Visual Support System for Selecting Reactive Elements in Intelligent Environments

Martin Majewski, Andreas Braun, Alexander Marinc, Arjan Kuijper
Fraunhofer Institute for Computer Graphics Research - IGD
Darmstadt, Germany
{martin.majewski, andreas.braun, alexander.marinc, arjan.kuijper}@igd.fraunhofer.de

***Abstract*—Concerning gestural interaction in realistic environments there often is an offset between perceived and actual direction of pointing that makes it difficult to reliably select elements in the environment. This work presents a visual support system that provides feedback to a user gesturing freely in an environment and thus enabling reliable selection of and interaction with reactive elements in intelligent environments. A prototype has been created that is showcasing this feedback method based on gesture recognition using the Microsoft Kinect and feedback provision using a custom laser-robot. Finally an evaluation has been performed, in order to prove the efficiency of such a system, acquire usability feedback and determine potential learning effects for gesture-based interaction.**

Keywords- Gesture based interaction, Human-computer interaction, Feedback, Ambient intelligence

I. INTRODUCTION

Ubiquitous Computing, Pervasive Computing, Ambient Intelligence and Smart Environments are common terms for a similar goal - envisioning a future of computing that shifts away from classical desktop applications and instead relies on devices so small and unobtrusive they can be placed throughout our immediate environment and thus enabling a new paradigm of human-computer-interaction, whereas the machines will be able to infer our intentions from our actions without having to rely on classical input devices such as mouse and keyboard [1]. This interaction is natural and multi-modal - that is we interact with devices similar to interacting with other human beings using speech, gesture and facial expressions [2]. A specific application of gestural interaction is the pointing-for-selection process (PFS) that allows selecting of elements by pointing at them and performing a specific selection gesture. While humans are particularly sophisticated regarding analyzing gestural selection of other humans this task is very complex for computers, considering different types of gesturing and the offset caused by the displacement between eyes and arm. In the following work we will give an introduction to specific challenges of the pointing process and provide a visual feedback solution that improves gesture analysis by computer systems and therefore enables the application of the PFS in intelligent environments without requiring a static visual output.

II. BACKGROUND & RELATED WORKS

In the last few decades there have been a lot of projects that investigated the potential applications, limitations and ramifications of intelligent environments. In their projects AIR&D [3] and Oxygen [4] Philips research and various partners have investigated the extensive use of home automation systems and user context to control typical living spaces. The University of Essex has done similar research with a particular focus on student dormitories [5].

Gestural interaction using the whole body has been a research interest for many years. Starting with early attempts in the 60s [6], [7] the current state-of-the-art is mostly driven by virtual reality and entertainment applications [8], [9]. A driving factor for this work is the availability of the Microsoft Kinect that provides real-time pose information of several human bodies [10].

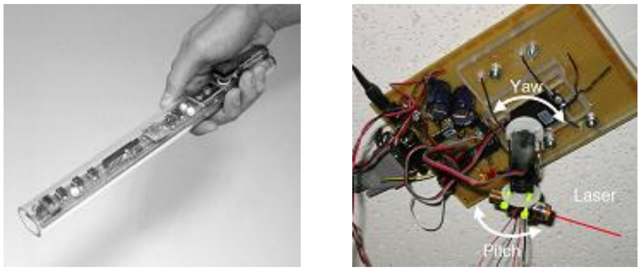

Figure 1. XWand input device (left) and WorldCursor (right)

A project similar to the work presented and source of inspiration has been realized in the 2000s by Wilson et al [11], [12]. The XWand is a dedicated input device based on inertial measurement units and infrared LEDs that allows determining position and orientation of the device in order to gather information about the device that is currently being pointed at and provides means for interaction. In later work this was augmented with the WorldCursor, a laser-pointing device that highlights the location currently selected by the XWand in the environment, in order to improve the selection process. Different working modes such as relative pointing and absolute pointing have been investigated and various application scenarios outlined. Our system improves upon this work by providing device-free gesture interaction, using a more sophisticated method of modeling devices in intelligent environments and providing an evaluation that proves the benefits of such a system and the existence of the PFS offset that was described previously.

III. Pointing-for-selection process

The process of pointing at and selecting a device via gestures can be abstracted to the more generic TOTE-model by Miller et al [13] that postulates that a user may achieve a certain goal by iteratively comparing the actual status with the desired target status and modify the actual until the target criteria is met, as shown in Figure 2.

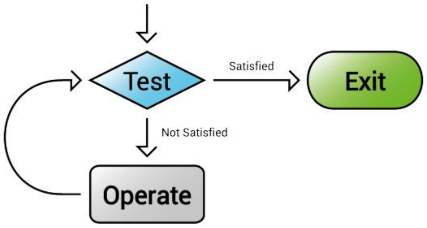

Figure 2. The TOTE-Model by Miller, Galanter and Pirbram

Concerning gestural input with a feedback mechanism this means that the user is adjusting the pointing position until the feedback indicates that the desired target criteria have been met thus performing a successful selection.

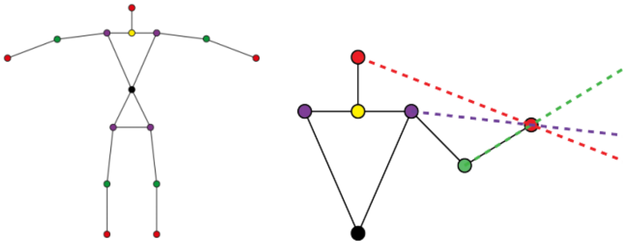

Figure 3. Joint model on the left and different methods to project rays on the right

The actual biomechanical process of pointing is complex and can vary from person to person. A simplified skeletal joint structure is used that allows modeling of different pointing methods. The chosen joint model coincides with the skeletal data acquired by the Microsoft Kinect and is sufficient to model pointing processes that disregard fingers. Nonetheless, it is possible to create different rays from this model, e.g. the extensions of the eye-hand-vector, shoulder-hand-vector or elbow-hand-vector. Both joint model and rays described are shown in Figure 3. A ray here is defined as the parametric equation of a line g with any point $\vec{r}$ defined through the origin vector $\vec{o}$, direction vector $\vec{d}$ and a multiplier λ in the following equation.

$$g: \vec{r} = \vec{o} + \lambda \vec{d}; \lambda \in \mathbb{R};\ \vec{o}, \vec{d} \in \mathbb{R}^3 \tag{1}$$

IV. Visual Assistance of Pointing-for-Selection Processes

The following section will describe the challenges that occur in typical PFS processes and propose a system that allows to overcome these disadvantages by providing a visual feedback system that allows both user and machine to properly evaluate the gestures and may provide a learning effect that allow users to perform better in PFS applications within actual environments.

A. *Challenges in pointing gesture recognition*

The challenges can be distinguished into two groups - target-actual errors and input-reaction delay.

Target-actual errors describe deviations in exactness and uniqueness of the desired target as opposed to the actual target. Exactness offset originate from two different sources - the offset between the user's mental ray projection and the actual ray direction as derived from biomechanical posture and limitations of the gesture recognition system.

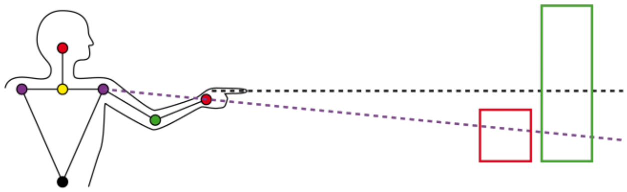

Figure 4. Offset between mental ray projection (originating from finger) and actual ray projection (originating from shoulder)

The use of the index finger as pointer and the ray's constructing source is a common choice by humans but is not supported by using our simplified skeleton model. This alternative ray construction can lead to significant confusion between the user's mental ray representation and system's ray construction, resulting from the appearing offset seen in Figure 4.

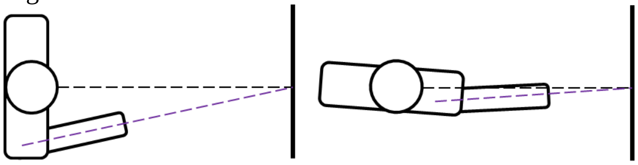

Figure 5. Parallax between eye viewpoint and shoulder viewpoint

Another factor is the parallax between eye viewpoint and arm viewpoint that is affecting the perceived direction of the pointing. Figure 5 shows this effect from top perspective. If the ray is constructed based on the shoulder or elbow joints there is a considerable difference between the two viewpoints, resulting in differently calculated angels. The closer the target object is, the higher the offset. In certain situations, this parallax can be reduced as highlighted on the right side. In some cases even elimination is possible, if eye viewpoint results in the same ray as shoulder-elbow viewpoint.

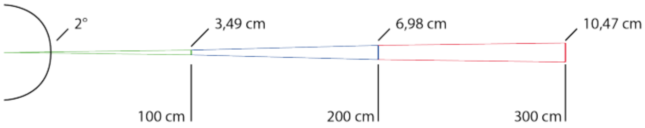

Figure 6. Influence of detection errors based on distance

Figure 6 highlights the influence of detection errors on the detected pointing target based on distance from the user. A minor error of 2° may result in an offset of several centimeters in a distance of a few meters, an example solution to the following equation.

$$offset = distance \cdot 2 \cdot \tan\left(\frac{angle}{2}\right) \qquad (2)$$

The second target-actual errors are uniqueness errors, whereas the user wants to select a target that is ambiguous. This means that in the line of the ray there are various reactive elements that could potentially be selected. There are various methods available to determine the intended target, such as always picking first on ray, selecting the largest target on the ray or associating a priority to all targets. In this work we have focused on the method mentioned first - considering the first reactive element hit by the ray as intended target.

The second group of challenges is associated with input-reaction delay. Concerning gesture input Kammer et al [14] distinguish between online gestures, where reaction follows within a timeframe of 100 milliseconds and offline gestures that are indicated by a reaction to a static gesture in the scope between several hundred milliseconds and a few seconds. If the visual feedback is delayed considerably that may lead to backtracking - the user trying to compensate for the delay by moving back in direction even though the system would have followed shortly after, which may lead to confusion for the users. If offline gestures are associated with a timeframe too long the user might stop the interaction without triggering a reaction.

Feedback and assistance may be provided by various means. A common method is using visual feedback on static displays. However this method is insufficient in intelligent environments if the task area is not within line-of-sight of the user. If he is pointing in opposite direction of the display it would be necessary to turn around in order to get feedback, resulting in a loss of focus. Audio feedback, haptic feedback or mobile display feedback are other available options but either require devices to be worn or are considerably slower than visual feedback.

B. Visual assistance method

A few prerequisites are required for the PFS process to be applied to an intelligent environment. First and most prominently a virtual representation of the environment is required that models all boundaries and reactive elements. The elements should be modeled in a way that they are applicable for ray intersection algorithms. Particularly well-suited are bounding boxes such as axis-aligned bounding boxes (AABB) or oriented bounding boxes (OBB) that form an outer shell of the actual object and can be calculated and intersected easily.

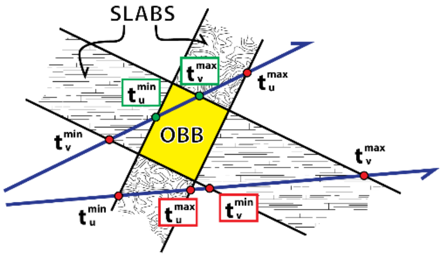

Figure 7. The slab method for ray intersection detection [15]

The intersection with an object and therefor focal point of the gesture can be calculated using the slab method that considers the bounding box as the space inside of three pairs of parallel planes [15]. The ray is clipped from all planes and an intersection considered if a portion of the ray remains, as shown in Figure 7. The second prerequisite is the presence of a gesture recognition device that allows gathering posture information in the skeletal model as described in Section III.

As previously mentioned the first object intersected is considered as intended target, thus intersections with all available objects are performed to determine a minimum λ value.

At the determined focal point visual feedback is given by means of projection from a device that is available in the device. This feedback is supporting both navigation and feedback on successful selection. Typically temporal gestures - that is gestures triggered by remaining on a selected target for a certain time - are suited for selection purposes. The feedback system can support this interaction method by various cues:

- Exact focus point tracking: To support navigation and give feedback on current pointing direction the focus point is continuously highlighted
- Object snapping: As soon as a reactive element is intersected the focus point is set to the center of this element highlighting that this element can be interacted with
- Highlighting: If selection or interaction is successful a specific lighting pattern, e.g. blinking, can be displayed to provide more detailed feedback

A system that implements all aspects previously mentioned is able to provide visual assistance for gestural interaction in intelligent environments.

V. System Prototype and the E.A.G.L.E. device

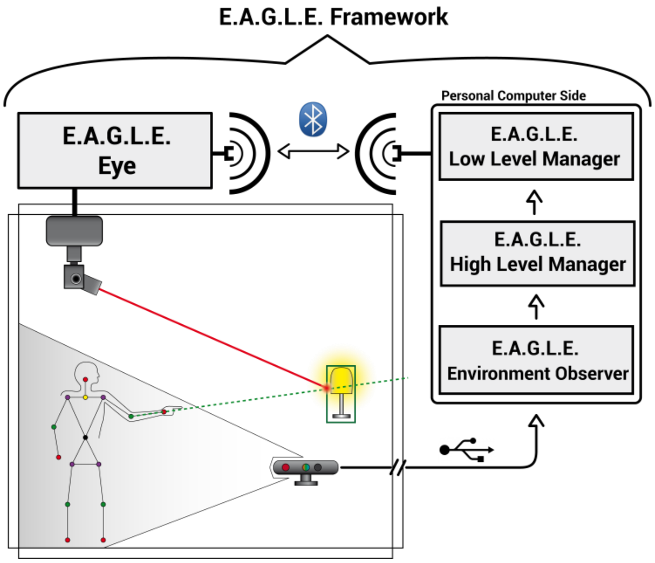

Figure 8. E.A.G.L.E. visual assistance platform

We have created a prototype of the visual support system based on the Microsoft Kinect device for gesture recognition, a regular PC system to perform all required computations and a custom-designed laser robot - the E.A.G.L.E. Eye - that is able to project a laser dot freely into an environment. The whole setup is displayed in Figure 8.

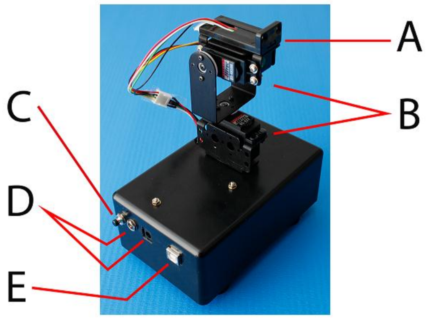

Figure 9. View on the E.A.G.L.E. Eye: A - laser unit, B - servo motors, C - reset switch, D - power supply, E - USB port

The E.A.G.L.E. Eye is based on an Arduino microcontroller board that is operating a laser mounted on two servo motors that allow free and precise positioning of a laser dot in the room. The device is communicating with a PC wirelessly using Bluetooth.

Underlying is the virtual representation of the environment that is modeling all reactive elements, the position of the gesture tracking device and the position of the E.A.G.L.E. Eye. The Kinect is interfaced using the OpenNI framework and NITE middleware that are providing a skeleton tracking algorithm resulting in the joint model mentioned previously that is registered into the virtual representation. The Environment Observer component is performing various filtering algorithms on this joint model and creates the pointing ray vector and performs intersection tests with all available objects.

The resulting focus point is then forwarded to the High Level Manager that calculates the E.A.G.L.E. Eye pointing vector based on spatial position of the device and calculated focal point. The Low Level Manager finally is determining control parameters for the laser robot and is handling the Bluetooth-based communication with the firmware on the Arduino microcontroller.

VI. Evaluation

An evaluation was performed that had three distinct goals:

- Verifying the existence of an offset between perceived and actual pointing accuracy
- Investigate the precision gain of a visual feedback system such as the E.A.G.L.E. framework
- Determine if there is a learning effect for non-feedback gestural interaction after having used a visual feedback system for a certain time

Accordingly it was decided to perform a combination of cross-evaluation, which allows testing of the latter two aspects and a questionnaire that amongst other things asked for perceived precision.

The evaluation was performed with 20 subjects between 22 and 65 years and a median age of 27 years. The average experience with gesture input was rated as 5.75 on a scale of 1 (no experience) to 10 (very experienced). They were required to aim and select a sequence of eight different targets of different size that were placed in a room sized 460 cm in length, 260 cm in width and 260 cm in height. These targets were made up of numbered paper sheets placed in the room. A successful selection was assumed by a target being aimed at continuously for two seconds. During the non-feedback run the subjects had to count manually, when they were supposedly aiming successfully, during the feedback run a quick blinking and snap-to-target indicated selection. Overall, each group could select up to 80 targets. Group one (G1) was first testing assisted, group two (G2) first unassisted.

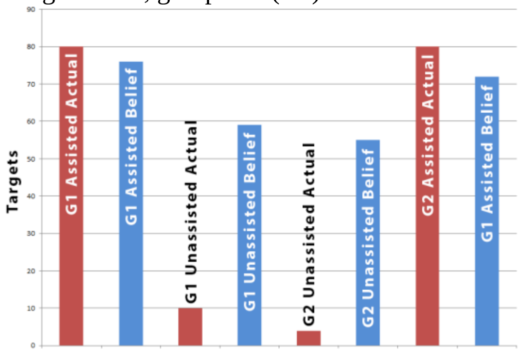

Figure 10. Selection accuracy for assisted and unassisted PFS evaluation - actual and belief

Figure 10 is showing the results of evaluation and questionnaire. The visual assistance system allowed both groups to successfully select all targets - without assistance the success rate is reduced severely to 8.75%. The perceived accuracy is in unassisted cases overestimated strongly, with subjects expecting to have selected 71.25% of all targets. This is a strong indicator of the existence of the offset between perceived and actual pointing accuracy. The underestimation of successful selection in assisted runs is difficult to explain since there was a visual cue (faster blinking) in this case. We assume that this is the lack of experience with such systems that is causing this effect.

While there is a difference in unassisted trials regarding the success rate it is not conclusive to either verify or disregard the learning effect for non-feedback gestural interaction. The number of successfully selected targets was too low overall to allow such a statement.

VII. Conclusion and Future Work

Above we have presented and successfully evaluated an intuitive solution to support the gesture-based control of intelligent environments. The visual feedback given by the laser of the E.A.G.L.E. Framework allows an exact selection of small and spatially close reactive elements. Basic drawbacks in pointing gesture recognition are compensated by providing an appropriate visual feedback, according to the current gesture. Like shown in our evaluation the accuracy in selecting a specific bounding volume is at 100%. Furthermore, we observed an effect by first training the pointing gestures using the laser as feedback and then shutting the laser off and perform similar gestures. The overall number of successfully selected targets in unassisted trials was small and thus it is necessary to validate these findings in a more focused study. Using the generic approach of the E.A.G.L.E. Framework it can be easily integrated into new environments and arbitrary positions of the laser.

During the work for this paper several potential future improvements have been identified. A main disadvantage is limitations of the currently available hardware. Microsoft's Kinect is currently not able to track fingers and therefore a clear pointing direction is harder to achieve. In this context it should also be feasible to evaluate various options for adapting the PFS process automatically to different postures to counteract challenges, such as parallax and mental ray offset, for example by tracking user posture more precisely and using the distance to the targeted object for parallax calculation. The servos that were used to position the laser-spot are limited in resolution and covered degree. The second group of open issues is related to software limitations. The current status of the user is a main issue to pay respect in future iterations. A central aspect is here to make sure that the laser does not point into the eyes of the user or other persons present. Furthermore, there is a need to improve the task of capturing the environment - that is the absolute position of all reactive devices - in a more generic way. Finally we need to compare the visual support of pointing gestures with other feedback alternatives such as acoustic signals.

However, we have shown that the E.A.G.L.E. system is already able to significantly improve the exactness of device selection by using pointing gestures. Further steps will help to improve the user experience and make control of the system even easier.

References

[1] M. Weiser, "The Computer for the 21st Century," *Scientific American*, vol. 265, no. 3, pp. 94-104, 1991.

[2] A. Valli, "The design of natural interaction," *Multimedia Tools and Applications*, vol. 38, no. 3, pp. 295-305, 2008.

[3] E. Aarts, "Ambient intelligence drives open innovation," *interactions*, vol. 12, no. 4, p. 66, Jul. 2005.

[4] Massachusetts Institute of Technology, "Oxygen project." [Online]. Available: http://oxygen.lcs.mit.edu/.

[5] S. Wright and A. Steventon, "Intelligent spaces — the vision , the opportunities and the barriers," *BT Technology Journal*, vol. 22, no. 3, pp. 15-26, 2004.

[6] M. L. Heilig, "Sensorama simulator," *US Patent 3050870*. Patent 3050870. US Patent Office, 1962.

[7] I. E. Sutherland, "Sketchpad: A man-machine graphical communication system," *Afips Conference Proceedings*, vol. 2, no. 574, pp. 329-346, 1963.

[8] G. D. Kessler, L. F. Hodges, and N. Walker, "Evaluation of the CyberGlove as a whole-hand input device," *ACM Transactions on Computer-Human Interaction*, vol. 2, no. 4, pp. 263-283, 1995.

[9] J. C. Lee, "Interaction Techniques Using The Wii Remote Nintendo Wii," *Applied Sciences*, 2008.

[10] J. Shotton et al., "Real-time human pose recognition in parts from single depth images," *Cvpr 2011*, pp. 1297-1304, Jun. 2011.

[11] A. Wilson and S. Shafer, "XWand: UI for intelligent spaces," in *Proceedings of the ACM Conference on Human Factors in Computing Systems*, 2003, no. 5, pp. 545-552.

[12] A. Wilson and H. Pham, "Pointing in intelligent environments with the worldcursor," in *INTERACT International Conference on HumanComputer Interaction*, 2003.

[13] G. A. Miller, E. Galanter, and K. H. Pribram, *Plans and the structure of behavior*. Holt, 1960, p. 226.

[14] D. Kammer, M. Keck, G. Freitag, and M. Wacker, "Taxonomy and Overview of Multi-touch Frameworks: Architecture, Scope and Features," *Architecture*, pp. 1-5, 2010.

[15] T. Akenine-Möller, E. Haines, and N. Hoffman, *Real-Time Rendering*, vol. 85. AK Peters, 2008, p. 1045.