Proceedings of the ACM International Conference on Multimedia,
ACM MM'2001, Ottawa, Canada, October 2001

Motion based Object Tracking in MPEG-2 Stream for Perceptual Region Discriminating Rate Transcoding

Javed I. Khan, Zhong Guo and Wansik Oh

Media Communications and Networking Research Laboratory
Department of Computer Science
Kent State University, Kent, OH 44242
Phone: 330-672-9038
javed|zguo@kent.edu

ABSTRACT

Object based bit allocation can result in significant improvement in the perceptual quality of relatively low bit-rate video. In this paper we describe a novel content aware video transcoding technique that can accept high-level description of video objects and extract them from incoming video stream and use it for perceptual encoding based extreme video downscaling.

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

1. INTRODUCTION

Current video transcoding techniques are based on requantization [8,13]. However, requantization do not seem to be capable enough to down scale a video to the very low bit-rate required by current the Internet scenario. In recent past, it has been shown that object based encoding plays an increasingly important role for carrying perceptually pleasant video at lower rates [2,4,9,12]. In this research we are studying how perhaps perceptual object based coding concept can be incorporated into transcoding for extreme rate transformation required in splicing asymmetric segments of the Internet.

Object based video encoding is an open problem. Recent MPEG-4 standard provides syntax to specify objects into video stream [11]. Despite the standardization of the syntax, object detection remains a serious open challenge [1,3,5]. It is much more difficult to detect an object than to compute a wavelet or DCT co-efficient. Indeed MPEG-4 objects are currently thought to have best chance for computer model generated synthetic video where objects do not need to be detected or in limited domain (such as head and shoulder) small format video [1,2]. Even in these special situations object detection algorithms are generally quite computation intensive [5]. Consequently, these techniques have been pursued primarily for first stage video encoding. Because most first stage encoding scenarios (except for live video) allow for off-line processing.

In this context, the transcoding scenario has several significant differences from conventional first stage encoding. First of all the original frame images are no longer explicitly available. The transcoder receives an encoded video stream. Secondly, the object detection has to be performed extremely fast at the rate of the stream. Thirdly, it already contains some refined information (such as motion vectors). Consequently, techniques for transcoding can generally be used in first stage encoding but the reverse in not always possible.

In this research we particularly describe a low computation based object tracking algorithm suitable for full motion focus region based perceptual video recompression. This novel technique accepts some logical and high-level initial description of the video objects in terms of initial position, and shape. It then automatically tracks the region covered by this object for subsequent perceptual encoding based on real-time motion analysis. We have restricted the problem that no pixel level decoding of DCT or image components is allowed to conform to the constraints of the transcoding scenario. In this paper we present the stream object tracking method and share some of the results from a novel perceptual transcoder, that we have recently implemented using this tracking algorithm.

2. SYSTEM MODEL

2.1 Transcoder Architecture

The perceptual transcoder system accepts and produces MPEG-2 ISO-13818 [6] video stream, and is capable of dynamically adjusting the incoming bit-rate to an outgoing *piece-wise constant bit-rate* (pCBR) [7,10]. The control is similar to the MPEG-2 test model TM-5 algorithm. Besides the pCBR operation the system can modulate the sample density both in temporal as well as spatial dimension based on region description. The detail of the rate control algorithm however is not within the scope of this paper but can be found in [7]. In this paper we particularly focus how the region of interest window required by the perceptual rate adaptation/control algorithm is tracked from the incoming stream for extreme video downscaling.

2.2 Tracking System Model

We view a frame F_t as a matrix of macroblocks $m_t(i,j)$, i and j being the column and row indices and subscript t the frames presentation sequence. The approach classifies frame macroblocks

into *active* (A_t), *monitored* (M_t) and *inactive* (I_t) sets, based on motion vector analysis. Macroblocks representing the same video object are grouped as A_t. Macroblocks surrounding the A_t belong to M_t, and those beyond are in I_t. The membership of the macroblocks can change from frame to frame. Fig-1(a) shows the typical active and monitored set and Fig-1(b) shows the transition model. The union of these sets for all objects is the frame set (F). Here superscript r is the index of the focal region.

$$F_t = \{\bigcup_r [A_t^r \cup M_t^r]\} \cup I_t$$

Each focal region is defined by a set of macroblock properties called *macroblock property set* (MPS or p_i^r). A focal region also has an *aggregate property set* (APS or $\overline{p}^r$), which is derived from the properties of the member macroblocks in its active set. The idea is that the active set defines the core focal object and is quantified by its aggregate property set. Both the region memberships and the aggregate properties are dynamic. The MPS properties of the active and the monitored sets both are continuously monitored. However, for the later is not used to update the APS. A macroblock, which is currently in M can be indoctrinated in A. Similarly, a macroblock, which is now in A can loose its membership and relegated to M or I in next frame. The transitions are determined by *set transition rules* defined based on distance measure between the APS and the MPS of individual macroblocks explained next.

2.3 Moving Object Model

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. Let $p^r = [v_t(x), v_t(y)]$ where $v_t(x)$ and $v_t(y)$ be the horizontal and vertical motion vectors associated with $m_t(i,j)$. We also denote $\overline{p}^r = [\overline{v}_t(x), \overline{v}_t(y)]$. An object is then defined by the following seven parameters:

Initial Shape (S): a region in a frame denoting the A_o.

Monitor Span (D): the width of M spans in number of macroblocks

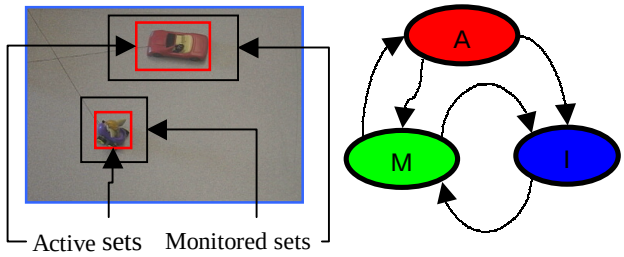


Fig-1(a) (left) A, M sets of macroblocks with a frame with two objects. I is the background. Fig-1(b) (right) Transition is bi-directional between A-M, and M-I, but unidirectional from A to I, A macroblock can not can not transition in A directly from I.

Deviator Thresholds ($P_1(x)$ and $P_1(y)$): if the difference magnitude between $V_t(x)$ of $m_t(i,j) \in A_t$ and $\overline{V}_t(x)$ is greater than $P_1(x)$ percent of $\overline{V}_t(x)$, or the difference magnitude between $V_t(y)$ of $m_t(i,j) \in A_t$ and $\overline{V}_t(y)$ is greater than $P_1(y)$ percent of $\overline{V}_t(y)$, $m_t(i,j)$ is not following the movement of observed object.

Follower Thresholds ($P_2(x)$ and $P_2(y)$): if the difference magnitude between $V_t(x)$ of $m_t(i,j) \in M_t$ and $\overline{V}_t(x)$ is less than $P_2(x)$ percent of $\overline{V}_t(x)$, or the difference magnitude between $V_t(y)$ of $m_t(i,j) \in M_t$ and $\overline{V}_t(y)$ is less than $P_2(y)$ percent of $\overline{V}_t(y)$, $m_t(i,j)$ is following the movement of observed object.

Deviator Persistence (N_1): if $m_t(i,j) \in A_t$ has not been following the movement of observed object for consecutively N_1 times, remove $m_t(i,j)$ from A_t.

Follower Persistence (N_2): if $m_t(i,j) \in M_t$ has been following the movement of observed object for consecutively N_2 times, add $m_t(i,j)$ to A_t.

Group Volatility (P_d): a maximum of P_d percent of macroblocks from A can be removed in one frame.

For each video object one set of the above seven parameters is given. The last six of the above have been defined as thresholds and they also determine the set transition rules. Below we now describe the algorithm.

3. INVERSE PROJECTION ALGORITHM

The tracking starts with the initial positions of the perceptual objects. Initial A^r_0, M^r_0 and I_0 sets are defined for the first frame corresponding to the objects. For each subsequent P frame in the presentation (as well as coding) sequence, A^r_t and M^r_t sets are predicted by shifting the A and M 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 are handled:

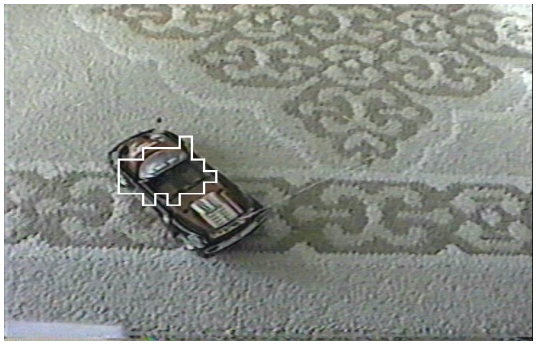
1. Initialize A and M sets

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 $m(x_1,y_1)$ and $m(x_2, y_2)$ are identified. The initial A will be composed of all m(i,j) where $x_1 \le i \le y_2$ and $x_1 \le j \le y_2$. If m(a,b) $\notin$ A, but within D macroblocks away from m(i,j) $\in$ A, let m(a,b) $\in$ M.

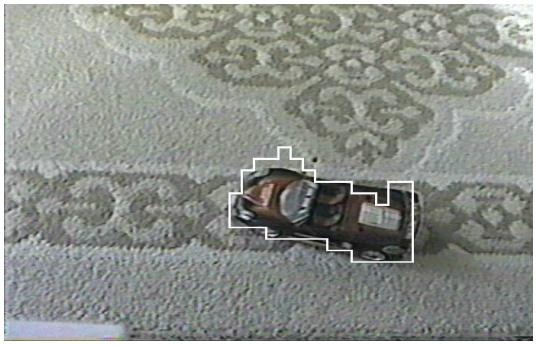
2. Predict A and M for frame t from previous frame t′

We obtain the A and M sets for frame t by shifting the A and M sets of a previous P frame t′ towards the object's movement direction.

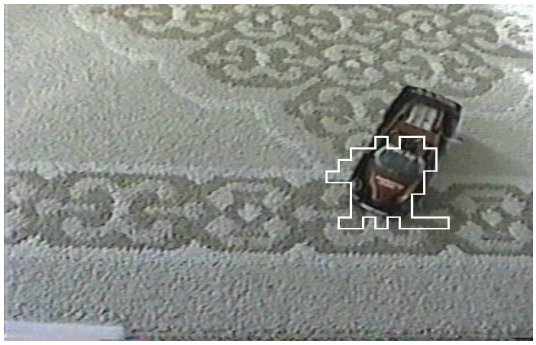
(a) Frame 0



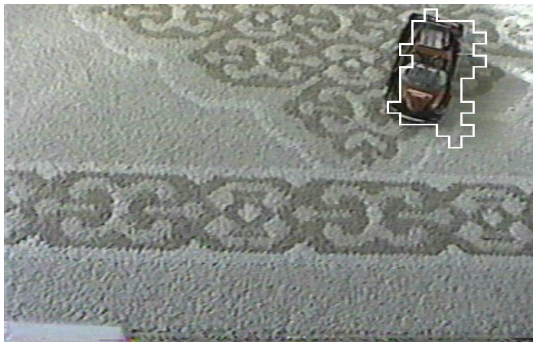
(b) Frame 40



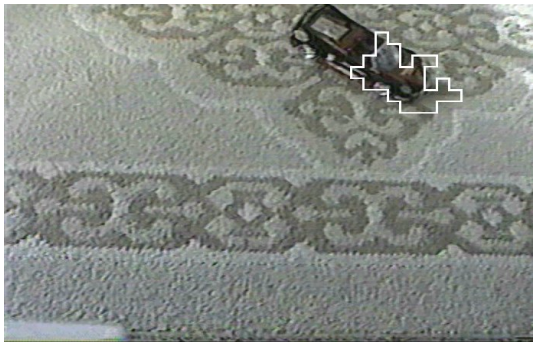
(c) Frame 80



(d) Frame 120

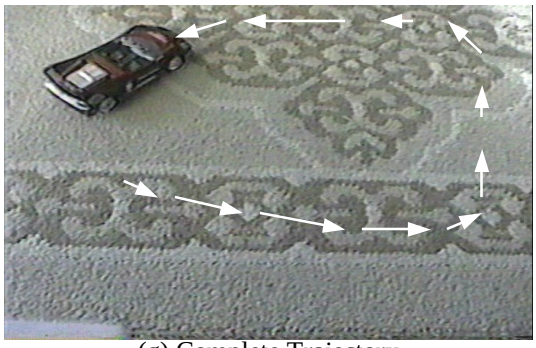


(e) Frame 160



(f) Frame 200

Fig-2 (a)-(f) Results of the motion tracking. The active window (shown with white boundary) detected by the algorithm mostly kept up with the objects. For this detection we used the parameters D=2, P_1=50, P_2=80, N_1=0, N_2=0, D_d=20. The algorithm occasionally (f) lost partial track when the vehicle was turning. However, it regained the track after a while. Fig-5(g) shows the trajectory of the car in our sample video sequence.



(g) Complete Trajectory

Given $\overline{V}_{t'}(x)$ and $\overline{V}_{t'}(y)$ of frame t′, since they were forward predicted, the shift direction should be opposite.

I. If frame t is I or P frame, A_t and M_t are generated by shifting $A_{t'}$ and $M_{t'}$ horizontally and vertically by the following number of macroblocks:

· $-\frac{\overline{V}_{t'}(x)}{16}$ and $-\frac{\overline{V}_{t'}(y)}{16}$

II. If frame t is B frame, assuming there are n B frames between two adjacent P frames and t is the ith one, We predict A_t and Mt by shifting $A_{t'}$ and $M_{t'}$ horizontally and vertically by the following number of macroblocks:

· $-\frac{\overline{V}_{t'}(x)\times i}{16\times(n+1)}$ and $-\frac{\overline{V}_{t'}(y)\times i}{16\times(n+1)}$

Step 3 to 5 are only performed on P frames.

3. Reshape the current frame A_t through motion analysis

I. Sort $V_t(x)$ of all $m_t(i,j) \in A_t$, choose the median value as $\overline{V}_t(x)$. Choose $\overline{V}_t(y)$ the same way.

II. For each $m_t(i,j) \in A_t$, if

· $\frac{|V_t(x)-\overline{V}_t(x)|}{|\overline{V}_t(x)|} > P_1(x)$ or $\frac{|V_t(y)-\overline{V}_t(y)|}{|\overline{V}_t(y)|} > P_1(y)$

We consider $m_t(i,j)$ is not following the object movement in current frame, if it has been not following in consecutively N_1 P frames. We then remove it from A_t. But we can only remove at most N_d percent of macroblocks from A_t. If more than N_d failed, the N_d percent macroblocks with largest values in following equation will be removed from A_t:

· $|V_t(x)-\overline{V}_t(x)|+|V_t(y)-\overline{V}_t(y)|$

III. for each $m_t(i,j) \in M_t$, if:

· $\frac{|V_t(x)-\overline{V}_t(x)|}{|\overline{V}_t(x)|} < P_2(x)$ and, $\frac{|V_t(y)-\overline{V}_t(y)|}{|\overline{V}_t(y)|} < P_2(y)$

we consider $m_t(i,j)$ follows object movement in current frame. If it has been following in consecutively N_2 P frames, let $m_t(i, j) \in A_t$.

4. Spatial Locality based adjustment to A_t

I. for each $m_t(i,j) \notin A_t$, if $m_t(i-1,j)$, $m_t(i+1,j)$, $m_t(i,j-1)$ and $m_t(i,j+1)$ all $\in A_t$, let $m_t(i,j) \in A_t$. Intra coded macroblocks that were removed form A_t because of their zero motion vectors can be recovered through this operation.

II. for each $m_t(i,j) \in A_t$, if $m_t(i-1,j)$, $m_t(i+1,j)$, $m_t(i,j-1)$ and $m_t(i,j+1)$ all $\notin A_t$, let $m_t(i,j) \notin A_t$.

5. Reset M_t

For all $m_t(a,b) \notin A_t$, if it is within D macroblocks away from $m_t(i,j) \in A_t$, let $m_t(a,b) \in M_t$.

The above procedure is now repeated for each object. The output of the system is the sequence of active sets $[A_0, A_1, A_2.....]$, which is fed to the transcoder rate controller as video object region. The transcoder then correspondingly generates the pCBR video stream with appropriate spatial distribution of bits for the specified outgoing bit-rate.

4. EXPERIMENTS

We use two parameters for the evaluation of the motion-tracking algorithm. The first one is the *object-coverage*, which is the percentage of the actual visual object successfully covered by the active set. The other is the *mis- coverage*, which is the percentage of the active set that did not cover the object (note these are not complement to each other). Here we share a typical result from a video shot with a toy car moving on carpet. 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-2(a)-(f) shows the tracking result for few frames (the boundary of A of each frame along with the original video picture is shown in white). Fig-2(g) shows the actual trajectory of the object on the final frame. To measure the tracking performance, we paused the video for every 10 frames, and did a direct count of the macroblocks covered by the object and compared them with the corresponding active sets. Fig-3 demonstrates the *object-coverage* and the *mis-coverage* with the frame sequence (x-axis). As visible the *object-coverage rate* is typically higher than 90%, particularly when the perceptual object has translation movement. During each turn it somewhat lost the tracking but after a while it recovered. The precision of the tracking is given by near zero *mis-coverage*. We let the algorithm run till it completely looses the track to approximate the maximum stability. The video set we tested shows (one continuous shot) stable tracking all the way for about 200-300 frames— over more than 10 typical GOPs. Finally Fig-4 shows the perceptual encoding (with temporal sample fusion [7]) results for a commercial movie clip sequence with high motion. Fig-4(a) and (b) shows the macroblock-wise bit distribution on the frame-plane without and with object detection activated. For both cases the total outgoing bit rate was same, but as can be seen in the right, we were able to allocated more bits in the regions of object.

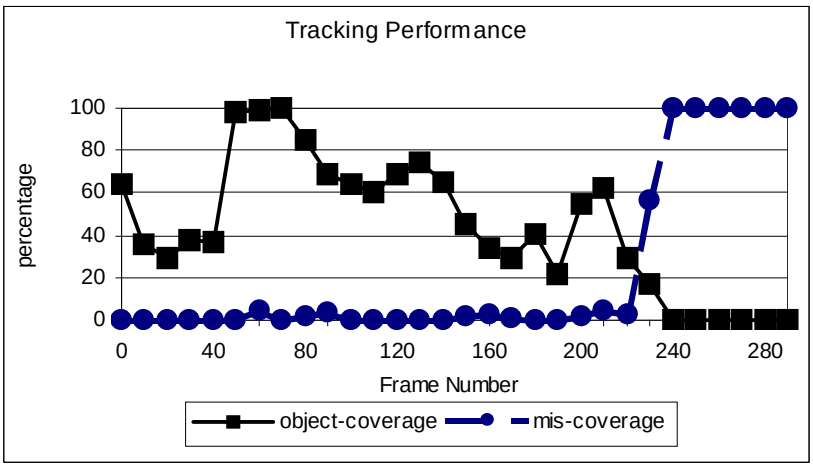


Fig-3 Tracking performance in a video sequence showing the *object-coverage* and *mis-coverage*.

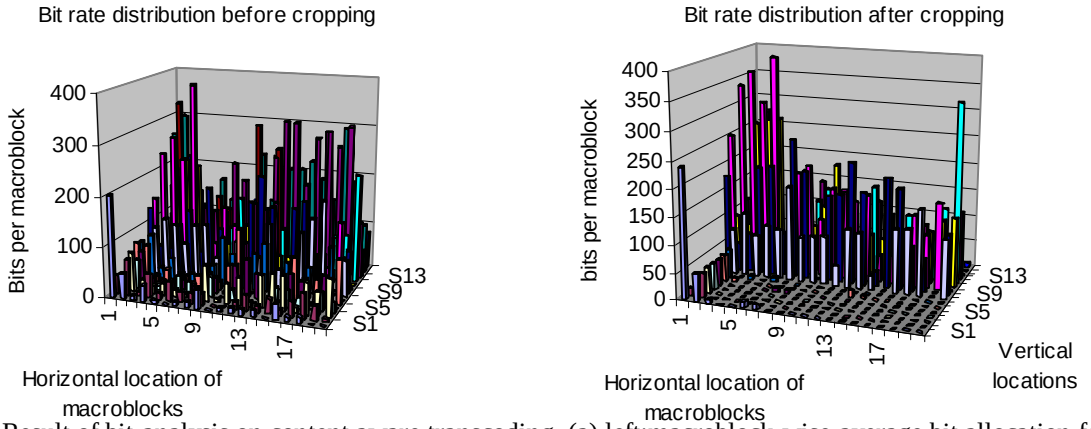


Fig-4 Result of bit-analysis on content aware transcoding. (a) left:macroblock-wise-average bit allocation for the entire video sequence for rate transcoding without object analysis. (b) right: the same with object analysis. The boundary area was encoded with much less bits in object analysis based transcoding.

5. CONCLUSIONS

It seems some form of object centered perceptual encoding will be inevitable in extreme rate (down) scalability. In this paper we have described a novel content aware video transcoding technique that can accept high-level description of video objects and use it for perceptual encoding based extreme video downscaling. Though we have implemented the system for MPEG-2/MPEG-2 transcoding but techniques such as this can play important role in the emerging MPEG-4/MPEG-2 splicing. Currently, the tracking complexity comprises only a negligible part of the overall transcoding. Computations are confined only within the pre-extracted motion vectors of the active and monitored sets. 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. We are currently performing additional investigation into tracking stability and spontaneous birth-and-death of video objects. This research has been supported by the DARPA Research Grant F30602-99-1-0515 under its Active Network initiative.

6. 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] De Silva, L.C., K. Aizawa, M. Hatori, "Use of Steer-able Viewing Window (SVW) to improve the visual sensation in face to face teleconferencing", ICA SSP Proceedings, v.5, 1994, pp421-424.

[4] Haskell B. G., Atul Puri and Arun Netravali, Digital Video: An Introduction to MPEG-2, Chapman and Hall, NY, 1997.

[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, [available at URL http://medianet.kent.edu/ technicalreports.html, also mirrored at http:// bristi.facnet.mcs.kent.edu/medianet] January, 2001]

[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, Hawaii, 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, (accepted as full paper, available from http://www.mcs.kent.edu/~javed.)

[11] Koenen, Rob (Editor), 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] Minami. T. et. al, Knowledge-based Coding of facial Images, Picture Coding Symposium, Cambridge, MA, pp. 202-209.

[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.