THIS PAPER APPEARED IN THE ICCV WORKSHOP, SANTIAGO, CHILE, 2015

Head Nod Detection from a Full 3D Model

Yiqiang Chen, Yu Yu and Jean-Marc Odobez
Idiap Research Institute and École polytechnique fédérale de Lausanne, Switzerland
yiqiang.chen@insa-lyon.fr, yyu@idiap.ch, jean-marc.odobez@idiap.ch

Abstract

As a non-verbal communication mean, head gestures play an important role in face-to-face conversation and recognizing them is therefore of high value for social behavior analysis or Human Robotic Interactions (HRI) modelling. Among the various gestures, head nod is the most common one and can convey agreement or emphasis. In this paper, we propose a novel nod detection approach based on a full 3D face centered rotation model. Compared to previous approaches, we make two contributions. Firstly, the head rotation dynamic is computed within the head coordinate instead of the camera coordinate, leading to pose invariant gesture dynamics. Secondly, besides the rotation parameters, a feature related to the head rotation axis is proposed so that nod-like false positives due to body movements could be eliminated. The experiments on two-party and four-party conversations demonstrate the validity of the approach.

1. Introduction

1.1. Head Nods

Head gestures have long been studied by psychologists. As the most common one, nod is often used in face-to-face conversation and has semantic functions. For listeners, they mainly nod to signal yes to a question, or show their interest, agreement and approval to the information they receive. Other functions may include enhancing communicative attention, or anticipating an attempt to capture the floor by occurring in synchrony with the others speech as conversational feedback [4, 5], along with other cues like gaze [2]. For speakers, they usually perform nods to emphasize their speech and in general convey the feeling of conviction or excitement. Therefore, the detection of head nods is a valuable module for social behaviors analysis and the study of social relations (e.g. [6]) and HRI design.

1.2. Related Work

A number of works on head gesture detection have been proposed. Head gestures are a series of head rotations performed around the neck. Among them, a head nod is the movement where the head is rotating up and down along the sagittal plane one or several times.

Kapoor et al. [1] present a technique to recognize head nods and shakes based on two Hidden Markov Models (HMMs) using 2D coordinate results from an eye tracker. In [12], the AdaBoost algorithm and anthropomorphic measures are applied to detect user's face and locate eye zone, respectively. Head movements are then derived from the eye location, and are then used within a discrete HMM to detect head nods and shakes.

Some works have been developed based on 3D head trackers. The approach in [8] models a nod as a velocity pattern of the pitch angle. The pattern is extracted when the 3D head tracker changes from a negative threshold to a maximum positive threshold within a certain time interval. The authors in [13] describes another method in which 5 head states (up, down, left, right and still) are distinguished. Then the head nod and shake are further recognized with two HMMs.

These approaches show good performance in simple scenarios where listeners use exaggerated head gesture to answer yes or no. But their performance drops significantly in detecting nods in natural face-to-face conversations where nods are more subtle and less explicit, because these methods tend to define nods as a sequence of head positions, which is a noisy feature to extract.

To better characterize and exploit the nod oscillating nature, other approaches use frequency features from the Fourier transform applied to head velocities. For instance, Morency et al. [7] use them as well as contextual features like lexical information or prosodic cues from an embodied conversational agent (ECA) to predict head nods of humans, in a scenario involving a human interacting with an ECA. Nguyen et al. [9] develop a multimodal method using frequency feature and audio based self-context by taking into account the influence of the speaking status of people on the dynamics of the head gestures. In this approach, the head velocities are computed at three arbitrarily defined points in a bounding box of a face tracker, using a robust and multiresolution optical flow computation method. [10].

The authors apply a Fourier transform with Gaussian temporal window to these velocities. Fourier features are then used to train two separate classifiers, one for speakers and the other one for listeners. Compared to [8] [13] [1] [12], frequency features result in a better description of fine head movements. These two approaches have shown good performance in the context of human-computer interaction (HCI) [7] and natural conversation [9]. The features characterizing the context, in which nods occur, have also been proved useful for improving the detection.

A main limitation of all these methods is the constraint linked to the position of the camera. They assume that in the training and test video, interlocutors have a similar head pose, and very often a frontal one. Therefore, these approaches cannot achieve pose invariance when camera position changes, or when people faces are oriented in more variable direction, e.g. when observing people in multiparty situations.

1.3. Main Contribution

In this paper, unlike previous approaches with 3D head tracker that extracts angular velocities directly by differentiating the Euler angles obtained from the pose expressed in the camera coordinate system, we propose to calculate the relative rotation at each instant with respect to the head pose at some instance before. The Euler angles from this relative rotation matrix are extracted. As they represent angular changes between two frames, they can be considered as a representation of angular velocities when using small time intervals. The advantage of this approach is that the measures are independent of the pose of the person with respect to the camera. This avoids some possible observation mismatches between training and testing due to the person being seen in a different pose with respect to the camera.

Furthermore, to fully characterize a rotation, only using the Euler angles (or visual velocities) is not sufficient. People may move their upper body back and forth, generating in this way oscillatory pose angles. The main difference is that here the rotation axis might be located around the pelvis rather than around the neck. Thus, our system also propose a feature related to rotation axis for classification. This feature could help distinguishing from which part of the body the rotation comes from, so that rotation movements not originating from the neck can be excluded.

The rest of the paper is organized as follows. In section 2 we describe our head nod detection method with 3D rotation model. In section 3, the experimental process is described. The results are discussed in section 4 and the conclusion is given in section 5.

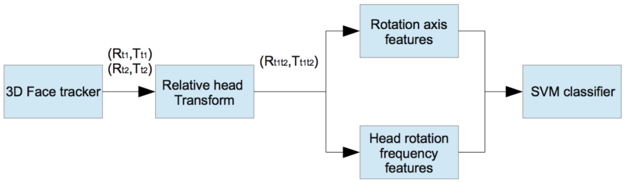

Figure 1. Overview of nod detection system.

2. Head Nod Detection with Relative Rotation and Distance to Rotation Axis

The overall procedure of the approach is shown in Fig.1. The head pose represented by a rotation and a translation of the face with respect to the camera coordinate system is first obtained from a 3D head tracker at each frame. Then the head rotation dynamic characterized by the head rotation $R_{t_1 t_2}$ and translation $T_{t_1 t_2}$ is computed within the head coordinate frame. In the next phase, two sets of features are extracted. First, similar to the work of Nguyen et al. [9], our system applies a Fourier transform with Gaussian window to the rotation angles derived from $R_{t_1 t_2}$. Second, rotation axis features are also extracted from the relative translation and rotation. Finally, a SVM classifier is applied to all features. A more detailed description of each step is given below.

2.1. Head Tracker

In order to estimate the head pose, the method in [3] is used. The method relies on a 3D Morphable Model (3DMM) to generate person specific 3D face templates. It is realized by fitting the 3DMM to a set of instances of the target to reduce the influence of the noise. The face tracker itself is based on the Iterative Closest Points (ICP) algorithm using point-to-plane constraints and the personalized template.

2.2. Relative Head Transform

Given a point P in the 3D space, we denote as $X^{cs}(P)$, the coordinates of this point in the coordinate system CS. In the face tracking system, there are two coordinate systems: the world coordinate system X^w which is fixed and located 1 meter away from the camera, and the face coordinate system $X_t^f(P)$, where t is the time since the face coordinate varies with time t. In the face coordinate system, the z axis is defined as the front direction of a person, whereas the x and y axis are defined as the side direction and the vertical direction respectively (see Fig.2).

For a point P_t, the outputs of the tracker relate the face coordinate and camera coordinate. The corresponding transformation for every frame is represented by a 3×3 rotation matrix R_t and a translation vector T_t, defining:

$$X^w(P_t) = R_t X_t^f(P_t) + T_t \, . \quad (1)$$

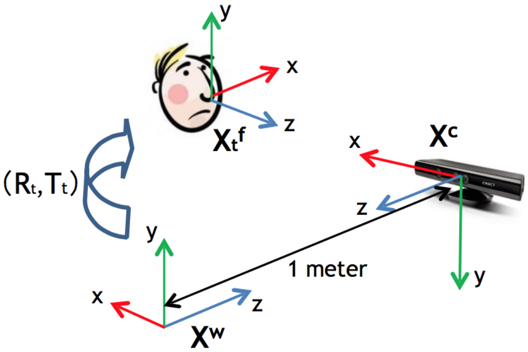

Figure 2. World and face coordinate system used by the head tracker.

We are interested in defining the transformation between the face coordinate systems at time $t_1 = t - m$ and time $t_2 = t$. Let us consider a face point P and let us denote by P_{t_1} and P_{t_2} its position in the 3D space at time t_1 and t_2. As the point is rigidly attached to the face, we have:

$$X^f_{t_1}(P_{t_1}) = X^f_{t_2}(P_{t_2})\ . \tag{2}$$

The point P_{t_2} at t_2 can be expressed in the face coordinate system at t_1 according to the transformation in Eq.3:

$$\begin{aligned} X^f_{t_1}(P_{t_2}) &= R^{-1}_{t_1}(X^w(P_{t_2}) - T_{t_1}) \\ &= R^{-1}_{t_1}(R_{t_2}X^f_{t_2}(P_{t_2}) + T_{t_2} - T_{t_1}) \\ &= R^{-1}_{t_1}R_{t_2}X^f_{t_2}(P_{t_2}) + R^{-1}_{t_1}(T_{t_2} - T_{t_1}) \\ &= R_{t_1t_2}X^f_{t_1}(P_{t_1}) + T_{t_1t_2} \end{aligned} \tag{3}$$

This equation represents the rigid rotation of a face point expressed in the coordinate system of the face at t_1. Thus the relative transformation between t_1 and t_2 which is represented by the relative rotation matrix $R_{t_1t_2}$ and relative translation $T_{t_1t_2}$ is given by:

$$\begin{aligned} R_{t_1t_2} &= R^{-1}_{t_1}R_{t_2}, \\ T_{t_1t_2} &= R^{-1}_{t_1}(T_{t_2} - T_{t_1}) \end{aligned} \tag{4}$$

2.3. Feature vector extraction

In this part, we present our encoding into features the transformation matrices $R_{t'-m,t'}$ defined for each time step t' of a short time window $[t-\Delta, t+\Delta]$ into features.

2.3.1 Rotation Frequency Features

At each time step t', we can extract from $R_{t'-m,t'}$ the three euler angles: roll, pitch, and yaw denoted by $(\alpha_{t'}, \beta_{t'}, \gamma_{t'})$ which are defined as the rotations around z-, x- and y-axis (Fig.3). We define $\alpha_{t-\Delta T:t+\Delta T}$ as the sequence of α observations within the temporal window $[t-\Delta T, t+\Delta T]$ (and similarly for β, γ), that is:

$$\alpha_{t-\Delta T:t+\Delta T} = [\alpha_{t-\Delta T}, \dots, \alpha_{t+\Delta T}]. \tag{5}$$

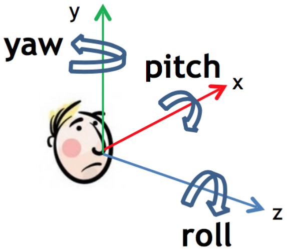

Figure 3. Euler angles defined in the head coordinate system.

In addition, we define a Gaussian window as:

$$\begin{aligned} &W_{2\Delta T+1} = [G(-\Delta T), \dots, G(\Delta T)] \\ &with\ G(n) = e^{-\frac{1}{2}(\frac{n}{\sigma})^2}. \end{aligned} \tag{6}$$

In order to characterize the oscillatory nature of head nods around time t, we apply a Fourier transform along with a Gaussian window to these three angle series, leading to:

$$\begin{aligned} A_{-\Delta f:\Delta f} &= DFT(\alpha_{t-\Delta T:t+\Delta T} \cdot W_{2\Delta T+1}), \\ B_{-\Delta f:\Delta f} &= DFT(\beta_{t-\Delta T:t+\Delta T} \cdot W_{2\Delta T+1}), \\ \Gamma_{-\Delta f:\Delta f} &= DFT(\gamma_{t-\Delta T:t+\Delta T} \cdot W_{2\Delta T+1}) \end{aligned} \tag{7}$$

We then compute the norm of the output of the Fourier transform, which is defined as follows for A:

$$|A_k| = \sqrt{Re(A_k)^2 + Im(A_k)^2}. \tag{8}$$

Finally, we take the positive part of the normalized frequency spectrum to avoid redundancy from the three angle series and concatenate them to obtain the vector used as frequency features for the frame located at the center of the window:

$$\begin{aligned} f^{rot}_m(t) = [&|A_0|, \cdots, |A_{\Delta f}|, \\ &|B_0|, \cdots, |B_{\Delta f}|, \\ &|\Gamma_0|, \cdots, |\Gamma_{\Delta f}|] \end{aligned} \tag{9}$$

Remember that the m in f^{rot}_m refers to the frame gap between the two frames needed to compute the relative rotation $R_{t'-m,t'}$.

2.3.2 Rotation Axis Features

Only looking at the angles is not enough to describe a head movement. Indeed there can be some motions leading to similar angle changes like back and forth motion with the upper body, but which are not head nods. In this paper, our goal is to capture that head rotations are done around an axis located near the neck. To do so, we compute the distance between the face and the rotation axis as additional but important feature characterizing the relative rotation.

Finding the distance d to the rotation axis. Let us consider a rigid transformation defined by the rotation R and

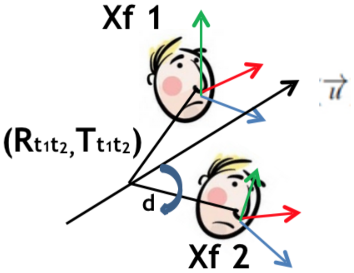

Figure 4. Head rotation around a fixed axis.

translation T. By definition, the rotation axis is the set of points invariant to the rigid transformation, which can therefore be obtained by solving the following equations:

$$\begin{aligned} X &= RX + T, \\ (I - R)X &= T \end{aligned} \tag{10}$$

There are three cases when solving for the above equation:

1. $R = I$ and $T \neq 0$. In this case, the set of points is empty. In practice, we will consider the distance to be at infinity and set the axis distance d to a large value.
2. $R = I$ and $T = 0$. In this case, the set of points is the 3D space. We consider that the distance is 0, and set $d = 0$.
3. $R \neq I$ and $T \neq 0$. In this case, the equation provides the rotation axis we are looking for.

Since we have a rotation around a fixed axis, the solution of Eq.10 is a line in the 3D space which means that $I - R$ is a singular matrix. To identify this axis, we can extract the direction of this line, and one point P^* from this line. For the latter one, we can use the least square solution as a particular solution of the equation. In our resolution, we use $(I - R) + \epsilon I$ instead of $I - R$ to stabilize the computation and avoid spurious values caused by noise (where ϵ is very small, we took 0.0001 in our calculation).

The null space of $I - R$ indicates the direction of the axis. In other words, to identify the axis direction, we can search for the unitary eigenvector $\overrightarrow{u}$ correspongding to the eigenvalue 1 of R, as every rotation matrix must have this eigenvalue (the other two being complex conjugates of each other), which can be found by solving:

$$R\overrightarrow{u} = \overrightarrow{u} \tag{11}$$

Then the distance between the origin O of the face coordinate system and the axis can be calculated as:

$$d = \|\overrightarrow{OP^*}\| \sin(\delta) \text{ with } \delta = \arccos\left(\frac{\overrightarrow{OP^*} \cdot \overrightarrow{u}}{\|\overrightarrow{OP^*}\| \|\overrightarrow{u}\|}\right).$$

Axis features. We can apply the above to the relative transformation defined by $R_{t'-m,t'}$ and $T_{t'-m,t'}$ and obtain the distance $d_{t'}$. Then, to summarize the axis information within the interval $[t - \Delta T, t + \Delta T]$, we take the maximum and average of the distance in the temporal window and define the axis features as:

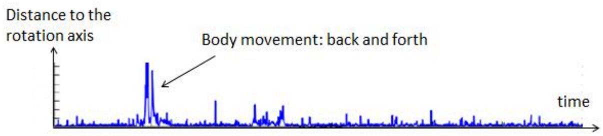

Figure 5. Distance of the relative axis of the relative rotation to the head frame origin.

$$f_m^{axis}(f) = [max(d_{t-\Delta T,t+\Delta T}), mean(d_{t-\Delta T,t+\Delta T})]$$

This feature can be used to eliminate false positives caused by body motion. Indeed, such motions like leaning forward and back, adjusting the sitting position, standing up and sitting down, may exhibit angular changes similar to nods. Our expectation is that the distance to the axis will be able to distinguish them from nods since nods are rotation around the neck, and these body motions usually have their axis distance much farther, as illustrated in Fig.5.

2.4. Classification

For the window $[t-\Delta, t+\Delta]$, we concatenate the Fourier features introduced in Section 2.2 as well as the maximum and average of the distance obtained in Section 2.3 into a single feature vector. Then, the system performs the classification of nods with this vector using a support vector machine (SVM). A support vector machine constructs a hyperplane which maximize its distance to the nearest training-data point of any class, since, in general, the larger the margin the lower the generalization error of the classifier. Some kernel functions can be used to implicitly project the data in a higher dimensional space where the date becomes more separable. The SVM classifier is applied at every frame. To filter out spurious detection, we applied a smoothing filter which eliminates detection events of very short duration (less than 7 frames).

3. Experimental Setting

In this section, we will present the design of our experiments, including the data we used, the parameter setting, training and evaluation method.

3.1. Dataset

In our experiments, two datasets are used.

Ubimpressed dataset. Acquired with a Kinect 2 sensor at 30fps, it consists of videos of job interviews. The camera is set about one meter away from the interlocutor and makes a little angle with the front direction (see Fig.6, left, people

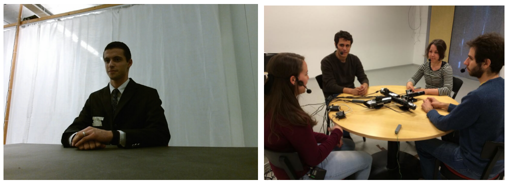

Figure 6. Ubimpressed sample (left) and view of the KTH-Idiap corpus setting (right).

Figure 7. Sample images of the KTH-Idiap corpus.

are seen from below and the side). The dataset comprises 12 videos, each containing different people, for a total time duration of 60 minutes.

KTH-Idiap corpus [11]. It features four people: one interviewer, and 3 interviewees who are applying for a funding grant. People are seated around a round table and each person was filmed by a Kinect 1 camera (See Fig.6, right). The video frame rate is also 30 fps. Since the conversation happend around a round table, the participants tend to look at each other and turn their sides to the camera in the videos (See examples of people in Fig.7). While full videos last around one hour, for the experiment we selected 5 minute excerpts from the videos of 9 different people.

3.2. Annotation

All the head nods were annotated manually. We annotated 13874 frames, for a total of 543 head nods in the two datasets (see Tab 1). The average duration of a nod is 25.5 frames ($\approx$ 0.85s). Nods in KTh-Idiap are longer on average because there are more continuous multi-nods.

Since head nods might be difficult to define and different people hold different opinions towards ambiguous ones, we annotated two classes of nods: obvious and subtle, according to the amplitude and duration of the rotation movement. Around 50% of the nods were considered as obvious.

Table 1. Nod statistics for Ubimpressed and KTH-Idiap dataset.

	#Nods	#Nod Frames	#Obvious Nods	Average Duration
Ubimpressed	407	10252	201	25.2
KTH-Idiap	136	3622	83	26.6
Total	543	13874	284	25.5

3.3. Parameter Setting

The algorithm comprises several parameters. First, we looked at the parameter m, the time interval (measured as the number of the spacing frames) used to compute the relative rotation mentioned in section 2.2. In general, larger m might be more robust against the noise of the head tracker but may lead to detectors which are less sensitive to movement details. In our experiment we tried with $m = 1, 3, 5, 7$.

Another parameter is the size of the Gaussian window $2\Delta T + 1$. In our experiments, we chose 31 frames (about 1 second) so $\Delta T = 15$. Note that the resulting window duration is larger than the duration of 90% of the annotated nods.

Apart from that, we chose the LIBSVM package as the SVM library tool. SVM parameters were chosen via 5-fold cross validation within the training set with a grid search. Note that in all cases, the feature vectors were z-normalized (i.e. the mean was subtracted and the result was divided by the standard deviation).

3.4. Performance Measure

The performance of the head nod detector is measured at the frame and event levels. At the frame level, we used the standard precision, recall and F-score measures. At the event level, we first need to match recognized events with the ground truth. To do so, suppose that there is a nod event e_i^{gt} (ground truth) happening in the time interval $I_i^{gt} = [t_{i,s}^{gt}, t_{i,t}^{gt}]$ and a detected nod event e_j^d in $I_j^d = [t_{j,s}^d, t_{j,t}^d]$ (see Fig.8). Then the event matching precision, recall, and F-score between e_j^{gt} and e_i^d are defined as:

$$P_{i,j}^{ov} = \frac{|I_i^{gt} \cap I_i^d|}{|I_i^{gt}|}, R_{i,j}^{ov} = \frac{|I_i^{gt} \cap I_i^d|}{|I_i^d|}, F_{i,j}^{ov} = \frac{2P_{i,j}^{ov} \cdot R_{i,j}^{ov}}{P_{i,j}^{ov} + R_{i,j}^{ov}}.$$

The events are said to match if their F-score is above a threshold (in that case e_i^{gt} is considered detected and e_i^d is considered correct). One difficulty arises in the case of long lasting multiple nods, which are difficult to annotate (as a single long nod, as separate short ones). In order to account for this situation, we set the threshold as 0.1

Then, given the matched events, we can compute the

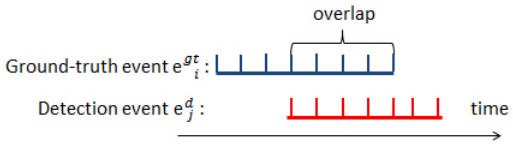

Figure 8. Nod matching.

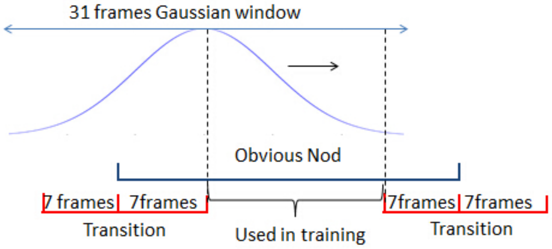

Figure 9. Transition frames around nod start and ends were not used for training

event-level precision, recall and F-score as follow:

$$P_{event} = \frac{\#\left\{e_j^d | \exists i, F_{i,j}^{ov} > threshold\right\}}{\# e^d},$$
$$R_{event} = \frac{\#\left\{e_i^{gt} | \exists j, F_{i,j}^{ov} > threshold\right\}}{\# e^{gt}}, \tag{12}$$
$$F_{event} = \frac{2 P_{event} \cdot R_{event}}{P_{event} + R_{event}}.$$

3.5. Test Classifier

Training classifiers. As very subtle head nods can be similar to non-nod movements during speaking, only obvious nods were used to collect the training samples[1]. Furthermore, to avoid introducing noise in the learning stage, we defined as transition frames 7 frames before and after the onset and offset frames of a nod, and did not use them as training samples (either as negative or positive samples). Note that by using only the central part of the nods as training data, we can guarantee that in general the most part (at least three quarters) of the Gaussian window used to compute the frequency overlaps the nods (See Fig.9).

Negative samples were chosen randomly, and more negative samples were chosen than positive ones since the space of negative gestures is larger than the space of positive ones. At the end we had 3100 positive samples and 10000 negative samples in the Ubimpressed data.

Tested classifiers. We trained 3 different support vector machines, with 3 different feature sets:

1. **Baseline**: this corresponds to the work in [8, 13, 7], where the Fourier transform outputs of sequences of Euler angle differences computed using the pose matrix defined with respect to the camera frame are used as feature.
2. **Relative rotation (RelRot)**: In this case, the Fourier features $f_m(t) = f_m^{rot}(t)$ of the angle extracted from the relative rotation matrix are used, as shown in section 2.2.
3. **Relative rotation + axis distance (RelRot-AxisDist)**: in addition to the rotation features, the average and maximum of the distance to the rotation axis is used. That is, $f_m(t) = [f_m^{rot}(t), f_m^{axis}(t)]$.

Table 2. Results of Ubimpressed data.

	EventLevel			*FrameLevel*		
	Precision	Recall	F-score	Precision	Recall	F-score
$linear\ SVM, m = 3$						
Baseline	0.8	0.83	0.81	0.8	0.73	0.76
RelRot	0.81	0.83	0.82	0.8	0.72	0.76
RelRot-AxisDist	0.81	0.83	0.82	0.78	0.75	0.76
$RBF\ SVM, m = 3$						
Baseline	0.84	0.8	0.82	0.84	0.68	0.75
RelRot	0.85	0.81	0.83	0.84	0.68	0.75
RelRot-AxisDist	0.87	0.8	0.84	0.83	0.7	0.76
$linear\ SVM, m = 5$						
Baseline	0.81	0.83	0.82	0.82	0.74	0.78
RelRot	0.84	0.83	0.84	0.82	0.75	0.78
RelRot-AxisDist	0.82	0.85	0.83	0.8	0.77	0.78
$RBF\ SVM, m = 5$						
Baseline	0.86	0.79	0.82	0.86	0.7	0.77
RelRot	0.86	0.8	0.83	0.87	0.71	0.78
RelRot-AxisDist	0.87	0.82	0.84	0.86	0.73	0.79

4. Experimental Results

Experiments are conducted on the two datasets separately. In section 4.1 we present the results obtained on the Ubimpressed dataset. Then in section 4.2, we apply the trained classifier on KTH-Idiap dataset to test the generalization performance.

4.1. Ubimpressed Data

A leave-one-person cross validation experiment was performed among the Ubimpressed dataset. That is, the SVM classifier was trained with samples from 11 videos and tested on the last one by applying nod detector to the entire video. All the videos make turns to be the test sample. We used both radial basis function kernel and linear kernel for $m = 1, 3, 5, 7$.

Overall results. Table.2 reports the results. In general, we obtain a F-score of 0.83 at the event level, and of 0.76 at the frame level. In this latter case, we can notice the higher precision and lower recall, which might be due to the use of only obvious nods during learning, shown in section 3.5. Overall, our results are quite high, when considering the subtleness of most of the examples.

Influence of time interval m. Table.2 and Fig.10 report

[1] Note that this only concerns training. Subtle nods were kept in the test set for evaluation.

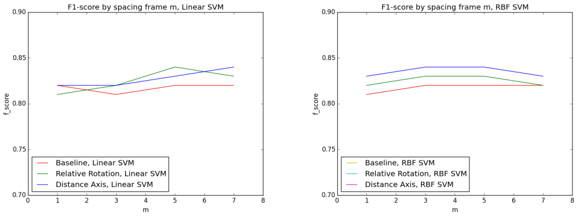

Figure 10. Event-level results in function of the time interval m. Left: linear SVM results; Right: RBF SVM results.

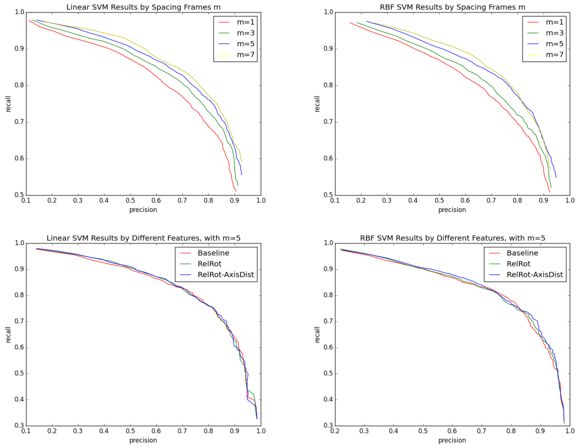

Figure 11. Frame-level results in function of the time interval and features. Top Left: linear SVM results; Top Right: RBF SVM results; Bottom Left: linear SVM results ($m = 5$); Bottom Right: RBF SVM results ($m = 5$).

the impact of changing the time interval used to compute the relative rotations (and hence, approximation of angular speed). We can notice that in general, the best results are obtained with $m = 3$ and $m = 5$, while results with $m = 1$ are lower (affected by potential tracker instability). With $m = 7$, results are more contrasted. These results are confirmed by the precision-recall curve measured at frame-level, shown at the top of Fig.11.

Model comparison. Table.2, Fig.10 and Fig.11 provide a comparison of the different feature vectors. As can be seen, the use of relative rotation features and axis distance produce slightly better results. Indeed, in the configuration of Ubimpressed (see Fig.6), all people are seen from the same viewpoint, and look towards the job interviewer, so we have very similar head pose. Since we train and test from the same dataset, the invariance does not bring much.

To validate that the relative rotation is more robust, we generate a series of synthetic data by systematically rotating the head, to simulate a change of viewpoint. To do so, we transformed the sequence of head pose R_t into $R_t^{'} = R^{vp}R_t$. R^{vp} can simulate a change of pitch and roll (which can be due to being seen from below/above or with an in-plane rotated camera). Thus we trained a model from the original data and tested it on the held out video with the modified viewpoint. The results are shown in Fig.12. While the relative rotation are by definition not affected by such a change, the sequence of Euler angles measured in camera are affected, with a performance reduction of 10% at 20° viewpoint change. Such behavior is also observed on the KTH data.

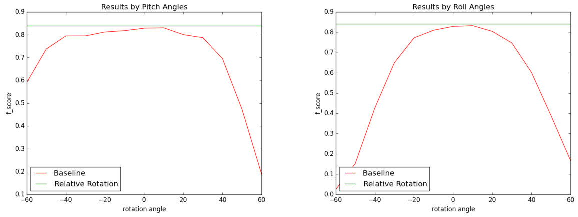

Figure 12. Results obtained by simulating on the test data a viewpoint change. Left: change in pitch (looking from above/bottom). Right: change in roll (looking with more in-plane rotation).

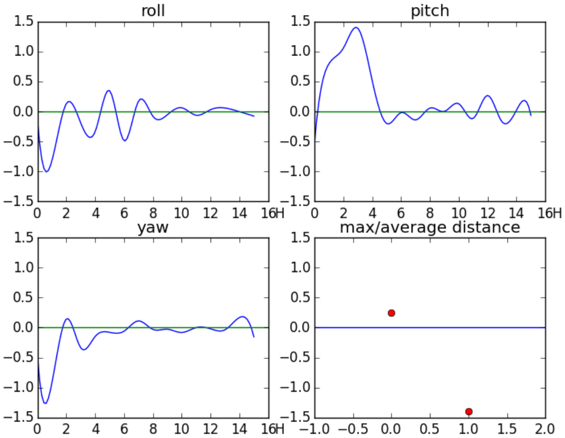

Figure 13. Weights for linear SVM, $m = 3$.

As motivated earlier, the distance to rotation axis could be a useful feature to filter out false alarms due to body movements. However, such behavior appears seldom in our dataset, and thus can not alter much the overall results. Results are thus unconclusive here. In a context where the participants have more degree of freedom (e.g. standing people), it could lead to better performance improvements.

Finally, we show in Fig. 13 the weights of the linear SVM. It can be seen that as expected, the filter reacts to rotation around the pitch in the 1-4Hz range, while is negatively affected by low frequency gestures around the yaw and roll rotations axis, which reflects the fact that real nods should only involve pitch, and not be a composite of rotations. In addition, we can notice that the rotation axis distance feature (esp. the average one) also negatively vote against the nod detection, as we could expect.

Error analysis. Qualitatively, most false positive errors are due to single stroke lowering or raising head gesture (with a small overshoot/oscillation at the end due to momentum control). False negatives come from very light nods or nods accompanied by other gestures, esp. during speaking periods. Note that during speaking time, some very subtle head gestures are difficult to label as nod or not, due to the presence of other head related motions/activities. Thus, some annotations inaccuracy and the trembling of the head pose tracker could influence the result.

4.2. KTH-Idiap Corpus Results

We first evaluated the generalization capabilities of our model and used the nod models trained on Ubimpressed data and tested them on the KTH-Idiap sequences. Results are reported in Table 3. Two main remarks can be made. First, results are lower than on the Ubimpressed data overall (F-score of 0.72 vs 0.84 for events), which can due to multiple factors. Most importantly, as people are more distant to the sensor, and are less facing the camera, tracking (remember that our tracker only relies on depth information) is more difficult and results in noisier head pose sequences. Furthermore, as there are four people, people behave more often as observers, producing some very light nods, often from a side view, which makes their recognition very challenging. Secondly, we can notice that the use of the relative head rotation features results in better performance (0.72 vs 0.68 for events), demonstrating their greater invariance to viewpoint and pose changes.

Finally, we also trained the classifiers on KTH-Idiap dataset using a one-person leave out scheme. Results are shown in Table 4. Surprisingly, results are not necessarily higher than with the model trained on Ubimpressed data, (0.68 vs 0.72 at the event level). This might be due to the use of noisier head pose tracking features during training, which affects the recognition behavior (see the drop in precision). Nevertheless, note that the proposed features still usually perform better than the baseline, especially at the event level.

Table 3. Results of KTH-Idiap data, with nod detector trained on Ubimpressed data.

	$EventLevel$			$FrameLevel$		
	Precision	Recall	F-score	Precision	Recall	F-score
$linear\ SVM, m = 3$						
Baseline	0.73	0.63	0.68	0.72	0.44	0.55
RelRot	0.75	0.69	0.72	0.78	0.47	0.59
RelRot-AxisDist	0.74	0.69	0.72	0.79	0.48	0.6
$RBF\ SVM, m = 3$						
Baseline	0.83	0.54	0.66	0.86	0.38	0.53
RelRot	0.83	0.62	0.71	0.87	0.42	0.57
RelRot-AxisDist	0.81	0.62	0.7	0.86	0.42	0.57
$linear\ SVM, m = 5$						
Baseline	0.74	0.63	0.68	0.76	0.46	0.58
RelRot	0.75	0.7	0.72	0.81	0.5	0.61
RelRot-AxisDist	0.73	0.69	0.71	0.82	0.5	0.62
$RBF\ SVM, m = 5$						
Baseline	0.81	0.57	0.67	0.86	0.43	0.57
RelRot	0.83	0.61	0.7	0.87	0.44	0.59
RelRot-AxisDist	0.83	0.63	0.72	0.87	0.45	0.59

Table 4. Results of KTH-Idiap data, trained on KTH-Idiap data.

	$EventLevel$			$FrameLevel$		
	Precision	Recall	F-score	Precision	Recall	F-score
$linear\ SVM, m = 3$						
Baseline	0.54	0.8	0.65	0.6	0.59	0.59
RelRot	0.54	0.79	0.64	0.59	0.58	0.59
RelRot-AxisDist	0.54	0.79	0.64	0.59	0.58	0.59
$RBF\ SVM, m = 3$						
Baseline	0.57	0.77	0.65	0.57	0.57	0.57
RelRot	0.6	0.8	0.68	0.59	0.58	0.59
RelRot-AxisDist	0.6	0.8	0.68	0.59	0.58	0.59
$linear\ SVM, m = 5$						
Baseline	0.53	0.79	0.63	0.6	0.61	0.61
RelRot	0.57	0.8	0.66	0.61	0.61	0.61
RelRot-AxisDist	0.57	0.8	0.66	0.61	0.61	0.61
$RBF\ SVM, m = 5$						
Baseline	0.55	0.77	0.64	0.57	0.6	0.58
RelRot	0.6	0.79	0.68	0.59	0.62	0.61
RelRot-AxisDist	0.6	0.79	0.68	0.59	0.62	0.61

5. Conclusion

In this paper, we developed a head nod detection system. The system exploits the 3D oscillatory characteristics of nods by relying on the Euler angles extracted from the relative rotation matrix expressed in the camera frame and on the distance of the rotation axis to the face origin. Compared to previous approaches, the method improves the detection and provides accurate results, even in the case of subtle nods. It is possible to extend this method to other head gestures like head shake. Future work can consist of further exploring the role of the rotation axis for recognition, e.g. by testing the system with standing people. In addition, investigation on the exploitation of a temporal model (e.g. CRF) as well as multiple instance learning would be helpful.

Acknowledgement. This work was funded by the UBIMPRESSED project of the Sinergia interdisciplinary program of the Swiss National Science Foundation (SNSF).

References

[1] A.Kapoor and R.Picard. A real-time head nod and shake detector. *In Proceedings from the Workshop on Perspective User Interfaces*, pages 1–5, 2001.

[2] S. Ba and J-M. Odobez. Multi-party focus of attention recognition in meetings from head pose and multimodal contextual cues. In *IEEE Int. Conf. on Acoustics, Speech, and Signal Processing (ICASSP)*, Las-Vegas, march 2008.

[3] K. Funes and J.M.Odobez. Gaze estimation from multimodal kinect data. *In Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2012.

[4] U Hadar, TJ Steiner, EC Grant, and FC Rose. Kinematics of head movements accompanying speech during conversation. *Human Movement Science*, 2:35–46, 1983.

[5] J.Allwood and L.Cerrato. A study of gestural feedback expressions. *In Proceedings of the First Nordic Symposium on Multimodal Communication*, 2003.

[6] D. Jayagopi, S. Ba, J.-M. Odobez, and D. Gatica-Perez. Predicting two facets of social verticality in meetings from five-minute time slices and nonverbal cues. In *Proc. Int. Conf. on Multimodal Interfaces (ICMI)*, 2008.

[7] LP Morency, C Sidner, C Lee, and T Darrell. Head gestures for perceptual interfaces: The role of context in improving recognition. *Artificial Intelligence*, pages 568–585, 2007.

[8] K Nakamura, T Watanabe, and M Jindai. Development of nodding detection system based on active appearance model. *Int. Symp. on System Integration*, 2013.

[9] L Nguyen, JM Odobez, and D Gatica-Perez. Using self-context for multimodal detection of head nods in face-to-face interactions. *Proc. of the 14th ACM Int. Conf. on Multimodal Interactions*, 2012.

[10] JM Odobez and P Bouthemy. Robust multiresolution estimation of parametric motion models. *Journal of visual communication and image representation*, 6:348–365, 1995.

[11] C Oertel, K Funes, S Sheikhi, JM Odobez, and J Gustafson. Who will get the grant? *Int. Conf. on Multimodal Interaction Workshop (UMMI)*, 2014.

[12] Wenzhao Tan and Gang Rong. A real-time head nod and shake detector using hmms. *Expert Systems with Applications*, 25:461–466, 2003.

[13] H Wei, P Scanlon, Y Li, DS Monaghan, and NE O'Connor. Real-time head nod and shake detection for continuous human affect recognition. *In Image Analysis for Multimedia Interactive Services workshop (WIAMIS)*, 2013.