Are You Referring to Me? - Giving Virtual Objects Awareness

René Stingl*
Human-Computer Interaction
University of Würzburg

Chris Zimmerer†
Human-Computer Interaction
University of Würzburg

Martin Fischbach‡
Human-Computer Interaction
University of Würzburg

Marc Erich Latoschik§
Human-Computer Interaction
University of Würzburg

Figure 1: A user selects an object using a speech & gesture interface (MMI) in virtual reality. On the left, non-verbal deixis is determined via ray-casting. On the right, non-verbal deixis is determined via the technique introduced in this work.

Abstract

This work introduces an interaction technique to determine the user's non-verbal deixis in Virtual Reality (VR) applications. We tailored it for multimodal speech & gesture interfaces (MMIs). Here, non-verbal deixis is often determined by the use of ray-casting due to its simplicity and intuitiveness. However, ray-casting's rigidness and dichotomous nature pose limitations concerning the MMI's flexibility and efficiency. In contrast, our technique considers a more comprehensive set of directional cues to determine non-verbal deixis and provides probabilistic output to tackle these limitations. We present a machine-learning-based reference implementation of our technique in VR and the results of a first performance benchmark. Future work includes an in-depth user study evaluating our technique's user experience in an MMI.

Index Terms: Human-centered computing—Interaction paradigms; Human-centered computing—Virtual reality; Computing methodologies—Machine learning

1 Introduction

Multimodal Interfaces (MMIs) that are based on the users' natural communication skills support the potential simultaneous use of at least two input modalities [11]. Speech and gestures are a powerful combination in Virtual Reality (VR) [8], especially for tasks where users have to refer to virtual objects (see Fig. 1 for an example). Speech excels at providing semantically rich information like visual appearances, while gestures can express extensive references to positions (deixis), shapes (iconics), or movements (kinemimics). Given this complementarity, such MMIs offer at least theoretically effective and flexible interactions [12].

However, concrete design choices often restrict these advantages in practice. Ray-casting is one noteworthy choice to determine non-verbal deixis (see Fig. 1, left), e.g., as done by [18, 19]. It is an interaction technique where users have to point directly at the target object with a tracked object, e.g., a VR controller, so that the ray it emits intersects with the visual representation of the target accurately. Despite its simplicity and intuitiveness [2], it poses limitations regarding the MMIs' flexibility and effectiveness:

(L_{Flex}) Ray-casting is a comparatively rigid interaction technique since users can only use the provided ray in the intended manner to point. It disregards additional non-verbal directional cues such as body posture and viewing direction.

(L_{Effect}) MMIs have the unique potential to handle and resolve ambiguous and error-prone interactions by jointly analyzing inputs from multiple modalities [4]. We can use speech to make gestures more robust and effective [7]. In the example of Fig. 1, the MMI can improve object inference by merging the verbally given information "cherry" with the user's nonverbal deixis. However, ray-casting is not compatible with this approach, due to its dichotomous nature, i.e., whether an object intersects with the ray or not. It does not provide any information about objects in the ray's vicinity if it does not intersect. The MMI is, in this case, essentially unimodal and relies only on the user's speech. Hand and tracking jitter exacerbates this problem by making it difficult to accurately intersect a ray with an object, especially if the target is small, occluded, and/or far away [2].

There are several more sophisticated selection techniques than ray-casting. Expand, for example, uses a sphere-shaped selection volume and additional refinement steps to facilitate object selection [1]. However, these techniques were developed for unimodal interfaces and not for synergistic use with speech in MMIs. They do not provide adequate solutions to the previously mentioned limitations.

In this work, we present an interaction technique designed for use in speech & gesture MMIs in VR. It considers a more holistic set of non-verbal directional cues to determine the user's non-verbal deixis and provides probabilistic output to overcome the limitations L_{Flex} & L_{Effect}. We present the interaction technique's concept, a machine-learning-based reference implementation in VR, and benchmarking results showcasing its feasibility.

2 Interaction Technique

Our technique endows virtual objects with an awareness of whether the user is currently referring to them based on her non-verbal deixis (see Fig. 1, right). In contrast to ray-casting, this assessment considers a more holistic set of directional cues rather than just a single ray. While conceptually all directional cues from head orientation and viewing direction over the finger, hand, arm positions and orientations to body posture shall be considered, this is limited by the utilized tracking systems. The awareness estimates a confidence value in $[0, 1]$, indicating how likely the user's non-verbal deixis

*e-mail: rene.stingl@stud-mail.uni-wuerzburg.de
†e-mail: chris.zimmerer@uni-wuerzburg.de
‡e-mail: martin.fischbach@uni-wuerzburg.de
§e-mail: marc.latoschik@uni-wuerzburg.de

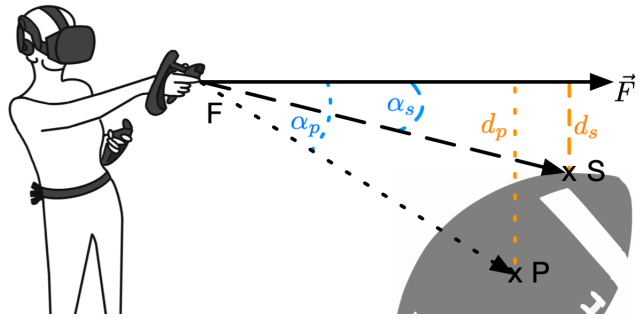

Figure 2: Example of the calculated input features for the right index finger and a virtual football (see Sect. 3 for details).

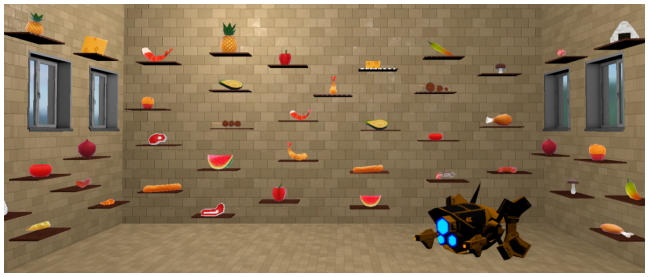

Figure 3: VR environment used for data recording.

is referring to, based on a predefined set of directional cues. This approach aims to overcome the aforementioned limitations: the user has much more freedom in choosing appropriate non-verbal directional cues to indicate non-verbal deixis and is not limited to accurately intersecting a single ray (L_{Flex}). Sorting objects based on these confidence values yields an n-best guess list of referred-to objects that can be merged with the information derived from the user's speech to improve the interface's effectiveness (L_{Effect}).

3 Reference Implementation

In the following, we provide a brief overview of our technique's reference implementation in VR. It is implemented in the game engine Unity3D 2019.4.20f1 [15] with the Steam VR plugin [17] and consists of the following aspects:

Tracking: We use an HTC Vive Pro Eye head-mounted display to track the user's head and eyes, Valve Index controller for hands and fingers, and a belt with an attached HTC Vive tracker for the torso. The Vive Eye and Facial Tracking SDK [3] provides access to the HMD's eye tracker. This configuration is capable of tracking the following sub-set of non-verbal directional cues: position and orientation of the user's eyes, head, hands, fingers, and torso.

Calculating Input Features: Based on the tracked directional cues, we calculate input features for each object in the virtual environment. Fig. 2 illustrates these calculations using the right index finger as an example. The ray $\vec{F}$ (black solid arrow) represents the user's pointing direction based on the index finger's position F and its forward vector. Point S determines the closest point on the object's surface to the ray $\vec{F}$, while point P represents the virtual object's pivot point. We calculate the shortest distance d_s (orange dashed line) from point S to ray $\vec{F}$ as well as the shortest distance d_p (orange dotted line) from point P to ray $\vec{F}$. Further, we calculate the angle α_s (blue dashed curve) between ray $\vec{F}$ and ray $\overrightarrow{FS}$ (black dashed arrow) as well as the angle α_p (blue dotted curve) between ray $\vec{F}$ and ray $\overrightarrow{FP}$ (black dotted arrow). In total, these calculations are performed for seven tracked directional cues, i.e., the user's viewing direction, head, hands, index fingers, and torso. This results in a set of 28 features per object: 14 distances D and 14 angles A.

Table 1: Comparison of different machine learning models.

Method	F1-Score	Precision	Recall
FNN (ours)	**94.46%**	**91.26%**	97.88%
SVC	93.91%	90.22%	**97.91%**
LR	93.14%	90.08%	96.42%

Recording Data: We implemented a recording module that stores all input features with a respective timestamp for each frame in a local .csv file. It records a video using the Open Broadcast Software [10], which captures the virtual environment from the user's perspective and the user's speech.

We created a simple Unity VR scene of a room containing differently shaped virtual objects placed on wall-mounted shelves (see Fig. 3). The task was to instruct a virtual agent, i.e., a flying drone situated in the virtual environment, to retrieve predefined objects from the shelves. We disclosed that the experimenter controlled this agent and thus can understand everything a human does. Participants were instructed to interact as natural as possible and were free to use speech and other non-verbal directive cues. In total, 11 participants (five male and six female, all members of a human-computer interaction chair with ample VR experience) performed 20 tasks, each with different target objects while our recording module captured video, audio, and the calculated features resulting in a data set with 220 interactions. We sorted out 30 interactions due to technical issues or participants relying solely on speech input with no visible head, eye, and body movements, leaving 190 interactions in our final data set.

Labeling Data: We used the annotation tool ELAN [9] to label the recorded videos manually. Time intervals with correct interactions were labeled positive, while incorrect or no interactions were labeled negative. These timestamps are fused with the recorded features to generate a labeled dataset. The labeled dataset contains a total of 77077 rows of data (15775 positive & 61302 negative).

Preprocessing: We balanced the labeled dataset by randomly removing 45527 negative labeled rows of data. Further, we analyzed the features in terms of their importance using scikit-learn [14]. The result revealed that the features regarding the torso have the least added value determining the user's non-verbal deixis. We sorted out the angles and distances of the torso, leaving 24 out of the 28 features. Thus, the preprocessed dataset contains 31550 rows of data (15775 positive & 15775 negative) with 24 features each.

Training and Testing: We implemented a feed-forward neural network with 24 nodes in the input layer, 16 nodes in the hidden layer (two-thirds of the nodes in the input layer [5]), and one node in the output layer using PyTorch [13]. The hidden layer uses the ReLU activation function, while the output layer uses the sigmoid function to transform the output values in $[0, 1]$. We used 80% of our dataset to train the network for 35 epochs with a learning rate $\alpha = 0.001$ and batch size 8 using the Adam optimizer [6]. Since the task is a binary classification problem, we use the Binary Cross-Entropy loss function. The trained network is exported to .onnx format.

We compared the neural network with *Logistic Regression* (LR) and *Support Vector Classifier* (SVC), also implemented in the scikit-learn package [14], using the remaining 20% of our dataset. The neural network performed best in the overall F1-score and Precision (see Table 1). Thus, we chose the neural network as our classifier.

Predicting: We use the Barracuda [16] plugin to use the exported neural network as a component in Unity. This component can be attached to every virtual object with a mesh (collider) component. It predicts the likelihood that the user is currently referring to the object in each frame. This likelihood is encoded in a confidence value in $[0, 1]$. The MMI can retrieve these confidence values for each respective object to recognize and interpret the user's interaction.

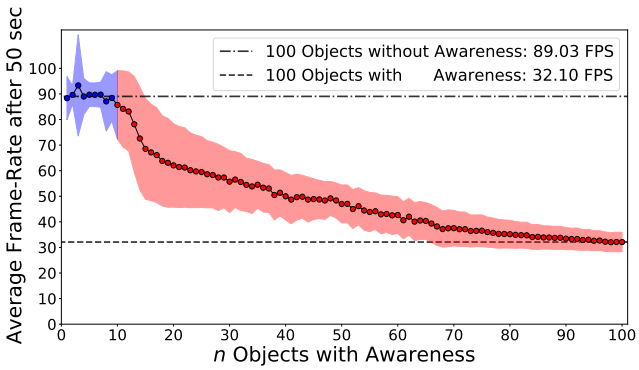

Figure 4: Frame rates (mean & standard deviation) depending on the number of awareness modules running simultaneously.

4 Performance Benchmark

We used Unity3D 2019.4.20f1 and the aforementioned hardware to create a VR application with 100 virtual objects to serve as a test environment for our performance benchmark. The benchmark measured frames per second (fps) while incrementally adding the awareness modules to the virtual objects resulting in a total of 101 measurements, i.e., average fps and standard deviation. For each measurement, the application was restarted and the respective amount of modules was added to the virtual objects. After a ten-second delay, the fps was captured and logged for 50 seconds. We conducted the benchmark on a PC equipped with an NVIDIA GeForce GTX 1080 Ti graphics card, an Intel (R) Core (TM) i7-8700K 3.70 GHz processor, and 16 GB of DDR4 memory running Windows 10. Barracuda was configured to CSharpBurst mode.

Fig. 4 depicts the results of all 101 runs. The baseline fps with 100 virtual objects without an awareness module was 89.03 (*SD*=6.80). The application ran stable until nine simultaneously active modules where we measured an fps of 88.41 (*SD*=9.09). However, the performance started to decline from the tenth object onwards with an fps of 32.10 (*SD*=3.83) with 100 modules.

5 Conclusion

We presented an interaction technique for determining the user's non-verbal deixis in VR applications. We designed it for the use in speech and gesture MMIs, where it provides an alternative design choice to the commonly used ray-casting. It addresses the limitations of rigid and dichotomous interaction techniques in this context and aims to increase the MMI's overall flexibility and effectiveness. This work showcased the technique's concept, a machine-learning-based reference implementation in VR, and validated its feasibility with a first performance benchmark. We identify the following limitations that open up space for subsequent future work: (1) During data collection, it seemed that some participants were unsure of how to interact, which later made it difficult to determine correct interactions when labeling the data. We need a more refined environment for data collection that encourages the user to interact as naturally as possible and enables clear labeling of the data. (2) The benchmark results show that the calculations of the awareness modules of each frame eventually lead to significant performance degradation. Performing these calculations only every second or third frame may yield a significant performance gain without negatively affecting the classifier's accuracy. However, the current reference implementation is sufficient for first investigations of our technique's user experience when combined with speech input in an MMI before addressing these limitations.

References

[1] J. Cashion, C. Wingrave, and J. J. LaViola Jr. Dense and dynamic 3d selection for game-based virtual environments. *IEEE Transactions on Visualization and Computer Graphics*, 18(4):634–642, 2012.

[2] G. d. Haan, M. Koutek, and F. H. Post. IntenSelect: Using Dynamic Object Rating for Assisting 3D Object Selection. In E. Kjems and R. Blach, editors, *Eurographics Symposium on Virtual Environments*. The Eurographics Association, 2005.

[3] HTC Corporation. VIVE Eye and Facial Tracking SDK. Retrieved June 14, 2022, from https://developer.vive.com/resources/vive-sense/eye-and-facial-tracking-sdk/.

[4] E. Kaiser, A. Olwal, D. McGee, H. Benko, A. Corradini, X. Li, P. Cohen, and S. Feiner. Mutual disambiguation of 3d multimodal interaction in augmented and virtual reality. In *Proceedings of the 5th International Conference on Multimodal Interfaces*, ICMI '03, pages 12–19, New York, NY, USA, 2003. Association for Computing Machinery.

[5] S. Karsoliya. Approximating Number of Hidden layer neurons in Multiple Hidden Layer BPNN Architecture. *International Journal of Engineering Trends and Technology*, page 4, 2012.

[6] D. P. Kingma and J. Ba. Adam: A Method for Stochastic Optimization. 2014.

[7] M. Latoschik, M. Frohlich, B. Jung, and I. Wachsmuth. Utilize speech and gestures to realize natural interaction in a virtual environment. In *IECON '98. Proceedings of the 24th Annual Conference of the IEEE Industrial Electronics Society (Cat. No.98CH36200)*. IEEE, 1998.

[8] M. E. Latoschik. A general framework for multimodal interaction in virtual reality systems: Prosa. In *The Future of VR and AR Interfaces-Multimodal, Humanoid, Adaptive and Intelligent. Proceedings of the Workshop at IEEE Virtual Reality*, number 138, pages 21–25, 2001.

[9] Nijmegen: Max Planck Institute for Psycholinguistics, The Language Archive. ELAN (Version 5.9), 2020. Computer software. https://archive.mpi.nl/tla/elan.

[10] OBS Studio Contributors. Obs studio (version 26.0.2), 2020. Retrieved June 14, 2022, from https://obsproject.com/de.

[11] S. Oviatt and P. Cohen. Perceptual user interfaces: multimodal interfaces that process what comes naturally. *Communications of the ACM*, 43(3):45–53, Mar. 2000.

[12] S. Oviatt, B. Schuller, P. Cohen, D. Sonntag, and G. Potamianos. *The Handbook of Multimodal-Multisensor Interfaces, Volume 1: Foundations, User Modeling, and Common Modality Combinations*. Association for Computing Machinery and Morgan & Claypool, 2017.

[13] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala. PyTorch: An Imperative Style, High-Performance Deep Learning Library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., 2019.

[14] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[15] Unity Technologies. Unity3D 2019.4.20f1, 2019. Retrieved June 14, 2022, from https://unity3d.com/get-unity/download/archive.

[16] Unity Technologies. Barracuda 1.0.4, 2020. Retrieved June 14, 2022, from https://github.com/Unity-Technologies/barracuda-release/tree/release/1.0.4.

[17] Valve Corporation. SteamVR Plugin, 2021. Retrieved June 14, 2022, from https://store.steampowered.com/app/250820/SteamVR/.

[18] E. Wolf, S. Klüber, C. Zimmerer, J.-L. Lugrin, and M. E. Latoschik. "paint that object yellow": Multimodal interaction to enhance creativity during design tasks in VR. In *2019 International Conference on Multimodal Interaction*. ACM, Oct. 2019.

[19] E. Zudilova, P. Sloot, and R. Belleman. A multi-modal interface for an interactive simulated vascular reconstruction system. In *Proceedings. Fourth IEEE International Conference on Multimodal Interfaces*. IEEE Comput. Soc, 2002.