AN AUTOMATIC, ADAPTIVE HELP SYSTEM TO SUPPORT GESTURAL OPERATION OF AN AUTOMOTIVE MMI

Ralf Nieschulz*, Michael Geiger*, Klaus Bengler+, Manfred Lang*

* Institute for Human-Machine Communication
Technical University of Munich, D-80290 Munich, Germany
+ BMW Group, dep. EV-22, D-80788 Munich, Germany

[nieschulz | geiger | lang]@ei.tum.de, klaus.bengler@bmw.de

ABSTRACT

In this study we surveyed an approach to determine an user's need of assistance while operating an automotive human-machine interface by means of gestures. In an extensive usability study we investigated three characteristic features that seemed to be suitable for this approach. The first stage of our 2-stage methodology is a neural network based classification that determines if a user needs help while performing a certain task. In the second stage a heuristic postprocess based on statistics determines which help this user actually needs in the given context. This approach was then evaluated in an offline application.

1. INTRODUCTION

Gesture controlled operation of in car devices can provide a comfortable way of interaction. But one can assume, that not many people are familiar to use it intuitively. Therefore we present an automatic, adaptive help system, which supports the novice user to operate the human-machine interface (MMI). It seems to be perspicuous that the user's need for assistance or his uncertainty does have an effect on his cognitive performance, i.e. the reflection time, as well as on the execution of the gestures concerning duration and quality. Based on this consideration, our goal is to provide an automatic, but unobtrusive assistance to minimize the amount of help requests.

2. METHODOLOGY

2.1. Experimental Environment

The experiment was conducted in the institute's driving simulation laboratory (cf. fig. 2.2a). It consisted of two parts: 1) user interaction in a parked car and 2) user interaction while driving (simulation). This paper will present the results of part 1. The study was carried out applying the so called 'Wizard of Oz' methodology. This means that an experimental manager ('wizard') telecontrols occurring events and is able to influence the system's behavior, while the test person is told to interact with an already implemented und functioning system. The test subject was seated inside the car and confronted with an automotive MMI. The experimental manager resided in the control room. He had visual connection to the subject from two points of view (cf. fig. 2.2b). Subject and wizard communicated via audio intercom.

The GUI was optimized for operation using right hand gestures and simulates devices such as radio, cassette-player, CD-player, CD-changer, telephone and navigation system. It is the result of a study about controlling automotive devices by gestures. Subjects had to interact with the MMI exclusively by gestures to avoid interfering side effects. The experimental manager observed the subject's right hand movements via monitor and interpreted them on the basis of a given vocabulary. This means that the 'wizard' acted like a gesture recognition system. He only accepted valid gestures that were part of the vocabulary corpus. The implemented gesture vocabulary consists of a set of intuitive right handed movements which resulted from former studies of our institute [Zob01]. The test person had no instructions at all, except to use only gestures for interaction. If the subject had any problems concerning the manner of performing gestures or the appliance of the MMI, he/she could push a 'help request button' (cf. fig. 2.2c) to get context sensitive help. This assistance is provided in several different audio-visual 'help packages' (cf. fig 2.1).

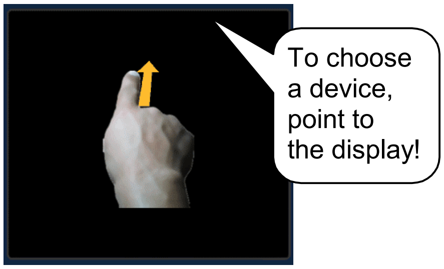


Figure 2.1: Example for a 'help package'

From: Proc. HCI 2001, Lawrence Erlbaum Ass.

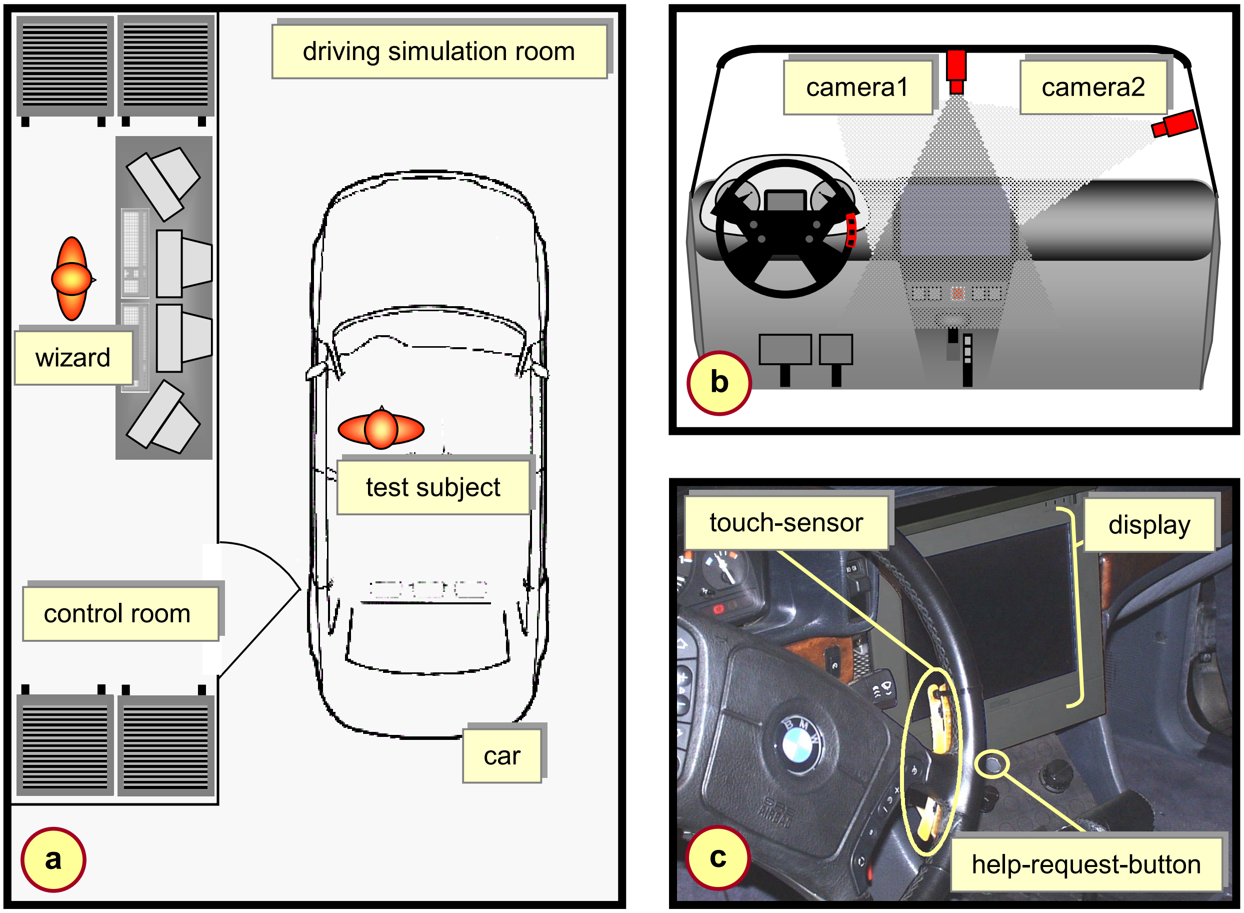


Figure 2.2: Schematic experimental setup: **a**) plan view of driving simulation laboratory; **b)** camera positions: camera 1 views the space where gestural input is performed, camera 2 shows a front view of the subject; **c)** layout of touch-sensor, help-request-button and display (also outlined in b)

During the experiment, the test subject was asked to perform a variety of typical tasks which were read out automatically by a virtual moderator. Such a task could be “Switch to CD number 4 and listen to track 5” or “Call Mr. Hunsinger”. The experiment was divided into several task blocks of increasing difficulty. After each block, the subject was interviewed. The person was asked how he/she gets along with the gestures and the operation of the MMI. Furthermore the experimental subject should explain whether he/she was satisfied with the given context sensitive assistance.

Throughout the whole test procedure all relevant data were automatically logged to hard disk with time stamp. These contain besides the status of the MMI (i.e. active device, current CD title, volume setting, current destination, etc.) every request for assistance and all moments the right hand of the experimental subject left the steering wheel and returned to it. This information is given by a touch sensor mounted to the steering wheel (cf. fig. 2.2c). The subject was instructed to keep this sensor always pressed unless gestural interaction had to be done.

2.2. Preprocessing (Feature Extraction)

In order to infer the user’s need of assistance, three features are determined as input for the help system. The first feature is the execution duration t_e of the gestures. This represents the time, while the right hand is off the steering wheel. The second feature maps the time the user has to think before executing a gesture. This feature is named ‘cognition time’ t_c. It is actually the time between the execution of separate gestures. Both were measured by the touch sensor located on the steering wheel (cf. fig. 2.2c). The third feature - execution quality of the gestures k - is estimated by the experimental manager and assigned to six categories: ‘unknown’, ‘very bad’, ‘bad’, ‘acceptable’, ‘good’ and ‘very good’. Thereby he has to consider two aspects: how well each gesture is executed itself and how well it can be distinguished from other gestures. The execution quality corresponds to a confidence measure of a real gesture recognition system. Later on the ‘Wizard of Oz’ will be replaced by our real-time gesture recognition system [Mor99]. All three features are then preprocessed to get a standardized feature space.

The first two features t_e and t_c are averaged by using former measurements of the actual user, as well as corresponding measurements of all test subjects. They are calculated as follows:

$$t_{e_{norm}}[n] = \frac{t_e[n]}{\left(1-\frac{1}{w_e}\right) t_{e_{norm}}[n-1] + \frac{1}{w_e} t_{e_0}} \qquad \text{and} \qquad t_{c_{norm}}[n] = \frac{t_c[n]}{\left(1-\frac{1}{w_c}\right) t_{c_{norm}}[n-1] + \frac{1}{w_c} t_{c_0}}$$

with			
	$t_e[n]$, $t_c[n]$	:	actual execution duration, resp. actual cognition time
	$t_{e_{norm}}[n]$, $t_{c_{norm}}[n]$	:	execution duration, resp. cognition time, normalized to previous averages
	t_{e_0}, t_{c_0}	:	each: average over all gestures and experimental subjects
	w_e, w_c	:	each: weight, to adjust the ratio between users and overall average

Then both features are non-linearly scaled to a data-range of 0 to 1 regarding standard deviation of all test data (cf. fig. 2.3). This leads to the features p_1 and p_2. Thereby the average durations $t_{e_{norm}}[n]=1$ and $t_{c_{norm}}[n]=1$ are mapped to $p_1[n]=0.5$ and $p_2[n]=0.5$. These operations provide adaptation to the users gestural behavior concerning execution duration and cognition time.

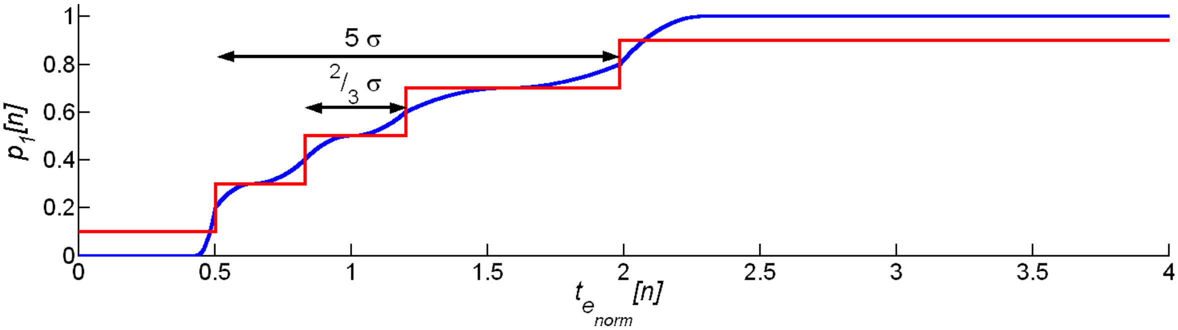


Figure 2.3: The blue line shows the non-linearly scaling function regarding standard deviation σ of all test data, the red line the five categories as described in sec. 2.3, both shown on $p_1[n]$ exemplary

The third feature - execution quality k - is averaged using a time interval, which is weighted with a memory function $w[n]$. The goal of $w[n]$ is to evaluate past gestures less than the current one. This is realized by a descending exponential characteristic: $w[n]=e^{-n}$. The feature is then scaled to a data-range of 0 to 1, too. This results in:

$$p_3[n] = \frac{1}{\sum\limits_{m=0}^{\infty} w[m]} \sum_{m=0}^{\infty} w[m]p[n-m] = \frac{1}{1+e^{-1}} \sum_{m=0}^{\infty} e^{-m} p[n-m]$$

Since $w[7]=e^{-7}<10^{-3}$ it is a good choice to only evaluate the past seven gestures, so that the time interval of $w[n]$ is seven gestures. To simplify matters p_3 can be determined recursively:

$$p_3[n] = (1-e^{-1})\, k[n-1] + e^{-1} p_3[n-1]$$

Feature p_3 can be called a current average of gesture quality. This is to avoid assistance for a user who performs a bad gesture only once. The bad quality of this gesture will be averaged in the context of the last gestures. That way this operation also provides an adaptation to the user’s gestural behavior in consideration of the last gestures.

Furthermore the statement, whether the subject actually needs assistance (‘yes’ or ‘no’), is determined as target feature $T[n]$.

2.3. Creating the underlying neural network

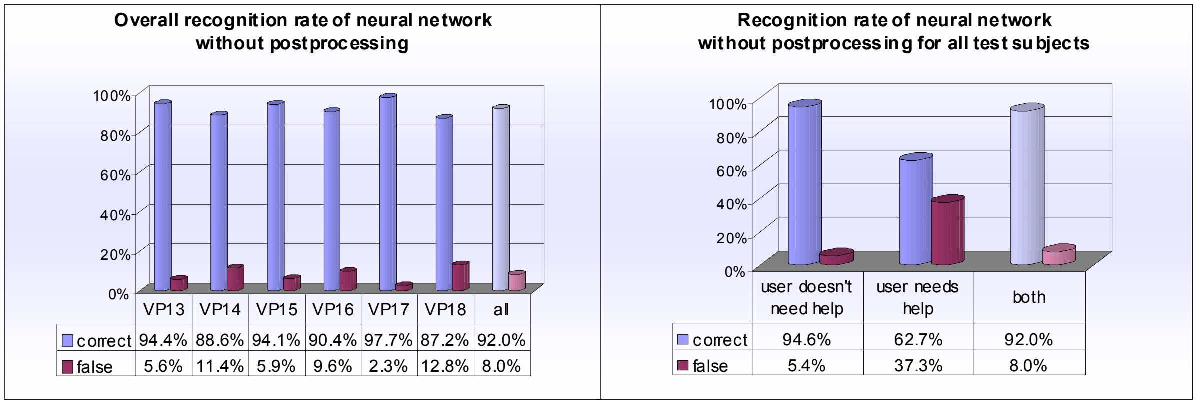
The feature vectors of the first two-thirds of the experimental subjects were then used to train a neural network, which supplies the statement, whether the subject needs help, or not. First the training data are categorized linearly: the first two features p_1 and p_2 in the five categories ‘very short’, ‘short’, ‘normal’, ‘fast’ and ‘very fast’ (cf. fig. 2.3) and the third feature p_3 into the same six categories as *k* above (cf. sec. 2.2). In consequence all feature vectors are mapped to a maximum of 150 training vectors. As the neural network is built as a probabilistic neural network (PNN) based on a radial basis network (RBN), the first layer contains a maximum of 150 neurons depending on the training material. As a positive side effect the memory usage of the neural network will be decreased as well as the system’s speed will be increased. As the training data do not cover the whole feature space there are areas which are represented only by very few neurons. In order to avoid an over-weighting, resulting from this, the individual neurons are weighted by their normalized average distance to all other neurons in the feature space. This is done because in training material of present quantity (the training material consists of about 2000 gestures) there are always outliers and because there is much more data for the statement ‘the user doesn’t need assistance’ than for the statement ‘the user needs assistance’. By these modifications it can be avoided that the network will almost always determine the class which represents the first statement. This means to the PNN that its recognition performance increases in stability.

2.4. Application and postprocessing

In the current status the help system is applied off-line. The logged data sets of the remaining one-third of the test subjects are used as test data. The individual features are preprocessed the same way as above, but not categorized. Then the input feature vector is fed into the neural network, which supplies the result, whether the user needs assistance, or not. As the network does not take into account the context of the MMI, it is necessary to find out the accurate assistance as a function of the context and the user’s operation history. Therefore the system searches the help database for the statistically most probable ‘help package’. Statistics of the current user (e.g. which gestures have been used recently, in which context, were they used in the correct manner, which assistance was already presented to the user, etc.?) as well as of all test subjects (e.g. with which gestures / with which functionalities of the MMI did the users get along best / worst, etc.?) are considered. If necessary the postprocessing algorithm sometimes suppresses assistance, even if the neural network determines that the user needs help. Such possible cases are checked heuristically and lead to an improvement of the overall recognition result. If, for example, the user was thinking a long time before the latest gesture and does a very slow and uncertain right hand move, yet, this caused what he wanted, e.g. to accept a telephone call for the first time, the MMI should not provide information of how to accept a telephone call.

3. RESULTS

In sum 2935 gestures of 18 experimental subjects were evaluated. In 304 cases subjects needed assistance. 2013 gestures were used as training data. The remaining 922 data sets were used for recognition. As can be seen in fig. 3.1 the overall recognition rate of the neural network is 92.0 % and the error rate is 8.0 %. The error rate can be reduced by relatively 24 % doing the postprocessing (cf. fig. 3.2). The recognition rate of the case, that the user doesn’t need



	VP13	VP14	VP15	VP16	VP17	VP18	all
■ correct	94.4%	88.6%	94.1%	90.4%	97.7%	87.2%	92.0%
■ false	5.6%	11.4%	5.9%	9.6%	2.3%	12.8%	8.0%

	user doesn't need help	user needs help	both
■ correct	94.6%	62.7%	92.0%
■ false	5.4%	37.3%	8.0%

Figure 3.1: Recognition rates of neural network without postprocessing

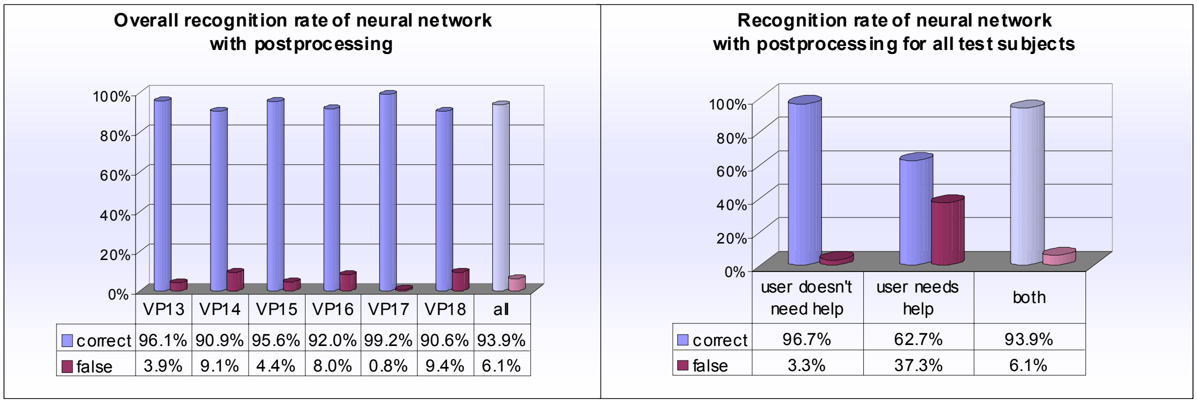


Figure 3.2: Recognition rates of neural network with postprocessing

help, is 94.6 %, while the rate of the other case, where the user needs assistance, is only 62.7 %. That is because of the fact that only about 10 % of the training material represents this target. The postprocessing has only effect in the first case and increases its recognition rate to 96.7 %.

Reclassification of the training material shows similar results (cf. fig. 3.3). But here the recognition rate for the target ‘user needs assistance’ is more than 10 % higher.

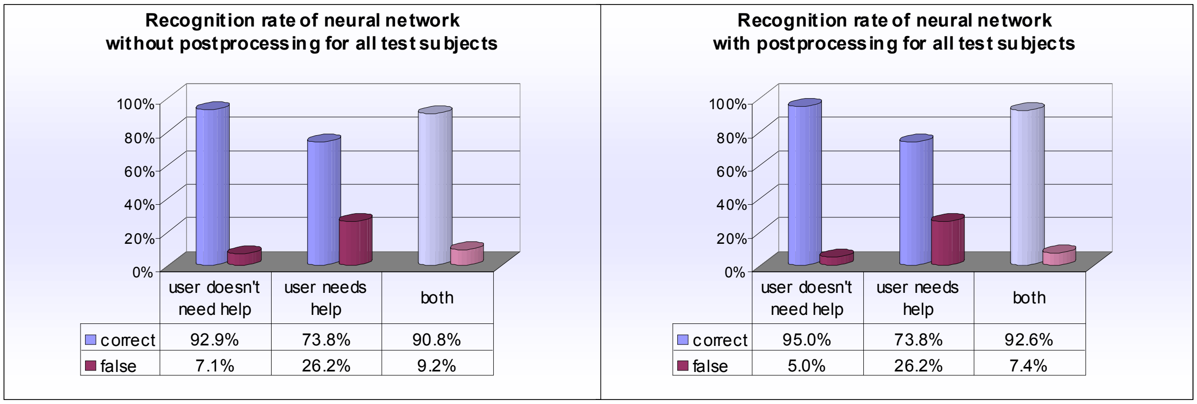


Figure 3.3: Reclassification with training data: Recognition rates of neural network for all experimental subjects

4. CONCLUSION

The results of our study show that the system is able to recognize reliably, whether the user is in need of assistance, or not. The statistical postprocessing ensures that most of the time the accurate assistance is presented to the user, so that the operation of the MMI is facilitated. We are quite confident that in the future the help system will also work very well with the input of our institute's real-time gesture recognition system [Mor99]. Follow-up studies will have to take into account that different gestures need different execution durations and that subjects need different cognition times in order to plan different gestures. In addition one has to consider that different gestures typically show different confidence measures.

REFERENCES

[Mor99] Morguet, P., Lang, M. (1999). Comparison of Approaches to Continuous Hand Gesture Recognition for a Visual Dialog System. *Proceedings ICASSP 99 (Phoenix, Arizona, USA), IEEE, Vol. 6*, 3549 (4 pp.).

[Zob01] Zobl, M., Geiger, M., Bengler, K., Lang, M. (2001). A Usability Study on Hand Gesture Controlled Operation of In-Car Devices. *Poster Proceedings HCII 2001 (New Orleans, Louisiana, USA), (this conference).*