A Discriminative Approach to Frame-by-Frame Head Pose Tracking

Jacob Whitehill and Javier R. Movellan
Machine Perception Laboratory
University of California, San Diego
La Jolla, CA 92093, USA
{jake,movellan}@mplab.ucsd.edu

Abstract

We present a discriminative approach to frame-by-frame head pose tracking that is robust to a wide range of illuminations and facial appearances and that is inherently immune to accuracy drift. Most previous research on head pose tracking has been validated on test datasets spanning only a small (< 20) subjects under controlled illumination conditions on continuous video sequences. In contrast, the system presented in this paper was both trained and tested on a much larger database, GENKI*, spanning tens of thousands of different subjects, illuminations, and geographical locations from images on the Web. Our pose estimator achieves accuracy of* $5.82°$*,* $5.65°$*, and* $2.96°$ *root-mean-square (RMS) error for yaw, pitch, and roll, respectively. A set of 4000 images from this dataset, labeled for pose, was collected and released for use by the research community.*

1. Introduction

Real-time, robust head pose tracking algorithms have the potential to greatly advance the fields of human-computer and human-robot interaction. The two main paradigms are *differential tracking* and *absolute tracking*. Differential trackers (e.g., [1–5]) track the position and orientation of the head through time, often using a geometrical model of the face. They benefit from temporal and/or optic flow information, but they are typically suscepetible to accuracy drift due to accumulated uncertainty over time. They usually also require the initial position and orientation of the head to be initialized, either manually or using a supplemental automatic system.

Absolute tracking approaches (e.g., [6–14]) detect head pose from single images without temporal information and without any previous knowledge of the user's appearance. Absolute trackers ignore all temporal or flow information and hence are inherently immune to accuracy drift. They are suitable for both independent image and video sequence analysis. Such trackers will likely play an important role in integrated pose tracking systems that combine the benefits of differential and absolute approaches.

Most work on automatic pose tracking reports accuracy statistics over test databases spanning a relatively small (< 20) number of subjects in continuous video sequences under laboratory lighting conditions. Even the CAS-PEAL dataset [15], which contains nearly 100,000 face images and is a valuable asset to the face analysis community, was collected under controlled lighting conditions in a laboratory. Such datasets do not give a clear indication as to how well a pose tracker would perform in unconstrained illumination conditions outside the laboratory or across a diverse range of human subjects.

In this paper, we propose a novel frame-by-frame (absolute) pose tracker using an array of Viola-Jones-style [16] classifiers that distinguish between different pose ranges. The outputs of these classifiers are integrated using a regression model to estimate the precise pose angle of all three pose parameters: yaw (side-to-side), pitch (up and down), and roll (rotation in-plane). We evaluate this system in terms of root-mean-square (RMS) pose estimation error on the *MPLab GENKI Dataset*, a large, diverse dataset of Web images, which we describe in Section 2, and of which we are releasing a sizable subset (*GENKI-4K*) for public use. We also showcase the usefulness of frame-by-frame pose analysis in a social interaction setting – detection of spontaneous head-shakes from video.

This paper is structured as follows: In Section 2, we briefly review the main pose tracking paradigms used in the recent literature. Sections 3 and 4 describe the algorithm at a high-level as well as implementation details. In Section 5, we measure the tracker's accuracy and examine quantitatively how it might be improved. Section 6 presents a qualitative example of how the tracker can detect a spontaneous head-shake from video. We summarize and conclude in Section 7.

978-1-4244-2154-1/08/$25.00

2. Previous Work

A number of different approaches exist within the frame-by-frame pose tracking paradigm. Some systems detect the face and its pose simultaneously [6, 17], perhaps using an Active Appearance Model [10]. Some approaches employ nearest-neighbor prototype methods, in which a query face is matched to a training set of prototype faces; the face can then be estimated as the pose of the nearest neighbor, or interpolated between several neighbors. Other approaches extract a set of features directly from the image pixels and then map, using some regression or classification function, directly to the pose [11–14]. Several hybrid systems spanning multiple static approaches exist as well [8, 9].

Li, et al [17] and Srinivasan and Boyer's [18] approaches are arguably the most similar to the one presented in our paper. Both methods estimate head pose by integrating the outputs of a bank of pose range classifiers. In [17], head pose (yaw) is estimated using kernel-PCA and kernelized support vector regression as belonging to 1 of n bins (spanning 10 degrees), and the system is evaluated on a large, presumably diverse database of images. (The dataset was not thoroughly described in the paper.) However, no exact-pose accuracy measurements were reported, only percent-correct statistics of bin assignment. [18] use a polynomial model to estimate the exact pose of query images after projecting them onto multiple eigenspaces. They evaluate their system on a dataset containing less than 10 test subjects.

In this paper, we train and evaluate the accuracy of our pose tracker on the GENKI dataset, which consists of over 60,000 images downloaded from publicly available Internet repositories of personal Web pages. The database spans a wide range of imaging conditions as well as variability in age, gender, ethnicity, and head pose. Human labels of head pose (yaw, pitch, and roll parameters in degrees) are available for most images, as are the locations of the eyes, nose, and mouth. (The pose tracker uses automatically detected eye locations, however.) Poses are labeled using a special labeling program containing a 3-D graphical model of the head. The labeler's task is to align the 3-D head model using the keyboard (to adjust yaw, pitch, and roll) so that its appearance matches that of the face contained in the GENKI image. The program enables both coarse-grained and fine-grained labeling to facilitate efficient and accurate pose coding. A subset of the GENKI database, entitled *GENKI-4K*, containing 4000 randomly selected, pose- and expression-labeled images is available for public use at `mplab.ucsd.edu`.

3. Architectural Overview

The pose tracker we present operates on each video frame independently and in real time. The system, portrayed graphically in Figure 1, works as follows:

1. Given an input video frame, the face is detected using a real-time face detection system (e.g., OpenCV [19]).
2. Facial features are detected automatically as (x, y) coordinates. Specifically, we detect the centers of both eyes (defined as the midpoint between the inner and outer eye corner), the tip of the nose, and the center of the mouth.
3. The face patch is registered and cropped using the locations of the eyes.
4. The cropped face pixels are passed through an array of pose range classifiers that are trained to distinguish between different ranges of yaw, pitch, and roll. Two types of such classifiers are used: *one-versus-one* classifiers that distinguish between two individual pose ranges (e.g., Yaw Range 1 and Yaw Range 4); and *one-versus-all* classifiers that distinguish between one individual and the remaining pose ranges (e.g., Yaw Range 2 and Yaw Ranges { 1, 3, 4, 5, 6, 7 }). The pose range discriminators are trained using GentleBoost on Haar-like box features and output the log probability ratio of the face belonging to one pose range class compared to another. The run-time performance of these classifiers is fast and small compared to the task of face detection.
5. The (x, y) coordinates output by the feature detectors, and the real-valued outputs of the pose range classifiers are integrated using linear regression to yield the estimate of the exact pose angles (yaw, pitch, and roll).

4. Implementation Details

This section describes in more detail some of the steps outlined in Section 3.

4.1. Face and Facial Feature Detection

For face detection, we employed a Viola-Jones-style face detector developed at our laboratory. For facial feature detection, we employed the system described in [20], which returns the maximum a posteriori estimates of the centers of the eyes, tip of the nose, and center of the mouth, given the appearance of the face, and using a prior over relative locations of facial features.

4.2. Face Registration

Using the automatically detected locations of the eye centers, the face region is cropped (see Figure 2) according to the following measurements: Given the distance d between the centers of the eyes, the size of the cropped face is set to be md. We found that $m = 3.125$ yielded good pose tracking accuracy. The top of the cropped face is set such that the distance from the top to the midpoint of the centers of the eyes equals kd. We set $k = 0.875$.

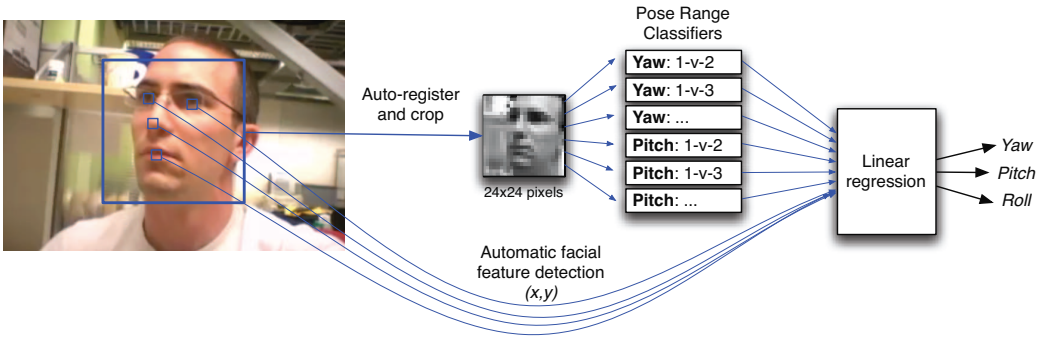

Figure 1. The high-level architecture of the pose tracker. The face and eye positions (x, y) are detected automatically. The face patch is cropped about the eyes and then classified by an array of pose classifiers that discriminate between different yaw and pitch ranges (defined in Figure 3). The classifier outputs and facial feature locations are then integrated, using linear regression, into an estimate of the three pose angles.

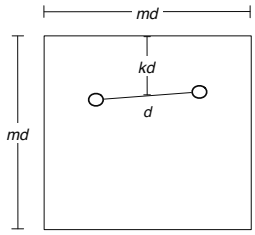

Figure 2. The face measurements by which faces were cropped from the surrounding image. Given the locations of the centers of the eyes (represented in the figure by the circles), the distance d between them is calculated. The size of the face md, and the y-distance kd between the midpoint of the centers of the two eyes and the top of the cropped face, are then calculated. We found that $m = 3.125$ and $k = 0.875$ yielded good pose tracking accuracy.

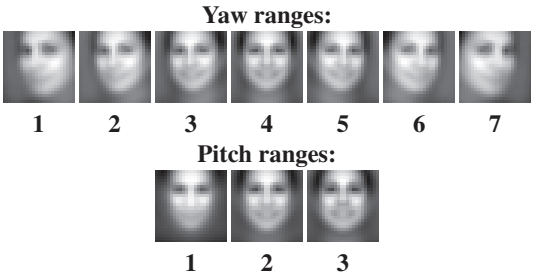

Figure 3. The average images from each of the seven yaw and three pitch ranges used in our implementation. The yaw ranges are (from 1-7): $[-45, -30]$, $[-30, -18]$, $[-18, -06]$, $[-06, +06]$, $[+06, +18]$, $[+18, +30]$, and $[+30, +45]$, in degrees. The pitch ranges (from 1-3) are: $[-45, -10]$, $[-10, +10]$, and $[+10, +45]$.

Once the face region is cropped, it is downscaled to 24x24 pixels, converted to grayscale, and normalized to zero mean and unit variance.

4.3. Training the Appearance-based Pose Range Classifiers

In our implementation we partitioned the Yaw space into seven ranges and the Pitch space into three ranges; these ranges are listed below Figure 3. Since the roll angle of a face can be accurately estimated using feature point positions alone, we did not train any classifiers to discriminate between ranges in Roll space. One-versus-one and one-versus-all classifiers were trained for Yaw and Pitch.

For training the classifiers, we used 80% (approximately 28,000 images) of the GENKI dataset (described in Section 2) for which pose labels existed and in which the face detector found a face. The remaining 20% of the GENKI images were used for validation. In particular, the now publicly available GENKI-4K dataset is contained completely in this validation set.

Each appearance-based pose range classifier was trained using GentleBoost [21], which is a variant of boosting whose final output is the log probability ratio of the class (yaw/pitch range) given the input (the face). For example, for the 1-v-{2,3} pitch classifier, the classifier's output is the log probability ratio of the face belonging to Pitch Range 1 to the face belonging to *either* Pitch Range 2 or 3. As a feature set, we used the same set of box filters (Haar-like wavelets) as Viola and Jones in their original face detector [16]. In essence, the individual pose range classifiers we use are single-cascade Viola-Jones face classifiers. All classifiers were trained for 500 rounds, i.e., they contain 500 weak classifiers.

4.4. Linear Regression

The (x, y) outputs of the facial feature detectors and the real-valued outputs of the pose range classifiers are combined to form an estimate of the exact head pose using standard linear regression. Specifically, the inputs to the linear regression function are the raw outputs of the facial feature detector, and the arctangent ($\arctan$) of the outputs of the pose range classifiers. We found that using $\arctan$ as a transfer function prior to regression improved results signficantly. This may be because $\arctan$ has bounded range and thus limits the effect that any single yaw range classifier can have on the final yaw estimate.

As alternatives to linear regression, we also tested ridge regression and ε-SVM regression (linear and RBF kernels) but did not find an improvement in accuracy.

5. Experimental Analysis

We measured the accuracy of the pose tracker on the GENKI-4K dataset (see Section 4.3), containing 4000 GENKI images not used for training. Accuracy was measured as the root-mean-square (RMS) error of estimating separately the yaw, pitch, and roll of the head with the human pose-labels as ground-truth. For comparison, we also estimated the accuracy of human pose labels by computing the average human labeling error (RMS) over 671 GENKI images which were labeled by at least four different human coders. The mean pose label for each image was taken as ground truth.

Overall Accuracy: The overall accuracy of the pose tracker, using both the facial feature locations and appearance-based information via the pose range classifiers, was $5.82°$, $5.65°$, and $2.96°$ RMS error for yaw, pitch, and roll, respectively, on the GENKI-4K dataset. Automatic pose tracking accuracy compared to human accuracy is portrayed graphically in Figure 4. For pitch estimation, the accuracy of the automatic system is comparable to that of human labelers. Estimation could be improved significantly for yaw. The inter-human accuracy for roll was particularly low because roll was computed automatically from eye coordinate positions that the humans coded.

In the following subsections, we examine which components of the system's architecture were most instrumental in achieving this accuracy, and also examine how error varies as a function of the pose itself.

5.1. Benefit of Feature Point Coordinates

The tracker's pose estimates are based on two input sources: the geometry of the face as represented by the location of automatically detected facial features, and appearance information extracted from the cropped face. To assess the contribution of each input source, we trained two additional classifiers by restricting the linear regression weights: Classifier *Appearance*, which uses only appearance information, and Classifier *Geometry*, which uses only facial feature locations. We refer to the original classifier which uses both input sources as *Combined*. The RMS yaw estimation error of the Appearance, Geometry, and Combined classifiers are shown in the following table:

Approach	**Estimation Error (RMS, in degrees)**		
	Yaw	**Pitch**	**Roll**
Appearance-based	5.96	5.82	6.81
Facial feature locations	8.94	6.27	2.93
Combined	5.82	5.65	2.96

These results indicate that nearly all of the useful yaw information comes from appearance information from the face. Since no pose discriminators were trained for roll, it is not surprising that appearance-based roll estimation was much less accurate. Pitch seems to benefit from both appearance-based and feature point-based information. It is also possible that, by using more facial feature locations, such as are available in an Active Appearance Model [22], the utility of facial feature locations would increase.

5.2. Benefit of *One-versus-One* versus *One-versus-Rest* Classifiers

To assess the relative importance of the one-versus-one compared to the one-versus-rest appearance-based classifiers, we measured yaw estimation accuracy using these input sources alone (without facial feature coordinates). We compared accuracy for yaw and pitch (since no classifiers were trained to discriminate roll ranges):

Approach	**Estimation Error (RMS, in degrees)**	
	Yaw	**Pitch**
One-versus-one	6.06	5.86
One-versus-rest	6.38	5.83
Combined	5.96	5.82

It thus seems that the one-versus-one pose range classifiers contain more signal than do the one-versus-rest classifiers.

5.3. Error as a Function of Pose

Figure 4 displays the RMS error of the automatic pose tracker as a function of the human-labeled pose. Each of the three RMSE figures form an approximately U-shaped curve, meaning that pose estimation is less accurate for poses farther from frontal ($0°$).

One possible explanation for this result is that the number of images in our training set is greater for near-frontal views than for views far from frontal, as displayed in Figure 5. The correlations between the number of training examples at each pose angle with the corresponding RMS error

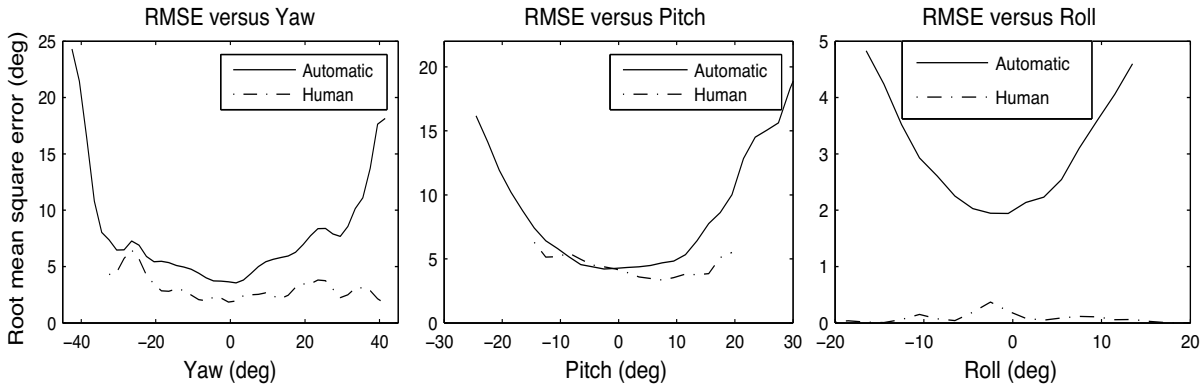

Figure 4. Smoothed root-mean-square errors (RMSE), as a function of human-labeled pose, for both the automatic pose tracker and the individual human labelers. RMSE for the automatic pose tracker was estimated over GENKI-4K using the average human labeler's pose as ground-truth. RMSE for humans was measured on a different subset of GENKI comprising 671 images on which at least 4 different humans had labeled pose. Human error for roll was not available since roll was computed automatically using the labeled eye positions.

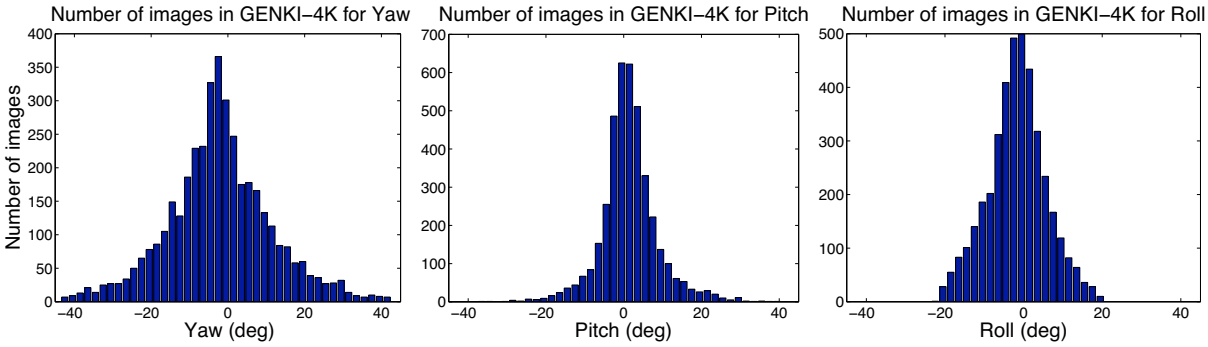

Figure 5. Histogram of training images as a function of the human-labeled pose.

of yaw pitch, and roll, were -0.55, -0.57, and -0.82, respectively. Not surprisingly, this means that, the more training examples that were available, the smaller the estimation error. This suggests that we can improve the accuracy of the pose tracker by collecting more training examples in the underrepresented yaw regions. We are currently in the process of expanding our GENKI dataset to include more non-frontal poses.

5.4. Synthetic Training Examples

In an effort to increase the number of training images in our dataset for the outer poses (see Section 5.3), we augmented GENKI with a number of synthetically generated face images. Collecting high-quality, accurately labeled image databases can be time-consuming and tedious, and the ability to generate precisely (pose-)labeled images automatically would be extremely useful. Previous research has used synthetically generated images for recognizing articulated full-body poses [23]. Here, we present our preliminary results for the task of *yaw* estimation:

We used the 3-D face rendering software *Poser* [24] to generate several thousands of face images of varying poses. Specifically, we automatically rendered 3200 face images in the yaw ranges of $[-35, -25]^\circ$, $[-10, 10]^\circ$, and $[25, 35]^\circ$ in 1° increments, using five different facial appearance models and a variety of different facial expressions (to increase the diversity of the images) using *PoserPython*, Poser's scripting language. Ethnicity parameters are, to our best deduction, not accessible from PoserPython and had to be set manually. The roll and pitch parameters were varied across the range of $[-8, 8]^\circ$ in 2° increments. All of the rendered images were then superposed onto a random set of background images (not containing faces) to simulate the geographic variability of GENKI. These synthetic images were added to the GENKI training set, but not to the test set. Examples of Poser-generated images and a random sample of GENKI are displayed, for comparison, in Figure 6.

Preliminary results have shown a marginal improvement

GENKI (natural):

Poser (synthetic):

Figure 6. **Above**: A random subset of GENKI images from Yaw Range 1. **Below**: A random subset of synthetic images generated using *Poser 7* and superimposed on random backgrounds from the yaw range of $[-10, 10]^\circ$.

.

in accuracy by augmenting the training set of naturally occuring images (GENKI) with synthetically generated ones (0.05° RMS improvement for yaw estimation). In our experiment, randomly selecting a subset ($\frac{1}{8}$th) of the artificially rendered faces yielded better performance than including all of them. An important question which we will address in future research is whether the statistics of the synthetically generated images – e.g., spatial frequencies of face images, diversity of ethnicity, illumination conditions – match those of GENKI and are thus suitable for training.

6. Head Gesture Recognition

Estimating head pose is an important signal in many human-computer interaction and human-robot interaction applications. The head yaw can indicate, for example, that the human user is looking toward the computer (attentive) or looking in a side direction (inattentive). Oscillation of the yaw may indicate the user is shaking her head to say "no."

As a qualitative example of the pose tracker's ability to capture such gestures, we show the pose tracker's output on a video clip of a user shaking her head "no." Sample video frames (every 4th frame) are shown beneath the graph as well as time-aligned video frames taken at the peaks and valleys of the yaw tracker's output. No information is shared between frames, and no smoothing was employed. Though the head-shake is subtle, the automated yaw output clearly shows the head-shake. While this example is only anecdotal, it is an encouraging indicator of the usefulness of the frame-by-frame pose tracker we developed.

7. Conclusions and Further Research

Our results on a large, diverse image database indicate that an array of discriminative pose range classifiers, integrated using linear regression, can yield accuracy levels close to that of humans using precise 3-D graphical labeling software.

In future work we will examine whether head pitch (up and down movement) can be estimated accurately using the same architecture as described in this paper. We will also collect more training examples in the yaw regions for which relatively few images exist in the GENKI dataset, and continue to experiment with synthetically generated faces for training.

Acknowledgement

This research was supported in part by NRL 55-05-03.

References

[1] M.J. Black and Y. Yacoob. Tracking and recognizing rigid and non-rigid facial motions using local parametric models of image motion. In *ICCV*, pages 374–381, 1995.

[2] M. La Cascia, S. Sclaroff, and V. Athitsos. Fast, reliable head tracking under varying illumination: An approach based on registration of textured-mapped 3D models. *PAMI*, 22(4):322–336, April 2000.

[3] S. Basu, I.A. Essa, and A.P. Pentland. Motion regularization for model-based head tracking. In *Proceedings. International Conference on Pattern Recognition*, 1996.

[4] A. Schodl, A. Haro, and I. Essa. Head tracking using a textured polygonal model. In *PUI98*, 1998.

[5] L. Wiskott, J.M. Fellous, N. Kruger, and C. von der Malsburg. Face recognition by elastic bunch graph matching. *PAMI*, 19(7):775–779, July 1997.

[6] Chang Huang, Haizhou Ai, Yuan Li, and Shihong Lao. High-performance rotation invariant multiview face detection. *Pattern Analysis and Machine Intelligence*, 29(4), 2007.

[7] Junwen Wu and Mohan M. Trivedi. An integrated two-stage framework for robust head pose estimation. In *International Workshop on Analysis and Modeling of Faces and Gestures*, 2005.

[8] Y. Fu and T. Huang. Graph embedded analysis for head pose estimation. In *Proc. IEEE Intl. Conf. Automatic Face and Gesture Recognition*, pages 3–8, 2006.

[9] Vineeth Nallure Balasubramanian, Sreekar Krishna, and Sethuraman Panchanathan. Person-independent head pose estimation using biased manifold embedding. *EURASIP Journal on Advances in Signal Processing*, 2008.

[10] T. F. Cootes, G. V. Wheeler, K. N. Walker, and C. J. Taylor. View-based active appearance models. *Image and Vision Computing*, Volume 20:657–664, August 2002.

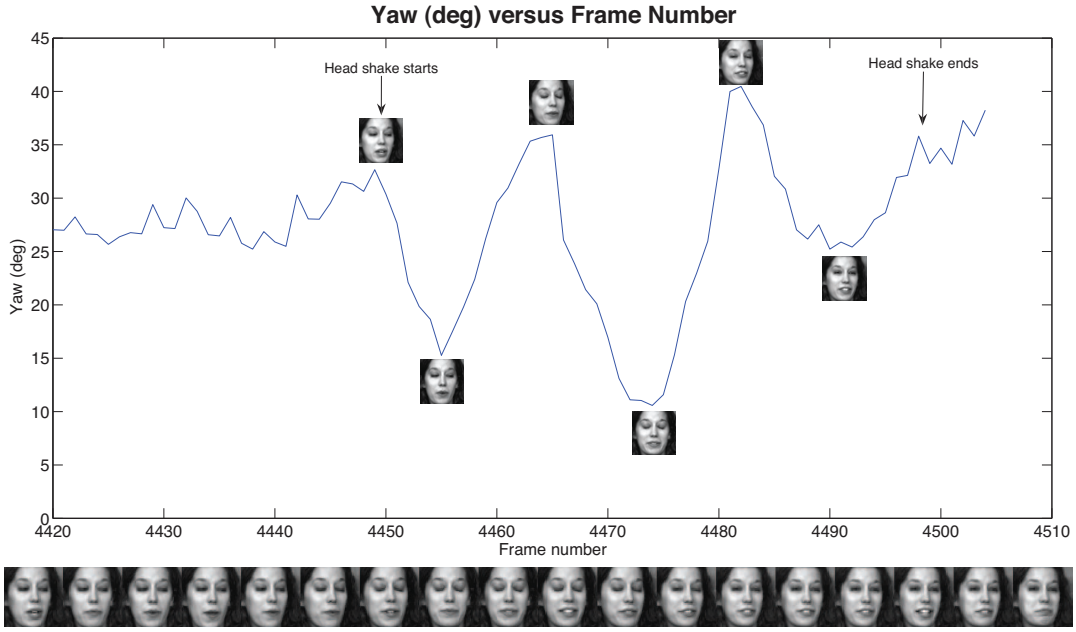

Figure 7. **Above**: Unsmoothed output of the pose tracker on a video sequence of a spontaneous head shake. **Below**: Sample video frames from the head-shake video. Images courtesy of Professor Mark Frank, SUNY Buffalo.

[11] J. Huang, X. Shao, and H. Wechsler. Face pose discrimination using support vector machines (svm). In *Proc. Intl. Conf. Pattern Recognition*, pages 154–156, 1998.

[12] Erik Murphy-Chutorian, Anup Doshi, and Mohan Manubhai Trivedi. Head pose estimation for driver assistance systems: A robust algorithm and experimental evaluation. In *Intelligent Transportation Systems*, 2007.

[13] N. Gourier, J. Maisonnasse, D. Hall, and J. Crowley. Head pose estimation on low resolution images. *ser. Lecture Notes in Computer Science*, 4122:270–280, 2007.

[14] A. Lanitis, C. Taylor, and T. Cootes. Automatic interpretation of human faces and hand gestures using flexible models. In *Proc. IEEE Intl. Conf. Automatic Face and Gesture Recognition*, pages 98–103, 1995.

[15] W. Gao, B. Cao, S.Shan, X.Chen, D.Zhou, X.Zhang, and D.Zhao. The cas-peal large-scale chinese face database and baseline evaluations. *IEEE Trans on System, Man and Cybernetics - Part A*, 38(1), 2008.

[16] Paul Viola and Michael Jones. Robust real-time object detection. *International Journal of Computer Vision*, 2002.

[17] S. Li, Q. Fu, L. Gu, B. Scholkopf, Y. Cheng, and H. Zhang. Kernel machine based learning for multi-view face detection and pose estimation. In *Proc. IEEE Intl. Conf. Computer Vision*, pages 674–679, 2001.

[18] S. Srinivasan and K.L. Boyer. Head pose estimation using view based eigenspaces. In *International Conference on Pattern Recognition*, 2002.

[19] Intel Corporation. OpenCV face detector, 2006.

[20] M.R. Eckhardt, I.R Fasel, and J.R Movellan. Towards practical facial feature detection. *Submitted to International Journal of Pattern Recognition and Artificial Intelligence (IJPRAI)*, 2008.

[21] Jerome Friedman, Trevor Hastie, and Robert Tibshirani. Additive logistic regression: a statistical view of boosting. *Annals of Statistics*, 28(2), 2000.

[22] T.F. Cootes, G.J. Edwards, and C.J. Taylor. Active appearance models. *PAMI*, 23(6):681–684, June 2001.

[23] G. Shakhnarovich, P. Viola, and T. Darrell. Fast pose estimation with parameter sensitive hashing. In *Proceedings of the International Conference on Computer Vision*, 2003.

[24] e frontier. Poser 7.