HIDDEN MARKOV MODEL FOR EYE GAZE PREDICTION IN NETWORKED VIDEO STREAMING

Yunlong Feng[o], Gene Cheung[#], Wai-tian Tan[*], Yusheng Ji[#]

[o] The Graduate University for Advanced Studies, [#] National Institute of Informatics,
[*] Hewlett-Packard Laboratories

ABSTRACT

With the advent of eye gaze tracking technology, eye gaze is increasingly being used as a media interaction trigger in a variety of applications, such as eye typing, video content customization, and network video streaming based on region-of-interest (ROI). The reaction time of a gaze-based networked system, however, is in practice lower-bounded by the round trip time (RTT) of today's networks, which can be large. To improve the efficacy of gaze-based networked systems, in the paper we propose a Hidden Markov Model (HMM)-based gaze prediction strategy to predict future gaze locations to lower end-to-end reaction delay. We first design an HMM with three states corresponding to human's three major types of intrinsic eye movements. HMM parameters are obtained offline on a per-video basis during training phase. During testing phase, a window of noisy gaze observations are collected in real-time as input to a forward algorithm, which computes the most likely HMM state. Given the deduced HMM state, linear prediction is used to predict gaze location RTT seconds into the future.

We demonstrate the applicability of our gaze prediction strategy by focusing on ROI-based bit allocation for network video streaming. To reduce transmission rate of a video stream without degrading viewer's perceived visual quality, we allocate more bits to encode the viewer's current spatial ROI, while devoting fewer bits in other spatial regions. The challenge lies in overcoming the delay between the time a viewer's ROI is detected by gaze tracking, to the time the effected video is encoded, delivered and displayed at the viewer's terminal. To this end, we use our proposed gaze-prediction strategy to predict future eye gaze locations, so that optimized bit allocation can be performed for future frames. Our experiments show that bit rate can be reduced by 21% without noticeable visual quality degradation when end-to-end network delay is as high as 200ms.

Index Terms— Eye-gaze prediction, network streaming

1. INTRODUCTION

Eye gaze tracking—the inference of a viewer's point of visual focus based on camera-captured images of the eyes—has been intensively studied by the computer vision community in the last decade [1], to the level of maturity that it is now a commercially available technology [2]. To unlock the potential of this new tool, many applications now employ eye gaze as a trigger for media interaction. One example is eye typing, where the gaze location on a monitor triggers typing of English alphabets for the physically disabled. Another example is video customization, where the video content is adaptively composed (e.g., ad insertion) according to gaze location.

For networked media systems, the gaze data are collected at a client and sent to a server to effect changes in media content. The reaction time of the gazed-based trigger is lower-bounded by the round trip time (RTT) of the transmission networks. For today's Internet, RTT can reach 200ms, which significantly exceeds the 60ms tolerance threshold [3] for lag between a change in viewer's visual focus and the corresponding content update in *gaze-contingent displays* (GCD) [4]. Predictive strategies are hence necessary for effective application of eye-gaze in a networked environment.

In this paper, we propose a novel gaze prediction strategy to estimate future gaze locations to lower end-to-end reaction delay in gaze-based networked media systems. We first design a Hidden Markov Model (HMM) with three latent states that correspond to human's three major types of intrinsic eye movements: *fixation*, *pursuit* and *saccade* [5]. HMM parameters are obtained offline on a per-video basis using data collected during training phase. During testing phase, a window of noisy gaze observations are collected in real-time for a forward algorithm (FA) to compute the most likely current latent state. Given the deduced HMM state, linear prediction is performed to predict gaze location RTT seconds into the future to reactively effect media content adaptation at server.

We demonstrate the applicability of our gaze prediction strategy through a network video streaming application that performs bit allocation based on Region-Of-Interest (ROI). In face of limited network transmission bandwidth, the conventional end-to-end streaming approach is to throttle sending rate, so that limited network bandwidth can be properly shared among competing users. Reduction of sending rate, however, causes a proportional degradation in video quality due to signal quantization, often resulting in unacceptable visual experience.

One can address this bandwidth-constrained problem by exploiting unique characteristics of the human perceptual system [6, 4, 3]. In particular, it has been shown [6, 7] that viewer's ability to perceive details away from the current focused ROI decreases drastically as the spatial distance from ROI increases. Thus, a smart bit allocation scheme [8, 9] can allocate more bits to ROI to minimize quantization noise and fewer bits elsewhere, so that the *perceived* quality of the video remains the same while encoded bit-rate can be decreased. The key challenge, however, is to overcome the unavoidable delay from the time a ROI is estimated, to the time the corresponding effected change in video bit allocation is executed, transmitted and rendered on the viewer's terminal. To overcome RTT delay, we use our proposed gaze-prediction strategy to predict future gaze locations, so that optimal bit allocation can be performed for future frames. Our experiments, using our developed real-time video coding and streaming system integrated with a web camera and a software gaze tracker [10], show that using our gaze-prediction strategy, transmission rate can be reduced by up to 21% without loss of perceived video quality for RTT as high as 200ms.

The outline of the paper is as follows. We first discuss related work in Section 2. We then discuss our proposed HMM for eye-gaze data in Section 3. For a given estimated HMM state, we discuss how linear prediction is used to predict future gaze location RTT sec-

978-1-61284-350-6/11/$26.00

onds into the future in Section 4. Having obtained a gaze prediction, the corresponding bit allocation scheme is discussed in Section 5. Experimentation and conclusions are discussed in Section 6 and 7, respectively.

2. RELATED WORK

Optimized bit allocation schemes for video with given ROI have been studied recently [8, 9]. The hard problem remains *how* ROI can be accurately estimated in the first place. In one approach, the ROI is determined *a priori* based on *saliency maps* [11] obtained solely based on content analysis, typically using low-level video features such as spatial contrasts in luminance, temporal changes in motion, appearances of machine-recognized human faces, etc. It has been shown [12, 13], however, that prior knowledge and context play important roles in affecting viewer's attention, and modeling these information when calculating saliency maps is a daunting task. In contrast, while we use video content to train HMM parameters during training phase, in operational phase we determine ROI based on real-time eye gaze tracking. The key challenge, which is the focus of this paper, is to reduce the effect of time lag due to server-client RTT delay in a networked video streaming setting.

Eye gaze prediction based on real-time collected gaze data has been recently studied in the literature [14]. In [14] a detailed motion model is presented to predict eye movements based on the mechanics of the human eye using a large number of parameters. Our gaze-prediction strategy differs from [14] in two major respects. First, we approach the gaze prediction problem from a statistical learning perspective, where our three-state HMM is simple and maps intuitively to human's three intrinsic types of eye movements, leading to low cost of implementation. Second, unlike [14] which predicts gaze movements in a content-independent manner, the few HMM parameters in our model are trained on a per-video basis, leading to more content-specific customization for better prediction performance.

3. HIDDEN MARKOV MODEL FOR GAZE-TRACKING

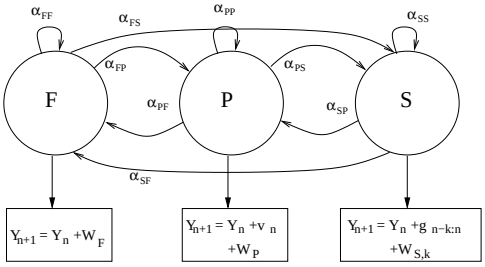


Fig. 1. Proposed hidden Markov model for eye gaze during video observation. Circles denote latent states of F (fixation), P (pursuit) and S (saccade). Boxes denote observations.

In this section, we discuss how we model eye gaze of a video viewer using a *hidden Markov model* (HMM) [15]. An HMM models transitions of sequential state X_n's, $n \in \mathcal{Z}^+$, in discrete time, where X_n is the *state variable* at time n. Each X_n can take on one of three possible *latent states*[1]. State F (*fixation*) models the case when eye gaze is fixated at a stationary object. State P (*pursuit*) models the case where gaze follows motion of a moving object. State S (*saccade*) models the rapid transition from one fixation point to another. As discussed in [5], these are the three major types of eye movements for the human eye.

[1]Since states $\{\mathsf{F}, \mathsf{P}, \mathsf{S}\}$ cannot be observed directly, they are commonly called *latent states* in the literature.

An HMM is Markovian in that the determination of state variable X_{n+1} at time $n+1$ depends solely on the value of X_n of previous time n. In particular, given $X_n = i$, the probability of $X_{n+1} = j$ is represented by *state transition probability* $\alpha_{i,j}$ of switching from state i to j. The model is hidden since the state variables X_n's are not directly observable; only observations[2] Y_n's are observed, where each Y_n is generated by a random process dependent on current latent state $X_n = i$. The most likely value of state variable X_n given observations $Y_1, \ldots, Y_n$ can be calculated using the known Forward Algorithm (FA) [15] (to be discussed). In our gaze tracking scenario, that means determining the most likely eye movement type of a viewer among $\{\mathsf{F}, \mathsf{P}, \mathsf{S}\}$ given captured eye gaze data. We describe the three random processes, corresponding to latent states, F, P and S, that generate observations next.

3.1. Fixation: observing a stationary object

Fig. 2. Eye gaze data on frame 220 of MPEG test sequence `kids`. Eye gaze data is marked by a white 5×5 square.

For the simplest of three latent states, F (fixation), we model the random process that emits observations Y_n's as follows. Given client resides in state $X_{n+1} = \mathsf{F}$ at time $n+1$, emitted observation Y_{n+1} is the sum of previous observation Y_n plus a random variable W_F:

$$Y_{n+1} = Y_n + W_F \tag{1}$$

where W_F is a zero-mean Gaussian random variable with variance σ_F^2. We denote the probability density function of W_F by $f_{\sigma_F^2}(w)$. The probability of observing Y_{n+1} given current state is F is $P(Y_{n+1}|Y_n, X_{n+1} = \mathsf{F}) = f_{\sigma_F^2}(Y_{n+1} - Y_n)$.

Noise modeling is important even for the fixation state, since instability of human vision and inaccuracy of eye tracking algorithms mean non-negligible noise is present in the eye gaze data. As an example, see Fig. 2 where a viewer is looking at the red ball, but the gaze tracker returns a gaze data point slightly away from the ball.

3.2. Pursuit: following a moving object

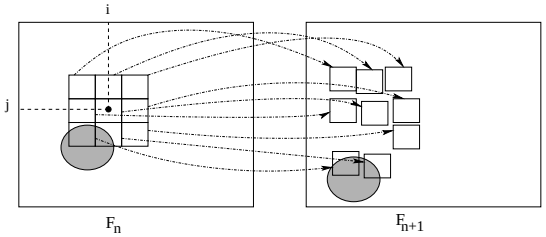


Fig. 3. Calculation of forward motion vector candidates in next frame F_{n+1} given eye gaze data Y_n at location (i, j) in frame F_n.

[2]For simplicity, we model observations of x- and y-components of gaze locations separately. We discuss here observations for one component only.

If the value of state variable X_{n+1} is P (pursuit) at time $n+1$, we model the emitted observation Y_{n+1} as the sum of previous observation Y_n plus a *pixel velocity vector* v_n plus random noise W_P:

$$Y_{n+1} = Y_n + v_n + W_P \tag{2}$$

where v_n is the *velocity vector* of the viewed pixel (the fixation point of the viewer) from frame F_n of time n to frame F_{n+1} of time $n+1$, and W_P is another zero-mean Gaussian random variable with its own unique variance σ_P^2. If the fixation point of the viewer in frame F_n is know precisely, v_n can be estimated easily: first identify the macroblock that contains the viewed pixel at time n, then find the best matched macroblock in frame F_{n+1} in pixel values and calculate the corresponding motion vector. The probability of observing Y_{n+1} given current state is P is hence $P(Y_{n+1}|Y_n, X_{n+1} = \mathsf{P}) = f_{\sigma_P^2}(Y_{n+1} - Y_n - v_n)$.

The difficulty here is that the true gaze point in frame F_n is not known precisely due to noise in observation. That means that if a viewer is actually following a moving object but gaze point is not on the object due to noise (as shown in Fig. 2), then the calculated motion vector will be erroneous.

To circumvent this problem, we perform multi-block search as shown in Fig. 3. For given observed gaze location Y_n, we first identify a *neighborhood* of macroblocks around Y_n. For each macroblock in the neighborhood, we search for a best matched block in the next frame F_{n+1} and calculate the corresponding motion vector v_n. Among all the calculated vectors v_n's, we identify the one that gives the largest conditional probability for state P:

$$P(Y_{n+1}|Y_n, X_{n+1} = \mathsf{P}) = \max_{v_n \in \mathcal{V}_n} f_{\sigma_P^2}(Y_{n+1} - Y_n - v_n) \tag{3}$$

where $\mathcal{V}_n$ is the set of calculated motion vectors.

Lastly, we see that (2) for state P and (1) for state F differ only by velocity vector v_n. In fact, if a viewer is following a very slow moving object, then $v_n \approx 0$ and we cannot distinguish between the two latent states. To disambiguate latent states P and F, we compare the discovered velocity vector v_n in (3) to a threshold τ_P. If $|v_n| < \tau_P$, we conclude that latent state P is not possible for X_n, and the computed probability for state P, $P(X_n = \mathsf{P})$, (using FA to be discussed in Section 3.4), is added to the probability for state F, $P(X_n = \mathsf{F})$.

3.3. Saccade: switching fixation points

If the viewer is in state $X_{n+1} = \mathsf{S}$ (saccade) at time $n+1$, the gaze of the viewer is switching from one fixation point to another. The transition process usually lasts a short duration (20 to 200ms), and the movement is fast [5]—saccade is said to be the fastest movement by the human body [16]. Fortunately, according to Listing's Law [16], movement of the eye during one saccade is restricted to rotation on a single axis; i.e., gaze moves in a single direction during saccade. Thus, if we are able to establish a *gaze vector* $g_{n-k:n}$ during saccade using previous observations Y_n's, then new observation Y_{n+1} is previous observation Y_n plus $g_{n-k:n}$ plus a noise term $W_{S,k}$. However, if the gaze vector during saccade cannot be established—gaze vector can only be estimated two samples *after* saccade has started—then gaze location can move in any direction at the start of saccade. We hence model the movement simply as a zero-mean Gaussian variable G with a fairly large variance σ_G^2.

Mathematically, we write observation Y_{n+1} given viewer resides in state $X_{n+1} = \mathsf{S}$ as follows:

$$Y_{n+1} = \begin{cases} Y_n + G & \text{if } X_n \neq \mathsf{S} \\ Y_n + g_{n-k:n} + W_{S,k} & \text{o.w.} \end{cases} \tag{4}$$

where $g_{n-k:n}$ is the mean eye gaze vector computed using most recent $k \geq 1$ observations $Y_{n-k+1}, \ldots, Y_n$ of state S *plus* one preceeding observation Y_{n-k} of state F or P. $W_{S,k}$ is a zero-mean Gaussian variable, whose variance $\sigma_{S,k}^2$ depends on the number of observations, $k+1$, used to compute $g_{n-k:n}$. The idea is to capture the notion that the more previous observations Y_n's we use to estimate gaze vector $g_{n-k:n}$, the smaller the corresponding variance $\sigma_{S,k}^2$ of Gaussian noise $W_{S,k}$ should be. $g_{n-k:n}$ can be computed using samples $(n-k, Y_{n-k}), \ldots, (n, Y_n)$ via linear regression (to be discussed in Section 4 in the context of linear prediction).

We can now write the probability $P(Y_{n+1}|Y_n, X_{n+1} = \mathsf{S})$ of observing Y_{n+1} given Y_n and state X_{n+1} is S as follows:

$$= \begin{cases} f_{\sigma_G^2}(Y_{n+1} - Y_n) & \text{if } X_n \neq \mathsf{S} \\ f_{\sigma_{S,k}^2}(Y_{n+1} - Y_n - g_{n-k:n}) & \text{o.w.} \end{cases} \tag{5}$$

As done previously, to disambiguate state S from F and P when $X_n = \mathsf{S}$, we do the following: if $|g_{n-k:n}| < \tau_P$, then $P(X_n = \mathsf{S})$ is added to $P(X_n = \mathsf{F})$. If $|g_{n-k:n}| > \tau_P$ but $|g_{n-k:n} - v_n| < \tau_S$, then $P(X_n = \mathsf{S})$ is added to $P(X_n = \mathsf{P})$.

3.4. Finding most likely latent states

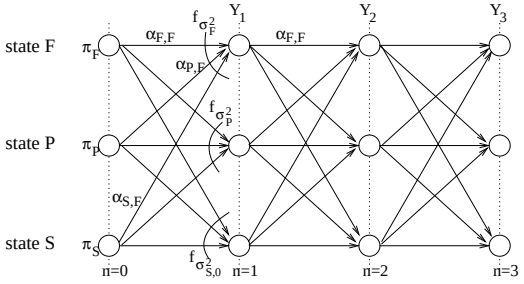


Fig. 4. Trellis corresponding to a 3-state HMM. A Forward Algorithm can find the most likely state X_n given observations $Y_1, \ldots, Y_n$'s.

To find latent state probability $P(X_n = j)$ given a window of observations $Y_1, \ldots, Y_n$, we use the well-known *forward algorithm* (FA) [15]. We first assume the initial probability π_i of each state $X_0 = i$ is known. We can then compute probability $P(X_n = j)$ recursively as follows:

$$\begin{aligned} P(X_n = j) &= \sum_i P(X_{n-1} = i)\, \alpha_{i,j}\, P(Y_n|Y_{n-1}, X_n = j) \\ P(X_0 = i) &= \pi_i \end{aligned} \tag{6}$$

The most likely latent state for state variable X_n is the one with the largest of three computed probabilities $P(X_n = j)$'s.

One interpretation of (6) is that we are building the state probabilities $P(X_n = j)$'s from $n = 0$ *forward* in a trellis, as shown in Fig. 4. As such, the computation can also be computed iteratively; i.e., the previously computed probabilities $P(X_n = j)$'s used to determine the most likely state X_n can be used to compute $P(X_{n+1} = j)$'s when new observation Y_{n+1} becomes available. The is particularly useful when the number of observations Y_n's becomes large and recursive definition (6) becomes computationally expensive.

4. LINEAR PREDICTION

We have just discussed how we find the most likely latent state X_n in HMM given observations $Y_1, \ldots, Y_n$. In this section, we discuss

how we predict a future gaze location $\bar{Y}_{n+RTT}$. Smart bit allocation can then be performed to assign finer QP for ROI centered on predicted location $\bar{Y}_{n+RTT}$, and coarser QP for other spatial regions in a coded frame (to be discussed in Section 5).

Note, however, that we perform prediction only if the most likely state is F or P. Because the duration in which a viewer stays in saccade state S is typically very short [16] and will soon stop at an unpredictable fixation point, we take the conservative approach and perform no prediction in state S. Further, even if the most likely state is F or P, we perform prediction only if the most likely state has probability $P(X_n = j)$ exceeding a threshold τ_C. In other words, we will predict gaze location only if we are confident enough in our state estimation.

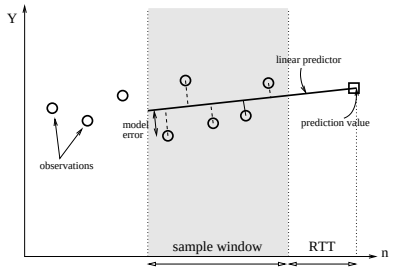


Fig. 5. Linear prediction using a window of ω observations. Circles denote observation point, square denotes predicted value.

To estimate Y_{n+RTT}, we use a window of ω observations $Y_{n-\omega+1}, \ldots, Y_n$ for *linear regression* [15]. In other words, using sample points $(n-\omega+1, Y_{n-\omega+1}), \ldots, (n, Y_n)$, we seek a linear function $Y(t) = \hat{\phi} + \hat{m}t$, so that the sum of errors (Euclidean distance) between sample points and the linear function is minimized. Fig. 5 shows an example where a best-fit linear function is constructed using five sample points. Statistically optimal $\hat{m}$ and $\hat{\phi}$ can be derived easily given samples (x, y)'s:

$$\begin{aligned} \hat{m} &= \frac{\overline{xy} - \bar{x}\bar{y}}{\overline{x^2} - \bar{x}^2} \\ \hat{\phi} &= \bar{y} - \hat{m}\bar{x} \end{aligned} \tag{7}$$

where $\bar{x}$ means the average of sample data x's.

Having computed linear regression parameters, $\hat{m}$ and $\hat{\phi}$, prediction $\bar{Y}_{n+RTT}$ is simply $\hat{\phi} + \hat{m}(n + RTT)$. Using this prediction, for state P, it means we assume the motion in the window of ω sample points remains constant. For state F, given the observed object is stationary, we force $\hat{m}$ to be zero; in other words, we simply compute the average of all sample points Y_n's as our estimate.

5. ROI BIT ALLOCATION FOR VIDEO ENCODING

In this section, we discuss a bit-rate allocation strategy as an application of our proposed HMM based eye-gaze prediction method. Conceptually, human ability to appreciate pixel fidelity decreases continuously away from the center of focus. It is therefore wasteful to encode visual information away from focus with high fidelity. In the previous sections, we already described how to predict the location of future eye gaze $\bar{Y}_{n+RTT}$. One approach to exploit this knowledge of user's visual focus is to continuously adapt each macroblock's quantization parameter (QP) according to a visual model [9]. Nevertheless, in this paper, we adopt a simpler approach in which a rectangular ROI is determined, and one QP is assigned to the ROI, while a coarser (higher) QP is assigned to spatial regions outside the ROI. This is due to its lower complexity, and the lower sensitivity to errors in focus determination. Furthermore, regions far away from focus will be not aggressively quantized, which results in little additional rate reduction, but displeasing quality during saccade when the viewer can suddenly change his focus.

5.1. Bit Allocation of ROI

As discussed in [7], the fall-off in human ability to appreciate pixel fidelity can be approximately modeled by the contrast sensitivity (CS) of humans, which is the reciprocal of the contrast threshold (CT) given by:

$$\begin{aligned} CT(f, e) &= CT_0 \exp\left(\alpha f \frac{e + e_2}{e_2}\right) \\ CS(f, e) &= 1/CT(f, e) \end{aligned}$$

where f is spatial frequency, e is the retinal eccentricity or the angle relative to the point of focus, and CT_0, e_2 and α are constants empirically determined to be 1/64, 2.3, and 0.106, respectively.

As done in [9], we determine the cutoff frequency, f_c, by setting CT to one:

$$f_c = \frac{e_2 \log \frac{1}{CT_0}}{\alpha(e_{max} + e_2)} \tag{8}$$

where e_{max} is the maximum eccentricity in the video frame, which is the largest angle the screen portends relative to the focus point. The average contrast threshold evaluated at spatial frequency f_c inside and outside an ROI are then computed, and the corresponding QP are chosen so that:

$$\frac{QP_{ROI}}{QP_{\overline{ROI}}} = \frac{CT_{ROI}}{CT_{\overline{ROI}}} \tag{9}$$

5.2. Determining ROI for State F

Given a video frame with width w and height h, we choose a ROI of size $w/2 \times h/2$ centered at the estimated gaze location. This allows at least 75% of the frame to be coded at a lower QP, while allowing a substantial region near the focus point to be at high quality. For experiments in Section 6 with a field of view of 80 degrees, this corresponds to a ROI with field of view of 45 degrees, which is large enough to comfortably capture regions of high visual sensitivity.

5.3. Determining ROI for State P

Due to the higher uncertainty in gaze position in State P compared to State F, a larger ROI should be employed. Let σ be the mean absolute difference between the linear predictor and ω observations $Y_{n-\omega+1}, \ldots, Y_n$ in the sample window. We can define a *dilation factor* $\rho_w = 1 + (2\sigma)/w$ for the horizontal axis, so that the width of the ROI is rescaled to $\rho_w * w/2$. The idea is to increase the size of ROI when the linear predictor is a poor fit to the ω data points. We can define a similar dilation factor ρ_h to rescale the height of ROI to $\rho_h * h/2$.

6. EXPERIMENTATION

To demonstrate the merit of our proposed HMM-based gaze prediction strategy, we conducted extensive experiments. We first describe the setup of our experiments and parameters selected for our model in Section 6.1. In part one of the experiment, described in Section 6.2, we examine the accuracy of our HMM state estimation, and the tradeoff between false positive (predicting HMM state to be F or P when ground truth is S) and false negative (predicting HMM state to be S when ground truth is F or P). In part two of the experiment, described in Section 6.3, we examine the accuracy of our

Table 1. State transition and steady state probabilities for `kids`

	F	P	S	π
F	0.965	0.019	0.016	0.494
P	0.017	0.965	0.017	0.365
S	0.015	0.029	0.956	0.141

Table 2. State transition and steady state probabilities for `table`

	F	P	S	π
F	0.949	0.012	0.039	0.422
P	0.046	0.927	0.027	0.292
S	0.028	0.056	0.916	0.286

HMM-based linear prediction. In part three of the experiment, described in Section 6.4, we examine the achievable bitrate saving for our proposed bit allocation scheme. We also show that our bit allocation scheme suffers no loss in perceived visual quality, through subjective user tests on our in-house developed real-time system.

6.1. Experimental Setup and HMM Training

Our gaze-based networked streaming system employs the freely available real-time gaze-tracking software `opengazer` [10], which is calibrated for sampling gaze location at 30 samples per second using an off-the-shelf web camera. The monitor used for gaze tracking and video experiments measured 22 inches diagonally ($473.7mm \times 296.1mm$). The distance between a user's head and the center of monitor screen is about $280mm$, resulting in a viewing angle of about 40 degrees to the edge of the screen.

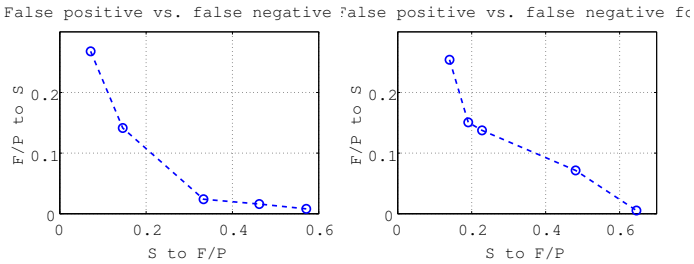
We used two 300-frame standard MPEG video test sequences, `kids` and `table`, at CIF resolution (354×288) for our experiments. For video compression, we use a fast implementation of H.263 [17] for real-time encoding. Each video was displayed in full-screen mode at 30 fps, the same sampling rate of `opengazer` for one-to-one correspondence between gaze samples and video frames.

To obtain ground truth gaze data to train our proposed HMM model, a trained user performed multiple viewings of a test sequence, each time continuously record his intention of fixation, pursuit or saccade by pressing keys on a keyboard. Using this "ground truth" data, we calculated the state transition probabilities $\alpha_{i,j}$'s from state i to j in the HMM and the steady state probabilities π_i's. State transition and steady state probabilities for `kids` and `table` are shown in Table 1 and 2, respectively.

We see that for both sequences, the probabilities $\alpha_{i,i}$'s of returning to the same states i's are very high. We see also from the steady state probabilities π_i's that the likelihood of being state F is much higher than the other two latent states. In contrast, the probability of being in state S is only 0.14 to 0.28. This is reasonable; in a typical video, a viewer tends to spend the majority of her time looking at objects of interest rather than switching fixation points.

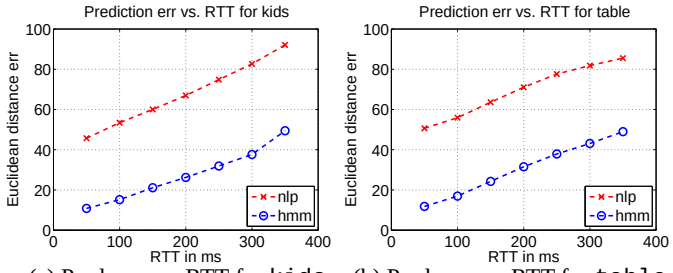
6.2. Results for HMM State Estimation

We now evaluate the accuracy of HMM state estimation using forward algorithm (FA), as discussed in Section 3.4. We denote an occurrence as *false positive* when FA estimates HMM state to be F or P but the ground truth state is S. In other words, false positive is when we wrongly deduced an opportunity to save coding bits by assigning coarser quantization parameter outside ROI, but the algorithm calls for high quality encoding for entire frame. In contrast, we denote an occurrence as *false negative* when FA estimates HMM state to be S but ground truth state is either F or P. This is the case where we miss a bit-saving opportunity.



(a) Err tradeoff for `kids` (b) Err tradeoff for `table`

Fig. 6. Tradeoff in false positive and false negative probabilities by adjusting threshold τ_C, for `kids` and `table`, respectively.



(a) Pred. err vs. RTT for `kids` (b) Pred. err vs. RTT for `table`

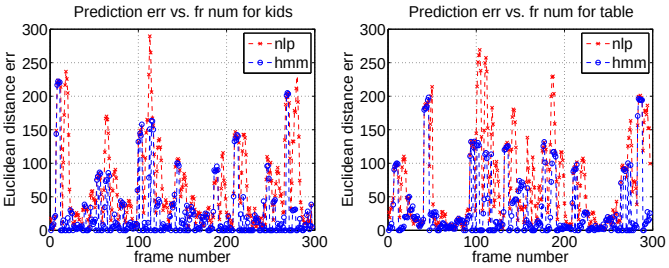
Fig. 7. Prediction Error in Euclidean distance as function of RTT for different prediction schemes, for `kids` and `table`, respectively.

As discussed in Section 4, a threshold τ_C can be adjusted according to our confidence in the estimated F or P state, resulting in a tradeoff between false positive and false negative probabilities. In Fig. 6, we see the said tradeoff in the two probabilities in our HMM state estimation for sequences `kids` and `table`, respectively. We see that though in general it is difficult to achieve very small false positive and false negative probabilities at the same time, it is possible to have reasonably small (≤ 0.15) values for both. This shows that FA can provide reasonable state estimates for our proposed HMM. To be shown later, this level of estimation accuracy is sufficient for our intended networked streaming application.

6.3. Results for HMM-based Linear Prediction

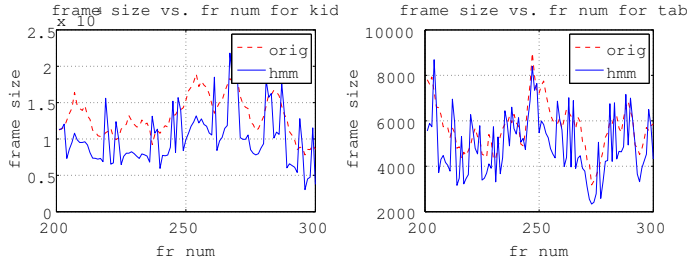
Given estimated HMM states, we next examine the accuracy of our proposed HMM-based linear prediction (`HMM`), as discussed in Section 4. We compare our prediction scheme to a naïve linear prediction scheme (`nlp`), where the last two gaze data points are used to construct a straight line, extrapolation of which to RTT seconds later yields a gaze location estimate. In Fig. 7, we see the performance of both schemes, in terms of Euclidean distance between the estimated gaze locations and true gaze locations, as function of RTT for both sequences `kids` and `table`. We see that as RTT increased, the estimation error increased for both `HMM` and `nlp`. However, `HMM` achieved much smaller errors than `nlp`. This is due to two reasons. First, to contain errors, `HMM` construct a linear prediction only when it is sufficiently confident it is in state F or P, while `nlp` makes an estimate for all data points.

Second, `HMM` uses different methods for prediction depending on the estimated state (F or P) using a window of data points. This is evident in Fig. 8, where prediction error was plotted against frame



(a) Pred. err vs. fr #. for `kids` (b) Pred. err vs. fr # for `table`

Fig. 8. Prediction Error in Euclidean distance as function of frame number for different prediction schemes, for `kids` and `table`, respectively, when RTT=200ms.



(a) Frame size vs. fr #. for `kids` (b) Frame size vs. fr # for `table`

Fig. 9. Frame size as function of frame number for different bit allocation schemes, for `kids` and `table`, respectively, when RTT=200ms.

number for $RTT = 200ms$. At frame numbers where `HMM` made prediction, the magnitude of resulting error was in general smaller than `nlp`.

6.4. Results for HMM-based Bit Allocation

We next show the achievable bit saving for our gazed-based bit allocation for networked video streaming. We use $QP = 10$ for a desired reference quality. For our gaze-based scheme (`hmm`) described in Section 5, the average QP outside the ROI is 15, as given by (9). For simplicity, we use dilation factors $\rho_w = \rho_h = 1$ for ROI construction for state `P`. An original scheme (`orig`) assigns $QP = 10$ for all blocks in a frame. The compressed frame size for the two schemes are given in Fig. 9 for both test sequences. We see that in frames where the estimated state was `F` or `P`, fewer bits were allocated to non-ROI regions, resulting in bitrate saving. In particular, we found that `hmm` achieved 21% and 17% bit saving compared to `orig` for sequence `kids` and `table`, respectively.

Of course, the bit saving must be achieved without the loss of perceptual quality. To verify this, we developed a real-time video coding / streaming system, with artificial delay inserted between encoder and decoder to emulate RTT=200ms. We performed user subjective test as follows. For each viewer, three runs of the same video were presented. One run was full quality video encoded at QP=10 for all blocks (`orig`). One run was our proposed HMM-based bit allocation (`hmm`). One run was bit allocation based on naïve linear prediction (`nlp`) based on the last two gaze data points. The order of the three runs was randomized for each viewer. A simple question was asked after viewing if one or more of the video suffered poor quality. Of the three viewers, two viewers reported no difference, while one identified `nlp` as having slightly worse quality. Though simple, this evaluation provided evidence that `hmm` was able to save bits without suffering perceived visual quality.

7. CONCLUSION

To improve the efficacy of gaze-based networked systems, in this paper, we proposed a hidden Markov model (HMM)-based gaze prediction strategy to predict future gaze locations round-trip-time (RTT) seconds into the future. The three HMM states correspond to human's three major types of intrinsic eye movements. The most likely HMM state is estimated via the forward algorithm (FA) using a window of observed gaze data. Given an estimated state, linear prediction is used to predict future gaze location. To validate our gaze prediction strategy, we apply our model to the bit allocation problem for network video streaming based on region of interest (ROI). Experiments show that bit rate can be reduced by 21% without noticeable visual quality degradation for RTT as high as 200ms.

8. REFERENCES

[1] Y. Sugano, Y. Matsushita, Y. Sato, and H. Koike, "An incremental learning method for unconstrained gaze estimation," in *European Conference on Computer Vision (ECCV2008)*, October 2008, pp. 656–667.

[2] LC Technologies, Inc., "Eyegaze Systems," http://www.eyegaze.com.

[3] L. Loschky and G. Wolverton, "How late can you update gaze-contingent multiresolution displays without detection?," in *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMCCAP)*, December 2007, vol. 3, no.7.

[4] A. Duchowski and A. Coltekin, "Foveated gaze-contingent displays for peripheral LOD management, 3D visualization, and stereo imaging," in *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMCCAP)*, December 2007, vol. 3, no.4.

[5] A. Duchowski, *Eye Tracking Methodology: Theory and Practice*, Springer, 2007.

[6] N. Jayant, J. Johnston, and R. Safranek, "Signal compression based on models of human perception," in *Proceedings of the IEEE*, October 1993, vol. 81, no.10, pp. 1385–1422.

[7] W. Geisler and J. Perry, "A real-time foveated multiresolution system for low-bandwidth video communication," in *SPIE Proceedings*, vol 3299, July 1998.

[8] Y. Liu, Z. G. Li, and Y. C. Soh, "Region-of-interest based resource allocation for conversational video communication of H.264/AVC," in *IEEE Transactions on Circuits and Systems for Video Technology*, January 2008, vol. 18, no.1, pp. 134–139.

[9] Z. Chen and C. Guillemot, "Perceptually-friendly H.264/AVC video coding," in *IEEE International Conference on Image Processing*, Cairo, Egypt, November 2009.

[10] "Opengazer: open-source gaze tracker for ordinary webcams," http://www.inference.phy.cam.ac.uk/opengazer/.

[11] O. Le Meur and P. Le Callet, "What we see is most likely to be what matters: Visual attention and applications," in *IEEE International Conference on Image Processing*, Cairo, Egypt, November 2009.

[12] N. Bruce and P. Kornprobst, "On the role of context in probabilistic models of visual saliency," in *IEEE International Conference on Image Processing*, Cairo, Egypt, November 2009.

[13] S. Davies, D. Agrafiotis, C. Canagarajah, and D. Bull, "A gaze prediction technique for open signed video content using a track before detect algorithm," in *IEEE International Conference on Image Processing*, San Diego, CA, October 2008.

[14] O. V. Komogortsev and J. Khan, "Eye movement prediction by oculomotor plant Kalman filter with brainstem control," in *Journal of Control Theory and Applications*, January 2009, vol. 7, no.1.

[15] C. Bishop, *Pattern Recognition and Machine Learning*, Springer, 2006.

[16] "Saccade," http://en.wikipedia.org/wiki/Saccade.

[17] ITU-T Recommendation H.263, *Video Coding for Low Bitrate Communication*, February 1998.