Adaptive Threshold Based Video Compression Technique

Prof. Meenakshi Sharma
HOD CSE Department, SSCET,
Badhani, India

Mukhinder Singh
M.Tech CSE Department, SSCET,
Badhani, India

Abstract—Edge detection and thresholding were used to locate edge areas in both the original and degraded video sequences. Degradation in the edge areas was calculated by measuring the peak signal-to-noise ratio (PSNR) between the edge areas of the original and degraded video clips. A common problem of global thresholding is that in practice it is impossible to find a single global threshold that works with all kinds of video materials. Therefore, global threshold should be avoided. Instead, an adaptive threshold can be a better alternative to enhance the detection precision. It uses the local thresholds of the feature or similarity function to be compared. Still now there are lower quality ratings to video clips with noticeably degraded edge areas even though the overall mean squared error (MSE) is not large. The experimental results show that the novel scheme has the better PSNR result as compare to the existing techniques.

Keywords— Macro block, Motion compensation, Threshold, Adaptive threshold, Psnr

I. INTRODUCTION

The fundamental aim of video compression is to minimize redundancy of the multimedia data in order to be able to store or transmit data in an efficient form. In most images or frames the neighbouring pixels are correlated and therefore contain redundant information. Three fundamental components of compression are Coding Redundancy, Spatial Redundancy and Temporal Redundancy, Irrelevant Information. There are two ways of classifying compression techniques are Lossless vs. Lossy compression and Predictive vs. Transform coding.

Quality metrics have an important application in video compression algorithms which make compression decisions in order to achieve optimal trade-off between picture quality and data rate. Hence, a strong correlation of these metrics with perceived quality is imperative. The subjective measurement of mean opinion score (MOS) is considered to be an accurate way to determine the perceived video quality. The general approach of MSE to choose the best coding option, MSE is a mathematical error measure which does not consider the human visual system and is therefore not an accurate measure of perceived quality. The quality of image is degraded by various noises in its acquisition and transmission. Image De noising has remained a fundamental problem in the field of image processing. There is various noise reduction techniques used for removing noise. Most of the standard algorithms use to de noise the noisy image and perform the individual filtering process which reduces the noise level. But the image is either blurred or over smoothed due to the loss of edges. Noise reduction is used to remove the noise without losing detail contained in the images.

In this paper we are proposing a technique by which we can separate the stationary and moving objects in real time so as to result in a lossless video compression. Now days the techniques which are being used for video compression are all lossy compression type unlike ours “Object repetition based video compression”. In this paper we present an object repetition based video coding approach that retains the relative advantages of both the hybrid based and block-based coders while minimizing the drawbacks of both. By employing motion segmentation techniques to separate moving objects from stationary backgrounds, the coder optimizes the bit allocation to those areas that are changing most frequently. This technique supports content-based functionalities such as object scalability and object manipulation easily.

In this paper we present an object repetition based video coding approach. In section II we show Motion estimation of video compression. Section III represents the threshold. In section IV describe about the proposed method and methodology.

II. MOTION COMPENSATION

The fundamental idea behind the block matching is to divide the current frame into a matrix of ‘macro blocks’ that are then compared with corresponding block and its adjacent neighbours in the previous frame to create a vector that stimulates the movement of a macro block from one location to another in the previous frame.

A. Motion estimation and motion vector

The motion estimation process is most computationally expensive and resource hungry operation in the entire compression. Since motion estimation is a process to match similar areas between frames, much information related to frame content correlation and object motion are already available from the motion estimation process. In most practical applications, videos are processed and stored in the compressed domain where motion estimation is performed during the
compression process to remove the temporal redundancy. It should be noted that the first frame is always sent full, and so are some other frames that might occur at some regular interval. A motion compensated image for the current frame is then created that is built of blocks of image from the previous frame. There are numbers of researches which use the motion- based or motion-related features for video processing.

The motion vectors for blocks used for motion estimation are transmitted, as well as the difference of the compensated image with the current frame is also JPEG encoded and sent. The compressed video provides the MV information which can be directly extracted from the bit stream. The encoded image that is sent is then decoded at the encoder and used as a reference frame for the subsequent frames. The decoder reverses the process and creates a full frame.

B. Motion Compensation

A motion compensated image for the current frame is then created that is built of blocks of image from the previous frame. The compensation of a video frame can be done by predicting the current frame from a reference frame. The frame is called to be a reference if it can be used for motion compensation prediction. The straightforward technique to do motion compensation is to use the previous frame to predict the current frame, and find the difference between the actual current frame and its prediction to make sure that the obtained prediction is true. The simplest algorithm is to subtract the previous frame from the current frame and encode that difference. This algorithm uses the entire previous frame as the prediction of the current frame. The prediction value of the pixels at location i, j can be expressed mathematically as

\[ f_d(i, j) = f_{n-1}(i, j) \]

Where \( f_d(i, j) \) is the pixel in frame n at position i, j and \( f_{n-1}(i, j) \) is the pixel at position i, j in frame n-1 and “ \( f_d(i, j) \) is the prediction value of the pixel at position i, j in frame note effects of the algorithm. The prediction of the current frame and the compression of the difference between the predicted frame and the actual frame are called hybrid coding.

Block-Based Motion-Compensated Prediction (MCP) relates to estimate frames in local motion the neighbour frame is divided into rectangular pixel blocks. In MCP the comparison is done between the current frame and the next frame, the comparison is done block by block. When compressing or sending the video frames from terminal to terminal, the next frame will not be sent, instead, a collection of motion vectors (MV) will be sent. The comparison is done by either the sum of absolute differences (SAD) or the sum of squared differences (SSD). There are mainly two methods Fixed Size Block-Matching (FSBM) and Variable Size Block-Matching (VSBM), for choosing the size of the block in the block based MCP.

When there is no motion in parts of the image, VM will hold the value \( V = (0, 0) \). However if there are other parts of the image that has motion, MV will hold non-zero values. Block based motion compensation have some artifacts such as blocking effect, Mosquito effect, ME Mismatches effect.

III. THRESHOLD

Segmentation involves separating an image into regions (or their contours) corresponding to objects. We usually try to segment regions by identifying common properties. Or, similarly, we identify contours by identifying differences between regions (edges). The simplest property that pixels in a region can share is intensity. So, a natural way to segment such regions is through thresholding, the separation of light and dark regions. Thresholding creates binary images from grey-level ones by turning all pixels below some threshold to zero and all pixels about that threshold to one. (What you want to do with pixels at the threshold doesn’t matter, as long as you’re consistent.) If \( g(x, y) \) is a threshold version of \( f(x, y) \) at some global threshold \( T_g \) is equal to 1 if \( f(x,y) \geq T \) and zero otherwise.

Methods of thresholding

A. Basic thresholding

Basic thresholding is done by visiting each pixel site in the image, and set the pixel to maximum value if its value is above or equal to a given threshold value and to the minimum value if the threshold value is below the pixels value. Basic thresholding is often used as a step in other thresholding algorithms.

B. Band thresholding

Band thresholding is similar to basic thresholding, but has two threshold values, and set the pixel site to maximum value if the pixels intensity value is between or at the threshold values, else it set to minimum.

C. P-tile thresholding

P-tile is a method for choosing the threshold value to input to the “basic thresholding” algorithm. P-tile means “Percentile”, and the threshold is chosen to be the intensity value where the cumulative sum of pixel intensities is closest to the percentile.

D. Optimal thresholding

Optimal thresholding selects a threshold value that is statistically optimal, based on the contents of the image.
E. Adaptive thresholding

Adaptive thresholding divides the image into patches, and each patch is threshold by a threshold value that depends on the patch contents (the threshold adapts). The threshold value of a patch is chosen to be a weighted sum of the mean intensity value of the patch and a global threshold value. The global threshold value is chosen to be the optimal threshold. The basic thresholding works well, and solves the problem of segmentation entirely. When using basic threshold, a threshold value has to be selected somehow. This was done manually. Interestingly, the method of optimal thresholding selected a better threshold value than what was found manually. Band thresholding is not meaningful to do for some images, because the histogram does not need to be divided into more than two to do correct thresholding. Adaptive thresholding manages to get the borders of the shapes slightly more correct, but also produces a little more junk.

IV. PROPOSED METHOD AND METHODOLOGY

A. Methodology/Planning of work

![Fig.1 Methodology](image)

Start with grabbing the video sequences through use of image acquisition. After grabbing of video sequences predict and extract the still frame in RGB format. In next step, loading the upcoming or next frame in MATLAB. Now, Compare these two frames and calculate the differences of these two frames and remove noise in the frame. Besides this, One apply the pre-processing filtering for eliminate the noise in the frame.

Next, Adaptive thresholding technique is applied and it can be a better alternative to enhance the detection precision. Adaptive thresholding divides the image into patches, and each patch is threshold by a threshold value that depends on the patch contents. The threshold value of a patch is chosen to be a weighted sum of the mean intensity value of the patch and a global threshold value. The global threshold value is chosen to be the optimal threshold. The basic thresholding works well, and solves the problem of segmentation entirely.
Adaptive thresholding manages to get the borders of the shapes slightly more correct. After prediction of adaptive threshold, Conclude objects are being saved in the memory and all of these followed steps are be used for upcoming or next frame in the video. If all these frames are over in the video then check compressed length of the video or the frames otherwise loading the next frame in the matlab. Now, predicting the bit error rate in the frames or video and finally get psnr results.

B. Methodology steps
1) Loading of data set in matlab environment using video acquisition toolbox.
2) Pre-processing steps for all acquired frames.
3) Adaptive thresholding for detecting moving parts for successive frames.
4) Storing unique features for compressing video frames.
5) Concluding various parameters to plot the percentage of improved results.

Now proposed technique is going to performing operation on “akiyo” video sequence and gets the efficient results. The Proposed method getting better PSNR and MSE results on different bit rates. Table 1 Shows that the new scheme provides the better mean psnr results at the very low bit rate as compare to the other methods.

<table>
<thead>
<tr>
<th>Bit Rate(kbps)</th>
<th>Original frame size(In bits)</th>
<th>Compressed Size(In bits)</th>
<th>MEAN PSNR</th>
</tr>
</thead>
<tbody>
<tr>
<td>35 kbps</td>
<td>53526528</td>
<td>5010920</td>
<td>56.91</td>
</tr>
<tr>
<td>48 kbps</td>
<td>40144896</td>
<td>4639960</td>
<td>54.98</td>
</tr>
<tr>
<td>50 kbps</td>
<td>35684352</td>
<td>4526560</td>
<td>54.87</td>
</tr>
<tr>
<td>60 kbps</td>
<td>31223808</td>
<td>4421880</td>
<td>53.67</td>
</tr>
<tr>
<td>80 kbps</td>
<td>22302720</td>
<td>4175120</td>
<td>51.66</td>
</tr>
<tr>
<td>100 kbps</td>
<td>17842176</td>
<td>4059400</td>
<td>50.75</td>
</tr>
<tr>
<td>150 kbps</td>
<td>13381632</td>
<td>3943960</td>
<td>48.69</td>
</tr>
<tr>
<td>200 kbps</td>
<td>8921088</td>
<td>3829200</td>
<td>45.6</td>
</tr>
<tr>
<td>250 kbps</td>
<td>8921088</td>
<td>3829080</td>
<td>45.82</td>
</tr>
<tr>
<td>300 kbps</td>
<td>8921088</td>
<td>3832480</td>
<td>46.1</td>
</tr>
<tr>
<td>350 kbps</td>
<td>8921088</td>
<td>3828680</td>
<td>46.1</td>
</tr>
<tr>
<td>400 kbps</td>
<td>8921088</td>
<td>3832680</td>
<td>46.1</td>
</tr>
</tbody>
</table>

![Graph showing PSNR and Bit Rates](image_url)

Fig. 2 Plot in between psnr and bit rates for “akiyo” video sequences of proposed method

![Two frames of “akiyo” video sequence](image_url)

Fig.3 (a) & (b) shows the two frame of the “akiyo” video sequence
In proposed method results are taken for akiyo video sequence on different bit rates. The novel method provides very efficient results at very low bit rates. Here figure shows the last two frames captured at the 35 bit rates of the akiyo video sequences.

![Frame Images]

Fig. 4 (a) Shows the edge detector or noise in the first two frames (b) Static image

V. CONCLUSION

The compression process is completely lossless means that the output from the decompressor is bit-for-bit identical with the original input to the compressor and compression ratio is high. A common problem of global threshold is that in practice it is impossible to find a single global threshold that works with all kinds of video materials. Therefore, global threshold should be avoided. Instead, an adaptive threshold can be a better alternative to enhance the detection precision. An adaptive threshold or novel technique provides the better psnr results as compare to other techniques. With the experiment results it was observed that the Compression Ratio increases with the increase of number of frames, thus, we can say that CR is directly proportional to the number of frames. The application of the novel technique in segmenting the Stationary and non stationary objects in the frames, Calculate the motion estimation in the frames. In our novel method we have used adaptively thresholding techniques which have given better result as compare to the global threshold. In future we wish to increase upon the speed and compression ratio.

REFERENCES