

Tracking Head Yaw by Interpolation of Template Responses

Mario Romero and Aaron Bobick

College of Computing
Georgia Institute of Technology
Atlanta, Georgia 30318
Email: {mromero, afb}@cc.gatech.edu

Abstract

We propose an appearance based machine learning architecture that estimates and tracks in real time large range head yaw given a single non-calibrated monocular grayscale low resolution image sequence of the head. The architecture is composed of five parallel template detectors, a Radial Basis Function Network and two Kalman filters. The template detectors are five view-specific images of the head ranging across full profiles in discrete steps of 45 degrees. The Radial Basis Function Network interpolates the response vector from the normalized correlation of the input image and the 5 template detectors. The first Kalman filter models the position and velocity of the response vector in five dimensional space. The second is a running average that filters the scalar output of the network. We assume the head image has been closely detected and segmented, that it undergoes only limited roll and pitch and that there are no sharp contrasts in illumination. The architecture is person-independent and is robust to changes in appearance, gesture and global illumination. The goals of this paper are, one, to measure the performance of the architecture, two, to assess the impact the temporal information gained from video has on accuracy and stability and three, to determine the effects of relaxing our assumptions.

1 Introduction

Understanding the user’s focus of attention is critical for a variety of human-machine interactions, and head orientation is a key component of assessing gaze and attention focus. Head orientation can be specified given three rotational parameters: roll, pitch and yaw. In the process of understanding human activity, yaw is the most important rotation of the head; our most common interactive environments, including the interpersonal, typically unfold horizontally.

Roll is the rotation about the sagittal-horizontal axis (the axis normal to the nose). Pitch is the rotation about the frontal-horizontal axis (the axis that passes through the

ears). Yaw is the rotation about the longitudinal axis (the vertical axis that passes through the center of the head). We report yaw as the angle between the optical axis of the camera and the sagittal-horizontal axis of the head, where a frontal view is 0° and the full profile views are $\pm 90^\circ$.

In this paper we present, first, a machine learning architecture that rapidly estimates full 180 degree yaw from a single image of the head. We purposely name the frame as “head image,” as opposed to “face image”, because large portions of the image are hair, specially when the view is not frontal. The architecture is appearance based. It interpolates the “response” to five uniformly sampled templates between full profile views of the head. We refer to the scalar value of the normalized correlation between the input image and a single template detector as the *response* of the template detector. The response of the input image to the five template detectors is the *response vector*. The interpolation mechanism is a Radial Basis Function Network (RBFN). Its input is the response vector normalized to have unit L2 norm and its output is a real-valued scalar estimating head yaw in degrees. We assume head detection, tracking, and segmentation have been performed. We place a white background behind the subjects’ heads to render these preprocessing steps simple, fast and accurate. Furthermore, we assume that the head is subject to only small changes in roll and pitch and that there are no sharp lights over the face that change its overall appearance. Ambient illumination changes, on the other hand, are part of the system. We do not use color or high resolution images nor do we calibrate the camera.

Second, we present two dynamic models and their impact on the tracking accuracy and stability of the system. The first model is a five-dimensional Kalman filter running over the response vector. It plots the response vector as a point in five-dimensional space and it tracks its position and velocity. The filtered response vector is the new input of the RBFN. The second Kalman filter is a one-dimensional first order running average that filters the scalar output of the network, i.e., angular position. We use two filters because

the RBFN amplifies the noise in the signal nonlinearly. In other words, although the input to the RBFN is sufficiently filtered at its own scale, the output suffers from noise unevenly amplified by the network. Next, we determine the impact of the the two filters independently and in conjunction. By measuring the influence the models have on the system we can determine the advantage temporal information from the image sequence has over single frame estimation. We do not model the physics of head motion.

Third, we report the empirical performance of the system and we expose the effects that relaxing our assumptions has on performance. Finally, we propose the requirements this system would need for a real-life online application in an indoor environment.

In section two, we present related work in frontal and multiple pose face detection, and face pose detection and tracking. In section three, we give an overview of the architecture. In section four we describe our methods, data structures and algorithms. In section five we present our experimental results and, we measure how our performance decays as our assumptions are relaxed and we propose the requirements for a real-life online indoor application.

2 Related Work

Face detection from a frontal view have been extensively studied since the early days of vision with some recent exceptional results [14, 15, 16]. Non-frontal view face detection and alignment, which is also necessary for our system, has been researched successfully at length as well [4, 8, 18]. Our approach to tracking head yaw assumes accurate multiple pose head detection based on these positive results.

There also exists an abundance of methods for face pose estimation and head orientation recognition. These methods can be classified into two different general categories: model based and appearance based [6]. The model centric systems construct a 3D model of the head and recover orientation by matching 2D features from one or more images to the 3D model [1, 3, 6]. In contrast, appearance based systems assume a structural relationship between the 3D rotations and motions of the head and its 2D projection [5, 7, 12, 13, 17]. Typically, these systems rely on large sets of training data. The assumption that the 2D projection can be mapped back to 3D holds given that the perspective projection is weak, that the changes in illumination are not drastic and that the background can be effectively removed. For many applications, like activity recognition in an indoor environment, these assumptions are reasonable. Our system belongs to this second group. Essa et. al. [3] present tracking facial motion with a continuous time Kalman Filter. Ong et. al [13] present a one-dimensional running average temporal filter on the output of their system. Finally,

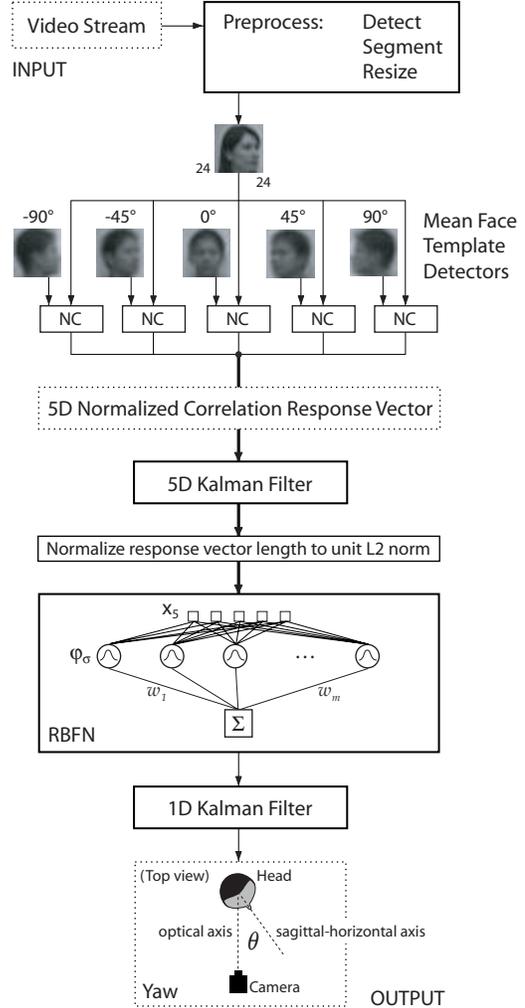


Figure 1: Architecture. Doted line boxes are data. Solid line boxes are processing steps. NC means normalized correlation. The bold data path represents response vector transfer.

Brunelli and Poggio present an interesting analysis of template response correlation dependency on yaw rotation that is reminiscent of our template response interpolation argument [2].

3 Architecture Overview

Figure 1 shows the architecture of the online system. The system receives as input a single grayscale video stream from a monocular uncalibrated camera. The preprocessing steps are detect, segment and resize the image of the face. A set of five templates have been precomputed by averaging training images of faces at five discrete yaw angles.

The first Kalman filter tracks the response vector as a

point with position and velocity in 5-dimensional space. We normalize the output of the filter to have unit L2 norm. The RBFN interpolates the 5-dimensional filtered response vector into a single scalar value that corresponds to the yaw angle.

We introduce a second Kalman filter that is equivalent to a 1-dimensional first order running average to smooth the output of the RBFN. The output of the system is a single scalar value for every frame in the image sequence. It is the estimate of yaw only.

4 Appearance based model, response interpolation and dynamic models

4.1 Appearance based model

Appearance based models for face orientation recognition rely on a large number of training images taken at intervals that are a few degrees apart [13]. We train our system with relatively few sample views, 5, that are 45° from each other (a significant gap). The key is that we interpolate not on the original images, but on the response vector. We collected 1000 training images: 20 frames with free varying gestures and the smallest possible pitch and roll from 10 subjects for each of the 5 views. Our samples include both light and dark skinned people, men and women, long and short hair, glasses and no glasses and one subject with facial hair.

The illumination is not controlled, but it is relatively constant. The original video resolution is 120 by 160 pixels. Since we are assuming a correct detection we have used images with white background to simplify the detection and segmentation. The head images are then cropped and aligned using a simple object detection algorithm. We find the edges of the head image using a Sobel edge detector with automatic thresholding [10]. We compute the vertical and horizontal extremes of the edge image, regardless of head pose, and we crop it to the tightest bounding square. We leave whatever background is left after cropping.

After the image of the head has been detected and segmented, it is resized to 24 by 24 pixels maintaining the aspect ratio of the head. Thus, not all images from different views include the same portion of the white background. For instance, frontal views have more background on the sides of the image than profile views because the head is narrower from the front (figure 2). The views are labeled by their yaw angle in degrees. We define yaw as the angle between the optical axis of the camera and the sagittal-horizontal axis of the head. The sagittal-horizontal axis of the head is the normal vector of the outermost tip of the nose or, equivalently, the normal vector of the cranial coronal plane. We present a top view visualization of yaw in the output of the architecture in figure 1. We sample the train-



Figure 2: Mean face template detectors at yaw angles -90° , -45° , 0° , 45° and 90° . The resolution of the templates is 24 by 24 pixels.

ing images at -90° (full right profile), -45° , 0° (frontal view), 45° and 90° (full left profile) while we maintain roll and pitch as constant as possible at 0° . The final step of this first part of the training process is building the five discrete view mean face template detectors as the average of the samples of the 200 preprocessed face images from each of the 5 views (figure 2).

Normalized correlation is a well known method for computing a template distance measurement. Given an image $I(x, y)$ and a template $T(x, y)$ of the same dimensions, the normalized correlation between them is defined as:

$$NC(I, T) = \frac{\sum_{x,y} (I - \bar{I})(T - \bar{T})}{\sqrt{\sum_{x,y} (I - \bar{I})^2 \sum_{x,y} (T - \bar{T})^2}} \quad (1)$$

where $\bar{I} = \frac{1}{n} \sum_{x,y} I$.

The crucial part of the system is computing a “stable” normalized correlation response vector from which interpolation is possible. We define a response vector to be *stable* if the response to varying input fluctuates smoothly and is reproducible.

Finally, we normalize the vector to have unit L2 norm. This normalization places the response vector from different subjects onto the same scale.

To visualize the stability of the response vector, we compute the normalized correlation between all the training images and the five template detectors. Then, we compute the mean and standard deviation of the response of each template to the 200 images of each view (figure 3). Notice that the response of the templates is symmetric, evidencing the vertical visual symmetry of the head. Note also that the falloff of the response vector from its peak is monotonic, gradual and has a low standard deviation (average standard deviation is 0.035). This yields a smoothly varying response vector as a function of angle. The gradual falloff and the low standard deviation is what enables the RBFN to interpolate continues angles from the response vector.

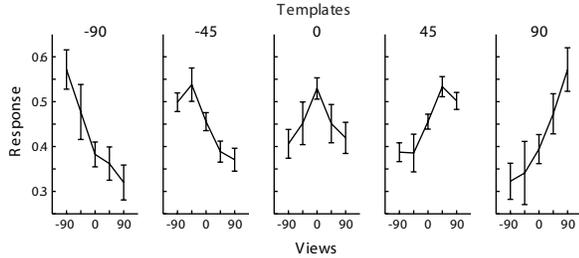


Figure 3: Mean and standard deviation of the normalized correlation response of the template detectors to the 200 samples from each view.

4.2 Response Interpolation

Radial Basis Function Networks (RBFN) are well studied methods for interpolating signals and for nonparametric regression [11]. We train an RBFN using Gaussian bases to interpolate the large gaps between the response of the five discrete views. Our training samples for the RBFN are the response vectors of the 1000 images (200 from each view) and their corresponding labels. The network interpolates the yaw angle θ as a function of the input response vector \mathbf{x} as:

$$\theta = f(\mathbf{x}) = \sum_{i=1}^m w_i \varphi_{\sigma}(\|\mathbf{x} - \mathbf{c}_i\|) \quad (2)$$

where $\varphi_{\sigma}(r) = \exp(-\frac{r^2}{2\sigma^2})$ is the gaussian basis, \mathbf{c}_i is the center and w_i is the weight of each basis. The two parameters of the RBFN are m , the number of bases, and σ , the spread of the bases. In our model, the spread is constant. Figure 1 shows the structure of the RBFN.

To determine the optimal number of bases and their spread we use gradient descent over the output of a training video 442 frames long at 15 frames per second of continuous rotating image samples. The samples are from a subject in our training set. We found that the best training performance was obtained with 24 bases and a spread of 0.6.

The gradient descent is performed on the testing output of the RBF network. The gradient is computed over the sum of the squared errors between the output of the network and the ground truth from an overhead camera. We label the continuous angle of yaw over the image sequences of the training and testing videos using two green markers invisible to the frontal camera but clearly detectable from an overhead camera. Both cameras run at the same frame rate and their input sequences are aligned and synchronized. Using straightforward color channel thresholding and 2D geometry we automatically label the overhead input image stream.

4.3 Dynamic models

The original response vector from the template detectors is cluttered by noise (figure 4b). To clear the noise we use two Kalman Filters (KF), one at the input and one at the output of the RBFN [9]. The first filter models the five-dimensional input response vector, \mathbf{x} , as position in 5D space and its velocity, $\dot{\mathbf{x}} = \frac{d\mathbf{x}}{dt}$. The second KF is a one-dimensional first order running average of the output of the network. We use two Kalman Filters because of the nonlinearity of the gaussian radial bases. In other words, even though the first filter outputs a smooth signal, the RBFN amplifies the noise unevenly (nonlinearly). The second filter is designed to smooth the output of the RBFN.

For the first KF, the state vector \mathbf{y} is ten dimensional, where the first 5 dimensions model position and the last 5, velocity, i.e., $\mathbf{y} = [\mathbf{x}\dot{\mathbf{x}}]^T$. The Kalman filter we use is:

$$\mathbf{y}_{t+1} = F\mathbf{y}_t + \mathbf{w}_t \quad (3)$$

$$\mathbf{z}_t = H\mathbf{y}_t + \mathbf{v}_t \quad (4)$$

where, \mathbf{z} models the 5D response of the detectors. $F_{[10 \times 10]}$ and $H_{[5 \times 10]}$ are the state transition model and measurement model, respectively. The noise of the model is \mathbf{w} and \mathbf{v} is the measurement noise. We model the process noise as $\mathbf{w} = N(0, Q)$ and the measurement noise as $\mathbf{v} = N(0, R)$. $Q_{[10 \times 10]}$ is the process noise covariance matrix and $R_{[5 \times 5]}$ is the measurement noise covariance matrix. Our model of the process noise is $Q = \eta \begin{bmatrix} 0 & 0 \\ 0 & I \end{bmatrix}$. The lower right quarter of Q is diagonal because we assume the noise due to view condition variation is independent and because we are adding noise to the velocity term only. It has a single smoothing parameter, η , because the feature vectors' lengths are normalized. To build R , we first subtract the mean template response from all the samples of each view and then we compute the covariance of the mean-subtracted response. R is:

$$10^{-3} \times \begin{bmatrix} 1.864 & 0.667 & -0.701 & -0.688 & -1.640 \\ 0.667 & 0.681 & -0.029 & -0.559 & -0.999 \\ -0.701 & -0.029 & 1.374 & 0.167 & -0.065 \\ -0.688 & -0.559 & 0.167 & 0.665 & 0.892 \\ -1.640 & -0.999 & -0.065 & 0.892 & 2.389 \end{bmatrix}$$

Notice the correlation in R decreases as the detectors are farther apart.

In Figure 4 we show the influence that filtering the response vector has on both the vector and the output of the network. The $\pm 90^\circ$ detectors benefit the most because they generate more noise. Accordingly, the stability of the filtered output also improves the most near full profile views of the head. Large angle views generate more noise because they produce greater inter subject variation at low resolution. The feature with greater influence on the output of the detectors is the contrast between face and hair areas. Without the Kalman filter, detection at angles greater than 70°

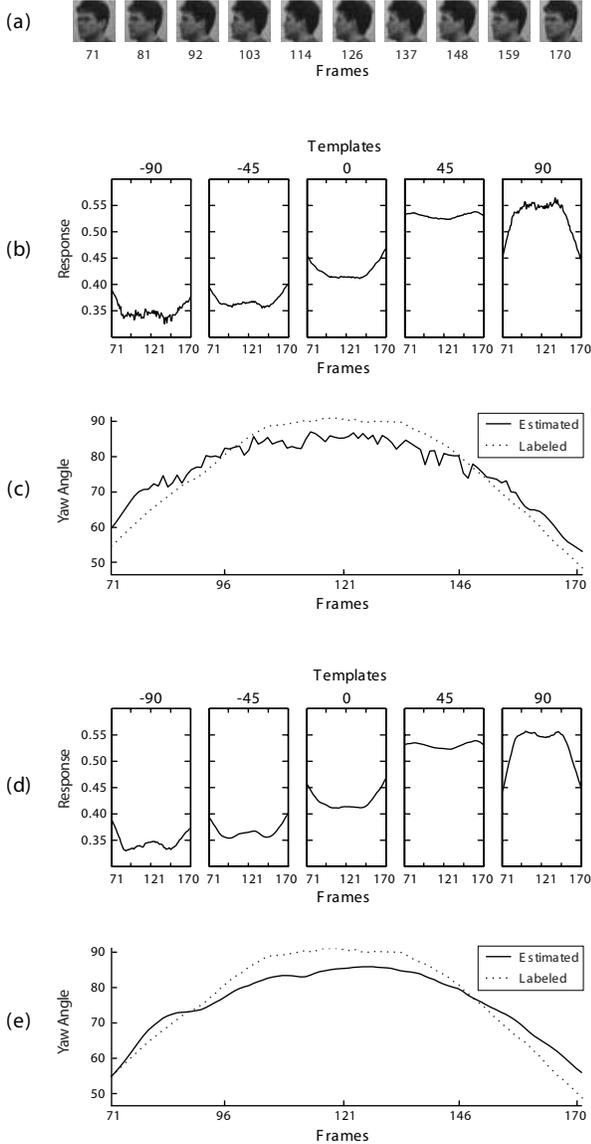


Figure 4: Normalized correlation response vector and RBFN output from a 100 frame subsequence of the training video. (a) 10 evenly distributed sample frames from the sequence, (b) unfiltered response vector (each template response is shown on a separate graph using the same scale), (c) RBFN output to unfiltered response vector, (d) Kalman filtered response vector and (e) RBFN output to filtered response vector. The dotted line is the ground truth yaw from the overhead camera.

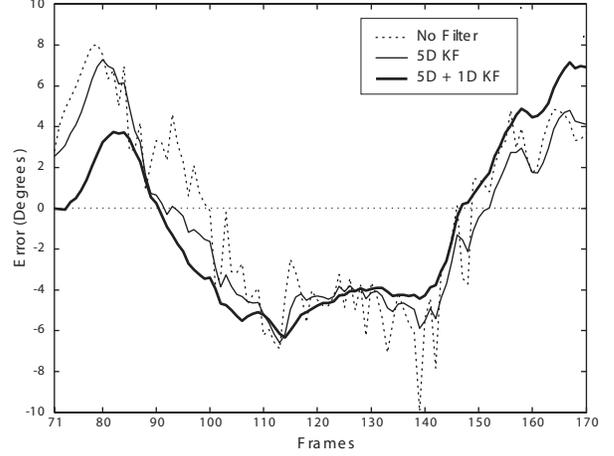


Figure 5: Error over the 100 frame subsequence of the training video. The error is reported in degrees with respect to ground truth. The dotted line is the unfiltered estimation error. The solid line is the result of just using the 5D KF. The bold line is the result of using both filters. Overall accuracy does not change. Filtering only stabilizes the signal.

is unstable. On the other hand, the overall accuracy of the estimate remains unchanged.

The second KF models the scalar network output, θ , as a position in a single dimension and computes a first order running average:

$$\theta_{t+1} = \beta\theta_t + (1 - \beta)\hat{\theta}_t, \quad (5)$$

where β is the smoothing parameter. We use a second filter because the RBFN is nonlinear. The second filter has effect on the unevenly amplified noise and on the overshooting of the first KF, which models velocity. In Figure 5 we show the effect of each filter. Notice the overall accuracy does not change. While some errors are reduced, others are increased. Only the stability of the signal is improved. Filtering with the 1D KF smooths out the remaining noise from the 5D KF, which was not evident from the response vector, but is amplified by the RBFN. The cost of using the running average is lag, evident in frames 155 through 170.

Finally, we optimize η and β by gradient descent on the stability of the training video output. The optimization returns $\eta = 3.1 \times 10^{-6}$ and $\beta = 0.7$.

5 Experimental Results

In order to test the performance of the system we gathered free moving continuous samples of subjects in the training set and new subjects. Figure 6 illustrates the online system. The system runs at over 19 frames per second, on Matlab code on a pentium 4 at 2.4 GHZ with 1GB of memory.

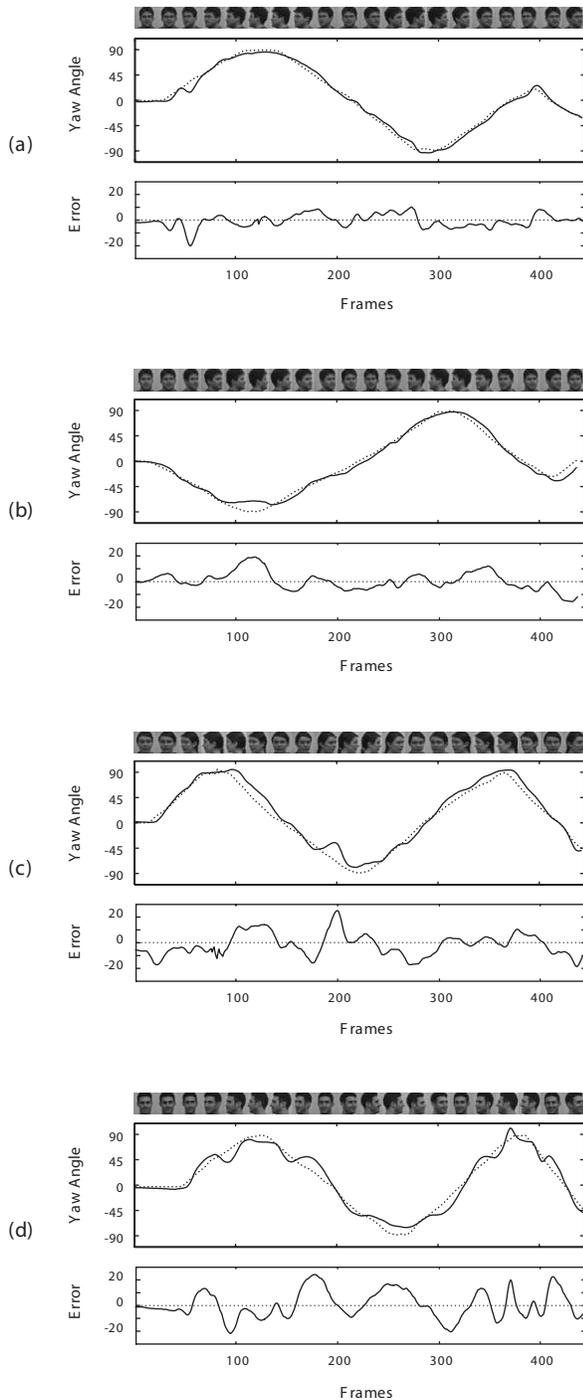


Figure 6: Tracking head yaw when assumptions hold. The figure shows sample frames, the estimate as a solid line, ground truth as a dotted line, and the error in degrees. (a) Training video. (b) New video of the same person as training video. (c) Another person in the template training set. (d) Person not in the training set.

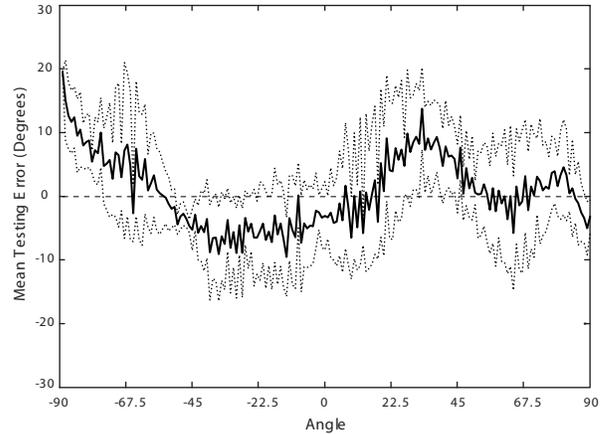


Figure 7: Mean and standard deviation of testing error as a function of angle.

5.1 Tracking head yaw given accurate detection and background removal

Our first set of experiments track yaw from image sequences that are well behaved: the head is correctly segmented, it undergoes minimum roll and pitch, the background is effectively removed and there are no sharp illumination contrasts. The range of free motion is full 180° and the subjects are free to change facial gestures. Figure 6 shows, for four video sequences, the output of the system, sample frames and the error in degrees. Figure 7 plots the mean testing error as a function of angle. The error is symmetric and it increases for input images that are farther away from the template angles.

5.2 Tracking head yaw as assumptions are relaxed

Even though our original assumptions are tight, there are environments where working with them seems plausible, for instance, indoor environments where the backgrounds and the illumination are relatively constant. Nevertheless, we wish to study the effects that relaxing this assumptions will have on our system. In these set of experiments we show results for frontal views only. Other views behave similarly.

To simulate segmentation inaccuracies, we horizontally and vertically shift a window over the image and compute yaw angle. Figure 8 shows the output to samples where our assumptions are relaxed. It is the RBFN estimate without filtering. (a) and (b) plot the output of the RBFN as a function of pixel shift. On the 24 by 24 images of a head, a pixel corresponds roughly to a centimeter. The response of the detectors to horizontal shift degrade symmetrically at

± 4 pixels. On the other hand, the failure for vertical shifting is asymmetric in this example. The response remains grounded until a 20 pixel up-shift because the detectors pick up the dark area still visible as hair. The response to vertical off-center is less brittle for these samples.

To measure the effect of roll and pitch on our system, we recorded a sequence of images with constant yaw at zero and varying pitch (figure 8c) and a second sequence with varying roll (figure 8d). Notice the scale of the vertical axis. Again, as we noted before, the response is more robust to vertical changes. The decay to the varying roll sequence is not symmetric because of misalignment. As the head is rolled, the detectors become more sensitive to misalignment.

To illustrate the effect of sharp illumination changes we capture a video of a frontal view with a controlled horizontally varying spotlight (figure 8e). The response in this sequence is closer to the ground truth when the head is illuminated from the sides.

Finally, to measure the effects of background subtraction error, we gradually add gaussian noise to the image (figure 8f). The signal remains grounded until the image is mostly noise.

6 Conclusions and future work

We have presented a system that estimates head yaw from an image stream and we have demonstrated the stabilizing impact temporal cues from video have on tracking yaw. For some applications, like focus of attention on a densely populated interactive surface, tracking stability is essential. For such applications, direction of gaze needs to be computed and head yaw computation can be used as a course approximation or as a preprocessing step.

We have shown positive laboratory experimental results, where conditions were under control. The first step we need to take from here is implementing a robust system in a real environment, such as an interactive office, where subjects are completely free to move and the lighting and background settings change. For such implementation, we need to collect more training data, both for building the templates and for training the RBFN.

We have not trained with a subject whose skin is darker than her hair. The system is trained to recognize exactly the opposite and would fail at recognizing these or other instances where the face area is darker than the hair area. For example, it would not track images with specular reflections on the hair or long hair covering large areas of the face. This is a bias that can easily be reverted based on the main users' appearances.

Given the orders of magnitude in the difference of the output of our system to facial images, even with noise and off-center, and non-facial images, an interesting future side

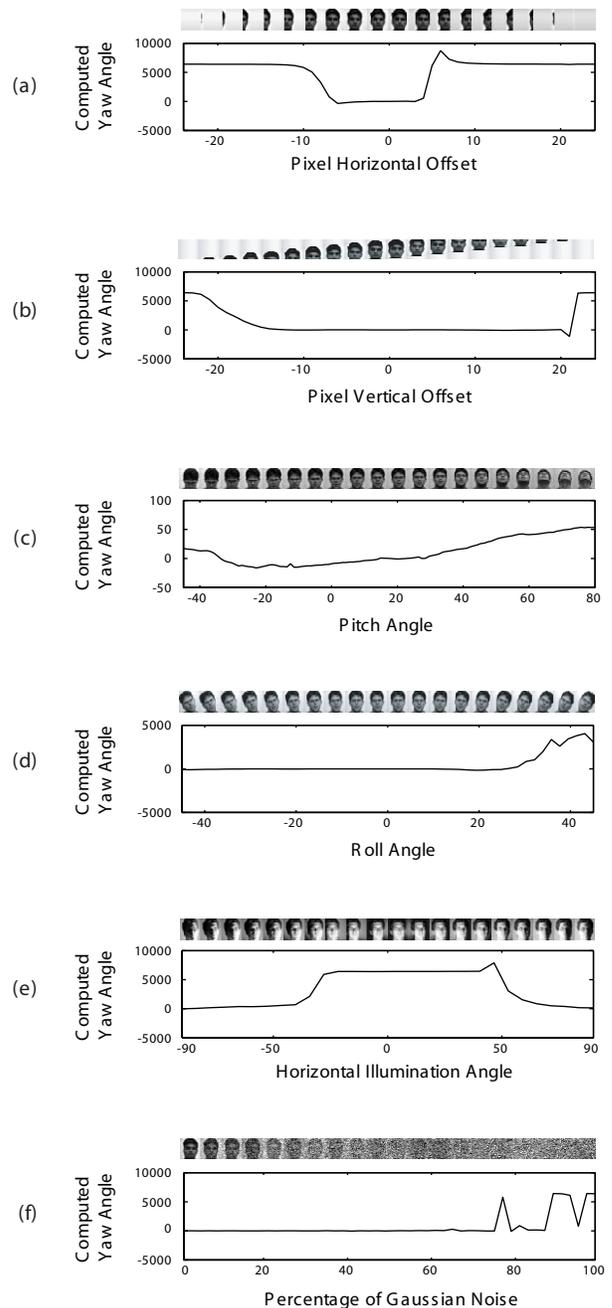


Figure 8: System decays as: (a) images are horizontally off-center, (b) vertically off-center, (c) the angle of pitch varies, (d) the angle of roll varies, (e) sharp illumination contrasts, and (f) noise is added (to simulate segmentation decay). Notice the vertical axis scales. When the system loses the signal its output rests at over 5000 degrees, a meaningless value. (d) is not symmetric because at greater roll angles, misalignment has different destabilizing effects.

effect application for our architecture is multiple view head detection.

References

- [1] S. Basu, I. Essa, and A. Pentland, "Motion regularization for model-based head tracking," *Proceedings of the IEEE International Conference on Pattern Recognition*, 1996.
- [2] R. Brunelli, T. Poggio, "Face Recognition: Features versus Templates," *PAMI*, volume 15(10), pages 1042-1052, 1993.
- [3] I. Essa, T. Darrell, and A. Pentland, "Tracking facial motion," *Proceedings of the IEEE Workshop on Nonrigid and Articulate Motion*, 1994.
- [4] R. Gross, I. Matthews, and S. Baker, "Eigen light-fields and face recognition across pose," *IEEE International Conference on Automatic Face and Gesture Recognition*, 2002.
- [5] T. Horprasert, Y. Yacoob, and L. Davis, "Computing 3-D Head Orientation from a Monocular Image Sequence," *IEEE International Conference on Automatic Face and Gesture Recognition*, 1996.
- [6] Q. Ji, "3D Face pose estimation and tracking from a monocular camera," *Image and Vision Computing*, Vol. 20 (7), pp. 499-511, 2002.
- [7] S. Li, X. Peng, X. Hou, H. Zhang, and Q. Cheng, "Multi-view face pose estimation based on supervised ISA learning," *IEEE International Conference on Automatic Face and Gesture Recognition*, 2002.
- [8] S. Li, Y. ShuiCheng, H. Zhang, and Q. Cheng, "Multi-view face alignment using direct appearance models," *IEEE International Conference on Automatic Face and Gesture Recognition*, 2002.
- [9] Matlab Control System Toolbox v.5.2, The MathWorks <http://www.mathworks.com/access/helpdesk/help/toolbox/control/control.html>
- [10] Matlab Image Processing Toolbox v.4.1, The MathWorks <http://www.mathworks.com/access/helpdesk/help/toolbox/images/images.html>
- [11] Matlab Neural Network Toolbox v.4.0.1, The MathWorks <http://www.mathworks.com/access/helpdesk/help/toolbox/nnet/nnet.html>
- [12] Y. Ming, G. Yingchun, and K. Youan, "Human Face Orientation Estimation Using Symmetry and Feature Points Analysis," *IEEE International Conference on Signal Processing*, 2000.
- [13] E. Ong, S. McKenna, and S. Gong, "Tracking Head Pose for Inferring Intention," *European Workshop on Perception of Human Action*, 1998.
- [14] D. Roth, M. Yang, and N. Ahuja, "A snowbased face detector," *Neural Information Processing*, 2000.
- [15] H. Rowley, S. Baluja, and T. Kanade "Neural network-based face detection," *PAMI*, volume 20, pages 22-38, 1998.
- [16] P. Viola and M. Jones, "Robust Real-time Object Detection," *International Journal of Computer Vision*, 2002.
- [17] Y. Wu and K. Toyama, "Coarse Head-Orientation Estimation with Bootstrap Initialization," *European Conference on Computer Vision*, 2002.
- [18] Z. Zhang, L. Zhu, S. Li and H. Zhang, "Real-time multi-view face detection," *IEEE International Conference on Automatic Face and Gesture Recognition*, 2002.