Published at Proceedings of
The 13th IEEE International Packet Video Workshop 2003
Nantes, France, April 2003

FLOCK-OF-BIRD ALGORITHM FOR FAST MOTION BASED OBJECT TRACKING AND TRANSCODING IN VIDEO STREAMING

Javed I. Khan and Zhong Guo

Media Communications and Networking Research Laboratory
Department of Computer Science
Kent State University, Kent, OH 44242
javed|zguo@kent.edu

ABSTRACT

Object based bit allocation enables video coding with high perceptual quality at significantly reduced bit-rate. In this paper, we describe a fast content aware video transcoding technique particularly designed for streaming video. The algorithm can accept an initial high-level description of the expected video objects. The proposed flock-or-bird algorithm then detects and tracks these target scene objects in real time. It uses macro-blocks group property for detection of video objects. Based on these moving objects it dynamically re-encodes a regular coded video stream into a perceptually coded video stream at a significantly reduced bit-rate. In this paper, we share the performance of its MPEG-2 compliant implementation.

Keywords: *perceptual encoding, video transcoding, content aware streaming.*

1. INTRODUCTION

The speed of network is increasing. However, with it the asymmetry in the Internet capacity is also growing dramatically. Increase of more multimedia communication oriented software, along with the impending adoption of high fidelity digital video standard such as DTV or HDTV is expected to generate an enormous flux of video content. However, the relatively slow increase of bandwidth at network edges and the advent of small devices (such as Personal Digital Assistant) seems to indicate that in near future the Internet applications have to deal with increased bandwidth asymmetry. Consequently, fast video transcoding with high transcoding ratio is expected to become an increasingly important problem for video streaming. Current video transcoding techniques are based on requantization [8,13, 16-20]. Unfortunately, requantization alone does not have enough capability to down scale a video to the very low bit-rate required by the current Internet. Research has shown that object based encoding can play an increasingly important tool for creating perceptually pleasant video at lower rates [1,2,5,9,12]. MPEG-4 has been proposed to transport object-based coded video stream. However, conversion of a regular video to an object-based stream remains an open challenge.

In this research we explore a perceptual object based video transcoding coding scheme that can provide substantial rate reduction capability at relatively high speed. The scheme is applicable for several content-based transcoding scenarios. These include transparent MPEG-2 to /MPEG-2 perceptual transcoding or converted MPEG-2 to MPEG-4 object based transcoding.

1.1 Related Works

Despite years of research in image processing, and the standardization of the syntax such as MPEG-4, object detection still is an illusive open problem [1-5]. Apparently, the first generation MPEG-4 [11] will only be available for computer generated synthetic video with model objects. For natural video its success is still limited to small format videos with known content model [1, 2] (such as head and shoulder video). Object detection algorithms for those special cases are also generally quite computation intensive [5].

Several methods have been investigated for video object detection. Most of them have been investigated under scene comprehension research. However, an interesting observation is that most has been conceived as an extension of object detection algorithms found in still image processing [14, 15]. Among the recent methods Ngo et. al. [14] described object detection based on motion and color features using histogram analysis. This technique could process less than two frames in one second. Unfortunately, most of the other techniques presented such as [15] did not provide any evaluation of their method’s time performance or measure of their tracking efficiency. These algorithms employ even more involved image processing such as active contour model, and may not be any faster. Recently few relatively faster partially compressed domain techniques have been proposed [3, 4]. For example [3] derives texture, spatial, temporal, and directional confidence measures from incoming stream based on DCT coefficients, motion vectors and spatial/temporal continuity of motion, buffering three adjacent frames. To identify multiple objects, they however, perform k-means and/or EM clustering based on spatial features. It achieved about 0.5 sec/frame for the CIF video. These systems however target video indexing or video description application rather than transcoding.

1.2 Transcoding vs. Encoding

The transcoding scenario has some notable differences from first stage encoding. (i) Most first stage encoding scenarios (except for live video) allow off-line processing. In contrast, in transcoding the object detection has to be performed extremely fast at the rate of the stream. (iii) Secondly, original pixel level information is no longer explicitly available. The transcoder receives an encoded video stream. It is expensive if an algorithm depends on them. (iii) Thirdly, however, the input stream typically contains highly structured coded and refined information such as motion vectors.

Consequently, techniques for transcoding are generally applicable in first stage encoding but the reverse in not always possible. There have been very little previous work that has explored techniques particularly for object based perceptual transcoder for streaming. However, with the advent of Internet video it is expected to become an important issue. In this research we focus on this particular problem and describe a low computation based object tracking algorithm particularly suitable for focus region based perceptual video coded stream recompression.

1.3 Our approach

First, to conform to the constraints of the transcoding scenario, we have restricted the problem that no pixel level decoding is allowed in detection and tracking. Thus, the detection algorithm is strictly a macro-block level compressed domain algorithm. In an MPEG-2 ISO-13818 [6] stream, it corresponds to video transport stream decoding only up to picture sequence and use of up to macro-block fields. It does not require and inverse DCT. Then we show that two characteristics of transcoding can be utilized to reduce the computational cost dramatically.

First, we observe that region-based recompression usually does not require precise specification of the object boundary. A region slightly larger than the interested object not only s acceptable rather is more desirable. Because, in region based perceptual encoding the human eye in effect scans both inside and outside areas near boundaries. Notably, most of the previous approaches has been derived from scene interpretation applications, and thus spend enormous effort is perfecting the boundaries and shapes. It is highly likely that an object detection technique for perceptual compression thus may not gain by paying the extra cost for detection precision. The algorithm we propose takes advantage of this approximation. Secondly, we observe that some useful information (such as motion vectors) have been made available in the stream by the original encoder's computation. Thus, the proposed algorithm uses this compressed domain information and avoid a great amount of processing delay in raw image data.

In addition, this frugal algorithm does not get involved into indiscriminate scene analysis to resolve the ambiguity inherently involved in scene understanding. A principal difficulty of image processing based object detection is to differentiate the target from non-target. Rather it accepts logical description of the expected target video objects in terms of high-level 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. The technique is based on analysis of group properties of the macro-blocks using strictly motion vectors. We call it **Flock-of-Bird** (FOB) algorithm. In this paper we present this fast perceptual transcoding algorithm.

The result presented is not a simulation. It has been implemented as a MPEG-2 based perceptual rate transcoder[1]. As a real implementation, the system faces a number of open challenges besides fast object detection. For example, bit-allocation models in objects and its impact on perceived video quality, smooth rate control beyond TM-5, coding scheme suitable for spatially varying images and video, etc. Each of these areas is subject of active research. However, the principal emphasis of this paper is the object detection and tracking model for transcoding. We have used specific known implemented solutions for these associated sub-problems. However, the proposed detection algorithm can be used with most other solutions to those associated problems.

Section 2 of this paper first describes the system architecture. Section 3 then presents the object-tracking algorithm. Section

[1] The incoming stream is a high rate MPEG-2 stream. The extracted perceptual objects can be fed into any outgoing stream format (for example MPEG-4). In this specific implementation we have shown that it can be also be fed into an MPEG-2 streaming, which does not have any explicit object representation syntax. If the outgoing format has explicit representation, certain additional syntax advantages can be exploited. But, we choose a fully transparent rate transcoder, where the end-points don't need to know the existence of the transcoder.

4 then presents how the object information is combined with MPEG-2 rate control. Finally, in section 5, we present performance of this system from the real MPEG-2 transcoder experiments.

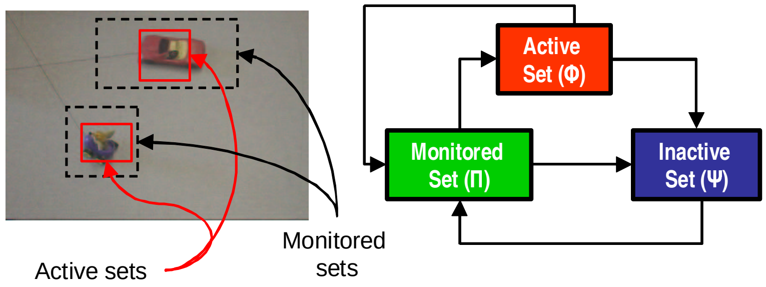

Fig-2(a) (left) Φ , Π sets of macroblocks with a frame with two objects. Ψ is the background. Fig-2(b) (right) Transition is bi-directional between Φ -Π , and Π -Ψ , but unidirectional from Φ toΨ , A macroblock can not can not transition in Φ directly from Ψ .

2. SYSTEM MODEL

2.1 Transcoder Architecture

The overall architecture of the transcoder in is shown in Fig-1. For this experiment, we have implemented a perceptual transcoder system, which both accepts and produces MPEG-2 ISO-13818 [6] video stream. Thus, it is fully transparent and does not require the switch of player or server while the rate transcoding is in effect. The rate transcoder has a full-logic decoder and encoder embedded in it. In between the FOB unit performs the object detection and provides object information to the re-encoder. The re-encoder is capable of dynamically adjusting the incoming bit-rate to an outgoing piece-wise constant bit-rate (pCBR) [7, 10]. The control has been implemented using a double loop feedback rate control mechanism, which is similar to the MPEG-2 TM-5 algorithm. Besides the pCBR operation, the system can modulate the sample density both in temporal as well as spatial dimension based on the detected objects. More detail of the rate control algorithm can be found in [7,12].

3. OBJECT DETECTION & TRACKING MODEL

3.1 The POP Model

We view a video to be a collection of macroblocks arranged both in 2D frame plane and in time axis. A frame is the viewing area in one time instant. Each viewing area is a mosaic of elementary shapes. For MPEG-2, 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. The concept of *frames* and *elementary mosaic* (macroblocks) provide the compression and packing view of a video.

From perceptual processing point of view, we reorganize 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. The projection of an object on the entire video is called *perceptual object projection* of POP. Each POP is defined by a set of descriptors based on some properties of the elementary regions (to be described shortly). Thus we define a frame F_t= {$b_t(i,j)$ all macroblocks at time t}. A video is the union of all frames:

$$V = \left\{ \bigcup_t [F_t] \right\} \qquad \text{……(1a)}$$

In addition, a particular perceptual object is:

$$POP(r) = \left\{ \bigcup_t [R_t(r)] \right\} . \qquad \text{……(1b)}$$

We also define a background macro-block set:

$$B = V - \left\{ \bigcup_r POP(r) \right\} \qquad \text{……(1c)}$$

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.

3.2 Tracking Model

For tracking, we introduce the notion of three mosaic sets. For each POP(r), the frame macro-blocks are classified into three sets *active* ($\Phi_t(r)$), *monitored* ($\Pi_t(r)$) and *inactive* ($\Psi_t(r)$) sets based on mosaic macroblock property analysis. Macroblocks representing the projection of same POP(r) on a frame are grouped as $\Phi_t(r)$={$b_t(i,j)$ where macro-block index i,j covers the POP}. The active set defines the core perceptual object. Macroblocks surrounding the $\Phi_t(r)$ belong to $\Pi_t(r)$ and those beyond are in $\Psi_t(r)$. The membership of the macroblocks in these three sets can change from frame to frame. Fig-2(a) shows the typical active and monitored set and Fig-2(b) shows the transition model. The union of these sets for all objects is the frame set (F_t). Here superscript r is the index of the focal region.

$$\cdot\ F_t = \{\bigcup_{r}[\Phi_t(r) \cup \Pi_t(r)]\} \cup \Psi_t(r) \qquad \ldots\ldots(2)$$

A macroblock, which is currently in Π(r) can join Φ(r). Similarly, a macroblock, which is now in Φ(r) can loose its membership and relegated to Π(r) or Ψ(r) in next frame. The transitions are determined by *set transition rules* defined based on distance measure between the *mosaic property set* (MoPS) and *projection object property set* (PoPS). The MoPS properties of the active and the monitored sets both are continuously monitored. However, the later is not used to update the PoPS. PoPS are the collective property determined by the MoPS in the active set. A macroblock can be member of the active sets of multiple POPs. It has no impact on recognition but it takes the best of their rendering quality attributes.

3.3 Base Property Set

The system accepts a high-level description of an object. For moving object detection in the MPEG-2 stream, we rely on macroblock motion properties. Here the MPD $MPS_t(i,j)= [u^x_t(i,j)\ u^y_t(i,j)]$ where, $u^x_t(i,j)$ and $u^y_t(i,j)$ are the horizontal and vertical motion vectors associated with macroblock $b_t(i,j)$. We also denote the aggregate property of an POD by $APS_t(r) = [\bar{U}^x_t(r), \bar{U}^y_t(r)]$. These two values are the medians of all $u^x_t(i,j)$ and all $u^y_t(i,j)$ respectively, where $b_t(i,j) \in \Phi_t(r)$.

3.4 Germination Descriptors

We define an object with two sets of parameters. The first is the *germination parameters*. These four parameters help in identifying the spontaneous birth and death of POP regions in a frame. These are as following:

Formation Mass ($m_F(r)$): start tracking a focal region for POP(r) as an object if its size is greater than $m_F(r)$ macroblocks.

Formation Velocity ($u_F(r)$): start tracking a focal region for POP(r) as an object if its speed is greater than $u_F(r)$ pixels per frame.

Dissolving Mass ($m_D(r)$): stop tracking POP(r) if the object size is less than $m_D(r)$ macroblocks.

Dissolving Velocity ($u_D(r)$): stop tracking POP(r) if the object speed is less than $u_D(r)$ pixels per frame.

The birth and death of POPs are not symmetric. The birth depends on the MoPS, while the death depends on the PoPs.

3.5 Flocking Descriptors

Once the birth of POP is detected, the POP is brought into the flocking state from germinal state. The flocking state POP in then handed over to the live flock-or-bird tracking process, which tracks the POP using the second set of six parameters called flocking parameters.

- Monitor Span (s_M): the width of $\Pi_t(r)$ in number of macroblocks.
- Deviator Thresholds ($v^x_D(r)$ and $v^y_D(r)$): if the difference in magnitude between the MPS parameter

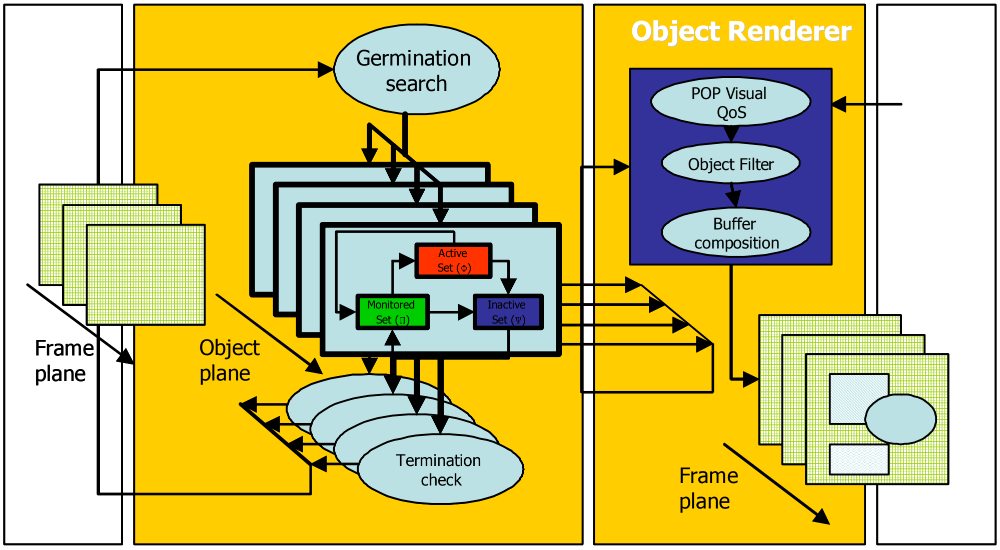

Fig-3. Internal organization of the tracking algorithm.

$u_t^x(i,j)$ of $b_t(i,j) \in \Phi_t(r)$ and APS parameter $\overline{U}_t^x(r)$ is greater than differential $v_D^x(r)$ percent of $\overline{U}_t^x(r)$, or the difference magnitude between $u_t^y(i,j)$ of $b_t(i,j) \in \Phi_t(r)$ and $\overline{U}_t^y(r)$ is greater than differential $v_D^y(r)$ percent of $\overline{U}_t^y(r)$, then $b_t(i,j)$ is not following the movement of the POP(r).

- Follower Thresholds ($v_F^x(r)$ and $v_F^y(r)$): if the difference in magnitude between the MPS parameter $u_t^x(i,j)$ of $b_t(i,j) \in \Pi_t(r)$ and APS parameter $\overline{U}_t^x(r)$ is less than the differential $v_F^x(r)$ percentage of $\overline{U}_t^x(r)$, or the difference magnitude between $u_t^y(i,j)$ of $b_t(i,j) \in \Pi_t(r)$ and $\overline{U}_t^y(r)$ is less than differential $v_F^y(r)$ percent of $\overline{U}_t^x(r)$, $b_t(i,j)$ is following the movement of the POP(r).
- Deviator Persistence ($p_D(r)$): if $b_t(i,j) \in \Phi_t(r)$ has not been following the movement of the POP(r) for consecutively $p_D(r)$ times then $b_t(i,j)$ is a persistent deviator in $\Phi_t(r)$.
- Follower Persistence ($p_F(r)$): if $b_t(i,j) \in \Pi_t(r)$ has been following the movement of POP(r) for consecutively $p_F(r)$ times then $b_t(i,j)$ is a persistent follower in $\Pi_t(r)$.
- Group Volatility ($g_V(r)$): a maximum of $g_V(r)$ percent of macroblocks from $\Phi_t(r)$ can be removed in one frame.

In the tracking phase, each flocking POP is also checked against the dissolve criterion. If it falls below then the tracking stops and the macroblocks are returned to the inactive pool, and the POP is taken back into germinal state.

Each POP is therefore represented by one set of the above eleven parameters called POP descriptor. Optionally a POP descriptor can have a germination frame range, and an instance limiter. A POP descriptor remains active only within the frame range. The default is full video range. In addition, the instance limiter describes how many POP can simultaneously exist with a given POP description. The default is unlimited.

3.6 Inverse Projection Algorithm

A VOD [11] can move into flocking state in two ways. It can either spontaneously detect the birth and death of VODs based on the given germination parameters, and lock it on for tracking. Alternatively, it can accept an explicit specification of VODs with the description of initial VOD position. Both can be done on any P frames during a video sequence. Figure-3 shows the overall internal structure of the tracking algorithm as well as the concept of region based reencoding.

3.7 Germination Search: (Initialize the Active and Monitored sets):

I. Spontaneous locking of new objects

It attempts to generate a new tracking object only when $m_F(r)$ is greater than 0. Search for a new possible tracking object is conducted only on P frames.

1) For all macroblocks $b_t(i,j)$ of a P frame those are not currently in any of the active sets: $b_t(i,j) \notin \left[\bigcup_k \Phi_t(k) \right]$ search from top-left corner of the frame the bottom-right corner. Mark all macroblocks that satisfy the following equation with the label L=r of the germinal VOD.

$$\sqrt{u_t^x(i,j) + u_t^y(i,j)} \geq u_F(r) \qquad \text{….(3)}$$

2) Search through marked macroblocks starting from top-left corner for consecutive regions of marked macroblocks until a region R with more than $m_F(r)$ macroblocks is found or the bottom-right corner is reached.

3) If such a region R is found let all macroblocks in R compose of the initial set $\Phi_0(r)$. For all $b(i,j) \notin \Phi_0(r)$, if it is within $s_M(r)$ macroblocks away from $b(i,j) \in \Phi_0(r)$, let $b(i,j) \in \Pi_0(r)$.

II. Explicit specification of an object position

Given the coordinates of top-left corner pixel (x_1, y_1) and bottom-right corner pixel (x_2, y_2) of a focal region, the corresponding macroblocks to these two points $b(x_1/16, y_1/16)$ and $b(x_2/16, y_2/16)$ are identified. The initial $\Phi_0(r)$ will be composed of all $b(i,j)$ where $y_1/16 \leq i \leq y_2/16$ and $x_1/16 \leq j \leq x_2/16$. If $b(i,j) \notin \Phi_0(r)$, but within $s_M(r)$ macroblocks away from $b(i,j) \in \Phi_0(r)$, let $b(i,j) \in \Pi_0(r)$.

3.8 Tracking

Once an object is detected, it is handed over to a tracking thread. There is one tracking thread corresponding to each POP instance (Figure 3). The tracking starts with the initial

Φ_0(r) and Π_0(r). For each subsequent P frame in the presentation (as well as coding) sequence, Φ_t(r) and Π_t(r) sets are predicted by shifting the Φ_{t-1}(r) and Π_{t-1}(r) sets of the previous P frame's motion analysis using inverse shift of the motion. These sets in I and B frames are computed by back interpolation while the computation follows coding sequence. Below are the steps how each of the objects in each frame is handled (Since, all the steps involve one object therefore we will omit the POP index r):

2. Predict Φ_t and Π_t for frame t from previous frame t′

Let t' is the last P frame before t, then we obtain Φ_t and Π_t sets for frame t by shifting $\Phi_{t'}$ and $\Pi_{t'}$ sets of a previous P frame t′ towards the object's movement direction.

Given $u^x_{t'}(i,j)$ and $u^y_{t'}(i,j)$ of frame t′, since they were forward predicted, the shift direction should be opposite.

If frame t is I or P type, Φ_t and Π_t are predicted by shifting $\Phi_{t'}$ and $\Pi_{t'}$ horizontally and vertically by the following number of macroblocks:

- $-\frac{\overline{U}^x_{t'}(r)}{16}$ and $-\frac{\overline{U}^Y_{t'}(r)}{16}$ ……(4a)

II. If frame t is a B frame, assuming there are n B frames between two adjacent P frames and t is the i^th one, Φ_t and Π_t are predicted by shifting $\Phi_{t'}$ and $\Pi_{t'}$ horizontally and vertically by the following number of macroblocks:

- $-\frac{\overline{U}^x_{t'}(r)\times i}{16\times(n+1)}$ and $-\frac{\overline{U}^Y_{t'}(r)\times i}{16\times(n+1)}$ ……(4b)

Steps 3 to 5 are only performed on P frames:

3. Reshape the current frame Φ_t through motion analysis

Sort $u^x_t(i,j)$ and $u^y_t(i,j)$ of all $b_t(i,j)\in\Phi_t(r)$ choose the median value as $\overline{U}^x_t(r)$ and $\overline{U}^Y_t(r)$ respectively.

For each $b_t(i,j)\in\Phi_t(r)$ if:

- $\frac{|u^x_t(i,j)-\overline{U}^x_t(r)|}{|\overline{U}^x_t(r)|}\ge v^X_D(r)$

or $\frac{|u^Y_t(i,j)-\overline{U}^Y_t(r)|}{|\overline{U}^Y_t(r)|}\ge v^Y_D(r)$

We consider b_t(i,j) is not following the object movement in current frame. If it has been not following in p_D consecutive P frames, we then remove it from Φ_t. But we can only remove at most g_V (group volatility) percent of macroblocks from Φ_t. If more than g_V macroblocks deviates, the g_V percent macroblocks with largest values in following equation will be removed from Φ_t:

- $|u^x_t(i,j)-\overline{U}^x_t(r)|+|u^Y_t(i,j)-\overline{U}^Y_t(r)|$

For each $b_t(i,j)\in\Pi_t(r)$ if:

- $\frac{|u^x_t(i,j)-\overline{U}^x_t(r)|}{|\overline{U}^x_t(r)|}<v^X_F(r)$

or $\frac{|u^Y_t(i,j)-\overline{U}^Y_t(r)|}{|\overline{U}^Y_t(r)|}<v^Y_F(r)$

We consider $b_{t,}$(i,j) is following object movement in current frame. If it has been following in p_F consecutive P frames, we let $b_t(i,j)\in\Phi_t(r)$.

4. Spatial Locality based adjustment to Φ_t :

For each $b_t(i,j)\notin\Phi_t(r)$, if b_t(i-1,j), b_t(i+1,j), b_t(i,j-1), and b_t(i,j+1) all ∈ Φ_t, we let b_t(i,j)∈ Φ_t. Intra coded macroblocks that were removed from Φ_t because of their zero motion vectors can be recovered through this operation.

Symmetrically, for each $b_t(i,j)\in\Phi_t(r)$, if b_t(i-1,j), b_t(i+1,j), b_t(i,j-1), and b_t(i,j+1) all ∉ Φ_t, we let m_t(i,j) ∉ A_t.

5. Tracking validation test

If Φ_t has less than $m_D(r)$ macroblocks or

$$\sqrt{\overline{U}_t^x(r)+\overline{U}_t^y(r)} \le u_D(r) \qquad \text{….(5)}$$

Then this object does not qualify to be tracked anymore. Let all $b_t(i,j) \in \Phi_t(r) \cup \Pi_t$ are returned to $\Psi_t(r)$.

6. Reset Π_t

For all $b_t(x,y) \notin \Phi_t(r)$ but within S_M(r) macroblocks away from any $b_t(i,j) \in \Phi_t(r)$, let $b_t(x,y) \in \Pi_t(r)$.

The above procedure is now repeated for each object. The output of the system is the sequence of active sets $[\Phi_0(r), \Phi_1(r), \Phi_2(r), \cdots]$, which is fed to the transcoder rate controller as video object region. The transcoder then correspondingly generates the pCBR stream with appropriate spatial distribution of bits for the specified outgoing bit-rate.

4. RATE CONTROL

4.1 Transcoder Rate Control Mechanism

Once the objects are detected, the active sets are fed into the following rate spatial rate control mechanism of he re-encoder. MPEG-2 TM-5 already defined a quantity called activity factor for taking into account of perceptual significance of the macroblocks. We refine this handle to perform the spatial varying coding. The proposed mechanism is also a double-loop feedback control mechanism where the output bit-rate is continually sensed to determine overall piecewise constant rate, with appropriate accounting for variations in frame/picture type like TM-5. A second internal feedback loop further tracks the efficacy of key conversion factors/constants for additional stability. Here, the perceptual content and activity in a particular picture area dictates the inherent amount of bits that may be required to encode it. In addition, the bit requirement per macro-block is dependant on the picture type (I, B or P) as well other subjective factors. Like TM-5 the bit-rate is controlled by the requantization-step of the DCT coefficients. The quantized output for intra- and non-intra frames are respectively given by:

$$y = \frac{f(x, quant\_step) + .75 \times mquant}{2 \times mquant} \qquad \text{……(6)}$$

$$y = \frac{16 \times f(x, quant\_step)}{mquant}$$

Here x is the DCT coefficient, y=f(x, quant_step) is determined from ISO/IEC 13818-2 tables [6]. As *mquant* increases, the effective quantization steps become larger, more information is lost, encoding requires lower bits, and also the quality of the picture degrades, and vice verse.

4.2 Quantization Factor Determination

To account for few of these factors, in the topmost level the value of mquant for each macroblock is calculated as a product of two primary factors (a) the buffer fullness and (b) the macroblock activity. The mquant for the jth frame is computed as a product of two parameters:

$$mquant_j = Q_j \times N\_act_j \qquad \text{….(7)}$$

The final value of *mquant$_j$* is coded either in the slice or in the macroblock. Qj is determined based on the frame and macroblock type and uses standard TM-5 model header [6]. The part that is relevant for this experiment is the N_act$_j$.[2] The motivation behind the original TM5 activity factor is that human visual perception is less sensitive to distortions in noisier textured areas and more sensitive to distortion in image areas with uniform texture. We used enhanced region based activity assignment algorithm for estimation of N_actj, based on the object tracking results. It allows spatial distribution of the bits to be controlled for a given allocation of frame bits.

4.3 Object based Activity Factor Determination

The relative bit allocation factor of the various objects and backgrounds detected is specified as a object resolution parameter α_i for each macroblock i. We maintain total per frame bit-allocation fixed. Thus:

$$\sum_i \log N\_act^{\text{mod}\,ified}{}_i \approx \sum_i \log N\_act_i \qquad \text{…(8)}$$

The macorblocks in the boundary set is assigned a macroblock resolution factor α_i (-8,0,+8). Based on the overall distribution of the α_i over a frame the N_activity(i) of a blocks is then calculated as:

[2] *Q$_{j.}$* It is a modulation parameter, that determines how the allocation of frame-bits itself is varied. The value *mquant* is calculated from the dynamic tracking of the bit-buffers. The detail involves prorated bit allocation based on frame and macroblock types.

$$N\_act^{\text{mod}\,ified}{}_i = N\_act_i \cdot \alpha_i \cdot 2^{\frac{\sum -\log \alpha_i}{n}} \qquad \ldots(9)$$

The log normalized value ensures that the bit distribution over the frame remains close to the original allocation of the TM-5 model.

5. EXPERIMENTS

5.1 Tracking Efficiency

We define two parameters for the evaluation of the motion-tracking algorithm.

The first one is the object *coverage*. This is the percentage of the actual visual object successfully covered by the active set.

The other one is the window *mis-coverage*. This is the percentage of the active set that do not cover the object (note these are not complement to each other).

Here we share some typical result from several video shots. The initial video (and the associated motion vectors) given to the transcoder was encoded as a standard MPEG-2 stream using an off the shelf commercial encoder (Ligos © MPEG-2 encoder) with GOP size 12 and distance between P frames 3 and frame rate 30 frames/second. Fig-4 shows the result of tracking two objects simultaneously for few frames in the *Two Tractors* sequence (the boundary of active set of each frame along with the original video picture is shown in black and white). To measure the tracking performance, for every 10th frame, we did a direct count of the macroblocks covered by the object and compared them with the corresponding active sets. Fig-5 demonstrates the object coverage and the window mis-coverage with the frame sequence (x-axis) for the two objects in this scene. It shows both germination and tracking. As can be seen the initial coverage is low near 40-60% because of the birth effect. However, soon after the algorithm determined the objects and the coverage reached 70-100%.

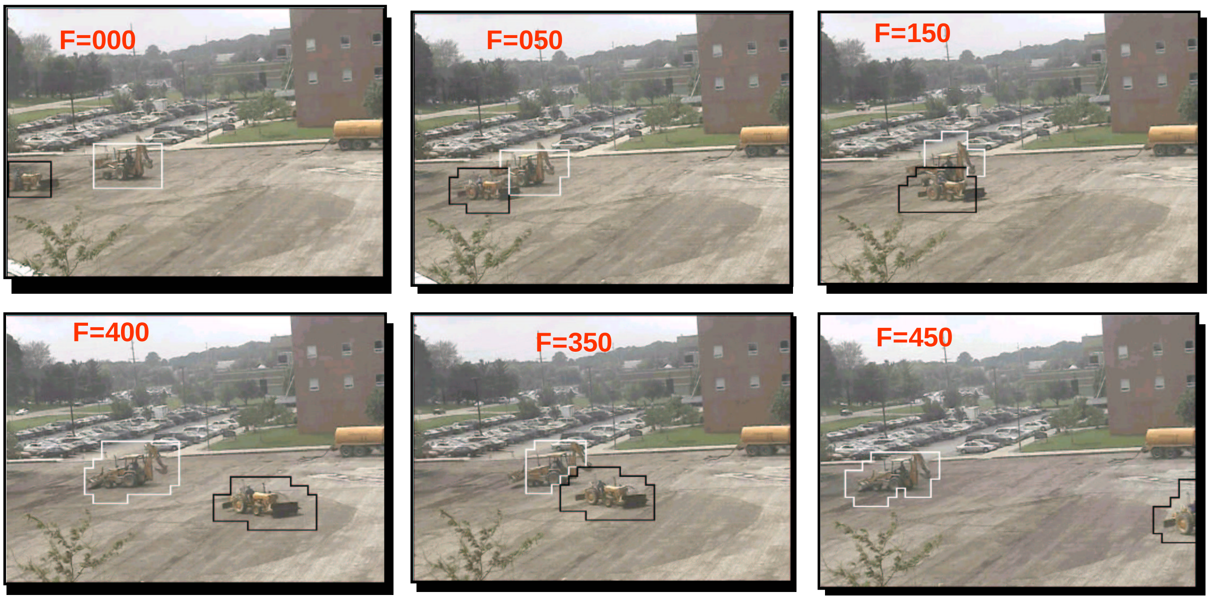

Fig-4.Results from tracking two objects simultaneously. The figures shows the boundary of the active block sets at 50 frame intervals. FOB algorithm tracks objects in P & B frames with reverse extrapolation on I frames.

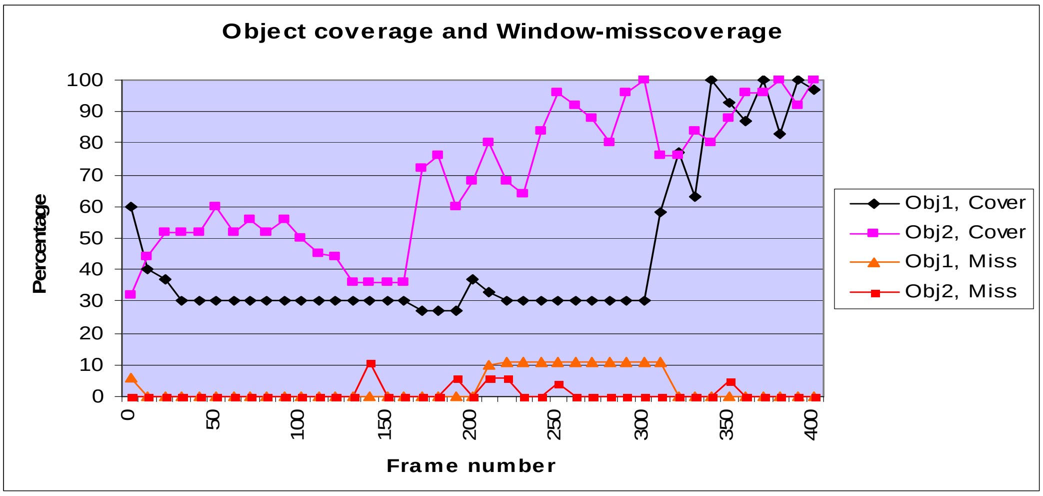

Fig-5 The evaluation of tracking efficiency. The object *coverage* and the window *miscoverage* evaluation of the "Two Tractors" sequence.

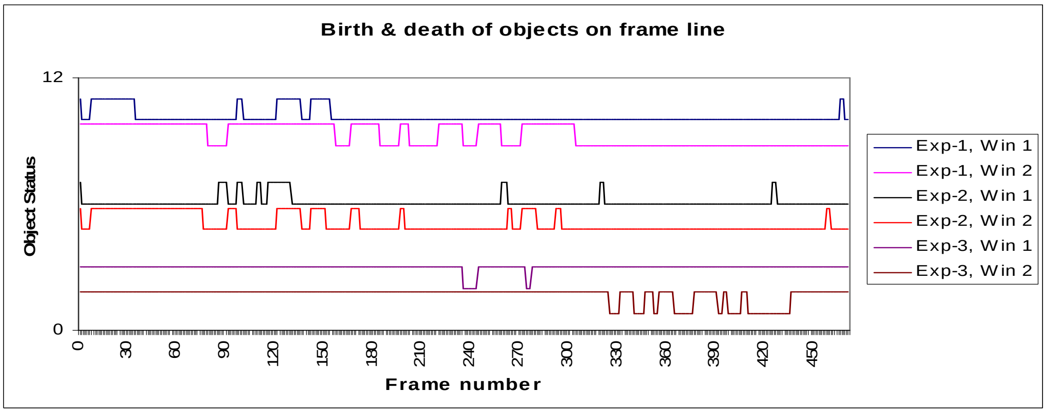

Fig-8 shows the window birth and death events under different germination and flocking parameters in Table-2.

5.2 Visual Performance

Fig-6 explains the visual effects of object-based and non-object-based re-encoding on the same sequence. It's bit rate was downsized from 5 mbps to 0.5mps. Fig-6(a) and 6(b) first shows a sample frame when encoded without perceptual encoding, and illustrates the advantage. With non-object-based method (top) the whole image region lost quality indiscriminately. As a result, the regions with more visual details create annoying discomfort. The result was significantly pleasant (for the same given target rate) when perceptual encoding was performed with automatic object tracking working at the background. As evident the two tractors, which are typically the objects of focus are much sharper, while the degeneration in the background is almost perceptually indistinguishable. (However, the still frame does not provide accurate perception of the video).

The corresponding bit allocation and SNR of different perceptual regions (objects vs. background) are also shown in Fig-7(a) and (b). As evident in Fig-7(a), the perceptual encoding kept all the bits (and corresponding high SNR) for the POP regions even with 10 times compression. Since, the overall object coverage was much smaller compared to the background;

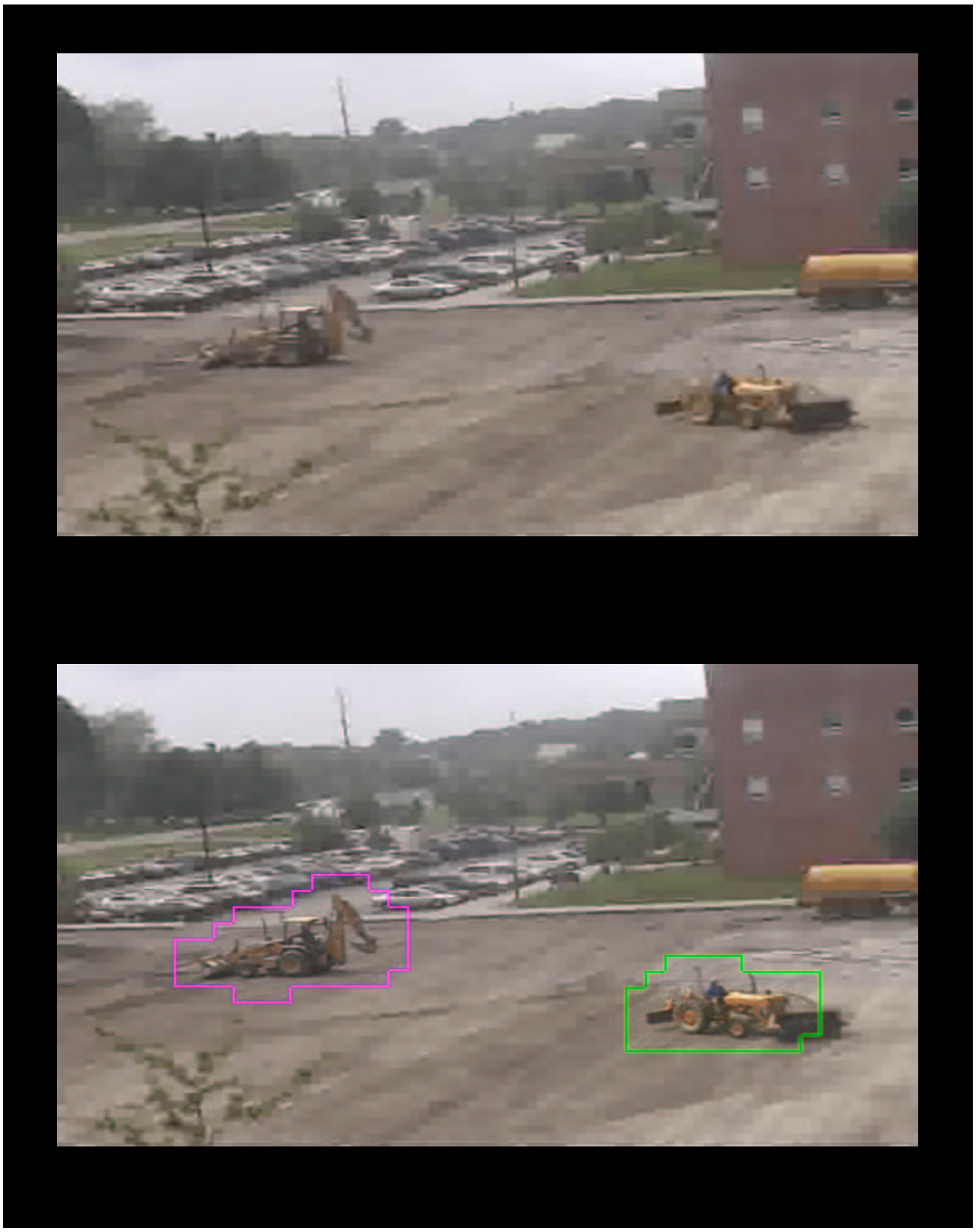

Fig-6 Stream (a) without and (b) with perceptual transcoding.

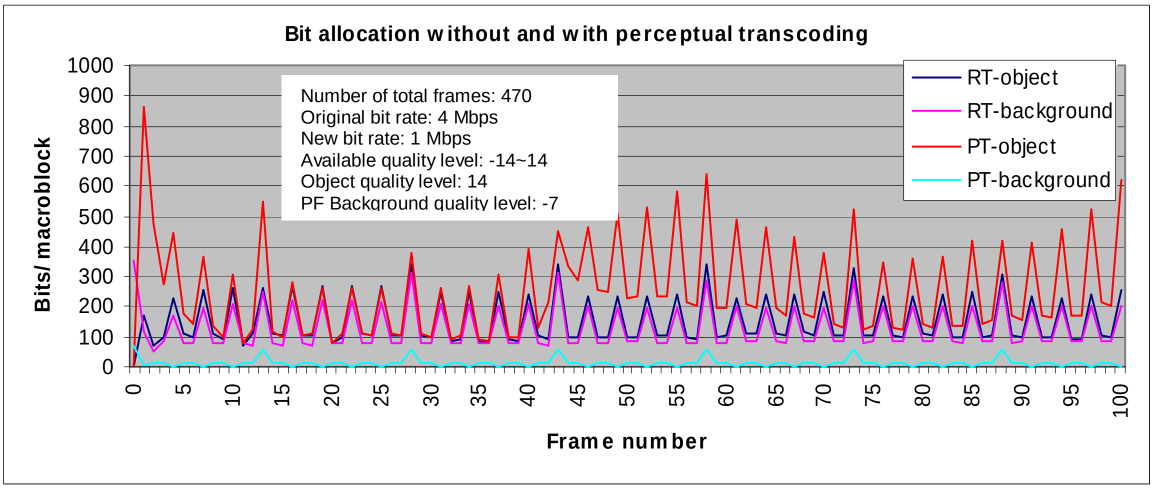

Fig-7 (a) In (a) The perceptual transcoder allocated more bits to the perceptually encoded object region (PT) in the frame and less bits to the background. The regular transcoding (RT) could not make the differentiation and makes the distribution uniform in both.

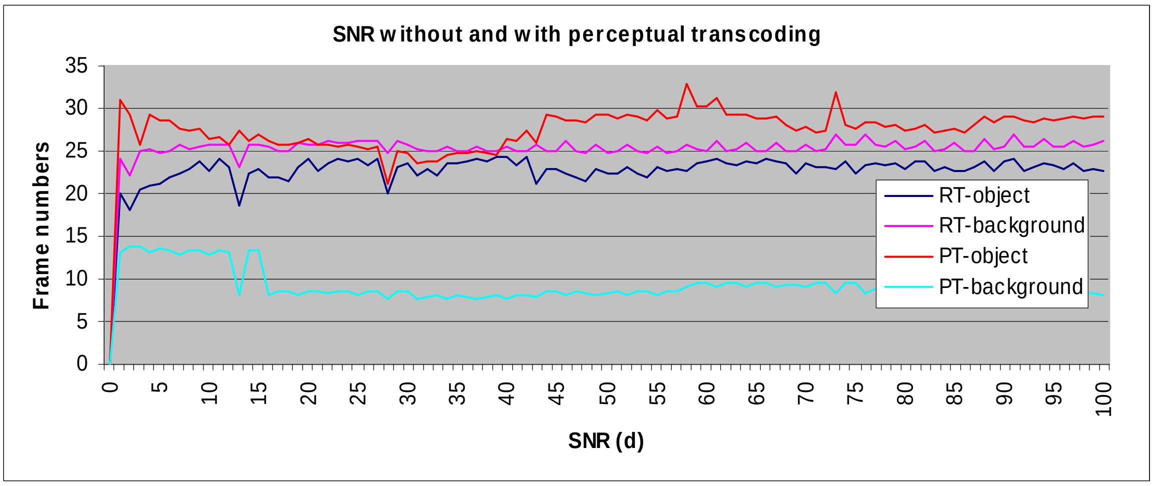

Fig-7 (b) Corresponding average SNR over the object and background regions in a frame without (RT) and with (PT) perceptual transcoding. .

the amount of bits (and loss of SNR) that had to be taken out from the background region was relatively small. As a result, background region is blurred almost to a similar degree as that in the non-object-based method, but object region, which will be the focus most of the time, still maintains its high quality.

The overall perceptual quality of the video sequence turn out to be significantly higher than the uniform blur effect of non-object-based re-encoding.

5.3 Descriptor and Content Sensitivity

We were also curious to see the sensitivity of the parameters. Like any other scene analysis algorithm the content and activity of the scene also affect the performance. We conducted large number of sensitivity tests for various descriptor variances using large number of other video samples. The system was able to lock in reasonably on objects and track them. In Table-2, we show the detail performance of the algorithm on ten clips. Object coverage and window miscoverage are calculated as the average of 10 sample frames each from every 30 frames for the first 300 frames after tracking starts. In general, the video sets we tested demonstrated (one continuous shot) stable tracking all the way for about 200-300 frames— or over more than 10 typical GOPs. We also varied the germination and flocking parameters. For this experiment, we selected three sets of v_F, v_D and u_D. Then for each set we tested two values of m_F and u_F. The germination and flocking parameters for these six sets are shown in Table-1. Fig-8 demonstrates the birth and death events as the result of detection and tracking when varying the germination and flocking parameters as per the six sets of Table-1. The transitions between these two states indicate birth or death events.

	Experiment 1		Experiment 2		Experiment 3	
	Win 1	Win 2	Win 1	Win 2	Win 1	Win 2
m_F	15	20	5	5	10	15
u_F	1	1	1	1	5	5
m_D	10	5	5	5	20	20
u_D	0	0	0	0	2	2
v_F	50	50	80	80	30	30
v_D	50	50	80	80	30	30

Table-1. The germination and flocking parameters for the experiments.

5.4 Speed

We performed detail speed analysis of the transcoder. The speed performance of the tracking system was less than 2% of the encoder part of the transcoding operation, and this was within the margin of system variance/error in our practical implementation based experiment. We achieved about 16-22 frame/sec by running the combined system (recoding+detection and tracking) on 1GHz P4 for QCIF video. Curious reader may consult [7,21] for stage-by-stage detail time performance for the entire transcoding system.

6. CONCLUSIONS

In this paper we have described a content aware video transcoding technique that can accept high-level description of video objects and use it for perceptual encoding based extreme video downscaling of a coded video stream in at frame rate. We have implemented an MPEG-2/MPEG-2 transcoder to demonstrate a fully transparent transcoding system for a streaming video, which will not require the player or server to be aware of the rate adaptation. However, the algorithm is also applicable for MPEG-2/MPEG-4 transcoding.

The advantage of the presented FOB algorithm is that its tracking complexity is much below frame rate. It also demonstrated reasonable tracking performance. Our steady-state system is able to operate with 80-100% coverage and less than 5% mis-coverage. The results on the tracking stability tend to indicate that about every few (~5-10) seconds one embedded video object description frame might be enough to support such object-based transcoding. It adds about 0.5-1.0 bits/object/frame overheads to the incoming high-speed video stream, which also should have negligible impact. Unfortunately, we are unable to provide comparison with other methods. Since, any such tracking efficiency has rarely been qualitatively measured. It is not known actually how much detection and tracking accuracy has been gained-- if any, even by incurring pixel level computational complexities.

The specification of expected object is a major challenge in almost any detection algorithm. The FOB algorithm allows user to fine tune the expectation about the target by the germination and flocking parameters. Scenarios where such specification is not available, the BOF algorithm too will face limitation like other autonomous detection systems.

Video sequence	(%) Object Coverage	(%) Object Mis-coverage
Car video 1	69	11
Car video 2	71	22
Car video 3	89	33
Parking Lot 1	85	0
Parking Lot 2	76	1
Toy video 1	53	5
Toy video 2	65	2
Toy video 3	74	22
Walk video 1	60	3
Walk video 2	74	19

Table-2 Tracking performance on ten video sequences. Object coverage and mis-coverage are calculated as the average of 10 sample frames each from every 30 frames for the first 300 frames after tracking starts.

The proposed logical POP descriptors in the form of germination and flocking parameters are intuitive. However, in future they can be expanded as a basis set for a high-level POP language. In our current implementation, we have developed a Graphical Monitor Interface (GMI). This allows a human observer to dynamically tune them. Descriptors can also be added into standards' private stream (MPEG-2 and MPEG-4 already has the mechanism). Alternatively, adhoc mechanisms are possible where XML type object schema's can be downloaded transparently.

It is also interesting to note that in the context the perceptual coding, the use of detail object boundary may or may not be as useful. This may appear counter intuitive. However, there is direct evidence that human eye samples both inside and outside of the boundary areas in its vicinity. Thus, there is a strong possibility that pure object boundary based multi-resolution coding may actually reduce the perceptual quality.

As a real implementation, the system has faced a number of other related open issues besides fast object detection. For example, bit-allocation models in objects and its impact on perceived video quality, fast re-encoding, smooth rate control beyond TM-5, coding scheme suitable for spatially varying images and video, etc. Each of these is active research area. Interesting solutions are still emerging. The principal emphasis of this paper is only the object detection and tracking model suitable for live transcoding. Further details about few others of these aspects can be found in [7, 12, 21, 22]. However, the proposed detection algorithm can be used with most other solutions to those associated problems.

This research has been supported by the DARPA Research Grant F30602-99-1-0515.

7. REFERENCES:

[1] Aizawa, K., H. Harashima, & T. Saito, Model-based Image Coding for a person's Face, Image Commun, v.1, no.2, 1989, pp 139-152.

[2] Casas, J. R., & Torres, L, Coding of details in very Low Bit-rate Video Systems, IEEE Transactions CSVT, vol. 4, June, 1994, pp. 317-327.

[3] R. Wang, H. -J. Zhang and Y. -Q. Zhang, ¡§A Confidence Measure Based Moving Object Extraction System Built for Compressed Domain,¡¨ in Proc. IEEE Symp. Circuits and Systems, vol. 5, pp.21 -24, May 2000, Geneva, Switzerland.

[4] H.-L. Eng and K. ¡VK. Ma, ¡§Spatiotemporal segmentation of moving video objects over MPEG compressed domain,¡¨ in Proc. IEEE International Conference on Multimedia and Expo, vol.3, pp.1531 ¡V1534,2000, New York

[5] Hotter, M., & R. Thoma, Image Segmentation based on Object-Oriented Mapping Parameter Estimation, Sinal Process., v. 15, 1998, pp.315-334.

[6] Information Technology- Generic Coding of Moving Pictures and Associated Audio Information: Video, ISO/IEC International Standard 13818-2, June 1996.

[7] Khan, Javed I., Darsan Patel, Wansik Oh, Seung-su Yang, Oleg Komogortsev, and Qiong Gu, Architectural Overview of Motion Vector Reuse Mechanism in MPEG-2 Transcoding, Technical Report TR2001-01-01, Kent State University, January, 2001, [available at URL http://medianet.kent. edu/ technicalreports.html, also mirrored at http:// bristi.facnet.mcs.kent.edu/medianet]

[8] Keesman, Gertjan; Hellinghuizen, Robert; Hoeksema, Fokke; Heideman, Geert, Transcoding of MPEG bitstreams Signal Processing: Image Communication, Volume: 8, Issue: 6, pp. 481-500, September 1996.

[9] Khan, Javed I. & D. Yun, Multi-resolution Perceptual Encoding for Interactive Image Sharing in Remote Tele-Diagnostics, Proc. of the Int. Conference on Human Aspects of Advanced Manufacturing: Agility & Hybrid Automation, HAAMAHA'96, Maui, Aug. 1996, pp183-187.

[10] Khan, Javed I. & S. S. Yang, Resource Adaptive Nomadic MPEG-2 Transcoding on Active Network, International Conference of Applied Informatics, AI 2001, February 19-22, 2001, Insbruck, Austria.

[11] Koenen, Rob, MPEG-4 Overview, Coding of Moving Pictures and Audio, V.16, La BauleVersion, ISO/IECJTC1/SC29/WG11, October 2000, [URL: http://www.cselt.it/mpeg/standards/ mpeg-4/mpeg-4.htm, last retrieved January, 2001]

[12] Javed I. Khan, Qiong Gu and Raid Zaghal, Symbiotic Video Streaming by Transport Feedback based Quality rate Selection, Proceedings of the 12th IEEE International Packet Video Workshop 2002, Pittsburg, PA, April 2002. http://www.pv2002.org (electronic proceeding)

[13] Youn, J, M.T. Sun, and J. Xin, "Video Transcoder Architectures for Bit Rate Scaling of H.263 Bit Streams," ACM Multimedia 1999', Nov., 1999. pp243-250.

[14] Ngo, Chong-Wah, Ting-Chuen Pong and Hong-Jiang Zhang, "On clustering and retrieval of video shots", ACM Multimedia 2001, Oct., 2001. pp51-60.

[15] Kuehne, Gerald, Stephan Richter and Mark Beier, "Motion-based segmentation and contour-based classification of veideo objects", ACM Multimedia

2001, Oct., 2001. pp41-50.

[16] U. Chong and S. P. Kim, Wavelet Trancoding of block DCT-based images through block transform domain processing, SPIE Vol. 2825, 1996, pp901-908.

[17] Niklas Björk and Charilaos Christopoulos, Video transcoding for universal multimedia access; Proceedings on ACM multimedia 2000 workshops, 2000, Pages 75 – 79

[18] J. Youn, M.T. Sun, and C.W. Lin, "Motion Vector Refinement for High Performance Transcoding," IEEE, Transactions on Multimedia, Vol. 1, No. 1, pp.30-40, March 1999.

[19] P. Assuncao and M. Ghanbari, "A frequency-domain video transcoder for dynamic bit rate reduction of MPEG-2 bit streams," Trans. On Circuits Syst. Video Technol., vol. 8, no. 8, pp. 953-967, 1998.

[20] Seo, Kwang-Deok; Lee, Sang-Hee; Kim, Jae-Kyoon; Koh, Jong S., Efficient rate-control algorithm for fast transcoding, ,SPIE Vol. 3528, 1998.

[21] Javed I. Khan and Seung Su Yang, Darsan Patel, Oleg Komogortsev, Wansik Oh, and Zhong Guo, Q. Gu, 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, Proceedings of the Active Networks Conference and Exposition, Active Network Conference and Exposition DANCE ’2002, San Jose, CA May 21-24, 2002, IEEE Computer Society Press, pp.409-426.

[22] Khan Javed I. Zhong Guo, & W. Oh, Motion based Object Tracking in MPEG-2 Stream for Perceptual Region Discriminating Rate Transcoding, Proceedings of the ACM Multimedia, 2001, October 2001, Ottawa, Canada, pp572-576.