WikNectVR: A Gesture-Based Approach for Interacting in Virtual Reality Based on WikNect and Gestural Writing

Vincent Kühn(✉), Giuseppe Abrami, and Alexander Mehler

Texttechnology Lab, Goethe University Frankfurt, Robert-Mayer-Strasse 10, 60325 Frankfurt am Main, Germany
vincent.kuehn@stud.uni-frankfurt.de, {abrami,mehler}@em.uni-frankfurt.de, https://www.texttechnologylab.org

Abstract. In recent years, the usability of interfaces in the field of *Virtual Realities* (VR) has massively improved, so that theories and applications of multimodal data processing can now be tested more extensively. In this paper we present an extension of VAnnotatoR, which is a VR-based open hypermedia system that is used for annotating, visualizing and interacting with multimodal data. We extend VAnnotatoR by a module for *gestural writing* that uses data gloves as an interface for VR. Our extension addresses the application scenario of WikiNect, a museum information system, and its gesture palette for realizing gestural writing. To this end, we implement and evaluate seven gestures. The paper describes the training and recognition of these gestures and their use within the framework of a user-centered evaluation system for virtual museums as exemplified by WikiNect.

Keywords: Gestural writing · WikiNect · VAnnotatoR · Multimodal annotation · Virtual Reality

1 Introduction

Recent developments of 3D glasses and other interfaces to *Virtual Realities* (VR) has enabled theories [11,12] and implementations of corresponding applications [15]. Such applications allow users to immerse themselves in virtual action contexts and scenarios. To achieve this immersion, the corresponding interfaces must be as intuitive as possible. This is no trivial task, as Erra et al. [3] make clear given the 3D glasses currently available. These glasses are controlled by hand controls, whose buttons can be assigned various functions. However, the use of such controllers leads to a limitation of the learning curves of users (due to a lack of intuitive handling) and the number of executable controls is limited by

J. Y. C. Chen and G. Fragomeni (Eds.): HCII 2020, LNCS 12190, pp. 299–312, 2020.
https://doi.org/10.1007/978-3-030-49695-1_20

the hardware. While such controllers may be sufficient for common applications, they are largely unsuitable for immersive control that requires the free use of hands for gesture formation.

A more realistic perspective is achieved by evaluating gestures or body movements to control applications. In this paper we present such a gestural control in the area of VR systems. More specifically, we describe an extension of VANNOTATOR [1,10,14] by means of a gesture-based control that uses data gloves as an interface for VR. VANNOTATOR is an open hypermedia system designed for annotating, visualizing and interacting with multimedia content (texts, images, their segments and relations, videos, audios, 3D objects, 3D buildings, etc.). Starting from the notion of *gestural writing* as introduced in [12] we develop a writing system that allows users to navigate and interact with objects in VANNOTATOR just by using gestures. Gestural writing means to use only indexical or iconic gestures to gesture parts of propositional acts and thus to make assertions about the reference objects involved. Analogous to pictograms, Mehler et al. [12] call such gestures *gestograms*. That is, gestograms are iconic or indexical gestures with a referential or predicative meaning. In this paper, we implement and test a subset of the gestures proposed in [12] for gestural writing. We describe the training and recognition of these gestures and their use within the framework of a user-centered evaluation system for virtual museums. This application scenario is based on WIKINECT [11], a museum information system that here is reimplemented in VR.

The paper is organized as follows: Sect. 2 gives a brief overview of related work. Sect. 4 describes the architecture of our implementation of gestural writing. Sect. 6 provides an evaluation of our system. Finally, Sect. 7 gives a summary and an outlook on future work.

2 Related Work

The gesture-based control of software is not a new topic. In this area there are a number of projects as described by Xue et al. [18]. Several of these projects use *Kinect* (c.f. [19]). Jambursia et al. [6], for example, describe a system for recognizing hand signs with *Kinect*. An alternative is proposed by Drossis et al. [2] who describe a system for navigating in virtual environments by the help of *Leap Motion*. Marin et al. [9] present a combination of *Leap Motion* and *Kinect*. It utilizes a multi-class SVM classifier without using 3D glasses. However, detection is not limited to data gloves or sensors such as those provided by Leap Motion. Ghosh and Ari [4] use image recognition techniques to detect gestures. The use of data gloves as hand-related data input for neural networks is described in [13,17]. As far as we know, there is no implementation yet that supports gestural writing in virtual environments, so it is worth taking this step, as explained in the next sections.

3 Application Context

The application scenario on which our implementation of gestural writing builds is WikiNect [12]. WikiNect is a museum information and annotation system that helps museum visitors to document and evaluate or even recommend their museum visit on site. WikiNect serves, among other things, to replace analog, usually offline communication in this area by digital online communication. In this context, Mehler et al. [12] developed a theory for the formation of gestures on the basis of Lakoff's theory of image schemata [8]. These gestures serve to implement gestural writing with the task of enabling the latter type of online communication. While WikiNect is designed as an information system that is used on site with *Kinect*, we opted for VAnnotatoR as the environment for transferring WikiNect to VR, henceforth called WikiNectVR. For this extension of VAnnotatoR, seven gestures were selected and implemented using data gloves (*Hi5*[1]). By the use of these gloves a more exact assignment of hand positions is possible. Compared to visual recognition systems such as *Kinect* and *Leap Motion*, these positions can also be recognized without direct visual contact. The usage of gestures is promising in several respects:

1. The control of software can be made more flexible and intuitive through gestures. Whether you need to navigate in a virtual browser, select, segment or link objects, all these operations can be simplified by gestures: selections are then made by indexical gestures, links are established by drawing edges from the respective source to the target, segmentations by drawing polygons or ellipses etc.
2. Even the input and use of linguistic information can be accelerated. In WikiNectVR, users can trigger Natural Language Processing (NLP) gesturally. For this purpose, WikiNectVR uses the NLP interface of VAnnotatoR for TextImager [5] to preprocess text input in a variety of languages. As a consequence, users may gesturally refer to a visual depiction of a discourse referent (e.g. a person) in order to automatically select all text segments referring to it.

As a result of these enhancements, WikiNectVR as a prototype allows its users to annotate, segment, link and evaluate virtualized museum tours and their virtualized artifacts and finally to network in online social communities for this purpose. As a special application scenario for testing gestural writing in WikiNectVR, we focus on the evaluation and rating system of WikiNect: The evaluation system in WikiNect allows users to rate a picture given one or two categories (e.g. emotions or style). In comparison to other ratings systems the results are displayed in a different way and show up as a new picture. The style is based on the paintings of Piet Mondrian. A picture is vertically and horizontally divided with black lines and each of the resulting rectangles is filled with a color. The rectangles may vary in size depending on the salience of the underlying evaluation. In WikiNect the size and the color of a rectangle is

[1] https://hi5vrglove.com/.

given through the user's answers in the rating system. The implementation of WikiNectVR is realized with Unity3D[2] as described in the following section.

4 Implementation

A first implementation and evaluation of WikiNectVR's gesture palette is described in [7]. The implementation contains two parts: gesture learning and gesture recognition. We start with explaining the second step.

4.1 Gesture Recognition

We use a feed-forward neural network based on [16] for gesture recognition. It uses all information collected from data gloves to recognize the corresponding gesture based on hand positioning data. In order to generalize input data, we perform preprocessing. To this end, we represent each finger by two vectors: The first one starts at the wrist, the second one at the basal finger joint, both vectors end at the fingertip. By these two vectors, we represent and distinguish the different finger orientations. In addition to the fingers' individual positions, the position and rotation of the respective hand is processed. Rotation data is represented in the normal vector of the hand. The vectors are collected by an agent as the binding element between the neural network and the rest of the gesture recognition system. Since learning is done with policy optimization, the agent/network is rewarded or punished with the same value as it has shown for the best results.

When a gesture is generated by the user, it is recognized by the network and the result is transmitted to WikiNectVR's *GestureController*. Since the gesture recognition takes place in real time, the controller is constantly confronted with new gestures. In order to prevent a gesture from being triggered unintentionally, the controller verifies whether or not past decisions contained the same gesture. In this way, we reduce the probability of false positives.

4.2 Gesture Learning

Since neural networks can hardly be trained in real time, the gestures to be learned must be collected in a training corpus. For this purpose, we generated the so-called *Frankfurt GESTural writing corpus* (FGEST) which is available via GitHub[3].

It stores each individual gesture token together with its hand and finger orientation and rotation data as required by our neural network for training. FGEST recorded and represented 300 tokens for each gesture to be learned where these tokens have been generated by five different persons. In principle, this approach makes it possible to calculate different models or classifiers for

[2] https://unity.com.
[3] https://github.com/texttechnologylab/FGEST.

the same or different gestures. We used a learning rate of $3e-4$ as the policy optimization showed the best results compared to a smaller one (e.g. $1e-5$) An LSTM was also used, but did not achieve as good learning outcomes as the Feed Forward network. This may be because the LSTM takes more time to perform a step and therefore requires longer training times (Fig. 1).

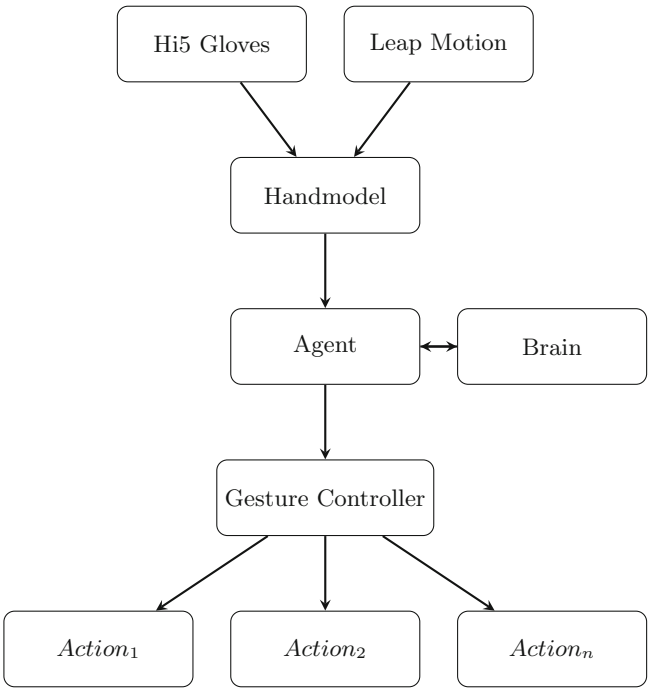

Fig. 1. Overview of the modules needed for the gesture recognition.

Gestures. Starting from the gesture palette described in [12], we trained and evaluated seven gestures (Table 1). First of all, this relates to the Point gesture (henceforth denoted by →) that is implemented to allow for selecting objects. To gesturally indicate a user's wish to go forward or backward within the rating system of WikiNectVR, and, thus, to enter the rating of the next or the last object, we introduce two gestures: the GreaterThan (>) and the SmallerThan (<) gesture. Both are created in front of the user's torso where the SmallerThan gesture inverts the direction into which the GreaterThan gesture is created (Fig. 2).

The search function requires two gestures to trigger searches in a search context. To this end, we implement the so-called IsEqual (==) and the IsUnequal (!=) gesture. The IsEqual gesture is formed with both hands, which iconically represent an equal sign. The IsEqual gesture is performed by positioning the flat palms orthogonally to each other (Fig. 3).

With these gestures a (museum) visitor of WikiNectVR can select pictures (P), artists (A) or museums (M) by means of the Point gesture and the gestures described before. Reference objects of a pointing gesture can denote classes (e.g. the class of paintings $\mathbb{P}$) or single entities (e.g. a concrete person such as an artist A_i). As there are different application scenarios which the user can select he needs a possibility to switch between them. This is done with the so called AND gesture.

Table 1. Gestures implemented in WikiNectVR. The functions in italics have been implemented only in the WikiNectVR for the specified purpose.

Gesture	Notation	Function
GreaterThan	$>$	go forward (rating system scenario)
SmallerThan	$<$	go backward (rating system scenario)
Negation	!	*deselect the active rating field (rating system)*
IsEqual	==	*search the cutting quantity of objects*
IsUnequal	!=	*search the symmetric difference of the objects*
And	$\wedge$	*switch between the museum graph and the search interface*
Point	$\rightarrow$	click on an object or select it

5 Two Applications

5.1 WikiNectVR's Graphical Navigation System

For evaluating the gestures distinguished so far, two use cases were implemented. As described in Sect. 1 the visualization and interaction with virtual objects is evaluated and tested using VR. For this purpose, a scenario was used from the museum context. To this end, museum content was visualized as a graph in VR (*museum graph*). The data refers to paintings, their artists, current exhibition venues and data about artists and their paintings. In this simplified virtual museum, visitors can view paintings and associated information, such as *label*, the *year of creation* and *artist* (Fig. 6).

At the same time, pictures of single artists can be selected. By visiting the virtual museum, a graph is created that enables various possibilities of interaction with the images. On the one hand, all functionalities described in [10], such as image segmentation and annotation, can be performed. On the other hand, the virtual paintings, based on WikiNect, can be rated along different categories. The latter is described in Sect. 5.2. The interface is controlled by gestures that are listed in Table 1 and can be assigned to various operations. In accordance with the implemented gesture set, various operations were defined.

As a visitor of this virtual museum, the user is positioned within the virtual museum graph, in which he or she can switch back and forth between individual nodes. The graph itself is projected onto the virtual museum floor, where neighboring nodes at a distance of 2 are displayed as shown in Fig. 5. By selecting a

node using the GreaterThan gesture, the user moves to the target node. At the same time, it is possible to search within the graph using the AND gesture.

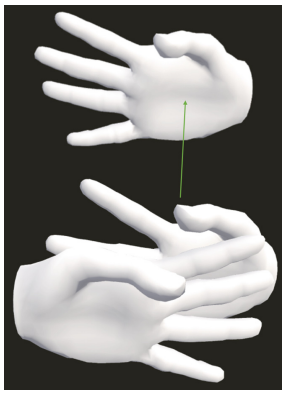

Fig. 2. Visualization of the GreaterThan gesture.

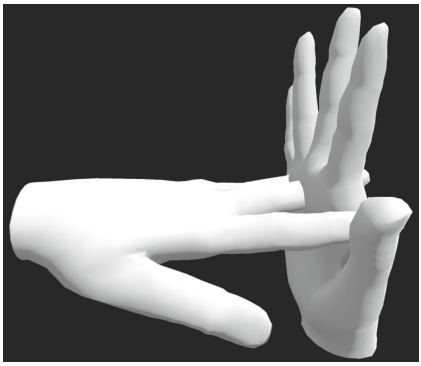

Fig. 3. Visualization of the IsUnequal gesture.

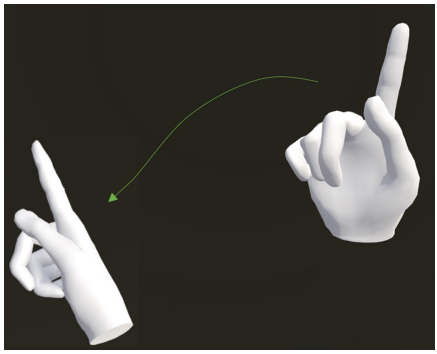

Fig. 4. Visualization of the And gesture.

The search interface is also represented through a graph structure. When a search is initialized, representations of searchable objects types (e.g. images or painters) show up around the user as nodes (Fig. 7). Afterwards there are several possibilities to use this view: The user can select certain objects for which additional information is to be displayed. Or a class of object is referred to, e.g. artists, so that all instances of this class are selected. After selecting a group objects (see Fig. 6) it is possible to make further selections in the resulting context by means of corresponding search gestures (Fig. 4).

5.2 WikiNectVR's Rating Interface

In addition to being able to obtain information about paintings, users can also rate them. The rating interface, which can be used for any object in the virtual museum (and not only for images), basically implements the interface model of [12]. The rating interface utilizes three gestures. The first one is the POINT gesture for object selection. The next two gestures are the GREATERTHAN and the SMALLERTHAN gesture. They allow users to move forward or backward. Before rating an object, the user has to select it together with at least two rating categories for spanning a matrix (see Fig. 8). Each rating is visualized through a cube showing a Mondrian-style painting on its surface. They are placed in front of a partially transparent image of the object to show the link between an object and its rating. The ratings are made visible by the color and size of the fields on the surface of the cube (for example, a larger field in the upper right area means that more good ratings were given).

When the user has made his or her choice, the valuation interface appears together with a row or matrix showing the valuation gradations (see Fig. 10). The Mondrian-style painting on the right-hand side (see Fig. 9) gets updated and the next evaluation criterion is displayed. If the user needs to review or change his previous ratings, he can return to the relevant criterion and update it. Once

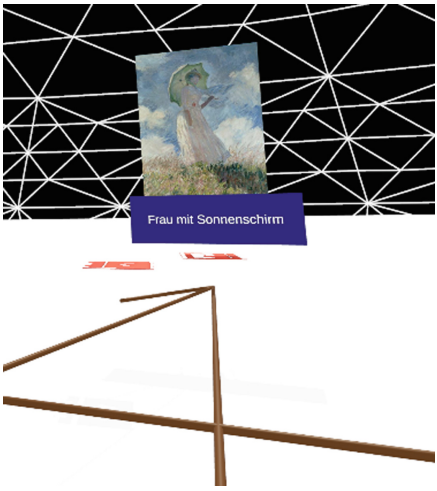

Fig. 5. Visualization of the virtual museum. The user is located relative to the node as part of the virtual museum graph representing an artist and can see one of his paintings.

Fig. 6. Further information about the selected painting can be displayed; in this case, the artist, the year of creation and the museum in which the painting is exhibited are shown. Red highlighted attributes were selected using the POINT gesture and joined as described in Sect. 5 using the EQUAL gesture. (Color figure online)

the evaluation is complete, the user has the possibility to see his results or get an overview of all evaluations.

6 Evaluation

We evaluate our gesture recognition tool in the context of gestural writing. To this end, the test persons had to write five test sentences gesturally using our gesture palette. In this case, the gestures are used to connect objects with each other. There are sentences which concern the gestural connection of two objects, and sentences in which three objects are affected. Therefore, the user annotates three sentences with two objects and two sentences with three different objects in the evaluation. For this purpose, the objects must be selected by the annotator and linked with a gesture of our gesture palette.

All objects represented in the annotation scenario can be annotated using the sentences in Table 2.

For illustration we look at the following example: “The armchair is bigger than the chair" (see Fig. 11). The annotators are now requested to select the objects *armchair* and *chair* and link them using the GREATHERTHAN gesture. The user input is displayed in three blue input fields, with a gesture expected in every second field. The input is only changed when the user looks at the specific field to prevent side effects such as incorrectly selected objects or gestures. After the annotator has selected the objects and the gesture, the input is evaluated and visual feedback is given: A correct annotation is visualized in green, a wrong in red.

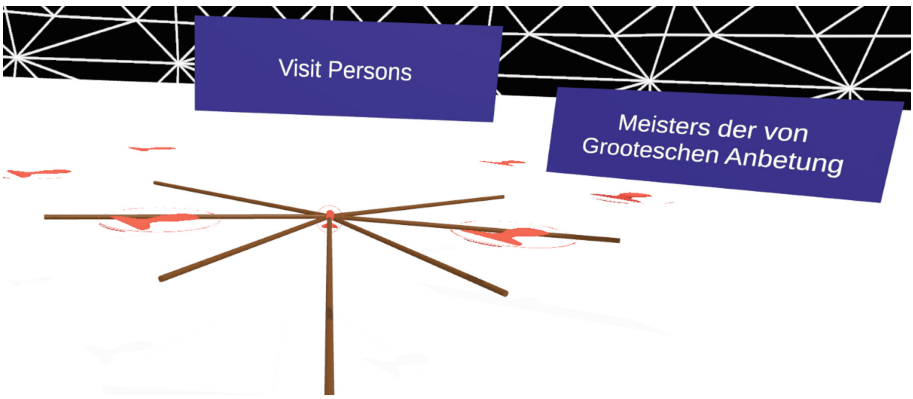

Fig. 7. A view on painters mapped by the *museum graph*. The user initialized the search layout while looking on the painters section.

Fig. 8. To rate a picture it has to be selected using the GreaterThan gesture. The resulting rating view shows the image (transparent) in the background with the available category matrix in the foreground. In this example, five evaluation categories connected with 10 questions, each of which can be used individually or in combination with another categories, are distinguished. The fields in the evaluation matrix are filled with the results of previous evaluations. The results are represented in Mondrian style. A new rating is performed simply by selecting the corresponding element. The diagonal items represent the one-dimensional evaluation categories.

In total, the evaluation included 12 participants, all studying computer science, nine of whom were male and three female. The evaluated parameters are the time it takes for the test person to manifest the sentence and whether he has selected the correct gestures and objects. This distinction is important in order to differentiate whether the selection was wrong or whether the target gesture was not recognized. For result No. 2 (Table 3), it should be noted that not all gestures have the same recognition rate. In addition, for result No. 3, some gestures were not separate enough, so that wrong objects were selected. The best gesture has a rate higher than 92%. The Equal gesture is the one with the lowest rate. This variance between the gestures may result from the training data. Since the Equal gesture is a static gesture, we assume that there was not enough accurate training data. To minimize this type of error, the neural network needs more training data to better distinguish the gestures.

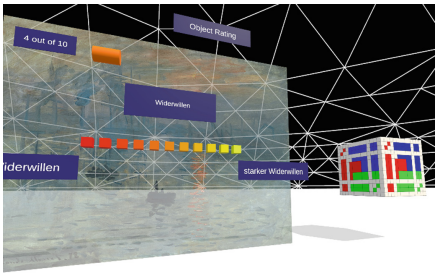

Fig. 9. The visualization of a one-dimensional rating category. The painting to be rated is transparent in the background and the current question is shown in the foreground. Below the rating scale one finds descriptions of the maximums. The scale covers a range from -5 to $+5$ including zero. A progress bar is displayed at the top. On the right side of the rating panel is a Mondrian visualization of the result, which changes with each rating.

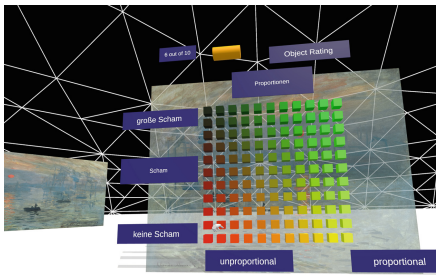

Fig. 10. The rating panel with two categories. The structure is similar to the panel with only one category. The difference, however, is the expansion from one to two scales. On the left side, the image is displayed non-transparent. Not visible, but still present is the result as a Mondrian-style visualization on the right side of the rating panel.

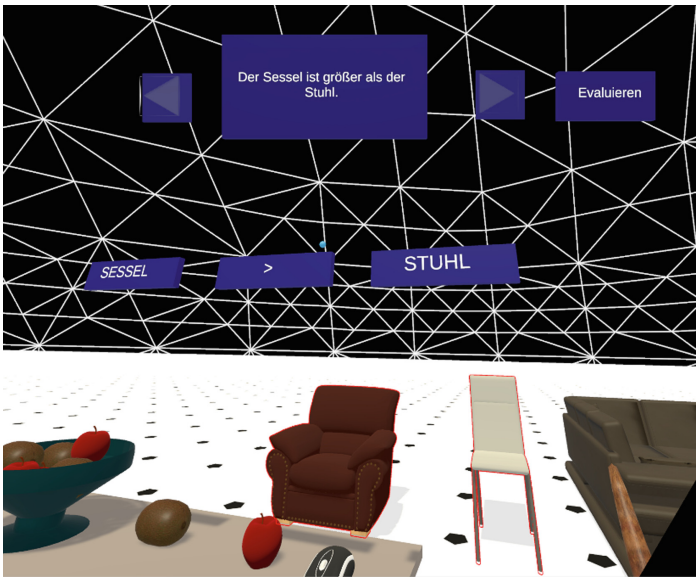

Fig. 11. An example sentence of the evaluation (in German language). Translation: “The armchair is bigger than the chair" and it is written on the big blue panel at the top. Below it, one sees three blue boxes that visualize the selected objects and the gestures of the evaluators. It means that *Armchair* > *Chair*. (Color figure online)

Table 2. Evaluation sentences and the corresponding objects to be selected with the linking gesture.

Sentence	Relation
The *chair* is smaller than the *armchair*	chair < armchair
The *plant* is more expensive than the *mouse*	plant > mouse
The *armchair* is bigger than the *chair*	armchair > chair
The *apple* and the *kiwi* are different	apple ≠ kiwi
The *kiwi* tastes as good as the *apple*	kiwi = apple
The *chair* is not an *armchair*	chair ≠ armchair
The *sofa* is not only larger than the *chair*, but also larger than the *armchair*	sofa > chair & sofa > armchair
The *armchair* is larger than the *chair*, but it is smaller than the *sofa*	sofa > armchair > chair

Table 3. Evaluation results based on 12 participants.

No.	Category	Result
1	Correct sentences reconstructed	70%
2	Correct gestures recognized	78%
3	Correct objects selected	82%

7 Conclusion and Future Work

In this paper, we presented a gesture-based, virtual approach to WikiNect by means of the VAnnotatoR. For recognizing our gesture alphabet, we trained a neural network by using a corpus of 400 data units per gesture. The gesture recognition was then evaluated by means of 12 test persons. As a result of our evaluation, promising results were achieved, although the gestures were not all recognized equally well. To minimize this type of error, the network needs more training data to better distinguish the gestures. The next development steps of our gesture-based control software will therefore be aimed at expanding the amount of training data as well as our gesture alphabet. The corresponding data of this study and the underlying software will be published on GitHub.

Acknowledgement. We are grateful to the *Scholarship Fund for Teaching at the University of Frankfurt* for the contribution of this work.

References

1. Abrami, G., Mehler, A., Spiekermann, C.: Graph-based format for modeling multimodal annotations in virtual reality by means of VAnnotatoR. In: Stephanidis, C., Antona, M. (eds.) HCII 2019. CCIS, vol. 1088, pp. 351–358. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-30712-7_44
2. Drossis, G., Birliraki, C., Margetis, G., Stephanidis, C.: Immersive 3D environment for data centre monitoring based on gesture based interaction. In: Stephanidis, C. (ed.) HCI 2017. CCIS, vol. 713, pp. 103–108. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-58750-9_14
3. Erra, U., Malandrino, D., Pepe, L.: Virtual reality interfaces for interacting with three-dimensional graphs. Int. J. Hum.-Comput. Interact. **35**(1), 75–88 (2019). https://doi.org/10.1080/10447318.2018.1429061
4. Ghosh, D., Ari, S.: A static hand gesture recognition algorithm using k-mean based radial basis function neural network (2011)
5. Hemati, W., Uslu, T., Mehler, A.: Textimager: a distributed UIMA-based system for NLP. In: Proceedings of the COLING 2016 System Demonstrations. Federated Conference on Computer Science and Information Systems (2016)
6. Jambusaria, U., Katwala, N., Kadam, M., Narula, H.: Finger writing in air using kinect. Int. J. Comput. Sci. Inf. Technol. (IJCSIT) **5**(6), 8119–8121 (2014)
7. Kühn, V.R.: A gesture-based interface to VR. Bachelor Thesis, Goethe University of Frankfurt (2018). http://publikationen.ub.uni-frankfurt.de/frontdoor/index/index/docId/50915
8. Lakoff, G.: Women, Fire, and Dangerous Things. University of Chicago press, Chicago (1987)
9. Marin, G., Dominio, F., Zanuttigh, P.: Hand gesture recognition with leap motion and kinect devices. In: 2014 IEEE International Conference on Image Processing (ICIP), pp. 1565–1569 (2014). https://doi.org/10.1109/ICIP.2014.7025313
10. Mehler, A., Abrami, G., Spiekermann, C., Jostock, M.: VAnnotatoR: a framework for generating multimodal hypertexts. In: Proceedings of the 29th ACM Conference on Hypertext and Social Media, Proceedings of the 29th ACM Conference on Hypertext and Social Media (HT 2018). ACM, New York (2018). https://doi.org/10.1145/3209542.3209572
11. Mehler, A., Lücking, A.: WikiNect: Towards a gestural writing system for kinetic museum wikis. In: Proceedings of the 2012 ACM Workshop on User Experience in e-Learning and Augmented Technologies in Education, UXeLATE 20(2012). https://doi.org/10.1145/2390895.2390899
12. Mehler, A., Lücking, A., Abrami, G.: WikiNect: image schemata as a basis of gestural writing for kinetic museum wikis. Univ. Access Inf. Soc. **14**(3), 333–349 (2014). https://doi.org/10.1007/s10209-014-0386-8
13. Murakami, K., Taguchi, H.: Gesture recognition using recurrent neural networks (1991)
14. Spiekermann, C., Abrami, G., Mehler, A.: VAnnotatoR: a gesture-driven annotation framework for linguistic and multimodal annotation. In: Proceedings of the Annotation, Recognition and Evaluation of Actions (AREA 2018) Workshop. AREA (2018)
15. Tecchia, F., Avveduto, G., Brondi, R., Carrozzino, M., Bergamasco, M., Alem, L.: I'm in vr!: using your own hands in a fully immersive MR system. In: Proceedings of the 20th ACM Symposium on Virtual Reality Software and Technology, VRST 2014, pp. 73–76. ACM, New York (2014). https://doi.org/10.1145/2671015.2671123, http://doi.acm.org/10.1145/2671015.2671123

16. Unity Technologies: Unity ML-Agents Toolkit Documentation (2018). https://github.com/Unity-Technologies/ml-agents/blob/master/docs/Readme.md
17. Weissmann, J., Salomon, R.: Gesture recognition for virtual reality applications using data gloves and neural networks (1999)
18. Xue, L., Parker, C.J., McCormick, H.: A virtual reality and retailing literature review: current focus, underlying themes and future directions. In: tom Dieck, M.C., Jung, T. (eds.) Augmented Reality and Virtual Reality. PI, pp. 27–41. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-06246-0_3
19. Zhang, X., Ye, Z., Jin, L., Feng, Z., Xu, S.: A new writing experience: finger writing in the air using a kinect sensor. IEEE MultiMedia **20**(4), 85–93 (2013). https://doi.org/10.1109/MMUL.2013.50