Human-Content and Gesture-Event Video Coding*

Robin and J. W. Davis
Dept. of Computer and Information Science
Ohio State University
Columbus, OH 43210
{robin,jwdavis}@cis.ohio-state.edu

Abstract

Currently, bandwidth limitations pose a major challenge for delivering high-quality multimedia information to users. In this research, we aim to provide a better compression of human-centered video sequences such as lectures, monologues, and presentations. Based on the idea that people pay more attention to face and hand regions in videos containing people speaking, our approach encodes those regions with higher resolution than the remaining image. Using computer vision techniques, we segment and track the subject's face and hands. The face region is assigned the highest salience value. Gesture analysis of the hands is then used to encode important gesture events at high salience and non-gestures at a lower value. We demonstrate the differential video coder with the production of three highly-salient, low-bandwidth video monologue sequences.

1. Introduction

Initially, the Internet was mostly used to communicate and share textual, non-multimedia types of data. Today, the Internet is a rich medley of text and multimedia data (such as audio and video), and has become centers of information and entertainment. For communication, the Internet has progressed from E-mail to video conferencing and Voice over IP technologies.

Unlike textual data, the transmission of multimedia data (e.g., streaming video) presents special challenges in its real-time delivery over the Internet. For video sequences, the frame rate of the video needs to be reasonably high (many agree that frame rate should be at least 16 frames per second [19]), and reasonably jitter-free with high picture quality. These problems have been addressed by employing client-side buffering, forward error correction, and piggy-backing of data. However, these approaches are still not enough to provide high user satisfaction. Currently, bandwidth limitations largely continue to force video sequences to have unacceptably low frame rates and/or picture quality.

In the domain of monologue presentation videos, comprised of a single speaker or lecturer (e.g., for distance learning), we present a gesture-based computer vision and coding method to enhance the communicative content of this type of multimedia video over the Internet. As viewers are typically most interested in watching the presenter in the video, we use computer vision algorithms to track the subject's face and hands and encode them at higher resolution than the background.

Additionally, viewers tend to pay more attention to gesturing, rather than to non-gesturing, hands (as demonstrated in [23]). We encode the hands at different qualities: gesturing hand events are encoded at high resolution and non-gesturing hands at lower resolution (though at higher quality than the background). This

*OSU Dept. of Computer and Information Science Technical Report OSU-CISRC-3/02-TR03.

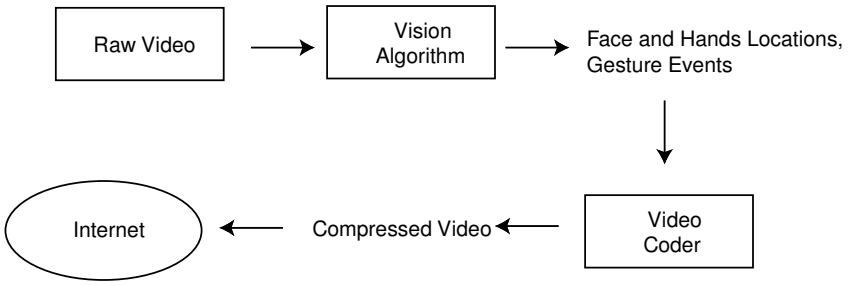


Figure 1: Differential video coder system diagram.

differential encoding of face and hands conserves the bandwidth without sacrificing much of the communicative quality of the video. To compensate for the increased resolution at these regions, we slightly reduce the remaining background quality.

The main contribution of this research is an innovative technique for integrating computer vision, gesture analysis, and multimedia networking research to provide users with a better multimedia experience over the Internet. In our approach, the raw video is first analyzed by vision algorithms to segment and track the presenter in the video. The system then detects the presenter's hand gesture events. Based on the results of the vision and gesture algorithms, the video coder performs a differential encoding on different parts of images with the face and hands regions at higher resolution than the background. The system diagram is presented in Fig. 1.

In Section 2, we discuss related work in computer vision, gesture analysis, and multimedia networking. Section 3 details the algorithms used for segmentation, tracking, and gesture detection. Section 4 presents the differential video coder. Section 5 presents the experimental results. In Section 6, we conclude this work with a summary of the research and discuss future work.

2. Related Work

As this research is multidisciplinary, encompassing computer vision, gesture analysis, and multimedia networking, we divide the related work into three sections. First, we present an overview of computer vision research related to multimedia event processing. Second, we present related gesture analysis work, and lastly describe current multimedia networking research.

2.1 Computer Vision

There has been much recent work in the detection of events and actions in video sequences [5]. The need to classify and data-mine huge multimedia databases has driven much of the computer vision work. Many of the applications have been in the areas of content-based retrieval, surveillance, and human-computer interaction.

IBM's CueVideo system [22, 28, 6, 29] combines video/audio analysis and speech recognition to provide automated indexing and hyperlinking of videos. The system also uses MSB (moving storyboard) and TSM (Time Scale Modulation) techniques to produce compact content-rich video summaries suitable for downloads and quick browsing. The system allows users to search for videos that have certain content from a large video database.

In [21], a method is proposed for automatic goal segmentation in basketball video sequences. This was accomplished mainly by recognizing certain key repetitive events, like crowd cheer, scoreboard display, and change in direction of the players. Since the text of scoreboard display in the scene is artificially embedded in the video, it can be detected by its sharp edges and high spatial frequency. A change in the direction of the

players is detected using motion vectors in the video. By developing temporal models of these key events, they reported a high accuracy rate.

In [24], the authors present a method to detect high-level human activity in compressed MPEG videos. From motion vector information in the MPEG movies, walking, kicking, and running sequences were trained and recognized using PCA. They also performed posture recognition using relational graph matching. To increase robustness, skin-color information was used.

In [26], the differences of the scene structures between talk-shows and advertisements were used to perform full advertisement-removal of such video sequences. First, a "shot" is classified by applying a threshold to the rate of change of the color histograms. Shots that have a blank screen are used to separate the show and commercials. The black screen is detected by checking if all color energy in the screen is concentrated in one single bin in the histogram. The number of frames and ratio of the number of repetitive shots in a story (shots having similar color statistics) are used to classify commercials and talk shows.

An algorithm to obtain automatic characterization of comedian monologue discourse was described in [10]. It was accomplished using pauses in the monologues, pitch of the voice, and hand positions and velocities. Isodata clustering was used to characterize the feature space. In their studies, they found that large gestures at long pauses in speech are likely to signify the ending of a joke.

In our research, we are interested in detecting the presence of a person and key gesture events to provide information for differential encoding of the face, hands, and background in presentation videos.

2.2 Gesture Events

The study of gesture has been important for designing a natural human-computer interface. A computerized map that interfaces with users through natural speech and gesture was demonstrated in [18]. In [11], a real-time perceptual user interface system was developed for recognizing natural head nod and shake gestures for naturally acknowledging "Yes" and "No" questions posed in a GUI dialog box.

McNeil [20] proposes several types of hand gesture: *Iconic*, *Metaphoric*, *Beat*, and *Diectic*. *Iconic* gestures are pictorial and have a "close formal relationship to the semantic content of speech". For instance, a speaker may use both hands to form a round shape when describing the circular shape of an object. *Metaphoric* gestures are also pictorial but present an abstract idea rather than a physical object. *Beat* gestures are mainly used to accompany words or phrases that are significant for its discourse-pragmatic content and are characterized by up/down or in/out movements – a politician making a strong verbal claim of "I will not raise taxes" accompanied by the "beating" (up/down movement) of the hand. *Diectic* gestures are pointing gestures that generally have the function of indicating objects and events in the physical world.

McNeil suggests that iconic and metaphoric gestures typically follow a tri-phasic model: preparation – stroke (followed by hold) – retraction. Mirror and anti-symmetric hand movements have also been found to be iconic or metaphoric as well. The evidence for a strong correlation between high-level discourse semantics and mirror/anti-symmetric hand movements is described in [25].

These gesture studies serve as the basis for our computational model for gesture detection. We seek to identify prominent iconic/metaphoric, beat, and diectic gestures to encode these events with higher salience in the video.

2.3 Multimedia Networking

In multimedia networking, there are two major research areas aimed at enhancing the multimedia experience over the Internet: network support for Quality of Service (QoS), and compression technologies.

Currently, networks only provide best-effort service. Under best-effort service, which was initially designed with robustness and scalability in mind, all data flows are treated the same and the networks do not provide any timeliness guarantees, such as maximum end-to-end delay. Hence, current networks cannot

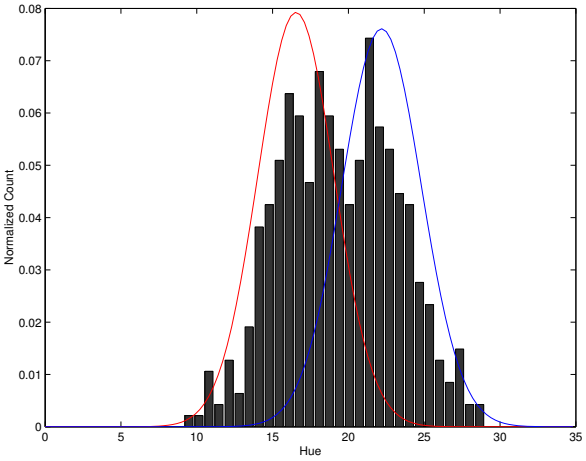


Figure 2: Computed EM skin-color model of Hue with two Gaussians.

fulfill the real-time requirement of multimedia data. To provide network QoS support, Integrated Service [27], Differential Service [7], and Stateless Core [30] frameworks have been proposed to augment the current best-effort service model. These models have yet to be deployed in the Internet.

Due to bandwidth limitations, compression has been an important aspect in providing efficient multimedia delivery. Image compression technologies are typically based on DCT or DWT (wavelet) methods. The Motion Picture Experts Group [1] has led in the development of several open standards for video compression, such as MPEG-1, MPEG-2 and MPEG-4. MPEG-1 was developed in 1988 and has an optimal bit rate of 1.5Mb/s. In 1996, MPEG-2 was designed for use in digital TV broadcasting and has a target bit-rate between 4 and 9Mb/sec. MPEG-4, introduced in 1999, was developed specifically for its mobile delivery, with an optimal bit rate of 385 to 768 Kb/sec. For streaming video, the most dominant proprietary standards are RealMedia [2], Quicktime [3], and Advanced Streaming Format [4]. These technologies exploit motion prediction and compensation of the video.

In our research, we segment the face and hands in each frame with the goal of coding these regions at higher resolution than the background. Our approach may be applicable to the MPEG-4 object layers specification.

3. Vision Algorithm

In our system, computer vision algorithms are initially used to segment and track the presenter's face and hands in the video, followed by gesture analysis to identify key gesture events. The implementation was done under a Windows environment using Intel IPL and OpenCV libraries [15].

3.1 Segmentation

Color-based image segmentation [9, 13, 14] is first used to detect skin-colored pixels. We employ a probabilistic segmentation of human skin pixels using the Hue component of the HSI color space [31].

We begin by training a statistical skin-color model. We manually select a number of skin pixels in the first image of the sequence and model the distribution of their Hue values as a mixture-of-Gaussians using the EM algorithm [12]. The probability function is given as:

$$P(H;\pi,\sigma,\mu) = \sum_{i}^{N_C} \pi_i \cdot \frac{1}{\sqrt{2\pi\sigma_i^2}} \cdot e^{-\frac{(H-\mu_i)^2}{2\sigma_i^2}} \quad (1)$$

The mixture is specified by the parameter set

$$\theta = \{\pi_i, \mu_i, \sigma_i\}_{i=1}^{N_c} \tag{2}$$

Given a training set of Hue values $\{H_{t=1}^{N_T}\}$, the mixture parameters can be estimated using the ML principle:

$$\theta^* = argmax[\Pi_{t=1}^{N_T} P(H^t|\theta)] \tag{3}$$

The estimation problem is solved using the EM algorithm, which consists of E- and M-Steps:

E-Step:

$$w_i^k(t) = \frac{\pi \cdot g(H^t; \mu_i^k, \sigma_i^k)}{\sum_{j=1}^{N_c} \pi_j^k \cdot g(H^t; \mu_j^k, \sigma_i^k)} \tag{4}$$

M-Step:

$$\pi_i^{k+1} = \frac{\sum_{t=1}^{N_T} w_i^k(t)}{\sum_{i=1}^{N_c} \sum_{t=1}^{N_T} w_i^k(t)} \tag{5}$$

$$\mu_i^{k+1} = \frac{\sum_{t=1}^{N_T} w_i^k(t) \cdot H^t}{\sum_{t=1}^{N_T} w_i^k(t)} \tag{6}$$

$$\sigma_i^{k+1} = \frac{\sum_{t=1}^{N_T} w_i^k(t) \cdot (H^t - \mu_i^{k+1})^2}{\sum_{t=1}^{N_T} w_i^k(t)} \tag{7}$$

where $g(\cdot)$ is a Gaussian probability. The EM algorithm is monotonically convergent in likelihood and converges to a local maximum in the total likelihood of the training set. The result for a set of skin pixel Hues using two Gaussians is shown in Fig. 2.

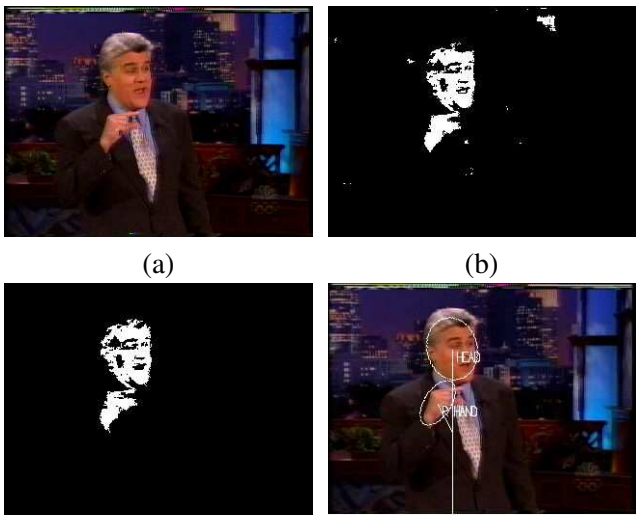
From the trained model, skin pixels in the video images are detected by examining the probability of each pixel Hue belonging to the skin class. If the probability of a pixel's Hue is greater than a certain threshold T_{skin}, it is classified as a skin-colored pixel (See Fig. 3.b).

From the detected skin-colored pixels, regions are formed using connected components. If the size of a region is below T_{size}, it is considered as noise and removed. Since the positions of head and hands in the current frame cannot be too far from their positions in the previous frame (from the tracking algorithm to be discussed), we also impose a maximum velocity constraint on the head and hand regions in the following frames. Skin pixels outside the predicted velocity constraint region are removed (See Fig. 3.c). A 2-D EM-Clustering algorithm is then used to map elliptical regions to the clusters [16] (See Fig. 3.d). This algorithm is robust for cases when two or more skin regions are joined (e.g., when the hand touches the face).

3.2 Tracking

From the detected face and hand regions, smooth trajectories are formed throughout the sequence using a Kalman Filter [17]. In many video sequences, the hands may "disappear" from the screen (e.g., in pockets or out of frame) and the face may be missing (presenter turns around). Whenever the head is missing, the search for the head in the subsequent frames is restricted to the upper half of the image, looking for the re-appearing head.

The missing hand(s) are detected in a similar manner, except that the search area is the bottom half of the image. In the case when both hands are missing, the classification of a re-appearing hand is done by its spatial relation to the head. If the new appearing hand is to the right of the head, it is classified as the right hand, else it is classified as the left hand (See Fig. 4). In the case of both hands re-appearing in the same frame, the classification of the handedness is done by their relative positions.



(a) (b)

(c) (d)

Figure 3: Face and hands segmentation. (a) Original image. (b) Detected skin pixels. (c) Noise removal. (d) EM ellipse fitting.

Figure 4: Classification of the reappearing right hand.

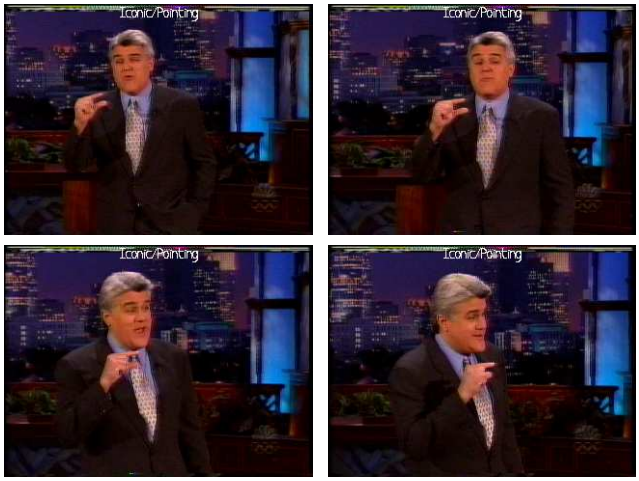


Figure 5: Using relative motion, this iconic gesture is continuously detected as the person turns.

3.3 Gesture Event Detection

For the hand gesture analysis, we aim to detect the following types of gestures: iconic/metaphoric, diectic, and beat. The gestures are detected based on the trajectories of the hands and position of the face. As discussed in Section 2.2, iconic and metaphoric gestures typically follow a tri-phasic model (preparation – stroke (hold) – retraction) or a form of mirror or anti-symmetry movement. For the diectic gestures, we intuitively believe that they also follow a similar tri-phasic model.

We first determine the natural rest-state hand position of the presenter in the video. The natural rest-state of the hands is important to differentiate the rest-state hand-holds and gesture hand-holds (as part of the tri-phasic model). The natural rest-state position is found by locating the lowest hand position in the video sequence, since it is the position at which the speaker does not need to spend any movement effort. Whenever a hand-hold is detected, if it is within a selected vertical distance from the rest position, then it is classified as a rest-state. Otherwise, it is considered as a possible gesture hold.

To detect the iconic, metaphoric, and deictic gestures, which follow a tri-phasic model, we detect the post-stroke hold part of the gesticulation process first since the meaning of the gesture is usually contained within this phase. To detect the post-stroke hold, the relative motion between the hand and head must be less than $T_{relative}$ for at least 2 consecutive frames (i.e., the hand is relatively stable in relation to head). The idea of using relative motion between the hand and head is that the presenter may move during the gesticulation or there may be camera motion (See Fig. 5). Since a post-stroke hold typically does not last long [8], we also temporally filter out post-stroke holds that are longer than 10 seconds.

Mirror and anti-symmetric movements of the hands are detected using the velocity and position of the hands. If both hands have a small vertical distance, share the same velocity sign, and have a similar velocity magnitude for at least 3 consecutive frames, they are classified as mirror symmetric. The hands are classified as anti-symmetric when they have the same conditions as above, except they have a different velocity sign.

The beat gestures are detected by identifying repetitive up-down movements of a hand. First, changes in the vertical hand direction with small horizontal motion are verified. Beat gestures are detected if there are at least 3 such consecutive vertical changes, with each occuring within 2/3 second.

The results of the vision and gesture process are the locations of the face and hands with accompanying gesture event labels. This information is used as input to the video coder to perform differential coding on each frame in the video.

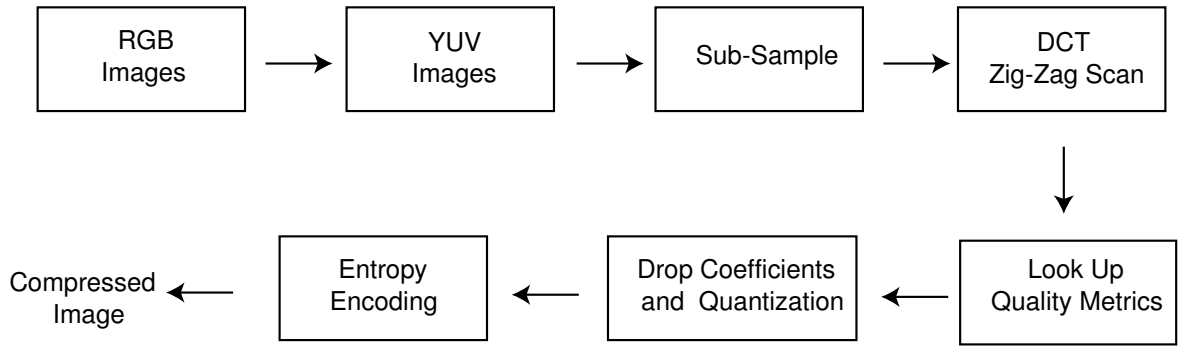


Figure 6: Compression overview.

4. Video Coder

In our system, the video coder performs differential compression on different parts of the image to accomodate efficient, yet perceptually communicative, transmission of the video. Like most existing video coders, we employ a DCT-based compression scheme. Our coder also exploits motion prediction of the video to provide temporal compression.

There are two inputs to our video coder. First is the location of face and hands to tell the video encoder where to retain higher resolution. The second input is the associated quality measurements (number of DCT coefficients, quantization factor) for the face, gesturing hands, non-gesturing hands, and background.

4.1 Image Compression

There are five major steps in our compression scheme (as shown in Fig. 6). First, the image is transformed from RGB to YUV, so as to later sub-sample with a ratio of 4:1:1. Second, forward DCT is applied in each 8x8 pixel block. Third, we calculate the quality measurement for a each macroblock. In the fourth step, we perform compression on the macroblock based on the information from the vision and gesture process. Lastly, entropy encoding is applied to provide further lossless compression.

In the first step, each image is first transformed from an RGB to YUV color space, where Y is the luminance and U,V are the chrominance of the pixel. The conversion formula is given by

$$\begin{bmatrix} Y \\ U \\ V \end{bmatrix} = \begin{bmatrix} 0.299 & 0.587 & 0.114 \\ -0.147 & -0.289 & 0.437 \\ 0.615 & -0.515 & -0.100 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} \tag{8}$$

Sub-sampling is then performed to exploit the psycho-visual redundancy of our eyes which are more sensitive to luminance rather than chrominance. Next, the image is divided into a set of 16x16 macroblocks with each containing 16x16 pixel information of luminance and two 4x4 pixel information of chrominance.

At this step, forward DCT is applied to each 8x8 pixel block to transform the pixels to the frequency domain. The DCT coefficients are then arranged (vectored) in order of increasing frequency. The quality of each macroblock is then calculated. To provide smoother degradation of quality for macroblocks near the objects of interests (face and hands), the quality of each macroblock is linearly scaled from the background quality to the object of interest quality, depending on the number of macroblock pixels associated with the object of interest.

Based on the quality metric calculated at the previous step, coefficients are dropped accordingly starting from the highest frequency. The remaining coefficients are then divided by the quantization factor to further decrease the number of bits needed.

The last step is entropy encoding. This is to further compress the data in a lossless manner. Currently, we employ Run-Length Encoding. Other possible schemes include Huffman and Arithmetic Encoding.

4.2 Motion Prediction

Since much of the scene usually does not change much, we can utilize motion prediction to send less information over the network. A macroblock is defined to be unchanged if the difference of the pixel values between the current the previous macroblock is less than T_{same} (1500). For a macroblock change less than T_{diff} (2500), we send only the difference. If the macroblock has a change larger than T_{diff}, the full information of the macroblock is sent. We therefore have three types of macroblock: TOTAL (where full information is sent), DIFF (where the difference is sent), and NONE (no information is sent).

4.3 Macroblock Format

The resulting format of a macroblock is as follows:

[**TYPE**]
[**QuantizationFactor**]
[**DCSize, DCCoefficient**] **(6 Fields)**
[**ACSize, ZeroRunLength, ACCoefficient**] **(6 fields)**

[TYPE] is a 2-bit field which specifies the macroblock format (TOTAL, DIFF, or NONE). [QuantizationFactor] is an 8-bit field specifying the quantization factor. There are six blocks of [DCSize, DCCoefficient] and [ACSize, ZeroRunLength, ACCoefficient] to encode four 8x8 pixel blocks of luminance and two 8x8 pixel blocks of chrominance. The [DCSize] field takes up 4 bits to specify the number of bits the DC coefficient requires. The [DCCoefficient] field is the quantized DC coefficient. The [ACSize] field takes up 4 bits to specify the number of bits needed to represent the leading AC coefficient. The 6-bit [ZeroRunLength] field is used to specify the number of zeroes preceding the AC coefficient. The [ACCoefficient] is the quantized AC coefficent. The end of the 8x8 vectored coefficients is signified by [ACSize = 0, ZeroRunLength = 0].

5. Experimental Results

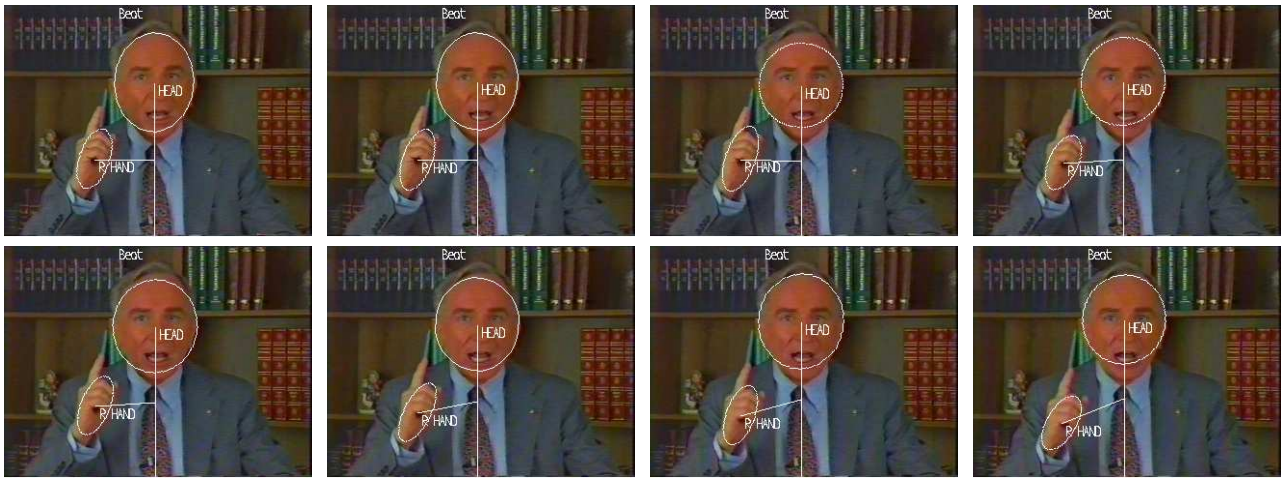
To test the proposed system, we examined three video sequences recorded from television, each containing a single person talking to the camera (or audience). Each sequence is approximately 1000 frames in length. We compare our method against an existing uniform compression method.

For each sequence, with manual identification of the face and hands in the initial frame, the output of segmentation and tracking in the subsequent frames were robust, always locating the face and hands in the image (See Fig. 7). However, it is important to note the shortcomings of our algorithms. The potential problems with any color-based segmentation and tracking approach are that the algorithms may give incorrect results when severe changes in lighting or camera motion exist. Fortunately, within our domain, we can safely assume that the imaging conditions are relatively stable. Also, during the tracking process, there is a possibility of handedness misclassification (reversing the left-right label assignment). This does not currently a pose problem for our system, as the handedness information is not used in the gesture analysis.

For the gesture analysis, it is difficult to measure the success rate due to the subjectivity of certain gestures. One possibility would be to first have the sequences linguistically transcribed and then compared to the event labels produced by our system. For our sequences, we focused on the more obvious gestures. Most of the prominent iconic/metaphoric/diectic gestures were correctly identified (See Fig. 8). Misclassifications happened when the hands were resting at a position away from the natural rest-state location (See Fig. 9).

(a)

(b)



(c)

Figure 7: Segmentation and tracking results. (a) Sequence 1. (b) Sequence 2. (c) Sequence 3.

Figure 8: Example Iconic/Metaphoric/Diectic gestures detected.

Figure 9: Misclassification of iconic gestures when the rest-state is located out of the frame.

The occurence of this specific error is minimized by using the temporal filter adopted in Section 3.3. The mirror and anti-symmetry of hand movements are also correctly determined (See Fig. 10), with only a few frames of misclassification. In sequence 3, the beat gestures in the video sequences are correctly detected (See Fig. 11), except when the hands enter and exit the frames. In the presence of any such errors as described, the cost of the resulting increase in bandwidth is minimal.

To quantitatively measure the quality of the resulting videos, we measure the Weighted Peak Signal to Noise Ratio (W-PSNR). This PSNR calculation is weighted in the manner that the face and hand regions have a higher signal content than the background (the underlying assumption of our approach). The W-PSNR is calculated as:

$$W\text{-}PSNR_{total} = \sum_{m=1}^{N} w_m \cdot PSNR_m \tag{9}$$

where N is the total number of macroblocks, and the weighting factors w_m are 12 (if the macroblock is in the

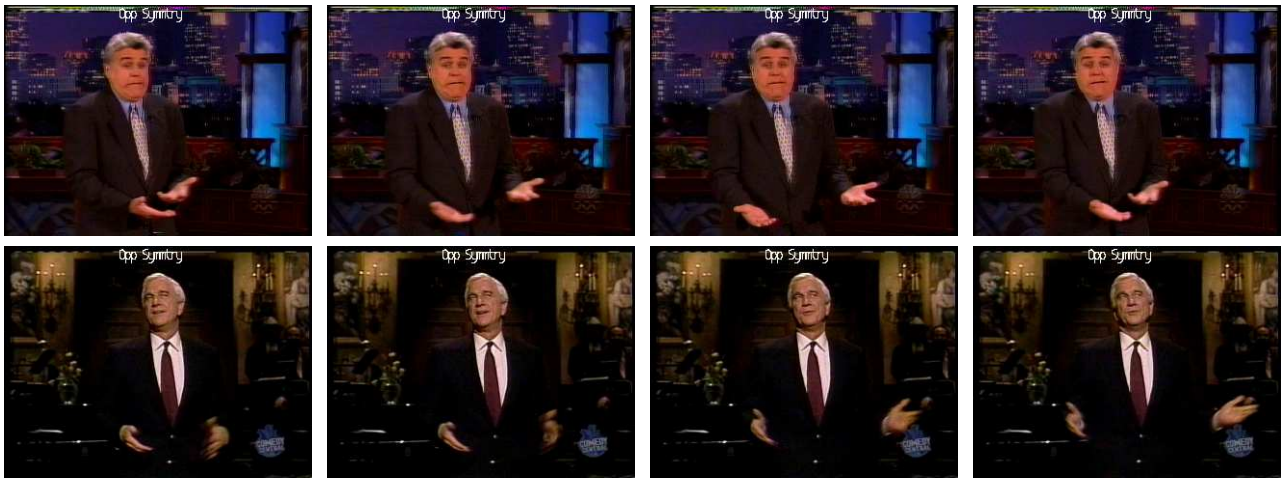


Figure 10: Example mirror and anti-symmetric hand movements detected.

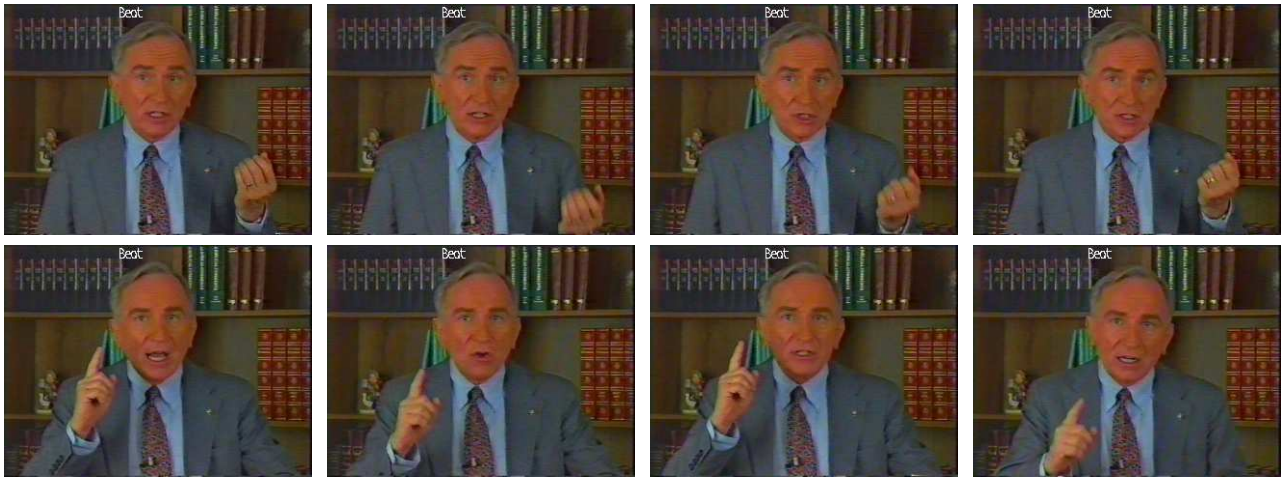


Figure 11: Example beat gestures detected.

Table 1: Comparison between uniform compression (UC) and the proposed method for a frame containing the face and no hands.

Seq #	W-PSNR (dB)		File Size KB	
	UC	Proposed	UC	Proposed
Seq. 1	11.53	11.77	2.57	2.32
Seq. 2	10.56	10.77	1.41	1.36
Seq. 3	30.68	31.12	3.12	3.04

Table 2: Average comparison between uniform compression (UC) and the proposed method for a frame containing the face and non-gesturing hand(s).

Seq #	W-PSNR (dB)		File Size KB	
	UC	Proposed	UC	Proposed
Seq. 1	17.47	18.27	2.03	2.23
Seq. 2	10.87	10.92	1.71	1.68
Seq. 3	34.13	34.77	3.09	3.13

face region), 5 (if the macroblock is in the gesturing hand region), 1 (if the macroblock is in a non-gesturing hand region), and 0.1 (if the macroblock is in the background). We summarize the W-PSNR and file-size compression results in Tables 1-4 and in Fig. 12. The proposed method usually results in a slightly smaller video sequence than produced by an existing uniform compression method, with our version perceptually more communicative.

6. Summary and Future Work

We presented a novel video compression scheme based on the presence and gesturing of a person in the video. We use computer vision algorithms to first segment and track the face and hand regions in the sequence. Next, we perform gesture analysis of the speaker to identify key gesture events. The output of the vision and gesture modules is fed into a video coder to perform differential encoding of the face, hands, and background. We tested our approach with three long-duration video sequences. The vision, gesture, and coding results were very encouraging. Under the same bandwidth limitation, our proposed method yielded a better quality video over an existing uniform compression method.

Since Internet traffic is constantly fluctuating, bandwidth availability is thus continuously changing. To adapt to this variability, our proposed video coder could make use of the output of the vision and gesture algorithms to provide a better adaption to the fluctuating bandwidth. When bandwidth availability increases, the video coder can increase the quality of the human first, then increase the quality of background if any bandwidth remains. When the bandwidth availability drops, the video coder can first choose to drop only the quality of the background.

Future work includes extending the regions of interest to include non-human content. For instance, in the case of distance learning, a white-board (or projector) region in the background could be selected (or identified) as a special area and thus be encoded at a constant high resolution. The segmenting and tracking of the human could also be made more robust by integrating other algorithms employing shape, motion, and templates. We also seek to track multiple people. As there is evidence that people tend to pay more attention to certain gestures than others, we may also do a further investigation on natural gesture to better provide insight into a more adaptive resolution mechanism.

The proliferation of multimedia through the Internet has provided us an additional source of entertainment and information. However, not all of us are able to enjoy it at the best quality, making online video

Figure 12: Selected key frames comparing uniform compression (left column) and our proposed approach (right column). Our differential coder produces cleaner face and hand regions.

Table 3: Comparison between uniform compression (UC) and the proposed method for a frame containing the face and gesturing hand(s).

Seq #	W-PSNR (dB)		File Size KB	
	UC	Proposed	UC	Proposed
Seq. 1	20.91	22.14	2.41	2.70
Seq. 2	11.22	11.59	1.60	1.53
Seq. 3	35.01	35.49	2.58	2.66

Table 4: Average comparison between uniform compression (UC) and the proposed method.

Seq #	Ave. W-PSNR (dB)		File Size KB/Frame	
	UC	Proposed	UC	Proposed
Seq. 1	15.4	15.95	3.023	3.021
Seq. 2	13.36	13.81	1.305	1.29
Seq. 3	32.54	33.17	2.80	2.79

streaming a poor experience. Our current and future research seek to help alleviate some of these problems by focusing on the human content in the video.

References

[1] `http://mpeg.telecomitalialab.com`.

[2] `http://www.real.com`.

[3] `http://www.apple.com/quicktime`.

[4] `http://www.microsoft.com/windows/windowsmedia`.

[5] IEEE Workshop on Detection and Recognition of Events in Video, July 2001.

[6] A. Amir, D. Ponceleon, B. Blanchard, D. Petkovic, S. Srinivasan, and G. Cohen. Using audio time scale modification for video browsing. In *Hawaiian Int. Conf on system Sciences*, January 2000.

[7] S. Black, D. Black, M. Carlson, E. Davies, Z. Wang, and W. Weiss. An architecture for differentiated services. In *Internet Draft RFC 2475*, December 1998.

[8] R. Bryll, F. Quek, and A. Esposito. Automatic hand hold detection in natural conversation. In *IEEE Workshop on Cues in Communication*, December 2001.

[9] J. Cai, A. Goshtasby, and C. Yu. Detecting human faces in color images. In *International Workshop on Multi-Media database Management Systems*, pages 124–131, August 1998.

[10] M. Casey and J. Wachman. Unsupervised cross-modal analysis of professional discourse. In *Proceedings of the Workshop on the Integration of Gesture and Language in Speech*, October 1996.

[11] J. Davis and S. Vaks. A perceptual user interface for recognizing head gesture acknowledgements. In *ACM Workshop on Perceptual User Interfaces*, November 2001.

[12] N. M. Dempster, A.P. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statististical Society B*, 39:185–197, 1977.

[13] M. Fleck, D. Forsyth, and Chris Bregler. Finding naked people. In *European Conference Computer Vision*, volume 2, pages 592–602, 1996.

[14] K. Imagawa, S. Lu, and S. Igi. Color-based hands tracking system for sign language recognition. In *Proc. International Conference on Face and Gesture Recognition*, pages 312–317, April 1998.

[15] Intel Open Source Computer Vision Library, `www.intel.com/research/mrl/research/opencv`.

[16] T. Jebara. Head and hand tracking. Technical Report 507, MIT Media Lab, September 1999.

[17] R.E. Kalman. A new approach to linear filtering and prediction problems. *Transactions of the ASME–Journal of Basic Engineering*, 82(Series D):35–45, 1960.

[18] S. Kettebekov and R. Sharma. Understanding gestures in multimodal human computer interaction. *International Journal of Artificial Intelligence Tools*, 2(9):205–223, September 2000.

[19] F. Kuo, W. Effelsberg, and J.J. Garcia-Luna-Aceves. *Multimedia Communications: Protocols and Applications*. Prentice Hall PTR, 1998.

[20] D. McNeil. *Hand and Mind: What Gestures Reveal About Thought*. University of Chicago Press, Chicago, 1992.

[21] S. Nepal, I. Srinivasan, and G. Reynolds. Automatic detection of 'goal' segments in basketball videos. In *ACM International Conference on Multimedia*, pages 261–269, September 2001.

[22] W. Niblack, S.Yue, R. Kraft, A. Amr, and N. Sundaresan. Web-based seaching and browsing of multimedia data. In *IEEE Int. Conf. on Multimedia and Expo*, July 2000.

[23] S. Nobe, S. Hayamizu, O. Hasegawa, and H. Takahashi. Are listeners paying attention to the hand gestures of an anthropomorphic agent? An evaluation using a gaze tracking method. In *International Gesture Workshop*, 1997.

[24] B. Ozer, W. Wolfe, and A. Akansu. Human activity detection in MPEG sequences. In *IEEE Workshop on Human Motion*, pages 61–66, December 2000.

[25] F. Quek, D. McNeil, R. Ansari, X. Ma, R. Bryll, S. Duncan, and K. E. McCullough. Gesture cues for conversational interaction in monocular video. In *International Workshop on Recognition, Analysis and Tracking of Faces and Gestures in Real-Time Systems*, pages 119–126, September 1999.

[26] M. Shah, O. Javed, and Z. Rasheed. A framework for segmentation of talk and game shows. In *IEEE International Conference on Computer Vision*, July 2001.

[27] S. Shenker, R.Braden, and D. Clark. Integrated services in the internet architecture: An overview. In *Internet Draft RFC 1633*, June 1994.

[28] S. Srinivasan, D. Petkovic, D. Ponceleon, and M. Viswanathan. Query expansion for imperfect speech: Applications in distributed learning. In *IEEE Workshop on Content-based Access of Image and Video Libraries*, June 2000.

[29] S. Srinivasan, D. Ponceleon, A. Amir, and D. Petkovic. What is in that video anyway? In search of better browsing. In *Proceedings of IEEE Int. Conf On Multimedia Computing and Systems*, pages 388–392, June 1999.

[30] I. Stoica. *Stateless Core: A Scalable Approach for Quality of Service in the Internet*. PhD thesis, Carnegie Mellon Univeristy, December 2000.

[31] B. Zarit, B. Super, and F. Quek. Comparison of five color models in skin pixel classfication. In *International Workshop on Recognition, Analysis and Tracking of Faces and Gestures in Real-Time Systems*, pages 58–63, September 1999.