To appear in proceedings of Multimedia Computing and Networking (MMCN 07), San Jose, California, Jan 28 – Feb 1.

Perceptual Multimedia Compression Based on Predictive Kalman Filter Eye Movement Modeling

Oleg V. Komogortsev, Javed I. Khan
Department of Computer Science
Kent State University, Kent, OH, USA 44242

ABSTRACT

In this paper we propose an algorithm for predicting a person's perceptual attention focus (PAtF) through the use of a Kalman Filter design of the human visual system. The concept of the PAtF allows significant reduction of the bandwidth of a video stream and computational burden reduction in the case of 3D media creation and transmission. This is possible due to the fact that the human visual system has limited perception capabilities and only 2 degrees out of the total of 180 provide the highest quality of perception. The peripheral image quality can be decreased without a viewer noticing image quality reduction. Multimedia transmission through a network introduces a delay. This delay reduces the benefits of using a PAtF due to the fact that the person's attention area can change drastically during the delay period, thus increasing the probability of peripheral image quality reduction being detected. We have created a framework which uses a Kalman Filter to predict future PAtFs in order to compensate for the delay/lag and to reduce the bandwidth/creation burden of any visual multimedia.

Keywords: Perceptual compression, Kalman filter, real-time communication.

1. INTRODUCTION

The digital world today is moving toward wide-format, high-definition, multimedia transmission. Multimedia streams with resolutions of 4Kx2K are viewed as the new standard in the future of home media [1]. The media with this level of resolution requires gigabits of information to be sent per second, but the significant part of the visual information that is being transmitted is redundant even after it is compressed by modern compression codecs. Human vision offers tremendous additional potential for visual data reduction. The diameter of the eye's highest acuity, the fovea, extends only to 2 degrees. The parafovea, the next highest acuity zone, extends to about 4 to 5 degrees, and acuity drops off sharply beyond that point [2]. This idea is used in perceptual adaptation schemes to achieve an additional reduction in bit-rate and computational burden [3, 4, 5, 6]. Such adaptation methods increase the resolution and the image quality around an eye-fixation point while reducing the image quality in the periphery in accordance with the eye's sensitivity. The instrument that is used to identify the position where a person is looking is called an eye-tracker. Today these devices are expensive, have approximately one degree of tracking accuracy and require a calibration procedure to be operated. There will be quite a few challenges present even after less-expensive, accurate eye-tracking technology is available. Many researchers have investigated the eye acuity degradation from an eye-fixation point [8, 9], however in situations where visual data is transmitted through a network with a high control loop delay or transmission lag, the accuracy of peripheral image degradation becomes much less important. The delay brings uncertainty into the system, due to the fact that the exact location and the size of a PAtF can change significantly during this delay period. In such a case the duration of the delay – particularly a rendering and network delay - plays a critical role in affecting the size of a PAtF, thus increasing the part of the image which requires the highest quality coding. Our paper is the first research, to our knowledge, to provide a Kalman Filter based framework, which predicts the perceptual attention focus, thus compensating for the feedback delay value. We tested our proposed framework with a range of delay values to better understand the impact these values have on the perceptual attention focus prediction and media compression. Practical implementation of this work can be applied to flight simulators, environment teleportation, virtual reality, telemedicine, remote vehicle operation, teleconferencing and any other scenario in which multimedia is involved.

2. PREVIOUS WORK

There have been quite a few studies that investigated various aspects of perceptual compression. The research in this area mainly focused on the study of contrast sensitivity or spatial degradation models around the foveation center and its impact on the perceived loss of quality by viewers [8, 10, 11, 12]. Geisler and Perry [13] presented pyramid coding and used a pointing device to identify the point of focus by a subject. Daly et. al. [14] presented an H.263/MPEG adaptive video compression scheme using face detection and visual eccentricity models. Bandwidth reduction of up to 50% was reported. Many of the early works were inspired by the objective to design good quality display systems [15, 16, 9]. For example, Daly [15] utilized a live eye tracker to determine the maximum frequency and spatial sensitivity for HDTV displays with a fixed observer distance. Lee and Pattichis [6] discussed how to optimally control the bit-rate for an MPEG-4/H.263 stream for foveated encoding. Stelmach and Tam [17] have proposed perceptually pre-encoding video based on the viewing patterns from a group of people. Babcock et. al. [18] investigated various eye movement patterns and foveation placements during different tasks, coming to the conclusion that those placements gravitate toward faces and semantic features of the image. A good summary of current research in the perceptual compression field is presented by Reingold et. al. [19] and Parkhurst et. al. [20]. A few researchers worked on the saliency maps and saccade target estimations in videos and 3D environments, based on the pre-computed image analysis [21, 22]. This saccade target estimation required off-line image processing and eye-data analysis. Some researchers used Kalman Filtering for the human interaction motion modeling [33], head-tracking [27], eye-position estimation during pursuit motion of the eye [34], for smooth-pursuit and saccade detection in the eye-movement analysis [36], eye-movement classification [35].

In our previous work we looked at the perceptual video compression for a single viewer [3]. This work was based on the simple eye-speed analysis. Based on this work we examined a multi-viewer scenario [23]. We researched the case when in which visual content information was extracted from a video source in real-time and applied to a perceptual compression scheme [24]. The purpose of this paper is to create a rigorous, easily extendable mathematically proven to optimal PAtF predictive model which can work in situations with high-feedback delay in real-time.

3. HUMAN VISUAL SYSTEM

Scientists have identified several intricate types of eye movements such as drift, saccade, fixation, smooth pursuit, and nystagmus. Among them, the following two play the most important role in the human visual system. (i) Fixations: - “eye movement which stabilizes the retina over a stationary object of interest” [25]. A human’s eye perceives the highest quality picture during a fixation. The world around us, as we see it, is broken into little fixation spots that our eyes provide for us. The whole picture is assembled later by the brain, but the exact process of this assembly is still an undiscovered mystery. (ii) Saccades: “rapid eye movements used in repositioning the fovea to a new location in the visual environment” [25]. It is widely believed that no vision occurs during saccades.

An eye contrast sensitivity function (CSF) is a visual sensitivity function which allows us to perform perceptual compression of any multimedia in a form of image degradation from a fixation point to the pheriphery. A contrast sensitivity function can be described by the Equation 3.1.1. We adopted this function from the work of Daly and Ribas-Corbera [14]). This function addresses the issue of entropy losses in the visual system. Also, this function accounts for the issues of cones, rods, and ganglion cells distributions.

$$S(x\_pix, y\_pix) = \frac{1}{1 + k_{ECC} \cdot \theta_E(x\_pix, y\_pix)} \qquad (3.1.1)$$

In this equation S is the visual sensitivity as a function of every pixel position on the image (x_pix, y_pix), k_{ECC} is a constant (in this model k_{ECC}=0.24), and θ_E(x_pix,y_pix) is the eccentricity measured in degrees. Within any lossy perceptual multimedia compression scheme, an eye sensitivity function has to be mapped to the spatial and temporal degradation functions of the selected encoding scheme. A visual sensitivity function diagram is presented in Figure 1. The possibility of up to 95% bandwidth reduction was previously reported using eye sensitivity function [32].

4. FEEDBACK DELAY

Feedback delay is the period of time between the time when the eye position is detected by an eye tracker (the device which identifies the current viewer’s eye position) and the moment when a perceptually compressed frame is displayed. This delay should be taken into consideration when using real-time perceptual media compression. This concern is important because a future perceptual attention focus should fall within the highest quality region of an image/video. Only then would a viewer not be able to detect the image spatial degradation used for perceptual coding. It is noteworthy that the properties of visual media transmission might change over time thus increasing or decreasing the feedback delay length. Typical network delays range from 20ms to a few seconds. Saccades can move the eye more than 10-100 degrees during that time, which has the potential of reducing the advantage of designing an accurate high acuity zone within the 2 degrees of an eye-fixation point. Our research focuses on the issue of containing the eye-fixations in a high-quality image area represented by the perceptual attention focus, given a value of the feedback delay.

5. OBJECTIVES

Our main objective was to create a framework that would enable us to make a prediction about a future perceptual attention focus based on the long-delayed and noisy eye position data supplied by an eye-tracker. This type of prediction should be fast enough to allow real-time performance for a PAtF-based multimedia compression system. We build this framework through a Kalman Filter design and we call our framework the *Attention Focus Kalman Filter* (**AFKF**).

6. KALMAN FILTERING

A Kalman filter works with a dynamic system which is modeled by an n-by-1 state vector which is updated through a discrete time equation:

$$x_{k+1} = Ax_k + Bu_k + w_k \tag{6.1.1}$$

In the equation above, **A** is an n-by-n state transition matrix, **B** is an n-by-m optional control input matrix, which relates control vector m-by-1 $\mathbf{u_k}$ to the dynamic system's state $\mathbf{x_k}$. **w** is an n-by-1 process noise vector with covariance $\mathbf{Q_k}$.

Every dynamic system's state has a j-by-1 observation/ measurement vector

$$z_k = Hx_k + v_k \tag{6.1.2}$$

H is a j-by-n observation model matrix which maps the true state into the observed space. **v** is an observation noise j-by-1 vector with covariance $\mathbf{R_k}$.

The Discrete Kalman filter has two distinct phases which are used to compute the next dynamic system's state estimate:

Predict:

project the state vector ahead:

$$\hat{x}_{k+1}^{-} = A\hat{x}_k + Bu_{k+1} \tag{6.1.3}$$

project the error covariance matrix ahead:

$$P_{k+1}^{-} = AP_k A^T + Q_k \tag{6.1.4}$$

The predict phase uses the state estimate information from the past time step to produce an estimate for the future state.

Update:

compute the Kalman gain:

$$K_{k+1} = P_{k+1}^{-} H^T (HP_{k+1}^{-} H^T + R_k)^{-1} \tag{6.1.5}$$

update the estimate of the state vector with a measurement z_k:

$$\hat{x}_{k+1} = \hat{x}_{k+1}^{-} + K_{k+1}(z_{k+1} - H\hat{x}_{k+1}^{-}) \tag{6.1.6}$$

update the error covariance matrix

$$P_{k+1} = (I - K_{k+1}H)P_{k+1}^{-} \tag{6.1.7}$$

Once the future step becomes current, the new measurement information is used to refine the prediction made in the predict phase, which allows the Kalman Filter to come to a more precise dynamic system's state estimate.

The Kalman filter is a recursive estimator that can be used for computing a future estimate of the dynamic system's state that is optimal in the sense of the least square error. Only the estimated state from the previous time step and the new measurements are needed to compute the new estimate for the current dynamic system's state. The majority of real dynamic systems do not exactly fit this model; however, because the Kalman filter is designed to operate in the presence of noise, an approximate fit is often good enough for the filter to be very useful [26, 27].

7. PREDICTING FUTURE ATTENTION FOCUS THROGUH KALMAN FILTERING

7.1 Attention Focus Kalman Filter

We create the Attention Focus Kalman Filter (AFKF) having in mind two key components which represent an eye – the location where the eye is looking and the current velocity of the eye. Thus we model an eye as a system which has two state vectors x_k and y_k.

$$x_k = \begin{bmatrix} \dot{x}_k \\ \ddot{x}_k \end{bmatrix} \qquad y_k = \begin{bmatrix} \dot{y}_k \\ \ddot{y}_k \end{bmatrix} \tag{7.1.1}$$

where $\dot{x}_k$ represents the horizontal and $\dot{y}_k$ represents the vertical eye position on the screen. $\ddot{x}_k$, $\ddot{y}_k$ represent the horizontal and vertical eye-velocity respectively at the time k. The reason why an eye velocity is broken into two components, horizontal and vertical, lies in previous observations presented in our work [3]. The horizontal and vertical components differ significantly from each other, thus shaping the PAtF as an ellipse.

The state transition matrix for both horizontal and vertical states is:

$$A = \begin{bmatrix} 1 & \Delta t \\ 0 & 1 \end{bmatrix} \tag{7.1.2}$$

where Δt is the system's eye-gaze sampling interval.

The observation model matrix for both state vectors is:

$$H = \begin{bmatrix} 1 & 0 \end{bmatrix} \tag{7.1.3}$$

Our system was initialized with the following two state vectors:

$$\hat{x}_0 = \begin{bmatrix} 130 \\ 0 \end{bmatrix} \qquad \hat{y}_0 = \begin{bmatrix} 120 \\ 0 \end{bmatrix} \tag{7.1.4}$$

This initialization was based on the assumption that a viewer looks at the middle of the screen at the very beginning of an experiment and the eye-tracker that we used (ASL 504) has a range of 260 units horizontally and 240 vertically.

Because we could not be sure that a viewer would definitely look at the middle of the screen, the initial error covariance matrix was initialized with ones on its diagonal:

$$P_0 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \tag{7.1.5}$$

The choice of covariance matrixes for process noise Q, and the choice of the measurement noise variance R is discussed in Section 10. We use static Q and R in our model and thus their time indices presented in Equations 6.1.4 and 6.1.5 are dropped. The choice of Q and R affects the performance of the proposed system greatly. In the case when static Q and R are used, both estimation error covariance P_k and the Kalman gain K_k stabilize quickly and become constant. There are no controls on the eye presented in Equation 6.1.1 and thus the **B** and **u** are not used in our model.

7.2 Perceptual Attention Focus Window Design

The goal of our Attention Focus Kalman Filter model is to predict future areas of perceptual attention. Such predictions, as was mentioned before, will allow us to compensate for a system's feedback delay and ensure additional multimedia bandwidth/computational burden reduction based on the idea of the visual sensitivity function presented in the Section 3.

We call the structure which uses the AFKF for a viewer's attention focus prediction the Perceptual Attention Focus Window (W^{PAFW}). The $W^{PAFW}(k)$ is constructed as an ellipse where the major axis equals $T_dV_x(k)$ and the minor axis equals $T_dV_y(k)$. The center coordinates of the $W^{PAFW}(k)$ are $x\_cen^{PAFW}(k)$ and $y\_cen^{PAFW}(k)$. T_d is the system's feedback delay. $V_x(k)$ and $V_y(k)$ are the eye-velocities predicted through the AFKF.

Through Equation 6.1.3 we calculate:

$$\hat{x}_{k+1}^{-} = \begin{bmatrix} \dot{x}_{k+1} \\ \ddot{x}_{k+1} \end{bmatrix} \quad \hat{y}_{k+1}^{-} = \begin{bmatrix} \dot{y}_{k+1} \\ \ddot{y}_{k+1} \end{bmatrix} \tag{7.2.1}$$

The eye velocity components for the W^{PAFW} are taken as absolute values of the velocity values predicted by the AFKF.

$$V_x(k) = | \ddot{x}_{k+1} | \quad V_y(k) = | \ddot{y}_{k+1} | \tag{7.2.3}$$

The center coordinates of the $W^{PAFW}(k)$ are calculated through the previously updated state vectors, which are calculated using Equation 6.1.6:

$$\text{x\_cen}^{\text{PAFW}}(\text{k}) = \hat{x}_k \qquad \text{y\_cen}^{\text{PAFW}}(\text{k}) = \hat{y}_k \tag{7.2.6}$$

A human's eye perceives the highest quality picture during an eye movement called an eye-fixation, described in Section 3. Eye-fixation presents the point of viewer's overt perceptual attention and is used by us to measure the performance of the W^{PAFW} construction algorithm. Thus, due to the delay presented in the system the W^{PAFW} construction is validated through an eye-fixation, which happens at $k + \frac{T_d}{eye\_sample\_rate}$ time interval. The formal algorithm for eye-fixation detection will be presented in section Section 8.2. The W^{PAFW} construction is accomplished in real-time due to fast nature of the Kalman Filter design, which is very important for real-time perceptual multimedia compression. The Perceptual Attention Focus Window construction is presented in the Figure 2.

The intuitive explanation behind the W^{PAFW} construction is based on the fact that the center W^{PAFW} updated eye-gaze coordinates (Equation 6.1.6) and the eye velocity predicted by the AFKF is used to find the size of the area where an eye is going to travel in the future.

7.3 Perceptual Multimedia Compression using Attention Focus Window

The logical structure represented by the W^{PAFW} can be applied to any type of multimedia for bandwidth reduction in the case of the video stream and computational burden reduction in the case of the 3D multimedia content. Once the W^{PAFW} parameters (discussed in Section 7.2) which define the size and location of the W^{PAFW} are calculated, the visual sensitivity function presented by Equation 3.1.1 translates to:

$$S_k(x\_pix, y\_pix) = 1 \quad when \sqrt{\left(\frac{x\_pix - \text{x\_cen}^{\text{PAFW}}(\text{k})}{v_x(k) / v_y(k)} \right)^2 + \left(y\_pix - \text{y\_cen}^{\text{PAFW}}(\text{k}) \right)^2} - V_y(k) \cdot T_d < 0, \; otherwise, \tag{7.3.1}$$

$$= \frac{1}{1 + k_{ECC} \cdot \frac{180}{\pi} \tan^{-1} \left(\frac{\sqrt{\left(\frac{x\_pix - \text{x\_cen}^{\text{PAFW}}(\text{k})}{V_x(k)/V_y(k)} \right)^2 + \left(y\_pix - \text{x\_cen}^{\text{PAFW}}(\text{k}) \right)^2} - V_y(t) \cdot T_d}{VD} \right)}$$

where x_pix, y_pix are coordinates of every pixel of the image presented. VD is the distance between the viewer and the image. Note: all distances need to be converted to the pixel distances for this equation to be true.

Figure 3 presents the sensitivity function specified in 7.3.1. The top, flat area of the surface is created by the W^{PAFW}. The slope is created by the eye sensitivity Equation 3.1.1. The peak point presented by the eye sensitivity function in Figure 1 is the ellipse of the W^{PAFW}. That means that any point inside of the W^{PAFW} has sensitivity equal to 1.

8. EXPERIMENT SETUP

8.1 Equipment

The proposed framework is implemented with an Applied Science Laboratories eye-tracker model 504. ASL 504 has the following characteristics: accuracy - spatial error between true eye position and computed measurement is less than 1 degree; precision - better than 0.5 of a degree. That model of eye-tracker compensates for small head movements within a few inches, so the subject's enforced head stabilization was not required. Nevertheless during the experiments every subject was asked to hold his/her head still. Before running each experiment, the eye-tracking equipment was calibrated for the subject and checked for the accuracy of the calibration, and if one of the calibration points was “off”, then the calibration procedure was repeated for that point. The eye-position-capturing camera worked at the rate of 60Hz. The ASL Eye Tracker User Interface version 1.51 and seventeen point calibration screen were used for the subject calibration procedure.

8.2 Eye Fixation Detection Algorithm

Surprisingly, there is no firm definition for an eye-fixation. It should be noted that from the practical point of view an eye-fixation is less a physiological quantity than a method for categorizing sections of a data stream. Sensible selection of criteria depends upon the experimental goal and the characteristics of the measurement as well as the underlying physiology. There are quite a few different algorithms in the literature for detecting eye-fixations [25], all of which represent logical strategies. Processing the same data with different algorithms or different parameters for a given algorithm, all of which may be justifiable, can easily result in a different number of eye-fixations and different set of eye-fixation start and stop times, and positions. All the facts presented make it important to define the algorithm used for the eye-fixation detection.

The method that we use for evaluation of the W^{PAFW} algorithm falls in the category that Duchowski [25] labels “dwell-time eye-fixation detection” as opposed to “velocity-based saccade detection”.

In our case to identify an eye-fixation, the dwell-time eye-fixation detection algorithm looks through raw eye-data samples provided by the eye-tracker. To "start an eye-fixation" the algorithm looks for a specified period (100 msec.) during which an eye-gaze has a standard deviation of no more than a targeted amount (0.5 degree of the visual angle). To end the eye-fixation, the algorithm looks for a specified number of sequential eye-gaze position samples (3 eye-gaze samples in our case) to be farther than a specified distance (1.5 degrees of the visual angle) from the initial eye-fixation position. The final eye-fixation position is the average position of all data samples between the beginning and end of the fixation. The exception is that any gaze coordinates that were farther than 1.5 degrees of the visual angle from the initial eye-fixation position are not included in the average [28].

8.3 Test Multimedia Content

Human eye movements are highly dependent upon the visual content. Some types of scenes inherently offer more opportunity for compression and some offer less. Any media compression algorithm should continuously analyze the complexity of a scene and provide the best performance possible. Unfortunately, there is no easy or agreed upon means of measuring the complexity of the content. As for our multimedia source we selected MPEG-2 encoded videos. We examined several video clips, each offering different combinations of subjective complexities. In this paper we considered three representative cases. One video clip is apparently “simpler” and one “harder” than the base video “Shamu.” Below are rough subjective complexity descriptions for each case:

Car: This video shows a moving car. It was taken from a security camera view point in a university’s parking lot. The visible size of the car is approximately one fifth of the screen. The car moves slowly. Nothing in the background of this video distracts the subjects’ attention. The snapshot is presented in Figure 4.

Shamu: This video captures a spotlighted, evening performance of Shamu at Sea World. This video consists of several moving objects such as Shamu, the trainer, and the audience. Each of them is moving at different speeds during various periods of time. The snapshot is presented in Figure 5.

Airplanes: This video depicts a performance of the Blue Angels on Lake Erie. The flight formation of supersonic planes changes rapidly as does their flight speeds. The camera movements are rapid zoom and panning. The snapshot is presented in Figure 6.

All three videos have the resolution of 720x480 pixels, are presented with a frame-rate of 30fps, and are between 1 and 2 minutes long. The original video clips are available at our website [29].

8.4 Participants

The evaluation experiments were done for one female subject whose vision was corrected to normal. The subject was not aware of the video content before the experiments and was asked to look at the presented content in any way she wanted. This type of setup is called free-viewing in eye-tracking literature. Test videos were presented on the screen of an 18 inch LCD monitor. The distance between subject’s eyes and the monitor surface was 43 inches. The size of the screen was measured 261x241 in eye-tracker units.

9. EVALUATION PARAMETERS

We selected two parameters – *average eye-fixation containmen*t and *average perceptual coverage* - to evaluate the correctness of the W^{PAFW} construction algorithm

9.1 Eye-gaze containment

An eye-fixation represents a point of viewer's overt attention, thus we measure the correctness of the W^{PAFW} by what we call *average eye-fixation containment* (**AEFC**). The AEFC is represented by a percentage of the eye-fixation samples contained within a W^{PAFW} over N eye-fixation samples.

$$\xi_{W^{PAFW}} = \frac{100}{N}\sum_{k=1}^{N} FIX^{W^{PAFW}(k)}\left(k + \frac{T_d}{eye\_sample\_rate}\right) \qquad (9.1.1)$$

where T_d is a system's feedback delay measured in seconds and the eye_sample_rate is the amount of eye-gaze-position samples the eye-tracker scans every second. Variable $FIX^{W^{PAFW}(k)}(k + \frac{T_d}{eye\_sample\_rate})$ equals 1 or 0. It equals one in the case if the eye-fixation, which happens at $k + \frac{T_d}{eye\_sample\_rate}$, is contained inside of the $W^{PAFW(k)}$ window. N is the number of corresponding eye-fixation samples that the AEFC is measured over. It is worth noting here that eye-fixations represent only 70%-90% of the eye-trace created by the raw eye-positions received from the eye-tracker [2, 25], thus N represents only eye-position samples belonging to the eye-fixations.

Examples in which eye-fixations are contained and missed by the W^{PAFW} are presented in Figures 4-6.

9.2 Perceptual Coverage

We have defined a second performance quantity called the *average perceptual coverage* (**APC**). The APC is a percentage of the image (video frame) covered by a W^{PAFW}. The average perceptual coverage is given by the equation (Δ for area or volume):

$$\chi_{W^{PAFW}} = \frac{100}{M}\sum_{k=1}^{M}\frac{\Delta(W^{PAFW}(k)\cap F)}{\Delta F} \qquad (9.1.2)$$

where F represents the size of the image or visual frame. M is the number of eye-gaze samples that the APC is measured over. Here M represents all eye samples collected during every test video. The diagram of the APC is presented in Figure 7.

10. EXPERIMENT RESULTS

The original selection of values for R and Q matrixes presented in Equations 6.1.4 and 6.1.5 is justified as follows. The standard deviation for the instrument noise relates to the accuracy of the eye-tracker equipment and is bounded by one degree of the visual angle, thus making the standard deviation of measurement noise 14 eye-tracker units $\sigma_v = 14$.

$$\mathrm{R} = \sigma_v^2 \qquad (10.1.1)$$

The standard deviation of the process noise, in our case the noise inside of the eye, has to do with three eye-sub-movements during and an eye-fixation: drift, small involuntary saccades and tremor [30]. Among those three, involuntary saccades have the highest amplitude - around half of the visual angle thus making the covariance matrix for eye noise:

$$Q = \begin{pmatrix} \sigma_w^2 & 0 \\ 0 & \sigma_w^2 \end{pmatrix} \qquad (10.1.1)$$

where $\sigma_w = 7$ in eye-tracker units.

The experiment results which evaluate the performance of the W^{PAFW} are presented in Figure 8 and Figure 9. Our framework had the best performance results for the **<u>Car</u>** video in both low delay and high delay scenarios. The average perceptual coverage was approximately 2.5% while the average eye-fixation containment was 65% in the case of the 166 msec. delay. The 1 sec. delay scenario gave an AFC of 81% while maintaining an APC at 36%.

The **Shamu** performance gave the AFC of 56%, while covering 7.7% of the video image in the case of the 166 msec. delay. The AFC was 70% with an APC of 39% for a 1 sec. delay scenario. From these numbers it is possible to see that our algorithm's performance decreased, probably due to the fact that the viewer had different objects to concentrate on – Shamu, the trainer and the crowd, while in the "car" video there is one main moving object.

Airplanes had the least amount of eye-fixation containment by the W^{PAFW} in the 166 msec. delay scenario containing 44% of the eye-fixations while having the coverage at approximately 4%. The 1 sec. delay scenario yielded 59% of the AFC and 36% of the frame coverage. The challenging part about this video is that it is quite "jerky"; the camera struggles to keep one of the planes in focus and this fact takes a toll on the performance of the algorithm.

Feedback delay performance: as it is possible to see from the figures the W^{PAFW} was able to contain more eye-fixations in the 1 sec. delay scenario - 59% to 81% - rather than in the 166 msec. delay scenario – 44% to 65%. The average perceptual coverage was low for the 166 msec. delay – 2.5% to 7.7% - while for the 1 sec. delay case it was 36% to 69%.

Compression: The actual amount of compression and computational burden reduction when using the W^{PAFW} provided by our framework will depend on two parameters: the size of the area which requires high quality coding (the perceptual attention window coverage) and visual degradation of the periphery. This compression is achieved by mapping the visual sensitivity function presented in Equation 7.3.1. Obviously the scenarios with the lowest APC will provide the highest amount of compression.

10.1 Tuning up Measurement Noise Covariance Matrix

We wanted to see if it was possible to improve the performance of our framework and increase the eye-fixation containment. Using MATLAB we found values for $0 \le \sigma_v \le 1000$ - the measurement noise standard deviation, presented in Equation 10.1.1 which provided the maximum AFC for each delay value and the test video we have considered. The standard deviation values which provided the highest average eye-fixation containment are given in the Table 1. σ_v values are presented in eye-tracker units. The σ_w values remained fixed and equal to 7.

Table 1. Optimal standard deviation values for measurement noise.

Feedback delay	Test video	σ_v
166 msec	"Car"	3
166 msec	"Shamu"	5
166 msec	"Airplanes"	6
1sec	"Car"	12
1sec	"Shamu"	19
1sec	"Airplanes"	28

The AEFC and APC results are presented in Figure 10 and Figure 11. In Figure 10 the AEFC values have increased slightly. Mostly it is visible for the scenario with a short feedback delay – in this case the AEFC rose by 5%. σ_v values were approximately two times smaller than the original standard deviation values presented in Equation 10.1.1. In Figure 11 the average perceptual coverage was doubled for all three test videos for the 166 msec. delay scenario. The question of whether a small increase in the AEFC justifies such an increase in the APC for a small delay scenario will depend on the choice of the application that uses the perceptual attention focus. Plus finding a better choice of σ_v requires an off-line analysis and thus it is not usable in real-time applications.

The σ_v selection for the 1 sec. feedback delay scenario yielded a very small increase in the AEFC – not more than 1.5% for all test videos. But the remarkable result was that for two test videos - "Shamu" and "Airplanes" - the average perceptual coverage decreased by up to 5% while the corresponding AEFC values became higher or remained at the same level.

As a conclusion to this section we would like to state that the choice of a covariance matrix - or the measurement noise in the case of our framework - does affect the AEFC and APC values. Depending on the application which uses the concept of perceptual attention focus it is possible to reselect σ_v which gives a higher AEFC if it is important to "hide" as much of the image degradation as possible or decrease the APC for increased bandwidth and computational savings.

11. DISCUSSION AND FUTURE WORK

It is a valid question to ask how "bad" it is for a system which uses the concept of a PAtF to miss an eye-fixation. The results might vary depending on how far the missed eye-fixation is from the W^{PAFW} boundary and on what mapping of the eye sensitivity function presented in Equation 7.3.1 is used for a particular choice of media. If a viewer notices the "blurred" effect and is unable to see a specific detail on the picture, the viewer can fixate his eyes on the point in question and the system will stabilize itself placing the W^{PAFW} on the spot under attention. This stabilization will happen at least in the time equal to the feedback delay value. If many fixations are missed then the application will require a significant amount of "stabilizations" from the viewer and such a viewing experience might be tiring or unacceptable.

As a part of the future work we would like to look into improving the proposed AFKF model. At the moment the AFKF framework takes into consideration only two dynamic eye states: eye-position and eye-velocity. We feel that incorporating additional eye-states may improve the performance of the AFKF model. As we can see from Section 10 of this paper, media content does affect the AEFC and APC values thus making real-time media content analysis a valuable addition to the PAtF concept [24]. The AFKF design can be possibly improved through a choice of dynamic covariance matrixes Q and R for Equations 6.1.4 and 6.1.5. This might prove practical in situations in which the feedback delay is low and more recent information about the latest eye-movement type is available.

It should be pointed out that there is an additional challenge for an eye-movement type detection and interpretation which is presented by the nature of a real-time multimedia compression system which uses a PAtF for additional compression. For example there is an uneven delay variation in the video transcoding mechanism because different types of video frames take various amounts of time to encode and process. When network jitter is present in the system, raw eye-gaze position samples start arriving at the transcoder unevenly, thus making the interpretation of eye-movement types even more difficult. Plus the transmission delay/lag (feedback delay) can be an order of magnitude larger than the duration of a basic eye movement (saccade or fixation). Thus by the time a current viewer's eye movement type is identified by the transcoder, that type of eye movement might be effectively over and useless for the prediction mechanism.

12. CONCLUSION

The use of perceptual attention focus for additional multimedia compression is a promising area. PAtF techniques will be critical in order to achieve the transcoding/compression ratio needed in emerging applications. The eye-tracker-based encoding might be especially viable in several scenarios such as the following: remote vehicles control and operation, remote surgery assistance, virtual reality teleporting and in applications where eye-gaze is used as an input or content evaluation tool. In this paper we have proposed a mechanism of how a viewer's perceptual attention focus area can be predicted in a dynamic environment with the help of the Kalman Filter and Perceptual Attention Focus Window design. We selected the Kalman Filter as a flexible and extendable framework which is designed to work in noisy environments. The results presented in this paper show that it is possible to predict up to 80% of a future viewer's through this framework in scenarios of the high transmission delays and have additional compression performed on the visual media presented.

Nevertheless, a number of formidable technical challenges still remain before all of the characteristics of the human eye can be exploited to the advantage of engineering. The framework proposed in this paper shows some interesting results through the experiments we conducted. In particular the feedback delay in the control loop (in a network, in media encoding, or even the delay within the eye-tracker) can be compensated for by a predicted high-quality coded area (W^{PAFW}) and depending on the delay value the W^{PAFW} can be many times larger than the para-fovea area studied by visual sensitivity function researchers. We suspect, in the case of live coding, a simple approximation of eye sensitivity function will bring the same performance results as a sophisticated one.

The important aspect of the research conducted in this paper is that it is media independent. Many of the point-gaze based researchers deeply integrate foveation schemes with the media. In contrast, we propose a perceptual attention focus window as a virtual area superposed on the rendering plane of any visual media. If a W^{PAFW} contains all eye fixations, then, in theory, the outer regions can be coded with lesser bits without any perceivable loss of quality. Once the window is obtained, then the actual fovea-matched encoding can be performed in numerous media specific ways with various computational-effort/quality/rate trade-off efficiencies. Our framework is capable of reducing the bit-rate of an already compressed video stream by more than 50%.

13. ACKNOWLEDGMENTS

The work had being funded by DARPA Research Grant F30602-99-1-0515. We would like to express special thanks to Sue Peti for help in preparation of this manuscript.

14. REFERENCES

1. Enami, K. Future of Home Media, In Proceedings of the 13th ACM International conference on Multimedia (ACM MM 05), Singapore, 2005, 1-1.
2. Irwin, D. E. Visual Memory Within and Across Fixations. In Eye movements and Visual Cognition: Scene Preparation And Reading, K. Rayner, Ed. Springer-Verlag, New-York, NY,1992, pp. 146-165. Springer Series in Neuropsychology.
3. Komogortsev O., Khan J. Predictive Perceptual Compression for Real Time Video Communication. In Proceedings of the 12th ACM International conference on Multimedia (ACM MM 04), New York, NY, 2004, 220-227.
4. Murphy, H., Duchowski, A. T. Gaze-contingent level of detail rendering. In EuroGraphics 2001, EuroGraphics Association.
5. Kortum, P., Geisler, W. S., Implementation of a Foveated Image Coding System for Image Bandwidth Reduction. In Proc. SPIE Vol. 2657, 1996, 350-360.
6. Lee, S., Pattichis, M., Bovok, A. Foveated Video Compression with Optimal Rate Control. In IEEE Transaction of Image Processing, V. 10, n.7, July 2001, pp-977-992.
7. Parkhurst, D. IPRIZE. at http://hcvl.hci.iastate.edu/IPRIZE/ last visited April 14th, 2006.
8. Kuyel T., Geisler W., and Ghosh J., “Retinally reconstructed images (RRIs): digital images having a resolution match with the human eye,” in Proc. SPIE Vol. 3299, 1998, 603-614.
9. Loschky, L. C. and McConkie G. W. User performance with gaze contingent multiresolutional displays. In *Proceedings of the symposium on Eye tracking research & applications 2000 (ETRA 00)*, (November 2000) pp. 97-103.
10. Duchowski, A. T. Acuity-Matching Resolution Degradation Through Wavelet Coefficient Scaling. In IEEE Transactions on Image Processing 9 (8), 2000, 1437-1440.
11. Duchowski, A. T. and McCormick, B. H. “Preattentive considerations for gaze-contingent image processing,” in Proc. SPIE Vol. 2411, 1995, 128-139.
12. Bergstrom P., Eye-movement Controlled Image Coding, PhD dissertation, Electrical Engineering, Linkoping University, Linkoping, Sweden, (2003).
13. Geisler W. S. and Perry J. S. Real-time foveated multiresolution system for low-bandwidth video communication. In Proc. SPIE Vol. 3299, 1998, 294-305.
14. Daly S., Matthews K. and Ribas-Corbera J. As Plain as the Noise on Your Face: Adaptive Video Compression Using Face Detection and Visual Eccentricity Models. In Journal of Electronic Imaging V. 10 (01), 2001, 30-46.
15. Daly S. Engineering observations from spatiovelocity and spatiotemporal visual models. In Proc. SPIE Vol. 3299, 1998, 180-191.
16. Duchowski A. T. and McCormick B. H. Gaze-contingent video resolution degradation In Proc. SPIE Vol. 3299, 1998, 318-329.
17. Stelmach L. B., and Tam W. J. Processing image sequences based on eye movements. In Proc. SPIE Vol. 2179, 1994, 90-98.
18. Babcock J.S., Pelz J.B., and Fairchild M.D. Eye tracking observers during color image evaluation tasks. In Proc. SPIE Vol. 5007, 2003, 218-230.
19. Reingold E. M., Loschky L.C., McConkie G. W., and Stampe D. M. Gaze-Contingent Multi-Resolutional Displays: An Integrative Review. In Human Factors, 45(2), 2003, 307-328.
20. Parkhurst D. J., and Niebur E. Variable Resolution Displays: A Theoretical, Practical, and Behavioral Evaluation. In Human Factors, 44(4), 2002, 611-629.
21. Carmy, R. and Itti L. Casual Saliency Effects During Natural Vision. In *Proceedings of the symposium on Eye tracking research & applications 2006 (ETRA 06)*, (March 2006), 11-18.
22. Peters, R. and Itti L. Computational mechanism for gaze direction in interactive visual environments. In *Proceedings of the symposium on Eye tracking research & applications 2006 (ETRA 06)*, (March 2006), 27-32.

23. Komogortsev O., Khan J. Perceptual Attention Focus Prediction for Multiple Viewers in Case of Multimedia Perceptual Compression with Feedback Delay. In *Proceedings of the symposium on Eye tracking research & applications 2006 (ETRA 06)*, (March 2006), 101-108.
24. Khan J., Komogortsev O. A Hybrid Scheme for Perceptual Object Window Design with Joint Scene Analysis and Eye-Gaze Tracking for Media Encoding based on Perceptual Attention. In Journal of Electronic Imaging 15(02), April 2006, pp. 1-12.
25. Duchowski A. T. Eye Tracking Methodology: Theory and Practic. Springer-Verlag, London, UK, (2003).
26. Brown R. and Hwang P. Introduction to Random Signals and Applied Kalman Filtering. 3rd ed., New York: John Wiley and Sons, 1997.
27. Foxlin E. Inertial Head-Tracker Sensor Fusion by a Complementary Separate-Bias Kalman Filter. In Proc of VRAIS '96, 1996, 185-195.
28. ASL Laboratories. Eyenal (Eye-Analysis) software Manual Windows version for use with ASL Series 5000 and ETS-PC Eye Tracking Systems. Copyright c 2001, by Applied Science Group, Inc.
29. Komogortsev, O., Khan, J. Perceptual Multimedia Compression Based on the Predictive Kalman Filter Eye Movement Modeling Video Set. At *www.cs.kent.edu/~okomogor/SPIE06VideoSet.htm*.
30. Yarbus L., Eye Movements and Vision, Institute for Problems of Information Transmission Academy of Sciences of the USSR, Moscow (1967).
31. Khan, J., Gu, Q., and Zaghal, R. Symbiotic Video Streaming by Transport Feedback based Quality rate Selection In Proceedings of the 12th IEEE International Packet Video Workshop 2002, Pittsburg, PA, April 2002.
32. Kortum, P. and Geisler, W. S. Implementation of a foveated image coding system for image bandwidth reduction. In Proc. SPIE Vol. 2657, 1996, 350-360.
33. Kohler M. Using the Kalman filter to track human interactive motion – Modelling and initialization of the Kalman filter for translational motion. Technical Report 629, Informatik VII, University of Dortmund, January 1997.
34. Abd-AlmageedW, Fadali MS, Bebis G (2002) A non-intrusive Kalman filter-based tracker for pursuit eye movement. In Proceedings of the 2002 American Control conference, Alaska, 2000.
35. Grindinger, T. Eye Movement Analysis and Prediction with the Kalman Filter, Masters thesis, Computer Science, Clemson University, Clemson, SC, USA, August 2006.
36. Sauter D, Martin BJ, Di Renzo N, Vomscheid C. Analysis of eyetracking movements using innovations generated by a Kalman filter. *Med Biol Eng Comput.* 1991;29:63–69.

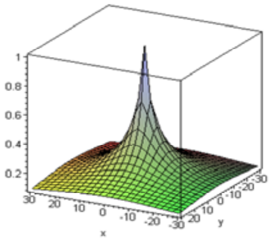

Figure 1: Visual Sensitivity for an eye-fixation function.

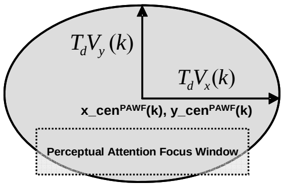
$T_d V_y(k)$
$T_d V_x(k)$
x_cen^{PAWF}(k), y_cen^{PAWF}(k)
Perceptual Attention Focus Window

Figure 2: Perceptual Attention Focus Window diagram.

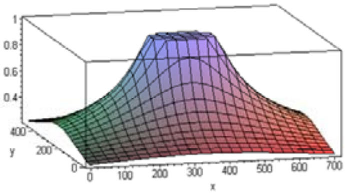

Figure 3: Visual Sensitivity function with perceptual attention focus window.

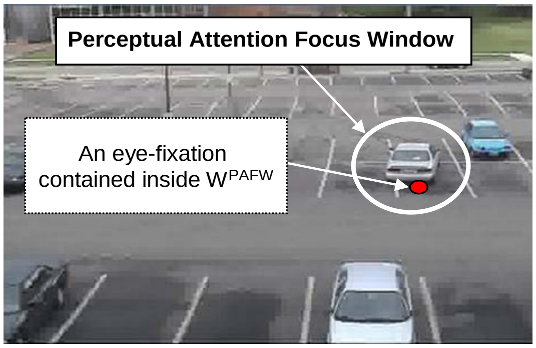
Perceptual Attention Focus Window
An eye-fixation contained inside W^{PAFW}

Figure 4. "Car" video. The frame is compressed from 10Mb/s to 1Mb/s based on the PAtF concept. The eye-fixation is contained inside of the W^{PAFW}.

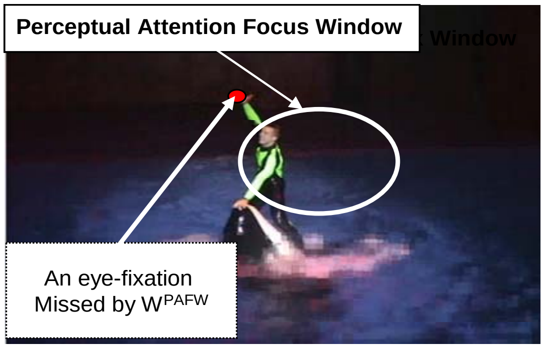
Perceptual Attention Focus Window
Window
An eye-fixation Missed by W^{PAFW}

Figure 5. "Shamu" video. The frame is compressed from 10Mb/s to 1Mb/s based on the PAtF concept The eye-fixation is missed by the W^{PAFW}.

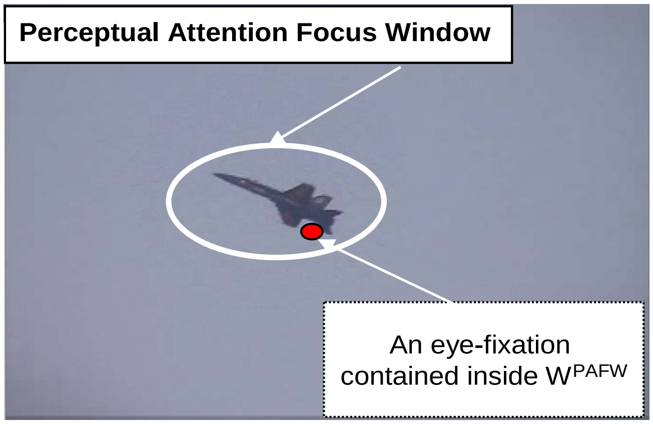
Perceptual Attention Focus Window
An eye-fixation contained inside W^PAFW

Figure 6. “Airplanes” video. The frame is compressed from 10Mb/s to 1Mb/s based on the PAtF concept The eye-fixation is contained inside of the W^{PAFW}.

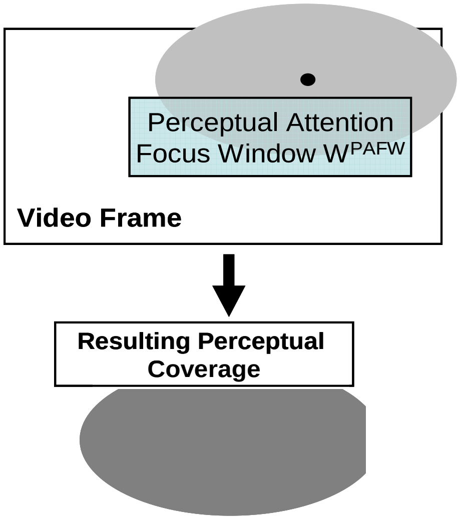
Perceptual Attention Focus Window W^PAFW
Video Frame
Resulting Perceptual Coverage

Figure 7. Perceptual coverage diagram.

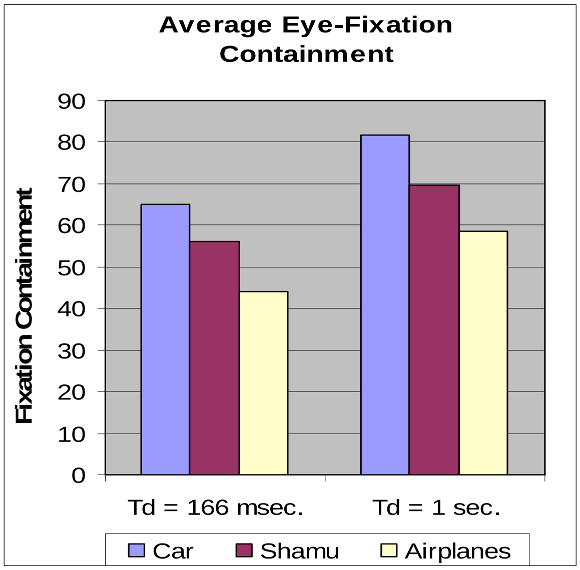

Figure 8. Average Eye-Fixation Containment for Perceptual Attention Focus Window. Original noise covariance matrixes are used.

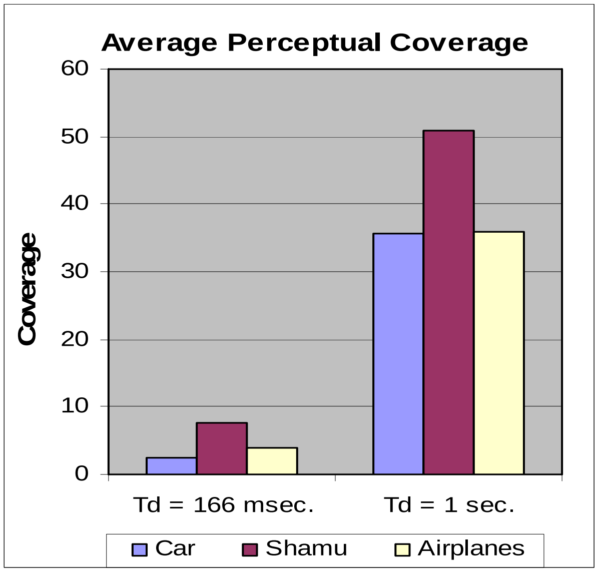

Figure 9. Average Perceptual Coverage for Perceptual Attention Focus Window. Original noise covariance matrixes are used.

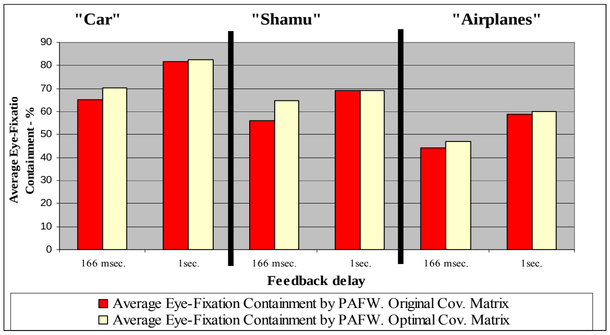

Figure 10. Average Eye-Fixation Containment for W^{PAFW}. Optimal covariance matrix provides highest AEFC values.

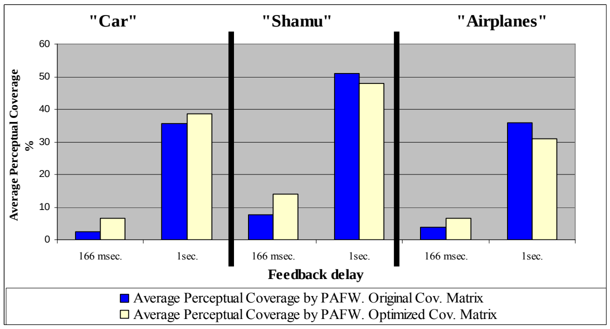

Figure 11. Average Perceptual Coverage for Perceptual Attention Focus Window. Original vs. optimal covariance matrixes are used.