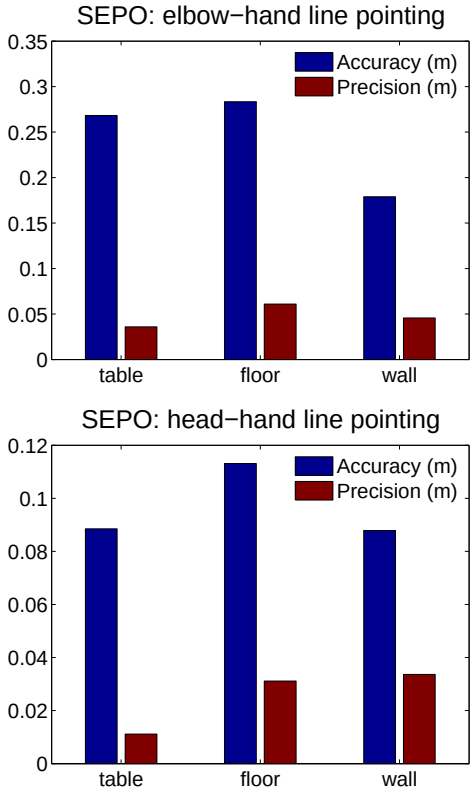

Fig. 8: Elbow and head hand-line pointing results

results confirmed published data which says that pointing is more accurate and precise in the case where the pointing direction is computed from the head-hand line than in the case where the pointing direction is computed from the elbow-hand line [12]. In addition our results suggest head-hand pointing on the table with SEPO interface is more precise than pointing on the Wall and the floor.

This can be explained by fact that the table setup presents an adequate ratio of the proximity of the subjects to the visually tracked area of the objects. The accuracy of pointing to the wall and on the table is respectively 2.5 cm and 2.4 cm better than for the floor.

After task 6 the following tasks were carried out with the more accurate strategy which was LHH, experiments are summarized in Figure 9 with 5 of our participants.

For the table scenario, the subjects point 2 times repeatedly to 4 different locations on a numbered grid (see the picture of the table in Figure 5). The results indicate an average accuracy of 8.8 cm across all participants within a $[5.4,\ 12.4]$ cm range. To give task related sense of the accuracy, we can say that on average users would have no trouble pointing to bigger objects (*e.g.,* cereal box, juice box); see bottom Figure 5). The average precision acquired for this experiment was 1.1 cm within a $[0.4, 5.2]$ cm range.

For the floor scenario, the subjects point repeatedly 4 times to 3 different locations on the grid floor (see Figure 6). The results indicate an average accuracy of 11.3 cm across all participants within a $[4.0,\ 18.6]$ cm range. The average precision acquired for this experiment was 3.1 cm within a $[2.7, 14.1]$ cm range, users would have no trouble pointing objects like toolbox, garbage bin, vacuum cleaner(see Figure 6).

For the wall scenario, the subjects point twice to each of 3 different locations on a grid pattern on the wall (see Figure 10). The results indicate an average accuracy of 8.8 cm across all participants within a $[3.6, 18.3]$ cm range. The average precision acquired for this experiment was 3.4 cm within a $[0.6, 11.5]$ cm range, users would have no trouble pointing objects like poster, ballon, wall-clock (see Figure 10).

D. Evaluation of 3D Point Hit Algorithm on Everyday Targets

To validate our experimetal results we performed a visual evaluation using our system visual output (Figure 4). The experiment include real objets for each of the scenarios. Through the visual system output we detect if the object was succesfully selected. If the object was missed we noted how many squares it is away from the object. The average of hit percentage was 58% for all the objects on the table, 100% for big objects (10,11 in Fig. 5), 57% for medium objects (7,8,9 in Fig. 5) and 41% for small objects (1,2,3,4,5,6 in Fig. 5). The average of hit percentage was 69% for all objects on the floor, 96% for big objects (6,7,8 in Fig. 6), 52% for medium objects (3,4,5 in Fig. 6) and 55% for small objects (1,2 in Fig. 6). The average of hit percentage was 51% for all objects on the wall, 89% for big objects (4,5,6 in Fig. 10), 33% for medium objects (3 in Fig. 10) and 22% for small objects (1,2 in Fig. 10). The difference in hit% between the scenarios is due to the relation between surface area projected in the user field of view and the distance to the target object. Bigger objects (e. g. table: cereal box, juice box;floor:vacuum cleaner, tool box;wall:balloon, poster) were hit more often than smaller ones (picture frame, flower pot or pin). When an object was missed the hit point was almost all the time in an area of one square (10 cm on the wall, 50 cm on the floor) from the desired object. The results are summarized in Figures 5,6,10).

E. Subjective evaluation

As some of the participants of the trial found raising the left hand for finally selecting an object was pretty annoying, alternative gestures should be considered. The subjects were asked to make suggestions. One of these was using verbal commands. An advantage of the verbal commands is that they could add another factor to make the selection more precise with for example telling the color or the shape of the desired object. Another solution that could be thought of is a handheld button with the drawback that when using a button you can't handle other objects, and as mentioned before we are particularly interested in not introducing any invasive device to our interface. Other proposals from trial subjects were making a fist, opening the hand or simple nodding.

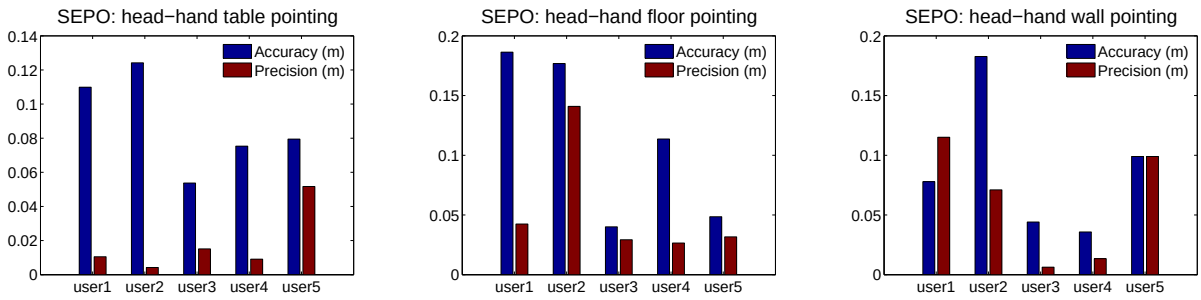

Fig. 9: Accuracy and precision results for LHH configuration

These are undeniably gestures which can be carried out with less effort.

Besides searching for an alternative to the gesture, making the gesture more precise or using different gestures for different commands could be a big advantage even if it would add more complexity to the interface. For example pointing with a special finger could mean that something in a special place (wall, floor or table) or something with a special color is meant.

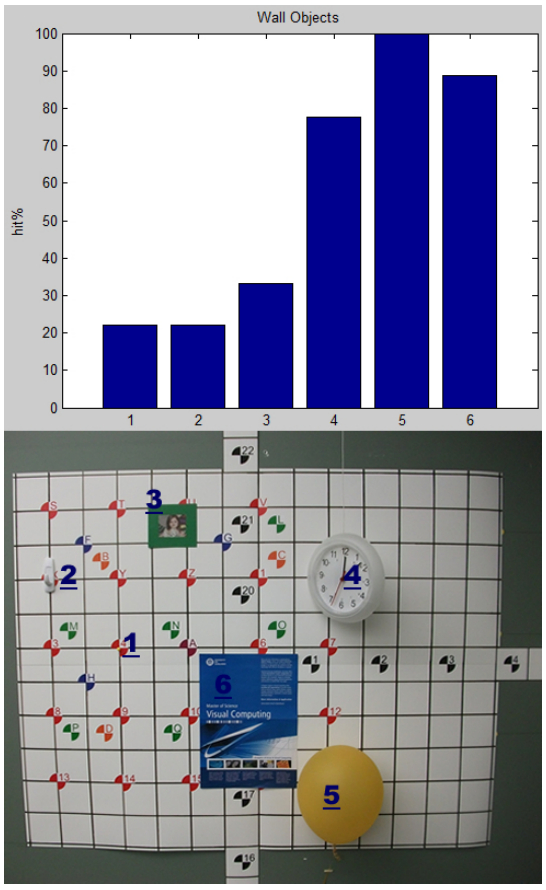

Fig. 10: Hit%,"wall" scenario, numbers in blue in the bottom image corresponds to numbers in bargraph

IV. Conclusions

We have proposed the selecting by pointing (SEPO) interface for HRI, based on the Microsoft Kinect sensor. We have studied three specific household scenarios: "table", "floor", and "wall", with two different pointing configurations LHH and LEH. Our results have demostrated that the LHH configuration consistently outperforms the accuracy and precision of the LEH configuration. This is consistent with other results in the literature. Our results has also shown that the proposed SEPO interface enables users to point to and select objects with an average position accuracy of 9.6 ± 1.6 cm in household situations, that means we can successfully selected objects similar to: cereal box on the kitchen table, a notebook on the desk, a poster on the wall, a vacuum cleaner on the floor, etc. A demonstration of the SEPO interface can be seen in the accompanying video or in our website `http://webdocs.cs.ualberta.ca/~vis/HRI/`.

References

[1] F. Simion and G. Butterworth, Eds., *The Development Of Sensory, Motor And Cognitive Capacities In Early Infancy: From Sensation To Cognition*. Psychology Press, 1998.

[2] R. E. Kahn and M. J. Swain, "Understanding people pointing: the perseus system," in *Proc. Symp. Int Computer Vision*, 1995, pp. 569–574.

[3] N. Jojic, B. Brumitt, B. Meyers, S. Harris, and T. Huang, "Detection and estimation of pointing gestures in dense disparity maps," in *Proc. Fourth IEEE Int Automatic Face and Gesture Recognition Conf*, 2000, pp. 468–475.

[4] K. Nickel and R. Stiefelhagen, "Pointing gesture recognition based on 3d-tracking of face, hands and head orientation," in *ICMI*, 2003, pp. 140–146.

[5] R. Kehl and L. Van Gool, "Real-time pointing gesture recognition for an immersive environment," in *Proc. Sixth IEEE Int Automatic Face and Gesture Recognition Conf*, 2004, pp. 577–582.

[6] D. A. Bowman, S. Coquillart, B. Froehlich, M. Hirose, Y. Kitamura, K. Kiyokawa, and W. Stuerzlinger, "3d user interfaces: New directions and perspectives," *IEEE Comput. Graph. Appl.*, vol. 28, no. 6, pp. 20–36, 2008.

[7] C. C. Kemp, C. D. Anderson, H. Nguyen, A. J. Trevor, and Z. Xu, "A point-and-click interface for the real world: Laser designation of objects for mobile manipulation," in *Proc. 3rd ACM/IEEE Int Human-Robot Interaction (HRI) Conf*, 2008, pp. 241–248.

[8] K. J. H. J. Herrera C., D., "Joint depth and color camera calibration with distortion correction," in *TPAMI*, 2012.

[9] *OpenNI User Guide*, OpenNI Organization, August 2012, last viewed 23-08-2012. [Online]. Available: http://www.openni.org/documentation

[10] R. Rusu, "3d is here:point cloud library (pcl)," in *Robotics and Automation(ICRA)*, 2011.

[11] B. G., "Computer vision face tracking for use in a perceptual user interface," in *Intel Technology Journal Q2*, 1998.

[12] K. Nickel, E. Scemann, and R. Stiefelhagen, "3d-tracking of head and hands for pointing gesture recognition in a human-robot interaction scenario," in *Proc. Sixth IEEE Int Automatic Face and Gesture Recognition Conf*, 2004, pp. 565–570.