Published in SPIE Journal of Electronic Imaging
15(02), 023018, April 2006

A Hybrid Scheme for Perceptual Object Window Design with Joint Scene Analysis and Eye-Gaze Tracking for Media Encoding based on Perceptual Attention

Javed Khan and Oleg Komogortsev
Media Communications and Networking Research Laboratory
Department of Math & Computer Science, Kent State University
233 MSB, Kent, OH 44242

ABSTRACT

The possibility of perceptual compression using live eye-tracking has been anticipated for some time by many researchers. Among the challenges of real-time eye-gaze-based perceptual video compression, are how to handle the fast nature of eye movements with the relative complexity of video transcoding and also take into account the delay associated with transmission in the network. Such a delay requires additional consideration in perceptual encoding because it increases the size of the area that requires high-quality coding. In this paper we present a hybrid scheme, one of the first to our knowledge, which combines eye-tracking with fast in-line scene analysis to drastically narrow down the high acuity area without the loss of eye-gaze containment.

Keywords: eye-gaze, perceptual encoding, MPEG-2.

1. INTRODUCTION

Perceptual coding is emerging as a new area of high-fidelity media coding. Our eyes see only a fraction of our visual plane at any given time. The intention of perceptual video compression is to calculate the spatial distribution of bits with close coherence of the perceptually meaningful shapes, objects and actions presented in a scene. A number of researchers have confirmed the effectiveness of this approach. However, there are many challenges in implementing such a scheme. It is difficult to detect the perceptual significance of a particular area in a given video scene. Indeed, the

nature of perception dictates that our previous experience plays an important role in the process of visualization. Thus, a scene itself may not have all the information that guides visual attention. Researchers of vision and eye-tracking have suggested the use of perceptual coding with direct eye-tracker based detection of perceptual attention focus. Approximately 2 degrees in our 170 degree vision span have sharp vision. A fascinating body of research exists in vision and psychology geared toward the understanding of the human visual system. These techniques involve processing information from eye and head-tracking devices, and then attempting to obtain the correct eye-gaze position with respect to a visual plane. Based on the acuity distribution characteristics of a human eye around the fovea, these methods use variable spatial resolution coding for image compression. These techniques require precise spatio-temporal information about the position of the eye. In the case of a large network delay, and/or encoding delay, an eye can move away from its detected location by the time the information is received and processed. This severely offsets the 2 degree acuity advantage.

We have performed several experiments to develop a hybrid technique that combines direct eye-gaze sampling with a video scene content analysis. It is widely believed that both scene content and the pattern of eye movement determine the precise area of human attention. Our hybrid scheme uses both these facts to calculate the area of perceptual attention focus and thus define the part of the image that requires high-quality coding. The hybrid scheme first creates a *Saccade Window* (W^{SW}) based on past eye-gaze information, predicting where a subject's visual attention is going to be directed. Then it calculates an *Object Tracking Window* (W^{TW}) based on a fast content analysis algorithm, refining the vicinity of the viewer's attention focus. After these two steps are completed a *Perceptual Object Window* (W^{POW}) is constructed based on the tentative attention areas identified by W^{SW} and W^{TW} increasing the advantages and reducing disadvantages of both.

We show that our technique reduces the image area requiring high-quality coding, thus increasing the scope of compression. Also, our method enables more eye-gazes to be contained within the W^{POW} for its size, thus retaining the perceptual quality. This technique is probably one of the first that merges the two major paradigms of perceptual encoding. The overall perceptual object window is media independent and can be applied to any video compression method. We have also recently completed an MPEG-2 implementation of the proposed hybrid scheme.

The two base techniques for scene analysis and eye-gaze-based saccade window are described separately in [5] and [8]. In this paper we present them briefly in the following two sections. In section 4 we present several possible schemes for combining them. Then in section 5 we will present experimental results and analysis.

1.1. Related Work

A large number of studies have been performed to investigate 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 subjects [2, 9, 11, 15, 19]. Geisler and Perry [4] presented pyramid coding and used a pointing device to identify the point of focus by a subject. Daly et. al. [26] presented an H.263/MPEG adaptive video compression scheme using face detection and visual eccentricity models. Bandwidth reduction of up to 50% was reported. Khan and Yan [6, 7] demonstrated a mouse-driven high resolution-window overlay interface for medical video visualization over bandwidth-constrained links. Many of the early works have been inspired by the objective to design good quality display systems [1, 3, 11]. For example, Daly [1] utilized a live eye tracker to determine the maximum frequency and spatial sensitivity for HDTV displays with fixed observer distance. Lee and Pattichis [10] discussed how to optimally control the bit-rate for an MPEG-4/ H.263 stream for foveated encoding. Stelmach and Tam [18] have proposed perceptually pre-encoding video based on the viewing patterns from a group of people. Babcock et. al. [27] 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. [20] and Parkhurst et. al. [21].

Among the methods that have been employed for object detection in video, Ngo et. al. [12] described object detection based on motion and color features using histogram analysis. This technique processes less than 2 frames in one second. Unfortunately, several of the other techniques presented did not provide any evaluation of time performance [14]. However, object detection depends on even more involved image processing methods, such as an active contour model, which puts considerable effort into determining the boundary of a shape, and is thus likely to be slower. More recently,

some compressed domain techniques have been suggested by Wang et. al. [13]. Their system achieved about 0.5 sec/frame for the CIF size on a Pentium III 450 MHz.

Virtually no analysis of perceptual compression techniques that combine eye gaze tracking and scene analysis exist.

1.2. Perceptual Transcoding

This section presents a brief description of our initial eye-tracker-based transcoding system paradigm.

The critical issue for a real-time perceptual video compression and transmission system is the *feedback delay*. Feedback delay is the period of time between the instance when the eye position is detected by an eye-tracker, and when the perceptually transformed frame is displayed to the viewer. The feedback delay originates primarily from the network during video and eye-gaze data transmission. Feedback delay should be compensated for with a prediction of viewer's perceptual attention for future video frames. If the area of perceptual attention is properly predicted the image can be perceptually compressed in a way that original image quality is retained in the predicted perceptual attention area and the background quality is degraded with the use of visual sensitivity function [26]. That way a viewer would not be able to notice the difference between the original and perceptually compressed frame. It is important to note that feedback delay can be large and also dynamically varying.

The primary goal of our perceptual transcoding system design was to address situations when different feedback delay values are present in the perceptual compression system as it is shown in the experiment section below. Our perceptual compression system uses the integrated approach of perceptual attention focus prediction, eye gaze containment, and scene analysis. First it predicts viewer's perceptual attention focus area that we call a saccade window. Its goal is to ensure that the targeted amount of the eye-gazes will remain within a certain area with a statistical guarantee, given some value of feedback delay. Second our system tracks moving objects presented in the video stream and based on the object information refines perceptual attention focus area predicted by the saccade window. We call perceptual attention area created by different methods a *Perceptual Attention Window* (W^{PAW}). The amount of viewer's attention towards a specific W^{PAW} is measured through eye gaze containment for that area. The area of perceptual attention window is encoded with high quality while the periphery is encoded with low quality. This approach allows bit-rate reduction while visual quality is maintained at the level close to the original.

Note that potential gain in perceptual video compression depends on the size of the high-quality area (perceptual attention window) on the video frame. Naturally, the goal of our design was to reduce the size of the high-quality area (W^{PAW}) without sacrificing the gaze containment, which determines the potential for further video bit-rate reduction. We have implemented our scheme in MPEG-2 based software transcoder.

2. SACCADE WINDOW

2.1. Human Visual Dynamics

Scientists have identified intricate types of eye movement that include drift, saccade, fixation, smooth pursuit eye-movement, involuntary saccade. Among them, the following two play the most important roles in the human's visual system. It is generally assumed that a human's eye perceives the highest quality picture during a fixation – "eye movement which stabilizes the retina over a stationary object of interest" [17]. It is widely believed that no vision occurs during saccades – "rapid eye movements used in repositioning the fovea to a new location in the visual environment" [17].

2.2. Saccade Window

The ultimate goal of the *Saccade Window* (W^{SW}) design is to contain eye fixations by estimating eye movement speed due to saccades. Analyzing previous eye speed behavior, the W^{SW} represents an estimated area where the eye is going to be in the future, thus predicting a future perceptual attention focus for a given viewer. We should note that, such eye characteristics as acceleration, rotation, and deceleration involved in ballistic saccades are determined by muscle dynamics, and demonstrate stable behavior. The latency, vector direction of the gaze, and the fixation duration, has been found to be highly dependent on the content, and hard to predict. Thus we model W^{SW} as an ellipse which is centered at the last known eye-gaze location, allowing the gaze to take any direction within the acceleration constraints. The current implementation of the saccade window contains all types of eye movements - fixations, saccades, drift, etc. - based on the eye positions provided by the eye tracker.

If (x_c,y_c) is the current detected eye-gaze position, then W^{SW} is an ellipse with center at (x_c,y_c) with half axis $x_R = T_d V_x(t)$ and $y_R = T_d V_y(t)$. See Figure 2.2.1. T_d is a feedback delay $V_x(t)$ and $V_y(t)$ are the *containment assured eye velocities* (CAV). CAV represents the predicted eye velocity, which will allow for the containment of the targeted amount of eye-gazes given a value of the feedback delay. The length of the T_d consists of the delay introduced by the network and eye tracking equipment plus the time it takes to encode a particular video frame. The saccade window is placed on the last available eye-gaze and is updated for every video frame. More detailed information about W^{SW} construction and CAV calculation is available in [8].

3. OBJECT WINDOW

The *Object Tracking Window* (W^{TW}) represents the part of the image that contains an object in the video frame. Object tracking window presents the approach of predicting a viewer's perceptual attention focus through a scene and object analysis. The assumption here is that the viewer looks at a particular object during perception of a visual scene.

Despite the availability of many algorithms for object detection, not all can be applied. The challenge is to select a scene tracking algorithm that can satisfy the speed and streaming constraints faced by a real-time transcoder. In the real-time transcoding, the object detection has to be performed extremely fast, at the rate of the stream. Also, in a streaming scenario, the entire bit stream is not available at any time (thus, techniques such as histogram can not be used). Transcoding the original pixel-level frame images are no longer explicitly available. Instead, the information is organized in an encoded transform space. A transcoder generally may receive some refined information (such as motion vectors). Techniques for transcoding can generally be used in first stage encoding but the reverse in not always possible. Therefore, for this system we have used a transformed domain approach presented in [5] called Flock-of-Birds (FOB) block motion algorithm. The approach has been developed for fast object approximation in video. It is suitable for our eye-tracking perceptual transcoding scheme for the reasons listed in the next two paragraphs.

3.1. Flock-of-Bird Approach:

The FOB approach is a compressed domain approach which depends on P frame motion vector analysis and a Kalman filter for detection of contiguous blocks covering the objects in a scene during transcoding. As explained earlier, it takes advantage of two characteristics of the scenario to reduce the cost dramatically. It builds its feature model on the encoded properties (such as motion vectors and DCT color energy) rather than on the raw pixel set. Secondly, this takes advantage of the observation that region-based reconstruction usually does not require precise specification of the object boundary. We will provide experimental verification of this shortly. Also, this algorithm can optionally accept a high-level cue of the expected target in terms of descriptors such as approximate initial position, size, and shape. It then automatically detects and tracks the region covered by these objects for subsequent perceptual encoding based on real-time motion analysis. Also, the entire tracking computation is based on the data from past frames. Predictions are based on Kalman filters only, and thus it is applicable to real-time video streaming.

3.2. POP Model

The projection of an object on the entire video is called *perceptual object projection* or POP. In POP, a video frame is a mosaic of elementary shapes. For an MPEG-2 stream these elementary shapes or *mosaics* are macroblocks. Thus, a frame F_t is a matrix of macroblocks $b_t(i,j)$, i and j being the column and row indices and subscript t the frames presentation sequence. It then reorganizes a video into a set of *objects* and their *projections* on the elementary mosaic set. The projection of an object on a frame F_t is denoted by a region $R_t(r)$. Each $R_t(r) \subseteq F_t$. Here, *t* is the frame index and *r* is the object index. Each object and (corresponding POP) is defined by a set of descriptors based on some properties of the elementary regions (to be described shortly). Thus a frame F_t= {$b_t(i,j)$ all macroblocks at time t}. A video is the union of all frames:

$$V=\left\{\bigcup_t [F_t]\right\} \tag{3.2.1}$$

And a particular perceptual object

$$POP(r) = \left\{ \bigcup_{t} [R_t(r)] \right\} \quad (3.2.2)$$

It also defines a background macro-block set:

$$B = V - \left\{ \bigcup_{r} POP(r) \right\} \quad (3.2.3)$$

Thus, the POP recognition problem can be stated as the task of detecting all the perceptually distinguishable POPs given a target video, and the POP descriptors. We also define the 'streaming constraint'. The detection of $R_t(r)$ is based on the past frames $\left\{ \bigcup_{k \le t} [F_k] \right\} \subseteq V$. Several of the previously reported approaches did not consider this restriction in frame dependency. These may not be easily applicable for streaming applications.

Tracking Model: Our tracking model introduces three mosaic subsets (for an MPEG-2 stream they are macroblocks) within each POP. For each POP(r), the frame macro-blocks are classified into (a) *active* set ($\Phi_t(r)$), (b) *monitored* ($\Pi_t(r)$) and (c) *inactive* ($\Psi_t(r)$) set based on mosaic macroblock property criterion. The macroblocks in the active set represent the POP projection.

Each POP is described with two sets of attributes (i) Germination, (ii) and Flocking descriptors. The germination parameters are used to identify the spontaneous birth and death events of the POP regions in a frame. Figure 3.2.1 illustrates the process. Once the birth of a POP is detected, it is brought into the flocking state from the germinal state. The flocking state POP is then handed over to the live flock-of-bird tracking process which tracks the POP using *flocking* parameters. In the flocking state the macroblocks are dynamically moved between the active, monitored and inactive sets based on the defined flocking descriptors. A separate three set tracking occurs for each of the detected objects.

Property Model: For each of the macroblocks a set of macroblock properties called the *mosaic property set* (MoPS) is estimated. The average on a collection of mosaics defining the active set of a POP provides the *object property set* (PoPS). The differential between the MoPS and PoPS are continuously monitored between frames for all macroblocks in active and monitored set. With each frame forward, the sets themselves are updated by inverse projection of the motion in the macro-blocks. In each P frame the macroblocks in active and monitored sets are then evaluated for membership. The set transitions are determined by *set transition rules* which are defined based on distance measure

$||V^{PoPS}-V^{MoPS}||$ of individual macroblocks. A macroblock can be a member of the active sets of multiple POPs. It has no impact on recognition but it takes the best of their rendering quality attributes.

For our experiment we used the following Germination Descriptors (i) *Formation Mass, (ii) Formation Velocity (iii), Dissolving Mass, and (iv) Dissolving Velocity*. The birth and death of POPs are not symmetric. The birth depends on the MoPS, while the death depends on the PoPs. The flocking process was tracked with the following six flocking parameters: (i) *Monitor Span (ii) Deviator Thresholds (iii) Follower Thresholds (iv) Deviator Persistence, (v) Follower Persistence*. In the tracking phase, each flocking POP is also checked against the dissolve criterion. The POPs which fall below, are returned to the inactive pool, and the POP is taken back into the germinal state. The algorithm demonstrated robust tracking under various video streaming scenarios. It was able to perform the tracking in real-time and added less than 1% computation cost to the transcoding operation. The detail performance of the algorithm has been presented in [5]. The tracked window is thus determined as the following:

$$W^{TW}(t) = \left\{ \bigcup_{r} [\mathrm{POP(r)}] \right\} \tag{3.2.5}$$

4. HYBRID VISUAL WINDOW

Both W^{SW} and W^{TW} are tools for perceptual attention focus estimation and prediction. Each of them is based on a different construction method. W^{SW} is based on the eye position detection and analysis. W^{TW} is based on the video scene analysis. We have thought of a possible hybrid scheme which takes both W^{SW} and W^{TW} into consideration to refine the predicted area of viewer's perceptual attention focus within a specific video frame. We have considered five different hybrid models. Two of them are improvements of the object tracking window and three of them are the hybrid windows or *Perceptual Object Windows* (W^{POW}) created with the help of both W^{SW} and W^{TW}. All windows are recalculated for every video frame.

4.1. Rectilinear Approximation

When we look at an object, we also tend to look at the area surrounding it [28]. Therefore, we decided to create an approximation around the object boundaries. For the rectilinear approximation of W^{TW} (W^{RTW}) all coordinates of the macroblocks (MB) in W^{TW} are sorted according to their values. W^{RTW} is constructed using the min max values of those coordinates. W^{RTW}= {(x_{min}, y_{max}), (x_{min}, y_{min}), (x_{max},y_{max}), (x_{max}, y_{min})}, where x_{max} = max{x_i}, x_{min}=min{x_i}, y_{max}=max{y_i}, y_{max}=max{y_i}, where x_i and y_i are MB coordinates and $x_i, y_i \in W^{TW}$. Rectilinear approximation algorithm is shown in the Figure 4.1.1.

4.2. Circular Approximation

Another form of W^{TW} approximation is a circular approximation W^{CTW} shown in the Figure 4.2.1. Let (x_i, y_i) represent the coordinates of a macroblock on the video frame. The distance between two macroblocks is calculated as: $D_{ij} = \sqrt{(x_i - x_j)^2 + (y_i - y_i)^2}$. Let D_{km} be the maximum possible distance between a pair of macroblocks in W^{TW} D_{km}=max{D_{ij}}. Suppose $MB_k[x_k,y_k]$ and $MB_m[x_m,y_m]$ are such macroblocks, then $D_{km} = \sqrt{(x_k - x_m)^2 + (y_k - y_m)^2}$. W^{CTW} is defined as a circle with radius R=0.5D_{km}, and center at ($x_c = \frac{x_k + x_m}{2}, y_c = \frac{y_k + y_m}{2}$).

4.3. Hybridization Method-A

(i) The idea behind this method is to monitor the center of W^{SW} and watch if it falls inside the boundary of W^{CTW}. When it happens, the resulting Perceptual Object Window (W^{POW}) is equal to the intersection of W^{CTW} and W^{SW}. (ii) W^{POW} is equal to W^{CTW} in the case when W^{SW} fully contains W^{CTW}. (iii) In all other cases W^{POW} is equal to W^{SW}. The algorithm for W^{POW} Method-A is represented in the Figure 4.3.1.

Assuming that $MB_{SW}[x_{cen},y_{cen}]$ is the macroblock which represents W^{SW} center, Method-A W^{POW} is defined as:

$$W^{POW} = \begin{cases} 1)\ W^{CTW} \cap W^{SW} \text{ if } MB_{SW}[x_{cen}, y_{cen}] \in W^{CTW} \\ 2)\ W^{CTW} \text{ if } W^{CTW} \in W^{SW} \\ 3)\ W^{SW} \text{ in all other cases} \end{cases} \quad (4.3.1)$$

4.4. Hybridization Method-B

Method-B has the same idea behind it as Method-A. Additionally, Method-B takes into consideration the relative position of W^{CTW} in respect to W^{SW}. In a case when W^{SW}'s center is not contained inside of W^{CTW}'s, Method-B creates W^{POW} as a half of the W^{SW} directed towards W^{CTW}. The algorithm for W^{POW} Method-B construction is represented in the Figure 4.4.1.

W^{DSW} is a Divided Saccade Window. It is constructed from W^{SW} by splitting the Saccade Window in half by a Divided Saccade Window Line (DSWL), which is orthogonal to the line going through W^{SW} center (x_{SW},y_{SW}) and W^{CTW} center (x_{CTW},y_{CTW}). See Figure 4.4.2. DSWL divides video frame F into two planes F' and F''. F' is the plane which contains the W^{CTW} center. W^{DSW} is created by the intersection of W^{SW} and F'. $W^{DSW} = W^{SW} \cap F'$. This is shown on the Figure 4.4.2.

Assuming that $MB_{SW}[x_{cen},y_{cen}]$ is the macroblock which represents W^{SW} center, Method-B W^{POW} is defined as:

$$W^{POW} = \begin{cases} 1)\ W^{CTW} \cap W^{SW} \text{ if } MB_{SW}[x_{cen}, y_{cen}] \in W^{CTW} \\ 2)\ W^{CTW} \text{ if } W^{CTW} \in W^{SW} \\ 3)\ W^{DSW} \text{ in all other cases} \end{cases} \quad (4.4.1)$$

4.5. Hybridization Method-C

First we should introduce W^{ECTW} – enhanced W^{CTW}, which is created from W^{CTW} by increasing the radius R of W^{CTW} by some number ε. Quantity ε is adjusted during video playback to provide better performance. In order to choose the best value for ε our algorithm measures how far eye gazes fall from the boundary of W^{CTW}. Value δ_i - deviation is defined as the distance between the W^{CTW} boundary and the eye gaze point S_i. Deviation is calculated only for those S_i located outside of the W^{CTW} boundary. See Figure 4.5.1. Deviation values are collected over some period of time, usually not exceeding the duration of m video frames. The deviation values are processed and new value for ε is chosen for each video frame based on some percentile target parameter ϖ. Values of m and ϖ are chosen based on empirical analysis. They are feedback and video content dependent. For this particular experiment m=10, and $\varpi = 0.7$. With a

new radius the W^{SW} center falls into the boundaries of W^{ECTW} much more often. (i) To create the final W^{POW} our algorithm chooses the intersection of W^{ECTW} and W^{SW} if the W^{SW} center lies inside of W^{ECTW}. (ii) W^{POW} is equal to W^{ECTW} in the case when W^{SW} fully contains W^{CTW}. (iii) In all other cases $W^{POW}=W^{SW}$. The algorithm for W^{POW} Method-A is represented in the Figure 4.5.2.

Assuming that $MB_{SW}[x_{cen},y_{cen}]$ is the macroblock which represents W^{SW} center, Method-C W^{POW} is defined as:

$$W^{POW} = \begin{cases} 1)\ W^{ECTW} \cap W^{SW} \text{ if } MB_{SW}[x_{cen}, y_{cen}] \in W^{ECTW} \\ 2)\ W^{ECTW} \text{ if } W^{ECTW} \in W^{SW} \\ 3)\ W^{SW} \text{ in all other cases} \end{cases} \quad (4.5.1)$$

5. EXPERIMENT

5.1. Setup

Our transcoding system was implemented with Applied Science Laboratories eye-tracker model 504. ASL 504 which has the following characteristics: accuracy - spatial error between true eye position and computed measurement is less than 1 degree; precision - better than 0.5 degree. That model of eye-tracker compensates for small head movements, so the subjects head fixation was not necessary. 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 calibration accuracy, 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 was used for each subject calibration procedure. The data from the eye-tracker equipment was streamed to the control center through the serial port, and then eye gazes were streamed through a socket connection to the player displaying the test video. The final data collection was performed at the video player’s end. Each video was projected onto a projection screen in a dark room. The projected physical dimensions of the image were

these: width 60 inches, height 50 inches. The size of each of the test videos was 704x480 pixels. The distance between the subject's eyes and the surface of the screen was about 100-120 inches. The data was collected from two male and one female subjects, with normal or corrected-to-normal vision. We selected three video clips with different content to provide a comprehensive challenge to our algorithm for accurate performance evaluation. We performed a uniform as well as a perceptual (where the perceptual window area had high quality and the area around it was encoded with reduced quality) bit-rate reduction. The reduction was made from 10 Mb/s bit-rate to 1 Mb/s bit-rate.

5.2. Test Data

We have selected three video sequences to test the performance of the methods discussed above. Each video was 66 sec. long.

"Video 1" is of a moving car. It was taped from the point of view of a security camera in a university parking lot. The visible size of the car is approximately one fifth of the screen. The car moves slowly, allowing the subject to develop smooth pursuit movement (our assumption).

"Video 2" had two radio-controlled toy cars moving at varying speeds. Both toy cars had rapid, unpredictable movements. In this video we asked the subjects to concentrate on just one of the cars.

"Video 3" had two relatively close up toy cars, offering a large area of attention. The cars moved in different directions inconsistently. Each subject was asked to concentrate on only one car.

The subjects who viewed the test videos were familiarized with them before the experiment.

Figure 5.2.1 shows the example of a frame from "Video 1" with different perceptual attention windows (W^{TW}, W^{SW}, W^{POW}) displayed. "Video 1" had bit-rate of 10MB/s and perceptual windows were constructed for feedback delay scenario of 1 sec. Perceptual object window for the "video 1" frame 441 was constructed with a hybrid Method-C, case 1, where saccade window center falls into the boundaries of enhanced circular tracking window and W^{POW} is created by intersection of W^{SW} and W^{ECTW}. In this example the original tracking window was not able to contain an eye-gaze, but W^{POW} contained it. Note that resulting size of W^{POW} Method-C was much smaller than of saccade window W^{SW}.

Figure 5.2.2 shows the example of a perceptually encoded frame, based of the Method-C W^{POW}. Note that the bit-rate was reduced by 10 times from 10Mb/s down to 1 Mbp/s. In this bit reduction scheme high resolution was maintained at

the macroblocks inside of W^{POW}. The MPEG-2 TM-5 rate control was used to control quantization levels of macroblocks inside and outside of W^{POW}. Perceptually encoded video samples, including the originals with different perceptual window boundaries displayed, can be found at our website [16].

6. PERFORMANCE ANALYSIS

To measure the effectiveness of our algorithm, we have defined the following two parameters: average eye-gaze containment and average perceptual coverage efficiency. All different window types that we mentioned above (W^{TW}, W^{SW}, W^{RTW}, W^{CTW}, $W^{ECTW,}$ and W^{POW} Method-A,B,C) are instances of P*erceptual Attention Window* (W^{PAW}). The area of a perceptual attention window represents the part of the image which requires high-resolution coding.

6.1. Eye-gaze Containment

In the current implementation of our perceptual media adaptation system, the performance of each perceptual attention window construction algorithm is measured by the containment of the eye-gazes inside of this window. The higher gaze containment for a particular W^{PAW}, the smaller the probability that peripheral image degradation will be detected. Thus, we defined the quantity *average eye-gaze containment* as the fraction of detected eye positions successfully contained within a perceptual attention window over N frames:

$$\xi = \frac{100}{N}\sum_{t=1}^{N}\frac{\left|\mathrm{E}^{\mathrm{PAW}}(t)\right|}{\left|E(t)\right|} \qquad (6.1.1)$$

Where, E(t) is the entire eye-gaze sample set for the frame F(t). $E^{W}(t) \subseteq E(t)$ is the eye-gaze sample subset contained within a perceptual attention window W^{PAW}(t). N is the number of frames in a video.

6.2. Perceptual Coverage

The other important design goal of our system is to decrease the size of the image area requiring high-quality coding (perceptual attention window size), while containing as many eye gazes as possible. It is obvious that with a large

W^{PAW}, more eye gazes can be contained. However, there will not be any perceptual redundancy to extract from a video frame. Therefore, we have defined a second performance parameter called *average perceptual coverage* for obtaining video frame coverage efficiency by a perceptual attention window. If ΔF(t) is the size of the viewing frame, and W^{PAW}(t) is a perceptual attention window, then the perceptual coverage is given by (delta for area or volume):

$$\chi = \frac{100}{N} \sum_{t=1}^{N} \frac{\left|\Delta(W^{W}(t) \cap F(t))\right|}{\left|\Delta(F(t))\right|} \qquad (6.2.1)$$

Next we present the performance of each method with respect to these two parameters.

6.3. Analysis of Results

After eye gaze containment and perceptual coverage was calculated for each subject and each perceptual attention window the final results were averaged between three subjects.

Performance of the Eye-Gazed based System:

Figures 6.3.1-6.3.3 provide the results. The left y-axis and the bar-graphs show the perceptual coverage efficiency of each method. The right y-axis and the line curves show the corresponding gaze containment. The leftmost TW (W^{TW}) and rightmost SW (W^{SW}) cases respectively show the performance of the strictly object-based method, and strictly eye-gaze-based method. In the absence of significant feedback delay, (166 ms or 5 video frames), the eye-tracker-based methods offered approximately a 3-6% frame coverage and roughly a 90% gaze containment. However, when the feedback delay was about 1 second (30 frames), the Saccade Window became quite large, close to 26-41%. With larger frame coverage there was less scope of compression.

Performance of the Pure Object-based System:

W^{TW} represents a case of scene-analysis-based perceptual video encoding. We can see that the advantage of W^{TW} is its smaller coverage area (about 3-5%). The small coverage by this perceptual attention window provides the potential for high compression. Conversely, its weakness is gaze containment. As it can be noted, despite the small coverage, W^{TW} actually misses a significant amount of the eye-gazes. Its containment is only about 45-67%. Thus, a perceptual compression based on just object detection is expected to lack high perceptual quality.

Improvement due to Approximations:

Before we move to the hybrid techniques, we also present the performances of the two approximations that are performed based on the pure-object approach. The plots CTW (W^{CTW}) and RTW (W^{RTW}) respectively provide corresponding performances. Compared to strict object-boundary-based TW (W^{TW}), these approximations increase the coverage area from 3-5% to 3-8%. However, at the same time these improve the gaze containment significantly from 45-67% to about 51-84%.

Hybrid Methods:

The incorporation of the scene analysis kept the eye-gaze containment near the level of the eye-tracking-only method (W^{SW}), but drastically reduced the perceptual coverage. For a 1 second feedback delay, Method-A kept the average eye-gaze containment around 75-82%, but the perceptual coverage was drastically reduced from 26-41% (Saccade Window containment) to about 13-22%. Among the methods used, Method-B was more conservative on the side of reducing the perceptual coverage. It offered coverage of about 8-12% with the average eye-gaze containment of about 61-75% for the 1 second feedback scenario. Method-C, on other hand, achieved eye-gaze containment almost at the level of the pure eye-tracker-based method W^{SW} - around 84-88% - but reduced the coverage to a level of 9-25%. As a conclusion we can say that the hybrid methods, particularly Method-C, were able to reduce the perceptual coverage from 26-41% to about 9-25%, without a significant loss of the eye-gaze containment.

Impact of Feedback Delay:

Feedback delay is a major performance factor in the proposed hybrid scheme. The longer the delay, the larger was the size of the constructed perceptual attention window. In the case of a 1 second delay, the sizes of the perceptual object windows were around 8-25% of the video frame. In case of a 166 ms delay the perceptual coverage went down to 1-5%. But in each feedback delay scenario, hybrid methods reduced the perceptual coverage significantly compared to simply object-based or eye-gaze-based methods.

Impact of Object Size & Quantity:

The hybrid approach has the ability to reduce the size of the perceptual attention window, making it even smaller than the size of the object itself. Any method solely using an eye-tracker must use a larger perceptual attention window due to the inherent feedback delay. This is evident in the experiment with the third video which features two objects. A scene-only analysis faces ambiguity, as it does not know precisely at which object a person is looking. In the hybrid

method, the eye gaze analysis helps resolve this uncertainty. In the case of the 166 ms delay experiment, the perceptual coverage of the hybrid Method-C for one of the subjects was around 4% with eye-gaze containment of 89% compared to 5% coverage of W^{TW} and gaze containment of 62%. Thus, in the case of a large object tracking, the hybrid technique was able to create a smaller-sized area with more viewer's attention directed to it, which was proved by increased gaze containment from 62% to 89%.

Perceptual Resolution Gain:

The actual amount of perceptual compression depends on the two perceptual attention window characteristics - the size of the window and the allocation of bits outside of the window, which should match a human eye acuity model. To estimate the advantage of perceptual video compression for a variable bit-rate transcoder in a delay scenario we introduce the quantity *perceptual resolution gain* (PRG). Perceptual resolution gain is directly proportional to the perceptual coverage depicted on the Figures 6.3.1-6.3.3. As we defined previously, perceptual coverage represents the area of the video frame which requires high-quality coding. More perceptual resolution gain can be achieved if the size of that area is small. As an example we can present a case in which each pixel outside a W^{PAW} requires half the amount of bits to encode on average, comparing to pixels inside of the perceptual attention window.

$$PRG = \frac{100}{\chi + 0.5(100 - \chi)} \tag{6.2.1}$$

We selected two methods to compare the perceptual resolution gain for the same level of gaze containment (90%). One method is the eye-tracker-only based saccade window and the other is the hybrid perceptual object window construction Method-C. Results are presented in Figure 6.3.4. It is possible to see that for the small feedback delay of 166ms PRGs values for those methods are practically the same and are close to two. In the case of a large feedback delay of 1 second the difference becomes more significant – the PRG for SW is around 1.42-1.58 and for the hybrid method the PRG varies from 1.6 to 1.82. Thus we can say that hybrid methods decrease the area which requires high-quality coding and thus allow achieving higher compression levels, especially in situations with high feedback delay/lag.

7. DISSCUSSION

There are a few challenges and limitations that we encountered in the implementation of our hybrid system, which might be of interest to the eye-tracking and perceptual media adaptation community.

Transcoding speed:

One of the difficulties that we encountered in the implementation of our perceptual video adaptation system is the speed of the software MPEG-2 transcoder, which produces a perceptually adapted MPEG-2 stream. The highest transcoding frame rate that we were able to achieve in our lab so far was 8fps on a Pentium 4 3Gz computer for 704x480 MPEG-2 video stream. Other components of our system perform in real-time (the assumption here is that a real-time system can run at 30fps). The saccade window construction algorithm is linear and based only on past eye-speed samples. The object tracking algorithm is based only on MPEG-2 motion vector analysis and is a linear algorithm. The approximation and hybrid (perceptual object) windows construction algorithms are performed at the macroblock level which makes them real-time for 704x480 resolution. Because of the 8fps transcoder limitation, a major part of the data analysis for the proposed system is done offline, but the system has the capability of being a real-time once the decoding-encoding speed reaches 30fps. The path to real-time transcoding performance lies through further optimization of decoding/encoding parts inside of the MPEG-2 transcoder or running the transcoder on more powerful computers. So far we have already performed some optimization to the MPEG-2 transcoder – motion vectors are directly sent to the encoder from the decoder which reduced the computational burden quite a bit. For more details about our MPEG-2 transcoder architecture and performance see [25].

Eye-position filtering and analysis:

In the current implementation of our system the eye-positions from the eye-tracker are sent through a network to a computer which runs MPEG-2 software. The eye-positions for which tracking has failed (eye positions identified as having 0,0 coordinates by the eye-tracker equipment) are filtered out and substituted by the previous valid eye-position. The amount of such eye-positions in the eye-data trace was extremely low – less than 0.1%.

For perceptual media adaptation (perceptual video transcoding) it is generally assumed that when calculating the gaze containment, more importance has to be given to the eye-position samples during a fixation than the eye-position samples during a saccade due to the definitions presented in the Section 2.1. However the precise rules for

saccades/fixations detection, their duration times, and their importance to the visual system are still under research. An example here is the recent research done by McConkie and Losckhy [22] which states that clear vision starts only 6ms after the end of a saccade. Foundational work to describe first basic eye-movement characteristics was done by Yarbus [23]. Later researchers refined and added to the definitions provided by Yarbus. Thus some investigators specified that the lowest fixation duration time is 30msec and that the saccade duration can be measured by the formula: $D = 2.2 * A + 21$, where is D is the duration of the saccade measured in msec. and A is the amplitude of the saccade measured in degrees [22].

The additional challenge for an eye-movement type detection and interpretation is presented by the nature of a real-time transcoding system. 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. In the case when network jitter is present in the system, 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. Thus by the time a current viewer's eye movement type is identified by the transcoder, that type of the eye movement might be effectively over.

These are the few issues that we wanted to point out that provide a significant challenge in the implementation of a real-time perceptual media adaptation and transmission system. The current implementation of our system does not yet address all the problems discussed, but we would like to address them in our future works.

8. CONCLUSIONS & CURRENT WORK

Eye trace-based media coding is a promising area. 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. In this paper we have addressed the mechanism of how a viewer's perceptual attention focus area can be further narrowed in a dynamic environment with augmentation from low-grade scene analysis. Opportunities exist for reduction of coverage without any loss of eye-gaze containment. Reduction in coverage provided by hybrid methods leads to the higher

perceptual compression as was shown in the example in section 6.3. It is important to note that in live video transcoding processing, speed is a critical consideration. Consequently, for both scene analysis and perceptual attention prediction we have used computationally low-cost approximation approaches. There are intricate schemes known for the detection of objects in videos, though these require massive image processing and we could not use them for this scenario.

It is also interesting to note that with few exceptions, most of the previous studies in eye-tracker-based media compression have focused on the study of the foveal window degradation around the point of an eye-gaze. Even when some type of fovea region was considered, it was of fixed size and static. In this paper, we have focused on a scenario where the perceptual video compression scheme is affected by a significant feedback delay/lag. This makes the dynamic estimation and prediction of the area that requires the highest quality coding more important than the precise calculation of the peripheral acuity degradation.

Further research should be performed to understand the media-dependent degradation and coding models when the perceptual attention focus prediction is affected by a delay. The ways of improving the saccade/hybrid window construction methods should be researched as well. At its current implementation the saccade window contains all eye movement types, but a possible improvement might be to change its shape during different types of eye movements. That might reduce perceptual coverage and increase compression while keeping perceptual quality perceived by a viewer at the original level.

9. ACKNOLEGMENTS

This work was supported in part by DARPA Research Grant F30602-99-1-0515. We would like to thank students, faculty and staff of Computer Science department of Kent State University. Additional thanks go to E. Liptle, A. Danilov, J. Chitalia for being the subjects in the experiments.

REFERENCES:

[1] S. J. Daly, "Engineering observations from spatiovelocity and spatiotemporal visual models," in *Proc. SPIE Vol. 3299*, pp. 180-191 (1998).

[2] A. T. Duchowski, "Acuity-Matching Resolution Degradation Through Wavelet Coefficient Scaling," *IEEE Transactions on Image Processing 9 (8)*, pp. 1437-1440 (2000).

[3] A. T. Duchowski and B. H. McCormick, "Gaze-contingent video resolution degradation," in *Proc. SPIE Vol. 3299*, pp. 318-329 (1998).

[4] W. S. Geisler and J. S. Perry, "Real-time foveated multiresolution system for low-bandwidth video communication," in *Proc. SPIE Vol. 3299*, pp. 294-305 (1998).

[5] J. I. Khan and Z. Guo, "Flock-of-Bird Algorithm for Fast Motion Based Object Tracking and Transcoding in Video Streaming," The 13th IEEE International Packet Video Workshop, April (2003).

[6] J. I. Khan and D. Yun, "Multi-resolution Perceptual Encoding for Interactive Image Sharing in Remote Tele-Diagnostics Manufacturing Agility and Hybrid Automation – I," in *Proc. of the International Conference on Human Aspects of Advanced Manufacturing: Agility & Hybrid Automation*, *HAAMAHA'96*, pp. 183-187 Aug (1996).

[7] J. I. Khan and D. Yun, "Perceptual Focus Driven Image Transmission for Tele-Diagnostics," i*n Proc. International Conference on Computer Assisted Radiology, CAR'96*, pp. 579-584, June (1996).

[8] O. Komogortsev and J. I. Khan, "Predictive Perceptual Compression for Real Time Video Communication," in *Proceedings of the 12th annual ACM international conference on Multimedia ACM Multimedia*, pp. 220-227 (2004).

[9] T. Kuyel, W. S. Geisler, and J. Ghosh, "Retinally reconstructed images (RRIs): digital images having a resolution match with the human eye," in *Proc. SPIE Vol. 3299*, pp. 603-614 (1998).

[10] S. Lee, M. Pattichis, and A. Bovok, "Foveated Video Compression with Optimal Rate Control," *IEEE Transaction of Image Processing, V. 10, n.7*, pp. 977-992, July (2001).

[11] L. C. Loschky and G. W. McConkie, “User performance with gaze contingent multiresolutional displays,” in *Proceedings of the symposium on Eye tracking research & applications*, pp. 97-103, November (2000).

[12] C. Ngo, T. Pong, and H. Zhang, “On clustering and retrieval of video shots,” in *Proceedings of the ninth ACM international conference on Multimedia,* pp. 51-60, October (2001).

[13] R. Wang, H. Zhang, and Y. Zhang, “A Confidence Measure Based Moving Object Extraction System Built for Compressed Domain,” *ISCAS 2000 - IEEE International Symposium on Circuits and Systems*, pp.21-24, May (2000)

[14] K. Gerald, S. Richter, and M. Beier, “Motion-based segmentation and contour-based classification of video objects,” in *Proceedings of the ninth ACM international conference on Multimedia*, pp. 41-50, October (2001).

[15] A. T. Duchowski, B. H. McCormick, “Preattentive considerations for gaze-contingent image processing,” in *Proc. SPIE Vol. 2411*, pp. 128-139 (1995).

[16] O. Komogortsev, J. Khan, “Video Set for Hybrid Perceptual Compression Test,” at *www.cs.kent.edu/~okomogor/SPIEJ05VideoSet.htm*.

[17] A. T. Duchowski, *Eye Tracking Methodology: Theory and Practice*, Springer-Verlag, London, UK, (2003).

[18] L. B. Stelmach, and W. J. Tam, “Processing image sequences based on eye movements,” in Proc. SPIE Vol. 2179, pp. 90-98 (1994).

[19] P. Bergstrom, *Eye-movement Controlled Image Coding*, PhD dissertation, Electrical Engineering, Linkoping University, Linkoping, Sweden, (2003).

[20] E. M. Reingold, L.C. Loschky, G. W. McConkie, and D. M. Stampe, "Gaze-Contingent Multi-Resolutional Displays: An Integrative Review," *Human Factors, 45(2)*, pp. 307-328 (2003).

[21] D. J. Parkhurst, and E. Niebur, E, "Variable Resolution Displays: A Theoretical, Practical, and Behavioral Evaluation," *Human Factors,* 44(4), pp. 611-629 (2002).

[22] G. W. McConkie, and L. C. Loschky, “Perception onset time during fixations in free viewing,” *Behavioral Research Methods, Instruments and Computers, 34*, pp. 481-490 (2002).

[23] L. Yarbus, *Eye Movements and Vision*, Institute for Problems of Information Transmission Academy of Sciences of the USSR, Moscow (1967).

[24] R. H. S. Carpenter, *Movements of the Eyes*, London : Pion (l977).

[25] J. I. Khan, S. S. Yang, D. Patel, O. Komogortsev, W. Oh, Z. Guo, Q. Gu, and P. Mail, "Resource Adaptive Netcentric Systems on Active Network: a Self-Organizing Video Stream that Automorphs itself while in Transit Via a Quasi-Active Network," in *Proc. of the Active Networks Conference and Exposition (DANCE '2002)*, IEEE Computer Society Press, pp. 409-426, May (2002).

[26] S. Daly, K. Matthews and J. Ribas-Corbera, "As Plain as the Noise on Your Face: Adaptive Video Compression Using Face Detection and Visual Eccentricity Models," *Journal of Electronic Imaging V. 10 (01)*, pp. 30-46 (2001).

[27] J.S. Babcock, J.B. Pelz, and M.D. Fairchild, "Eye tracking observers during color image evaluation tasks", in *Proc. SPIE Vol. 5007*, pp. 218-230 (2003).

[28] O.V. Komogortsev and J.I. Khan, "Contour Approximation for Faster Object based Transcoding with Higher Perceptual Quality," in *Proceedings of the Computer Graphics and Imaging (CGIM 2004)*, pp. 441-446, August (2004).

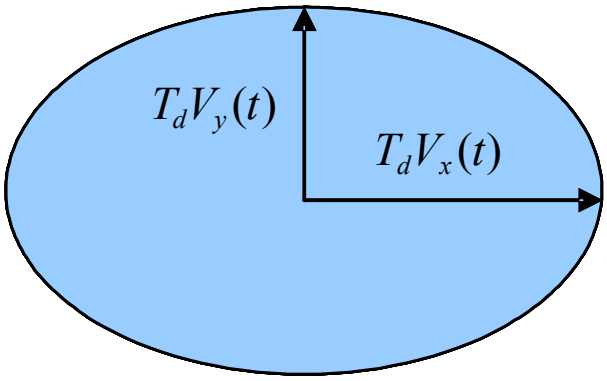

Fig. 2.2.1 Saccade Window diagram

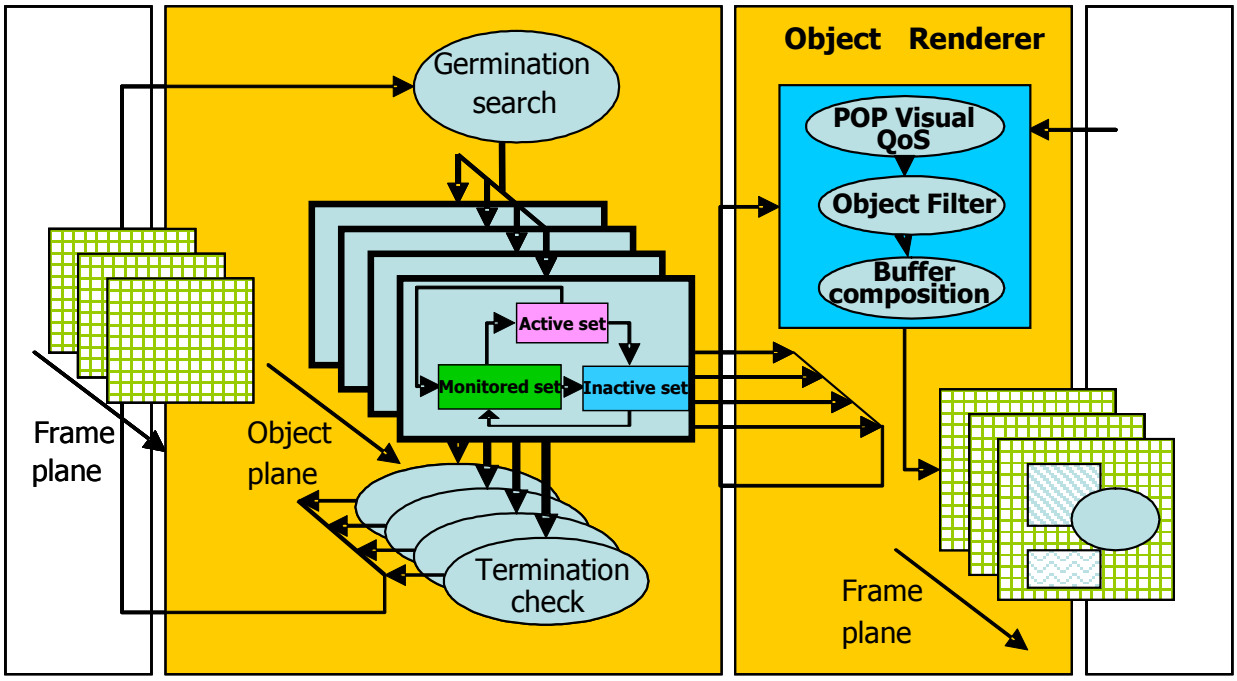

Figure 3.2.1 Internal organization of the object tracking algorithm.

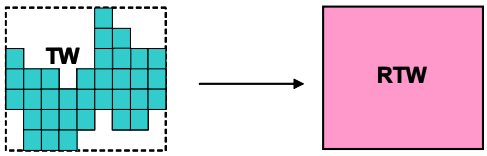

Figure 4.1.1 Rectilinear Approximation of the Tracking Window.

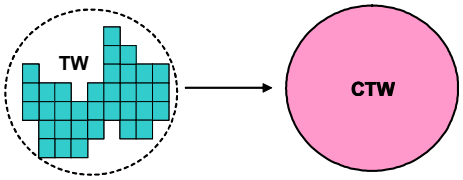

Figure 4.2.1 Circular Approximation of the Tracking Window.

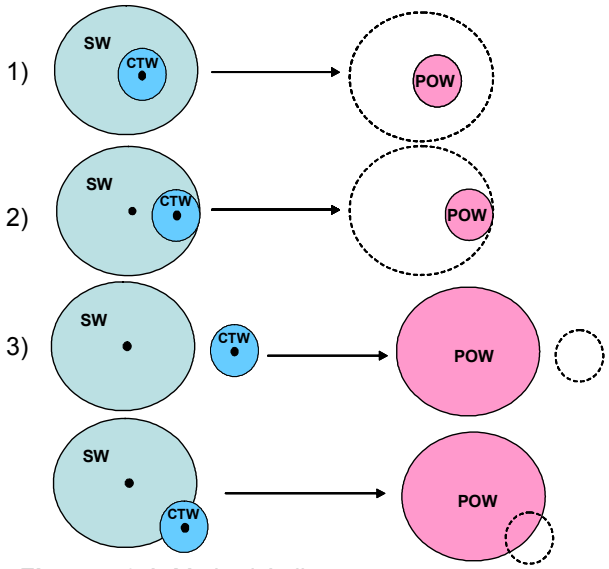

Figure 4.3.1 Method A diagram.

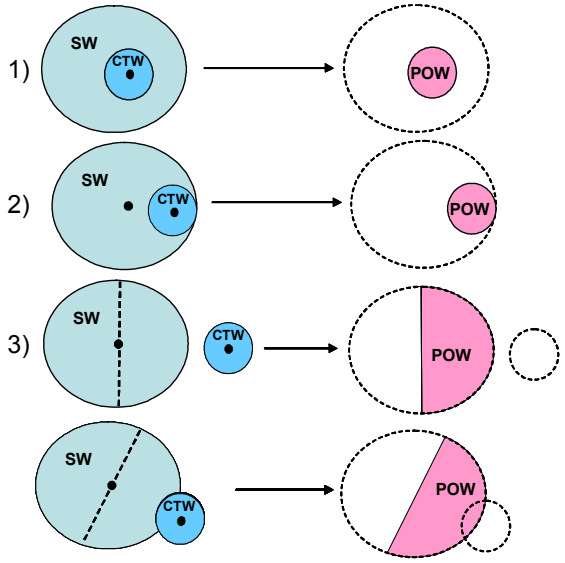

Figure 4.4.1 Method B diagram.

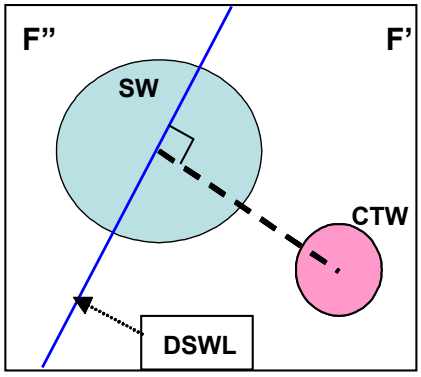

Figure 4.4.2 DSWL construction diagram.

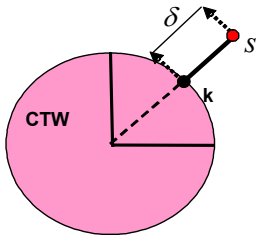

Figure 4.5.1 Deviation representation.

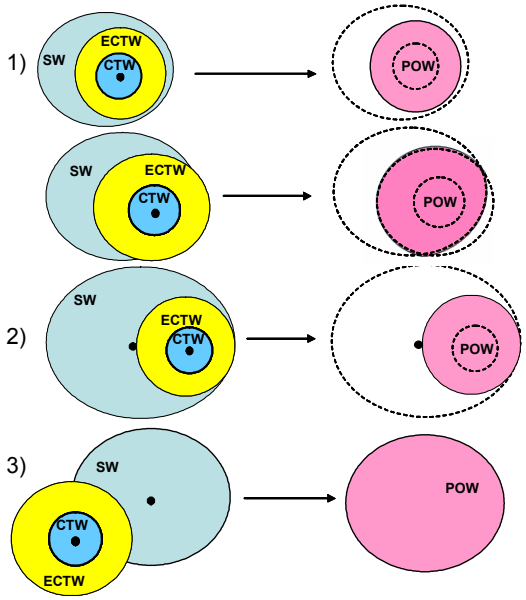

Figure 4.5.2 Method C diagram.

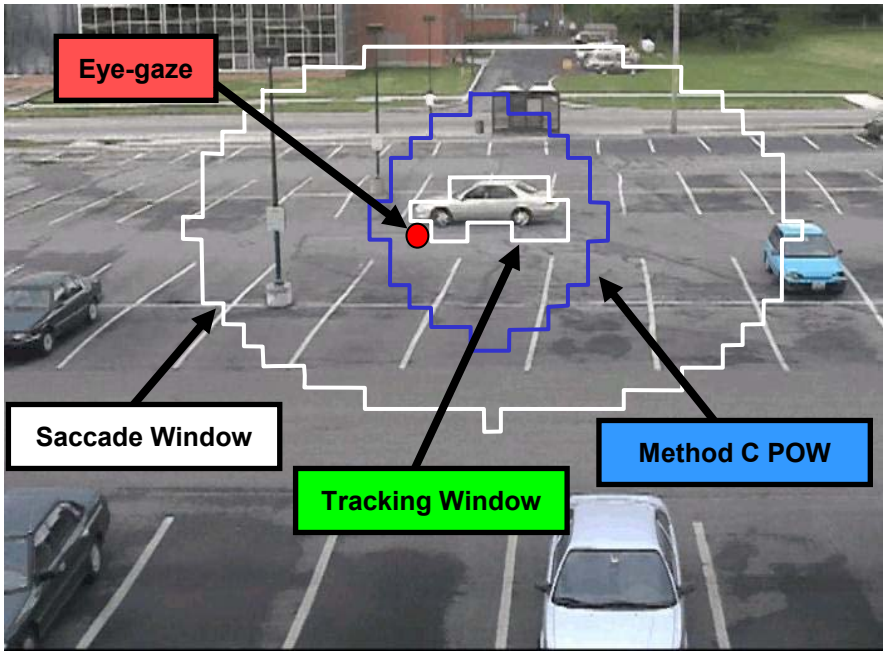

Figure 5.2.1. "Video 1". Frame number 441. Window. Method C POW – perceptual object window created using hybrid method C, case 1. This picture also presents the case when tracking window misses an eye-gaze, but perceptual object window contains it. Bit-rate is 10MB/s.

Figure 5.2.2. "Video 1". Frame number 441. Perceptually encoded. Target bit-rate rate is 1MB/s

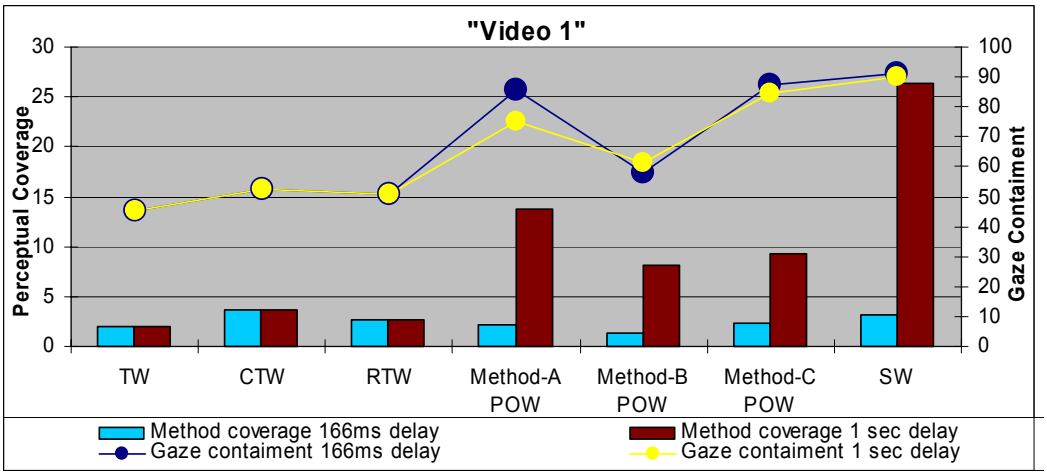

Figure 6.3.1 "Video 1". Average perceptual coverage and average gaze containment data for different POW build methods.

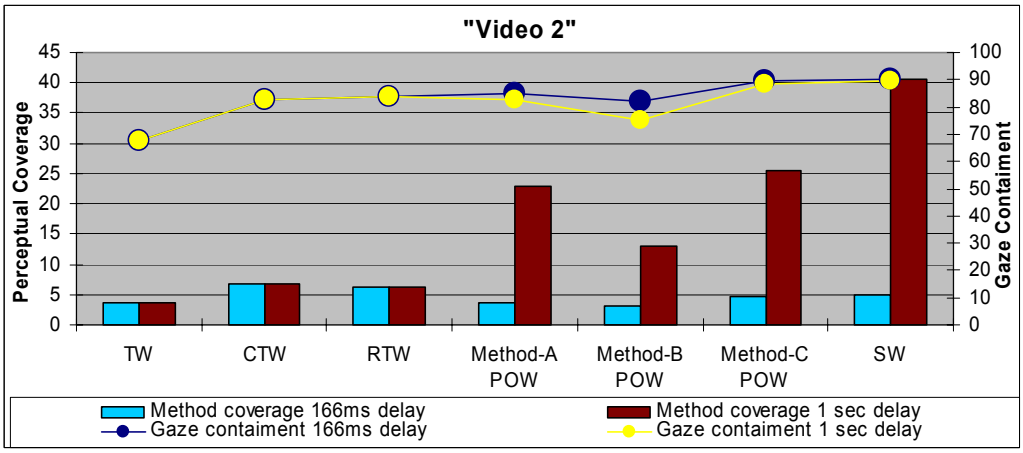

Figure 6.3.2 "Video 2". Average perceptual coverage and average gaze containment data for different POW build methods.

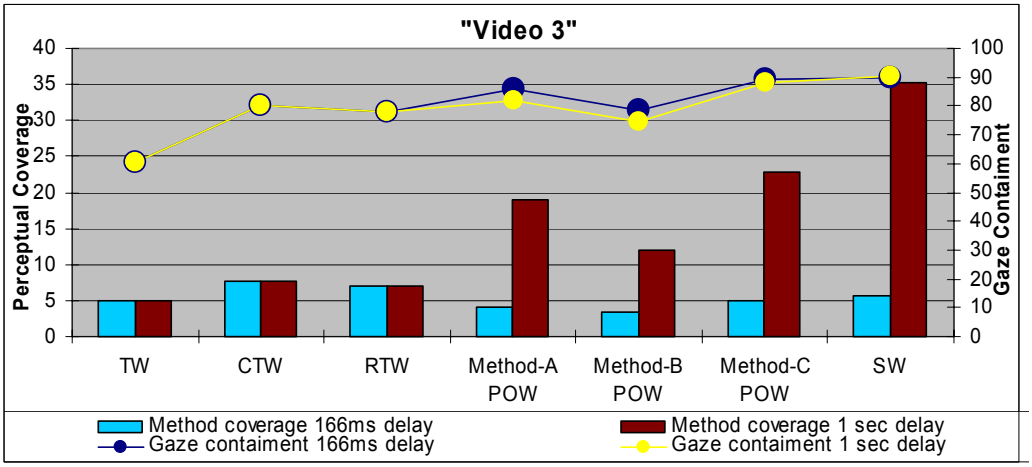

Figure 6.3.3 "Video 3". Average perceptual coverage and average gaze containment data for different POW build methods.

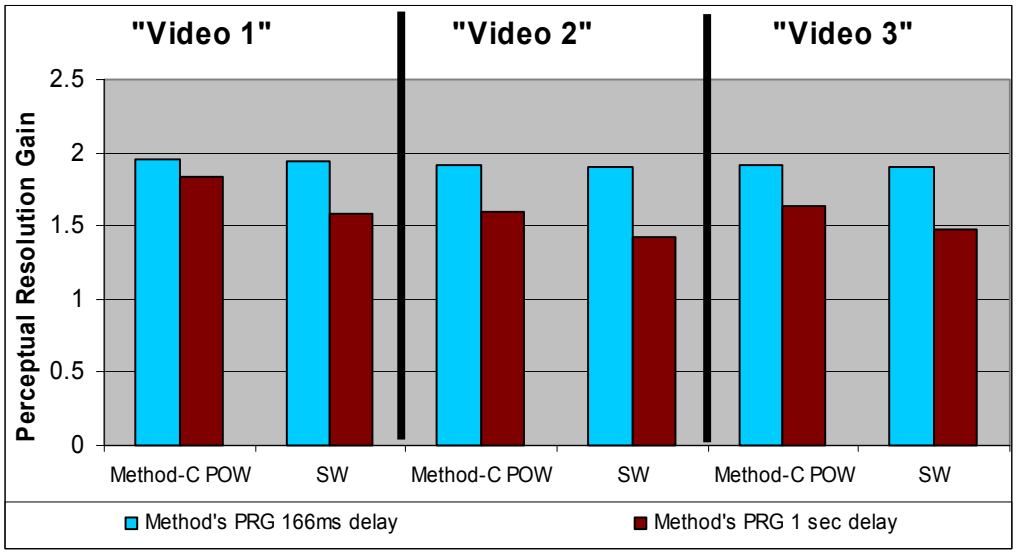

Figure 6.3.4 **"**Video 1". Perceptual resolution gain for saccade window and Method-C perceptual object window methods for "Video 1", "Video 2", "Video 3". Feedback delay is 1 sec.

Author Biographies

Javed Khan

Dr. Javed I. Khan is currently an Associate Professor at Kent State University, Ohio. He has received his PhD from the University of Hawaii and B.Sc. from Bangladesh University of Engineering & Technology (BUET).

His research interest includes intelligent networking and advanced network based applications and systems, and perceptual information processing. His research has been funded by US Defense Advanced Research Project Agency and National Science Foundation. He has also worked at NASA for Space Communication Team. He is member of ACM, IEEE and Internet Society. More information about Dr. Khan's research can be found at medianet.kent.edu

Oleg Komogortsev

Oleg Komogortsev received a B.S. Applied Math degree in 2000 from Volgograd State University, Russia, M.S. degree in 2003 from the Kent State University, Ohio. Currently he is doing his Ph.D. in Computer Science at Kent State University, Ohio.

From January 2001 to December 2001 he was a research assistant at the department Computer Science at Kent State University. From January 2002 till present he is a graduate assistant teaching courses and performing research duties for the department of the Computer Science at Kent State University. He received Ohio Board of Reagents Research Assistantships Award for Fall 2003 – Summer 2004. His research interests are networking, perceptual media compression and adaptation, perceptual attention focus estimation and prediction, human computer interaction.