Online Nod Detection in Human-Robot Interaction

Eduard Wall[1], Lars Schillingmann[1] and Franz Kummert[1]

***Abstract*— Nodding is an important factor in human communication, providing a physical cue for socially communicative acts such as turn taking, backchanneling, and confirmation. In this article, we describe a vision-based online head nodding detector that works with monocular camera images. Using SVM regression, our system estimates the head pose based on facial landmarks. Subsequence dynamic time-warping is then used to compare head pose features against nod templates. In contrast to many other previous implementations, our system was evaluated with study participants who were not instructed to reply by nodding, and shows good results while maintaining a low false positive rate.**

I. INTRODUCTION

Nodding is one important type of head movement in social interaction and communication, and the physical action transmits information on different functional levels [1]. On a semantic level, for instance, nodding indicates affirmation. In social interaction, nodding is used for backchanneling. In Japanese culture, nodding has also been observed to mark the end of one's turn speaking. Because nodding is so widespread in human communication and social interaction, robots and agents designed to interact with humans should be able to detect head nods. Such a cue can support various functions in robots, such as a dialogue system or attention monitoring, as for instance in engagement detection.

Various systems for nod detection have been proposed. Since nodding induces the relative movement of facial landmarks such as the eyes, their relative movements are typically used as features for nod detection. For example, in [2], pupil location is tracked using an infrared camera system. After this data is gathered, the location is smoothed and a symbolic direction of movement is calculated. A Hidden Markov Model (HMM) is used to detect nodding based on directional symbols. This model was trained using a corpus of 25 head nods from instructed users answering questions to an agent, and the system is capable of real-time processing. A similar approach is used in [3], but it differs from the previous approach in that the visual spectrum is used. In this system, the user's face and eyes are first detected. The relative changes in the eyes' positions on the face are used as a nod feature. The training data consists of 37 nod samples collected from instructed users. Instead of relying solely on the eyes as a landmark, optical flow can be used to detect facial motion [4]. The flow vector's angle is discretized into directional symbols and an HMM is used to classify nodding based on these directional symbols. This model is trained with 100 nod samples and depth sensing is also used for nod detection. In [5], the Kinect sensor is used to estimate head pose. A symbolic direction of movement is calculated based on pitch and yaw changes, and an HMM is used to classify nodding based on directional symbols. This model is trained with 150 samples of nodding. A more sophisticated approach is used in [6] than in the previously described models: a 3-D head tracker operating with the Kinect depth sensor is used to estimate the rotation matrix of the face and within a temporal window, frequency and axis features based on changes in the rotation matrix are calculated. A support vector machine is used to classify nodding. The model is trained on 543 nods taken from a corpus containing conversational interactions, such as job interviews and grant applications.

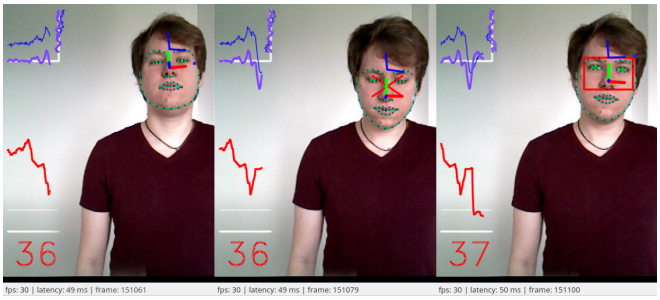

Fig. 1. Left to right: Three frames of a recorded head nod. Blue lines visualize the head pitch and yaw. Purple lines depict the derivatives and red lines show the actual distance to the nodding prototype. The current nod detection count is indicated by the red number.

Most of the existing work uses databases with instructed users. This might lead to more pronounced nodding patterns that are easy to detect. Furthermore, the corpora often contain a relatively high frequency of nods. Nodding behavior, however, tends to differ significantly among individuals: some people nod rarely, while others nod frequently [7]. Accordingly, a system with a low false positive rate is required to accommodate this. In addition, the amount of training data can be very low. We therefore propose a nod detection system that is based on subsequence dynamic time warping. In domains such as gesture recognition, dynamic time warping has been shown to outperform HMMs [8]. Our system is able to detect nods online, based on prototypical nods (see Fig. 1). We show that our nod detector performs well on human-agent conversational data, including cases with low nod frequencies. Furthermore, our approach relies on monocular VGA camera images and thus does not require specialized hardware such as depth sensors or high resolution cameras.

[1]Eduard Wall, Lars Schillingmann and Franz Kummert are with the Cluster of Excellence Cognitive Interaction Technology (CITEC), Bielefeld University, 33615, Bielefeld, Germany {lschilli, ewall, franz} at techfak.uni-bielefeld.de

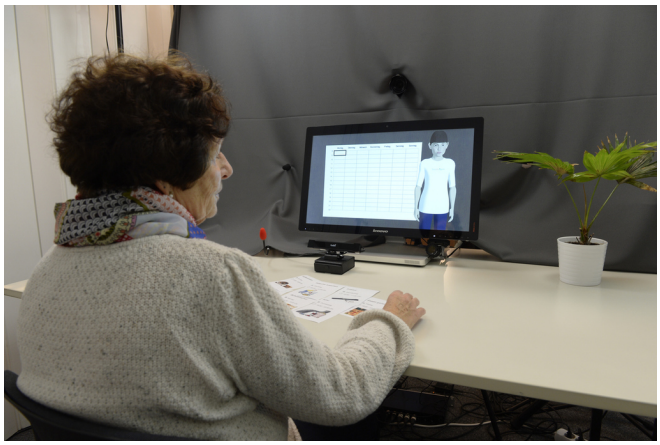

Fig. 2. Person interacting with the virtual agent BILLIE.

This paper is structured as follows: Section II introduces the scenario our nod detection approach is applied to. In Section III our methodology is explained step by step, including feature extraction, head pose estimation, and dynamic time warping. Finally, evaluation results based on existing datasets and a user study are presented in Section IV.

II. SCENARIO

Our work is part of the research project KOMPASS at Bielefeld University, where the virtual agent BILLIE is currently being developed to provide assistance to elderly or cognitively impaired people in planning their daily activities. BILLIE maintains a person's schedule by interacting with the user (see Fig. 2), for example by suggesting activities. In order to be able to successfully interact with people, BILLIE requires the capability to perceive cues from its conversation partner relevant to the interaction. One possibility is to rely on verbal communication for these cues. However, if the user's speech is not clear, visual cues such as nodding might help the system better understand the users' intentions or desires. Furthermore, such visual cues can also help the system to ascertain if the person is actually engaged in the interaction with the agent.

III. HEAD NOD DETECTION

The guiding idea behind our approach is to detect head nods based on changes in the head pose. Our system estimates the user's head pose based on facial landmark features and then uses dynamic time warping to compare changes in the actual head pose with prototypical changes that occur during nodding. Nodding is detected if the comparison results fall below a threshold that defines sufficient similarity. An overview of the approach is shown in Fig. 3. In the following sections, a detailed description of each step is provided.

A. Facial Feature Extraction

For face detection and facial landmark estimation, we rely on an implementation provided by dlib [9], [10]. The landmark estimation algorithm uses a cascade of regressors to estimate the face's landmark positions (see Fig. 4) directly

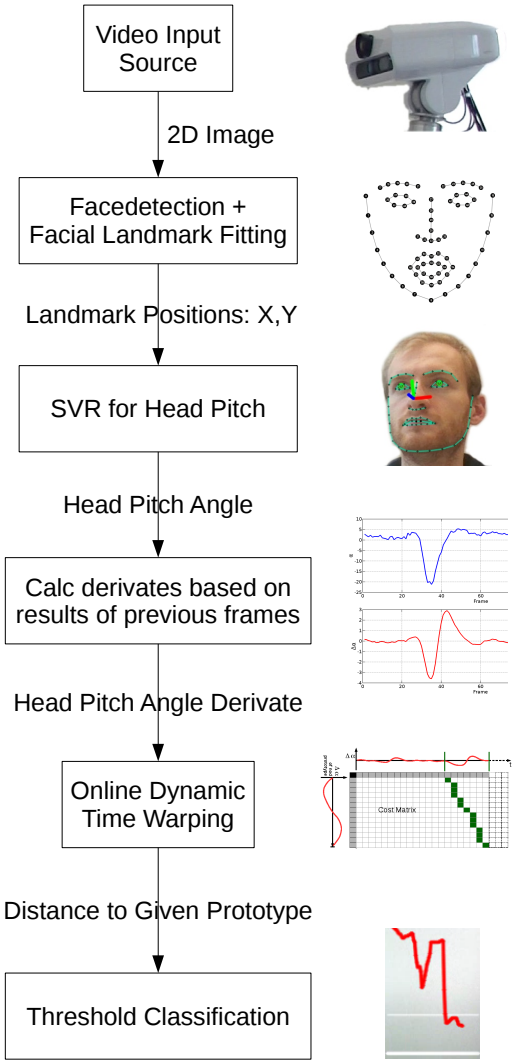

Fig. 3. Nod Detection System Overview

from a minor set of pixel intensities. Coordinates of the resulting landmarks are expressed relative to the nose root and are normalized for the rotation and size of the face. The result is a 136-dimensional vector consisting of the X and Y values of each normalized landmark position.

B. Head Pose Estimation

Various methods exist to estimate head pose, including geometric methods, tracking methods, and regression methods [11]. To robustly estimate the pitch and yaw angles of the head pose based on few assumptions, we decided to use support vector regression (SVR). The advantage of this approach is that it can be trained with head pose data from multiple users in varying light conditions. The resulting classifier does not need to be calibrated with an initial head pose or adapted to specific users. The Biwi Kinect Head Pose Database [12] was used for training the SVR models.

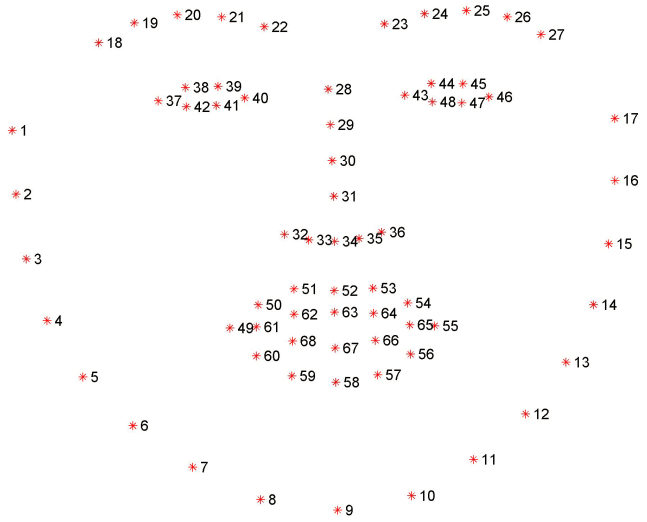

Fig. 4. Facial Landmark Features.

TABLE I

RESULTS OF HEAD POSE VALIDATION

	#people	#img	svm-c	svm-epsilon	svm-epsilon intensitivity	standard deviation
Yaw	24	5621	0.09	0.05	3.5	6.2°
Pitch	20	1617	0.09	0.1	3.0	6.3°

The database was recorded using the Kinect depth sensor (including 2-D RGB images) and consists of 24 people turning their heads in different directions, as well as reference head pose data. Images with faces not reliably detected by dlib's face detector were excluded. The data was further filtered to approximate a uniform distribution of pitch and yaw angles so as to optimize the training data for support vector regression. After filtering, 5,621 2-D images with annotated head orientations for yaw and 1,617 images for pitch were used as training data. Two independent SVR models for estimating pitch and yaw were trained based on the feature vector described in the previous section. Figure 5 depicts the head pose estimator visualizing detection results. Optimal model parameters were determined using a grid search.

For each parameter set, a three-fold cross validation was conducted to estimate the regression error. The best model resulted in a standard deviation of 6.3 degrees for pitch and 6.2 degrees for yaw. Results are shown in Table I. The head pose angles are used as features for our nod detection approach, and are described in the next section.

C. Dynamic Time Warping of Head Pose Changes

The idea guiding our approach to head nod detection is to compare the pitch angles estimated by the head pose SVR against reference nods. Humans exhibit head nods differing in terms of duration and strength. To measure the similarity between the estimated pitch and reference nods, dynamic time warping is used to compensate for length variations. Dynamic time warping is a widely used method

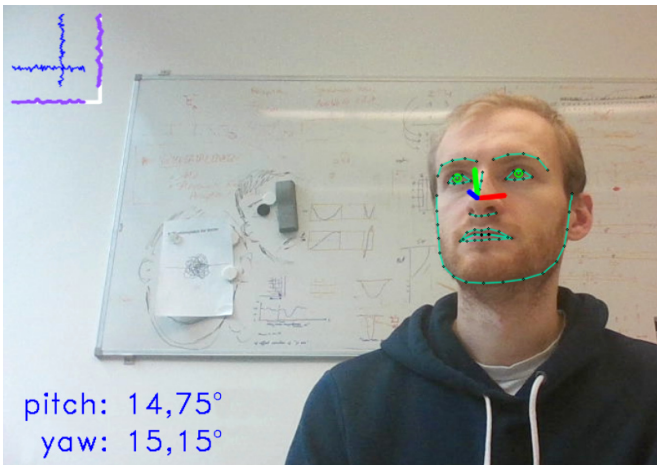

Fig. 5. Head Pose Estimation.

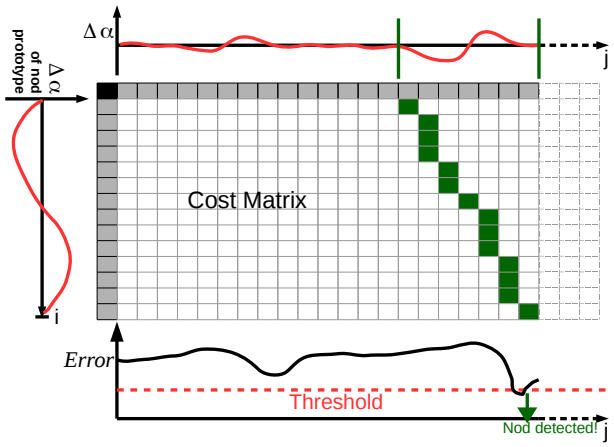

Fig. 6. Head nod detection with DTW.

for comparing time series. However, constant offsets and amplitude of the time series (for example, a different neutral position of the head and different nod strength) cannot be compensated for by dynamic time warping. These problems can be eliminated, though, by taking the derivative of the time series as an input for the DTW [13].

The idea of dynamic time warping is to warp two time series by pausing or continuing one of them such that the sum of distances between them becomes minimal. This requires a distance function. Given two discrete time series A and B of length M and N, a distance function $d(i,j)$ represents the distance between $A(i)$ and $B(j)$. In our implementation we chose Euclidean distance. To find the minimal distance the optimal warping path needs to be determined. First, all possible $d(i,j)$ are composed in a cost matrix C. The optimal alignment between A and B is determined by evaluating all paths from $C(0,0)$ to $C(M,N)$ and selecting the path where the sum of their elements is minimal. By using dynamic programming distance calculation and path evaluation can be handled in one step. This is accomplished by calculating an accumulated cost matrix C_A according to Equation 1.

Each distance calculation $C_A(i,j)$ also provides the cost of the minimal warping path to $C_A(0,0)$. Specifically, $C_A(M,N)$ holds the distance between the time series A and B.

$$C_A(i,j) = \begin{cases} d(0,0), & i,j = 0 \\ d(0,j) + C_L, & i = 0, j > 0 \\ d(i,0) + C_T, & i > 0, j = 0 \\ d(i,j) + min(C_L, C_{LT}, C_T), & i,j > 0 \end{cases} \quad (1)$$

where

$$C_L = C_A(i, j-1),$$
$$C_{LT} = C_A(i-1, j-1),$$
$$C_T = C_A(i-1, j).$$

D. Online Dynamic Time Warping

Processing data online means that our implementation has to be able to detect nods in a continuously expanding time series. The prototype time series is fixed in size. Thus, it is not sufficient to compare time series of fixed lengths using DTW. Multiple instances of a subsequence, i.e. the nod prototype, have to be detected in a continuous stream of head pitch values. This problem is solved by employing subsequence dynamic time warping which is a modification of the DTW algorithm to handle incremental updates. At each time step the accumulated cost matrix C_A is extended with a new column and filled with the accumulated costs for the new time point according to Equation 1. The bottom row C_A indicates the minimal distance $D_{M,j}$ for a subsequence ending at the current time step j. In the offline case the DTW warping path always ends on $C_A(0,0)$. However, a dynamic target point is required within the incremental series B. This is achieved by redefining the case $(i = 0, j > 0)$ in Equation 1 to $d(0,j)$ which leads to Equation 2.

$$C_A(i,j) = \begin{cases} d(0,j), & i = 0 \\ d(i,0) + C_T, & i > 0, j = 0 \\ d(i,j) + min(C_L, C_{LT}, C_T), & i \neq 0 \end{cases} \quad (2)$$

According to this method each element of the bottom row of C_A holds the minimal distance $D_{M,j}$ of each possible alignment of $A = [a_0, ..., a_M]$ and $B_k = [b_k, ..., b_N]$. In the event that $D_{M,j}$ falls below a given threshold, a nod is detected within the starting point k and end point N of the corresponding minimal path. To avoid detection overlaps, a cost of infinity is assigned to the column with the nod end point N. A schematic representation of online dynamic time warping for head nod detection is shown in Figure 6.

E. Slope Constraints in Online Dynamic Time Warping

In the previous section, we defined the start and ends point for the DTW warping path, but we did not impose any constraints on the route of the warping path. In some cases, the alignment of two series could pause one of the series while continuing on the other one for long time, thus matching very short to very long parts. This undesired behavior can be prevented by constraining the warping path's slope with a restriction on how many consecutive steps in the same horizontal or vertical direction are allowed. When the accumulated cost matrix is calculated, the slope constraint can be enforced by checking how many steps were taken in consecutive directions of the warping path. This prevents the warping path from deviating too far from the diagonal of the DTW distance matrix. Based on empirical testing our system allows a maximum of 2 steps in the same direction.

F. DTW Cost Normalization

As mentioned in Section III-C, we take the derivative of the time series to compensate for the constant offsets and amplitude differences of the pitch angle data. However, the amplitude differences of the derivative are not compensated for, although head nods may have been performed at different velocities. In general, it can be observed that two high amplitude time series with high similarity have a higher DTW distance than two head nods with comparable similarity but low amplitude. We therefore use the standard deviation of the subsequence in B with the length of A as a cost normalization factor. This heuristic is possible because the standard deviation correlates approximately linearly with the time series amplitude. The normalization is applied to the accumulated cost matrix by dividing the bottom row by the subsequence standard deviation.

G. Data Smoothing and Differentiation

For each new frame, a new head pose is estimated by the SVR model. The resulting time series tends to be noisy due to estimation errors. Furthermore, our approach relies on the derivative of the head pitch making it more sensitive to noise. We therefore use a polynomial filter introduced by Savitky and Golay [14]. The advantage of this filter over rolling mean or Gaussian smoothing is the better preservation of peaks including their characteristics. In addition, it combines differentiation and smoothing in one filter operation. The general Savitzky-Golay formula is shown in Equation 3, where n is the filter length.

$$y_t = \frac{1}{h} \sum_{i=-\frac{n-1}{2}}^{\frac{n-1}{2}} a_i x_{t+i} \quad (3)$$

For calculating the smoothed derivatives a_i should be set to i. Note that x_t refers to the raw signal data. The normalization factors h are listed in the following table:

filter length	h
5	10
7	28
9	60
11	110

Based on empirical testing we set the coefficients to $n = 9$ and $h = 60$.

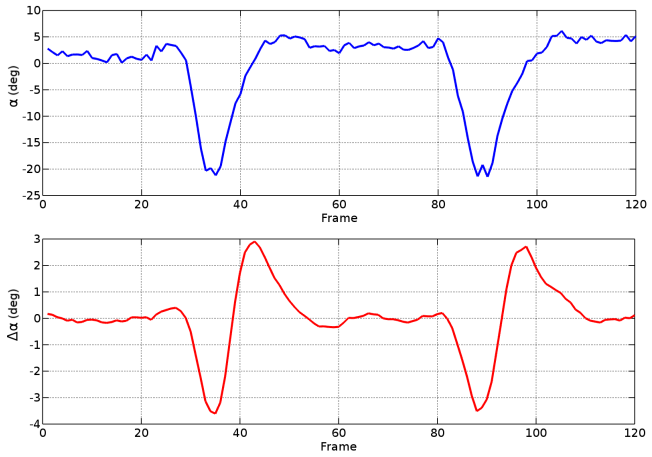

Fig. 7. Pitch angle (blue) and smoothed derivative (red) of two typical head nods.

TABLE II

THE PART OF THE WOZ1 DATASET USED FOR CROSS-EVALUATION.

dataset	#People	#Nods	Entire Video Length	Average Nod Duration
WOZ1	21	497	246 min	803 ms

IV. EVALUATION

A. Datasets

The KOMPASS WOZ1 dataset contains recordings of 51 participants, consisting of students, seniors and cognitively impaired individuals interacting with the virtual agent BILLIE. The participants were instructed to schedule their next week of activities with the help of BILLIE. Nodding observed in the participants' behavior can be assumed to occur naturally, since no instructions were given to participants with regard to nodding. For the purpose of this evaluation, a VGA camera recording was selected. The videos were captured at 30fps, for an entire video length of 629 minutes.

All head nods were annotated manually by two independent observers. Annotators were instructed to make their annotations according to the following definition: a head nod begins with the head starting to move down. Then it continues with a down-up movement and ends when the raising of the head has stopped. Figure 7 depicts the pitch data of two head nods in degrees over time, and the corresponding derivatives.

The length of all annotated nods combined accumulates to 12 minutes (1.9%) of video, with a total nod count of 690. This finding demonstrates that nodding and non-nodding classes are strongly imbalanced in natural interaction data. However, both annotators agreed on only 72% of nodding instances, indicating that some nodding behavior is difficult to judge. According to our observations, it is primarily subtle nods that contribute to the complexity of detecting nods. For further processing the annotations both annotators agreed with is chosen as ground truth.

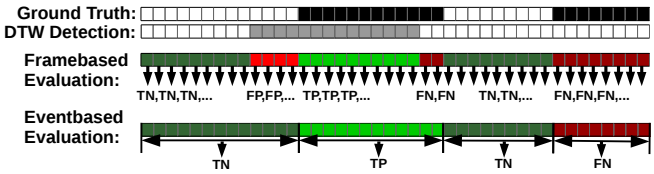

Fig. 8. Framewise and eventwise performance measure.

B. Frame- and Event-Based Evaluation Methodology

A single instance of nodding typically spans several frames. Thus, one way of evaluating the performance of the nod detector is to consider each frame by comparing it with the reference annotation. Each comparison can have four outcomes: True Negative, False Negative, True Positive, and False Positive (see Fig. 8). However, small temporal shifts between the ground truth and the nod detector are counted as errors even though the nod in general has been detected, the alignment is not fully identical. This motivates an event-based evaluation of the nod detector. Here, nodding is seen as a single event, which is detected correctly if the event falls within a certain temporal tolerance window of the reference annotation (see Fig. 8).

Frame-based evaluation is able to handle imbalanced datasets well. However, the false positive rate is higher than a human observer would agree with. Event-based evaluation, on the other hand, is not able to handle imbalanced datasets well. Nevertheless, the positive rate is closer to human judgment. Our datasets are imbalanced since they contain only few nodding frames compared to the frames for which no nodding was annotated. We therefore propose a combined evaluation strategy. Negatives (i.e. non-nodding frames) will be evaluated frame-wise, since they would otherwise form large blocks of non-nodding, which would count as only one event. Positives will be evaluated event-wise, since a small alignment error can be tolerated given the low number of nod events.

C. Nod Prototype and Threshold Selection for DTW

As described in Section III-C, a prototypical nodding time series that generalizes most nod types is required. This nod prototype will serve as the basis for calculating the DTW distance to arbitrary subsequences of live data. The prototype is selected from a set of annotated nod sequences from a training dataset. The sequence that performs best with respect to the training dataset is selected as the prototype. This requires a quality criterion to assess the classification performance.

First, the desired true positive rate (recall) is specified in advance. Subsequently, the DTW threshold is adjusted to match the desired recall. For each prototype candidate, we iterate over the DTW threshold and test the prototypes on the training dataset until the desired recall is reached. After all prototype candidates have been evaluated, the precision of each prototype for classifying the training data is used as a quality criterion. The sample with the highest precision is chosen as the nod prototype.

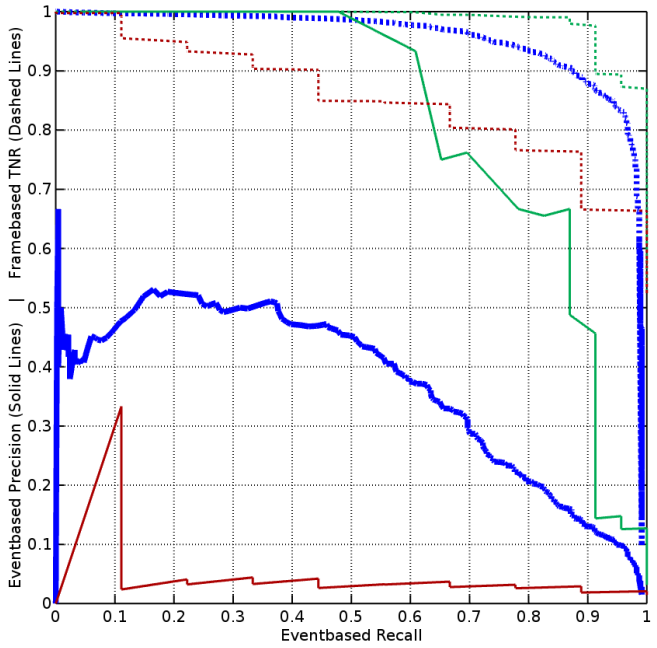

Fig. 9. Precision (y-axis) and recall (x-axis) of cross validation runs on the WOZ1 dataset: Solid lines indicate event based precision. True negative rate is plotted additionally on the y-axis as dashed lines. The solid blue line represents the averaged results over all participants of the test data. To visualize the large performance variation between participants' data, the green and the red line render a good and a bad per-participant result, respectively.

D. KOMPASS WOZ1 Results

To estimate the performance of the nod detector, a cross validation is performed on the Kompass WOZ1 dataset. For cross validation, we excluded videos with a face detection failure ratio of more than 90% as well as videos with less than four annotated nods. An overview of the remaining part of the WOZ1 dataset is shown in Table II. In each iteration, one person is left out and the nod prototype selection procedure described in the previous section is carried out on the remaining data. Each nod prototype is thus tested on the data from the excluded individual. The averaged results are presented in Figure 9. Relatively low results of event-wise precision (blue solid line) are misleading and are caused by the highly unbalanced dataset. Here, our combined frame and event-wise scheme (see IV-B) is used to handle the imbalanced training data, demonstrating high true negative (or low false positive) rates (blue dashed line).

For comparing our results with other methods, a comparable dataset is required. The nod detection approach introduced by Chen et. al., was evaluated on the KTH-Idiap dataset [6], where 4.5% of the frames contain nodding. To make our data comparable, a subset of the WOZ1 corpus (WOZ1-SUB) was composed. Random five minute long excerpts from the students and senior groups of the corpus were selected in a way that the same distribution of nodding and non-nodding frames was achieved. Our results on this subset and the results by Chen et. al., are compared in Table III.

TABLE III
COMPARISON OF OUR APPROACH WITH OTHER METHODS

dataset	Precision (event)	Recall (event)	F-Score (event)	True Negative Rate (frame)	#FP /min
WOZ1-SUB	0.43	0.69	0.53	0.97	3.1
Chen-KTH	0.60	0.79	0.68	0.98	1.6

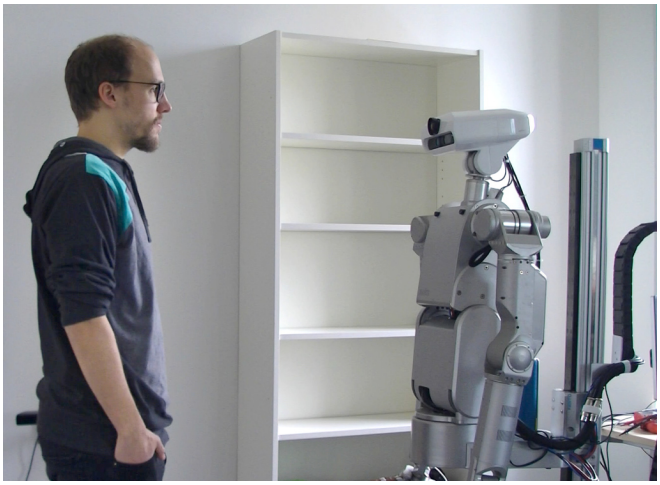

Fig. 10. Human-robot interaction scenario with the Meka robot.

E. Human-Robot Interaction User Study

In order to verify our approach in a human-robot interaction scenario and to evaluate the generalization capabilities of our model, we conducted a user study in which participants interacted with the humanoid robot Meka (see Fig. 10). An Asus xtion pro is integrated in the head of the Meka robot. However, in this evaluation only the RGB sensor is used, which captures the participants' heads for nod detection at 30fps in VGA resolution. Meka's voice was synthesized by using the text-to-speech system MaryTTS [15]. Before the interaction, participants were not instructed to use head gestures. The following dialogue was designed to induce natural nods: to engage participants in an interaction, Meka told the participants that he bought some food and needs help to make a fruit salad. After this, Meka went through a list of 15 food items and asked if each item can be an ingredient for fruit salad. The list contained 10 different fruits and five non-fruits. After giving the participant some time to respond, each question was followed by a short verbal confirmation. The steps in the dialogue were advanced by a Wizard-of-Oz to ensure correct timing between utterances.

Meka was programmed to return a head nod by performing a short down up movement of its head after detecting a nod from the human. The dialogue was not controlled by the nod detection module. A total of 10 participants (three female, seven male) took part in the study, four of whom wear eyeglasses.

The interaction and system activity was recorded. The total video length adds up to 22 minutes. For evaluation, all nodding was annotated event-wise to evaluate detection results on the event level. The results are shown in Table IV.

TABLE IV

NOD DETECTION RESULTS ON THE HUMAN-ROBOT INTERACTION DATA

	#Nods	Precision	Recall	F-Score	#FP/min
Total	62	0.56	0.61	0.58	1.5
Total [no glasses]	41	0.82	0.68	0.74	0.8

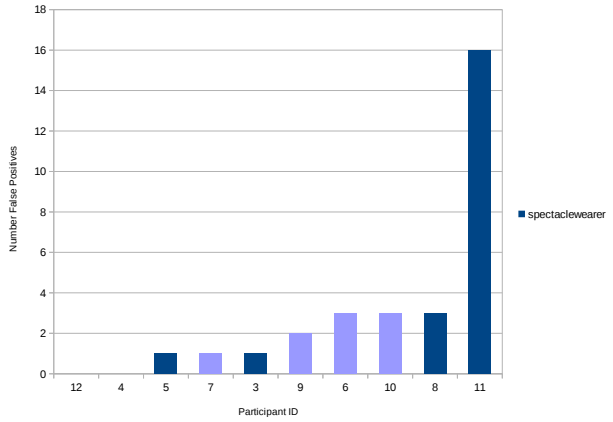

Fig. 11. Total number of false positives per participant. Note the strong outlier.

As shown in Figure 11, there was a strong outlier in our dataset with many false positives. The main cause of false positives was poor alignment of facial features, which frequently occurred with participants who wore glasses.

V. CONCLUSION

In this paper, we presented a nod detection system based on dynamic time warping that works on monocular VGA images. Support vector regression was used to estimate head pose angles based on facial landmark features. Noise reduction and differentiation was carried out using the Savitzky Golay filter. We extensively evaluated our approach on a corpus of human-agent interaction. Our approach already shows a relatively high true positive rate of 0.70, while maintaining high true negative rate of 0.97. This behavior is important because nodding is transient, and thus no nodding is observed most of the time in typical interactions. Furthermore, we conducted a user study to evaluate our approach using the Meka robot, which shows that our approach also works in human-robot interaction.

At the moment, our approach uses only one nod prototype. To achieve better coverage of the feature space of different nod types, we aim to extend our approach to multiple nod prototypes. Representative prototypes could be selected by a clustering algorithm. Furthermore, we plan to apply the same methodology to detect head shakes allowing the system to detect some forms of negation.

ACKNOWLEDGMENT

The authors gratefully acknowledge the German Federal Ministry of Education and Research (BMBF) for providing funding to Project KOMPASS, within the framework of which our research was able to take place. Furthermore, the authors would like to thank Sebastian Meyer zu Borgsen for his support with using the Meka robot and our student worker Kirsten Kästel for data annotation.

REFERENCES

[1] E. Z. McClave, "Linguistic functions of head movements in the context of speech," *Journal of Pragmatics*, vol. 32, no. 7, pp. 855–878, 2000. [Online]. Available: http://linkinghub.elsevier.com/retrieve/pii/S037821669900079X

[2] A. Kapoor and R. W. Picard, "A real-time head nod and shake detector," in *Proceedings of the 2001 workshop on Perceptive user interfaces - PUI '01*. New York, USA: ACM Press, nov 2001, p. 1. [Online]. Available: http://dl.acm.org/citation.cfm?id=971478.971509

[3] W. Tan, "A real-time head nod and shake detector using HMMs," *Expert Systems with Applications*, vol. 25, no. 3, pp. 461–466, oct 2003. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0957417403000885

[4] H. Gunes and M. Pantic, "Dimensional emotion prediction from spontaneous head gestures for interaction with sensitive artificial listeners," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 6356 LNAI, pp. 371–377, 2010.

[5] H. Wei, P. Scanlon, Y. LI, D. Monaghan, and N. E. O'Connor, "Real-time head nod and shake detection for continuous human affect recognition," in *Image Analysis for Multimedia Interactive Services (WIAMIS)*, jul 2013. [Online]. Available: http://doras.dcu.ie/19586/1/06616148.pdf

[6] Y. Chen, Y. Yu, and J.-M. Odobez, "Head Nod Detection from a Full 3D Model," in *2015 IEEE International Conference on Computer Vision Workshop (ICCVW)*. IEEE, 2015, pp. 528–536. [Online]. Available: http://ieeexplore.ieee.org/document/7406424/

[7] J. Lee and S. C. Marsella, "Predicting speaker head nods and the effects of affective information," *IEEE Transactions on Multimedia*, vol. 12, no. 6, pp. 552–562, 2010.

[8] J. M. Carmona and J. Climent, "A Performance Evaluation of HMM and DTW for Gesture Recognition," in *Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications*, ser. Lecture Notes in Computer Science, L. Alvarez, M. Mejail, L. Gomez, and J. Jacobo, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, vol. 7441, pp. 236–243. [Online]. Available: http://www.springerlink.com/index/10.1007/978-3-642-33275-3

[9] D. E. King, "Dlib-ml: A Machine Learning Toolkit," *The Journal of Machine Learning Research*, vol. 10, pp. 1755–1758, 2009. [Online]. Available: http://dl.acm.org/citation.cfm?id=1577069.1755843

[10] V. Kazemi and J. Sullivan, "One Millisecond Face Alignment with an Ensemble of Regression Trees," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2014)*, 2014.

[11] E. Murphy-Chutorian and M. M. Trivedi, "Head pose estimation in computer vision: a survey." *IEEE transactions on pattern analysis and machine intelligence*, vol. 31, no. 4, pp. 607–26, 2009. [Online]. Available: http://www.ncbi.nlm.nih.gov/pubmed/19229078

[12] G. Fanelli, T. Weise, J. Gall, and L. V. Gool, "Real time head pose estimation from consumer depth cameras," in *DAGM'11 Proceedings of the 33rd international conference on Pattern recognition*. Springer-Verlag, 2011, pp. 101–110. [Online]. Available: http://dl.acm.org/citation.cfm?id=2039976.2039988

[13] E. J. Keogh and M. J. Pazzani, "Derivative Dynamic Time Warping," in *First SIAM International Conference on Data Mining (SDM'2001)*, 2001. [Online]. Available: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.23.6686

[14] A. Savitzky and M. J. E. Golay, "Smoothing and Differentiation of Data by Simplified Least Squares Procedures." *Analytical Chemistry*, vol. 36, no. 8, pp. 1627–1639, 1964. [Online]. Available: http://dx.doi.org/10.1021/ac60214a047

[15] M. Schröder and J. Trouvain, "The German Text-to-Speech Synthesis System MARY: A Tool for Research, Development and Teaching," *International Journal of Speech Technology*, vol. 6, no. 4, pp. 365–377, 2003. [Online]. Available: http://dx.doi.org/10.1023/A:1025708916924