Video Compression Technique Using Facial Landmarks on Mobile Devices

Student Name: Ruchika Banerjee

IIIT-D-MTech-CS-MC-16-MT14053
Jun, 2016

Indraprastha Institute of Information Technology
New Delhi

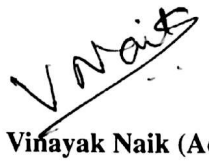
Thesis Committee
Dr. Vinayak Naik
Kuntal Dey
Dr. A V Subramanyam
Nitendra Rajput

Submitted in partial fulfillment of the requirements
for the Degree of M.Tech. in Computer Science,
with specialization in Mobile Computing

Keywords:Video Compression;Facial Landmarks;Dynamic thresholding for facial video compression

Certificate

This is to certify that the thesis titled **"Video Compression Technique Using Facial Landmarks on Mobile Devices"** submitted by **Ruchika Banerjee** for the partial fulfillment of the requirements for the degree of *Master of Technology* in *Computer Science & Engineering* is a record of the bonafide work carried out by her our guidance and supervision in the Mobile Computing group at Indraprastha Institute of Information Technology, Delhi. This work has not been submitted anywhere else for the reward of any other degree.



Dr. Vinayak Naik (Advisor)
Indraprastha Institute of Information Technology, New Delhi

Kuntal Dey (Co Advisor)
IBM Research Lab, New Delhi

Abstract

Wide proliferation of smart mobile phones has manifolded the bandwidth demand, as video streaming applications have significantly gained popularity. At the same time, technical challenges, such as requirements of resources, as well as practical challenges such as limited availability of the mobile bandwidth spectrum, have acted as inhibitors of transmission of unlimited video over the wire in an ubiquitous manner. In this thesis, we propose a first-of-its-kind methodology for compressing videos that stream human faces. Our technique is amenable for streaming transmission of live videos. Our framework relies upon detecting facial landmarks on-the-fly, and compressing the video by storing a sequence of distinct frames extracted from the video, such that the facial landmarks of a pair of successively stored frames are significantly different. The compression technique uses a dynamic thresholding technique to detect significance of difference, and stores meta-information for reconstructing the missing frames. We measure the goodness of our technique by evaluating the time taken to compress, the entropy of successively stored images, and a comparison with several static thresholds of significance. We validate our work with a user study, observing user satisfaction at different compression ratios. Our work will also be useful in applications that require live streaming of facial videos.

Acknowledgments

I would like to express my deepest gratitude to my advisors Vinayak Naik and Kuntal Dey for their constant guidance and support. The quality of this work would not have been nearly as high without their well-appreciated advice. Their ideas and suggestions have given me the motivation to work harder on this project. I would also like to thank my esteemed committee members A.V. Subramanyam and Nitendra Rajput for agreeing to evaluate my thesis work. I would also like to thank all my friends who agreed to be a part of the user study of the project and gave their valuable time. Last but not the least, I would like to thank all my supportive family and friends who encouraged and kept me motivated throughout the project.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Introduction

The adoption of smart mobile phones has proliferated over the past decade. Mobile applications have firmly established their presence among the users of smart phones [21] [14] [13]. Many of the popular applications today use end-to-end streaming video transmission [38] [27] , some noteworthy examples being Skype [1], Youtube [2] and Facebook [3], amongst myriads. Many of these are social applications that generate live user video streams, and video playback applications that have large video repositories; hence, they tend to have a large number of videos that involve human faces. Thus, practically, a large number of videos, comprising of human faces, are transmitted over the network, to mobile applications and web, on a daily basis. [35]

In spite of advent of the newer generations of mobile data network with enhanced data transmission capacity, such as 3G and 4G, the transmission bandwidth and commercially imposed usage limits (such as monthly usage quota) with higher monetary cost on the end-user for larger usage volumes, remains a concern. Video data requires much more network data transmission volume, compared to text and image. In addition, with huge video repositories that have started to exist for video providers, such as Youtube, the space occupied at the server side for videos is also massive.

Clearly, the more compressed a video is, the lesser will be the stress on the precious resources, such as the mobile data network in case of transmission, and server storage space for storing the videos. Since, a large volume of the videos comprise of human faces as one of the primary constituents, we observe that there is an opportunity of significant resource savings, by compressing human face videos. This serves as an incentive to explore the possibility of creating a robust compression methodology for human face videos.

We follow a multi-step approach to compress the human face videos. We adopt a frame-by-frame compression approach, wherein we examine each distinct frame from the video, and decide to either include the frame for transmission, or discard it. In order to decide whether to retain or discard a frame, we first find the facial landmarks, such as eye corners, eye center, lip corners and cheek muscles. This is important because the process of communication of a person leads to variation of facial expressions [23],

[1] https://www.skype.com
[2] https://www.youtube.com
[3] https://www.facebook.com

and thus, facial landmarks. After transmitting a frame f_i, we choose to include the first among the following frames f_j, where the difference between f_j and f_i are significant enough, and otherwise discard the frame. The significance of difference is measured using a dynamic thresholding technique. The system adapts to update the dynamic threshold to permit a higher transmission rate for rapid changes to the facial landmarks (video segments with high variation rates), and a lower transmission rate for video segments that practically have a lesser rate of variations. The number of frames that are discarded is counted, and in a video transmission scenario, when the next frame is selected for transmission by virtue of being significantly different from the previously transmitted frame, the number of discarded frames is also transmitted. This information is used at the receiving end to reconstruct the video, and thereby have the capability to seamlessly play any audio associated with the original video. We measure our dynamic threshold with several static baselines as well as an objective image entropy based measurement, and validate with a user study over different compression ratios. The dynamic threshold is empirically observed to be effective in maximizing the user satisfaction.

It is interesting to note that, our system is easy to use along with traditional video compression systems. While it is indeed true that videos are stored and transmitted in compressed forms and formats, our system selects a subset of frames for storage and transmission, and does not keep the remaining frames except for simple statistical numbers (such as count of discarded frames). Traditional video compression techniques can simply use the output of our system, and compress only the retained subset of frames, instead of all the frames that it would normally compress, thereby allowing to be benefitted from our facial video compression methodology. We experiment with real-life Youtube videos, that supports our claim of obtaining significant additional compression with facial videos, over and beyond what traditional video storage and transmission systems use.

The key contributions of this work are the following.

- We propose a facial landmark based framework for compression of streaming videos of the human face.
- We follow a frame-by-frame facial landmark comparison based decision of frame transmission, using a dynamic threshold that adapts to different variation rates of the facial videos. We maintain meta-information that helps in reconstructing the compressed videos at the receiving end, in spite of the discarded frames.
- We provide a lightweight implementation encompassing the entire framework, and successfully demonstrate the system to perform well on a mobile platform with inherent resource limitations.
- We study the goodness of our system against multiple static baselines, and validate with a robust user study. We further provide an entropy-based objective measurement, showing that the entropy amongst a successive pair of transmitted frames to be higher, than the entropy amongst a pair of successive frames in which one frame was transmitted and the other was not.

The rest of the thesis is organized as follows. In chapter 2, we review the existing literature. In chapter 3, we provide the details of our methodology. Subsequently, in chapter 4, we present our experimental setup, including the platform, application, data and the user study we perform. This is followed by a coverage of the experimental results, in chapter 5. Finally, we have concluded in the chapter 6.

Chapter 2

Related Work

2.1 Related Work

Video compression had been an area of interest in research for the last two decades [19] [20] [28] [31] [33]. The aim of video compression is to reduce image data, which makes storage of videos, transmission of videos easy. Different video compression techniques provide efficient solutions to improve the storage, transmission bandwidth of video files. [24] [25] Video files, as compared to other media files, contain a greater amount of redundant or repetitive data. Compression techniques work on removing different types of redundancies, such as perceptual, temporal and spatial redundancies [15].Removing perceptual redundancies [4] refers to the removal of those minute details in an image which a human eye cannot recognize.Since videos are nothing but a set of frames with each frame composed of an array of pixels, the extent of similarity between two successive frames depends on their frame interval.Temporal compression compresses the amount of video data by detecting similarities between these adjacent pixels in subsequent video frames and encodes the redundant information [3] whereas techniques that perform only spatial compression reduces the data to represent a single frame of video by detecting regions within a frame with similar pixel data and compresses the video data corresponding to those regions [1].

The standard of video compression techniques are decided by the two standard organizations, ITU-T and ISO/IEC. JPEG [34], Motion JPEG [10] and MPEG [15] are the three well-used terms used to describe different types of compression formats. In broader terms, JPEG is associated with still digital pictures whereas MPEG and MJPEG are used for digital video sequences. As described in the US Patent numbered *US 4717957 A* [26], one such way of video compression is where the transmitter detects the areas of the current image that are changed with respect to the previous image, and this information along with any new information of the current picture is sent into the transmission channel. In the receiver a new image is reconstructed on the basis of the previous image and the information received. One of the applications of video compression in the field of security and authentication was proposed by [29]. A video encryption technique is presented in this work, that relies upon examination of the nature of the data to be secured. They aim to identify the sensitive portions of a compressed video to reduce the amount of data to be encrypted.

These techniques presents some of the existing work on video compression. Our work combines video compression with facial landmark detection in video sequences for human face videos.Ekman and Friesen [6] developed the Facial Action Coding System(FACS) for describing facial expressions by a means of

44 Action Units.Each Action Unit is a state of a feature of the face, for example: AU 4 is Brows lowered and drawn together, AU 25 is lips are relaxed and parted. Thus, combinations of many AU's can form a certain expression. Many research works have been done since then to identify these action units mainly by training a mathematical model on existing databases using Neural Networks [32], HMM [17], SVM's and Adaboost [2] etc. The paper [17] focuses on recognizing only 15 AU's in a face. Once the input image is aligned, dense flow extraction module is applied on it to track the flow on the entire face image. Facial feature extraction module is applied on the aligned image to track some pre-selected feature. High gradient combination detection module is separately again applied on the image to detect and track changes in the standard and transient facial lines and furrows with the help of horizontal,vertical and diagonal line and edge detectors. HMM and DBN both were compared to give a recognition rate of 93% and 89% respectively. Another paper [22] has worked on spotting the segments that display facial expressions from image sequences using HMM. Here, the feature extraction is performed using gradient-based optical flow algorithm and the classification is done using HMM. A different attempt has been made in that paper to recognize the intermediary states of the basic emotions such as relaxed, contracting, apex and relaxing between two standard emotions. Interesting relations between recognition rate and frame rate of image sequences, spotting rate and interval between two emotions have been presented and discussed. Emotion Detection from mobile phones is done in this [30] where template matching method is used for the feature extraction and SVM for expression recognition with an accuracy of 72%. In [11], authors have worked on mobile platform using Neural Network and CK Database. Many facial animation systems work by tracking facial features and use the information derived from these features to animate cartoon characters [7]. Any change in expression is caused due to a change in the several features of a face. Facial landmark detection in video sequences have proved to be an important step towards facial recognition and tracking in videos [16] [12], recognizing human facial expressions to understand the level of interests in a video [37],facial gesture recognition [8]. There has been no specific work on implementing video compressing techniques based on facial landmarks. Though a very unique technique of video compression is described in [9] which predicts the priority region in a video stream using a neurobiological model of visual attention and the compresses the video file according to its priority regions. This paper uses pixel wise video compression technique by identifying relevant areas based on the model trained on the human eye movements on the unconstrained video input.

Chapter 3

Methodology

We discuss the algorithm in this chapter. The algorithm takes a video as an input and implements the algorithm on it. A detailed step by step explanation is given in the following section.

3.1 Detecting Facial Landmarks

Facial landmarks are the facial feature points, such as corners of eyes and tip of nose, corners of lips. When a person communicates, these landmarks change from frame to frame in a video. Our algorithm first divides input video into frames. Within each frame, it identifies the face of the person and then identifies eight major landmarks as shown in figure 3.1. The landmarks are two eye centers, nose tip, two cheek muscle tip points, two lip corners, and one point in the lower lip jaw. It then records the coordinates of these landmarks.

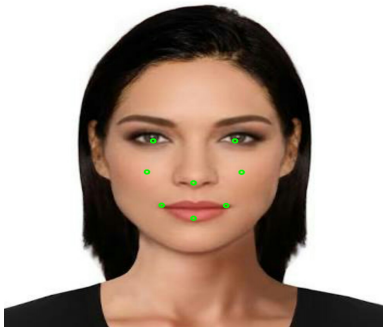


Figure 3.1: Image showing the eight selected facial landmarks for a frame.

3.2 Detecting Distinct Frames

A frame, which differs in position of the landmarks from its previous one, more than a certain threshold is called a *distinct frame*. In principle, the set of distinct frames carry more information than the others.

Once the facial landmarks are detected in a frame, the next step is to store those landmarks and check the difference of the landmarks' position of the next frame in the queue.

Introducing Static and Dynamic Thresholds: The difference in the facial landmarks is compared using two methods: *using dynamic threshold* and *using static threshold*. The threshold to detect distinct frames can be the same for the entire video or it can be variable. If the threshold is constant, it speeds up the processing. However, the downside is that it will hamper the visual quality of the compressed video if the frames have a variable rate of change in a video.

In our framework, we compute the threshold dynamically, instead of using any predefined static threshold value. The dynamic threshold value automatically adjusts itself with respect to change of rate of frames. The threshold is decided by looking at the maximum and minimum change in the difference for any of the eight landmarks' coordinates in two successive frames. A frame is considered to be distinct if the minimum change is greater than half of the maximum change. We experimentally validate the qualitative and quantitative performance of our dynamically computed threshold, against several baseline static threshold values.

3.3 Formulation of the Compression Algorithm

The mathematical formulation of the algorithm follows.

- We calculate the Euclidean distance of each landmark in the current frame from the stored landmark information of the previously transmitted frame. For any landmark j in the $(i+1)^{th}$ frame, the Euclidean distance $d_{i+1,j}$ is given as:

$$d_{i+1,j} = \sqrt{(cx_{i+1,j} - cx_{i,j})^2 + (cy_{i+1,j} - cy_{i,j})^2} \tag{3.1}$$

- A frame *i* is categorized as a distinct frame if it satisfies the following equation:

$$min(d_{i,j}) >= max(d_{i,j})/2 \tag{3.2}$$

where j = 1 to M, where *M* is the number of landmarks detected.

Our algorithm requires only a pair frames to calculate the threshold, thus making it efficient. As an intuitive example, let's say a video comprises of 100 frames, namely $f_1, f_2, ..., f_{100}$. Out of these 100 frames, say frames f_1, f_{12}, f_{23}, f_{24}, f_{25}, f_{47}, f_{55} and f_{90} are distinct frames, and the rest are not. Our algorithm will transmit only these distinct frames, as f_1, f_{12}, $\cdots$, f_{23}, f_{24}, f_{25}, $\cdots$, f_{47}, $\cdots$, f_{55}, $\cdots$, f_{90}, and discard the rest. The sequence numbers of the discarded frames are transmitted as $2, 3, \cdots, 11, 13, \cdots$, and so on, so that any audio associated with the video can be appropriately overlayed with the reconstructed video; however, the image content of these frames are not transmitted. Note that the audio will be transmitted separately, using any standard audio transmission methodology. The algorithm is given in Algorithm 1.

3.4 Decompression

We extract the original video from the compressed one by storing the distinct frames along with the number of times they are to be repeated. Thus, while reconstructing the original video, only the missing frames need to be inserted between the two distinct frames. This makes the compressed video equivalent to the size of the original video. Since the audio is sent along a different channel, it will not get affected by the compression. Thus, same length video can be obtained at the user end by only transferring lesser amount of data. When the compressed video algorithm is used for real time communication, the time for which no distinct image is identified, the previous distinct frame only displayed on the front screen of the other user.

The proposed decompression method works as follows: Continuing with the same example given in section 3.3, we received the compressed video as $f_1, ..., f_{22}, f_{24}, ..., f_{54}, f_{56}, ..., f_{89}, f_{91}, ..., f_{100}$. According to our proposed decompression method, we replicate the previous frame of the discarded frames in their place at the receiver's end. Thus the final output at the receiver's end becomes as follows: $f_1, ..., f_{22}(3times), f_{25}, ..., f_{54}(2times), f_{56}, ..., f_{89}(2times), f_{91}, ..., f_{100}$. The audio was received on a different channel and since the length of the received video after decompression becomes same the original video, the audio remains in synchronization with the original video file.

3.5 Metrics to Evaluate Accuracy and Efficiency of the Algorithm

In this subsection, we mention metrics to measure performance of our algorithm and our method of comparing it.

3.5.1 Metrics

Time Taken By the Algorithm: We measure the efficiency of the algorithm in terms of time taken to compress the video, the time taken to run the algorithm was measured using a separate clock.

Entropy: A successful compression of a video is to reduce its size without affecting much, its quality or information content. To determine the information stored in the video with respect to the entire length of the video, the following evaluation metric was used.

- Calculate the pixel-wise difference between each consecutive frames for all the frames of the entire video.

$$Iz = Ix - Iy \tag{3.3}$$

  where I_x and I_y are 2 consecutive frames of a video

- Calculate the entropy of the difference calculated for each consecutive pair of frames and then take an average value by dividing it to the total number of frame in the video.

$$\frac{\sum_{i=1}^{N-1} E(Iz_i)}{N} \tag{3.4}$$

  Here, N is the total no. of frames in the video, Iz_i is the difference of i^{th} and $i + 1^{th}$ frames, and $E(Iz_i)$ gives the entropy of the difference.

Size of the Compressed Video: One of the most important evaluation metric is to determine the size of the compressed videos in case of each threshold used. These values help us to compare between the size of the original video and the compressed ones.

Baseline using Static Threshold

For comparison, we use a static value of threshold as the baseline. Between two successive frames, every time any landmark changes its position by a margin above that of the threshold, the frame is categorized as a distinct frame. As evident from Figure 3.2, the landmarks positions changes with changing expression on face. If the change is above a certain threshold, the frame is considered to be distinct.

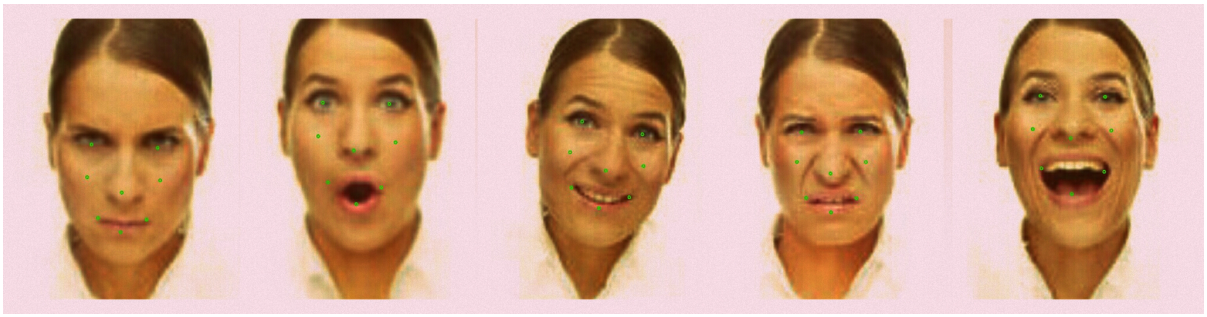


Figure 3.2: A set of images that shows how the facial landmarks changes with different expressions of a person

We considered four static threshold values of 3, 5, 8, and 10. We chose these values empirically.

Algorithm 1 VIDEO COMPRESSION USING DYNAMIC THRESHOLD

$N \leftarrow$ No. of frames
$SequenceNo \leftarrow 1$
Transmit and store ¡the first frame, SequenceNo¿
$lx = 0 \leftarrow$ Landmark information of the x coordinate of the first transmitted frame
$ly = 0 \leftarrow$ Landmark information of the y coordinate of the first transmitted frame
for $k = 2 \rightarrow N$ **do**
    $\{face_k\} \leftarrow$ An array of points detected as a face for the k^{th} frame using Face API
    for $landmark : face_k.getlandmarks()$ **do**
        $cx_{k,j} \leftarrow$ landmark.getposition().x
        $cy_{k,j} \leftarrow$ landmark.getposition().y
        $j = j + 1$
    end for
    $M \leftarrow$ No. of landmarks
    for $j = 1 \rightarrow M$ **do**
        $d_{k,j} = \sqrt{(cx_{k,j} - lx_j)^2 + (cy_{k,j} - ly_j)^2}$
    end for
    for $j = 1 \rightarrow M$ **do**
        $min\_diff \leftarrow MIN(d_{k,j})$
    end for
    for $j = 1 \rightarrow M$ **do**
        $max\_diff \leftarrow MAX(d_{k,j})$
    end for
    if $min\_diff >= max\_diff/2$ **then**
        Mark the k^{th} frame as a distinct frame
        Transmit and store ¡the k^{th} frame, SequenceNo¿
        $lx \leftarrow cx_k$
        $ly \leftarrow cy_k$
    else
        Discard the k^{th} frame
        SequenceNo = SequenceNo + 1
        Transmit SequenceNo
    end if
end for

Chapter 4

Experimental Set Up

In this chapter, we provide the outline of the experimental setup that was used to implement and validate our framework. We provide a brief overview to the mobile platform application we developed for the purpose of our experiments, the facial landmark detection tool used for the development of our application, the database used for our testing purposes, and the user study.

4.1 The Mobile Platform and Application

We conduct our experiments using Google Nexus 5 phone with 1.3 MP front camera Quad-core 2.3 GHz Krait 400 CPU. The mobile application we developed, uses the front camera for ingesting video input, which is technically equivalent to clicking pictures in the burst mode and subsequently storing the frames in the phone memory. This video, treated as a sequence of frames, is fed to our algorithm, wherein the distinct frames are identified for the purpose of retaining (transmission), and the rest are discarded after the meta-information is extracted for compression. Note that, the mobile application is an entirely client-side one, and does not involve any server-side component.

We use Google API for detecting facial landmarks. This provides a robust and well-established platform for facial landmark detection that in turn is used in our application, and makes our code portable across Android devices. However, our approach is not restricted to facial landmarks detected by Google API. It can use any other API that detects facial landmarks.

4.2 Data for Evaluation

We evaluate our methodology on publicly available databases for benchmark. We use two benchmark databases: the Talking Face Video [5] and the Youtube Faces DB [36].

The Talking Face video [5] consists of $1,000$ frames, that corresponds to about 26MB in size, recorded from a video of a person engaged in a conversation. The video was recorded while the person was talking to another person, and was subsequently broken into frames. This database is suitable for evaluation as the subject displayed different facial expressions while talking, such as smiling, laughing, staring silently, etc, as seen in Figure 4.1. This database was released by the research team PRIMA of INRIA Grenoble Rhone-Alpes Research Center, France.

Video Names	Gender	Talking	Smiling / Laughing	Staring Quietly	Hand Gestures	Multiple Expressions	Background Lighting Conditions	Head Movements
$Video_1$	Male	Low	High	High	Medium	High	High	Low
$Video_2$	Female	High	Low	Medium	Low	Low	High	Medium
$Video_3$	Male	High	Low	Low	High	Low	High	Low
$Video_4$	Male	Medium	Medium	Low	High	Medium	Low	Medium
$Video_5$	Female	High	Low	Low	Low	Low	Low	High

Table 4.1: Tables showing the various characteristics of all the Databases rated on a scale of *High, Medium and Low*

We use the Youtube Faces DB [36] as the other benchmark database. These images are the frames received after breaking up videos of people, while they are either interviewing or participating in a press conference. Thus, the main focus of each video is the single person, shown in the frame.

We use 4 videos of 4 different subjects from this database. Figure 4.2, 4.3, 4.4, and 4.5 shows the chosen videos, $Video_2$, $Video_3$, $Video_4$, and $Video_5$ respectively. Table 4.1 contains a description of the characteristics of these videos, e.g. Talking, Smiling/Laughing, etc. These characteristics are found in most of the conversations. The 5 selected video cover the spectrum of characteristics' scale, from Low to High.

Figure 4.1: Facial landmarks marked on the frames of $Video_1$

4.3 User Study for Subjective Evaluation

In addition to objective evaluation of our approach, We perform subjective evaluation as quality of video is a subjective matter. In order to perform human validation of the outcome of our human facial video compression technique, we conducted a user survey. We showed the compressed videos in no specific order to 10 users and surveyed them to assess their subjective perception of the quality of each video. The original videos from the two benchmark databases and the 5 compressed videos corresponding to each database were presented to user. Each video was shown at 5 different threshold levels, 4 static and 1 dynamic threshold. Thus, each user was shown a total of 30 videos, 5 originals and 25 compressed videos.

Each user was asked to rate the compressed videos in terms of three key dimensions.

Figure 4.2: Facial landmarks marked on the frames of $Video_2$

Figure 4.3: Facial landmarks marked on the frames of $Video_3$

1. Perceived video quality: The quality of the videos were defined to the users, as their perception of resolution of the videos. This effectively captures the pixel quality of the video, in the perception of the viewers.

2. Smoothness of frame transition: This dimension captures the perception of smoothness (continuity) as a video moves from one frame to the next.

3. Perception of loss of information: This dimension captures whether the user feels any information has been lost.

Since, in our setting, we choose to selectively retain frames, finding the user perception regarding the video quality, smoothness of the transition of one frame transitioning to the next, and user perception of loss of information due to the video compression, all become important tasks.

The user study was conducted by first showing the original video to the users, and then at a random order show the different compressed videos, without conveying the threshold for any of the randomly-ordered videos. They were then asked to rate them on a Likert scale [18] of 1 to 5 along each of the three dimensions, with 5 being the highest (as good as the original video) and 1 being the least (much poorer than the original video). The results of our experiments are presented below, in Section **??**.

Figure 4.4: Facial landmarks marked on the frames of $Video_4$

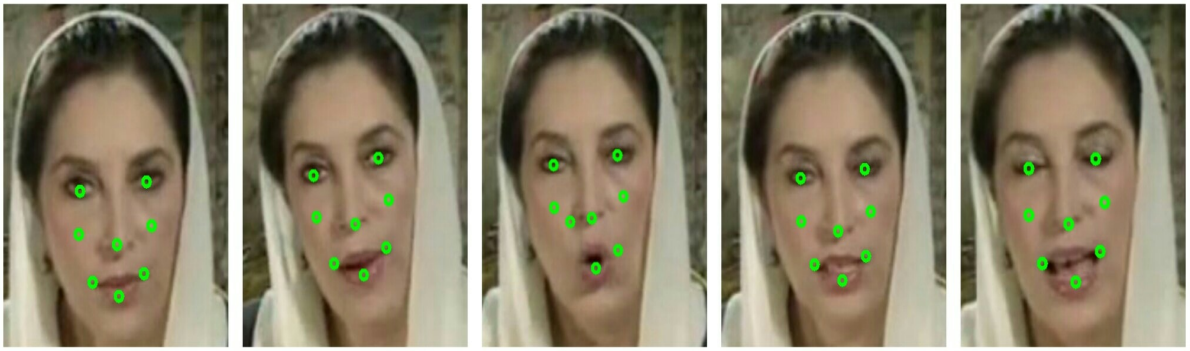


Figure 4.5: Facial landmarks marked on the frames of $Video_5$

Chapter 5

Results

In this chapter, we present the results of conducting our experiments. We also provide the perception about our system, of the users that participated in the study.

5.1 Impact of Static Thresholds

Our algorithm is designed to identify the distinct frames using the methodology outlined in Chapter 3, and compress the video by retaining the distinct frames and not retaining the others. This would inherently favor retaining the frames that have significant changes compared to the previous frames, specifically of the facial landmarks. For experimentation with real-life videos containing facial data, this translates to the fact that videos having a degree of movement of facial landmarks, such as while talking, while showing emotions (laughing, crying *etc.*) or while moving the entire face from one position to another, the expectation of having a higher number of distinct frames within a given duration of time is higher, compared videos with a lower degree of movement of facial landmarks.

Intuitively, the above phenomenon indicates that while for some videos, or for some stages of a given video, having a static threshold to identify a distinct video frame would suffice, it is not feasible to have a unique and ubiquitous static threshold that would yield a satisfactory performance for all video segments with varying degrees of movement of facial landmarks. In our user study, outlined in Tables 5.1, 5.2, 5.3, 5.4 and 5.5, we observe the above hypothesis to hold true. On all the axis of measurement that we conduct our user study on, namely the perceived video quality, the smoothness of tansition from one frame to the next, and the perceived information loss, conform with the above hypothesis. Further, as the static threshold increases, although the video compression ratio increases as well as the information content per unit length of the compressed video increases, the user experience quality consistently deteriorates along all the three axis.

Thus, experimenting with static threshold values, and the outcome of the user study, clearly reveals the need of a dynamically determined threshold, that would provide significant compression ratio to optimize storage and transmission, but would also cater a smooth and satisfactory user experience with minimally perceivable information loss, by automatically adopting to the rate of movement of facial landmarks across the different segments of a given video as well as across videos. The design of our dynamic thresholding is motivated by the above observations, and is presented below.

Original Size (MB)	Threshold	Compressed Size (MB)	Quality (/5)	Trans-ition (/5)	Information Loss (/5)	Overall Rating (/15)	Information Content per unit length (Initial Value = 1.6843)
25.89	3	11.30	4.5	4.5	4.4	13.4	1.9566
	5	6.61	4.5	4.3	4	12.8	2.1290
	8	3.28	4.5	4	3.4	11.9	2.3669
	10	2.51	4.4	3.4	3.6	11.4	2.4579
	Dynamic	8.57	4.4	4.6	4.1	13.1	1.9042

Table 5.1: Results showing original size, compressed size of the videos, time taken by our algorithm to compress the videos, and the user ratings across each threshold values for $Video_1$

Original Size (MB)	Threshold	Compressed Size (MB)	Quality (/5)	Trans-ition (/5)	Information Loss (/5)	Overall Rating (/15)	Information Content per unit length (Initial Value = 0.95675)
25.25	3	12.29	4.6	4.2	4	12.8	1.2806
	5	3.66	4.6	4.1	3.8	12.5	1.5324
	8	2.89	4.5	3.4	3.3	11.2	1.7998
	10	1.13	4.4	3.5	3.2	11.1	2.2142
	Dynamic	4.75	4.5	4.2	4.2	12.9	1.7018

Table 5.2: Results showing original size, compressed size of the videos, and the user ratings across each threshold values for $Video_2$

Original Size (MB)	Threshold	Compressed Size (MB)	Quality (/5)	Trans-ition (/5)	Information Loss (/5)	Overall Rating (/15)	Information Content per unit length (Initial Value = 1.4089)
20.14	3	13.59	4.4	4.2	3.9	12.5	1.4909
	5	6.61	4.5	4.2	3.8	12.5	1.6607
	8	4.55	4.5	4.1	3.8	12.4	1.6985
	10	3.99	4.5	3.8	3.7	12	1.7033
	Dynamic	4.59	4.5	4.4	4.1	13	1.8007

Table 5.3: Results showing original size, compressed size of the videos, and the user ratings across each threshold values $Video_3$

Original Size (MB)	**Threshold**	**Compressed Size (MB)**	**Quality (/5)**	**Trans-ition (/5)**	**Information Loss (/5)**	**Overall Rating (/15)**	**Information Content per unit length (Initial Value = 0.97167)**
2.85	3	2.35	4.3	4.5	4.4	13.2	1.1123
	5	1.32	4.4	4.4	4.3	13.1	1.2473
	8	0.815	4.4	3.8	3.6	11.8	1.3640
	10	0.597	4.4	4.1	3.3	11.8	1.4706
	Dynamic	0.677	4.5	4	3.5	12	1.4314

Table 5.4: Results showing original size, compressed size of the videos, and the user ratings across each threshold values for $Video_4$

Original Size (MB)	**Threshold**	**Compressed Size (MB)**	**Quality (/5)**	**Trans-ition (/5)**	**Information Loss (/5)**	**Overall Rating (/15)**	**Information Content per unit length (Initial Value = 0.93228)**
21.20	3	17.10	4.3	3.5	4	11.8	0.99685
	5	14.13	4.4	3	3.8	11.2	1.0387
	8	5.02	4.4	3.1	3.5	11	1.4048
	10	4.06	4.1	2.6	3.3	10	1.502
	Dynamic	3.58	4.4	4.4	4	12.8	1.4840

Table 5.5: Results showing original size, compressed size of the videos and the user ratings across each threshold values for $Video_5$

Overall User Rating						**Mean User Rating**	**Std. Dev. of User Ratings**
Threshold	$Video_1$ (/15)	$Video_2$ (/15)	$Video_3$ (/15)	$Video_4$ (/15)	$Video_5$ (/15)	Mean of User Ratings	Standard Deviation of User Ratings
3	13.4	12.8	12.5	13.2	11.8	12.74	0.6309
5	12.8	12.5	12.5	13.1	11.2	12.42	0.7259
8	11.9	11.2	12.4	11.8	11	11.66	0.5639
10	11.4	11.1	12	11.8	10	11.26	0.7861
Dynamic	13.1	12.9	13	12	12.8	12.76	0.4394

Table 5.6: Overall User Rating for all the Databases

5.2 Impact of Dynamic Threshold

The dynamic threshold based facial video compression technique, the methodology of which has been outlined earlier in Section 3, is geared towards providing a varying degree of compression of different segments of the video, that would auto-adopt to the varying rate of movement of facial landmarks within any given video. Practically, this would lead to different compression ratios within different segments of a given video, that in turn would be proportionate to the degree of movement of the facial landmarks. In other words, the segments where the information content per unit time is higher within a given duration covering a portion of the video, will have a lower degree of compression, compared to the same video at other portions where the information content per unit time has a lower value.

The user study indicates multiple interesting observations. An inspection of the compression ratio obtained for each of the videos, reveals a significantly good performance. For all the videos, the compression ratio we achieve with our dynamic thresholding method, is higher compared to the lowest static threshold that we experimented with. Specifically, we observe the following.

- In two out of the five videos, namely $Video_1$ and $Video_2$, the compression ratio with the dynamic threshold is higher than the one with static threshold 3, but lower than the remaining static thresholds.
- In case of $Video_3$, the compression ratio with the dynamic threshold outperforms the ones with static thresholds 3 and 5.
- For $Video_4$, it outperforms all but the video with a static threshold 10 - the maximum static threshold that we conduct our experiments with.
- And finally, for $Video_5$, the compression pario with the dynamic threshold outperforms all the videos compressed with static thresholds.

The user ratings, that capture the user perception of all the videos along all the three axis, namely vidoe quality, transition smoothness, and perceived loss of information, indicate that the dynamic threshold based compression consistently caters a quality that is comparable to the statically thresholded videos. We observe the following ranking characteristics.

- In one case, namely in $Video_1$, the dynamic threshold based video ranks second after the static threshold video, where the threshold value is set to 3.
- In another case, namely in $Video_4$ it ranks third, after the static threshold videos with thresholds set at 3 and 5.
- In all other cases, namely $Video_2$, $Video_3$ and $Video_5$, it is perceived to be the best (ranked the highest) for all the other videos.

The user perception scores along the three individual factors, along with the total scores, are presented in Tables 5.1, 5.2, 5.3, 5.4 and 5.5. Thus, our experiments indicate that the dynamically thresholded videos tend to outperform a significant number of the statically thresholded videos in terms of compression, as well as user experience.

Overall Compression Ratio (C.R.)						Mean of C.R.	Std. Dev. of C.R.
Threshold	$Video_1$ 25.89 MB	$Video_2$ 25.25 MB	$Video_3$ 20.14 MB	$Video_4$ 2.85 MB	$Video_5$ 21.20 MB	Mean of Compression Ratio	Standard Deviation of Compression Ratio
3	0.4365	0.4867	0.6748	0.8246	0.8066	0.6458	0.1787
5	0.2553	0.1449	0.3282	0.4632	0.6665	0.3716	0.2013
8	0.1267	0.1145	0.2259	0.2860	0.2368	0.1980	0.0743
10	0.0969	0.0448	0.1981	0.2095	0.1915	0.1481	0.0732
Dynamic	0.3310	0.1881	0.2279	0.2375	0.1689	0.2307	0.0627

Table 5.7: Overall Compression Ratio for all the Databases

	Dynamic Threshold			Static Threshold = 5		
Video Names	**Time Taken By Facial Landmark API per frame (ms/frame)**	**Time Taken By Our Algorithm per frame (ms/frame)**	**Time Taken By Our Algorithm per MB (s/MB)**	**Time Taken By Facial Landmark API per frame (ms/frame)**	**Time Taken By Our Algorithm per frame (ms/frame)**	**Time Taken By Our Algorithm per MB (s/MB)**
$Video_1$	26	288	**11.124**	25	225	**8.691**
$Video_2$	23.6	334.2	**10.059**	25	280.3	**8.436**
$Video_3$	25.6	410.3	**9.533**	29.9	322.6	**7.498**
$Video_4$	18.5	137.1	**12.982**	18.5	96.29	**9.123**
$Video_5$	24.1	181.95	**11.415**	22.5	153.4	**9.623**

Table 5.8: Time taken by the API and the Algorithm for static and dynamic thresholds

5.3 Comparison Based on Time taken

The time taken by our algorithm is in the order of a few hundred milli-seconds per compression frame, both for static thresholds (demonstrated with a threshold value of 5, for illustrative purposes) and dynamic compression thresholds, as shown on Table 5.8. Multiple interesting observations emerge.

- The compression time per frame varies significantly across videos. This can be attributed to the fact that the compression time taken per frame varies with the rate of movement of facial landmarks, which in turn, translates to higher information content per frame in our settings. Thus, the videos with higher information content per unit length (per frame, per MB *etc.*) take much longer to compress, compared to the ones with lower information content per unit length.
- The static threshold based system compresses faster compared to the dynamic threshold based system, across all the videos. The overhead of computing the dynamic thresholds requires the additional time. It is visibly obvious from Table 5.8 that the additional overhead of dynamic compression is consistently in the range of 20%-30% over its statc counterpart.
- A significant fraction of the time is invested in the actual compression process, while computing the landmarks does not provide too heavy an overhead.

Chapter 6

Conclusion

6.1 Discussion

Our experimental results establish the validity of our primary hypotheses, both in terms of achieving video compression as well as retaining perception of users about the given videos, in that: (a) facial video compression technique using the change of facial landmarks over successive video frames, is an effective solution, and (b) dynamic threshold based compression is often more effective compared to static threshold based videos. While it is not feasible to provide any quantitative metric to jointly capture the user satisfaction and video compression ratio for facial videos, we observe that, with dynamic thresholding, as a whole the user satisfaction is maximally or near-maximally retained, as well as the compression ratio is optimal or near-optimal, across several videos having different rates of movements of facial landmarks.

It is clear from Table 5.6 that, the average overall rating, which is a measure of our subjective rating, is the best with dynamically thresholded video. The compression size with dynamic thresholding is not the maximum for a few of the videos; however, in those cases, the user perception factor significantly tilt the weights in favor of the dynamic thresholding technique, over the several static threshold values. The standard deviation of the user ratings is the smallest for the dynamically thresholded video, when compared with all the statically thresholded videos.

Also, Table 5.7 shows that, while dynamically adopting to the difference in the rate of change of facial landmark movements prohibits the dynamically thresholded algorithm from having the highest compression ratio, it caters the most consistent compression ratio over all the videos. This is reflected by the fact that the standard deviation is the smallest in case of the dynamically thresholded video.

The combination of the above observations, demonstrate the consistent user satisfaction that our dynamic thresholding based methodology is capable of producing, over statically thresholded methods.

Facial landmark detection has been earlier used to solve the problems in facial expression analysis, head pose estimation, facial recognition. Video compressing can also be one such application which requires facial landmark detection as shown in this paper. Compressing human face videos by this algorithm paves way to more such related applications. Facial landmark positions identified can be used to replicate the same facial expressions on different cartoon characters or digital avatars on a live chat. With this algorithm, only the landmark positions can be exchanged between the end to end users and the digital avatars can make their faces according to the changed position of the landmarks. To further improve

the network bandwidth we can extend the idea of creating a geometrical head model of the user at the other end by using only the facial landmark points in a live chat. The first frame of the user can be sent to the other end along with the facial landmark points. After that only the information of the changed landmarks needs to be sent and the algorithm will automatically construct a geometrical model of the head of the user using the first frame and a mesh created from the landmark points.

6.2 Conclusion

In the current work, we proposed a methodology to compress videos that comprise of human faces. We presented a technique to compress videos based upon detected movements of facial landmarks across video frames. We explored two compression scenarios: one where the decision to compress is statically computed and a compression as carried out as soon as the change of any given facial landmark satisfies the static threshold; and another where the decision to compress is dynamically made and a compression automatically adjusts with the change rate of frames. We tested our methodology on smart mobile phones that have inherent resource limitations, thereby showing it to be practicable on a majority of modern-day devices. We benchmarked our system against two databases: Talking Face Video DB and YouTube Face DB, and obtained significant compression in both the cases. The goodness of our system was validated by a user study on 5 videos. The dynamic threshold based implementation was seen to deliver a more consistent performance compared to the static one, and often delivered the highest user satisfaction as well as high compression ratios. Our system can be used to compress real-life human face based videos, supplementing traditional video compression systems used in practice.

Bibliography

[1] ASCENSO, J., BRITES, C., AND PEREIRA, F. Improving frame interpolation with spatial motion smoothing for pixel domain distributed video coding. In *5th EURASIP Conference on Speech and Image Processing, Multimedia Communications and Services* (2005), Smolenice, Slovak Republic, pp. 1–6.

[2] BARTLETT, M. S., LITTLEWORT, G., FASEL, I., AND MOVELLAN, J. R. Real time face detection and facial expression recognition: Development and applications to human computer interaction. In *Computer Vision and Pattern Recognition Workshop, 2003. CVPRW'03. Conference on* (2003), vol. 5, IEEE, pp. 53–53.

[3] BAUCHSPIES, R. A. Temporal compression and decompression for video, Dec. 28 1999. US Patent 6,008,847.

[4] CHUN, K., LIM, K., CHO, H., AND RA, J. An adaptive perceptual quantization algorithm for video coding. *IEEE Transactions on Consumer Electronics 39*, 3 (1993), 555–558.

[5] COOTES, T. Talking face video database. Images. `https://www-prima.inrialpes.fr/FGnet/data/01-TalkingFace/talking_face.html`.

[6] EKMAN, P., AND FRIESEN, W. V. Measuring facial movement. *Environmental psychology and nonverbal behavior 1*, 1 (1976), 56–75.

[7] FACERIG, F. Facial animation system. `https://facerig.com/`.

[8] HEIZMANN, J., AND ZELINSKY, A. Robust real-time face tracking and gesture recognition. In *IJCAI* (1997), pp. 1525–1530.

[9] ITTI, L. Automatic foveation for video compression using a neurobiological model of visual attention. *Image Processing, IEEE Transactions on 13*, 10 (2004), 1304–1318.

[10] JACKSON, J., ET AL. Low-bit rate motion jpeg using differential encoding. In *Signals, Systems and Computers, 2004. Conference Record of the Thirty-Eighth Asilomar Conference on* (2004), vol. 2, IEEE, pp. 1723–1726.

[11] JO, G.-S., CHOI, I.-H., AND KIM, Y.-G. Robust facial expression recognition against illumination variation appeared in mobile environment. In *Computers, Networks, Systems and Industrial Engineering (CNSI), 2011 First ACIS/JNU International Conference on* (2011), IEEE, pp. 10–13.

[12] KIM, M., KUMAR, S., PAVLOVIC, V., AND ROWLEY, H. Face tracking and recognition with visual constraints in real-world videos. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on* (2008), IEEE, pp. 1–8.

[13] LANE, N. D., MILUZZO, E., LU, H., PEEBLES, D., CHOUDHURY, T., AND CAMPBELL, A. T. A survey of mobile phone sensing. *IEEE Communications magazine 48*, 9 (2010), 140–150.

[14] LANE, N. D., MOHAMMOD, M., LIN, M., YANG, X., LU, H., ALI, S., DORYAB, A., BERKE, E., CHOUDHURY, T., AND CAMPBELL, A. Bewell: A smartphone application to monitor, model and promote wellbeing. In *5th international ICST conference on pervasive computing technologies for healthcare* (2011), pp. 23–26.

[15] LE GALL, D. Mpeg: A video compression standard for multimedia applications. *Communications of the ACM 34*, 4 (1991), 46–58.

[16] LEE, K.-C., HO, J., YANG, M.-H., AND KRIEGMAN, D. Video-based face recognition using probabilistic appearance manifolds. In *Computer Vision and Pattern Recognition, 2003. Proceedings. 2003 IEEE Computer Society Conference on* (2003), vol. 1, IEEE, pp. I–313.

[17] LIEN, J. J.-J., KANADE, T., COHN, J. F., AND LI, C.-C. Detection, tracking, and classification of action units in facial expression. *Robotics and Autonomous Systems 31*, 3 (2000), 131–146.

[18] LIKERT, R. A technique for the measurement of attitudes. *Archives of psychology* (1932).

[19] LIOU, M. Overview of the p$\times$ 64 kbit/s video coding standard. *Communications of the ACM 34*, 4 (1991), 59–63.

[20] MARPE, D., WIEGAND, T., AND SULLIVAN, G. J. The h. 264/mpeg4 advanced video coding standard and its applications. *Communications Magazine, IEEE 44*, 8 (2006), 134–143.

[21] MOHAN, P., PADMANABHAN, V. N., AND RAMJEE, R. Nericell: rich monitoring of road and traffic conditions using mobile smartphones. In *Proceedings of the 6th ACM conference on Embedded network sensor systems* (2008), ACM, pp. 323–336.

[22] OTSUKA, T., AND OHYA, J. Spotting segments displaying facial expression from image sequences using hmm. In *Automatic Face and Gesture Recognition, 1998. Proceedings. Third IEEE International Conference on* (1998), IEEE, pp. 442–447.

[23] PALEARI, M., AND LISETTI, C. L. Toward multimodal fusion of affective cues. In *Proceedings of the 1st ACM international workshop on Human-centered multimedia* (2006), ACM, pp. 99–108.

[24] RICHARDSON, I. E. *H. 264 and MPEG-4 video compression: video coding for next-generation multimedia*. John Wiley & Sons, 2004.

[25] RIJKSE, K. H. 263: video coding for low-bit-rate communication. *Communications Magazine, IEEE 34*, 12 (1996), 42–45.

[26] SANTAMAKI, H., LEPPANEN, J., HAIKONEN, P., AND KORHONEN, I. Video compression method, Jan. 5 1988. US Patent 4,717,957.

[27] SETTON, E., YOO, T., ZHU, X., GOLDSMITH, A., AND GIROD, B. Cross-layer design of ad hoc networks for real-time video streaming. *IEEE Wireless Communications 12*, 4 (2005), 59–65.

[28] SIKORA, T. The mpeg-4 video standard verification model. *Circuits and Systems for Video Technology, IEEE Transactions on 7*, 1 (1997), 19–31.

[29] SPANOS, G. A., AND MAPLES, T. B. Performance study of a selective encryption scheme for the security of networked, real-time video. In *Computer Communications and Networks, 1995. Proceedings., Fourth International Conference on* (1995), IEEE, pp. 2–10.

[30] SUK, M., AND PRABHAKARAN, B. Real-time mobile facial expression recognition system–a case study. In *Computer Vision and Pattern Recognition Workshops (CVPRW), 2014 IEEE Conference on* (2014), IEEE, pp. 132–137.

[31] SULLIVAN, G. J., TOPIWALA, P. N., AND LUTHRA, A. The h. 264/avc advanced video coding standard: Overview and introduction to the fidelity range extensions. In *Optical Science and Technology, the SPIE 49th Annual Meeting* (2004), International Society for Optics and Photonics, pp. 454–474.

[32] TIAN, Y.-L., KANADE, T., AND COHN, J. F. Recognizing action units for facial expression analysis. *Pattern Analysis and Machine Intelligence, IEEE Transactions on 23*, 2 (2001), 97–115.

[33] TUDOR, P. Mpeg-2 video compression. *Electronics & communication engineering journal 7*, 6 (1995), 257–264.

[34] WALLACE, G. K. The jpeg still picture compression standard. *IEEE transactions on consumer electronics 38*, 1 (1992), xviii–xxxiv.

[35] WANG, J., AND COHEN, M. F. Very low frame-rate video streaming for face-to-face teleconference. In *Data Compression Conference* (2005), IEEE, pp. 309–318.

[36] WOLF, L., HASSNER, T., AND MAOZ, I. Face recognition in unconstrained videos with matched background similarity. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on* (2011), IEEE, pp. 529–534.

[37] YEASIN, M., BULLOT, B., AND SHARMA, R. Recognition of facial expressions and measurement of levels of interest from video. *IEEE Transactions on Multimedia 8*, 3 (2006), 500–508.

[38] ZHANG, Z.-L., WANG, Y., DU, D. H., AND SHU, D. Video staging: a proxy-server-based approach to end-to-end video delivery over wide-area networks. *IEEE/ACM Transactions on Networking (TON) 8*, 4 (2000), 429–442.