Human Interactive System for Networked Video Streaming

YUNLONG FENG

A dissertation submitted to the Department of Informatics
School of Multidisciplinary Sciences
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

at
The Graduate University for Advanced Studies (SOKENDAI)
September 2014

Abstract

In this thesis, we discussed the application and development using human interactive technology and visual saliency map analysis for the networked video streaming. As we know, for current video streaming systems, there are more and more devices to record videos, more and more powerful servers to process encoding, larger and larger databases to store them, and also faster and faster network speed to transmit video packages to viewers. However, viewers still feel discontented on passively transmitted videos, for preferring a real time interactive system. In this thesis, we developed a Human Interactive System to make network video streaming more efficient, not only to benefit the network traffic, but also to adapt video content based on viewers' interests and real-time requirement.

For interactive media applications, eye gaze is now used as a content adaptation trigger, such as customized advertisement in video, and bit allocation in streaming video based on region-of-interest (ROI). The reaction time of a gaze-based networked system, however, is lower-bounded by the network round trip time (RTT). Furthermore, only low-sampling-rate gaze data is available when commonly available webcam is employed for gaze tracking. To realize responsive adaptation of media content even under non-negligible RTT and using common low-cost webcams, we propose a Hidden Markov Model (HMM) based gaze-prediction system that utilizes the visual saliency of the content being viewed. Due to the strong prior of likely gaze locations offered by saliency information, accurate runtime gaze prediction is possible even under large RTT and using common webcam. And region-of-interest (ROI) bit allocation is real-time performed based on predicted future gaze location to adapt the video content for reducing bit size and perceived visual quality.

By the latter half of the thesis, a novel method using saliency map for detecting video busyness, which is called visual attention deviation (VAD) is improved, to develop the gaze prediction system. We all know that analyzing human perception is time-consuming, for subjective evaluation is necessary. The experiment always cost much time and need a lot of subjectors. However, saliency map analysis is able to find out the most salient region by low-level figures using given picture or frames. Based on the existed analysis methods, we proposed our metric VAD to detect video busyness by analyzing the saliency regions along the whole timeline with the presented HMM in first half work. Through experiments, we show that VAD was able to detect the video busyness by analyzing the saliency objects transition probability, using trained HMM. And our comparison results show that it's much sensitive than other metrics, and the most important is that VAD result is matching subjective evaluation, which means it's reflecting human perception while the video is play-backed.

Acknowledgments

I am indebted to many people who both directly and indirectly contributed to this thesis. Most of all, I'm grateful for the help and friendship of Zhi LIU. Zhi is my classmate and best friend on all matters for the past nine years, since undergraduate school. I'm also grateful for my collaboration with all lab-mates, who performed many of the subjective experiments described in Chapter 2 and Chapter 3. Finally, I would like to thank Gene CHEUNG, who straddled the line between collaborator and advisor. Gene taught me an attention to detail and a penchant for the LINUX command line and programming.

Most importantly, I am indebted to Yusheng JI, my advisor, whose guidance and support was unwavering these past five years. Prof. Ji is always positive in the face of my many failures along the way. She also gave me the freedom to pursue projects of my own choosing, which contributed greatly to my academic independence, if not the selection of wise projects.

I am much obliged to my thesis committee members: my advisor Professor Yusheng JI, Professor Gene CHEUNG, Professor Shigeki YAMADA, Professor Noboru SONEHARA, Professor Kensuke FUKUDA and Professor Keita TAKAHASHI. My committee was always flexible in scheduling and judicious in their application of both carrots and sticks.

Contents

List of Figures

List of Tables

Chapter 1

Background

1.1 Introduction

With the development of technology, people are able to access millions of videos over network. The conventional networked video streaming system is only able to transmit the required video packages to viewers, which is discontented due to unable responding to viewers' requirement or visual interest in real-time. Fortunately, in the last decade, eye gaze tracking—the inference of a viewer's point of visual focus based on camera-captured images of the eye(s)—has been intensively studied by the computer vision community [43, 47], to the level of maturity that it is now a commercially available technology [33, 51]. To unlock the potential of this new tool, many applications now employ eye gaze as a content adaptation trigger for media interaction. One example is large display customization [46], where the visual content rendered is adaptively composed (e.g., insert customized advertisements) according to tracked past and current gaze locations. Another example is immersive gaming [44], where different animated non-player characters (NPC) react differently depending on which NPC the viewer is currently looking at and showing facial expressions.

For networked media systems, gaze data are collected at a client in real-time and sent to a server to effect changes in media content. The reaction time of the gazed-based trigger, however, is lower-bounded by the round trip time (RTT) of the transmission networks. For today's Internet, RTT can be as large as 200ms, which significantly exceeds the 60ms threshold [36] for tolerable lag between a change in viewer's visual focus and the corresponding content update in gaze-contingent displays (GCD) [15]. This large RTT delay severely limits the efficacy of gaze-based networked media systems. Hence, predictive strategies are necessary for effective application of eye gaze to networked interactive media systems.

In first part of this thesis, we propose a low-cost gaze prediction system using proposed Hidden Markov Model (HMM) to model viewer's gaze behavior and Kalman Filter (KF) to predict viewer's gaze location in the future (RTT seconds from the present), so that the server can adapt media content using the predicted gaze locations instead of the most recently tracked gaze locations, reducing end-to-end reaction delay. The key idea is to establish correlation between tracked eye-gaze movements and the current video content being watched, so that future gaze locations can be predicted with the help of content analysis of video that is about to be displayed. Such analysis can be performed offline computation-efficiently. Specifically, we first design an HMM with two latent states that correspond to two of human's intrinsic gaze behavioral movements: tracking and saccade [14]. Tracking means a viewer is following the movement of an identifiable object in video. Saccade means a viewer is shifting his visual attention from one object of interest to another. Thus, if a viewer following an object in tracking state, then his future gaze location will likely be correlated with the future position of the object.

HMM parameters (most importantly, state transition probabilities) are derived offline at server on a per-video basis via analysis of the video's visual saliency maps [24, 26, 39]. In bottom-up visual saliency models, by computing weighted combinations of detected low-level features in a video frame such as lighting / color contrast, flicker, motion, etc., a saliency map reveals, as a first order approximation, the amount of visual attention (saliency) each spatial region in the frame will draw from the viewer. By analyzing how spatial saliency in video frames changes over time, we can estimate the regions-of-interest (ROI) a viewer may choose to observe and how he may switch ROIs over time, resulting in HMM state transition probabilities.

During actual streaming, 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, gaze prediction using Kalman filtering [17] is performed to predict gaze location RTT into the future to reactively effect media content adaptation at server.

We demonstrate the applicability of our gaze prediction strategy through a networked video streaming application that performs bit allocation based on ROI. In face of limited network transmission bandwidth, the conventional end-to-end streaming approach [20, 21] 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 more aggressive signal quantization, often resulting in unacceptable visual experience.

One can alleviate this bandwidth-constrained problem by exploiting unique characteristics of the human perceptual system [15, 29, 36]. In particular, it has been shown [23, 29] that viewer's ability to perceive details away from the current gaze focal point falls precipitously as the angle away from the focal point increases. Thus, a smart bit allocation scheme [10, 35] can allocate more bits to ROI to minimize noticeable quantization noise, and fewer bits elsewhere. In this way, the perceived video quality remains the same while encoded bit-rate can be decreased. The technical 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 system to predict future gaze locations, so that optimal bit allocation can be performed for future frames. Experiments using our developed real-time video coding and streaming system, integrated with an off-the-shelf web camera and a software gaze tracker [2], show that transmission rate can be reduced by up to 29% without loss of perceived video quality for RTT as high as 200ms.

In the second half of this thesis, we are improving the proposed system by using saliency map analysis. Like the variable of presented HMM to correctly model a viewer's eye-gaze movements during playback of a video clip, HMM model parameters appropriate for the observed video clip must be derived. It's obviously that the different video contents contain different visual excitation through stimuli properties, inducing different amount of eye-gaze movements from viewers. For example, a video capturing a head-and-shoulder sequence of the president addressing the nation may induce very few gaze movements, while a dance music video with lots of new objects entering and leaving the scene may induce a lot. Thus, finding suitable HMM parameters given the visual activities of the video is important for eye-gaze movement modelling. Or even during one video sequence, it will contain different scenarios. Through the saliency map analysis, we can also partition the video into temporal segments of roughly stationary gaze statistics—each a set of consecutive frames that induce observer's gaze movements well described statistically by the same set of HMM parameters.

One brute-force method to derive appropriate HMM parameters for a given video content is to conduct extensive eye-gaze experiments [18], using a real-time gaze tracking system [2], with a sizeable group of test subjects. This, however, is clearly too time-consuming and cost-ineffective for a large number of video clips. Instead, we propose an alternative method to derive them by analysing the visual saliency maps [26] of individual video frames across time. Using saliency map analysis, we can compute the salient region of detected low-level features in a video frame such as lighting / color contrast, flicker, motion, etc. Then the relationship between salient objects within each two frames could be built by motion estimation. The probability of transition between salient objects and non-salient regions are treated as how often viewers may switch their regions-of-interest (ROI), which we call Visual Attention Deviation (VAD). By proposed VAD, we are able to measure and segment the video sequences, and to classify the video busyness avoiding extensive subjective experiment.

1.2 Motivation

With the development of video processing technologies, the conventional networked video streaming system is still transmitting pre-encoded video packages to viewers regarding on their requests. Facing more and more larger resolution and dimension videos, a human interactive system, which could adapt video content based on viewers' interest in real-time, will improve both streaming system efficiency and viewers' perceived visual quality.

We first proposed one video content adaptation system based on human gaze behavior, for the viewer's interest will be directly reflected by his visual attention during video playback. By tracking the gaze movement, we are able to predict his future gaze location by combining the current gaze information and video content. If the viewer keeps focusing on the same object of the frames over time, only the regions around this object will be labelled to encoded at a high quality, and other regions far away from it will be encoded at a low quality to reduce the frame size. At the same time, our human visual characters prove that the degradation outside the focal points is hard to notice. Then, a high percentage of bit saving will be gained without noticeable degraded visual quality as long as the future gaze location is predicted correctly.

As we know, there are millions of thousand videos created every minute, and different video content will cause different gaze movements, as different people have their own visual attention. If the viewer move his gaze location from one video object to another, then we say that he shifts his visual attention. A video that causes a viewer to shift his attention often is a “busy” video. Determination of which video content is busy is an important practical problem; a busy video is difficult for encoder to deploy region of interest (ROI)-based bit allocation, and hard for content provider to insert additional overlays like advertisements, making the video even busier. One way to determine the busyness of video content is to conduct eye gaze experiments with a sizeable group of test subjects, but this is time-consuming and cost-ineffective. We proposed one novel method using saliency map analysis to achieve it, which is also important and efficient for video adaptation based on gaze behavior.

1.3 Related work

1.3.1 Gaze tracking technology

While eye-gaze tracking has been studied extensively in the literature [43, 47]—including newer systems that do not require active calibration [9, 48] —there are relatively few prior work on eye-gaze prediction. Assuming a viewer's eye-gaze movements are either fixation or saccade, [30] first proposed a Kalman-filter-based eye-gaze movement prediction scheme to predict viewer's gaze location in the future. The same authors later improved their model by integrating it with a linear horizontal oculomotor plant mechanical model, a detailed motion model to predict eye movements based on the mechanics of the human eye using a large number of parameters [31, 32].

Our gaze-prediction strategy differs from [31, 32] in two major respects. First, rather than modelling the mechanics of the human eye, we approach the gaze prediction problem from a pure statistical learning perspective, where our two-state HMM is simple and maps intuitively to two of human's intrinsic gaze behavioral movements.

Second, unlike [31, 32] which predicted gaze movements in a content-independent manner, the major insight in our approach is to establish correlation between eye-gaze movements and the video content being watched. We do so because it has been shown in numerous subjective experiments in a variety of viewing scenarios [26, 42] that human visual attention is very often driven by innate visual stimulus in the observed content. Hence it is quite reasonable to assume that the aforementioned correlation exists and can be exploited for gaze prediction. This content-dependent approach has two implications: i) we only need to estimate very few parameters in a simple HMM model, and ii) only coarsely sampled gaze data are required to estimate the HMM state (tracking or saccade) an observer's gaze is currently in, so that a low-cost web camera capturing video at low frame rate (30 fps was used in our system) can be used in place of more expensive standalone gaze trackers used in [31, 32], lowering the barrier to mass deployment[1].

1.3.2 Smart bit allocation

The idea of preferentially allocating more resources to a region of interest during video encoding is not new [10, 35, 40]. While our primary interest is to use ROI-based bit allocation as a demonstration of the applicability of eye gaze prediction, the availability of real-time eye-gaze information does provide a firm basis for determination of ROI. In contrast, prior research without eye gaze information has to rely solely on video analysis such as high frequency content [40] and motion content [35],

[1] We note that because our low-cost gaze prediction system only makes predictions when the estimated state is tracking (saccade is deemed too complex to predict given the low sampling rate), the intended interactive media applications are limited to non-mission-critical ones, such as ROI bit allocation as detailed in this paper, and others as described in the first paragraph in the Introduction.

with the aforementioned saliency map also a suitable candidate. Nevertheless, it has been shown [6, 13] that prior knowledge and context play important roles in affecting viewer's attention. Thus, video analysis can at best provide a rough estimate of where viewers may look, in the absence of real-time information.

In contrast, we use saliency maps of video content to train HMM parameters during offline analysis, but combine real-time eye tracking information during stream time to determine ROI. 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 scenario. We will show in conducted subjective testing in Section 2.6.5 and Section 3.5.2.2 that ROI-based video encoding, where ROI is determined solely by saliency analysis with no real-time gaze tracking, is noticeably poorer in quality compared to video encoded in high quality for all spatial regions. On the other hand, our proposed ROI-based scheme with real-time gaze tracking performs much better in comparison.

1.3.3 Saliency Map Analysis

Visual attention (VA) modelling has focused many research efforts in the last decade following up efforts from the community of vision science and perception to better understand the fundamentals of visual attention. Several computational models to emulate VA have been consequently proposed, detecting the locations that attract the eye gaze. Most of the models compute a saliency map that values each pixel according to its visual saliency. While top-down visual saliency modelling is also possible [16], we focus our discussion in bottom-up visual attention process.

Several approaches, more or less biological, have been proposed. All the approaches share the same main principle: saliency is closely related to singularity or rareness. They can be classified into three different categories:

1. Hierarchical models [8, 26, 38, 39] based on computational architecture characterized by a hierarchical decomposition followed by ad hoc processing on each sub-band (e.g. DOG to mimic receptor field properties to seek for singularities) to estimate the salience. Different techniques are then used to aggregate this information across levels in order to build a unique saliency map.

2. Statistical models [7, 22, 41] based on probabilistic analysis of the content. Following the plausible link between saliency and singularity, the saliency at a given location is defined as a measure of the deviation between features at this location with respect to its neighborhood.

3. Bayesian models [25, 55] are useful to introduce prior-knowledge (e.g. contextual information like statistic of natural scene) and another alternative to cope with the saliency/singularity link. For instance, Itti and Baldi [25] introduced a Bayesian definition of surprise in order to measure the distance between posterior and prior beliefs of the observers. They proved that this measure, the surprise, is related to visual attention.

The quantitative assessment of the performances of these different models is still an open issue, but it appears that all these models reach similar results, whatever the assessment technique [37]. Our goal here is not to propose new visual saliency maps, but to use saliency maps, computed using previously established techniques, to derive HMM parameters offline in a computationally efficient way. This motivation is not unlike previous proposals that use saliency maps to resolve uncertainty in gaze estimates [9, 48, 52], except that our derived HMM parameters reflect the temporal aspect of expected gaze behavior, rather than the spatial aspect. In this paper, we selected methodology in [26] to compute saliency maps, based on a plausible model of bottom-up visual attention. Considering previous comments on performance, this model offers good performance with reasonable computational cost. An existing implementation of the model is available at [4]. We note, however, that our proposed gaze prediction strategy is agnostic to the particular type of saliency model, and thus can be made interoperable to other saliency models such as [16].

1.4 Contribution

In the first half of this thesis, we propose a novel dynamic gaze prediction strategy to estimate future gaze location to lower end-to-end reaction delay in gaze-based networked media systems. We first design a Hidden Markov Model (HMM) with two latent states that correspond to human's major types of intrinsic eye movements: Tracking: (fixation, pursuit) and Saccade [14]. HMM parameters are obtained offline by content analysing using saliency map analysis. During video playback, 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, dynamic Kalman Filter (KF) 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 networked video streaming application that performs bit allocation based on Region-of-Interest (ROI). Our experiments, using proposed real-time video coding and streaming system integrated with a web camera and a software gaze tracker [2], show that using our gaze-prediction strategy, transmission rate can be reduced by up to 29% without loss of perceived video quality for RTT as high as 200ms.

And by the latter half, we continue to strength the proposed human interactive system, like video stationarity detection, video content classification and so on. Those video characters would highly effect system performance for causing different human visual reaction. To obtain these necessary parameters, we developed a novel method by using saliency map analysis, which is called Video Attention Deviation (VAD).

The VAD implies saliency map calculation on each frame of wanted video to gain their saliency map. According to the salient probability distribution, we claim the salient objects for each frame, whose relationships between each two frames will be estimated by motion estimation. Then the transition probabilities between each existed salient objects or other non-salient regions would be obtained. And VAD value is proposed to measure how often people will switch their interested point between salient object and non-salient regions to achieve the scene change detection and video busyness classification. Our experiments show that VAD is able to detect the scene change and video busyness by computing saliency regions with given videos only, and the more important is that our VAD is matching subjective evaluation during our proposed logo-insertion application, which means it's reflecting human perception, besides it's much sensitive than other metrics.

1.5 Organization

The outline of the thesis is as follows: we first present the background and related work in Chapter 1. For gaze prediction strategy, we discuss how we model the gaze performance and predict gaze location in future one round trip time in Chapter 2. With the trained hidden markov model, we improved our prediction system using saliency map analysis in Chapter 3. And finally, the conclusion, discussion and future work are presented in Chapter 4.

Chapter 2

Gaze prediction based bit allocation scheme

2.1 Introduction

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 this section, 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 two states corresponding to human's major types of intrinsic eye movements. HMM parameters are obtained offline using saliency map analysis 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, Kalman filter prediction is used to predict gaze location RTT seconds into the future.

To validate our gaze prediction strategy, we focus 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.

2.2 Motivation

We demonstrate the applicability of our gaze prediction strategy through a networked video streaming application that performs bit allocation based on ROI. In face of limited network transmission bandwidth, the conventional end-to-end streaming approach [20, 21] 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 more aggressive signal quantization, often resulting in unacceptable visual experience.

One can alleviate this bandwidth-constrained problem by exploiting unique characteristics of the human perceptual system [15, 29, 36]. In particular, it has been shown [23, 29] that viewer's ability to perceive details away from the current gaze focal point falls precipitously as the angle away from the focal point increases. Thus, a smart bit allocation scheme [10, 35] can allocate more bits to ROI to minimize noticeable quantization noise, and fewer bits elsewhere. In this way, the perceived video quality remains the same while encoded bit-rate can be decreased. The technical 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 system to predict future gaze locations, so that optimal bit allocation can be performed for future frames. Experiments using our developed real-time video coding and streaming system, integrated with an off-the-shelf web camera and a software gaze tracker [2], show that transmission rate can be reduced by up to 29% without loss of perceived video quality for RTT as high as 200ms.

2.3 System Overview

First, a two-state Hidden Markov Model (HMM) is discussed to model eye gaze of a human observer watching video in section 2.4. An HMM describes transitions of sequential state X_n's, in discrete time, $n \in \mathcal{Z}^+$, where X_n is the state variable at time n, and $\mathcal{Z}^+$ denotes the set of positive integers. Each X_n can take on one of two possible latent states. State $\mathtt{T}$ (tracking) models the case when the gaze of the human observer is following the motion of an identifiable object in the video. In the gaze literature [14], it is common to further categorize gaze movements into fixation, which models the case when eye gaze is fixated at a stationary object, and smooth pursuit, which models the case where gaze follows a slowly moving object. However, for our intended purpose of gaze prediction, we only need to estimate the likelihood that the human observer has identified an object of interest and is currently tracking it—doing so would mean his/her gaze location will likely coincide with the locations of the moving object in future frames as well. Thus, for simplicity we use a combined state $\mathtt{T}$ to model observer's tracking of object in video.

State $\mathtt{S}$ (saccade) models the rapid transition of observer's gaze from one object of interest to another. More precisely, for the purpose of gaze prediction, we interpret state $\mathtt{S}$ simply to mean gaze statistics that do not conform to that of tracking state $\mathtt{T}$. No gaze prediction is made when state is estimated to be $\mathtt{S}$ due to saccade's more unpredictable nature compared to state $\mathtt{T}$[1]. Note that this definition of saccade deviates slightly from others in the literature [14], e.g., pursuit of a fast moving object (called catch-up saccade) will also be included in our definition of saccade. Nevertheless, this classification is more practical for our purpose of gaze prediction. Further, note that while other classifications of eye movements for the human eye are also possible [34], broadly speaking, fixation, pursuit and saccade are the three most frequently cited and major eye movement types in the literature [14].

We construct our HMM to be first-order 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

[1] Since these two states cannot be observed directly, they are commonly called latent states in the literature.

state variables X_n's are not directly observable; only observations 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 state X_n given observations $Y_1, \ldots, Y_n$ can be calculated using a simplified version of the forward algorithm (FA) (section 13.2.2, pp.618, [5], to be discussed). In our gaze tracking scenario, observations Y_n's can be either x- or y-coordinates of tracked eye-gaze locations on the display terminal; while the HMM transition probability is trained by saliency map analysis, which is introduced in Section 2.4.1. for simplicity, we construct the same HMM (but possibly with different parameters) to model gaze movements in x- and y-coordinates separately, while it is possible to construct a single HMM to jointly consider both x- and y-coordinates, we choose to construct separate HMMs for x- and y-coordinates for two reasons: i) a simpler model requires fewer data samples for the few model parameters to converge, and ii) a simpler model has lower complexity (our gaze prediction algorithm must be executed in real-time). And How we reconcile two latent states during gaze prediction is discussed in Section 2.4.3. Also we will show in Section 2.6 that we can achieve good gaze prediction results nonetheless.

With predicted future gaze location, we discuss a bit-allocation strategy in section 2.5. Conceptually, human ability to appreciate pixel fidelity decreases continuously away from the center of focus. Hence we adopt a simpler approach in which a rectangular region-of-interest (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. Thus a large percentage of bit saving could be achieved while the perceived visual quality won't be degraded, both objective and subjective experiments are designed to verify the performance of our proposed system.

2.4 Gaze Prediction

2.4.1 Hidden Markov Model

A hidden Markov model (HMM) is a statistical Markov model in which the system being modeled is assumed to be a Markov process with unobserved (hidden) states. An HMM can be considered as the simplest dynamic Bayesian network.

In a regular Markov model, the state is directly visible to the observer, and therefore the state transition probabilities are the only parameters. In a hidden Markov model, the state is not directly visible, but output, dependent on the state, is visible. Each state has a probability distribution over the possible output tokens. Therefore the sequence of tokens generated by an HMM gives some information about the sequence of states. Note that the adjective 'hidden' refers to the state sequence through which the model passes, not to the parameters of the model; even if the model parameters are known exactly, the model is still 'hidden'.

Hidden Markov models are especially known for their application in temporal pattern recognition such as speech, handwriting, gesture recognition, part-of-speech tagging, musical score following, partial discharges and bioinformatics. A hidden Markov model can be considered a generalization of a mixture model where the hidden variables (or latent variables), which control the mixture component to be selected for each observation, are related through a Markov process rather than independent of each other.

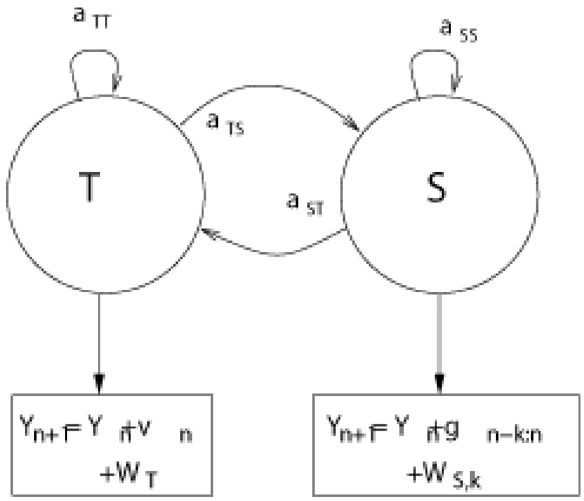


Figure 2-1: Proposed HMM for eye gaze during video observation.Circles denote latent states `T` (tracking), which includes fixation and smooth pursuit gaze movements, and `S` (saccade). α's denote state transition probabilities. Y's denote the observations. v is the pixel velocity vector. g is the gaze velocity vector. W's are the additive noise terms. Boxes denote observations.

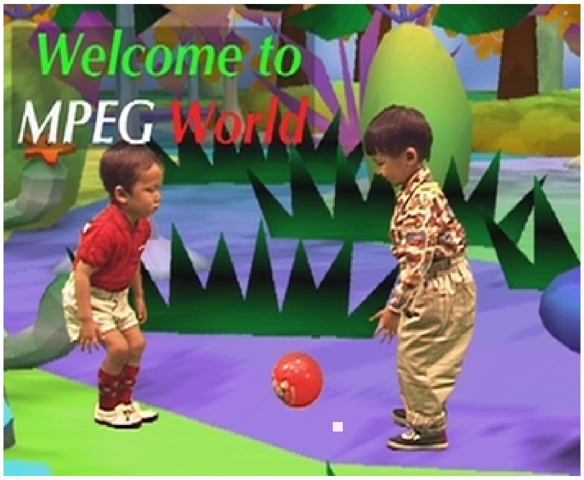


Figure 2-2: An unreliable eye gaze tracker often produces noisy observations.

In our proposed system, the HMM is trained to describe transitions of sequential state: T and S in discrete time. State T (tracking) models the case when the gaze of the human observer is following the motion of an identifiable object in the video. State S (saccade) models the rapid transition of observer's gaze from one object of interest to another. More precisely, for the purpose of gaze prediction, we interpret state S simply to mean gaze statistics that do not conform to that of tracking state T. No gaze prediction is made when state is estimated to be S due to saccade's more unpredictable nature compared to state T.

2.4.1.1 Tracking: following the motion of an identifiable object

If the value of state variable X_{n+1} is $\mathtt{T}$ (tracking) 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[2] $v_n(Y_n)$, plus random noise W_T:

$$Y_{n+1} = Y_n + v_n(Y_n) + W_T \tag{2.1}$$

$v_n(Y_n)$ is the velocity vector of the viewed pixel, as indicated by gaze point Y_n, from frame F_n of time n to frame F_{n+1} of time $n+1$, and W_T is a zero-mean Gaussian random variable with variance σ_T^2. If the gaze point of the observer in frame F_n is known precisely, $v_n(Y_n)$ can be estimated straightforwardly via video content analysis. For example, one can use optical flow algorithms [49], or more computation-efficient block search commonly used in video coding standards like H.263 [28], H.264 [54]: first identify the macroblock that contains the viewed pixel at time n, then find the best matched macroblock in frame F_{n+1} in terms of RGB pixel values, and calculate the corresponding motion vector. The probability of observing Y_{n+1} (emission probability) given current state is $\mathtt{T}$ is hence:

$$P_T(Y_{n+1} \| Y_n) = f_{\sigma_T^2}(Y_{n+1} - Y_n - v_n(Y_n)) \tag{2.2}$$

Unfortunately, the problem with (2.2) is that the true gaze point in frame F_n is not known precisely due to noise in the observation. That means that if a viewer is

[2] While pixel velocity v_n can also be considered as an observation, it is essentially a derivative of observation Y_n—movement of observed pixel located at Y_n from frame F_n to F_{n+1}. Thus we will only write Y_n as the sole independent observation value for each instant n.

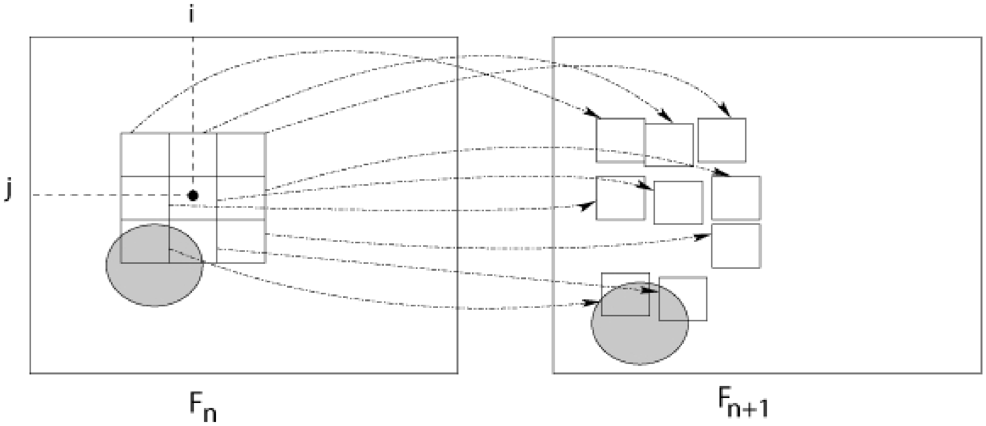


Figure 2-3: Calculation of forward motion vector candidates in next frame F_{n+1} given eye gaze data Y_n at location (i, j) in frame F_n.

actually following a moving object but gaze point is not on the object due to noise (as shown in Fig. 2-2, In this example, a viewer has focused on the red ball in this frame 220 of MPEG test sequence `kids`, but an eye tracker reports gaze location marked by the 5×5 white square.), then the calculated motion vector $\nu_n(\mathsf{Y}_n)$ will be erroneous.

To circumvent this problem, we perform multi-block search as shown in Fig. 2-3. 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 ν_n. Among all the calculated vectors ν_n's, we identify the one that gives the largest conditional probability for state `T`:

$$\begin{aligned} P_T(Y_{n+1}\|Y_n) &= \max_{v_n \in \mathcal{V}_n(Y_n)} f_{\sigma_T^2}(Y_{n+1} - Y_n - v_n) \\ v_n^* &= \arg\max_{v_n \in \mathcal{V}_n(Y_n)} f_{\sigma_T^2}(Y_{n+1} - Y_n - v_n) \end{aligned} \tag{2.3}$$

where $\mathcal{V}_n(Y_n)$ is the set of calculated motion vectors for a neighborhood of macroblocks around detected gaze point Y_n, and v_n^* is the motion vector in $\mathcal{V}_n(Y_n)$ that maximizes the tracking emission probability $P_T(Y_{n+1}\|Y_n)$ as expected.

2.4.1.2 Saccade: switching fixation points

If the viewer is in state $X_{n+1} = S$ (saccade) at time $n+1$, the gaze of the viewer is not following an identifiable object in the video, and thus is very likely switching from one object of interest to another. The transition process usually lasts a short duration (20 to 200ms), and the movement is fast [14]—saccade is said to be the fastest movement by the human body [14]. Fortunately, very often movement of the eye during one saccade is along a straight line [14]. Thus, if we are able to establish a gaze vector $g_{n-h+1:n}$ during saccade using previous observations Y_n's, then new observation Y_{n+1} is previous observation Y_n plus $g_{n-h+1:n}$, plus a noise term $W_{S,h}$.

There are many possible ways to model the complex saccade movement; we choose the simplest linear motion model for complexity reason. Mathematically, we write observation Y_{n+1} given viewer resides in state $X_{n+1} = S$ as follows:

$$Y_{n+1} = Y_n + g_{n-h+1:n} + W_{S,h} \tag{2.4}$$

where $g_{n-h+1:n}$ is the mean eye gaze vector computed using most recent $h \geqslant 2$ observations $Y_{n-h+1}, \ldots, Y_n$. $W_{S,h}$ is a zero-mean Gaussian variable, whose variance $\sigma^2_{S,h}$ depends on the number of observations, h, used to compute $g_{n-h+1:n}$. The idea is to capture the notion that, in general, the more recent observations Y_n's we use to estimate gaze vector $g_{n-h+1:n}$, the smaller the corresponding variance $\sigma^2_{S,h}$ of Gaussian noise $W_{S,h}$ should be. $g_{n-h+1:n}$ can be computed using samples $(n-h+1, Y_{n-h+1}), \ldots, (n, Y_n)$ via linear regression (section 3.1, pp.138, [5]). On the other hand, if gaze movement does not follow a straight line but a curvature instead (again, in rare cases), then more samples do not lead to better estimate of gaze vector $g_{n-h+1:n}$. In practice, we cap the maximum number of samples used to be no larger than a parameter H ($H = 15$ is used in our experiments).

We can now write the emission probability $P_S(Y_{n+1} \| Y_n, \ldots, Y_{n-h+1})$ of observing Y_{n+1} given previous h observations $Y_n, \ldots, Y_{n-h+1}$ and current state is S as follows:

$$P_S(Y_{n+1} \| Y_n, \ldots, Y_{n-h+1}) = f_{\sigma^2_{S,h}}(Y_{n+1} - Y_n - g_{n-h+1:n}) \tag{2.5}$$

We notice that $P_T(Y_{n+1} \| Y_n)$ in (2.3) and $P_S(Y_{n+1} \| Y_n, \ldots, Y_{n-h+1})$ have similar forms and would evaluate to have similar values if v^*_n and $g_{n-h+1:n}$ are similar (if the corresponding variance σ^2_T and $\sigma^2_{s,h}$ are also similar). This is the case when the observer is tracking an object in the video with slow linear motion, so that the motion vector and gaze vector coincide. Clearly, we should label this case as tracking state T, indicating that we can predict future gaze location with high probability. To disambiguate state S from T in this case, we do the following: we add a weighting parameter $1 - e^{\gamma \| v^*_n - g_{n-h+1:n} \|}$ to probability $f_{\sigma^2_{S,h}}$, so that if motion vector v^*_n is close

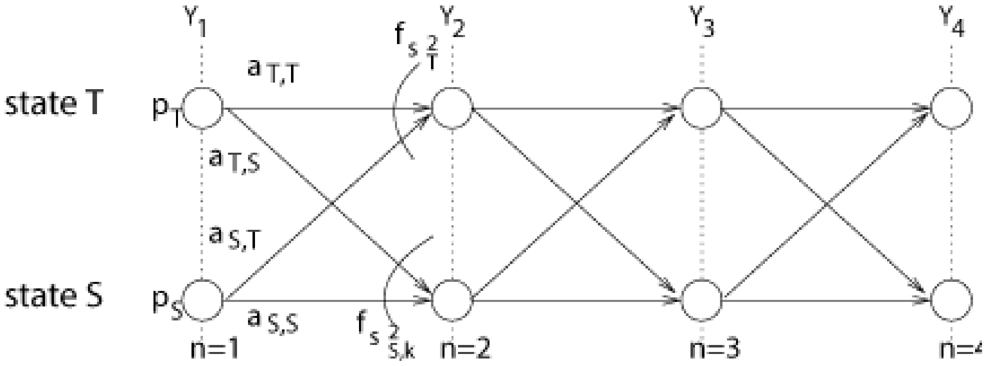


Figure 2-4: Trellis corresponding to a 2-state HMM. A Forward Algorithm can find the most likely state X_n given observations $Y_1, \dots, Y_n$'s.

to gaze vector $g_{n-h+1:n}$, then emission probability $P_S(Y_{n+1} \| Y_n, \dots, Y_{n-h+1})$ is small. To summarize, we can replace the earlier (2.5) with the following:

$$P_S(Y_{n+1} \| Y_n, \dots, Y_{n-h+1}) = (1 - e^{\gamma \| v_n^* - g_{n-h+1:n} \|}) \; f_{\sigma^2_{S,h}}(Y_{n+1} - Y_n - g_{n-h+1:n}) \qquad (2.6)$$

where γ is a parameter to control the weight factor (γ is set to -0.25 in our scheme).

2.4.1.3 Finding most likely latent states

To find latent state probability $P(X_n = j)$ given a window of observations $Y_1, \dots, Y_n$, we derive a simplified version of the forward algorithm (FA), which is the first half of the well known forward-backward algorithm (section 13.2.2, pp.618, [5]). It is a simplified version because, unlike the general case posed in [5], we do not have future observations $Y_{n+1}, \dots$ when estimating X_n given real-time collected observations

$Y_1, \ldots, Y_n$.

Mathematically, we seek to find latent variable X_n^* that maximizes the posterior probability $P(X_n \| Y_1, \ldots, Y_n)$ given observations $Y_1, \ldots, Y_n$. Using Bayes' theorem, we can write:

$$\begin{aligned} X_n^* &= \arg\max_{X_n} P(X_n \| Y_1, \ldots, Y_n) \\ &= \arg\max_{X_n} \frac{P(Y_1, \ldots, Y_n \| X_n) P(X_n)}{P(Y_1, \ldots, Y_n)} \\ &= \arg\max_{X_n} P(Y_1, \ldots, Y_n, X_n) \end{aligned} \tag{2.7}$$

This last line follows since the choice of X_n does not affect $P(Y_1, \ldots, Y_n)$. As done in [5], let $a(X_n) = P(Y_1, \ldots, Y_n, X_n)$. $a(X_n)$ can be written recursively (equation (13.36), pp. 620, [5]):

$$a(X_n) = P(Y_n \| X_n) \sum_{X_{n-1}} a(X_{n-1}) P(X_n \| X_{n-1}) \tag{2.8}$$

Note that (2.8) is computed in a recursive manner, meaning that as a new observation Y_{n+1} arrives, previously computed $a(X_n)$'s can be used for the computation of $a(X_{n+1})$'s, instead of computing the entire observation sequence $Y_1, \ldots, Y_{n+1}$ again. This is equivalent to constructing a new stage of a trellis of two states representing $a(\mathtt{T})$ and $a(\mathtt{S})$ at instant $n+1$, reusing computed states of the previous stage at instant n, as shown in Fig. 2-4.

To solve (2.8) we still need to complete two additional practical details. First is initial conditions, which can be calculated easily as follows:

$$\alpha(X_1) = P(Y_1, X_1) = \pi_X P(Y_1 \| X_1) \tag{2.9}$$

where π_X is the steady state probability of the Markov chain for latent state X.

The second is scaling factor. Because for each recursive call in (2.8) we need to multiply emission probability $P(Y_n \| X_n)$ which can be much smaller than 1, $\alpha(X_n)$ can become very small very quickly, leading to numerical instability. As done in section 13.2.4, pp.627, [5], we add in a coefficient c_n so that the sum of all $\hat{\alpha}(X_n)$'s is 1. (2.8) thus becomes:

$$c_n \hat{\alpha}(X_n) = P(Y_n \| X_n) \sum_{X_{n-1}} \hat{\alpha}(X_{n-1}) P(X_n \| X_{n-1}) \tag{2.10}$$

where c_n is chosen so that $\sum_{X_n} \hat{\alpha}(X_n) = 1$.

2.4.2 HMM Parameters Estimation

For the previously presented HMM to correctly model a viewer's eye-gaze movements during playback of a video clip, model parameters (most importantly, HMM state transition probabilities) appropriate for the observed video clip must be derived. Different video contents contain different visual excitation through stimuli properties, inducing different amount of eye-gaze movements from viewers. For example, a video capturing a head-and-shoulder sequence of the president addressing the nation may induce very few gaze movements, while a dance music video with lots of new objects entering and leaving the scene may induce a lot. Thus, finding suitable HMM parameters given the visual activities of the video is important for eye-gaze movement modeling.

One brute-force method to derive appropriate HMM parameters for a given video content is to conduct extensive eye-gaze experiments [18], using a real-time gaze tracking system [2], with a sizable group of test subjects. This, however, is clearly too time-consuming and cost-ineffective for a large number of video clips. Instead, we propose an alternative method to derive HMM parameters per video clip by analyzing the visual saliency maps [26] of individual video frames across time.

The saliency map was designed as input to the control mechanism for covert selective attention. Koch and Ullman (1985) posited that the most salient location (in the sense defined above) in a visual scene would be a good candidate for attentional selection. Once a topographic map of saliency is established, this location is obtained by computing the position of the maximum in this map by a Winner-Take-All mechanism. After the selection is made, suppression of activity at the selected location (which may correspond to the psychophysically observed "inhibition of return" mechanism) leads to selection of the next location at the location of the second-highest value in the saliency map and a succession of these events generates a sequential scan of the visual scene. This role of the saliency map in the control of which locations in the visual scene are attended is close to that of the "master map" postulated in the "Feature Integration Theory" proposed by Treisman and Gelade in 1980. The Koch and Ullman study was purely conceptual. The first actual implementation of a saliency map was described by Niebur and Koch in 1996. They applied their saliency map model which made use of color, intensity, orientation and motion cues both to simplified visual input (as is typically used in psychophysical experiments) and to complex natural scenes and they demonstrated sequential scanning of the visual scene in order of decreasing salience (see below). Later work refined the model [26]. The source code to compute saliency maps is freely available at http://ilab.usc.edu/toolkit/downloads.shtml.

Computed saliency maps for individual video frames describe visual attention variation spatially. For our purpose, we seek to describe visual attention variation of a video temporally, i.e., how a viewer will shift visual attention from one object of interest to another over time, which requires additional steps.

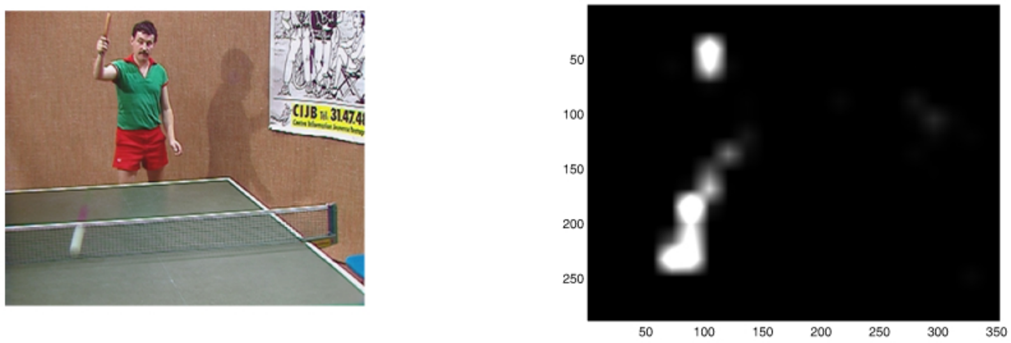
Our methodology is as follows. First, we define saliency objects within each video frame given calculated saliency maps; as a first-order approximation, saliency objects are the only regions a viewer may observe at that particular frame. Then, we derive HMM transition probabilities of a Markov model by solving consistency equations written for different saliency objects across consecutive frames. We describe these steps in order next.

2.4.2.1 Finding Saliency Objects

We first compute visual saliency maps for all video frames using methodology in [26]. [3]. As an example, in Fig. 2-5 we see an original video frame, frame 157 of MPEG test sequence `table`, and its corresponding computed saliency map. We see that saliency values are highest around the ping-pong ball and the hand, agreeing with our expectation of visual attention for this frame.

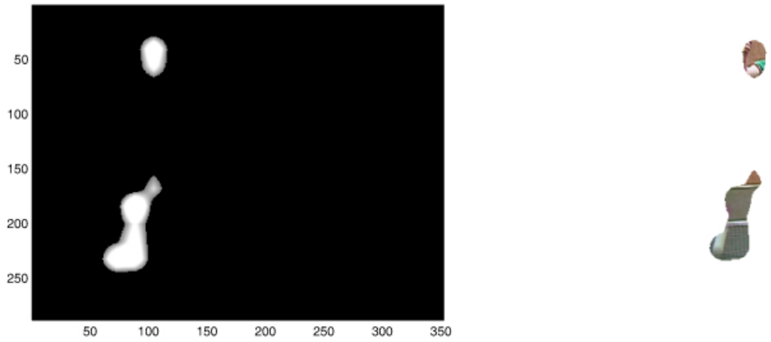
Having computed visual saliency maps, we first normalize each individual map, so the sum of all saliency values in a map equals to one. We then find a set of saliency

[3] We note again that our method of deriving HMM parameters is agnostic to the particular model used for saliency calculation, and thus our method can be easily adapted to other saliency models such as [16].



(a) original video frame (b) corresponding saliency map

Figure 2-5: Original video frame 157 of MPEG test sequence `table`, and the corresponding visual saliency map, calculated using method in Itti's model.



(a) saliency map w/ threshold (b) saliency objects

Figure 2-6: Normalized saliency map after applying threshold, and resulting salient objects in video frame 157 of sequence `table`.

objects in each map. We define a saliency object as a spatially connected region with per-pixel saliency value larger than a pre-defined threshold τ_s. As a first order of approximation, we assume these are the only video objects a viewer will observe in the given frame. A viewer may of course have gaze locations outside of these saliency objects; we assume that such occurrence means the viewer is in the process of switching from one saliency object to another; i.e., he is in saccade state at this frame time. Returning to our earlier example, we see in Fig. 2-6(a) the normalized saliency map with normalized saliency values below threshold τ_s set to zero, leaving only two saliency objects in the map. Correspondingly, we see the saliency objects in Fig. 2-6(b).

2.4.2.2 Merging of Saliency Objects

Because the computed saliency maps can be noisy, it turns out that finding a single appropriate threshold τ_S a priori that can identify reasonable saliency objects in all saliency maps of a video in time is difficult. To ease the burden of the threshold selection, we perform the following two procedural steps after initial saliency objects are found in a frame. First, if only a single saliency object is found in a frame, we incrementally lower threshold τ_S until a second saliency object is discovered. We do so because, by definition, the probability of a viewer being in saccade state in any frame is non-zero (i.e., there is a non-zero chance of a viewer switching objects of interest in any frame), and having a single saliency object means there are no other objects to switch to. Performing such procedure usually means the size of the original single saliency object increases as threshold τ_S is lowered. This agrees with intuition: the original object remains the main object with the strongest visual attention despite the decrease in threshold.

Second, for each discovered saliency object, we search within a radius r_s of the object's center to check if another object is in the vicinity. Note that an object center (c_x, c_y) is the Cartesian center of the object, where c_x and c_y are the arithmetic

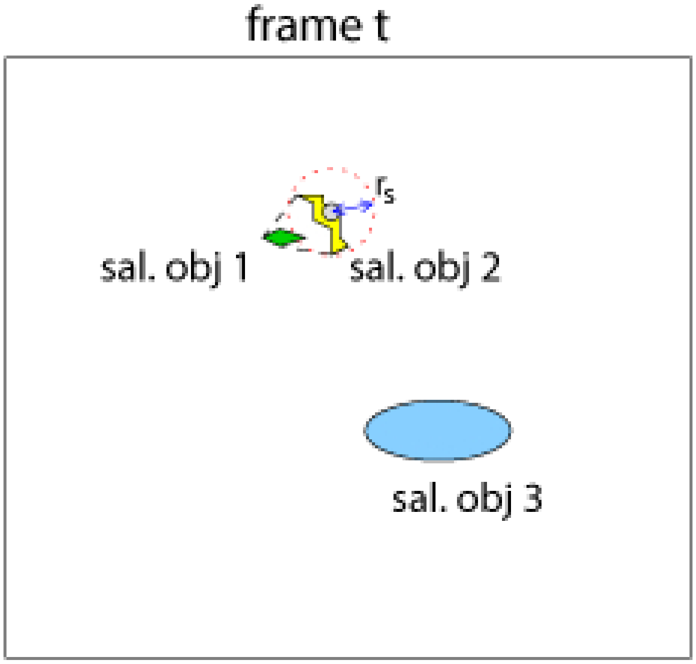


Figure 2-7: Example of how two small saliency objects (obj. 1 and obj. 2) are merged using search radius r_s.

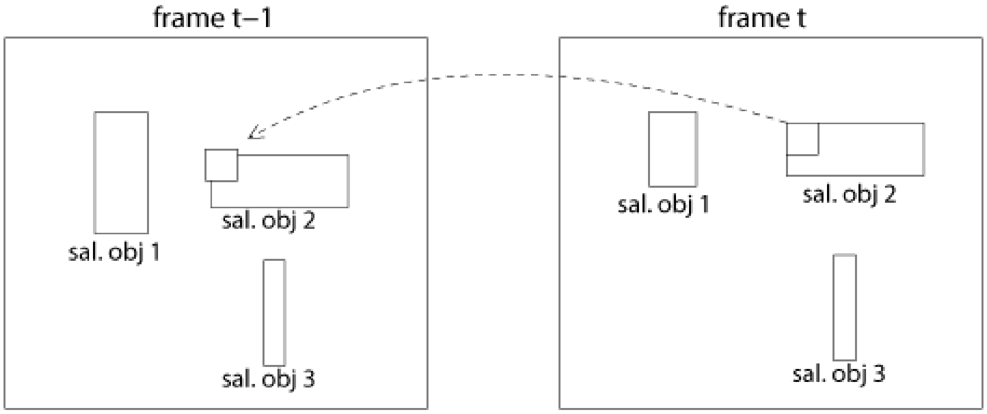


Figure 2-8: Example of how correspondence of saliency objects located in pairs of consecutive saliency maps are found using motion estimation (ME).

means of the x- and y-coordinates of every pixels in the object. If so, we merge the two objects via convex combination. In Fig. 2-7, we see saliency object 1 is within radius r_s of object 2's center, so we merge the two objects into one. The motivation of saliency object merging is the following. A viewer necessarily looks at a group of pixels at a time. So if a very small object $o_{t,i}$ (smaller than a circle of radius r_s) is in the vicinity of another object $o_{t,j}$, then the viewer is also looking at object $o_{t,j}$ when observing $o_{t,i}$. Thus it is sensible to merge the two objects.

2.4.2.3 Correspondence of Saliency Objects

We can establish correspondence among saliency objects in consecutive frames using motion estimation (ME), commonly used in video compression algorithms [28, 54]. In details, for each block k (we use 8×8 in our experiments) in a saliency object $o_{t,i}$ in saliency map of instant t, we find the most similar block in saliency map of instant $t-1$, i.e. the block with corresponding RGB pixel values in the original video frame $t-1$ that most matches RGB pixel values corresponding to block k in frame t. If the most similar block in saliency map $t-1$ belongs to a saliency object $o_{t-1,j}$, then object $o_{t-1,j}$ in map $t-1$ and object $o_{t,i}$ in map t could potentially be the same object. If a sufficiently large fraction of blocks k's in $o_{t,i}$ map to the same object in $o_{t-1,j}$, then we declare they are the same object. If no such object exists in previous map $t-1$, then we declare object $o_{t,i}$ to be a new object appearing for the first time in map t. As an example, in Fig. 2-8, we see that a block in object 2 in frame t has found a matching block in object 2 in frame $t-1$.

2.4.2.4 Deriving Transition Probabilities

Having identified saliency objects across frames, we now derive state transition probabilities for our eye-gaze HMM. As an illustrative example, we examine the simple case where there are the minimum two salient objects in consecutive frames t and $t+1$. Denote by $p_{t,1}$ and $p_{t,2}$ the probability that a viewer will fix his gaze in each of the two objects, respectively, in frame i. Similarly, denote by $p_{t+1,1}$ and $p_{t+1,2}$ the corresponding probabilities for frame $t+1$. Let s_t and s_{t+1} be the probabilities that a viewer is in saccade state in frame t and $t+1$. Because we know the volume of visual saliency for each saliency object (sum of computed saliency pixel values within each object) and saccade spatial region (area not covered by saliency objects), we can calculate the relative probability size of objects by comparing their volumes in each frame:

$$\begin{aligned} s_t &= p_{t,1}/\beta_{t,1} = p_{t,2}/\beta_{t,2} \\ s_{t+1} &= p_{t+1,1}/\beta_{t+1,1} = p_{t+1,2}/\beta_{t+1,2} \end{aligned} \tag{2.11}$$

where β's are the scaling factors among objects in each frame.

Further, we know that the sum of probabilities in each frame must equal 1:

$$\begin{aligned} p_{t,1} + p_{t,2} + s_t &= 1 \\ p_{t+1,1} + p_{t+1,2} + s_{t+1} &= 1 \end{aligned} \tag{2.12}$$

Together with (3.1), we can determine the gaze probability of each object in each frame. This is true no matter how many saliency objects are in each frame.

To calculate the state transition probabilities α's, we apply the definition of state transition to the objects of these two frames. We can write the probability $p_{t+1,1}$ of object 1 in frame $t+1$ as the sum of probabilities of objects in previous frame scaled by view transition probabilities α's:

$$p_{t+1,1} = p_{t,1}\ \alpha_{TT} + s_t\ \alpha_{ST} \left(\frac{\beta_{t+1,1}}{\sum_{i=1}^{2} \beta_{t+1,i}} \right) \tag{2.13}$$

Note that the probability $s_t\ \alpha_{ST}$ from state $\mathtt{S}$ to $\mathtt{T}$ must be split between the two objects, according to their relative volumes.

We can write a similar equation for probabilities $p_{t+1,2}$ of moving objects 2 in frame $t+1$. Further, we can similarly write state transition equation for the saccade state as well:

$$s_{t+1} = \alpha_{TS} \sum_{i=1}^{2} p_{t,i} + s_t \ \alpha_{SS} \tag{2.14}$$

Note that we have now three state transition equations for the four unknown α's. In general, one can obtain $k+1$ state transition equations for k saliency objects. In addition, we know the sum of probabilities leaving a state in a HMM must also be one:

$$\begin{aligned} \alpha_{TT} + \alpha_{TS} &= 1 \\ \alpha_{SS} + \alpha_{ST} &= 1 \end{aligned} \tag{2.15}$$

These two linear equations, together with the earlier derived three linear state transition equations, means that we have more equations than unknowns. We hence compute α's as follows. We rewrite each linear equation i with an additional noise term n_i at the end. The set of linear equations becomes:

$$\mathrm{Ca} = \mathrm{b} + \mathrm{n} \tag{2.16}$$

where $\mathrm{a} = [\alpha_{TT}, \alpha_{TS}, \alpha_{SS}, \alpha_{ST}]^T$ is the vector of α's we are seeking, C is the coefficient matrix, b and n are the constant and noise vectors, respectively. It is well known that the a^* that minimizes the noises n in a mean square sense is computed as follows:

$$\mathrm{a}^* = C^+\mathrm{b} \tag{2.17}$$

where $C^+ = (C^TC)^{-1}C^T$ is the Moore-Penrose pseudo-inverse of matrix C.

Having computed sets of transition probabilities α's each using different pairs of neighboring saliency maps in time, the transition probabilities for the video is simply the average of the computed sets of transition probabilities. We can then also compute the steady state probabilities π's of the HMM by performing eigen-analysis as typically done in the literature.

2.4.3 Prediction Schemes

We have discussed how to determine the most probable latent state X_n in HMM given observations $\mathsf{Y}_1, \ldots, \mathsf{Y}_\mathsf{n}$ in Section 2.4.1. In this section, we discuss how a future gaze location $\bar{\mathsf{Y}}_{\mathsf{n}+\mathsf{RTT}}$ can be estimated RTT gaze samples into the future. Smart bit allocation can then be performed to assign finer quantization parameter (QP) for ROI centered on predicted location $\bar{\mathsf{Y}}_{\mathsf{n}+\mathsf{RTT}}$, and coarser QP for other spatial regions in a coded frame (to be discussed in Section 2.5).

We stress here that we perform gaze prediction only if the most likely state is $\mathtt{T}$. This may seem counter-intuitive, since it is commonly accepted that the human eyes cannot perceive any visual details when in saccade state $\mathtt{S}$ [34], and so it appears that, for the ROI bit allocation application, the greatest bit-saving can be achieved when the viewer is in state $\mathtt{S}$. However, the duration in which a viewer stays in state $\mathtt{S}$ is typically very short [14], and gaze will soon stop at an unpredictable new object of interest. Thus, reducing bit-rate through coarser quantization of the video frames when viewer is in state $\mathtt{S}$ poses a significant risk of not reacting fast enough to improve video quality back up when viewer suddenly switches from state $\mathtt{S}$ to $\mathtt{T}$. This is particularly the case when a low-cost web camera capturing video at a low frame rate is used for gaze tracking. Hence, we take the conservative approach and perform no gaze prediction in state $\mathtt{S}$.

Further, even if the most likely state is $\mathtt{T}$, we perform prediction only if probability $P(X_n = \mathtt{T} \| Y_1, \ldots, Y_n)(\alpha_{TT})^{RTT}$ exceeding a threshold τ_C for both x- and y-coordinate state estimation. In other words, we employ prediction of gaze location to perform optimized bit-allocation only if:

1. We have confidence in our state estimation $P(X_n = \mathtt{T} \| Y_1, \ldots, Y_n)$; and,

2. The likelihood of the observer staying in state $\mathtt{T}$ RTT gaze samples into the future remains high.

For example, a long RTT between server and client, or a video content that contains many salient objects and induces much gaze movement (small α_{TT}), will limit the fraction of time we actually make gaze prediction.

2.4.3.1 Linear Prediction

At first beginning, we employ the Least Squares Linear prediction for the future gaze location. To estimate Y_{n+RTT}, we use a window of ω observations $Y_{n-\omega+1}, \ldots, Y_n$ for linear regression [5]. Specifically, using observations $(n - \omega + 1, Y_{n-\omega+1}), \ldots, (n, Y_n)$, we seek a linear function $Y(t) = \hat{\phi} + \hat{m}t$, so that the sum of squared errors between sample points and the linear function is minimized. This is illustrated in Fig. 2-9 for a window of five observations. The slope $\hat{m}$ and y-intercept $\hat{\phi}$ of the straight line that realize least squared error can be readily obtained for a set of observations y's taken at instant x's as: where the bar above $\bar{x}$ signifies the mean.

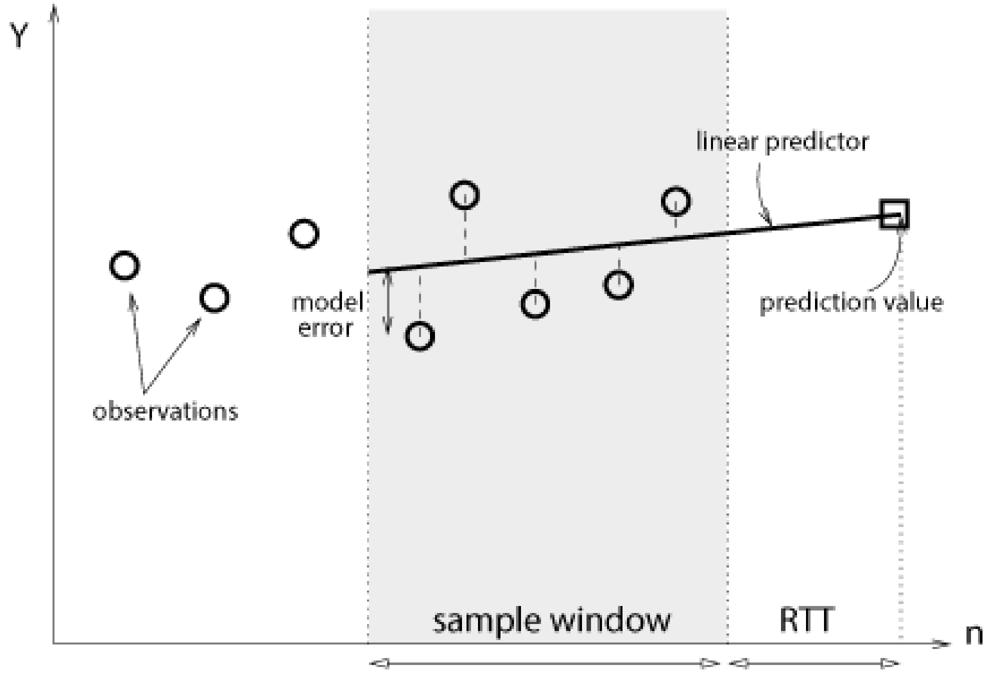


Figure 2-9: Least squares linear regression can be employed over a sample window to form a predictor for future gaze locations.

Given linear regression parameters, $\hat{m}$ and $\hat{\phi}$, predicted gaze point RTT into the future, $\bar{Y}_{n+\mathsf{RTT}}$, is extrapolated to be $\hat{\phi} + \hat{m}(n + \mathsf{RTT})$. For state $\mathtt{T}$, $\hat{m}$ is simply the constant velocity of the motion in the window of ω sample points.

2.4.3.2 Kalman Filter Prediction

Considering the accuracy of linear prediction, we developed our prediction model with Kalman Filter. The Kalman filter, also known as linear quadratic estimation (LQE), is an algorithm that uses a series of measurements observed over time, containing noise (random variations) and other inaccuracies, and produces estimateds of unknown variables that tend to be more precise than those based on a single measurement alone.

Given $\mathsf{P}(X_n = \mathtt{T} \| Y_1, \ldots, Y_n)(\alpha_{\mathtt{TT}})^{\mathsf{RTT}} \geqslant \tau_C$, we first denoise D latest samples of noise-corrupted observations $Y_{n-D+1}, \ldots, Y_n$ into estimated gaze points $\hat{Y}_{n-D+1}, \ldots, \hat{Y}_n$ using Kalman filtering (KF) [17]. D is the size of a small window of previous gaze samples (for complexity reason) that have been estimated to be state $\mathtt{T}$ during HMM state estimation.

To conform to the standard KF formulation, we modify previous notation to the following. Let $\hat{Y}_n$ and $\dot{\hat{Y}}_n$ be the true gaze location and velocity at time n. Denote by $\hat{\mathrm{Y}}_n = [\hat{Y}_n \ \dot{\hat{Y}}_n]^T$ the 2×1 vector that contains the true gaze location and velocity at time n. We write the evolution of $\hat{\mathrm{Y}}_n$ recursively as a linear dynamic system (LDS):

$$\hat{\mathrm{Y}}_n = \underbrace{\begin{bmatrix} 1 & (1-\beta) \\ 0 & (1-\beta) \end{bmatrix}}_{\mathrm{F}_n} \hat{\mathrm{Y}}_{n-1} + \underbrace{\begin{bmatrix} \beta \\ \beta \end{bmatrix}}_{\mathrm{B}_n} v^*_{n-1} + \begin{bmatrix} W_P \\ 0 \end{bmatrix} \tag{2.18}$$

where F_n and B_n are respectively the state transition and control-input models, ν^*_{n-1} is the control vector (also the emission probability maximizing block motion vector in (2.1)), and W_P is a zero-mean Gaussian process noise with variance σ^2_P . In words, (2.18) states that the next true gaze location $\hat{Y}_n$ is the previous gaze location $\hat{Y}_{n-1}$, plus $(1-\beta)$ times the gaze vector $\dot{\hat{Y}}_{n-1}$, plus β times the maximizing block motion vector ν^*_{n-1}, plus a noise term W_P. β is a parameter to control the convex combination of previous gaze velocity vector and motion vector of the scene. In our experiments, β is set close to 1. Note that having first derived ν^*_{n-1} using (2.1) during HMM state estimation in Section 2.4.2, it is then possible to write (2.18) as a LDS in each given instant n.

The observation $\mathrm{Y}_n = [\mathsf{Y}_n\ \dot{\mathsf{Y}}_n]^T$ is simply $\hat{\mathrm{Y}}_n$ plus an observation noise term:

$$\mathrm{Y}_n = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \hat{\mathrm{Y}}_n + \begin{bmatrix} W_O \\ 0 \end{bmatrix} \tag{2.19}$$

where W_O is a zero-mean Gaussian observation noise with variance σ^2_O.

Having written the evolution and observation equation (2.18) and (2.19), we can compute the estimated gaze location $\hat{Y}_{n-D+1}, \ldots, \hat{Y}_n$ using standard KF predict and update equations. See [17] for details.

Given estimated gaze point $\hat{Y}_n$, we predict gaze RTT samples into the future using a similar LDS setup. However, because there are no future observations available beyond Y_n, Kalman filtering reduces to a simpler LDS setup with no data denoising.

We write a similar evolution equation for Y_n into the future as follows:

$$Y_n = \underbrace{\begin{bmatrix} 1 & (1-\beta) \\ 0 & (1-\beta) \end{bmatrix}}_{F_n} Y_{n-1} + \underbrace{\begin{bmatrix} \beta c_1 & \dots & \beta c_Z \\ \beta c_1 & \dots & \beta c_Z \end{bmatrix}}_{B_n} \underbrace{\begin{bmatrix} v^1_{n-1} \\ \vdots \\ v^Z_{n-1} \end{bmatrix}}_{u_n} \qquad (2.20)$$

where $v^1_{n-1}, \dots, v^Z_{n-1}$ are the Z block MVs around gaze point Y_{n-1}.

In words, (2.20) states that gaze location Y_n is the previous location Y_{n-1} plus $(1-\beta)$ times previous velocity $\dot{Y}_{n-1}$, plus β times a weighted combination of MVs of the surrounding blocks $v^1_{n-1}, \dots, v^Z_{n-1}$, where weights c_i's sum to 1, $\sum_{i=1}^{Z} c_i = 1$. The weights $c_1, \dots, c_Z$ are used to compensate for the fact that observation Y_{n-1} is not available to select the MV that maximizes the emission probability, as done in (2.3). To predict gaze location RTT samples into the future, we repeatedly compute (2.20), starting from the last estimated gaze location $\hat{Y}_n$. After RTT iterations, we have an estimated gaze location $\hat{Y}_{n+\mathsf{RTT}}$ into the future.

2.5 ROI based Bit Allocation

In this section, we discuss a bit-rate allocation strategy as an application of our proposed HMM based eye-gaze prediction method. And we already described how to predict the location of future eye gaze $\hat{Y}_{n+\mathsf{RTT}}$ in the previous sections, 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 [10]. Conceptually, human ability to appreciate pixel fidelity decreases continuously away from the center of focus. Hence it is wasteful to encode visual information away from focus with high fidelity. 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. Specifically, regions far away from focus is no longer subjected to extreme quantization, which yields little additional rate reduction, but may attract unwanted attention due to large quantization artifacts, changing the visual saliency of the original video frames [53].

2.5.1 Determining ROI

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 2.6 with a field of view of 55 degrees, this corresponds to a ROI with field of view of 30 degrees, which is quite large to comfortably capture regions of high visual sensitivity.

When the estimated state is state S, which means the views are switching between different tracking points, therefore, we withdraw the optimization, and ROI should be the whole display, all the details in the screen will be encoded as high quality without optimization.

2.5.2 ROI based Encoding Scheme

As discussed in [23], 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:

$$\begin{aligned} \mathsf{CT}(f, e) &= \mathsf{CT}_0 \exp\left(\alpha f \frac{e + e_2}{e_2}\right) \\ \mathsf{CS}(f, e) &= 1/\mathsf{CT}(f, e) \end{aligned}$$

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 [10], we determine the cutoff frequency, f_c, by setting CT to one:

$$f_c = \frac{e_2 \log \frac{1}{\mathsf{CT}_0}}{\alpha(e_{\max} + e_2)} \tag{2.21}$$

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:

$$\frac{\mathsf{QP}_{\mathsf{ROI}}}{\mathsf{CT}_{\mathsf{ROI}}} = \frac{\mathsf{QP}_{\overline{\mathsf{ROI}}}}{\mathsf{CT}_{\overline{\mathsf{ROI}}}} \tag{2.22}$$

For ease of computation, we are primarily interested in having only two regions, namely inside and outside the ROI, and having rectangular ROI. Nevertheless, the scheme can be trivially extended to multiple regions, and to non-rectangular ROI.

In addition, the saliency map will change corresponding to the QP change. To avoid the effect, QP also should be selected carefully according to:

$$D_{KL}(QP_{ROI} + QP_{\overline{ROI}} \| \| QP_{Full}) < \rho \tag{2.23}$$

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 2.6 with a field of view of 55 degrees, this corresponds to a ROI with field of view of 30 degrees, which is quite large to comfortably capture regions of high visual sensitivity.

To ensure that the predicted observer's gaze movement does synchronize with an identified moving object in the video from frame F_n to F_{n+RTT}, we perform one final check to see if the predicted gaze location $\hat{Y}_{n+RTT}$ lands inside the same saliency object in frame F_{n+RTT} as it did in frame F_n. If it does not, then we declare uncertainty in the prediction, and the entire frame is encoded in high quality.

2.6 Performance Evaluation

We demonstrate the benefit of our proposed HMM-based gaze prediction strategy through both objective and subjective experiments. We first describe the setup of our experiments in Section 2.6.1. In part one of the experiment in Section 2.6.1, we show that our proposed saliency map analysis can be used to derive accurate HMM parameters. In part two, described in Section 2.6.2, we examine the accuracy of our HMM state estimation, and the tradeoff between false positive (predicting HMM state to be T when ground truth is S) and false negative (predicting HMM state to be S when ground truth is T). In part three, described in Section 2.6.3, we examine the accuracy of our HMM-based gaze prediction using Kalman filtering. In part four, described in Section 2.6.4, we examine the achievable bit-rate saving for our proposed bit allocation scheme. Finally, through an extensive subjective study, we show that our bit allocation scheme suffers no statistically meaningful loss in perceived visual quality, using our in-house developed real-time system, in Section 2.6.5.

2.6.1 Experiment Setup

Our gaze-based networked streaming system employs the free real-time gaze-tracking software opengazer [4] [2], which is calibrated for sampling gaze location at 30 samples per second using an off-the-shelf web camera.

In our experiment, we used two kinds of sequences: i) 300-frame standard MPEG video test sequences at CIF resolution (352 × 288), and ii) 150-frame video sequence at SD resolution (720 × 576) that can be downloaded from [1]. To mitigate viewer frustration from repetitive viewing, we used five CIF videos: silent, table, mother, foreman, kids, and five SD videos: captain, group, racing, rowboat, concert.

The monitor used for gaze tracking and video experiments measured 24 inches

[4] Though opengazer requires a viewer to hold his/her head still for accurate gaze tracking, we note that newer gaze trackers allow a viewer to move his head naturally (within an acceptable range) without affecting gaze tracking performance. Thus it is conceivable that a gaze tracking / prediction system utilizing advanced gaze tracking technologies can be practically deployed.

diagonally ($522.3\mathrm{mm} \times 329.6\mathrm{mm}$), with resolution of 1920×1200. Brightness and contrast are set to 30% and 50%, respectively. The distance between a user's head and the center of monitor screen is about $500\mathrm{mm}$, resulting in a viewing angle of about 55 degrees to the side-edge of the screen.

For video compression, we use a fast implementation of H.263 [28] for real-time encoding. For subjective testing, videos were displayed in full-screen mode at 30 fps (for CIF) and 15 fps (for SD), either the same or half the sampling rate of `opengazer` for one-to-one or two-to-one correspondence between gaze samples and video frames.

2.6.2 Results for HMM State Estimation

We now validate our proposed saliency map analysis discussed in Section 2.4.2, i.e., whether HMM state transition probabilities derived from saliency map analysis are roughly the same as "ground truth data". The ground truth model parameters are derived as follows. First, a trained user performed multiple viewings of each test sequence, each time recorded his intention of tracking state `T` or saccade state `S` by pressing keys on a keyboard during state transitions. This data set serves as initial guess Θ of the ground truth HMM model parameters.

Then, we use the forward-backward algorithm (section 13.2.2, pp.618, [5]) to refine model parameters Θ as follows. In Section 2.4.1.3, we defined forward probability $\alpha(X_n)$ in (2.8). We now define its counterpart—backward probability—as follows (equation (13.38), pp.622, [5]):

$$\begin{aligned} b(X_n) &= P(Y_{n+1}, \ldots, Y_N \| X_n) \\ &= \sum_{X_{n+1}} b(X_{n+1}) P(Y_{n+1} \| X_{n+1}) P(X_{n+1} \| X_n) \end{aligned} \tag{2.24}$$

where X_n is the latent state at instant n, and Y_n is the observed gaze location at instant n, $n \in \{1, \ldots, N\}$. Like (2.8), (2.24) can also be computed recursively.

Using forward probability $a(X_n)$ and backward probability $b(X_n)$, we can calculate the following quantity (equation (13.43), pp.623, [5]):

$$\xi(X_{n-1}, X_n) = \frac{a(X_{n-1}) P(Y_n \| X_n) P(X_n \| X_{n-1}) b(X_n)}{P(Y)} \tag{2.25}$$

Finally, we can estimate transition probability $\alpha_{j,k}$ from state j to k using $\xi(X_{n-1}, X_n)$ (equation (13.19), pp.617, [5]):

$$\alpha_{j,k} = \frac{\sum_{n=2}^{N} \xi(X_{n-1} = j, X_n = k)}{\sum_{l=1}^{K} \sum_{n=2}^{N} \xi(X_{n-1} = j, X_n = l)} \tag{2.26}$$

where l takes on all possible latent state values, which in our case is simply state `T` and `S`.

HMM parameters can be calculated by repeating the above equations until the differences of the HMM parameters between iterations are all lower than a pre-set threshold $\sigma =$ 1e-05. We use the resulting HMM model parameters as “ground truth data”.

State transition and steady state probabilities for `silent` and `table` are shown

Table 2.1: State transition and steady state probabilities for silent

(a) Forward-Backward algorithm

	T	S	π
T	0.891	0.109	0.841
S	0.577	0.423	0.159

(b) saliency map analysis

	T	S	π
T	0.885	0.115	0.836
S	0.588	0.412	0.164

Table 2.2: State transition and steady state probabilities for table

(a) Forward-Backward algorithm

	T	S	π
T	0.893	0.107	0.598
S	0.159	0.841	0.402

(b) saliency map analysis

	T	S	π
T	0.865	0.135	0.546
S	0.162	0.838	0.454

in Table 2.1(a) and 2.2(a), respectively. Notice that silent is a relatively "quiet" video [19]—one with little visual attention shifts, with the saccade steady state probability π_S much smaller than table. For comparison, the state transition probabilities derived via our proposed visual saliency map analysis for silent and table are shown in Table 2.1(b) and 2.2(b), respectively. We see that the derived HMM parameters using saliency maps analysis are fairly close to the ground truth gaze data trace. In particular, we see that the analytical saccade steady state probability π_S for both silent and table are very close to the ground truth trace numbers, even though π_S's for silent and table are very different. This shows accuracy of our proposed saliency map analysis.

We performed the same experiment for the two SD test sequences, captain and group as well. captain is a "quiet" video, while group is a "busy" video. The

Table 2.3: State transition and steady state probabilities for `captain`

(a) Forward-Backward algorithm

	`T`	`S`	π
`T`	0.882	0.118	0.699
`S`	0.274	0.726	0.301

(b) saliency map analysis

	`T`	`S`	π
`T`	0.924	0.076	0.643
`S`	0.137	0.863	0.357

Table 2.4: State transition and steady state probabilities for `group`

(a) Forward-Backward algorithm

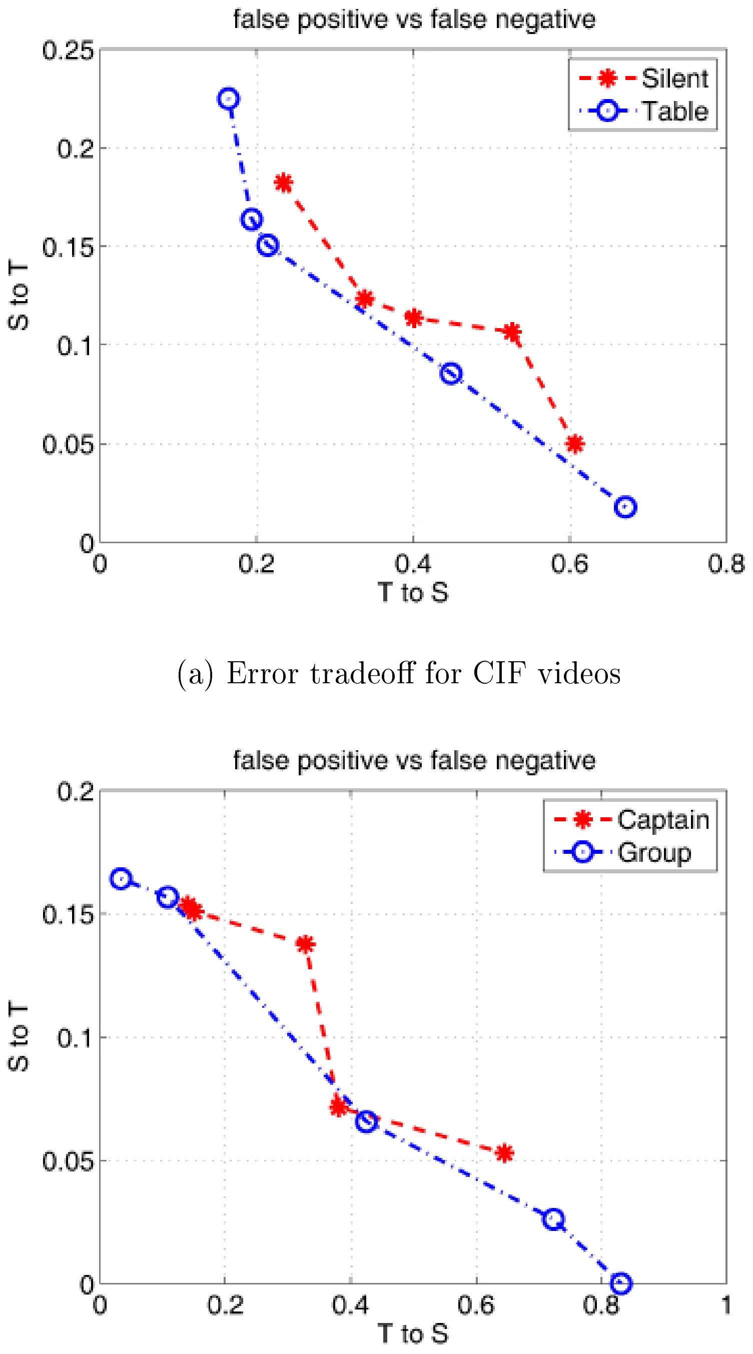
	`T`	`S`	π
`T`	0.823	0.177	0.356
`S`	0.122	0.878	0.644

(b) saliency map analysis

	`T`	`S`	π
`T`	0.879	0.121	0.367
`S`	0.067	0.933	0.633

resulting HMM state transition and steady state probabilities are shown in Table 2.3 and 2.4. We again see very similar numbers between HMM parameters derived using saliency map analysis and ones obtained using eye-gaze data trace. Having validated our approach, we will henceforth use HMM parameters derived from saliency map analysis.

We now evaluate the accuracy of HMM state estimation using forward algorithm (FA), as discussed in Section 2.4.1.3. We denote an occurrence as false positive when FA estimates HMM state to be `T` but the ground truth state is `S` (`S` to `T`). 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 the entire frame. In contrast, we denote an occurrence as false negative when FA estimates HMM state to be `S` but ground truth state is `T` (`T`



(a) Error tradeoff for CIF videos

(b) Error tradeoff for SD videos

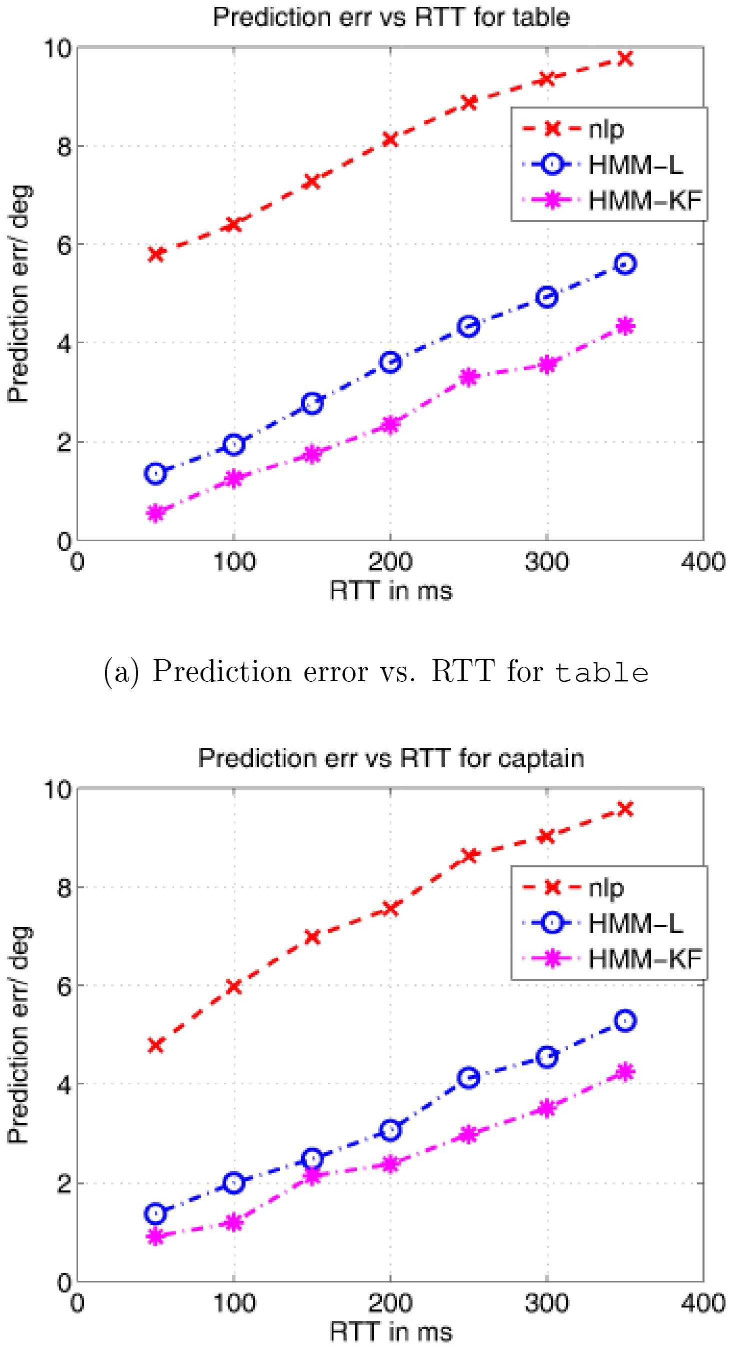
Figure 2-10: Tradeoff in false positive and false negative probabilities by adjusting threshold τ_C, for CIF and SD videos, respectively.

to `S`). This is the case where we miss a bit-saving opportunity.

As discussed in Section 2.4.3, a threshold τ_C can be adjusted according to our confidence in the estimated `T` state, resulting in a tradeoff between false positive and false negative probabilities. In Fig. 2-10, we see the said tradeoff in the two probabilities in our HMM state estimation for the two CIF (`silent` and `table`) and SD (`captain` and `group`) sequences, 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 ($\leqslant 0.15$ for false positive and $\leqslant 0.2$ for false negative) values for both. This shows that FA can provide reasonable state estimates for our proposed HMM. As we will discuss later, this level of estimation accuracy is sufficient for our intended application of ROI-based bit allocation for streaming video.

2.6.3 Results for Kalman Filter Prediction

Given estimated HMM states, we next examine the accuracy of our HMM-based gaze prediction using Kalman filter (`HMM-KF`), as discussed in Section 2.4.3. We compare first `HMM-KF` to a naïve linear prediction scheme (`nlp`), where the last two gaze data points are used to construct a straight line, which is then extrapolated to RTT seconds later to yield a gaze location estimate. We also compare `HMM-KF` to our previous HMM-based linear prediction scheme (`HMM-LP`) [18], where linear regression is used to construct a straight line using a window of previous gaze samples, then extrapolated into the future for gaze estimate as done for `nlp`. In Fig. 2-11, we see the performance of all schemes, in terms of visual degree between the estimated gaze



(a) Prediction error vs. RTT for `table`

(b) Prediction error vs. RTT for `captain`

Figure 2-11: Prediction Error in degree as function of RTT for different prediction schemes, for `table` and `captain`, respectively.

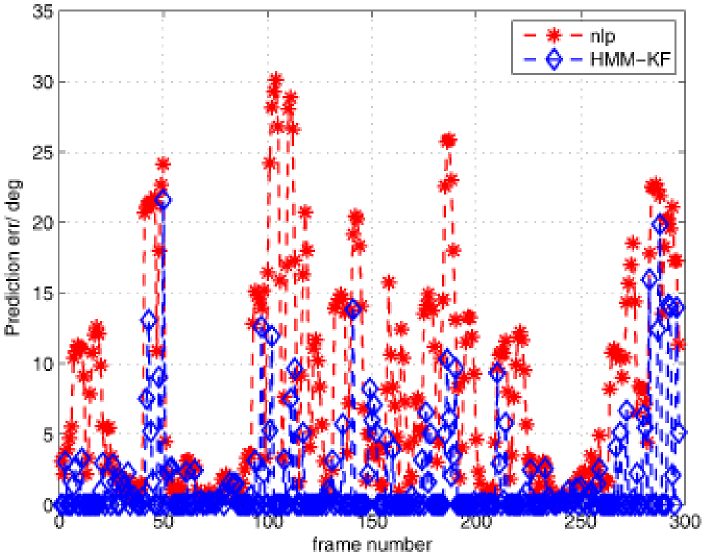
locations and true gaze locations, as function of RTT for CIF sequence `table` and SD sequence `captain`. We see that as RTT increased, the estimation error increased for all schemes. However, `HMM-LP` and `HMM-KF` achieved much smaller errors than `nlp`. This is because, to contain errors, `HMM-LP` and `HMM-KF` construct a prediction only when they are sufficiently confident that the viewer's gaze is in tracking state `T`, while `nlp` makes an estimate for all data points.

Second, we observe that `HMM-KF` performed better than `HMM-LP`. This is because the linear dynamic system employed in `HMM-KF` is able to deduce the true motion of an identifiable object in future video, while `HMM-LP` simply assumes linear motion.

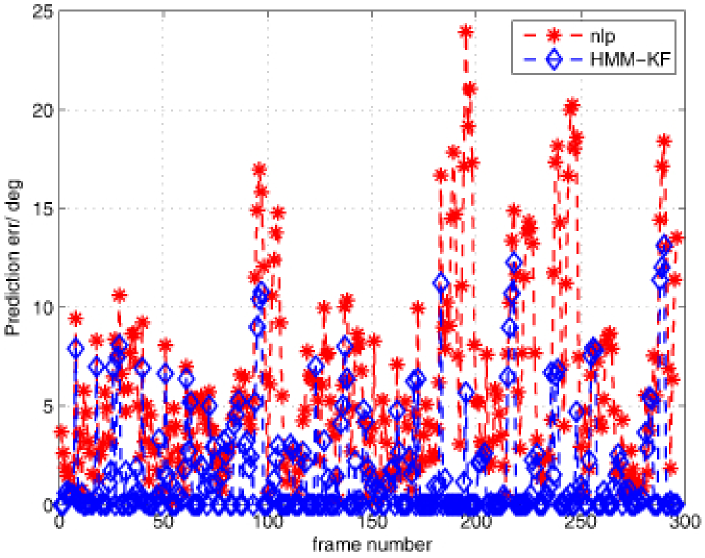
We also plotted the resulting prediction error in Fig. 2-12 against frame number for $\mathsf{RTT} = 200\mathsf{ms}$ for `HMM-KF` and `nlp`. At frame numbers where `HMM-KF` made prediction, we observe that the magnitude of resulting error was in general smaller than `nlp`.

2.6.4 Results for HMM-based Bit Allocation

We next show the achievable bit saving for our gaze-based bit allocation for networked video streaming. We use $\mathsf{QP} = 10$ for a desired reference quality. For our gaze-based scheme (hmm) described in Section 2.5, the average QP outside the ROI is 15, as given by equation (2.22) and (2.23), where $\rho = 5\text{e-}09$. An original scheme (`orig`) assigns $\mathsf{QP} = 10$ for all blocks in a frame. The compressed frame size for the two schemes are given in Fig. 2-13 for CIF sequence `table` (`orig` bit-rate is 300kbps) and SD sequence `captain` (`orig` bit-rate is 800kbps). We see that in frames where the estimated state was tracking state `T`, fewer bits were allocated to non-ROI regions,

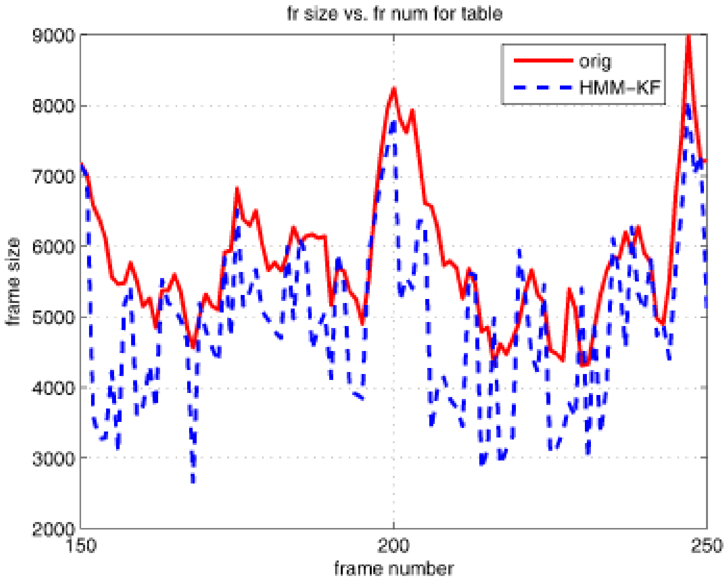


(a) Prediction error vs. frame number for `table`

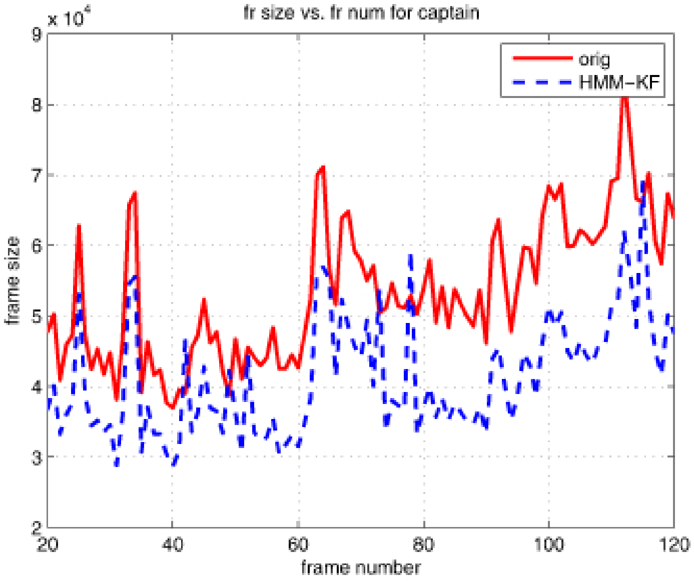


(b) Prediction error vs. frame number for `captain`

Figure 2-12: Prediction Error in degree as function of frame number for different prediction schemes, for `table` and `captain`, respectively, when RTT=200ms.



(a) Frame size vs. frame number for `table`



(b) Frame size vs. frame number for `captain`

Figure 2-13: Frame size as function of frame number for different bit allocation schemes, for `table` and `captain`, respectively, when RTT=200ms.

Table 2.5: Comparing the saliency-based method with the HQ without real-time gaze tracking.

	FQ : HQ		
	Quiet-video	Busy-video	sum
votes	7:16	10:13	17:29
p-value	0.0461	0.5372	0.0698

resulting in bit-rate saving. In particular, we found that hmm achieved 20% and 29% bit saving compared to orig for sequence table and captain, respectively.

2.6.5 Results for subjective testing

Of course, the bit saving must be achieved without significant loss of perceptual quality. To quantify this, we developed a real-time video coding / streaming system for subjective testing, with delay introduced between encoder and decoder to emulate $\mathsf{RTT} = 50ms, 100ms, 150ms, 200ms, 250ms$. A Two Alternative Forced Choice (2AFC) method [50] was used to compare subjective video quality.

We first establish through subjective testing that using ROI-based video encoding without real-time gaze tracking / prediction will often not lead to sufficient perceptual quality. We performed the testing as follows. FQ encodes saliency objects in a video frame in high quality and other regions in low-quality, saving bit-rate. No gaze tracking / prediction is employed. HQ encodes entire video frames as the same high quality, resulting in a higher bit-rate. The subjective result could be seen in Table 2.5.

We see in Table 2.5 that a substantially larger proportion of test subjects preferred HQ over FQ. That means test subjects were able to construe a difference in perceived visual quality between `HQ` and `FQ`. Looking more closely, this perceived difference in visual quality is most pronounced when the video content itself is quiet—steady state probability π_{T} is large.

We can explain the results as follows. It is clear that a pre-encoded ROI video coding scheme can handle gaze behavior of the mean user at best; idiosyncratic gaze behavior by individual users that deviate from the mean user—which happens more often for quiet videos—cannot be handled by offline encoded scheme. In contrast, our real-time gaze-based scheme can fully account for such personal idiosyncrasies, which explains our better subjective experimental results.

Next, to validate our gaze prediction strategy, two videos are randomly selected among the following three: the original HQ scheme `hq`, our proposed gaze-based ROI bit allocation scheme `hmm`, and the naïve linear prediction `nlp`. In each trial, participants looked at two videos back-to-back (with 3 seconds break in-between). Each video lasted for 10 seconds as recommended by ITU-R BT.500 [27]. After these presentations, each participant was asked to indicate which of the two videos looks better (First or Second), regardless of how certain they were of their response. Participants did not know which video was obtained by which kind of method. Full random combinations of two from `hq, hmm, nlp`, using 5 different `RTT`, gave a total of $2 \times 3 \times 5 = 30$ pairs.

The experiment was run in a quiet room with 23 participants (17 males and 6 females, and of age between 21 and 40). All participants had normal or corrected to normal vision. The illumination in the room was in the 300-320 Lux range. Each participant was familiarized with the task before the start of the experiment via a short instruction. During video playback, the viewer's gaze points were tracked and

Table 2.6: Comparing the proposed method with the HQ and NLP method based on the subjective results at 5 different RTTs.

$\mathsf{RTT/ms}$		HMM : HQ	HMM : NLP	HQ : NLP
50		23:23	37:9	40:6
	p-value	1	2.65E-07	1.82E-13
100		24:22	37:9	42:4
	p-value	0.7703	2.65E-07	8.08E-23
150		20:26	39:7	43:3
	p-value	0.3774	8.25E-11	3.36E-32
200		18:28	41:5	42:4
	p-value	0.1352	3.35E-17	8.08E-23
250		13:33	41:5	44:2
	p-value	0.0012	3.35E-17	5.68E-51

sent to the streaming server.

The subjective testing results are shown in Table 2.6, where we indicate the number of responses showing preference for `hq, hmm, nlp` at different RTT values. We used the two-sided chi-square χ^2 test [45] to examine the statistical significance of the results. The null hypothesis is that there is no preference for either two of `HQ, HMM, NLP`. Under this hypothesis, the expected number of votes is 23 for each method. The p-value [45] is also indicated in the table. In experimental sciences, as a rule of thumb, the null hypothesis is rejected when $p < 0.05$. When this happens in Table 2.6, it means that the two methods cannot be considered to have the same subjective quality, since one of them has obtained a statistically significantly higher number of votes, and therefore seems to have better quality.

As seen in Table 2.6, in all of the pairs of `HMM−NLP` and `HQ−NLP`, the p-value is much smaller than 0.05, which indicates that subjects showed a statistically significant preference for our proposed method HMM and HQ. Further, looking across all pairs of HMM−HQ, the results show that participants only noticed significant difference when RTT is larger than 200ms.

Our results clearly shows that our proposed method is always superior to nlp. Furthermore, it can achieve about 29% bit savings compared to HQ with only minor loss of subjective quality.

2.7 Complexity Analysis

2.7.1 Computing Complexity

As noted, the completed system is running in real-time while the video playbacks at 30fps, although the detailed computing overhead is not provided. we believe that it's much smaller than one RTT delay. And from subjective result table 2.6, we can tell that people can even not notice the effect by the delay from RTT.

But it's true that the real-time encoding is quite heavy for any server part. We success to build the system to imply ROI-based encoding method to encode cif-format video at 30fps and SD-format video at 15fps in real-time, and it will be much harder for larger resolution videos. We improved one streaming switch system to avoid the real-time encoding scheme with real-time gaze tracking and prediction strategy, which will be discussed in Section 4.1.1.

Table 2.7: Bit saving on different video types

format	CIF	SD	720P
resolution	352*288	720*576	1280*720
Bit saving	21%	29%	52%

2.7.2 Video Complexity

In this section, we will discuss how would video content types effect the system performance. First, the different video formats will gain different bit saving. From table 2.7, it's obviously that our system will gain better performance on videos those have larger resolutions.

Also as we know, for different videos has their own motions, and objects, to achieve the best prediction in our system, each type of video must has their own HMM parameters, and even for one video, its different scenes also be able to have different corresponding gaze behaviors. We next illustrate through examples how video can be partitioned into segments of roughly stationary gaze statistics using computed Kullback-Leibler (KL) divergence of motion-compensated saliency maps.

For our illustration, we constructed two composite video clips. The first CIF video clip consists of 100-frame of `silent`, plus 100-frame of `table`, plus 100-frame of `silent`. Since we know the visual activities in `silent` and `table` are very different, we know a priori that there is a change in gaze statistics at frame 101 and 201. Similarly, we constructed a second composite SD video clip consisting of 100-frame of captain, plus 100-frame of `group`, plus 50-frame of `captain`.

We first compute motion-compensated saliency maps: after identifying saliency objects in saliency map t and $t+1$, for each corresponding saliency object pair in map t and $t+1$, we relocate the object in map $t+1$ to match the location of the corresponding object in map t. Such relocation process allows easier comparison of saliency characteristics frame-to-frame in terms of gaze statistics, particularly when a salient object is in motion.

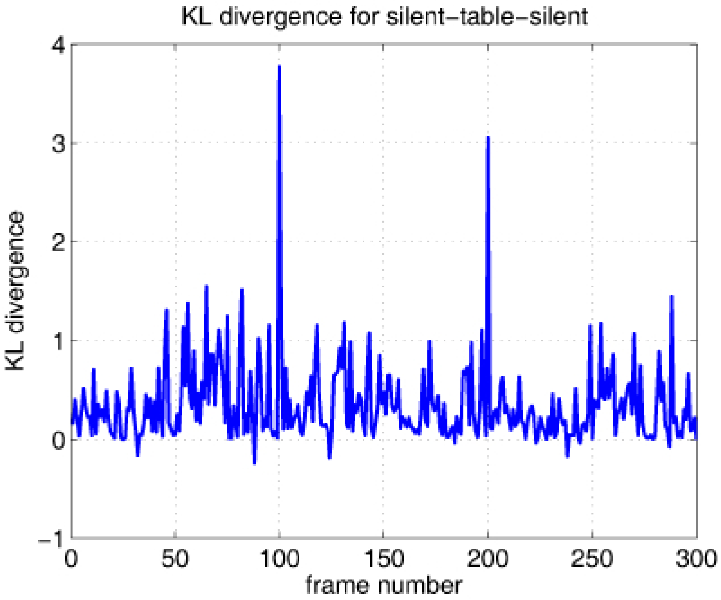
The comparison is achieved treating saliency map ϕ_t at t and motion-compensated saliency map ϕ_{t+1} at $t+1$ as probability distribution functions, and compute the KL divergence as follows:

$$d_{KL}(\phi_t \| \| \phi_{t+1}) = \sum_i \phi_t(i) \log \left(\frac{\phi_t(i)}{\phi_{t+1}(i)} \right) \tag{2.27}$$

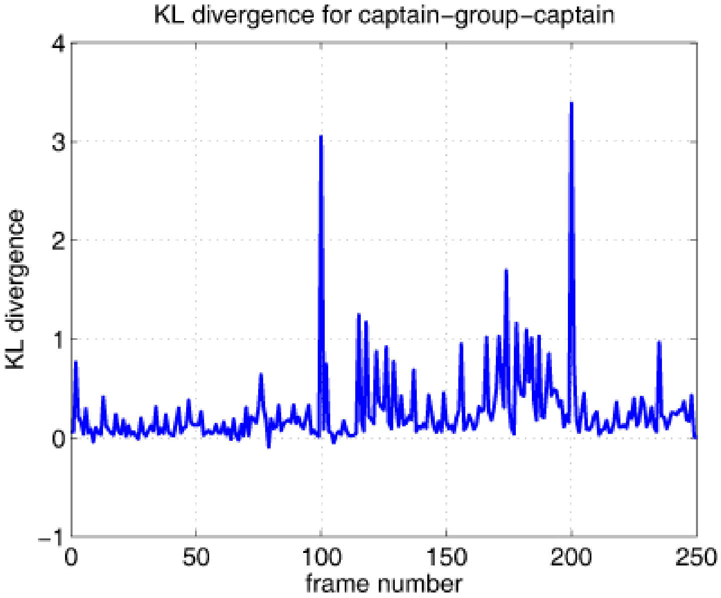
If the computed KL divergence exceeds a certain threshold τ_{KL}, then we declare there is an abrupt change in statistics, and we can partition the video clip into two segments of roughly stationary gaze statistics.

The computed KL divergence for each frame could be seen in Fig. 2-14(a) and (b), respectively, for the two composite sequences. We can clearly see spikes around composition frames 101 and 201, indicating a significant change in gaze statistics. This suggests that KL divergence using motion-compensated saliency maps can be an effective method to partition video into segments of different gaze statistics (even though other methods may also be appropriate).

Further, one proposed method to mearsure the video busyness using saliency map analysis will run at first step to classify the video will be discussed in Section 3.



(a) KL for `silent-table-silent`



(b) KL for `captain-group-captain`

Figure 2-14: KL Divergence as function of frame numbers

2.8 Summary

To improve the efficacy of gaze-based networked systems, in this Chapter, we proposed a hidden Markov model (HMM)-based gaze prediction strategy to predict future gaze locations one round-trip-time (RTT) into the future. The two HMM states correspond to two of human's intrinsic gaze behavioral movements. HMM parameters are derived offline by analyzing the video's visual saliency maps of per-pixel visual attention weights. The most likely HMM state is estimated via the forward algorithm (FA) using real-time collected gaze data. Given an estimated state, a prediction strategy using Kalman filtering 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 up to 29% without noticeable visual quality degradation for RTT as high as 200ms.

Chapter 3

Video Attention Deviation Analysis

3.1 Introduction

As presented in previous section, a viewer's visual attention during video playback is the matching of his eye gaze movement to the changing video content over time. If the gaze movement matches the video content (e.g., follow a rolling soccer ball), then the viewer keeps his visual attention. If the gaze location moves from one video object to another, then the viewer shifts his visual attention. A video that causes a viewer to shift his attention often is a "busy" video. Determination of which video content is busy is an important practical problem; a busy video is difficult for encoder to deploy region of interest (ROI)-based bit allocation, and hard for content provider to insert additional overlays like advertisements, making the video even busier. One way to determine the busyness of video content is to conduct eye gaze experiments with a sizable group of test subjects, but this is time-consuming and cost-ineffective. In this section, we propose an alternative method to determine the busyness of video—formally called video attention deviation (VAD)—by analyzing the spatial visual saliency maps of the video frames across time. We first derive transition probabilities of a Markov model for eye gaze using saliency maps of a number of consecutive frames. We then compute steady state probability of the saccade state in the model—our estimate of VAD. We demonstrate that the computed steady state probability for saccade using saliency map analysis matches that computed using actual gaze traces. Further, our analysis can also be used to strengthen the performance of gaze prediction system presented in Section 2.4.

3.2 Motivation

During playback of a video clip displayed on a reasonably large screen, a viewer sitting at a comfortably close distance from the screen cannot observe all spatial regions simultaneously and clearly in a given video frame. In fact, it has been shown [15] that the ability of a viewer to discern details away from his focal point of visual attention drops off precipitously as a function of the viewing angle. So driven by top-down motivation (e.g., a task in mind) and/or bottom-up stimulus (e.g., low-level features of the visual scene), a viewer typically shifts his visual focal attention from time to time, to study new local spatial regions of interest. This movement is known as saccade in eye gaze literature [14]. Saccade contrasts with another eye gaze movement tracking, where a viewer's gaze point simply follows a identified moving video object, like a rolling soccer ball across the screen. In the latter case, the viewer does not shift his visual attention, but rather, maintains his attention on the same video object.

Thus, one can determine the extent in which a viewer shifts his visual attention by comparing his eye gaze trajectory with the video content being observed [18]. It is apparent that different video contents contain different degrees of bottom-up stimulus, inducing different amount of visual attention shifts from viewers. A video content that induces very few shifts of visual attention from the viewer, like a stationary camera, head-and-shoulders presidential address, is called a "quiet" video. On the contrary, a video content that induces frequent shifts of visual attention, like a dance music video, is called a "busy" video.

Determination of which video contents are busy is an important practical problem. For example, in a region of interest (ROI)-based bit allocation scheme [10, 18, 35], more bits are allocated to the spatial region containing the viewer's current focal attention point, and fewer bits elsewhere, so that the overall bitrate can be reduced without degrading perceptual visual quality. If a video is busy, frequent saccade movements—which are known to be very fast and unpredictable in speed and duration [3]—will limit the effectiveness of such ROI-based allocation scheme [18]. Identified busy videos can then be encoded in more traditional methods to reduce bitrate, e.g., using uniform quantization across the entire frame.

In another example, given an identified busy video, a content provider is limited by the extent in which visual overlays [11] like advertisements can be painted on top. Doing so will make the video even busier, as too many objects compete for viewer's visual attention. For an identified busy video content, advertisements can instead be inserted as full frames temporally, extending the running time of the video.

One straightforward way to determine the busyness of a video content is to conduct eye gaze experiments, using a real-time gaze tracking system [2], with a sizable group of test subjects. This, however, is clearly too time-consuming and cost-ineffective for a large number of video contents. In this chapter, we propose an alternative method to determine the busyness of video—formally called video attention deviation (VAD)—by analyzing the visual saliency maps [26] of individual video frames across time. Saliency maps, gray-scale images that reflect per-pixel visual attention weight in original video frames, are constructed using combinations of low-level neuronal feature maps—e.g., color and intensity contrasts—that have been found to attract attention in humans and monkeys. Saliency maps that are computed for individual video frames describe visual attention deviation spatially. In contrast, we seek to describe visual attention deviation of a video temporally, i.e., how often a viewer will likely shift attention over time.

Our methodology is as follows. First, we derive transition probabilities of a Markov model using saliency maps of a number of consecutive frames. Then, we compute steady state probability of the saccade state in the model, which becomes our estimate of VAD. We demonstrate that the computed steady saccade state using saliency map analysis matches that computed from actual gaze traces for a range of videos with different degrees of busyness. Further, our inter-frame saliency map analysis can also be used to measure the video busyness, which would be much helpful for foreseeing the bit saving gain and related implements with video encoding and compression.

3.3 Saliency Map Analysis

3.3.1 Introduction

A brief introduction has been presented in Section 1.3.3. Visual attention modeling has been focused many research efforts in the last decade. Several computational models to emulate visual attention have been consequently proposed, detecting the locations that attract the eye gaze. Most of the models compute a saliency map that values each pixel according to its visual saliency. While top-down visual saliency modeling is also possible [16], we focus our discussion in bottom-up visual attention process.

Our goal here is not to propose new visual saliency maps, but to use saliency maps, computed using previously established techniques, to derive HMM parameters offline in a computationally efficient way. This motivation is not unlike previous proposals that use saliency maps to resolve uncertainty in gaze estimates [9, 48, 52], except that our derived HMM parameters reflect the temporal aspect of expected gaze behavior, rather than the spatial aspect. In this chapter, we selected methodology in [26] to compute saliency maps, based on a plausible model of bottom-up visual attention. Considering previous comments on performance, this model offers good performance with reasonable computational cost. An existing implementation of the model is available at [4]. We note, however, that our proposed gaze prediction strategy is agnostic to the particular type of saliency model, and thus can be made interoperable to other saliency models such as [16].

3.3.2 Saliency Map Models

The Saliency Map is a topographically arranged map that represents visual saliency of a corresponding visual scene. One of the most severe problems of perception is information overload. Peripheral sensors generate afferent signals more or less continuously and it would be computationally costly to process all this incoming information all the time. Thus, it is important for the nervous system to make decisions on which part of the available information is to be selected for further, more detailed processing, and which parts are to be discarded. Furthermore, the selected stimuli need to be prioritized, with the most relevant being processed first and the less important ones later, thus leading to a sequential treatment of different parts of the visual scene. This selection and ordering process is called selective attention. Among many other functions, attention to a stimulus has been considered necessary for it to be perceived consciously (see Attention and Consciousness and Visual Awareness; but see Koch and Tsuchiya (2007) for a different viewpoint).

What determines which stimuli are selected by the attentional process and which will be discarded. Many interacting factors contribute to this decision. It has proven useful to distinguish between bottom-up and top-down factors. The former are all those that depend only on the instantaneous sensory input, without taking into account the internal state of the organism. Top-down control, on the other hand, does take into account the internal state, such as goals the organisms has at this time, personal history and experiences, etc. A dramatic example of a stimulus that attracts attention using bottom-up mechanisms is a fire-cracker going off suddenly while an example of top-down attention is the focusing onto difficult-to-find food items by an animal that is hungry, ignoring more "salient" stimuli.

Given the difficulty of accurately measuring or even quantifying the internal states of an organism, those aspects of attentional control that are independent of these, i.e., bottom-up attention, are easier to understand than those that are influenced by internal states. Possibly the most influential attempt at understanding bottom-up attention and the underlying neural mechanisms was made by Christof Koch and Shimon Ullman (Koch and Ullman, 1985). They proposed that the different visual features that contribute to attentive selection of a stimulus (color, orientation, movement etc) are combined into one single topographically oriented map, the Saliency map which integrates the normalized information from the individual feature maps into one global measure of conspicuity. In analogy to the center-surround representations of elementary visual features, bottom-up saliency is thus determined by how different a stimulus is from its surround, in many submodalities and at many scales. To quote from Koch and Ullman, 1985 (p. 221), Saliency at a given location is determined primarily by how different this location is from its surround in color, orientation, motion, depth etc.

The saliency map was designed as input to the control mechanism for covert selective attention. Koch and Ullman in 1985 posited that the most salient location (in the sense defined above) in a visual scene would be a good candidate for attentional selection. Once a topographic map of saliency is established, this location is obtained by computing the position of the maximum in this map by a Winner-Take-All mechanism. After the selection is made, suppression of activity at the selected location (which may correspond to the psychophysically observed "inhibition of return" mechanism) leads to selection of the next location at the location of the second-highest value in the saliency map and a succession of these events generates a sequential scan of the visual scene. This role of the saliency map in the control of which locations in the visual scene are attended is close to that of the "master map" postulated in the "Feature Integration Theory" proposed by Treisman and Gelade in 1980.

The Koch and Ullman study was purely conceptual. The first actual implementation of a saliency map was described by Niebur and Koch in 1996. They applied their saliency map model which made use of color, intensity, orientation and motion cues both to simplified visual input (as is typically used in psychophysical experiments) and to complex natural scenes and they demonstrated sequential scanning of the visual scene in order of decreasing salience (see below). Later work refined the model (Itti et al, 1998; Itti and Koch 2001). The source code to compute saliency maps is freely available at http://ilab.usc.edu/toolkit/downloads.shtml.

3.4 Video Attention Deviation Method

3.4.1 Identification of Salient Objects

We first compute visual saliency maps for all video frames using methodology in [26], which is the same as presented at Section 2.4.2.1 As shown in Fig. 2-5 we see an original video frame, and its corresponding computed saliency map. And each individual computed visual saliency map is normalized, so the sum of all saliency values in a map equals to one. Then we find a set of saliency objects in each map. A viewer may also have gaze locations outside of these saliency objects; we assume that such occurrence means the viewer is in the process of switching from one saliency object to another; i.e., he is in saccade state at this frame time. Returning to our earlier example, we see in Fig. 2-6(a) the normalized saliency map with normalized saliency values below threshold τ_s set to zero, leaving only two saliency objects in the map. Correspondingly, we see the saliency objects in Fig. 2-6(b).

Because the computed saliency maps can be noisy, it turns out that finding a single appropriate threshold τ_S a priori that can identify reasonable saliency objects in all saliency maps of a video in time is difficult. To ease the burden of the threshold selection, we perform the following two procedural steps after initial saliency objects are found in a frame. First, if only a single saliency object is found in a frame, we incrementally lower threshold τ_S until a second saliency object is discovered. We do so because, by definition, the probability of a viewer being in saccade state in any frame is non-zero (i.e., there is a non-zero chance of a viewer switching objects of interest in any frame), and having a single saliency object means there are no other objects to switch to. Performing such procedure usually means the size of the original single saliency object increases as threshold τ_S is lowered. This agrees with intuition: the original object remains the main object with the strongest visual attention despite the decrease in threshold.

Second, for each discovered saliency object, we search within a radius r_s of the object's center[1] to check if another object is in the vicinity. If so, we merge the two objects via convex combination. In Fig. 2-7, we see saliency object 1 is within radius r_s of object 2's center, so we merge the two objects into one. The motivation of saliency object merging is the following. A viewer necessarily looks at a group of

[1] An object center (c_x, c_y) is the Cartesian center of the object, where c_x and c_y are the arithmetic means of the x- and y-coordinates of every pixels in the object.

pixels at a time. So if a very small object $o_{t,i}$ (smaller than a circle of radius r_s) is in the vicinity of another object $o_{t,j}$, then the viewer is also looking at object $o_{t,j}$ when observing $o_{t,i}$. Thus it is sensible to merge the two objects.

We can establish correspondence among saliency objects in consecutive frames using motion estimation (ME), commonly used in video compression algorithms [28, 54]. In details, for each block k (we use 8×8 in our experiments) in a saliency object $o_{t,i}$ in saliency map of instant t, we find the most similar block in saliency map of instant $t-1$, i.e. the block with corresponding RGB pixel values in the original video frame $t-1$ that most matches RGB pixel values corresponding to block k in frame t. If the most similar block in saliency map $t-1$ belongs to a saliency object $o_{t-1,j}$, then object $o_{t-1,j}$ in map $t-1$ and object $o_{t,i}$ in map t could potentially be the same object. If a sufficiently large fraction of blocks k's in $o_{t,i}$ map to the same object in $o_{t-1,j}$, then we declare they are the same object. If no such object exists in previous map $t-1$, then we declare object $o_{t,i}$ to be a new object appearing for the first time in map t.

3.4.2 Deriving Transition Probabilities

Having identified saliency objects across frames, we now derive state transition probabilities for our eye-gaze HMM. As an illustrative example, we examine the simple case where there are the minimum two salient objects in consecutive frames t and $t+1$. Denote by $p_{t,1}$ and $p_{t,2}$ the probability that a viewer will fix his gaze in each of the two objects, respectively, in frame i. Similarly, denote by $p_{t+1,1}$ and $p_{t+1,2}$ the corresponding probabilities for frame $t+1$. Let s_t and s_{t+1} be the probabilities that a viewer is in saccade state in frame t and $t+1$. Because we know the volume of visual saliency for each saliency object (sum of computed saliency pixel values within each object) and saccade spatial region (area not covered by saliency objects), we can calculate the relative probability size of objects by comparing their volumes in each frame:

$$\begin{aligned} s_t &= p_{t,1}/\beta_{t,1} = p_{t,2}/\beta_{t,2} \\ s_{t+1} &= p_{t+1,1}/\beta_{t+1,1} = p_{t+1,2}/\beta_{t+1,2} \end{aligned} \tag{3.1}$$

where β's are the scaling factors among objects in each frame.

Further, we know that the sum of probabilities in each frame must equal 1:

$$\begin{aligned} p_{t,1} + p_{t,2} + s_t &= 1 \\ p_{t+1,1} + p_{t+1,2} + s_{t+1} &= 1 \end{aligned} \tag{3.2}$$

Together with (3.1), we can determine the gaze probability of each object in each frame. This is true no matter how many saliency objects are in each frame.

To calculate the state transition probabilities α's, we apply the definition of state transition to the objects of these two frames. We can write the probability $p_{t+1,1}$ of object 1 in frame $t+1$ as the sum of probabilities of objects in previous frame scaled by view transition probabilities α's:

$$p_{t+1,1} = p_{t,1}\ \alpha_{TT} + s_t\ \alpha_{ST} \left(\frac{\beta_{t+1,1}}{\sum_{i=1}^{2} \beta_{t+1,i}} \right) \tag{3.3}$$

Note that the probability $s_t\ \alpha_{ST}$ from state $\mathtt{S}$ to $\mathtt{T}$ must be split between the two objects, according to their relative volumes.

We can write a similar equation for probabilities $p_{t+1,2}$ of moving objects 2 in frame $t+1$. Further, we can similarly write state transition equation for the saccade state as well:

$$s_{t+1} = \alpha_{TS} \sum_{i=1}^{2} p_{t,i} + s_t\ \alpha_{SS} \tag{3.4}$$

Note that we have now three state transition equations for the four unknown α's. In general, one can obtain $k+1$ state transition equations for k saliency objects. In addition, we know the sum of probabilities leaving a state in a HMM must also be one:

$$\begin{aligned} \alpha_{TT} + \alpha_{TS} &= 1 \\ \alpha_{SS} + \alpha_{ST} &= 1 \end{aligned} \tag{3.5}$$

These two linear equations, together with the earlier derived three linear state transition equations, means that we have more equations than unknowns. We hence compute α's as follows. We rewrite each linear equation i with an additional noise term n_i at the end. The set of linear equations becomes:

$$\mathrm{Ca} = \mathrm{b} + \mathrm{n} \tag{3.6}$$

where $\mathrm{a} = [\alpha_{TT}, \alpha_{TS}, \alpha_{SS}, \alpha_{ST}]^T$ is the vector of α's we are seeking, C is the coefficient matrix, b and n are the constant and noise vectors, respectively. It is well known that the a^* that minimizes the noises n in a mean square sense is computed as follows:

$$\mathrm{a}^* = \mathsf{C}^+\mathrm{b} \tag{3.7}$$

where $\mathsf{C}^+ = (\mathsf{C}^\mathsf{T}\mathsf{C})^{-1}\mathsf{C}^\mathsf{T}$ is the Moore-Penrose pseudo-inverse of matrix C.

3.4.3 Generating VAD values

Having computed sets of transition probabilities α's each using different pairs of neighboring saliency maps in time, the transition probabilities for the video is simply the average of the computed sets of transition probabilities. We can then also compute the steady state probabilities π's of the HMM by performing eigen-analysis as typically done in the literature.

3.5 Performance Evaluation

3.5.1 Validation of VAD

To show the potential of our proposed VAD estimation using inter-frame visual saliency map analysis, we used four test sequences as input to USC's visual saliency map calculation software [4]: i) two 300-frame standard MPEG video test sequences, `silent` and `table`, at CIF resolution (352 × 288) and ii) two 250-frame higher resolution video sequences, `captain` and `group`, at SD resolution (720 × 576) and downloadable from IRCCyN Lab website [1]. All videos have 30 frames per second (fps) playback speed. `silent` and `captain` are "quiet" videos with few visual activ-

(a) frame 15 of SD test sequence `captain` (b) frame 199 of SD test sequence `group`

Figure 3-1: Sample frame of SD resolution sequence

ities, while `table` and `group` are “busy” videos with lots of visual activities. Single frames of `captain` and `group` are shown in Fig. 3-1.

We first validate our proposed saliency map analysis discussed in Section 3.3, i.e., whether saccade steady state probability (VAD) derived from saliency map analysis are roughly the same as ones obtained using actual eye-gaze data traces. To obtain ground truth gaze data, a trained user performed multiple viewings of each test sequence, each time continuously recorded his intention of fixation, pursuit or saccade by pressing keys on a keyboard. Using this “ground truth” data, we calculated one set of state transition probabilities in the HMM and then the saccade steady state probability π_s. The saccade steady state probabilities for all test sequences are shown in Table 3.1. Notice that `silent` and `captain` are indeed a quieter video: the saccade steady state probability π_S is much smaller than `table` and `group`.

For comparison, we see that the derived HMM parameters using saliency maps analysis are quite close to the ground truth gaze data trace. In particular, we see

Table 3.1: Comparison of computed VAD values

	π_S
gaze data for Silent (300 frames)	0.063
saliency map analysis for Silent (300 frames)	0.089
gaze data for Table (300 frames)	0.432
saliency map analysis for Table (300 frames)	0.442
gaze data for Captain (250 frames)	0.152
saliency map analysis for Captain (250 frames)	0.181
gaze data for Group (250 frames)	0.439
saliency map analysis for Group (250 frames)	0.457

that the analytical saccade steady state probability π_S for both silent and table are very close to the ground truth trace numbers, even though π_S's for silent and table are very different. This shows accuracy of our proposed saliency map analysis. We also performed the same experiment for the two SD test sequences. We again see very similar numbers between saccade steady state probabilities derived using saliency map analysis and ones obtained using eye-gaze data trace.

Besides, we note that the VAD value is related with the bit saving gained by our gaze prediction system presented at section 2.6.4, while its performance is matching the ground truth. As presented VAD is reflecting the video busyness, the bit saving is smaller while VAD value is larger, which is because large-VAD videos always contain much busy motions, noise informations to attract people switching their interested points between each other, thus the credibility of our gaze prediction strategy will be lower down to ensure perceived visual quality, leading to a smaller bit saving gain. Oppositely, small-VAD videos always have a clear object in its own contents, so people is easily following with the single or less salient objects, thus a large bit saving is achieved.

3.5.2 Logo-insertion application

To show the potential of our proposed VAD estimation using inter-frame visual saliency map analysis, we used four test sequences as input: `Golf` and `Dance In the Woods`, `Captain` and `Group Disorder` at HD resolution (1920×1080). All videos have 30 frames per second (fps) playback speed. `Golf` and `Captain` are "quiet" videos with few visual activities, while `Dance In the Woods` and `Group Disorder` are "busy" videos with lots of visual activities. We performed two experiments to validate the effectiveness of our proposed VAD for logo insertion. The first experiment consists in determining whether our proposed VAD is an appropriate metric for detecting changes in gaze behavior when inserting a logo, compared to other metrics. The second experiment consists in checking whether there actually is a gaze behavior change when inserting a logo.

3.5.2.1 Logo-insertion Experiment 1

This experiment aims at determining whether our proposed VAD is an appropriate metric for detecting changes in gaze behavior when inserting a logo, compared to other ROC and to the correlation-based measure. To generate a gaze behavior change, we inserted a logo with varying opacity values. We expect that the higher the opacity of the logo, the more noticeable and the most likely the gaze behavior change. To achieve this goal, we consider five different opacity values per test sequence (`Golf`, `Dance In the Woods`, `Captain` and `Group Disorder`): 20%, 40%, 60%, 80% and 100%. The logo is inserted in every frame of the test sequence. This leads to twenty new sequences. Visual attention maps are then generated from the new sequences. The gaze behavior change should be detected by using an appropriate metric comparing the visual attention map of the original sequence and that of the logo-inserted video sequence. We used ROC, correlation coefficient and VAD to achieve this comparison. Figures. 3-2 and 3-3 give the results of this comparison when using ROC and correlation-based measure respectively.

3.5.2.2 Logo-insertion Experiment 2

To quantify the metrics and obtain the ground truth of how many percentage would the inserted logo be perceived, we developed experiment system for subjective testing, with introduced four test sequences and logo inserted ones, they all have the same bit-rate.

Our experiment system employs the high accurate gaze-tracking equipment `Tobii X60`, which is calibrated for sampling gaze location at 60 samples per-second. In our experiment, we used three groups of sequences: i) quiet video, ii) busy video and iii) video with median motions, which were grouped by their vad value.

The monitor used for gaze tracking and video experiments measured 46 inches diagonally ($1018\mathtt{mm} \times 573\mathtt{mm}$), with resolution of 1920×1080. Brightness and contrast are set to 30% and 50%, respectively. The distance between a user's head and the center of monitor screen is about $1100\mathtt{mm}$, resulting in a viewing angle of about 53 degrees to the side-edge of the screen, while the traceable degree of X60 is 70 degree.

To measure how many percentage would the inserted logo be perceived, the orig-

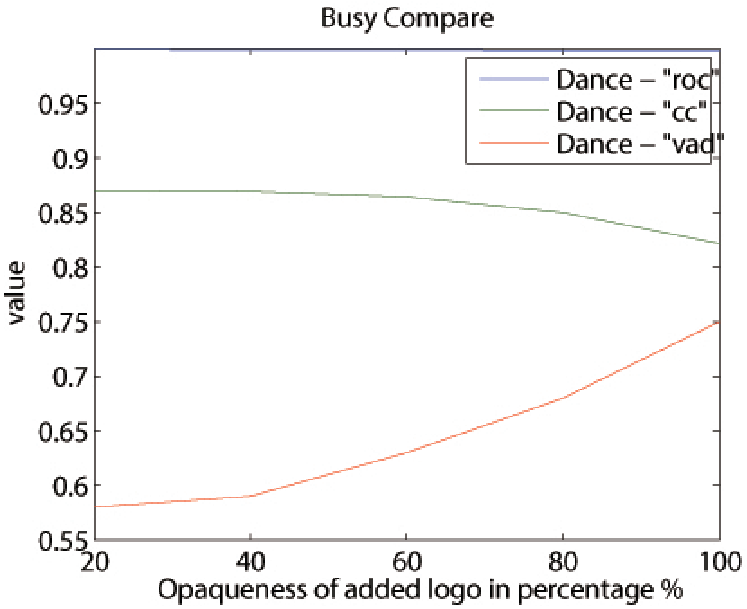


Figure 3-2: Comparison of ROC, CC and VAD depending on the opaqueness of the inserted logo of busy video.

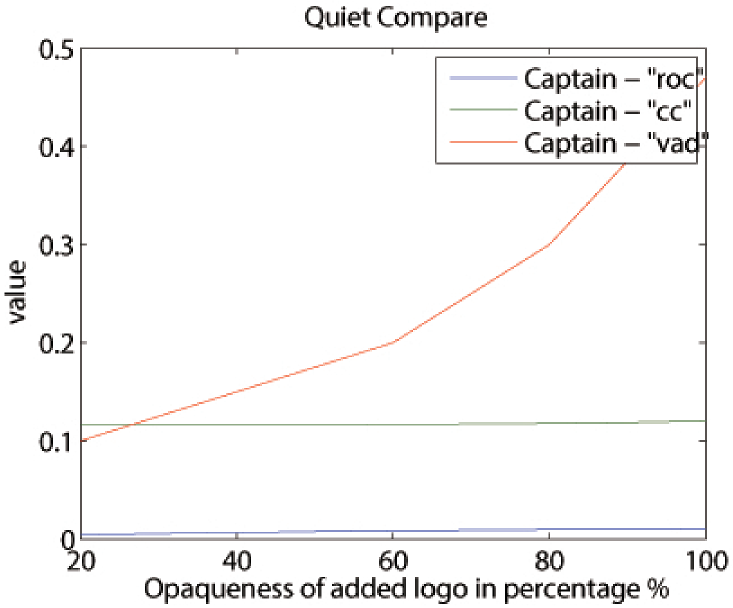


Figure 3-3: Comparison of ROC, CC and VAD depending on the opaqueness of the inserted logo of quiet video.

inal and comparison videos should be prepared carefully. During our subjective experiment, each 8 videos were prepared for each group, as total 24 videos. For logo insertion, we have 3 different logos, and each logo has two candidate locations for different video, while all of them have the similar resolution, to ensure the ratio between logo-area and frame-area should be the same.

And the organization of each test should follow these rules: i) each sequence should, and could only be watched once for each viewer, ii) all 24 sequences should be shown to viewer, with randomly chosen logo insertion, but for all viewers, all options of logo insertions should be watched. and iii) for each two consecutive videos, the same logo with the same location should be avoid.

For each test, the viewer will start with a short introduction and the calibration. He/She will be asked to answer few questions for a reasonable visual acuity examination. During the whole test, there are no leading instructions on video contents, all viewers are required to sit naturally and keep their head stable. Besides a 5-second break is inserted into each two sequences with gray picture showing on the display. And for this subjective evaluation, there were totally 36 people invited.

Figures 3-4 gives the result of this subjective test when showing quiet-busy videos with inserted logo respectively. We notice that the higher VAD value is, the lower inserted logo is perceived. And comparing with Figures 3-5, we notice that our proposed method would be more sensitive on measuring video busyness than traditional metrics.

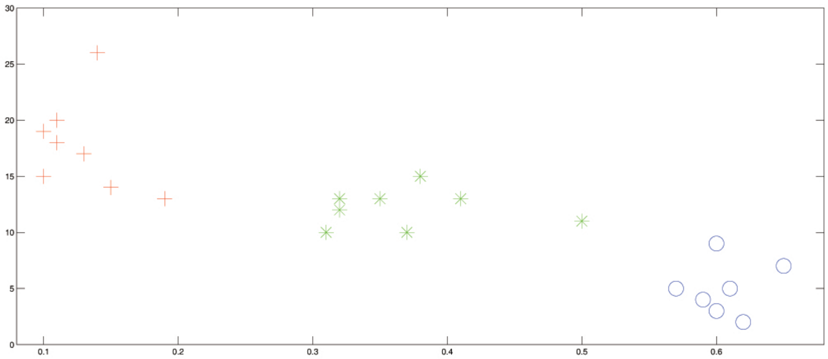


Figure 3-4: Percentage of inserted logo being perceived via video's VAD value.

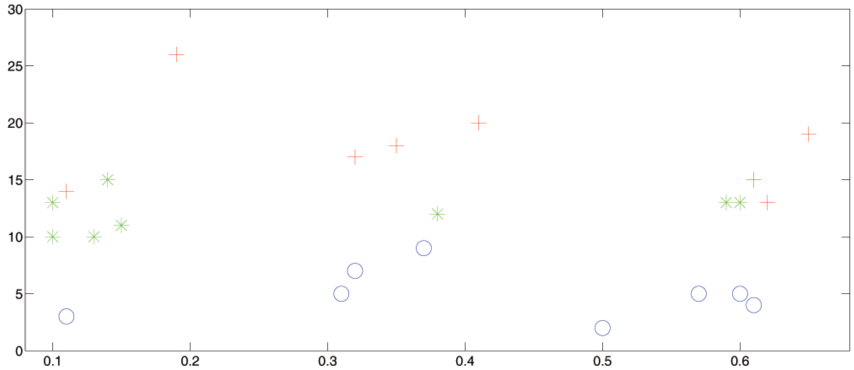


Figure 3-5: Percentage of inserted logo being perceived via video's KLD value.

3.6 Summary

In this chapter, we continued to improve our gaze prediction system by using saliency map analysis to detect scene change, classify the video content for bit saving, also we were able to measure video busyness without time consuming subjective experiment. And our comparison results show that it's much sensitive than other metrics, also the VAD result is matching subjective evaluation by collecting ground truth gaze data. And the most important is that VAD result is matching subjective evaluation, which means it's reflecting human perception while the video is playbacked.

Chapter 4

Conclusion

4.1 Discussion

As presented, the experiments prove that our straight-forward algorithms were able to achieve the better performance than previous work, but there are more problems to be resolved:

1. Our gaze prediction is using single gaze tracker, but there always will be multiple people before the display. Is that possible to improve it for multi-users?

2. The proposed gaze prediction is highly based on hidden markov model, which is very simple and might be not able for variable videos and viewers. Is there any other more suitable prediction strategy?

3. The VAD metric is able to recognize quiet or busy video, but the performance is based on the right saliency map analysis and our proposed HMM. Is there

other ways to produce the more stable metric?

4.1.1 Dual-stream switching frame structure

In section 2.7.1, we talked about the system complexity of ROI bit allocation, and to avoid the real-time encoding, we propose one store-and-playback video streaming system to employs two pre-encoded video streams with the same content in different qualities: HQ stream has all spatial regions encoded in HQ, while MQ stream only has visually salient regions encoded in HQ. DSC frames are inserted periodically every T frames to facilitate stream-switching depending on real-time tracked gaze locations. The system essentially switches to HQ stream when a viewer's gaze travels outside visually salient regions, and switches back to MQ stream when the server is confident that the viewer's gaze will remain in visually salient regions in the foreseeable future.

In theory, it is possible to set T small enough so that zero visual degradation is observed. This is because when human gaze shifts from one object of interest to another—a movement called saccade—the observer cannot perceive visual details until his vision has settled on the new object [36]. Hence if T is small enough that the server can switch from MQ stream to HQ stream before saccade has completed, the observer will always perceive HQ. Doing so would require a very small T, however, which is not practical given the non-negligible overhead of encoding stream-switching DSC frames. Hence, we take the alternative approach of limiting the probability of observing LQ regions to be below an application-specific ϵ instead.

We now describe the frame structure used to facilitate periodic switching between two pre-encoded bitstreams, shown in Fig. 4-1. HQ stream is encoded in HQ for entire

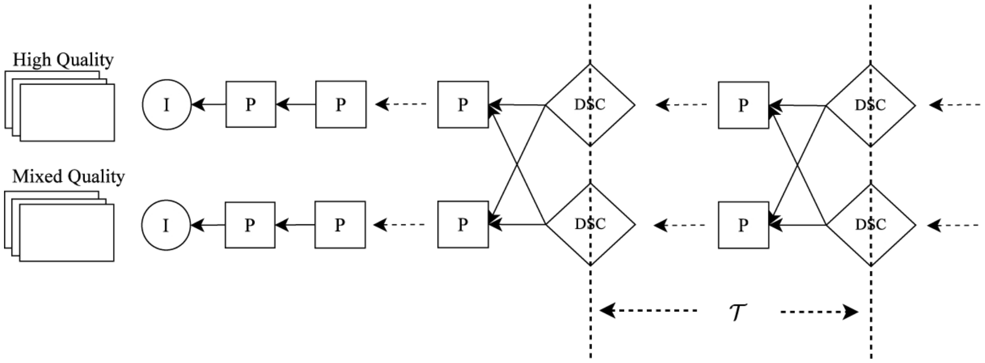


Figure 4-1: Proposed frame structure in gaze-driven video streaming system. I-, P-, and DSC frames are denoted as circles, squares, and diamonds, respectively. DSC frames are inserted every T frames.

video frames. MQ stream is encoded in two quality: spatial regions with per-pixel visual saliency values above a saliency threshold τ (ROI) are encoded in HQ, while the other regions are encoded in LQ. Frames in each stream are encoded in IPPP structure, with DSC frames [12] periodically inserted with period T frames to enable stream-switching at DSC frame boundary. More specifically, each DSC frame W^{q}_{nT} of instant nT, $n \in \mathcal{Z}^{+}$ and $q \in \{\text{HQ}, \text{MQ}\}$, is encoded with two predictor P-frames of previous instant $nT-1$ from the two streams, P^{HQ}_{nT-1} and P^{MQ}_{nT-1}. The reconstruction property of DSC frame guarantees that W^{q}_{nT} can be correctly decoded if any one of the predictor frames is available at decoder buffer as side information. Thus, a client can switch from P^{HQ}_{nT-1} in HQ stream or P^{MQ}_{nT-1} in MQ stream to W^{q}_{nT} in q stream at DSC boundary nT.

4.1.2 Applications of Gaze/Video-based video content adaptation

It's obviously that the proposed video content adaptation using gaze behavior or video content analysis can be implied widely not only in video streaming society. Now the google glass make it possible to track your gaze point and share your life with your friends in real-time. With those more and more popular wearable devices, video content adaptation could be more and more intelligent for daily life.

For example, there's one patient need a operation immediately while the doctor at the scene does not have related experience on it, then he might choose accepting remote instructions during the surgery while transmitting the live operation to other doctors. In this case, the high quality video content is necessary for other doctors to clearly see what's happening during the surgery and to provide correct instructions, besides the delay by video transmission will be quite dangerous for it may lead to inadequate treatment or even death. Our proposed system could solve it by video content adaptation using gaze tracking, gaze prediction and saliency map analysis. Using saliency map analysis and related medical informations, we could lock the lesion parts and make sure them encoded at high quality. Then the gaze prediction strategy will make sure other doctors are able to see exactly how the operation is going on, and ROI based bit allocation could reduce the video size and transmission time.

Furthermore, 4K videos and displays are more and more popular and 4K TV has already shown on Japan's market, so more and more high definition video contents will be needed. Take the live ball-game as an example, like the current World Cup 2014, everyone want to see the live broadcast clearly while they have their own favorite team and players. To detect viewers' interest, and to adapt video content automatically, and to insert different advertisements or logos according to viewers' interest are also capable of our proposed system.

4.2 Conclusion

In this thesis, we first proposed a Hidden Markov Model (HMM)-based gaze prediction strategy to predict future gaze locations one round-trip-time (RTT) into the future for improving the efficacy of gaze-based networked system. The two HMM states correspond to two of human's intrinsic gaze behavioral movements. HMM parameters are derived offline by analyzing the video's visual saliency maps of per-pixel visual attention weights. The most likely HMM state if estimated via the forward algorithm using real-time collected gaze data. Given an estimated state, a prediction strategy using Kalman filtering 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 up to 29% without noticeable visual quality degradation for RTT as high as 200ms.

By the second half of this thesis, we presented our metric VAD using saliency map analysis to detecting scene change and video busyness of given videos. without time consuming subjective experiment. And our comparison results show that it's much sensitive than other metrics, also the VAD result is matching subjective evaluation by collecting ground truth gaze data. And the most important is that VAD result is matching subjective evaluation, which means it's reflecting human perception while the video is playbacked.

4.3 Future Work

Human interactive system would be more and more popular for future life, considering the more and more convenient devices, and more and more powerful services. In this thesis, we are starting with simple method to resolve the transition problems for networked video streaming, using hidden markov model and saliency map analysis.

For the future work, we would improve our low-cost gaze prediction system for multiple users, and consider more accurate model for gaze prediction. Besides, the real-time bit allocation with known ROIs is also difficult, especially for high throughout servers. We also developed some optimized ways to avoid real time encoding while using the real time gaze tracking. Furthermore, we are more interested at VAD application for future video analysis or encoding methods. Because VAD is not only representing the change on pixel/color level, but also reflecting the human reaction when the video was perceived by people. It's matching the bottom-up analysis model with human top-down cognition model together. By implying VAD, we would be able to foresaw or even guide the human vision and perception while producing, encoding or recovering videos.

Bibliography

[1] IRCCyN lab platforms & databases. http://www.irccyn.ec-nantes.fr/ lecallet/platforms.htm. 64, 95

[2] Opengazer: open-source gaze tracker for ordinary webcams. http://www.inference.phy.cam.ac.uk/opengazer/. 20, 21, 28, 33, 46, 64, 86

[3] Saccade. http://en.wikipedia.org/wiki/Saccade. 85

[4] iLab neuromorphic vision C++ toolkit. http://ilab.usc.edu/toolkit/downloads.shtml. 27, 87, 95

[5] C. Bishop. Pattern Recognition and Machine Learning. Springer, 2006. 35, 42, 43, 44, 45, 56, 65, 66

[6] 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. 25

[7] N. Bruce and J. Tsotsos. Saliency, attention, and visual search: An information

theoretic approach. In Journal of Vision, volume 9, no.3, pages 1–24, March 2009. 26

[8] A. Bur and H. Hugli. Optimal cue combination for saliency computation: A comparison with human vision. In Lecture Notes in Computer Science, volume 4528, pages 109–118. Springer Verlag, 2007. 26

[9] J. Chen and Q. Ji. Probabilistic gaze estimation without active personal calibration. In IEEE International Conference on Computer Vision and Pattern Recognition, Troy, NY, June 2011. 23, 27, 87

[10] Z. Chen and C. Guillemot. Perceptually-friendly H.264/AVC video coding. In IEEE International Conference on Image Processing, Cairo, Egypt, November 2009. 20, 24, 33, 60, 62, 85

[11] G. Cheung, W.-T. Tan, B. Shen, and A. Ortega. ECHO: A community video streaming system with interactive visual overlays. In IS&T/SPIE 15th Aunnual Multimedia Computing and Networking (MMCN'08), San Jose, CA, January 2008. 85

[12] N.-M. Cheung, A. Ortega, and G. Cheung. Distributed source coding techniques for interactive multiview video streaming. In 27th Picture Coding Symposium, Chicago, IL, May 2009. 107

[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. 25

[14] A. Duchowski. Eye Tracking Methodology: Theory and Practice. Springer, 2007. 18, 27, 33, 34, 41, 55, 84

[15] 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), volume 3, no.4, December 2007. 18, 20, 32, 84

[16] M. C. et al. Predicting human gaze using low-level saliency combined with face detection. In Neural Information Processing Systems, 2007. 25, 27, 47, 87

[17] R. Faragher. Understanding the basis of the Kalman filter via a simple and intuitive derivation. In IEEE Signal Processing Magazine, volume 29, no.5, pages 128–132, September 2012. 19, 58, 59

[18] Y. Feng, G. Cheung, W. t. Tan, and Y. Ji. Hidden Markov model for eye gaze prediction in networked video streaming. In IEEE International Conference on Multimedia and Expo, Barcelona, Spain, July 2011. 21, 46, 70, 85

[19] Y. Feng, G. Cheung, P. L. Callet, and Y. Ji. Video attention deviation estimation using inter-frame visual saliency map analysis. In IS&T/SPIE Visual Information Processing and Communication Conference, Burlingame, CA, January 2012. 67

[20] S. Floyd and K. Fall. Promoting the use of end-to-end congestion control in the internet. In IEEE/ACM Trans. Networking, August 1999. 19, 32

[21] S. Floyd, M. Handley, J. Padhye, and J. Widmer. Equation-based congestion control for unicast applications. In ACM SIGCOMM, Stockholm, Sweden, August 2000. 19, 32

[22] D. Gao, V. Mahadevan, and N. Vasconcelos. On the plausibility of the discriminant center-surround hypothesis for visual saliency. In Journal of Vision, volume 8, no.7, pages 1–18, March 2008. 26

[23] W. Geisler and J. Perry. A real-time foveated multiresolution system for low-bandwidth video communication. In SPIE Proceedings, vol. 3299, July 1998. 20, 33, 62

[24] L. Itti. Automatic foveation for video compression using a neurobiological model of visual attention. In IEEE Transactions on Image Processing, volume 13, no.10, pages 1304–1318, October 2004. 19

[25] L. Itti and P. Baldi. Bayesian surprise attracts human attention. In Advances in Neural Information Processing Systems, pages 802–817, Cambridge, MA, 2006. MIT Press. 26

[26] L. Itti, C. Koch, and E. Niebur. A model of saliency-based visual attention for rapid scene analysis. In IEEE Transactions on Pattern Analysis and Machine Intelligence, volume 20, no.11, pages 1254–1259, November 1998. 19, 21, 24, 26, 27, 46, 47, 86, 87, 90

[27] ITU-R. Recommendation BT.500-8: Methodology for the subjective assessment of the quality of television pictures. Technical report, ITU, 1998. 76

[28] Video Coding for Low Bitrate Communication. ITU-T Recommendation H.263, February 1998. 39, 51, 65, 92

[29] N. Jayant, J. Johnston, and R. Safranek. Signal compression based on models of human perception. In Proceedings of the IEEE, volume 81, no.10, pages 1385–1422, October 1993. 20, 32, 33

[30] O. Komogortsev and J. Khan. Perceptual multimedia compression based on predictive Kalman filter eye movement modeling. In ACM Multimedia Computing and Networking Conference, San Jose, CA, January 2007. 23

[31] O. Komogortsev and J. Khan. Eye movement prediction by Kalman filter with integrated linear horizontal oculomotor plant mechanical model. In Eye Tracking Research & Applications Symposium, Savannah, GA, March 2008. 23, 24

[32] O. V. Komogortsev and J. Khan. Eye movement prediction by oculomotor plant Kalman filter with brainstem control. In Journal of Control Theory and Applications, volume 7, no.1, January 2009. 23, 24

[33] LC Technologies, Inc. Eyegaze Systems. http://www.eyegaze.com. 17

[34] R. J. Leigh and D. S. Zee. The Neurology of Eye Movements. Oxford University Press, 2006. 34, 55

[35] 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, volume 18, no.1, pages 134–139, January 2008. 20, 24, 33, 85

[36] 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), volume 3, no.7, December 2007. 18, 20, 32, 106

[37] O. L. Meur and P. L. 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. 26

[38] O. L. Meur, P. L. Callet, D. Barba, and D. Thoreau. A coherent computational approach to model the bottom-up visual attention. In IEEE Transactions on Pattern Analysis and Machine Intelligence, volume 28, no.5, pages 802–817, May 2006. 26

[39] O. L. Meur, P. L. Callet, and D. Barba. Predicting visual fixations on video based on low-level visual features. In Vision Research, volume 47, no.19, pages 2483–2498, September 2007. 19, 26

[40] D. K. N. Doulamis, A. Doulamis and S. Kollias. Low bit-rate coding of image sequences using adaptive regions of interest. IEEE Transactions on Circuits and Systems for Video Technology, 8(8):928–34, 1998. 24

[41] A. Oliva, A. Torralba, M. S. Castelhano, and J. M. Henderson. Top-down control of visual attention in object detection. In IEEE International Conference on Image Processing, Barcelona, Spain, September 2003. 26

[42] R. Peters and L. Itti. Computational mechanism for gaze direction in interactive

visual environments. In Proceedings of the Symposium on Eye Tracking Research & Applications, San Diego, CA, March 2006. 24

[43] M. Reale, T. Hung, and L. Yin. Pointing with the eyes: Gaze estimation using a static/active camera system and 3D iris disk model. In IEEE International Conference on Multimedia and Expo, Singapore, July 2010. 17, 23

[44] M. Reale, P. Liu, and Y. Lijun. Using eye gaze, head pose, and facial expression for personalized non-player character interaction. In IEEE International Conference on Computer Vision and Pattern Recognition Workshops, Colorado Springs, CO, June 2011. 18

[45] D. J. Sheskin. Handbook of Parametric and Nonparametric Statistical Procedures. Chapman & Hall/CRC, 2007. 77

[46] A. Sippl, C. Holzmann, D. Zachhuber, and A. Ferscha. Real-time gaze tracking for public displays. In Lecture Notes in Computer Science, volume 6439/2010, pages 167–176, 2010. 17

[47] Y. Sugano, Y. Matsushita, Y. Sato, and H. Koike. An incremental learning method for unconstrained gaze estimation. In European Conference on Computer Vision (ECCV2008), pages 656–667, October 2008. 17, 23

[48] Y. Sugano, Y. Matsushita, and Y. Sato. Calibration-free gaze sensing using saliency maps. In IEEE International Conference on Computer Vision and Pattern Recognition, San Francisco, June 2010. 23, 27, 87

[49] D. Sun, S. Roth, and M. Black. Secrets of optical flow estimation and their principles. In IEEE International Conference on Computer Vision and Pattern Recognition, San Francisco, CA, June 2010. 39

[50] M. Taylor and C. Creelman. PEST: Efficient estimates on probability functions. J. Acoustical Society of America, 41:782–787, 1967. 75

[51] Tobii Technology AB. Eye tacking and eye control for research, communication and integration. http://www.tobii.com. 17

[52] R. Valenti, N. Sebe, and T. Gevers. What are you looking at? improving visual gaze estimation by saliency. In International Journal on Computer Vision, volume DOI 10.1007/s11263-011-0511-6, 2011. 27, 87

[53] M. V. Venkatesh and S. c. S. Cheung. Eye tracking based perceptual image inpainting quality analysis. In IEEE International Conference on Image Processing, Hong Kong, September 2010. 61

[54] T. Wiegand, G. Sullivan, G. Bjontegaard, and A. Luthra. Overview of the H.264/AVC video coding standard. In IEEE Transactions on Circuits and Systems for Video Technology, volume 13, no.7, pages 560–576, July 2003. 39, 51, 92

[55] L. Zhang, M. H. Tong, T. K. Marks, H. Shan, and G. W. Cottrell. SUN: A bayesian framework for saliency using natural statistics. In Journal of Vision, volume 8, no.7, pages 1–20, December 2008. 26

List of my Publications

[1] Yunlong Feng, Gene Cheung, Wai-tian Tan, Patrick Le Callet, Yusheng Ji. Low-cost eye gaze prediction in interactive networked video streaming. IEEE Transactions on Multimedia, vol.15, no.8, pp.1865–1879, December 2013.

Jounal:

[2] Pengfei Wan, Yunlong Feng, Gene Cheung, Ivan V. Bajic, Oscar C. Au. 3-D motion estimation for visual saliency modeling. IEEE Signal Processing Letters, vol.20, no.10, pp.972–975, October 2013.

Refereed Published:

[3] Pengfei Wan, Yunlong Feng, Gene Cheung, Ivan V. Bajic, Oscar C. Au, Yusheng Ji. 3D Motion in Visual Saliency Modeling IEEE International Conference on Acoustics, Speech and signal Processing(ICASSP 2013).

[4] Yunlong Feng, Gene Cheung, Wai-tian Tan, Yusheng Ji. Gaze-driven Video

Streaming With Saliency-based Dual-stream Switching IEEE Visual Communications and Image Processing(VCIP), Nov.2012, doi:10.1109/VCIP.2012.6410732.

[5] Yunlong Feng, Gene Cheung, Patrick Le Callet. Video attention deviation estimation using inter-frame visual saliency map analysis SPIE Visual Information Processing and Communicatoin conference(VIPC 2012).

[6] Yunlong Feng, Gene Cheung, Wai-tian Tan, Yusheng Ji. Hidden Markov Model for Eye Gaze Prediction in Networked Video Streaming IEEE International Conference on Multimedia and Expo(ICME 2011).

Others:

[7] Yunlong Feng, Gene Cheung, Yusheng Ji. QoE-aware gaze-based bit allocation for networked video streaming. IEICE Tech. Report, vol.112, no.476, pp.5-7, CQ2012-84, March 2013.

[8] Yunlong Feng, Gene Cheung, Yusheng Ji. Gaze Prediction using Kalman Filter for Networked Video Streaming. IEICE General Conference, BS-1-14, Gifu, March 2013.

[9] Yunlong Feng, Gene Cheung, Yusheng Ji. Evolution of Eye Movement Classification for Interactive Streaming System. IEICE Society Conference, BS-5-36, Toyama, September 2012.

[10] Zhi Liu, Yunlong Feng, Gene Cheung, Yusheng Ji. Districuted Source Coding

for Interactive Multiview Video Unicast. IEICE General Conference, BS-3-28, Okayama, March 2012.

[11] Yunlong Feng, Zhi Liu, Gene Cheung, Yusheng Ji. Swichable Mix-Quality Frame Structure for Gaze-based Video Streaming. IEICE General Conference, BS-3-27, Okayama, March 2012.

[12] Yunlong Feng, Gene Cheung, Yusheng Ji. Estimating Visual Attention using Inter-Frame Salliency Map Analysis for Gaze-based Video Streaming. Picture Coding Symposium of Japan 2011/Image Media Processing Symposium 2011(PCSJ/IMPS 2011).

[13] Yunlong Feng, Gene Cheung, Wai-Tian Tan, Yusheng Ji. Hiddent Markov Model for Gaze-Tracking in Networked Video Streaming. IEICE General Conference(Meeting Cancelled), March 2011.