CHAPTER 4

BLOCKING ARTIFACT REDUCTION

4.1 INTRODUCTION

Blocking artifacts are very annoying and degrade the visual quality of reconstructed images after compression-decompression process. In this chapter a new post processing strategy has been presented based on Lagrange’s interpolation.

Video coding remains the most widely used method in transmission and storage of video in the rapidly developing world on internet and computing technologies. Due to limitations of transmission bandwidth and storage devices we are moving to low-bit compression. Hence we require very efficient data compression methods to fit a large amount of video data into a narrow bandwidth of communication channels or small storage devices while acceptable quality of reconstructed data is preserved. This is, however, challenging for most existing coding algorithms and standards, since highly visible artifacts appear when the coding bit rate is low.

Visual artifacts in digital image and video are due to various factors. The major factors that lead to visually annoying artifacts are:

- Due to inadequate acquisition of video, which is caused by out of focus of the camera, smearing, dust, scratch, blotches in the visual path of video capturing.
- Due to inadequate processing of signal while conversion from analog to digital and other purposes.
- Due to transmission loses: Like while transferring data in a network, there may be loss of blocks, frames, discoloring due to congestion and other network problems, and
- Watermarking of the images.

Generally at low bit-rate encoding, blocking artifacts are visibly more perceptible and annoying among all kinds of artifacts due to
compression of image data. In this chapter a new post-processing algorithm is proposed to reduce such a blocking artifact. The proposed algorithm is based on image enhancement using interpolation, in the DCT domain. A distortion that appears in compressed video material has abnormally large pixel blocks. Also called "macroblocking," it occurs when the encoder cannot keep up with the allocated bandwidth. It is especially visible with fast motion sequences or quick scene changes. Video uses lossy compression, and higher the compression rate, the more content is removed. At decompression, the output of certain decoded blocks makes surrounding pixels appear averaged together and look like larger blocks. As the size of output devices get larger, blocking and other artifacts become more noticeable.

### 4.2 STATE OF ART ON ARTIFACTS

Video coding technique based on the block coding is the most popular and has found many applications. Especially, block coding is used in both still image and moving image coding standards, such as JPEG [Wall92] and MPEG [Gall91]. Other block coding that involves partitioning such as vector quantization [Gers82], block truncation coding [Delp79] and fractal-based compression [Barn88] are found in the literature.

But the main drawback of the block coding is the blocking artifacts [Reev84]. This is due to the encoding of blocks without considering the correlation between adjacent blocks. In transform coding the high quantization of transform coefficients leads to blocking effect [Stev93], but it becomes a major problem in block truncation coding and vector quantization. Blocking artifact reduction is due to the attenuation of high frequency terms during compression, which can turn the slowly varying regions in to a series of visible step changes. To reduce blocking artifacts in the decode images two approaches are adopted:

- One is to solve the problem at the encoder end, which is known as pre-processing technique.
- The other uses a post processing at the decoder side.
4.2.1. Pre-Processing

Pre-processing techniques are used to enhance the quality of reconstructed images for a given bit rate constant. Pre-processing is commonly utilized in video compression systems, though it is not widely discussed in the literature. The goal of a pre-processing algorithm is to remove noise and small features from a video sequence, while preserving salient information. This affects the visual quality of the decoded sequence in two ways. First, the noise component of the original imagery is not transmitted to the decoder. Second, bits previously allocated to the noise are reassigned to more critical image areas. Pre-processing algorithms appear at different locations within a video compression system. Perhaps the most obvious is to position the pre-processing algorithm before the encoding procedure. In this location, the pre-processor operates directly on the intensity data of the original image sequence. In H.261, the prediction is processed with a low-pass filter before calculating the error residual. The operation is called loop filtering. In H.263+, the decoder filters the block boundaries of the intensity data, with the amount of smoothing varying relative to the quantization parameter. Another approach is to allow blocks to overlap so that the adjacent ones can be dependent to some degree [Reev84].

4.2.2 Post Processing

The post processing [Meie99, Shen98] algorithms aiming at the reduction of compression artifacts improves the overall perceptual quality for a given bit rate or equivalently, increases the compression ratio with respect to a given quality requirement.

The post-processing algorithms are derived from two different viewpoints: i) Image restoration and ii) Image enhancement.

4.2.2.1 Image Restoration

In restoration methods the images are recovered and must satisfy some measurement of “goodness” which can be accomplished with either deterministic or stochastic models. In deterministic model, the algorithm reduces compression artifacts by finding a solution that remains faithful to the decoded result while also satisfying a definition of smoothness.
These definitions can include linear models within a Lagrangian approach or non-linear models within a projection onto convex sets (POCS) methodology [Yong97]. In the Lagrangian techniques, attempt is made to minimize the function,

\[ j(f) = \|f - g\|^2 + \lambda \| Cf \|^2 \]  

(4.1)

where, \( f \) denotes the post-processed image, stored row by row in a one-dimensional vector, \( g \) denotes the observation provided by the compressed bit-stream, \( C \) is an operator (or filter) that defines the measurement of goodness, and \( \lambda \) is a multiplier that controls the influence of the regularizing constraint. In most realizations, \( C \) is a high-pass filter that enforces a smoothness constraint at the block boundaries.

In some applications, it may be more convenient to express the ideal properties of the post-processed image as a combination of closed and convex sets of solutions. For example, instead of penalizing high-frequency content, we could say that the energy across the block boundaries is less than some threshold. If all of the ideal properties describe the original image sequence, a solution is guaranteed to exist within the union of the set of constraints. To find the solution, a projection operator is first defined for each set. These operators map solutions outside of the constraint into the allowable set of solutions. Then, the projection operators are applied with the iteration.

\[ f^{k+1} = P_1 P_2 \ldots P_{n-1} P_n f^k \]  

(4.2)

where \( P_i \) is the projection operator for the \( i \)th set. The algorithm continues until \( f^{k+1} \) and \( f^k \) are very close, which denotes that the solution is within the intersection of the sets.

In stochastic methods, the post-processed solution corresponds to the maximum a posteriori (MAP) estimate of the image sequence provided to the encoder. Thus, after applying Bayes’ rule, the post processed image must satisfy,
\[
f = \arg \min \log p(f_{\text{allowable}})
\]  

where, \(f_{\text{allowable}}\) denotes the set of images represented by the compressed result and \(p\) is a probability distribution. In most applications, the distributions used for post-processing penalize images with significant high-frequency information, such as the expression.

\[
p(f) = \frac{1}{Z} \exp \left\{ -\frac{1}{\beta} \sum_c \varphi(d_c(f)) \right\},
\]  

where \(Z\) is a normalizing constant, \(\beta\) is a non-negative number, \(d_c\) returns the simple difference between the pixel of interest and the neighbor in the \(c\)th direction and \(\varphi\) is a potential function that returns large numbers for large differences. Since the restoration algorithms usually take much computation time, the filtering technique is preferred for real-time implementation [Lin97, Karu95].

### 4.2.2.2 Image Enhancement

Designing an image enhancement algorithm typically relies on a two-step procedure. First, objectionable distortions are identified and modeled within the decompressed video sequence. Then, an operator is constructed to attenuate the distortion. For low-bit applications, enhancement techniques primarily remove blocking artifacts. For example, filtering the decoded image with a low-pass filter attenuates blocking.

The technique by H. C. Reeves et.al [Reev84] suggests using a 3×3 Gaussian spatial domain filter. This approach is very fast; however, it cannot reduce artifacts that are not confined to pixels next to block boundaries. In particular, it has difficulties in suppressing ringing that extends over a larger area. Kim et.al [Kim99] propose a separable adaptive two-dimensional filter that progressively transforms from a traditional median filter within blocks to a low-pass filter when it gets closer to the block boundaries. Kim, Jang and Hong proposed a blocking artifact reduction algorithm using wavelet transform as a set of one-dimensional signals reducing the step-wise discontinuities in the spatial domain.
But, excessive smoothing also removes important high-frequency content. Several techniques adjust the amount of smoothing to address this problem. Filtering may be restricted to the block boundaries and reduced when significant intensity differences are detected between neighboring blocks [Kim96] and this preserves significant region boundaries. Alternatively, the visibility of each block boundary may be estimated through the use of a human visual system model and boundary is then smoothed until it is no longer visible as proposed in [Karu95]. Adaptive filtering [Yan94] for video signals is applied to both spatial and temporal components iteratively to reduce blocking artifacts. An important application of spatial filtering techniques is in the post processing of images degraded by coding.

Linear, space-invariant filters are inadequate to reduce the noise produced by block coders [Rama86]. With block-based compression approaches for both still images and sequences of images annoying blocking artifacts are exhibited, primarily at high compression ratios. They are due to the independent processing (quantization) of the block-transformed values of the intensity or the displaced frame difference. M. Javier et.al [Javi00] proposed an application of the hierarchical Bayesian paradigm to the reconstruction of block discrete cosine transform (BDCT) compressed images and the estimation of the required parameters. The reconstructed images from highly compressed MPEG data have noticeable image degradations, such as blocking artifacts near the block boundaries, corner outliers at cross points of blocks, and ringing noise near image edges because the MPEG quantized the transformed coefficients of 8 X 8 pixel blocks. A post-processing algorithm was presented in [Hyun99] to reduce quantization effects, such as blocking artifacts, corner outliers, and ringing noise, in MPEG-decompressed images. Motivated by error concealment applications N. Aria [Aria02] proposed a method for the post-processing of JPEG-2000 compressed images at very low bit rates. This method counter intuitively employs further compression to achieve image enhancement. This approach, although not widely known, is not entirely new; it is an adaptation of a technique originally designed for the removal of block-transform coding artifacts.

Nakajima, Hori and Kanok proposed a pel adaptive image restoration algorithm for coded video signals, where ringing effects are removed without blurring edges at the decoder. Kuo and Hsieh proposed
an adaptive postprocessor for the removal of blocking artifact, where the adaptation is achieved by changing filter coefficients according to the local characteristics of images and the blocking effect. Chen, Ren Wu and Qiu proposed a post processing approach that works in the transform domain to alleviate the accuracy loss of transform coefficients, which is introduced by the quantization process. Huang Jiwu proposed an adaptive space-variant filter based edge map for reducing block effects by analyzing the environment of each pixel and based on that 1-D directional filter and 2-D mean filter are applied on those pixels. These adaptive methods reduce the computational complexity of ordinary filters and increase the adaptation to local statistics of images.

However these methods reduce the blocking artifacts considerably, when they are applied for video coding. At the decoder they are not viable because of the high computational complexity [Atzo00] relative to video decoder. Since already the decoding of video compressed data remains a huge task and if additionally these algorithms are applied the efficiency of the video decoder reduces considerably. The computational complexity of these algorithms is mainly due to: 1) pixel or block classification - as the artifact are mainly due to boundary pixels between two successive blocks, rather this process is applied to all the pixels in the block and 2) the adaptively of filters is not robust.

In this chapter we propose a new post-processing algorithm, which exploits the statistical properties of pixels in the boundary of the blocks and classifies block boundary into flat, texture and edge regions using pixel difference in the block boundary. Based on this classification, we adaptively interpolate the pixels. The proposed method is developed considering the subjective quality of images, signal-to-noise ratio (SNR) and computational complexity. Since we implemented a simple procedure for block classification and adaptive filtering of pixels, the computational complexity is considerably reduced along with error.

Our proposed algorithm is compared with three post processing methods Modular Post-processing Scheme (MPS) [Kim98], Reducing Quantization Effect (RQE) method (BPM) [Park99] and Low Pass Filter method [Nam00] for comparative study. In the subsequence subsection, we describe these schemes.
4.2.3 Modular Post-Processing Scheme (MPP)

In this method a modular post-processing algorithm appropriate for conventional block-based video coding is developed. The modular feature of the proposed filter makes it possible to adjust the overall complexity depending on the available computational power of the decoder. The proposed algorithm is aimed to improve both subjective and objective image quality prominently. This algorithm, limits the pixels be smoothed to two block boundary pixels, v4 and v5. In the figure 4.1 we use the 4-point DCT kernel as a frequency analysis tool to get the feature of the pixel array. As shown in figure 4.1 if a 4-point pixel array is located across the block boundary. We define a01, a11, a21 and a31 as the 4-point DCT coefficients of S1, and then the high frequency component a31 is a major factor affecting the blocking artifact. Thus, in this mode the magnitude of a31 is scaled down using a factor as,

$$\text{MIN} \left( \left| a_{3,0} \right|, \left| a_{3,1} \right|, \left| a_{3,2} \right| \right) / \left| a_{3,1} \right|$$  \hspace{1cm} (4.5)

And its value is between 0 and 1. By doing this, the difference between block boundaries is lowered in smooth region and is not affected in complex regions so that undesirable blurring can be prevented. If the magnitude a31 is greater than a certain value however the filter is not applied to preserve the image details.

![Figure 4.1 Typical 8x8 block boundaries](image)

This algorithm is tested on a various sample and the quality of the output image is reported. Here the images i) Girl (256 x 256) and ii) Flower
Garden (256 x 256) are selected and they are subjected to the artifact reduction by this algorithm.

Figure 4.2
4.2.4 Reducing Quantization Effect (RQE) METHOD (BPM)

This method deals with removal of the blocking artifacts effect in images at low bit rate compression. This algorithm is proposed to reduce the blocking artifacts in a low bit-rate image keeping the algorithm complexity to minimum. It is based on:

1. Smoothing artificial discontinuities between block boundaries.
2. Smoothing actual image edges degrades the image quality. So smoothing needs to be done.

The algorithm comprises of edge detection logic to differentiate between false edges and real edges and applying smoothing filter to modify the pixel values in the false edge and real edge. The edge detection is done to prevent the filtering of real edges and the loss of sharpness in the picture content. Edge differentiation is done by simple threshold.

As shown in the figure 4.3, four pixels on the top vertical and left horizontal edges of each block are filtered. A, B, C and D represent the pixel values the block boundary after decoding. The A, B, C and D represent the pixel values after applying the de-blocking on the pixel A, B, C and D. The fig 4.3a depicts the filtering mechanism in detail. As shown in the figure the four pixels at boundary are modified. As seen the algorithm requires very simple control of mechanism for applying filtering in comparison to other known algorithms. One-dimensional view of the block boundary after deblocking is shown in the fig 4.3b.

![Figure 4.3a One-dimensional view of the block boundary after deblocking](image-url)

```plaintext
Figure 4.3a One-dimensional view of the block boundary after deblocking
```
Figure 4.3b One-dimensional view of the block boundary after decoding

Fig 4.4a: Original Image

Fig 4.4b: Compressed Image

Fig 4.4c: Enhanced Image

Fig 4.4d: Original image

Fig 4.4e: Compressed Image

Fig 4.4f: Enhanced Image

Figure 4.4
4.2.5 MODELING OF ANALOG LOW PASS FILTERS (LPF)

Most recent video compression coding standards such as H.261, H.263, MPEG-1, 2, 4 use the block-based discrete cosine transform (DCT) and quantization to reduce spatial redundancy, the temporal prediction to diminish temporal redundancy, and the variable length coding (VLC) for Statistical redundancy reduction. The basic picture types in video compression are the inter-picture if the motion prediction is applied, and the intra-picture if no the temporal prediction is applied. As the degree of compression increases, a coding algorithm employing the block-based transform starts presenting the so called blocking artifacts which show structured discontinuities of pixel values along block boundaries. This is primarily due to independent block-based transform of each block and subsequent coarse quantization of transform coefficients neglecting correlation over adjacent blocks. According to the characteristic of human visual system (HVS), the blocking artifacts are more perceivable in low frequency region than in high frequency region. Therefore, this makes a picture looking very unpleasant. The artifacts differently appear in an intra and an inter picture.

The intra picture carries out transform and quantization without temporal prediction, so the reconstructed intra picture typically shows more structured blocking artifacts than the inter picture. But, the inter picture actually has more complex appearance of blocking artifacts due to the temporal prediction an inverse transform. It means to have an additional complexity in reconstruction. Since we are interested in applying the post-processing in real-time applications, we propose a post-processing which carries out adaptive IIR low pass filtering twice in the spatial domain. Finally, we extend the algorithm to apply to the chrominance components simultaneously.

Low pass filtering is a very simple but an efficient tool to reduce the blocking artifacts in low frequency region. But, it is also easy to over-smooth high frequency components. Blocking and ringing effects resulting from block DCT coding are high frequency artifacts. To reduce this, this algorithm applies low pass filtering to the region where artifacts occur. A space-invariant filtering method to reduce median filter was applied to
edges. It was reported that combination of these nonlinear filters can effectively re-block artifacts in image coding was first proposed by Reeves and Lim (Reev84), to maintain the sharpness of the reduced artifacts in block DCT coded images. However, considering additional temporal artifacts such as age statistics often causes the loss of high frequency details such as edges. Therefore, a number of adaptive spatial flickering and motion unsmoothness, it is usually integrated with temporal filtering.

Hence instead of direct filtering, we can consider the scan line as a continuous–time signal. Then the output of the filter is estimated with respect to the shape of the input without actual filtering. It is found through extensive experiments that the objective quality of the post processed image by the sine wave model is better than that of exponential model.

This algorithm is tested on a various sample and the quality of the output image is presented. Here the images i) Girl (256 x 256) and ii) Flower Garden (256 x 256) are selected and they are subjected to the artifact reduction by this algorithm.

Fig .4.5a:Original Image  
Fig .4.5d:Original image
4.3 PROPOSED POST-PROCESSING METHOD

Interpolation is a very simple way to correlate the values with respect to its neighboring pixels. Using this we create a linear polynomial which is smooth and continuous along the block boundary reducing the blocking artifact. Since the pixels in the decoded images are not linear and correlated it is possible to over smooth the high frequency components and the edges. Hence we classify the block boundary so that we adaptively adjust the level of interpolation with changing the characteristics of the images. As shown in the figure 4.6, the reconstructed frames from video decoder are sent to a simple block boundary classification method and edge detection method before get interpolated. The interpolation is done based on the output of both block boundary classification and edge detection method.
Fig. 4.6. The Flow Diagram of complete post-processing

4.3.1 Block Boundary Classification

The classification of blocks remains a major issue in all the adaptive post-processing methods because they i) increase the adaptivity of the method and ii) preserves the high frequency coefficients and edges. The block classifications are done either in pixel-domain or in DCT-domain. The DCT-domain approach remains the most efficient if the artifacts created are mainly due to DCT-based based block coding [Kim99]. However this method is unable to detect the blocking artifacts created by block based motion-estimation and vector quantization. Hence the pixel domain approach remains the most widely used method for block classification.

In most pixel-domain block classification algorithms [Rama86, Apos99] the blocks are classified based on analyzing all the pixels in the block. Thus block classification is done based on local characteristics of pixels. Liu and Jayant proposed an adaptive post-processing algorithm which recognizes that the visibility of noise depends on local signal characteristics and classifies the video signals into different classes. Apostolopoulos and Jayant [Apos99] proposed a post processing algorithms that uses pixel-by-pixel processing to identify and reduce blocking artifact. The major drawback in above methods is they do not compare pixels in the ambient blocks with which it actually creates the blocking artifact effect. Thus the computational complexity of this type of block classification is high.
Hence the classification is applied in the block boundaries thus reducing the number of pixels. Chae, Sohn and Jun Oh proposed a new post-processing technique in which classification of blocks is done based on block boundary of all the sides. Similarly Jeon and Jeong proposed a block boundary discontinuity measure as the sum of the squared differences of pixel values along the block boundary.

In our proposed method we are implementing a simple block boundary classification method, i.e., we are classifying only the block boundary and not the blocks in the video frames, since the blocking artifact results mainly due to signal discontinuities across the block boundary. We are motivated by the observation that,

Artifact is more annoying when there is same level of transition or discontinuity in the values of all pixels along the block boundary. (as shown in figure 4.7b). Artifact is not notable if the signal discontinuity is not consistent along the boundary. (as shown in figure 4.7c)

![Fig. 4.7a](image)

![Fig. 4.7b](image)  ![Fig. 4.7c](image)  ![Fig. 4.7d](image)

The above points can be easily visualized from the images shown in the figure 4.7 the table tennis video frames of MPEG 2. The three regions of the reconstructed image are boxed and shown in figures 4.7b, 4.7c and 4.7d for reference. In the figure 4.7b we can see that blocks are more
annoying as the difference in the pixels values is very consistent along the boundary between two flat pixel blocks. In the same figure the flat block that forms with an edge and texture block doesn’t exhibit a noticeable artifact as due to non-consistency in signal distortion. In figure 4.7c, as there are artifact, as the blocks are texture region and since we do not have a consistent signal distortion, the artifact are not notably annoying. In figure 4.7d the block boundaries that lies along the edges do not produce much artifact effect. Thus artifact effect emitted by a block boundary between two blocks depends upon the surrounding pixels.

Hence we classify the block boundary by considering the pixels present around each boundary between the blocks. The difference D between the pixels lying on either side of the boundary in calculated as

\[ D_x = \text{abs} \left( p_{i,j} - p_{i,j+1} \right), \quad 0 \leq x \leq n, \quad (4.6) \]

where, \( n \) is the size of the block and \( p_{i,j} \) is the pixel at position \( (i,j) \). The values of \( i \) remains multiple of \( n \) for horizontal boundaries of blocks and the values of \( j \) remains multiple of \( n \) for vertical boundaries of blocks. The mean \( E \) and the variance \( V \) are calculated from the values of \( D \) which are used to classify the block boundary.

If variance \( V < \text{th\_flat} \) and mean \( E \) lies between certain values i.e., \( m < E < n \) then block boundary is classified as Flat boundary.

Else If variance \( V < \text{th\_texture} \) and \( E \) is less than certain threshold value \( E < p \)

To classify block boundary as Texture boundary

And if \( V > \text{th\_edge} \) then the block boundary are classified as Edge boundary.

The threshold values \( \text{th\_flat}, \text{th\_texture}, \text{th\_edge} \) and the limits \( m, n \) and \( p \) determine the sensitivity of the images.
4.3.2 Reduction of Blocking Artifact by Interpolation

Since interpolation is used for the construction of polynomials, we are using the Lagrange interpolation for the reconstruction of block boundaries in the video coded frame. The video frame, which are considered as a two-dimensional matrix is considered as a separate one-dimensional array of polynomials in each row and column format. Interpolation is performed to this low frequency block. In a large flat area, the difference of coefficients from adjacent blocks can cause severe blocking effects, which are not limited on block boundary area. Hence there is a break in the normal flow of pixel. In our method we are considering the values of pixel in row wise and column wise as continuous polynomial function. Here we are considering artifact as a discontinuity in the normal flow of the polynomial. Hence using interpolations, we predict pixel values at the block boundaries. These predicted values are used for the reconstruction of polynomials and thus reduce artifacts.

In this method interpolation is applied to the pixels in the boundary in a step-by-step manner. Since during interpolation we are taking more than two pixels for consideration, the values of the pixel determined is more accurate. For a set of data points $n$ and $n > 1$ the elementary Lagrange interpolation formula is given by,

$$l_i^n(x) = \prod_{j=1, j \neq i}^{n} \frac{x - x_j}{x_i - x_j}, \quad i = 1, 2, 3, \ldots, n$$

(4.7)

$l_i^n(x)$ is a polynomial with degree no greater than $n-1$. Its value at any data point $x_k$ within the data set is either 1 or 0, which is equivalent to

$$l_i^n(x_k) = \delta_{ik} = \begin{cases} 1, & i = k \\ 0, & i \neq k \end{cases}$$

(4.8)

Here we are following a three-step procedure, so that we have better reconstruction of signals in the block boundaries.
Fig. 4.8. Position of interpolated pixels in vertical and horizontal directions

Thus we interpolate the pixels from \( v_0 \) to \( v_9 \) along the block boundary based upon the block boundary classification and edge detection as shown in figure 4.8. The algorithm for the proposed interpolation is given by,

**Step 1** : For all block boundaries both horizontal and vertical do the following steps 2 to step 4.

**Step 2** : If the block boundary is classified as flat block then, the interpolating formula (3) is applied to pixels \( v_2 \) to \( v_7 \)

\[
v_n = (-1 \times v_{n-2} + 4 \times v_{n-1} + 4 \times v_{n+1} - v_{n+2}) / 6
\]  

(4.9)

**Step 3** : Else if it is a texture block the eqn 4 is applied for pixels \( v_4 \) and \( v_5 \)

\[
v_n = \left( \frac{(v_{n-1} + v_{n+1})}{2} + v_n \right)
\]  

(4.10)

**Step 4** : Stop the execution.

4.4. EXPERIMENTS AND RESULT

Experiments are carried out to evaluate the performance of various post-processing algorithms. The algorithm is checked with various standard sample sequences and the error is calculated. The samples are decoded with MPEG2 standard with a uniform quantization factor. The proposed algorithm is also checked with other algorithms like i) Modular Post-processing Scheme (MPS) ii) Reducing Quantization Effect (RQE) method (BPM) and iii) Low Pass Filter method for comparative study.
Table 4.1 shows the SNR values averaged over for hundred frames for various video sequences for different post-processing algorithms and compared with the original video sequence. The SNR performance of various post-processing algorithm is given in graphical format for first hundred video frames of table tennis is given in figure 4.7.

**Table 4.1. SNR comparisons of video files processed by various post-processing algorithms**

<table>
<thead>
<tr>
<th>Video</th>
<th>Size</th>
<th>bpp</th>
<th>SNR values</th>
<th>MPP</th>
<th>RQE</th>
<th>LPF</th>
<th>Prop</th>
</tr>
</thead>
<tbody>
<tr>
<td>Table Tennis</td>
<td>352×240</td>
<td>0.0335</td>
<td>13.31</td>
<td>12.36</td>
<td>12.67</td>
<td>13.26</td>
<td></td>
</tr>
<tr>
<td>Flower Garden</td>
<td>352×240</td>
<td>0.0387</td>
<td>12.59</td>
<td>11.36</td>
<td>12.08</td>
<td>12.58</td>
<td></td>
</tr>
<tr>
<td>Bicycle</td>
<td>352×240</td>
<td>0.0328</td>
<td>11.66</td>
<td>10.08</td>
<td>11.01</td>
<td>11.57</td>
<td></td>
</tr>
<tr>
<td>Mobile</td>
<td>352×288</td>
<td>0.0319</td>
<td>12.86</td>
<td>10.84</td>
<td>12.60</td>
<td>12.81</td>
<td></td>
</tr>
<tr>
<td>Susie</td>
<td>352×240</td>
<td>0.0289</td>
<td>13.58</td>
<td>13.10</td>
<td>12.50</td>
<td>13.46</td>
<td></td>
</tr>
</tbody>
</table>

The table 4.2 shows the computing performance of the post-processing algorithm calculated on AMD-Athlon XP -1700 MHz machine with 128 MB ram and running REDHAT™ Linux 8 system. The table 4.2 gives the time for which each process used the CPU in seconds.

![Graph showing SNR Comparisons of various post-processing algorithm for the TABLE TENNIS video sequence](image)

**Fig. 4.9. SNR Comparisons of various post-processing algorithm for the TABLE TENNIS video sequence**
Table 4.2. Computational time comparison of various post-processing algorithms

<table>
<thead>
<tr>
<th>Video</th>
<th>Time (in seconds)</th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>MPP</td>
<td>RQE</td>
<td>LPF</td>
<td>Proposed</td>
</tr>
<tr>
<td>Table Tennis</td>
<td>6.44</td>
<td>3.35</td>
<td>2.81</td>
<td>2.97</td>
</tr>
<tr>
<td>Flower Garden</td>
<td>4.23</td>
<td>3.38</td>
<td>2.71</td>
<td>2.85</td>
</tr>
<tr>
<td>Bicycle</td>
<td>4.02</td>
<td>3.64</td>
<td>2.71</td>
<td>2.99</td>
</tr>
<tr>
<td>Mobile</td>
<td>4.84</td>
<td>4.35</td>
<td>3.27</td>
<td>3.58</td>
</tr>
<tr>
<td>Susie</td>
<td>7.07</td>
<td>3.32</td>
<td>2.81</td>
<td>2.99</td>
</tr>
</tbody>
</table>

From table 4.1 and graph in figure 4.9, that the MPP has the best SNR performance, but the processing time is much longer than other methods due to high computational complexity. The LPF, which has the best computational time but the SNR performance, is low. The proposed method
Figure 4.10

provides good SNR performance and is comparable with MPP and at the same time has low computational complexity equivalent to LPF.

Though the SNR performance gives the level of error present in the images, but when considered to human perception they are not the best evaluator of post-processing algorithm. Hence to analyse the subjective quality of sample images, the original, compressed and reconstructed images are shown in figure 4.10. It was found the MPP and proposed method achieves better subjective quality. The LPF and RQE method does not remove much artifact from the images.

4.5 CONCLUSION

In this chapter, we propose a new post-processing method to reduce the blocking artifacts by processing the spatial components so that it readily applied to real time video decoder. The proposed algorithm classifies the block boundary by analyzing the statistical property of the pixels and reduces the artifact adaptively by correlating the pixels by interpolation based on the block boundary classification.

The experimental results obviously show improvement of subjective quality while maintaining image details with low computational complexity. Hence the proposed method can effectively be used for artifact reduction in block coded and low bit compressed video coding.