Proceedings of the ACM International Conference on Multimedia, ACM MM04
New York, October 2004, best student paper contender, pp.220-227

Predictive Perceptual Compression for Real Time Video Communication

Oleg Komogortsev
Kent State University
Computer Science Department
okomogor@cs.kent.edu

Javed Khan
Kent State University
Computer Science Department
javed@kent.edu

ABSTRACT
Approximately 2 degrees in our 140 degree vision span has sharp vision. Many researchers have been fascinated by the idea of eye-tracking integrated perceptual compression of an image or video, yet any practical system has yet to emerge. The unique challenge presented by real time perceptual video streaming is how to handle the fast nature of the human eye and provide its integration with computationally intensive video transcoding scheme. Difficulty arises due to the delay introduced by video transmission in the network. This delay creates a problem when we try to use information about eye movements for perceptual encoding. In this paper we discuss a new approach to the eye-tracker based video compression. Rather than relying on the point of gaze, this novel scheme tracks a vicinity of interest and offers a prediction mechanism for eye movements. The described system compensates the interim eye movements between the sampling and actual coding. The proposed scheme can be applied to a large variety of today's video compression standards. We have developed an eye gaze-aware MPEG-2 transcoder that can perceptually re-encode a live video stream in real time. The experiments we have conducted illustrate the substantial impact this integrated prediction method has on perceptual video compression and bit-rate reduction.

Keywords
Perceptual compression, video transcoding.

1 INTRODUCTION
Human vision offers a tremendous scope of perceptual data reduction. The diameter of the highest acuity - fovea subtends only to 2 degrees. The parafovea (next highest acuity zone) extends to about 4 to 5 degrees, and acuity drops off sharply beyond [4]. A fascinating body of research exists in vision and psychology geared towards the understanding of the human visual perception system. A number of previous attempts have studied how the acuteness degrades around the gaze points. An important factor in such integrated perceptual encoding is the delay between the time an eye-gaze can be tracked and the time the coding response arrives on the screen. This delay is particularly significant in systems that involve network transmission. The delay value increases due to computational overhead when large format media is going to be perceptually encoded. In this paper we propose a solution which allows for coping with such a delay during real time video transmission with perceptual encoding. The ability of a perceptual video transcoding system to compensate for the delay is extremely important in applications where the video is transmitted in real-time through the high delay link. Let us imagine the situation where a Moon exploration vehicle, which is capable of capturing high definition video and transmitting it to Earth in real-time is being controlled by a video data that is transmitted to the operator who wears eye-tracking equipment. This scenario introduces the need for video bandwidth reduction for which perceptual encoding can be used to achieve a significant video compression on top of a standard (MPEG-2/MPEG-4) compression scheme. It also takes barely over a second for a light signal to reach the Moon from the Earth. In that case, the delay for the video transmission will be one second or higher. This delay should be compensated for by the transcoder to provide same viewing experience to the operator as if the video stream wasn't perceptually compressed.

1.1 Previous Works
Since the era of digital imaging, researches have investigated the potential of perceptual video compression. Contrast Sensitivity Function (CSF) and spatial degradation around the foveation center is an area of active research. The issue of perceptual quality loss in respect to these models was investigated in [2, 8, 10,15]. These studies observed a potential for bandwidth reduction as high as 94.7%.

Some techniques for variable spatial resolution coding have been suggested. Examples include Wavelet-based Spatial Coding [11, 12], Retinal Coding [8]. Several investigations of CSF and coding techniques were done for videos in particular [3, 6, 9, 12, 13, 14]. Daly Scott [1] used image analysis instead of eye-gaze to detect the area of visual attention. Geisler [3] presented pyramid coding with a pointing device to identify focus. Khan [6] suggested mouse driven video transformation for medical visualization. Recently Wang [12] discussed a solution to the frame prediction problem found in compressed DCT domain transcoding techniques. Lee [9] discussed how to optimally control the bit-rate for MPEG-4/ H.263 stream for foveated encoding.

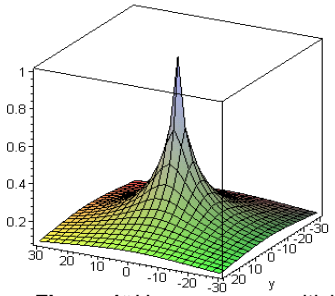

Figure 1. Human eye sensitivity distribution.

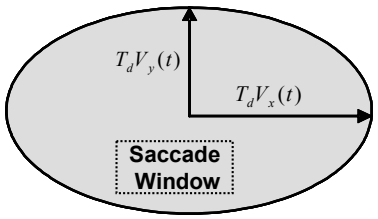

Figure 2. Saccade Window diagram.

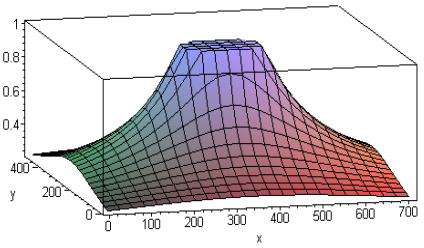

Figure 3. Visual Window in 3D.

1.2 Impact of Feedback Delay & Eye Speed

The CSF oriented studies are mostly point or gazes based, and have yet to incorporate dynamic eye velocity with feedback delay into a video transcoding model. Feedback delay is the period of time between the instance when the eye position is detected by an eye tracker, and the moment when perceptually encoded frame is displayed. This delay should be taken into consideration while using real time perceptual media transcoding. This concern is important because future eye movements should fall within the region of video with the highest quality. Only then would a subject not be able to detect video spatial degradation used for perceptual coding. It is noteworthy that the properties of video transmission might change over time thus increasing or decreasing feedback delay length. A typical network delay ranges from 20ms to a few seconds. Saccades can move the eye position more than 10-100 degrees during that time, while potentially reducing the advantage of designing an accurate parafoveal window within the 2 degrees of eye fixation point. Our research focuses specifically on that issue. It has been proven to be very challenging to predict individual eye gazes. Thus, a point-gaze based CSF coding system studied traditionally will be very difficult to extend for real time transcoding.

1.3 Our Approach

We show a new approach that can withstand dynamically varying delay. Here, instead of relying only on the eye sensitivity matching model, we propose a coding scheme based on a saccade window that has greater stability than individual point gazes. Then we show an integrated approach of gaze proximity prediction and containment, where the eye speed information is used to construct a saccade window.

Based on this approach, we have implemented a real-time foveation aware perceptual video streaming transcoder with a fixed target bit-rate. Our transcoder system intakes live perceptual information related to the subject's eye position. Head movement is monitored via an eye-tracker, and a magnetic head-tracker. Using the collected data, our system controls the spatio-temporal resolution of the video. The eye-tracker tracks the eye-gaze with respect to a human head. The magnetic head-tracker detects the movement of the head with respect to a scene plane. The eye-tracker and magnetic head-tracker together determine the eye movements with respect to the video plane. The system uses a new full-logic MPEG-2 high-resolution region-based motion-vector reprocessing transcoder (which is drift free) [5].

2 SACCADE WINDOWING

2.1 Human Visual Dynamics

Scientists have identified several intricate types of eye movements such as drift, saccade, fixation, smooth pursuit eye-movement, involuntary saccades. Among them, the following two play important role in the design of the proposed system: (i) Saccades: are identical and simultaneous very rapid rotations of the eyes that occur between two points of fixations, and (ii) Fixations: eye movements that take place when the object of perception is stationary relative to the observer's head. Eye fixations also consist of small involuntary saccades, drift, and tremor.

2.2 Overview of the Scheme

Our initial approach was to derive a parafoveal window based on the eye sensitivity function. This was followed by a correction that takes into consideration saccadic eye movements between the time the eye position is tracked and the moment when the perceptually encoded frame would be seen by a subject. We define three types of windows:

$W^{EW}(t)$ is an eye sensitivity window. $W^{EW}(t)$ ensures that bits on the visual plane are distributed according to eye sensitivity function. Sensitivity function corresponds to the eye acuity distribution around an eye fixation point.

$W^{SW}(t)$ is a *saccade window*. $W^{SW}(t)$ represents an area where eye-gazes are contained and sensitivity windows are placed on the border of that area.

$W^{VW}(t)=f(W^{EW}(t),W^{SW}(t))$ is a *visual window*. $W^{VW}(t)$ is a combination of $W^{EW}(t)$ and $W^{SW}(t)$. The idea of visual window is to create an undetectable perceptual compression using eye sensitivity function and address the issue of the feedback delay.

All three windows are described in the following three sections in more detail.

2.3 Eye Sensitivity Window

Numerous functions have been suggested for the eye sensitivity function (SF). For this experiment we have used SF described by equation 2.3.1 (a variation of the SF presented by [1]). This SF is based on entropy losses in the visual system. Also, this function accounts for the issues of cones, rods, and ganglion cells distributions. Figure 1 shows how SF looks like on the visual plane if eye fixation is located in the center of an image.

$$S(x,y) = \frac{1}{1 + k_{ECC} \cdot \theta_E(x,y)} \qquad (2.3.1)$$

Here S is the visual sensitivity as function, k_{ECC} is a constant (in this model k_{ECC}=0.24), and $\theta_E(x,y)$ is the eccentricity in visual angle. Within any lossy compression scheme, the eye sensitivity S has to be mapped to the spatial degradation functions of a given encoding scheme.

2.4 Saccade Window

The objective of the saccade window is to contain eye fixations by estimating an eye speed due to the saccades. Given a set of past eye samples, the saccade window represents a zone where the eye will be at a certain point in the future from its current position with target likelihood given a specific value of the feedback delay T_d. The acceleration, rotation and de-acceleration involved in ballistic saccades are guided by the 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 unpredictable. Therefore we model the saccadic window as an ellipse centered at the last known eye location, allowing the gaze to take any direction within the acceleration constraints. The size of the ellipse is proportional to the length of the feedback delay. If (x_c,y_c) is the current known eye location. Then $W^{SW}(t)$ is an ellipse with the center at (x_c,y_c) and radial components $x_R=T_dV_x(t)$ and $y_R=T_dV_y(t)$. Saccade Window is presented on Figure 2. T_d is a feedback delay introduced by the network transmission and video transcoding overhead. $V_x(t)$ and $V_y(t)$ are the *gaze containment assured speed* (CAS). CAS is calculated with the help of the past eye speed samples. The algorithm for calculating CAS is described in section 2.6.

2.5 Visual Window

A visual window takes into account both the eye sensitivity distribution as well as the eye motion. The eye speed and the feedback delay value determine the size of the saccade window ellipse. The eye position can be expected anywhere within $W^{SW}(t)$. Figure 3 presents visual window in three dimensions. The top flat area of the surface is created by the saccade window. The slope is created by the eye sensitivity window. We modify equation 2.3.1 to present the eye sensitivity formula for visual window. Visual window sensitivity function should correctly incorporate saccadic window into eye sensitivity distribution. Thus the peak point presented by the eye sensitivity function on the Figure 1 is the ellipse of the saccadic window. That means that any point inside of $W^{SW}(t)$ has sensitivity equal to 1. Thus modified Eq. 2.3.1 becomes:

$$S(x,y)=1 \quad when \quad \sqrt{\left(\frac{x-x_x}{v_x(t)/v_y(t)}\right)^2-(y-y_c)^2}-<y_R, otherwise, \qquad (2.5.1)$$

$$=\frac{1}{1+k_{ECC}\cdot\frac{180}{\pi}\arctan\left(\frac{\sqrt{\left(\frac{x-x_C}{V_x(t)/V_y(t)}\right)^2+(y-y_C)^2}-V_y(t)\cdot T_d}{VD}\right)}$$

Use of visual window sensitivity function in perceptual video compression decreases overall bandwidth while maintaining same perceptual quality. Performance results are presented in section 3.6.

2.6 Eye Speed Prediction

We base the CAS calculation on the past eye positional variances. The idea behind our algorithm is to estimate an average eye's speed over a fixed period of time and use the result in estimation of the future eye's speed. We use the following percentile approach to determine this. Suppose there are n eye samples detected during frame "t". Each eye sample $e(t_i)$ has (x_i,y_i) position on the frame F(t) (position is measured in units of pixels). We introduce *running average eye speed* (RAS) as:

$$VR_x(t)=\sum_{i=1}^{n-1}|x_{i+1}(t-T_F)-x_i(t-T_F)| \qquad (2.6.1)$$

$$VR_y(t)=\sum_{i=1}^{n-1}|y_{i+1}(t-T_F)-y_i(t-T_F)| \qquad (2.6.2)$$

Here "n" is the number of eye samples on a particular frame "t". "n" can vary per frame. T_F is the value of the feedback delay in the system measured in frames. $T_F=T_d/FrRt$. T_d is the value of the feedback delay in the system measured in seconds. FrRt is the transcoder's current frame rate per second. Notation $x_i(t-T_F)$ and $y_i(t-T_F)$ mean that eye samples that transcoder received for frame F(t) are T_d seconds late. Thus delayed eye gazes are represented by coordinates $x_i(t-T_F)$ and $x_i(t-T_F)$, where $1\le i\le n$, and n is number of eye-gazes received by the transcoder from the eye-tracker while encoding frame F(t). The coordinates of the $W^{SW}(t)$ center for the frame F(t) would be $x_n(t-T_F)$ and $y_n(t-T_F)$. Let, λ be *target eye gaze containment (TGC)* - the percentage of eye gazes requested to be contained in the saccade window. We calculate CAS values $V_x(t)$ and $V_y(t)$ based on the past m RAS samples. We use notation RASs=m. As a result we work with two sets of RAS samples $\{V_x^R(t-RASs)\ ,..,\ V_x^R(t)\}$ and $\{V_y^R(t-RASs)\ ,..,\ V_y^R(t)\}$. We sort both sets in the ascending order creating two new sorted sets $\{V_x^S(t-RASs)\ ,..,\ V_x^S(t)\}$ and $\{V_y^S(t-RASs)\ ,..,\ V_y^S(t)\}$. After it is done CAS is calculated in the following way:

$$V_x(t)=V_x^S\left(\left\lfloor\frac{\lambda}{100}RASs\right\rfloor\right) \qquad (2.6.3)$$

$$V_y(t)=V_y^S\left(\left\lfloor\frac{\lambda}{100}RASs\right\rfloor\right) \qquad (2.6.4)$$

CAS values are recalculated every frame. When given "m" is a large number and current frame number "t" is less than $m+T_F$, then current RASs for the frame F(t) equals to $t-T_F$. In this case RASs value increases by one every frame until the frame number $t=m+T_F$ is reached to make RASs=m. As we can see from CAS calculations the size of the saccade window is dynamic and might change every frame.

2.7 Impact of the RAS on CAS

Our model considers "m" RAS samples so that it encompasses at least one eye sample from saccade latency, acceleration, di-acceleration, and fixation or pursuit within that period. The parameter "m" impacts performance results greatly. In our experiments we have chosen two extreme cases when the RASs or "m" equals to 20 and 2000. Section 3.6 provides resulting data.

3 EXPERIMENT RESULTS

Our transcoding system was implemented with integrated Applied Science Laboratories high speed eye tracker model 501. The eye position capturing camera worked at the rate of 120 samples per second. The resolution of the test videos was 720x480 pixels. Each video was projected onto a projection screen in a dark room. The projected physical dimensions of the image were: width 60 inches, height 50 inches. The distance between subject's eyes and the surface of the screen was about 100-120 inches. We selected three video clips with different content to provide a good challenge to our algorithm for accurate performance evaluation. We performed a uniform (through standard MPEG bit-rate control mechanism) as well as a perceptual bit-rate reduction from 10 Mb/s to 1 Mb/s. A diagram presenting $W^{SW}(t)$'s boundary is superimposed on the video frames. The area inside of the saccadic window is encoded with

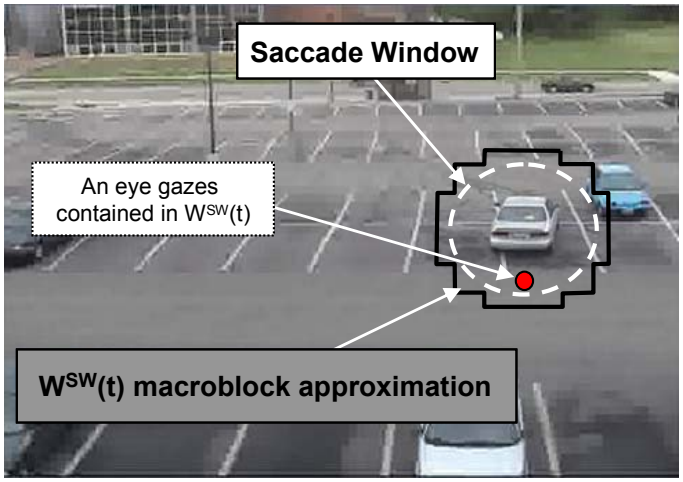

Figure 4a. "Car" – perceptual bit-rate reduction. The only eye gaze is contained in side of $W^{SW}(t)$. Target bit-rate is 1Mb/s.

Figure 4b. "Car" – uniform bit-rate reduction. Target bit-rate is 1Mb/s.

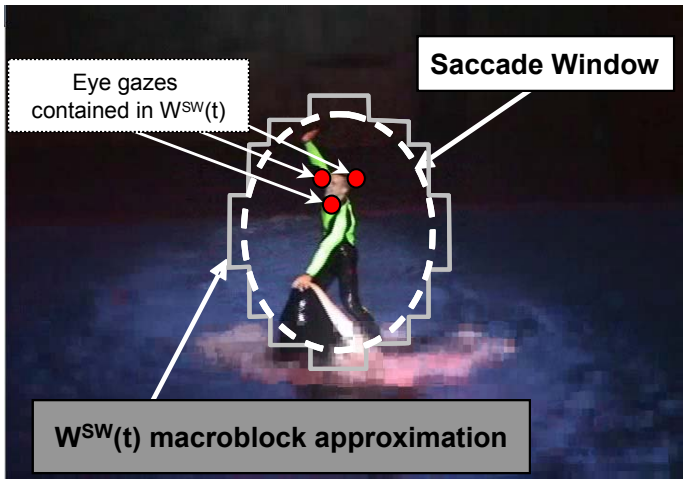

Figure 5a. "Shamu" – perceptual bit-rate reduction. All eye gazes are contained in side of $W^{SW}(t)$. Target bit-rate is 1Mb/s.

Figure 5b. "Shamu" – uniform bit-rate reduction. Target bit-rate is 1Mb/s.

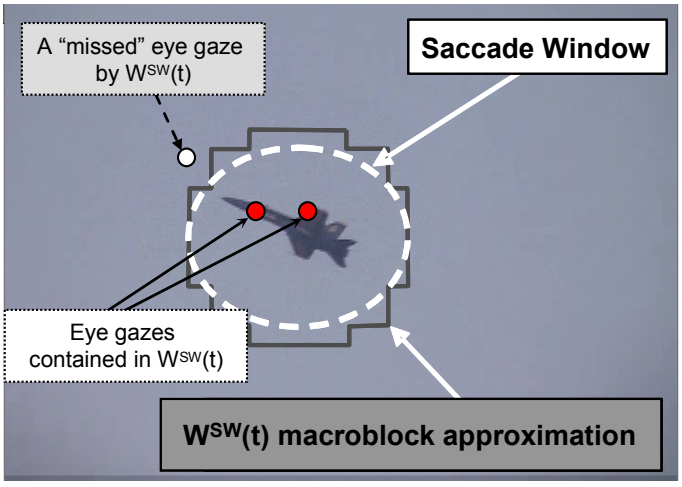

Figure 6a. "Airplanes" – perceptual bit-rate reduction. Two eye gazes are contained in side of $W^{SW}(t)$. One eye gaze is "missed". Target bit-rate is 1Mb/s.

Figure 6b. "Airplanes" – uniform bit-rate reduction. Target bit-rate is 1Mb/s.

high resolution while the rest of the frame is encoded with lower resolution.

3.1 Impact of Scene Complexity:

Human eye movements are highly dependent on the video content. Inherently, some types of scenes offer more opportunity for compression and some offer less. A compression algorithm should continuously analyze the complexity of a scene and provide the best performance possible. Unfortunately, there is no easy or agreed means of measuring the complexity of the content. With the presence of subjective impact a gross average performance is generally not meaningful. We have performed case-by-case analyses of 10 video clips, each offering different combination of subjective complexities. In this paper we consider 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 the university parking lot. The visible size of the car was approximately one fifth of the screen. The car was moving slowly, letting subject to develop smooth pursuit movement (our assumption). Nothing on the periphery of the main object on the video distracts subject attention. Video's background is still. Video duration was 1min 10sec.
- Shamu: This video captures an evening performance of Shamu at a Sea World, during night time under a tracking spotlight. This video consists of several moving objects: Shamu, trainer, and the crowd. Each of them moving at the different speed during various periods of time. The interesting aspect of this video is that a subject can concentrate on different objects and it would result in variety of eye-movements: fixations, saccades, pursuit. The background of the video is constantly moving due to the fact that camera was trying to follow moving Shamu. Such environment suits the goal of challenging our algorithm to deal with different types of eye movements. The fact that the clip was taken during the night provides an interesting aspect of the video perception by a subject. Video duration was 2 min.
- Airplanes: This video depicts formation flying of supersonic planes – performed by Blue Angels on Lake Erie, rapidly changing their flying speeds. Number of planes varies from one to five during the clip. Scene recording camera movements were: rapid zoom and panning. This video provides a challenge for our algorithm to build a compact window to contain rapid eye-movements of the saccades and pursuit. Sometimes camera could not focus very well on a plane and subject had to search for it. This aspect brought additional complication to the general pattern of eye movements. Background of this video is in the constant motion and presents a blue sky. Video duration was around 1 min.

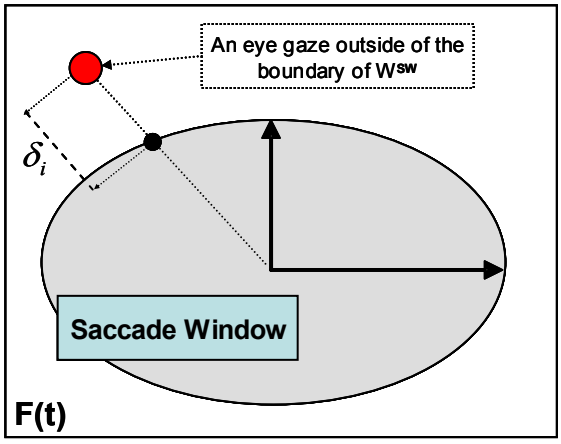

Figure 7. Gaze deviation example.

Figures 4-6 show two snapshots from each of the three test videos. They show perceptual bit-rate reduction as well as uniform bit reduction. It is possible to see that the boundaries of the objects are blurred due to uniform bit-rate reduction on the Figures 4b-6b, but are clear on Figures 4a-6a which are perceptually encoded. The snapshots also present the idea of saccade window and eye gazes in correlation to it. The original video clips, their encoded versions, and videos with eye gaze superimposed on them are available on our website [7].

We have designed several objective measures to estimate our system performance, as described in the next two sections.

3.2 Gaze Containment

Ideally, if the main bulk of eye gazes falls within the saccade window, then it is possible to design optimum perceptual encoder, which ensures undetectable perceptual compression. Thus, we define the quantity *average* eye *gaze containment (AEGC)* as the fraction of gazes successfully contained within a saccade window:

$$\xi = \frac{100}{N}\sum_{t=1}^{N}\frac{\left|\mathrm{E}^{\mathrm{SW}}(t)\right|}{\left|E(t)\right|} \qquad (3.2.1)$$

Where, E(t) is the entire sample set and $E^{SW}(t) \subseteq E(t)$ is the sample subset contained within the saccade window $W^{SW}(t)$ for the frame F(t). N is the number of frames in a video sample.

3.3 Gaze Deviation

While eye gaze containment displays the hits and misses, we were further interested to see how far-off were the misses. We defined a quantity called *gaze deviation*. Gaze deviation δ_i is the distance between an eye gaze $e_i(t)$ and the boundary of the W^{SW}. Gaze deviation is measured in pixels. The idea is presented in Figure 7.

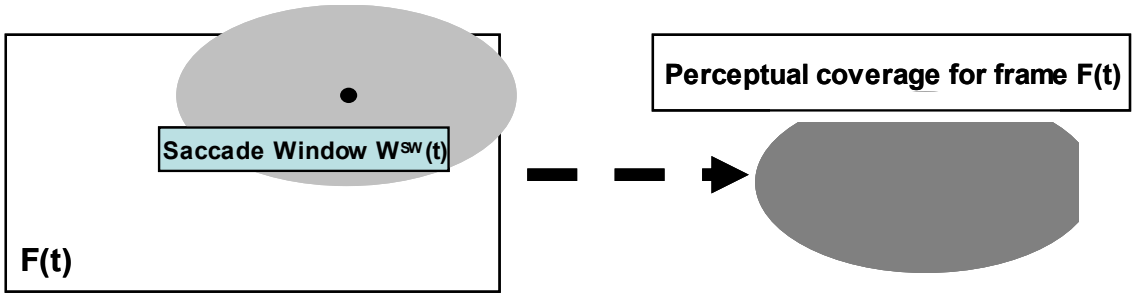

Figure 8. Perceptual coverage calculation example for frame F(t)

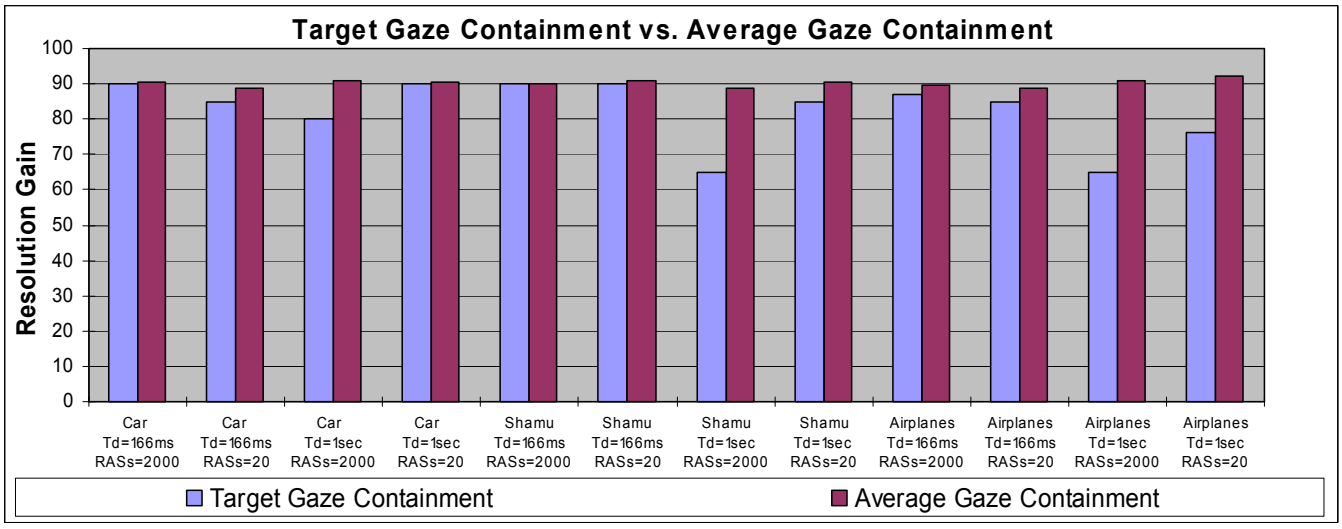

Figure 9. Target gaze containment values to achieve average gaze containment of approximately 90%. Presented for all test videos different delay scenarios and number of RASs considered.

Average gaze deviation is calculated for all gaze samples which fall outside of the saccade window with the distance more than one pixel.

$$\mu = \frac{1}{P}\sum_{i=1}^{P}\delta_i \qquad (3.3.1)$$

Here δ_i is deviation for the eye gaze $e_i(t)$. P is the number of eye gazes which fell outside of saccade window during the test.

3.4 Perceptual Coverage

Another important design goal of our transcoding system was to reduce the size of the saccade window, as there would not be any perceptual redundancy to extract if its size was too large. We have defined a second performance quantity called *average perceptual coverage*. Average perceptual coverage is a percentage of the video image covered by a saccade window, thus requiring high resolution coding in that region. If F(t) is the size of the total viewing frame, and $W^{SW}(t)$ is the predicted saccade window, then the average perceptual coverage is given by equation (delta for area or volume):

$$\chi = \frac{100}{N}\sum_{t=1}^{N}\frac{\left|\Delta(W^{SW}(t)\cap F(t))\right|}{\left|\Delta(F(t))\right|} \qquad (3.4.1)$$

Example of perceptual coverage presented in Figure 8. As indicated, a key factor that determines the size of the saccade window in our algorithm is the *containment assured velocity* (CAV).

3.5 Perceptual Resolution Gain

The actual amount of perceptual compression depends on the two visual window characteristics: the size of the saccade window and allocation of bits from periphery to fovea i.e. sensitivity window. To estimate the advantage of perceptual video compression mathematically for a variable bit-rate transcoder in a delay scenario we introduce the quantity *average perceptual resolution gain* (APRG). APRG is defined as following:

$$APRG = \frac{N*H*W}{\sum_{t=1}^{N}\int_{0}^{W}\int_{0}^{H}S_t(x,y)dxdy} \qquad (3.5.1)$$

N is the number of frames in the video sequence. $S_t(x,y)-$ is the sensitivity function from equation 2.5.1 for the frame "t". W and H are the width and height of the video frame respectively.

3.6 Performance results

Gaze containment: One of the goals of our experiments was to match average eye gaze containment (AGC) with target eye gaze containment (TGC) described in section 2.6. Figure 9 shows the TGC values which make it possible to approximately achieve AGC of 90%. 90% value is chosen, because we feel that it provides us with the best quality/compression ratio. As we can see from the figure in most cases saccade window construction algorithm performed quite well – AGC values were close to those

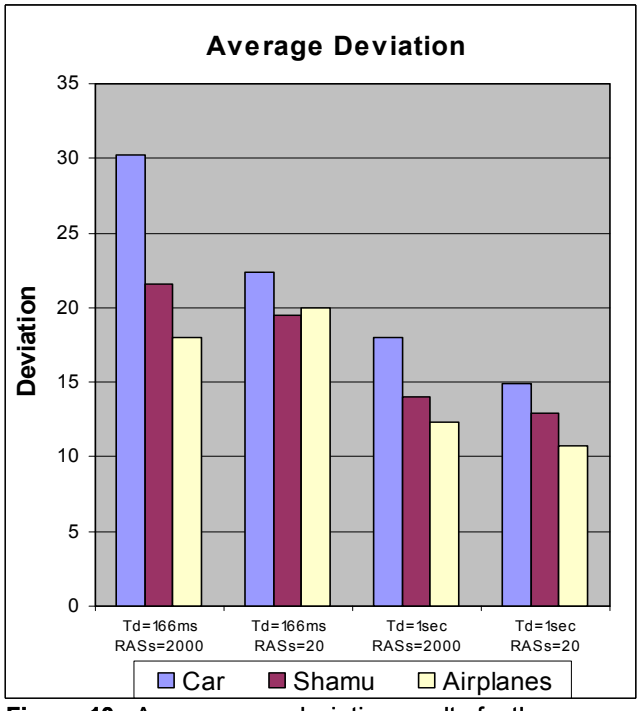

Figure 10. Average gaze deviation results for three test videos.

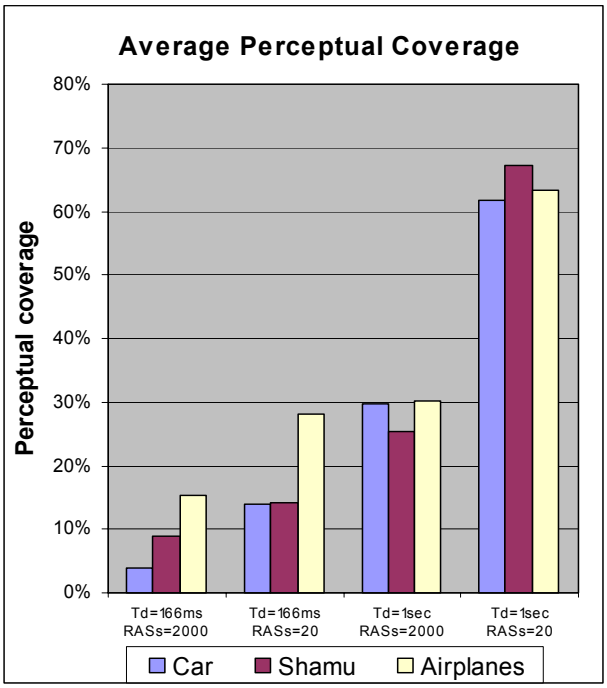

Figure 11. Average perceptual coverage results for three test videos.

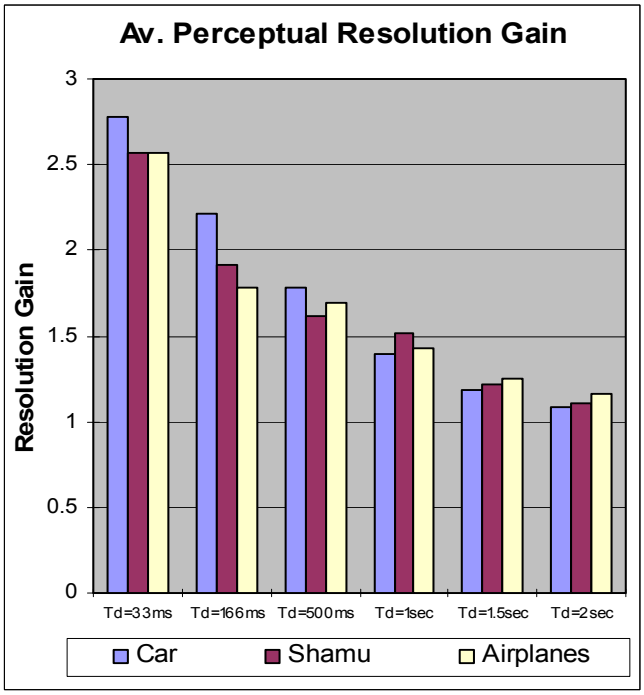

Figure 12. Average perceptual resolution gain results for three test videos.

of TGC. There were few exceptions though. The 1 sec. delay scenario with RASs=2000 provided a challenge to Shamu and Airplanes video. In that case for Shamu TGC had to be 65% to achieve 89% of AGC. For Airplanes TGC had to be 65% for AGC to be 90%.

Figures 10-12 plot performance results described bellow when TGC and AGC are equal to those presented in Figure 9.

Gaze deviation: Average gaze deviation results are presented in the Figure 10. It is interesting to see that for low delay situation deviation numbers are higher than for high delay scenario. It can be explained by the fact that the saccade window size is smallest in the low delay situation. Thus if an eye gaze is not contained by the saccade window chances are that it fell far from the boundary of W^{SW}. In the high feedback delay scenario the saccade window size is large and thus uncontained gaze is usually closer to the W^{SW} boundary. It is possible to see the same tendency in regard to the type of the test video. For the "Car" video where the saccade window size is generally smaller the deviation values are higher than for "Airplanes" video where the saccade window is comparably large. We can also note that the number of RAS samples used for calculating CAS has some impact on the deviation values. For most cases lesser RASs provided better performance, deviation wise, than the case with larger RASs pool. For all videos, delays scenarios and RASs values average gaze deviation stayed in 10-30 pixel range.

Perceptual coverage: Figure 11 shows average perceptual coverage for our test set. We can see that a system operating with about 166 msec. delay would require around 15% of the frame to be encoded with high resolution. In case of 1 sec. delay perceptual coverage goes to 30% and higher. Number of RASs has a significant impact on the perceptual coverage. In the scenario presented, perceptual coverage was the lowest when a large pool (RASs=2000) of running average eye speed samples was considered for CAS calculation. Such tendency is valid for both high delay and low delay scenario. But in case of the 1 sec. delay scenario the difference between perceptual coverage values was between 30% and 65% for different amount of RASs considered. That was a double increase. This can be explained by the fact that 20 is a very low number of RASs to consider in the situation of 1 sec. delay. In this case W^{SW} has a tendency to "lag" behind, because during 1 sec. eye movement behavior can change drastically. The video content also plays significant role in perceptual coverage. We can say that the longer "memory," with a larger number of RAS samples considered brought substantial improvement in the coverage efficiency. "Car" video had smallest average perceptual coverage due to the smooth moving nature of the perceiving object. Perceptual coverage was 4%, 13% for 166 msec. delay scenario and 30%, 62% for 1 sec. delay scenario depending on number of RASs. The performance of the "Shamu" video, with more rapid and complicated object movements, was next best. Average perceptual coverage was 9%, 14% for 166 msec. delay scenario and 25%, 67% for 1 sec. delay scenario depending on number of RASs considered. As expected, "Airplanes" gave the worst performance due to the fast, airplane movements inside of the video. Perceptual coverage was 15%, 27% for 166 msec. delay scenario and 30%, 63% for 1 sec. delay scenario depending on number of RASs considered.

Perceptual resolution gain: Figure 12 shows average resolution gain values for our video test set. APRG value was highest for slow, one object moving video such as "Car". APRG decreased for the "busier" videos such as "Shamu" and "Airplanes". A "busy" scene with multiple perceptual activities will naturally offer less opportunity for perceptual compression, due to the more rapid eye movements and thus larger saccade window. The APRG values were between 2.5 and 2.7 for 33 msec. delay, 1.7 and 2.2 for 166 msec. delay scenario and between 1.3 and 1.5 for 1 sec. delay. It is possible to see that the higher feedback delay was the smaller APRG value was achieved. APRG was very close to 1 in case of 2 sec. delay. It means that our perceptual compression technique would beneficial up to a certain delay value. We should also mention that actual compression values

will depend on the particular encoding scheme. It is worth saying that a lot of modern codec's can encode still part of the background with a very few bits and can reduce overall bandwidth that way, but in the case of a video where everything is moving (such as presented Shamu video) modern codec will fail to reduce the bandwidth without quality loss. In such scenario our technique can provide a specific region which requires high quality coding. That can be seen from the Figure 12, where APRG values for the videos with amoving background (Shamu, Airplanes) are almost the same and sometimes better, than of Car video with a still background.

4 CONCLUSIONS & CURRENT WORK

Perceptual techniques will be critical to achieve transcoding/compression ratio needed in emerging applications. The eye tracker can formulate object-based encoding might be extremely viable in some scenarios: remote vehicles control and operation, remote surgery assistance, virtual reality teleporting. However, there are several challenges yet to overcome before such a concept system can become a reality. The results of this system suggest some interesting discourse from contemporary research. The feedback delay in control loop (in network, in media encoding, or even the delay within the eye-tracker) creates an area that we call a saccade window which is several times larger than the para-fovea area studied by CSF researchers. We suspect, in the case of live coding, a simple approximation of eye sensitivity function will bring same performance results as a sophisticated one.

Another important aspect of the proposed approach is that it is media independent. Many of the point-gaze based research deeply integrate foveation schemes with the media. In contrast, we proposed the visual window as a virtual area superposed on the rendering plane of any video media. If a saccade window 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. Mapping of eye sensitivity to bit-allocation is a separate problem by its own merit. This would require perceptual tests involving loss perception. Though this is not the main focus of this paper, but we used a transcoding scheme given in [5]. It can be easily changed if needed. The actual bit-saving will depend on this coding model. However, the perceptual resolution gain can be determined by estimating the sensitivity distribution over the video plane, which can bring 50% and more additional compressibility depending on a particular encoding scheme. The work is currently being funded by DARPA Research Grant F30602-99-1-0515.

5 ACKNOWLEDGMENTS

Our special thanks to Jessica Erin for help in preparation of this manuscript.

6 REFERENCES

[1] Daly, S. J., Matthews, K., Ribas-Corbera, J. *Visual eccentricity models in face-based video compression.* In Human Vision and Electronic Imaging IV, May 1995, SPIE.

[2] Duchowski, A. T. *Acuity-Matching Resolution Degradation Through Wavelet Coefficient Scaling.* IEEE Transactions on Image Processing 9, 8. August 2000.

[3] Geisler, W. S., Perry, J. S. *Real-time Foveated Multiresolution System for Low-bandwidth Video Communication.* In Human Vision and Electronic Imaging III, July 1998, SPIE.

[4] 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.

[5] 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,

[6] Javed, K., Yun D. *Multi Resolution Perceptual Encoding for Interactive Image Sharing in Remote Tele-Diagnostics, Manufacturing Agility and Hybrid Automation –I.* In Proceedings of the International Conference on Human Aspects of Advanced Manufacturing: Agility & Hybrid Automation, HAAMAHA'96, Hawaii, Aug. 1996, pp183-187.

[7] Komogortsev, O., Khan, J., *Video Set for Predictive Perceptual Compression Test.* At *www.cs.kent.edu/~okomogor/ACM04VideoSet.htm.*

[8] Kuyel, T., Geisler, W. S., Ghosh, J. *Retinally reconstructed images (RRIs): digital images having a resolution match with the human eye.* In Human Vision and Electronic Imaging III, July *1998, SPIE* .

[9] 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.

[10] Loschky, L., McConkie, G. *User performance with gaze contingent multiresolutional displays.* In Proceedings of the symposium on Eye tracking research & applications. November 2000.

[11] Niu, E. L. *Gaze-based video compression using wavelets.* M.S. Thesis. University of Illinois at Urbana-Champaign. The Graduate College. August 1995.

[12] Wang, Z., Lu, L., and Bovik, A. *Rate scalable video coding using a foveation-based human visual system model.* IEEE International Conference on Acoustics, Speech, & Signal Processing, May 2001.

[13] Westen, S. J., Lagendijk, R., Biemond, J. *Spatio-temporal model of human vision for digital video compression.* In Human Vision and Electronic Imaging II, June 1997, SPIE.

[14] Yoon, S., Ratakonda, K., Ahuja, N. *Region-Based Video Coding Using A Multiscale Image Segmentation.* In Proceedings of 1997 International Conference on Image Processing (ICIP '97).

[15] Kortum, P., Geisler, W. S., *Implementation of a Foveated Image Coding System for Image Bandwidth Reduction.* In Human Vision and Electronic Imaging, April 1996, SPIE.