Available online at www.sciencedirect.com
SCIENCE DIRECT®
Computer Vision and Image Understanding 96 (2004) 200–215
Computer Vision and Image Understanding
www.elsevier.com/locate/cviu

Differential video coding of face and gesture events in presentation videos

Robin Tan, James W. Davis*

Computer Vision Laboratory, Department of Computer Science and Engineering, Ohio State University, USA

Received 14 March 2002; accepted 2 February 2004
Available online 7 August 2004

Abstract

Currently, bandwidth limitations pose a major challenge for delivering high-quality multimedia information over the Internet to users. In this research, we aim to provide a better compression of presentation videos (e.g., lectures). The approach is based on the idea that people tend to pay more attention to the face and gesturing hands, and therefore these regions are given more resolution than the remaining image. Our method first detects and tracks the face and hand regions using color-based segmentation and Kalman filtering. Next, different classes of natural hand gesture are recognized from the hand trajectories by identifying gesture holds, position/velocity changes, and repetitive movements. The detected face/hand regions and gesture events in the video are then encoded at higher resolution than the remaining lower-resolution background. We present results of the tracking and gesture recognition approach, and evaluate and compare videos compressed with the proposed method to uniform compression.
© 2004 Elsevier Inc. All rights reserved.

1. Introduction

Initially, the Internet was mostly used to communicate and share textual forms of data. Today, the Internet includes a rich medley of multimedia audio-visual

* Corresponding author. Fax: +1 614 292 2911.
E-mail address: jwdavis@cse.ohio-state.edu (J.W. Davis).

1077-3142/$ - see front matter © 2004 Elsevier Inc. All rights reserved.
doi:10.1016/j.cviu.2004.02.008

data, and has become a center of information, education, and entertainment. But the real-time network delivery of multimedia data presents special challenges. For video sequences, the frame rate of the video needs to be reasonably fast (at least 16 FPS [15]) with jitter-free, high picture quality. These issues have been addressed by employing client-side buffering, forward error correction, piggy-backing, and streaming of data. However, these general approaches are still not enough to provide high user satisfaction in many cases. Computer vision techniques are now being explored as a means to help analyze and segment video to select particular contextual objects, regions, or events of interest to achieve a higher-salience encoding of video (e.g., [26,16,31,19]). In this paper, we examine a special class of presentation videos in which we identify and track face/hand regions and gesture events to produce a compressed video that retains its visually communicative content.

Presentation videos consist of a single person giving a talk or lecture to the camera or an audience (e.g., for distance learning). Viewers of these types of videos typically focus on the presenter, rather than the background scene (except for perhaps an accompanying projected display). Therefore, a video coder could give special emphasis (more resolution) to the face and hand regions, which are the most communicative regions in the images, while reducing the quality of the remaining background. Furthermore, since viewers tend to pay more attention to hands when they are *gesturing* [20], higher resolution can be assigned to the gesture event regions and medium quality to non-gesturing hand regions (though still more resolution than the background). The main idea is that presentation videos could be differentially encoded such that image pixels belonging to the face and (gesturing) hand regions receive more emphasis than the surrounding non-informative background pixels (assigned lower quality resolution). The result should provide a compact, yet informative and communicative viewing of the video.

The main contribution of this research is a multidisciplinary approach that integrates computer vision, gesture analysis, and multimedia networking to enhance the communicative content of compressed presentation videos. Initially, skin-colored pixels are detected in the images using a Gaussian mixture-model trained on skin hue to locate candidate face and hand regions. After region-growing, the selected regions are tracked throughout the sequence using a Kalman filter, taking care to handle cases when the hands enter and exit the scene. We next apply special event detectors to recognize basic hand-gesture categories. These gesture events are used to give special emphasis (higher resolution) to the gesturing hand regions. The tracking and gesture recognition results in each frame are then used to assign each coding block in the image a quality of either {LOW = background, MEDIUM = non-gesturing-hands, or HIGH = gesturing-hands/face}. Lastly, the encoder uses these labels to compress the video, where the higher quality blocks are given higher resolution (more bits). This differential encoding method conserves the bandwidth without sacrificing much of the communicative video quality.

2. Related work on video event analysis

There has been much recent work in the detection of events and actions in video sequences (we point the reader to the recent video-event workshops of [1,2]). The need for classifying and mining large multimedia databases has driven much computer vision research. Many of the applications have been in the areas of video compression, content-based retrieval, surveillance, and human–computer interaction.

IBM's CueVideo system [29] employed audio, video, and text modalities for automatic video summarization, cross-linking of events, and indexing. Keyframes were automatically detected in the video based on its segmentation into shots. The system employed moving storyboards (animated keyframes synchronized with the audio track) for browsing the video content. Indexing was achieved through co-occurences in visual and auditory modalities using color region hashing and speech recognition.

In [18], a method was proposed for automatic goal segmentation in basketball video sequences. This was accomplished mainly by recognizing certain key repetitive events, such as crowd cheer, scoreboard update, and direction changes of the players. The text of the scoreboard display in the scene is artificially embedded in the video, and therefore can be detected by its sharp edges and high spatial frequency. A change in the direction of the players was detected using the motion vectors in the video. The use of temporal models of the key events enabled the system to achieve a high rate of accuracy.

A method to detect human activity in compressed MPEG videos was presented in [22]. From the motion vector information in the MPEG movies, activities such as walking, kicking, and running were modeled and recognized using principal components analysis. Posture recognition was also examined using relational graph matching. Skin-color information was employed in the approach to increase the robustness of person detection.

Differences in the scene structures of talk-shows and advertisements were used in [11] to perform full advertisement-removal in such video sequences. First, a shot was classified by applying a threshold to the rate of change of the color histograms for the images. Shots that have a blank screen were used to separate the show and commercials. The blank screen was detected by checking if all of the color energy in a frame was concentrated in a single histogram bin. The number of frames and the ratio of the number of repetitive shots in a story (shots having similar color statistics) were then used to classify the commercial and talk show segments.

An algorithm to obtain automatic characterization of comedian monologue discourse in video was presented in [6]. Pauses in the monologues, pitch changes of the voice, and hand positions/velocities were clustered using Isodata to characterize the target feature space. In their studies, it was found that large hand gestures occurring at long pauses in speech were likely to indicate an event corresponding to the ending (punchline) of a joke.

In our research, we are interested in detecting the presence of a person (face and hands) and the key hand-gesture events in presentation videos to provide a saliency map for differential encoding of the video to preserve the visually communicative content.

3. Person detection and tracking

We begin by locating the presence of the face and hand regions in the video. A Gaussian mixture-model trained on skin-colored pixels is employed to locate the potential face and hand pixels in the images. The top candidate pixels are selected after region growing and removal of small noise components. A Kalman filter is then used to track the head and hand regions over multiple frames.

3.1. Skin-color segmentation

Our first task is to locate the face and hand regions of the person in the video. Our approach is to train a probabilistic skin-color model using the hue component of the HSI color space, and examine the probability of each pixel (in the image sequence) as belonging to this model. The hue color component was shown to be a robust measure of general skin color [3], and many approaches have been examined for skin-color image segmentation (e.g., [5,9,10,32,3,25,28]).

Those pixels having a high likelihood of the skin-color class are labeled as potential face/hand pixels. In our approach, the likelihood that a pixel (with hue H) belongs to the class of skin pixels is given by

$$P(H|\theta) = \sum_{i=1}^{N_C} \frac{\omega_i}{\sqrt{2\pi\sigma_i^2}} \cdot \mathrm{e}^{-\frac{(H-\mu_i)^2}{2\sigma_i^2}}, \tag{1}$$

where the N_c component Gaussian mixture-model (we currently set $N_c = 2$) is specified by the parameter set $\theta = \{\omega_i, \mu_i, \sigma_i^2\}_{i=1}^{N_c}$ of mixture-weights ω_i, means μ_i, and variances σ_i^2 learned from a set of training data.

To train the model, we manually selected a large number (N_t) of skin pixels (with hues H_t) from the first image of the video sequence and estimated the model parameters θ using the EM algorithm [8] which employs the maximum likelihood principle

$$\theta^* = \mathrm{argmax}\left[\prod_{t=1}^{N_T} P(H_t|\theta)\right]. \tag{2}$$

The approach revises the parameter set θ to increase the total likelihood of training data. This problem is solved iteratively, first estimating the probabilities of the expected labels of the training examples to each Gaussian (E-Step), and then maximizing the joint likelihood of the data and labels (M-Step):

E-Step (iteration k):

$$h_{m,t}^k = \frac{\omega_m^{k-1} \cdot g(H_t; \mu_m^{k-1}, (\sigma^2)_m^{k-1})}{\sum_{j=1}^{N_c} \omega_j^{k-1} \cdot g(H_t; \mu_j^{k-1}, (\sigma^2)_j^{k-1})} \tag{3}$$

M-Step (iteration $k + 1$):

$$\omega_m^{k+1} = \frac{\sum_{t=1}^{N_T} h_{m,t}^k}{\sum_{i=1}^{N_c} \sum_{t=1}^{N_T} h_{i,t}^k}, \tag{4}$$

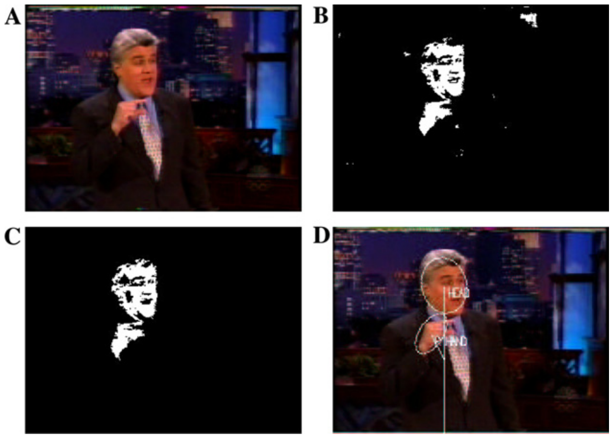


Fig. 1. Face and hands color segmentation. (A) Original image. (B) Candidate skin pixels. (C) Final selected regions. (D) EM clustering of face and hand.

$$\mu_m^{k+1} = \frac{\sum_{t=1}^{N_T} h_{m,t}^k \cdot H_t}{\sum_{t=1}^{N_T} h_{m,t}^k}, \tag{5}$$

$$(\sigma^2)_m^{k+1} = \frac{\sum_{t=1}^{N_T} h_{m,t}^k \cdot (H_t - \mu_m^{k+1})^2}{\sum_{t=1}^{N_T} h_{m,t}^k}, \tag{6}$$

where $g(\cdot)$ is a Gaussian probability and m corresponds to which mixture component is being estimated. The E-Step and M-Step are iterated back-and-forth until the system converges or reaches a maximum number of iterations. The EM algorithm is monotonically convergent to a local maximum in the total likelihood of the training set.

Using the trained model, skin pixels are detected in the remainder of the video sequence by computing the skin-color likelihood of each pixel and comparing it to a threshold T_{skin} (determined empirically). If the likelihood is greater than T_{skin}, the pixel is classified as a candidate skin-colored pixel. The detected skin-colored pixels for the image of a person in Fig. 1A are shown in Fig. 1B.

3.2. Region growing

From the detected skin-colored pixels, regions are formed in each image using a connected components algorithm. If the size of any region is below T_{size} (determined from the minimum expected sizes of the head and hand regions for a given image size), the region is considered as noise and removed from consideration. Also, since the head and hand positions do not typically move far between two frames (for 30 Hz video), we additionally impose a maximum velocity constraint to speed up the detection process by limiting the possible locations of the head and hands in the next frame (i.e., pixel regions outside the predicted velocity constraint area in the next image are ignored). An image of the final selected regions (after applying the connected components algorithm and the velocity constraints) for Fig. 1A is shown in Fig. 1C.

There are cases that can occur when two or more skin regions are joined in the connected components result. For example, the binary image in Fig. 1C contains only one skin region due to the adjacency of the face and hand. As we will need to separately track the face and hands for gesture analysis, a region assignment of {FACE, LEFT-HAND, RIGHT-HAND} is required. Given information on which of the objects are expected to be present in the current image (predicted from the result in the previous frame), we employ a 2-D pixel clustering and segmentation algorithm using the EM algorithm. The location for each region present in the previous image is given to the EM algorithm along with the current connected component image. The result after applying EM is a set of best-fit ellipses (selected at a 2σ Gaussian contour) for the target regions in the current image. The final detected head and hand regions for Fig. 1A are shown in Fig. 1D. Notice that even though the face and hand pixels were merged (connected) in the image (see Fig. 1C), the EM algorithm produced a plausible segmentation of the two regions given the previous frame information in which the two regions were not joined.

3.3. Tracking

In presentation videos, there may be cases when the person turns away from the camera, and hence the face region will not be detected by the skin-color process (or any other face detection scheme). Furthermore, the person may be moving (translating) while facing away from the camera. If the face cannot be located in the image, we search for the face to re-appear in subsequent video frames and restrict the search to the upper half of the image (to account for the person moving in the scene, but not to falsely map to the lower hand regions that may appear). The face tracker is re-initialized when a potential face region is found.

The hands may also frequently disappear from the camera view. For example, the person may have placed his/her hands in clothing pockets. Additionally, when the person turns away from the camera, a hand may be occluded by the body. Furthermore, the camera may at times have a fairly closeup (tight) view of the face, where the hands may often enter and exit at the bottom or sides of the frame. We re-initialize the hand regions in a similar manner as the face, except that the search area is restricted to the lower half of the frame. In the case when both hands are missing, the classification of a re-appearing hand in the next frame is guided by the horizontal location of the hand with respect to the head location. 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. This case is shown in Fig. 2. In the case of both hands re-appearing in the same frame, the handedness is assigned by their relative left–right positions to one another.

From the detected face and hand regions in the images, motion trajectories are formed throughout the sequence using a Kalman Filter [12]. The Kalman filter uses a popular recursive state-based model that can be employed to smooth noisy trajectories and predict future observations. Our state model is composed of the state-update and observation equations

Fig. 2. Classification of the re-appearing right hand in a subsequent frame.

$$\mathbf{S}_{t+1} = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \mathbf{S}_t + \mathbf{R}, \tag{7}$$

$$\mathbf{O}_t = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \mathbf{S}_t + \mathbf{Q}, \tag{8}$$

where the state $\mathbf{S}_t = [x(t), y(t), \mathrm{d}x(t), \mathrm{d}y(t)]^{\mathrm{T}}$ contains the position and velocity of a hand or face region, and $\mathbf{O}_t$ contains the filtered output positional observations $(\hat{x}(t), \hat{y}(t))$ in the image. Kalman recursions are used on the above state-space model to find the optimal state and observation sequences. The Gaussian noise parameters R and Q were determined empirically from our data sets. Currently, the face and each hand are assigned a separate Kalman filter, though a single joint model could be devised. From these trajectories, we next perform the gesture event analysis.

4. Gesture event detection

Recognizing gestures has been an important component in designing a more natural human–computer interface (e.g., [13,7]). We present a simple method to recognize basic natural gesture categories by examining the motion of the hand trajectories. The results will be employed in a video coding scheme to aid in constructing a more communicatively salient compressed video.

4.1. Gesture categories

We define a hand gesture event as either an *Iconic*, *Deictic*, or *Beat* gesture category as described by McNeill [17]. *Iconic* gestures are pictorial and have a close relationship to the semantic content of accompanying speech. For instance, a speaker may use both hands to form a round shape when describing the circular shape of an object. Mirror and anti-symmetric hand movements strongly correlate with high-level discourse semantics and can also be used as Iconic gestures [23]. *Deictic* gestures generally have the function of indicating objects and events in the physical world by "pointing" with the hand. *Beat* gestures are mainly used to accompany words or phrases that are significant for its discourse-pragmatic content and are commonly characterized by repetitive up/down motions (e.g., a politician making a strong verbal claim of "I will not raise taxes" accompanied by the "beating" up/down movement of the hand).

These gesture categories serve as the basis for our model of gesture event detection in presentation videos. We seek to identify the prominent Iconic, Deictic, and Beat gesture events to encode the hand regions with the highest quality when these events are detected in the video sequence. Each class of gesture is currently given equal importance in our coding approach (each assigned the same high quality label).

4.2. Gesture detection approach

The gesture events are recognized based on analysis of the trajectories of the hands and their relative position to the face region. To detect the Iconic and Deictic gesture categories, we look for a post-stroke hold (pause) of the hand (the hold is part of the tri-phasic gesture model: preparation—stroke/hold—retraction). A hold is detected if the motion of a hand is less than T_{motion} for at least two consecutive frames, and the location of the hold is far from the natural rest-state hand position (located automatically by finding the lowest hand position in the video sequence). If the body then begins to move (e.g., the person or camera is moving), the gesture hold can be continuously labeled by checking for a similar relative speed of the head. An example of a hold with a moving/turning body is shown in Fig. 3. Since a post-stroke hold typically does not last long [4], we also temporally filter out potential holds that are longer than 10 s. Other more complex temporal models could also be applied (e.g., Hidden Markov Model).

Other Iconic gestures with mirror and anti-symmetric movements of the hands are detected using position and velocity changes of the hands. If both hands have a small vertical separation, share the same velocity direction, and have a similar velocity magnitude for at least three consecutive frames, they are classified as mirror symmetric. The hands are classified as anti-symmetric when they have the same conditions as above, except that the hand motions have different (opposite) velocity directions. We also temporally filter the gesture labels, where we assign the surrounding gesture label to any small gap of non-labeled frames between the detected gesture frames.

Beat gestures are recognized by identifying any repetitive up–down hand movements. Rather than requiring many beat cycles to perform a frequency analysis, we simply look for changes in the vertical hand motion. We also require small hor-

Fig. 3. Using relative motion, this Iconic gesture is continuously detected as the person turns.

izontal motion in relation to the vertical motion. We label a beat gesture if there are at least three consecutive vertical changes, each occurring within 2/3 s of another (a natural beat frequency).

The identified gesture events, along with the detected face and hand pixel regions, are then passed on to a differential video coder as a guide to compress the video.

5. Differential video coder

Due to bandwidth limitations, compression technology has been an important aspect in providing efficient multimedia delivery. The Motion Picture Experts Group (MPEG) has led in the development of several open standards for video compression. The most dominant proprietary standards for streaming video include RealMedia, Quicktime, and Advanced Streaming Format (ASF). Many of these technologies are based on DCT or DWT (wavelet) methods, and exploit motion prediction/compensation of the video. In our approach with presentation videos, the video coder performs a differential compression of the video where the face and hand image regions are given more resolution than the background to produce a compact, yet perceptually communicative, video for transmission.

There are three inputs to our video coder. First is the video to be compressed. Second is the image mask of the detected face and hand ellipse regions in each frame with any gesture events labeled. Third is the quality values to be associated with the regions for the face, gesturing hands, non-gesturing hands, and background. Together, the image mask and quality labels form a saliency map for each video frame.

Like most existing video coders, we employ a general DCT-based compression scheme, but we additionally include our saliency map to set the quality factors for the image regions. First, the image is transformed from an RGB to YUV colorspace using

$$\begin{bmatrix} Y \\ U \\ V \end{bmatrix} = \begin{bmatrix} 0.299 & 0.587 & 0.114 \\ -0.147 & -0.289 & 0.437 \\ 0.615 & -0.515 & -0.100 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix}, \tag{9}$$

where the RGB color of each pixel is decoupled into its luminance (Y) and chrominance (U, V). Sub-sampling the YUV image is then performed to take advantage of the fact that the human visual system is more sensitive to changes in luminance than chrominance. The image is compressed into 4:1:1, where the U and V images are reduced to a quarter of the size of the Y image (by averaging 2×2 pixel blocks). Next, a forward DCT is applied to each 8×8 pixel block in the YUV image. The DCT coefficients are arranged (vectored) in order of increasing frequency. The YUV image is then partitioned into a set of 16×16 pixel macroblocks, where each macroblock contains a 16×16 pixel block of Y and the corresponding 8×8 blocks of U and V (4:1:1 reduced).

The quality factors assigned to the head/hands in the saliency map (attained from the tracking and gesture recognition algorithms) are used to select the compression level for the DCT macroblocks. The assigned quality values are {LOW = background, MEDIUM = non-gesturing-hands, and HIGH = gesturing-hands/face}. The quality value for each macroblock is used to determine the number of DCT coefficients to retain for each 8×8 DCT sub-block (in Y, U, V) and also to select the corresponding quantization factor. Coefficients are dropped starting from the highest frequency, and the remaining coefficients are divided by the assigned quantization factor. To provide smoother degradation of quality for macroblocks near the face/hand borders, the quality of each macroblock is linearly scaled from the background quality to the region of interest (face/hands) quality based on the percentage of pixels in the macroblock that are within the face/hand region ellipses.

Lastly, entropy encoding of the sequence is applied to provide further compression. Currently, we employ lossless Run-Length Encoding (RLE). Other possible schemes include Huffman and Arithmetic Encoding. Since large portions of the scene typically do not change much, we also utilize a general motion prediction scheme to transmit less information for static regions in the scene. As our approach is object and event driven, it may therefore be applicable to the MPEG-4 object layers specification.

6. Experimental results

To test the proposed approach, we examined three presentation video sequences recorded from television (VHS quality). Each video is approximately 1000 frames in length (33 s), and contains a single person talking to the camera (or audience). We first examined the tracking and gesture recognition approach. Next, we compressed the videos with our differential encoder and compared the results to a uniform compression method in a simple user study. The vision algorithms were implemented under a Windows development environment using the optimized Intel IPL and OpenCV libraries.

6.1. Tracking and gesture recognition results

For each video sequence, the segmentation and tracking results were quite robust, yielding reasonable estimations for the face and hand locations in the video. A sample of the detection and tracking results is shown in Fig. 4 for the three video sequences. Notice that the camera–person distance is different in each video. Due to the subjectivity of certain gestures, it is difficult to rate the success of the gesture analysis method. One possibility would be to first have the sequences linguistically transcribed and then compared to the event labels produced by our system. Instead, we focused on detecting and reporting the more obvious macro-gestures. From manual inspection, most of the prominent Iconic and Deictic gestures were correctly identified. We show a few examples of correctly identified gestures in Fig. 5. In Fig. 6, we show some gesture misclassifications that occurred when the hands were believed to be away from the computed rest-state position (falsely determined to be out of the frame). Most of the mirror and anti-symmetry hand movements were also correctly identified, with only a few frames of misclassification throughout the se-

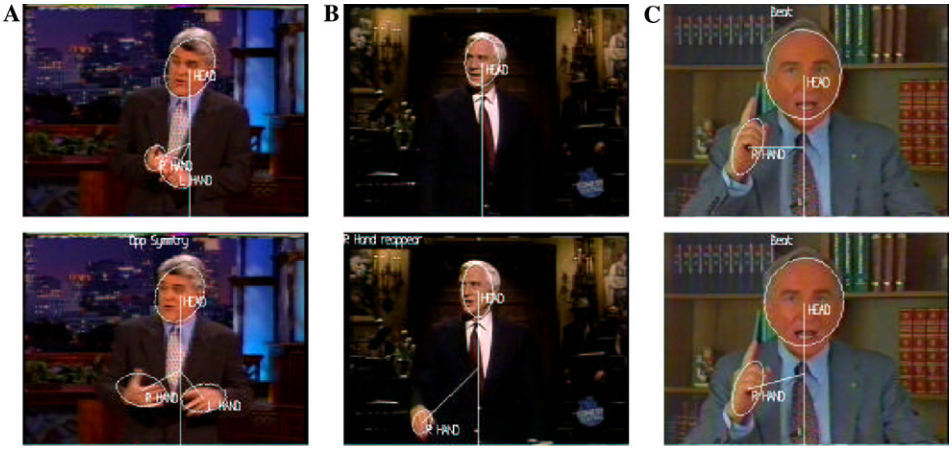


Fig. 4. Example segmentation and tracking results. (A) Video 1. (B) Video 2. (C) Video 3.

Fig. 5. Example Iconic and Deictic gestures detected.

Fig. 6. Misclassification of Iconic/Deictic gestures when the rest-state was falsely computed to be out of the frame.

quences. A detected anti-symmetry (opposite movement) gesture is shown in Fig. 7. Several beat gestures were especially present in video sequence three, and two detected examples are shown in Fig. 8.

Though the overall tracking and gesture recognition approach performed reasonably well, it is important to note the shortcomings of the method. A potential problem with any color-based segmentation and tracking approach is that the algorithm may give incorrect results when there are severe lighting changes. Fortunately, within our domain of presentation videos, the imaging conditions are relatively stable. Another issue to be addressed is that the skin-color model was trained independently for each sequence using only the first frame. Given a larger number of sequences, we could construct a more generalized skin-color model to overcome this limitation. Similarly, the tracking and gesture models were initialized with a manual segmentation of the face and hands in the first frame. Automatic bootstrapping could be accomplished using a model-based approach [30] or employing a face detection algorithm [24,27]. Additionally, there is a possibility of handedness misclassification during the tracking process when the hands join together then separate (potentially

Fig. 7. Example anti-symmetric (opposite) hand movements detected.

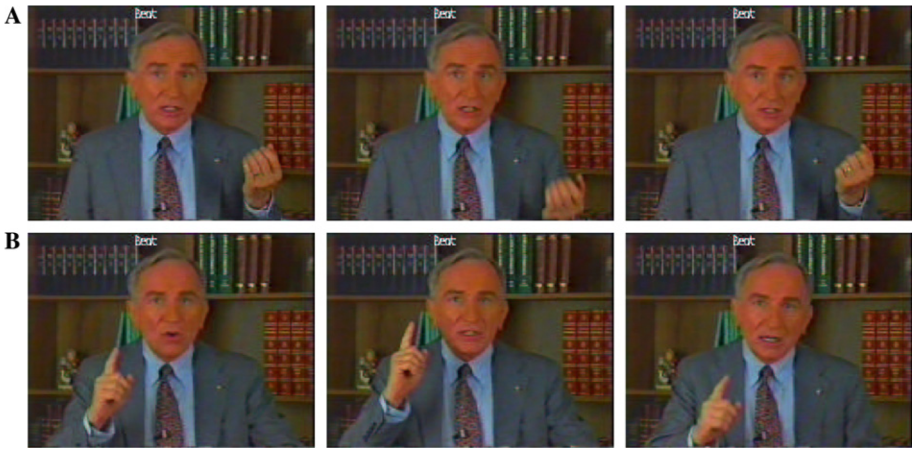
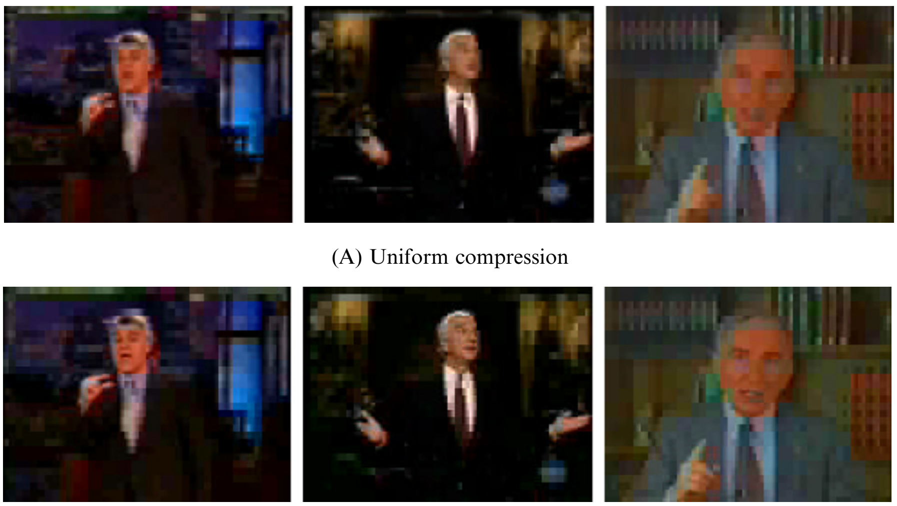


Fig. 8. Two example Beat gestures detected.

reversing the left/right hand assignment), but this does not currently a pose problem for our gesture recognition method as handedness is not formally encoded in the gesture model. In the presence of any minor detection errors that may occur at ceratin frames, the cost of the resulting increase in the overall bandwidth of the video is minimal. Therefore, the coding system is quite tolerant of errors in small portions of the video.

6.2. Comparison of compression results

To evaluate if our differential video encoder (employing the tracking and gesture results) would produce a noticeable visual improvement, we conducted a simple user study comparing the output of our differential encoder to a uniform (flat) compression of the videos. To begin, we encoded the three video sequences with a uniform compression method, where all macroblocks were assigned a low quality. Then we compressed the original sequences with our differential method, with the quality values {LOW = background, MEDIUM = non-gesturing-hands, and HIGH = gesturing-hands/face}. To generate differential file sizes comparable to (yet less than) the uniform compression results, we set the quality of the background region in the differential method to a value slightly less than the uniform compression quality. The average KB/frame for the sequences using the uniform and differential compression methods were 3.02/3.02 for video 1, 1.31/1.29 for video 2, and 2.80/2.79 for video 3. A single frame from each of the sequences for the two coding methods is shown in Fig. 9 for comparison. Notice that the differentially compressed method has fewer artifacts in the face and hand regions.



(A) Uniform compression

(B) Differential compression

Fig. 9. Selected key frames from three video sequences comparing (A) uniform and (B) differential compression.

We examined the compressed video sequences in a simple user comparison experiment. Five subjects were shown the two versions of each video sequence side-by-side (random left–right placement) and asked to select which of the two sequences was more pleasing to watch. Our differentially compressed videos were always found to be more desirable to watch than the uniformly compressed videos. Thus, even under the same bandwidth constraints, our method produced more visually pleasing results.

7. Summary and future work

We presented a multidisciplinary approach to video coding that employs aspects of computer vision, gesture analysis, and multimedia networking. The goal was to develop a compression method for presentation videos (containing a single person talking to the camera/audience) that retains higher resolution in the face and (gesturing) hand regions to achieve low bit-rates while retaining high communicative salience.

Computer vision algorithms are initially employed to segment and track the face and hand regions throughout the video sequence. Next, we performed gesture analysis of the hand trajectories to identify key Iconic, Deictic, and Beat gesture events. The outputs of the vision and gesture modules are used by a differential DCT video coder to compressed the face, hands, and background at different resolutions (with the background assigned the lowest quality). As demonstrated, the proposed method can reduce the non-informative background quality to compensate for the added resolution in the face and hand regions. We tested our approach with three different presentation video sequences. The vision, gesture, and coding results were very encouraging. Under the same bandwidth limitations, the resulting differentially encoded videos were unanimously selected over a uniform compression method by subjects in a side-by-side comparison.

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 temporal adaption to fluctuating bandwidth. For example, when the bandwidth availability drops, the video coder could choose to drop only the quality of the background to retain the most communicative aspect of the video (face and hands). When bandwidth availability increases, the video coder could then increase the quality of the background.

Our future work includes extending the regions of interest to include other non-human content. For instance, in the case of distance learning, a white-board (or projector) region in the background could be selected (or recognized) as a special area and thus could be encoded at a constant high resolution. The segmentation and tracking algorithms could also be made more robust by integrating other techniques employing shape, motion, and templates. We also would like to extend the method to track multiple people in the scene. Since presentation videos also contain an audio track, we could integrate the speech information into a multi-modal system to help identity the gesture events more reliably [14,21,6].

References

[1] IEEE Workshop on Detection and Recognition of Events in Video. July 2001.
[2] IEEE Workshop on Event Mining: Detection and Recognition of Events in Video. June 2003.
[3] G. Bradski, Real time face and object tracking as a component of a perceptual user interface, in: IEEE Workshop on Applications of Computer Vision, 1998, pp. 214–219.
[4] R. Bryll, F. Quek, A. Esposito. Automatic hand hold detection in natural conversation, in: IEEE Workshop on Cues in Communication, 2001.
[5] J. Cai, A. Goshtasby, C. Yu, Detecting human faces in color images, in: International Workshop on Multi-Media database Management Systems, 1998, pp. 124–131.
[6] M. Casey, J. Wachman, Unsupervised cross-modal analysis of professional discourse, in: Workshop on the Integration of Gesture and Language in Speech, 1996.
[7] J. Davis, S. Vaks, A perceptual user interface for recognizing head gesture acknowledgements, in: ACM Workshop on Perceptual User Interfaces, 2001.
[8] A. Dempster, N. Laird, D. Rubin, Maximum likelihood from incomplete data via the EM algorithm, J. R. Stat. Soc. B 39 (1977) 1–38.
[9] M. Fleck, D. Forsyth, C. Bregler, Finding naked people, in: European Conference on Computer Vision, 1996, pp. 592–602.
[10] K. Imagawa, S. Lu, S. Igi, Color-based hands tracking system for sign language recognition, in: Internat. Conf. on Automatic Face and Gesture Recognition, 1998, pp. 462–467.
[11] O. Javed, Z. Rasheed, M. Shah, A framework for segmentation of talk and game shows, in: Internat. Conf. on Computer Vision, 2001, pp. 532–537.
[12] R. Kalman, A new approach to linear filtering and prediction problems, Trans. ASME J. Basic Eng. 82 (Series D) (1960) 35–45.
[13] S. Kettebekov, R. Sharma, Understanding gestures in multimodal human computer interaction, Int. J. Artif. Intell. Tools 9 (2) (2000) 205–223.
[14] S. Kettebekov, M. Yeasin, R. Sharma, Improving continuous gesture recognition with spoken prosody, in: IEEE Conf. on Computer Vision and Pattern Recognition, 2003, pp. 565–570.
[15] F. Kuo, W. Effelsberg, J. Garcia-Luna-Aceves, Multimedia Communications: Protocols and Applications, Prentice Hall, Upper Saddle River, NJ, 1998.
[16] C. Lin, Y. Chang, Y. Chen, Low-complexity face-assisted video coding, in: IEEE Internat. Conf. on Image Processing, 2000, pp. 207–210.
[17] D. McNeill, Hand and Mind: What Gestures Reveal About Thought? University of Chicago Press, Chicago, 1992.
[18] S. Nepal, U. Srinivasan, G. Reynolds, Automatic detection of goal segments in basketball videos, in: ACM Internat. Conf. on Multimedia, 2001, pp. 261–269.
[19] A. Nguyen, J. Hwang, Scene context dependent rate control, in: ACM Internat. Conf. on Multimedia, 2001, pp. 309–318.
[20] S. Nobe, S. Hayamizu, O. Hasegawa, 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, pp. 49–59.
[21] H. Nock, G. Iyengar, C. Neti, Assessing face and speech consistency for monologue detection in video, in: ACM Internat. Conf. on Multimedia, 2002.
[22] B. Ozer, W. Wolf, A. Akansu, Human activity detection in MPEG sequences, in: IEEE Workshop on Human Motion, 2000, pp. 61–66.
[23] F. Quek, D. McNeill, R. Ansari, X. Ma, R. Bryll, S. Duncan, K. McCullough, Gesture cues for conversational interaction in monocular video, in: Internat. Workshop on Recognition, Analysis and Tracking of Faces and Gestures in Real-Time Systems, 1999, pp. 119–126.
[24] H. Rowley, S. Baluja, T. Kanade, Neural network-based face detection, IEEE Trans. Pattern Anal. Mach. Intell. 20 (1) (1998) 23–38.
[25] E. Saber, A. Tekalp, Frontal-view face detection and facial feature extraction using color, shape, and symmetry-based cost functions, Pattern Recogn. Lett. 19 (8) (1998) 669–680.

[26] R. Schumeyer, E. Heredia, K. Barner, Region of interest priority coding for sign language videoconferencing, in: IEEE Workshop on Multimedia Signal Processing, 1997, pp. 531–536.
[27] K. Sung, T. Poggio, Example-based learning for view-based human face detection, IEEE Trans. Pattern Anal. Mach. Intell. 20 (1) (1998) 39–51.
[28] T. Syeda-Mahmood, Y. Cheng, Indexing colored surfaces in images, in: Internat. Conf. on Pattern Recognition, 1996.
[29] T. Syeda-Mahmood, S. Srinivasan, A. Amir, D. Ponceleon, B. Blanchard, D. Petkovic, CueVideo: a system for cross-modal search and browse of video databases, in: IEEE Conf. on Computer Vision and Pattern Recognition, 2000.
[30] C. Wren, A. Azarbayejani, T. Darrell, A. Pentland, Pfinder: real-time tracking of the human body, IEEE Trans. Pattern Anal. Mach. Intell. 19 (7) (1997) 780–785.
[31] Z. Wu, C. Chen, A new foreground extraction scheme for video streams, in: ACM International Conference on Multimedia, 2001.
[32] B. Zarit, B. Super, F. Quek, Comparison of five color models in skin pixel classification, in: Internat. Workshop on Recognition, Analysis and Tracking of Faces and Gestures in Real-Time Systems, 1999, pp. 58–63.