Reduction of Blocking Artifacts
Using Side Information

by

Fatih Kamisli

B.S. Electrical and Electronics Engineering
Middle East Technical University, 2003

Submitted to the Department of Electrical Engineering and Computer Science in Partial Fulfillment of the Requirements for the Degree of

Master of Science in Electrical Engineering

at the

Massachusetts Institute of Technology

February 2006

© 2006 Massachusetts Institute of Technology
All rights reserved

Signature of Author..........................

Department of Electrical Engineering and Computer Science
January 20, 2006

Certified by....................

Jae S. Lim

Accepted by..........................

Chairman, Committee for Graduate Students
Reduction of Blocking Artifacts
Using Side Information

by

Fatih Kamisli

Submitted to the Department of Electrical Engineering and Computer Science
on January 20, 2005 in Partial Fulfillment of the Requirements for the Degree of
Master of Science in Electrical Engineering

Abstract

Block-based image and video coding systems are used extensively in practice. In low bit-rate applications, however, they suffer from annoying discontinuities, called blocking artifacts. Prior research shows that incorporating systems that reduce blocking artifacts into codecs is useful because visual quality is improved. Existing methods reduce blocking artifacts by applying various post-processing techniques to the compressed image. Such methods require neither any modification to current encoders nor an increase in the bit-rate.

This thesis examines a framework where blocking artifacts are reduced using side information transmitted from the encoder to the decoder. Using side information enables the use of the original image in deblocking, which improves performance. Furthermore, the computational burden at the decoder is reduced. The principal question that arises is whether the gains in performance of this choice can compensate for the increase in the bit-rate due to the transmission of side information. Experiments are carried out to answer this question with the following sample system: The encoder determines block boundaries that exhibit blocking artifacts as well as filters (from a predefined set of filters) that best deblock these block boundaries. Then it transmits side information that conveys the determined block boundaries together with their selected filters to the decoder. The decoder uses the received side information to perform deblocking. The proposed sample system is compared against an ordinary coding system and a post-processing type deblocking system with the bit-rate of these systems being equal to the overall bit-rate (regular encoding bits + side information bits) of the proposed system. The results of the comparisons indicate that, both for images and video sequences, the proposed system can perform better in terms of both visual quality and PSNR for some range of coding bit-rates.

Thesis Supervisor: Jae S. Lim
Title: Professor of Electrical Engineering
Acknowledgements

First and foremost, I would like to thank my thesis advisor Professor Jae S. Lim for his guidance and support throughout my thesis work. He has introduced me to the problem addressed in this thesis and steered the research towards meaningful directions. I would like to thank also to the other members of the Advanced Telecommunications and Signal Processing Group (ATSP); Brian Heng, Chang D. Yoo and Cindy LeBlanc.

I would like to thank my mother, my father, and my sisters Filiz and Sema. Their support, encouragement and love have been invaluable to me.

Finally, I would like to thank Basak for her support and love from so far distances.
Table of Contents

1. Introduction ................................................................................................................... 8
   1.1 Thesis Motivation and Overview ......................................................................... 9

2. Background .................................................................................................................. 13
   2.1 Overview of Image and Video Coding ............................................................... 13
      2.1.1 Image and Video Processing Definitions ............................................... 13
      2.1.2 Image Compression Basics ...................................................................... 15
      2.1.3 Video Compression Basics ...................................................................... 17
   2.2 Previous Research ............................................................................................. 20
      2.2.1 Adaptive Filtering Methods ..................................................................... 20
      2.2.2 Estimation Theoretic Methods .................................................................. 22
      2.2.3 POCS Based Methods ............................................................................. 22
      2.2.4 Wavelet Based Methods ......................................................................... 23
      2.2.5 Lapped Orthogonal Transforms ............................................................... 24

3. A System To Reduce Blocking Artifacts Based On Side Information ................... 25
   3.1 Overview of the Proposed System .................................................................... 25
   3.2 Blocking Artifact Detection .............................................................................. 28
      3.2.1 Block Classification ............................................................................... 29
      3.2.2 Adaptive Block Boundary Analysis ....................................................... 30
   3.3 Filter Selection .................................................................................................. 41
   3.4 Coding of Side Information ............................................................................. 41

4. Adaptation Of The System To Video Applications ............................................... 44

5. Experimental Results ............................................................................................. 46
   5.1 Experimental Results for Images .................................................................... 46
   5.2 Experimental Results for Video Sequences .................................................... 66

6. Conclusions .............................................................................................................. 79
   6.1 Summary ........................................................................................................... 79
   6.2 Future Research ............................................................................................... 80

Appendix ......................................................................................................................... 82
   A. The Filter Set Used in the Experiments ............................................................ 82
   B. Run-length Encoding of Side Information ....................................................... 85
   C. Training Data for the Huffman Codes of the Side Information ..................... 88

References ....................................................................................................................... 95
## List of Figures

1.1 Compressed images .................................................................................. 9
2.1 A high-level view of a typical image encoder ........................................ 19
2.2 A high-level view of a typical video encoder .......................................... 19
3.1 Abstract view of the proposed deblurring system based on side information ........................................................................................................ 27
3.2 Overview of Blocking Artifact Detection ............................................... 28
3.3 Block Classification Algorithm ................................................................ 29
3.4 Sample output of the block classification algorithm ............................... 30
3.5 Characteristics of different boundary types ............................................. 31
3.6 Graphical illustration of Step-1 of ss-boundary analysis .......................... 33
3.7 Graphical illustration of Step-3 of ss-boundary analysis .......................... 33
3.8 Graphical illustration of Step-4 of ss-boundary analysis .......................... 34
3.9 Graphical illustration of Step-1 of sn-boundary analysis .......................... 37
3.10 Graphical illustration of Step-3 of sn-boundary analysis ........................ 38
3.11 Graphical illustration of Step-4 of sn-boundary analysis ........................ 38
3.12 Blocking Artifacts detected on the Cameraman and Lena images .......... 40
3.13 Scan orders of block boundaries for run-length encoding ........................ 42
4.1 Frame 80 of the Foreman sequence ......................................................... 44
5.1 The 256x256 pixel images that are used in the experiments .................... 48
5.2 The Sailboat image that is used in the experiments .................................. 49
5.3 The Airplane image that is used in the experiments ................................. 50
5.4 (a) PSNR vs bit-rate for the Lena image (256x256 pixels) .......................... 53
(b) PSNR vs bit-rate for the Cameraman image (256x256 pixels) .............. 53
5.5 (a) PSNR vs bit-rate for the Goldhill image (256x256 pixels) ................. 54
(b) PSNR vs bit-rate for the Tree image (256x256 pixels) ......................... 54
5.6 (a) PSNR vs bit-rate for the Sailboat image (512x512 pixels) ................. 55
(b) PSNR vs bit-rate for the Airplane image (512x512 pixels) .................. 55
5.7 Plot of side information bit-rate vs overall coding bit-rate for the Lena .... 56
5.8 Comparison of the Lena image coded at 0.25 bpp ................................ 58
5.9 Comparison of the Lena image coded at 0.3 bpp ................................ 59
5.10 Comparison of the Lena image coded at 0.35 bpp ................................ 60
5.11 Comparison of the Lena image coded at 0.4 bpp ................................ 61
5.12 Comparison of the Lena image coded at 0.45 bpp ................................ 62
5.13 Comparison of the Cameraman image coded at 0.25 bpp ..................................... 63
5.14 Comparison of the Cameraman image coded at 0.35 bpp .................................... 64
5.15 Comparison of the Goldhill image coded at 0.30 bpp ............................................ 65
5.16 Test sequences ............................................................................................................. 68
5.17 PSNR values for each frame of the Foreman sequence are plotted for the standard encoder, the encoder with deblocking method in [8], and the deblocking method based on side information ................................................................. 72
5.18 PSNR values for each frame of the Silent Voice sequence are plotted for the standard encoder, the encoder with deblocking method in [8], and the deblocking method based on side information ................................................................. 73
5.19 Bit-rate of side information for each frame of the Foreman sequence is plotted ........ 74
5.20 Frame 13 and frame 180 of the Foreman sequence coded at overall bit-rate of 292 Kb/s are shown ..................................................................................................................................... 75
5.21 Frame 93 and frame 147 of the Foreman sequence coded at an overall bit-rate of 369 Kb/s are shown ..................................................................................................................................... 76
5.22 Frame 45 and frame 135 of the Silent Voice sequence coded at an overall bit-rate of 210 Kb/s are shown ..................................................................................................................................... 77
5.23 Frame 15 and frame 205 of the Silent Voice sequence coded at an overall bit-rate of 293 Kb/s are shown ..................................................................................................................................... 78
List of Tables

5.1 Comparison chart of PSNR values of the non-processed image, the method in [15], and the method based on side information at different bit-rates .........................52

5.2 Comparison of the average PSNR values of the test sequences for the standard encoder, the encoder with deblocking method in [8], and the encoder with deblocking method based on side information .................................................68

B.1 Huffman Code table for the coding of the side information .............................................85
Chapter 1
Introduction

A large number of communication applications contain pictures or video. For example, many webpages on the Internet contain pictures. Commonly used video applications include video conferencing, video over the Internet, videophones and high definition television (HDTV) among many others. Uncompressed pictures or video require very high data rates. The transmission or storage of this data may be impractical, or even impossible for many applications. However, there is significant redundancy both in video and in pictures, allowing compression of data. Yet, compression ratios above a certain level are achieved at the expense of some loss of detail in the image or video. The amount of compression is determined by the bandwidth requirements of the application. Applications having a very small bandwidth available need very high compression ratios. Under very high compression ratios, however, some visually annoying artifacts known as blocking artifacts occur, in addition to loss of detail. Blocking artifacts severely reduce the visual quality of the image or video, making it unpleasant to the viewer, as shown in Figure 1.1. Reduction of blocking artifacts is essential to render the compressed image or video acceptable to the human viewer.

The image or video coding standards that are widely used, such as JPEG, MPEG2 and MPEG4, code images or frames on a block-by-block basis. Each block is coded in three main steps: transform, quantization, and entropy coding. These steps are applied to each block independently of other blocks. Therefore, as the compression factor increases, the correlation between two adjacent pixels that fall into different blocks decreases because the reconstruction of these pixels becomes poorer in an independent manner. Consequently, an artificial discontinuity appears along this block boundary. This can be seen as an intuitive explanation for the reasons for the blocking artifacts. Various methods have been proposed in the literature to reduce blocking artifacts. In this thesis, a
new approach to reduce blocking artifacts is examined. The approach is based on sending side information from the encoder to the decoder to aid the decoder in performing deblocking.

1.1 Thesis Motivation and Overview

Compression systems based on Transform Coding operate on a block-by-block basis to exploit spatial redundancy in images. Since each block is coded independently of other blocks, this scheme gives rise to artificial discontinuities, often called blocking artifacts, at block boundaries in low bit-rate applications of image and video coding. Even though the overall imagery can easily be understood, the blocking artifacts destroy the natural appearance of the image and greatly annoy the human viewer. Therefore, reducing blocking artifacts is essential to increase the visual quality the viewer perceives.

Reducing blocking artifacts, also called deblocking, is the process of reducing the artificial discontinuity between two blocks below the visibility threshold. This is done by modifying pixel intensities on both sides of the block boundary. However, several
questions arise: Is the discontinuity at the block boundary completely artificial, or is part of it also due to the original signal in the image? How much reduction in the discontinuity is good? Is it sufficient to modify only pixels next to the block boundary or do we also need to change pixel intensities in the interiors of the blocks? Finding answers to all of these questions is difficult if only the compressed image is used. However, if the original image is available, then much better answers can be provided because the original image contains the most reliable information that can be achieved.

One advantage of the availability of the original image is that it enables better detection of the blocking artifact strength and of the amount of detail around the block boundary because comparisons of the compressed image and the original image are possible. For example, during the coding of strong edges, ringing artifacts arise due to loss of significant high frequency information. Ringing artifacts show themselves as artificial fluctuations on both sides of the edge. These fluctuations can be misleading as they can be interpreted as high frequency content or detail by existing methods and thus might not be reduced for means of maintaining details. However, using the original image, it is possible to understand that these fluctuations are artificial and do not exist in the original image, and therefore must be removed.

Another, and maybe the most important, advantage of the availability of the original image is the fact that the result of any deblocking action can be compared against the original. Hence, the success in removing the blocking artifact and maintaining details can be evaluated. This property is very important since, for example, it enables to see if the performed deblocking action brings the block boundary closer to the original image or causes unnecessary blurring.

One motivation for the deblocking method based on side information is the ability to use the original image in deblocking. For that, however, deblocking must be performed at the encoder and then necessary information must be sent to the decoder. There is another motivation for deblocking based on side information. The computational burden related to the analysis of the local regions prior to deblocking is moved from the decoder to the encoder. The decoder only needs to decode the side information and do what the side information requires. No computation to extract information about the surrounding
region of block boundaries is required. This property can be of great value for some applications where decoder simplicity is of concern.

Sending side information results in an overhead in the bit-rate. This might seem as a bottleneck of the proposed method. However, in deblocking based on side information, the encoder dictates the deblocking at the decoder; the decoder only does what the side information sent by the encoder requires. Since the encoder has access to the original image, the deblocking dictated by the encoder will be more successful than a so-called blind deblocking performed by the decoder using only the compressed image. As a result, the loss coming from the overhead in the bit-rate due to sending side information can be compensated by the use of the original image in deblocking. Indeed, it is only such situations in which deblocking based on side information makes sense. We have observed that in low bit-rate applications of image coding, this is the case for some important range of bit-rates.

One major disadvantage of reducing blocking artifacts using side information is that current image and video coding standards do not allow it. Current encoders need to be changed to enable the reduction of blocking artifacts using side information. However, it is worth noting that the idea of side information is not new in video coding; side information has been used for various purposes. For example, video encoders choose between intra and inter coding modes for each macroblock. The encoder must convey its choice to the decoder by coding some bits, which can be seen as side information, in the bit-stream.

In light of the above discussion, a sample system that deblocks the compressed image using side information is proposed. It is worth noting that the main purpose of this thesis is to demonstrate the benefits that can be obtained from using side information for deblocking. We do not claim that the proposed system is optimal. Other implementations that use side information for deblocking are possible and can achieve better results. In the proposed system, the encoder, first, determines block boundaries that exhibit blocking artifacts as well as filters, from a predefined set of filters, that best deblock these block boundaries. Then it transmits side information that conveys the determined block boundaries together with their selected filters to the decoder. The decoder uses the received side information to perform the corresponding filtering for each boundary.
An overview of the remaining chapters is as follows: In the second chapter of this thesis, first, a background for image and video coding is given, then, a summary of existing deblurring methods in the literature is provided. Methods based on Adaptive Filtering Theory, Estimation Theory, Projection onto Convex Sets and Lapped Orthogonal Transforms are each explained in a separate section of Chapter 2. Chapter 3 provides an in-depth discussion of the proposed deblurring system based on side information. This discussion is based on images. The first section presents an overview of the system. The following three sections explain the three complementary parts of the system in detail. Chapter 4 discusses the adaptation of the method described in Chapter 3 to video applications. Similarities of this method to and differences of this method from the image based deblurring method are given. Experimental results of the proposed methods are presented in Chapter 5. We discuss what gains can be obtained (and under what conditions) with deblurring based on side information. Finally, Chapter 6 summarizes the main conclusions of this thesis and points to future research directions.
Chapter 2
Background

This chapter is devoted to the background necessary for discussing the work done in this thesis. Section 2.1 presents definitions used in image and video coding, and provides an overview of the fundamentals of image and video coding. Section 2.2 summarizes the previous research done on the reduction of blocking artifacts.

2.1 Overview of Image and Video Coding

2.1.1 Image and Video Processing Definitions

An image signal can be considered as a continuous function of two dimensions; horizontal and vertical position. This signal must be discretized for digital processing and transmission. This involves sampling the signal along the two dimensions and quantizing those samples. The resulting image samples are often referred to as picture elements or pixels. The number of pixels used for the entire image is often called the spatial resolution of the image.

To represent color, additive combination of three primary colors, red, green and blue, is used. This representation of color information is called the RGB color system. Quantization is often done with 8 bit accuracy for each of the color components. Consequently, each pixel is associated with three color components and each of them is represented with 8 bits. There is another representation called the luminance-chrominance representation, which can also be used to represent color information. One example for this representation is the YIQ system. While the Y component, called the luminance component, is primarily responsible for the perception of brightness of a color image, I and Q components, called the chrominance components, are responsible for the perception of hue and saturation of a color image. Using a linear transformation, the RGB
system can be transformed to the luminance-chrominance system and vice versa. In image or video compression, the luminance-chrominance system is preferred for reasons that will be explained in the next sections.

Similar to an image signal, a video signal can be considered as a continuous function in three dimensions; horizontal position, vertical position and time. Sampling the video signal in the temporal dimension gives us a sequence of images. These images can be sampled in the spatial dimensions and then quantized, as explained above. The sampled and quantized version of each of these images is called frame in video processing terminology. This way of discretizing the video signal is called progressive scanning. A video sequence can then be seen as a collection of frames, with equal dimensions, sampled at equally spaced time intervals.

In video processing, sampling in the temporal and spatial dimensions can also be coupled. In particular, while sampling in the temporal dimension, even lines and odd lines of a frame are used interchangeably. In other words, while at one temporal sampling instant, only even lines of the frame are used, at the next temporal sampling instant only odd lines are used. This is called interlaced scanning. It enables doubling the display rate without a frame store. This is important for minimizing the perceived flicker of the display, without reducing the total number of lines per frame. Interlaced scanning is used in conventional television systems. Progressive scanning, on the other hand, is used mostly in computer displays and flat panel television displays.

Compression algorithms for image or video are often lossy. This means that the reconstructed version of the compressed data is not identical to the original. For the purpose of this thesis, the distortion caused by losses during data compression will be quantitatively measured using the peak-signal-to-noise-ratio (PSNR). The PSNR of a frame is defined as

\[
PSNR = 10 \cdot \log_{10} \left( \frac{255^2}{MSE} \right) \quad (1)
\]

where the mean-square-error (MSE) is the average squared difference between the original and distorted frames, \( F_0 \) and \( F_d \).
\[ \text{MSE} = \frac{1}{M \cdot N} \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} (F[m,n] - F_d[m,n])^2 \]  
(2)

Even though PSNR provides a useful estimate for assessing image or video quality, perceived quality and PSNR are not always directly correlated. However, due to its simplicity, its use is common practice in the image or video processing communities. Hence, this thesis also uses PSNR for evaluating image or video quality.

2.1.2 Image Compression Basics

One goal of image compression is to represent an image with as few bits as possible. The reduction in bit rate is achieved by exploiting the redundancy and irrelevancy present in the image. Sources of redundancy are

- **color space**: RGB components are correlated among themselves.
- **spatial**: Nearby pixels are often correlated with each other.

Among various compression methods, *Transform Coding* is widely used. In this section, we explain how compression systems based on Transform Coding exploit the above mentioned redundancies and outline the basic structure of a typical image compression system based on Transform Coding.

The redundancy in the RGB components is exploited by converting RGB values to luminance-chrominance values, such as YIQ. In the YIQ domain, there exists less correlation among the components allowing for better approximation of the original data upon compression. Furthermore, most of the high-frequency content of the image is packed into the luminance component, leaving the chrominance component with significantly smaller amount of high-frequency content. Therefore, downsampling the chrominance components does not seriously affect the high-frequency details of a color image and allows coding of a smaller number of samples than in RGB domain.

The redundancy in the spatial domain is evident from the fact that pixel intensities of each of Y, I, Q components in a small spatial region are highly correlated. A transform that decorrelates pixel intensities in this small spatial region (e.g. 8x8 blocks) and packs the energy in as few coefficients as possible would be ideally suited for this task, because
in this case, a good approximation to the original data would be possible by coding only the few high energy coefficients. The Karhunen-Loeve Transform is the statistically optimal transform for this task. However, it is signal dependent and is computationally very involved. Instead, the Discrete Cosine Transform (DCT) is widely used in practice. For most natural images, the DCT is also shown to be very effective at decorrelation and energy compaction. In particular, the DCT results in high energetic low spatial frequency components for most natural images since they possess significant low frequency content. Furthermore, fast algorithms for the computation of the DCT are available. The transform idea is the heart of compression systems based on Transform Coding.

After the transformation step, the resulting transform coefficients will have a significant portion of their energy compacted in a small fraction of the coefficients. For compression purposes, not only can we discard low energy coefficients but we can also reduce the representation accuracy of the high energy coefficients. This can be done by quantizing the transform coefficients. Quantization is a lossy operation, that is, it is irreversible. Quantization is an essential part of image compression systems since it also exploits some Human Visual System (HVS) features enabling further compression. In particular, consider the quantization of only one specific transform coefficient; if we go from very fine quantization to very coarse quantization, there will be a specific quantization interval up to which the reconstructed image data will be indistinguishable from the original for the HVS. Hence, by selecting the quantization interval in light of this observation, more compression can be achieved. It should also be noted that different transform coefficients have different perceptual importance for the HVS. As a result, different quantization intervals can be associated with different transform coefficients. For example, high spatial frequency coefficients of the DCT may be quantized more coarsely than low spatial frequency coefficients of the DCT.

In light of the above discussions, the basic structure of a typical image compression system can be given as follows: First, the acquired digital image in the RGB domain is converted to the YIQ domain and, I and Q components are downsampled by a factor of 2, both in the horizontal and vertical direction. Then, each of the resulting components undergoes block-based DCT with a block size of 8x8 pixels. In other words, each of the components is divided into pieces of 8x8 blocks and each block is
transformed using the DCT resulting in 64 DCT coefficients for each block. Next, for each block, quantization of the DCT coefficients is performed using a quantization table specifying possibly different quantization intervals for each of the 64 DCT coefficients. The quantization tables used for luminance blocks and chrominance blocks might be different as well. Many of the 64 DCT coefficients in a block may be quantized to zero. Furthermore, for most blocks, the nonzero quantized coefficients will reside in the lower spatial frequency region. To take advantage of these facts and also exploit all remaining statistical redundancy, entropy coding schemes are used. Common practice is to scan one block of quantized DCT coefficients in zigzag fashion starting from the lowest spatial frequency coefficient to the highest spatial frequency coefficient. The resulting string of quantized coefficients undergoes run-length encoding. Run-length encoding produces symbols that carry two pieces of information: number of zeros in the run, and the non-zero value that terminates the run. These symbols are coded using variable-length-codes (VLC). Huffman codes are the most frequently used VLCs. Arithmetic codes are another choice.

Figure 2.1 shows all of these steps graphically. To reconstruct the image at the decoder, reverse of these steps must be performed. For an in-depth understanding of the most widely used image coding system, the JPEG, see [1].

2.1.3 Video Compression Basics

A video sequence was defined to be a sequence of frames sampled at fixed time intervals. For each frame, all of the steps which were explained above for image compression can be used. However, there is one additional significant redundancy in video,

- temporal: Most frames are similar to previous and next frames.

Consecutive video frames typically contain the same imagery, although possibly at different spatial locations [2]. This redundancy can be exploited by coding a given frame and then using it to form a prediction for the next frame, while compensating for the motion between the two frames. To accomplish this, an initial frame must be coded independently of other frames and transmitted to the decoder. Then the motion between the coded frame and the current frame to be coded must be estimated, and an appropriate
prediction of the current frame must be made. The error in the prediction, or residual, is then coded and transmitted. The process of estimating the motion between frames is known as motion estimation (ME). The process of forming a prediction while compensating for the motion is known as motion-compensated prediction or MC-prediction.

Similar to the block-based transform in image compression, ME is also performed on a block-by-block basis. This is called block-matching. In block-matching, the current frame is partitioned into rectangular regions or blocks of pixels, and for each block, a search is performed in the previous frame to find the displacement that provides the best match in a nearby region. The displacement of the best match is represented by a motion vector, which is coded into the bit stream so that it can be used in the decoder. The residual of the prediction is then coded in a similar manner to block-based coding of images. That is, the residual of the block-matching prediction is first transformed by DCT and the coefficients are then quantized and entropy coded.

There are many instances when MC-predictive coding may fail, either globally or locally. For example at a scene change, MC-prediction may produce a residual that can be more difficult to code than the original frame. In this case, MC-prediction should be suppressed and the original frame or block should be coded with intraframe coding. In other words, it should be coded as in image coding; by first applying the DCT on the image data and then quantizing and entropy coding the transform coefficients. Figure 2.2 shows a high-level view of a typical video encoder.
Figure 2.1 A high-level view of a typical image encoder.

Figure 2.2 A high-level view of a typical video encoder.
2.2 Previous Research

Block-based coding is used extensively both in image and video coding systems. In low bit rate applications, this scheme gives rise to blocking artifacts which severely reduce the visual quality of the image or video. Reducing blocking artifacts is essential to render the compressed visual data acceptable to the viewer.

Various deblocking methods have been proposed in the literature to reduce blocking artifacts. Many of these methods are based on the post-processing idea. In other words, these methods take only the compressed image (or data that is used by the decoders to reconstruct the compressed image) as their input and process it to reduce the blocking artifacts. There are also approaches that look at the blocking artifact problem from a broader perspective. One such approach is based on using Lapped Orthogonal Transforms (LOT) instead of the block-based DCT.

We provide a brief summary of our literature search in this section. Our literature search indicates that the use of side information to reduce blocking artifacts has not been explored. Since the reduction of blocking artifacts is an important problem, especially in low bit-rate video coding, new approaches should be explored, even if existing methods work reasonably well. Therefore, this thesis discusses the advantages and disadvantages of using side information to reduce blocking artifacts and investigates if one particular example of such methods can perform similar, if not better, to other existing methods.

In the following subsections, we summarize the main ideas of some of the previous research on deblocking. Post-processing-type deblocking methods are based on different theoretical frameworks including Adaptive Filtering, Estimation Theory, Projection onto Convex Sets (POCS) and Wavelets. The basic ideas behind each of these theories and the LOT are explained in the following subsections.

2.2.1 Adaptive Filtering Methods

Since blocking artifacts are artificial discontinuities along block-boundaries, they can be considered as high frequency artifacts. Then, a simple solution is to apply low-pass filtering to the regions where they occur. This is the basis of the filtering method. A
space-invariant filtering method was first proposed in [3]. Maintaining sharpness of detailed regions and edges while sufficiently smoothing blocking artifacts in smooth regions of an image requires adaptation of the filtering to the local characteristics of an image. This idea of adapting the filtering to local characteristics is also supported by the masking effect, which says that the HVS is less sensitive to artifacts in texture regions than in smooth regions of an image.

The above observations have led to a number of adaptive filtering methods in the literature [4-8]. For example, in [5], first blocks with visible blocking artifacts are detected, then an edge map for these blocks is created, and finally edge-sensitive filtering is performed. Edge-sensitive filtering is performed by low pass filtering pixels near edges using only pixels on the same side of the edge. Pixels on the edges are left untouched. In [6], the filtering method has two modes, filtering for smooth regions and filtering for other regions. Mode decision is performed for each row of a vertical block boundary or each column of a horizontal block boundary by examining the flatness of this row or column respectively. Filtering for smooth region is performed by using a 9 tap one dimensional low pass filter. For example, using this filter along each row of a vertical block boundary, the innermost four pixels are modified on either side of the block boundary. Filtering for other regions is performed by modifying only the pixels adjacent to the block boundary, based on a row wise 4-point DCT analysis on the pixels across the vertical block boundary. There is also an option that leaves some block boundaries untouched. Other methods based on adaptive filtering are very similar in spirit to the two methods which were explained above.

The main issue with methods based on adaptive filtering is the tradeoff between sufficient smoothing and maintaining details. This tradeoff is accounted for by the adaptive nature of the algorithm. However, still some problems occur, especially in some complex regions of an image. Furthermore, there are regions in an image where some texture is introduced by the lossy compression algorithm which was actually not present there in the original image. For example, ringing artifacts are caused during the coding of strong edges. Ringing artifacts manifest themselves as artificial fluctuations on both sides of the edge. However, in the original image both sides of the image are quite smooth. Ringing artifacts can mislead the algorithm as the algorithm might interpret blocks with
blocking artifacts to have high-frequency content that is spread over the entire block, whereas the high-frequency content is actually concentrated only on the edge.

Methods based on adaptive filtering have been developed to reduce blocking artifacts both for images and video. While for images, methods based on other theoretical frameworks such as Estimation Theory and POCS provide good alternatives to methods based on adaptive filtering, the same cannot be claimed for video. This is due to the fact that methods based on both Estimation Theory and POCS require very involved signal processing which make them infeasible for video due to its computational requirements.

2.2.2 Estimation Theoretic Methods

In methods based on Estimation Theory [9-12], probabilistic models are used for the compressed and desirable deblocked images. Then, estimators are derived based on these probabilistic models. To be able to model image characteristics well, complex probabilistic models are used, which then lead to quite involved estimation procedures. For example, in [9], the Hueber-Markov random field model is used to model image characteristics. The corresponding estimation procedure then leads to a constrained minimization problem which the authors solve using an optimization technique, called gradient projection. This requires several iterations until a sufficiently deblocked image is obtained.

These methods require excessive amount of computation. While this may be acceptable to a certain degree for reducing blocking artifacts in images, it is not acceptable for reducing blocking artifacts in video.

2.2.3 POCS Based Methods

POCS theory was first applied to image restoration problems by Youla [13]. After the blocking artifact problem began to be studied by the image coding community, many methods based on POCS theory for reducing blocking artifacts flourished [14-18]. In methods based on POCS theory, the desired properties for the deblocked image are captured by defining appropriate convex sets. For example, one local convex set could be
defined as the set of all possible neighboring two blocks whose sum of absolute values of the difference of pixels along the block boundary is smaller than a threshold. Attention should be paid to the convexity of the sets because that is what enables the resulting algorithm to converge to a meaningful deblocked image.

After defining convex sets, their projection operators are established. Projecting a point (or vector) onto a convex set gives the element of that set that is closest to the point (or vector) that was projected in some appropriate norm. The $l_2$ norm is used widely. For example, for the above defined convex set, the projection would give us the neighboring two blocks which are closest (in terms of $l_2$ norm) to the initial neighboring two blocks and also satisfy the requirements of the convex set, which is to have sum of absolute values of the difference of pixels along the block boundary smaller than the threshold.

If we want the deblocked image to satisfy several properties, we have to define several convex sets. The desired deblocked image can then be found after several iterations of projections. For example, if three convex sets were defined, then iterations of projections would be as follows: Project the initial image onto convex set 1, then project the result onto convex set 2, then project the result onto convex set 3, then project the result onto convex set 1, then project the result onto convex set 2 and so on. The iterations are guaranteed to converge to a unique image by the theory of POCS. Even though the number of iterations for convergence is infinite, a sufficiently deblocked image is likely to be obtained after a small number of iterations.

These methods require excessive amount of computation. While this may be acceptable to a certain degree for reducing blocking artifacts in images, it is not acceptable for reducing blocking artifacts in video.

### 2.2.4 Wavelet Based Methods

Some methods reduce blocking artifacts using the wavelet domain representation of images. These methods start by transforming the compressed image with blocking artifacts to the wavelet domain. Then, the wavelet coefficients related to block boundaries with blocking artifacts are modified. The algorithms to modify these coefficients differ
between the various methods based on wavelets and they are the most distinguishing feature between these methods.

In [23], an over-complete wavelet representation is used. In other words, wavelet representations in all scales have the same number of coefficients as the image with blocking artifacts. First, an edge map is created using the correlation of wavelet coefficients across scales. Locations above a threshold are identified as edges. Then, wavelet coefficients at non-edge locations are set to zero while coefficients at edge locations are untouched. Finally, the low-pass component (the scaling coefficients) is averaged at the block-boundary locations with its neighbors. The inverse wavelet transform using the modified coefficients gives the deblocked image.

2.2.5 Lapped Orthogonal Transforms

Lapped Orthogonal Transforms (LOT) provide a way for avoiding the occurrence of blocking artifacts in the first place, rather than reducing them after they have occurred. They are different from regular block transforms. They have basis functions that overlap to adjacent blocks. In other words, to produce transform coefficients for one block, not only samples of that block are used, but also samples from neighboring blocks are used. This means that each block is no more coded independently of other blocks. Hence, blocking artifacts are less prominent. However, other artifacts tend to appear, such as increased ringing effects around edges due to longer basis functions. Detailed discussions on image coding based on LOT can be found in [20,21].

Even though the LOT might seem to be an attractive solution, many standards were developed relying on the block-based DCT and large amounts of investment were made. The amount of improvement the LOT provided could not justify a shift from the DCT to LOT. Overall, the DCT is used immensely in practice, especially in video coding. Therefore, it is important to find ways to reduce blocking artifacts in block-based DCT coded image and video coding applications.
Chapter 3
A System To Reduce Blocking Artifacts Based On Side Information

From a broad perspective, side information can be any information aiding the decoder in reducing blocking artifacts. In this chapter, we describe a system to reduce blocking artifacts based on side information. We do not claim that this particular system is optimal. Other systems based on side information are possible, and may achieve better results. Our primary goal is to demonstrate that the approach to use side information for deblocking has significant benefits and should be explored further in the future research on deblocking. The discussion of the system is based on images.

In the proposed system, the encoder determines block boundaries that exhibit blocking artifacts as well as filters (from a predefined set of filters) that best deblock these block boundaries. Then it transmits side information that conveys the determined block boundaries together with their selected filters to the decoder. The decoder uses this side information to perform the corresponding filtering for each block boundary. This is the essence of the proposed deblocking system based on side information.

In the next section, we give an overview of the proposed deblocking system. The system can be broken down into three complementary parts. Therefore, the next section is followed by three other sections that explain complementary parts of the proposed system in detail.

3.1 Overview of the Proposed System

The essence of the proposed deblocking system is as follows: For each block boundary, the filter that best deblocks that block boundary is selected from a predefined filter set. The filter is selected by comparing the deblocked block boundary with the corresponding
region in the original image. Then, a code that represents that filter is transmitted as side information to the decoder. The decoder filters the block boundary with the filter that is signaled by the received side information.

This simple scenario requires that filtering is performed on each block boundary. However, filtering is not necessary on all block boundaries of an image because there are, most of the times, many block boundaries that do not exhibit visible blocking artifacts. Furthermore, sending side information for those block boundaries increases the bit-rate of the side information.

Considering the above observations, we include a blocking artifact detection step before selecting the best filter for deblocking. The blocking artifact detection step determines whether a block boundary exhibits a visible blocking artifact (using also the original image). If it does, then filter selection is performed on that block boundary and the selected filter is signaled by transmitting that filter's code in the side information. If, however, the blocking artifact detection step decides that no visible blocking artifact exists, then a code signaling that there is no blocking artifact is transmitted for that block boundary. Hence, no filtering is performed on that block boundary. At this point, one might argue that instead of coding that there is no blocking artifact, it would make more sense to code a filter which still improves the appearance of that block boundary. However, this is not true because coding a filter improves the deblocking performance slightly whereas it requires more bits than coding that there is no blocking artifact. Therefore, including the blocking artifact detection step is useful from a rate-distortion point of view. Indeed, it results in a bit-rate reduction of at least a factor of two with small performance degradation, according to our experiments.

There is significant statistical redundancy in the side information resulting from both the blocking artifact detection step and the filter selection step. Therefore, a coding system that removes this redundancy is employed. We use a system that is analogous to the well-known run-length encoding system.

In summary, the proposed deblocking method based on side information can be given in three main steps:

1. Blocking artifact detection
2. Filter selection
3. Coding of side information.

All of these steps are performed at the encoder. The decoder only needs to decode the received side information and perform corresponding filtering operations on the compressed image. Figure 3.1 summarizes the overall view of the proposed method.

![Diagram of the proposed deblinking system based on side information.](image)

**Figure 3.1** Abstract view of the proposed deblinking system based on side information. (a) Encoder side of the proposed method. (b) Decoder side of the proposed method.
3.2 Blocking Artifact Detection

Blocking artifact detection refers to identifying the block boundaries, in the compressed image, which exhibit visible blocking artifacts. There are very few studies in the literature which, given the original image, consider the measurement of how high the blocking artifact degradation in the whole image is [19]. Rather than quantifying the level of blockiness in the whole image, our goal is to decide whether each block boundary exhibits a visible blocking artifact or not. Deciding whether a block boundary exhibits a visible blocking artifact is a difficult task, which requires careful analysis of the surrounding region in the compressed image as well as in the original image.

Blocking artifacts have different characteristics from region to region in an image. For example, blocking artifacts with equally strong discontinuities at the block boundary may look very annoying in smooth regions of the image while they may not be as disturbing in texture regions. Another example is ringing artifacts. Ringing artifacts are artificial fluctuations created on both sides of an edge upon compression. When a block with ringing artifact comes together with a smooth block, independent of the discontinuity at the block boundary, a highly disturbing picture arises.

To account for the varying characteristics of blocking artifacts, we first classify block boundaries into three classes and then process each class differently. This allows us to adapt our processing to different characteristics of the blocking artifacts as well as to simplify and more easily control the overall blocking artifact detection process. In summary, we break down the blocking artifact detection step into two parts: block classification and adaptive block boundary analysis. See Figure 3.2. In the following two

![Figure 3.2 Overview of Blocking Artifact Detection](image-url)
subsections, we explain the details of these parts.

### 3.2.1 Block Classification

Block Classification classifies each block in the compressed image into either smooth block or non-smooth block. The classification is based on the sum of absolute values of the DCT coefficients of the block. The DC coefficient is excluded in the sum since it does not represent high frequency information. The resulting sum is compared against a threshold. The flowgraph in Figure 3.3 summarizes the block classification. In Figure 3.4, a sample compressed image and the corresponding block classification is shown.

Classification of blocks results in three block boundary types:
- block boundary between two smooth blocks (ss-boundary)
- block boundary between a smooth and a non-smooth block (sn-boundary)
- block boundary between two non-smooth blocks (nn-boundary)

Each block boundary type undergoes different processing in the adaptive block boundary analysis step to decide whether a blocking artifact exists.

\[
S = \sum_{m=0}^{7} \sum_{n=0}^{7} |C(m,n)| - |C(0,0)|
\]

**Figure 3.3** Block Classification Algorithm. C represents the DCT coefficients of the compressed block. S is the sum of absolute values of the DCT coefficients, excluding the DC coefficient. 

29
Figure 3.4 Sample output of the block classification algorithm. (a) Lena image compressed at 0.37 bpp. (b) Output of block classification. Dark blocks represent smooth blocks, bright blocks represent non-smooth blocks.

3.2.2 Adaptive Block Boundary Analysis

Different block boundary types have different blocking artifact characteristics. For example, while the ss-boundary has a sharp discontinuity at the block boundary with relatively smoothly varying interior pixels, the sn-boundary has unsteady discontinuities at the block boundary with smoothly varying pixels on the smooth side and rapidly changing pixels on the non-smooth side. The nn-boundary has quickly changing pixels on both sides of the block boundary. The different block boundary types also have different perceptual characteristics. For example, ss- and sn-boundaries tend to be more disturbing relative to the nn-boundaries. These observations can be seen in Figure 3.5 and have led to adaptive processing of block boundaries. The goal of each different processing is the same: to decide whether a blocking artifact is visible on that boundary. The processing for each type of block boundary is presented in the following three subsections.
3.2.2.1 SS-Boundary Analysis

The essence of the processing in ss-boundary analysis is to compare the difference of pixels across the block boundary against the difference of pixels across the inner pixels. This is performed on the pixels of the compressed image. The exact algorithm is given in the following four steps for horizontal block boundaries (extension to vertical block boundaries is made by exchanging all columns by rows and all rows by columns):

**Step-1:** Process each of the eight columns of pixels (each column consists of six pixels; three pixels on one side, three on the other side) across the block boundary as follows. First, calculate successive differences vertically. Second, take absolute values of entries. Change entries which are 0 or 1 to 2, except for the middle entry. Third, divide the middle entry by each of the other entries and sum up the ratios. Call the sum \( \text{sum}(i) \). See Figure 3.6 for a graphical illustration of this step.

**Step-2:** Combine the results of column processing by summing up squares of \( \text{sum}(i) \), as shown below. Call this sum \( \text{BlockingArtifactMeasure}_1 \).

\[
\text{BlockingArtifactMeasure}_1 = \sum_{i=1}^{8} (\text{sum}(i))^2
\]
Step-3: Scale BlockingArtifactMeasure1 by checking the types of the surrounding blocks to obtain BlockingArtifactMeasure2, as shown below. Block types are either smooth or non-smooth, as determined by the block classification stage. A look-up table is used to determine the scaling factor. See Figure 3.7 for a graphical illustration of this step.

\[ \text{BlockingArtifactMeasure2} = \text{ScalingFactor} \times \text{BlockingArtifactMeasure1} \]

Step-4: Compare BlockingArtifactMeasure2 against a threshold (Threshold2). If threshold is exceeded, blocking artifact exists, otherwise it does not exist. See Figure 3.8 for a graphical illustration of this step.
For each column (length 6; 3 pixels on one side 3 on the other side) in the compressed blocks, do the followings:

1. Calculate successive differences vertically.
2. Take absolute values of entries. Change entries which are 0 or 1 to 2, except for the middle entry.
3. Divide the middle entry by each of the other entries and sum up the ratios. Call the sum $\text{sum}(i)$.

**Intuition:**
The higher the ratios, the higher the Blocking Artifact

Figure 3.6 Graphical illustration of Step-1 of ss-boundary analysis

Figure 3.7 Graphical illustration of Step-3 of ss-boundary analysis
The intuition in the first step is that the higher the ratios, the higher the blocking artifact. If the difference across the block boundary is about the same as the difference across the inner pixels (which means that the block boundary has similar characteristics as the inner regions of the blocks), then we will have smaller ratios leading to a small sum(i). If, however, we have high ratios then that means that the block boundary has relatively higher changes than the inner pixels, leading to a large sum(i) and therefore implying the existence of a blocking artifact. Entries which are 0 or 1 are changed to 2 because such entries in the denominator cause the ratios to become too high, causing misleading sums.

In Step-2, rather than summing sum(i) directly, we sum the squares of sum(i) because we want to put increasingly more weight on large ratios. This is in conformance with the fact that the disturbance the eye perceives from the discontinuity at the block boundary increases faster, with the discontinuity strength, than at a linear rate [19]. In Step 3, we apply a scaling to BlockingArtifactMeasure1 to account for the effects of the surrounding blocks. In particular, the block boundary discontinuity appears more disturbing in regions where all the surrounding blocks are smooth blocks, relative to regions where many of the surrounding blocks are non-smooth blocks. Step-4 finalizes the ss-boundary analysis with a comparison against a threshold (Threshold2).
3.2.2.2 SN-Boundary Analysis

On an sn-boundary, a smooth and a non-smooth block come together. This happens mostly if we have a sharp edge on a smooth region. The non-smooth block contains the edge while the smooth block contains a portion of the smooth region, for example background. The non-smooth block containing the edge exhibits ringing artifacts, in other words artificial fluctuations. When these fluctuations come together with a smooth block, a very annoying picture results, as shown in Figure 3.5. In the original image, the smoothness continues until the edge, while in the compressed image, the smoothness stops at the block boundary and an abrupt artificial fluctuation starts there going up to the edge.

The essence of the processing here is to compare the behavior of pixel differences in the reconstructed image against the behavior of pixel differences in the original image. The pixel differences in the compressed image will quantify the strength of the artificial fluctuations. The exact algorithm is again given in four steps for horizontal block boundaries (extension to vertical block boundaries is made by exchanging all columns by rows and all rows by columns):

**Step-1:** Process each of the eight columns of pixels (each column consists of five pixels; one pixel on the smooth side, 4 pixels on the non-smooth side) across the block boundary, both in the compressed and original blocks, as follows. First, calculate successive differences vertically. Second, take absolute values of entries. Change entries which are 0 or 1 to 2, except for the middle entry. Third, divide the resulting entries from the compressed image by the entries from the original image and sum up the ratios. Call the sum sum(i). See Figure 3.9 for a graphical illustration of this step.

**Step-2:** Combine the results of column processing by summing up squares of sum(i), as shown below. Call this sum BlockingArtifactMeasure1.

$$BlockingArtifactMeasure1 = \sum_{i=1}^{8} (sum(i))^2$$
Step-3: Scale \( \text{BlockingArtifactMeasure1} \) by checking the types of the surrounding blocks to obtain \( \text{BlockingArtifactMeasure2} \), as shown below. Block types are either smooth or non-smooth, as determined by the block classification stage. A look-up table is used to determine the scaling factor. See Figure 3.10 for a graphical illustration of this step.

\[
\text{BlockingArtifactMeasure2} = \text{ScalingFactor} \times \text{BlockingArtifactMeasure1}
\]

Step-4: Compare \( \text{BlockingArtifactMeasure2} \) against a threshold (Threshold3). If threshold is exceeded, blocking artifact exists, otherwise it does not exist. See Figure 3.11 for a graphical illustration of this step.
1. Calculate successive differences vertically.

2. Take absolute values of entries. Change entries which are between 0 and 1, to 2; entries from the original image only.

3. Divide the resulting entries from the compressed image by the entries from the original image and sum up the ratios. Call the sum $\text{sum}(i)$.

Intuition:
The higher the ratios, the higher the Blocking Artifact

Figure 3.9 Graphical illustration of Step-1 of sn-boundary analysis
The intuition in the first step is that the higher the ratios, the higher the blocking artifact. If the differences are high in the compressed image while they are low in the original image, then that implies the existence of an artificial fluctuation and thus the existence of a blocking artifact. However, if they are high both in the compressed and original image, then there is no greatly annoying structure because nothing is perceived as artificially generated. Again, entries from the original image which are 0 or 1 are
changed to 2 because such entries in the denominator cause the ratios to become too high, causing misleading sums.

Steps 2, 3, and 4 are similar to the corresponding steps in the ss- boundary analysis. In Step-2, we again sum the squares of sum(i) because we want to put increasingly more weight on larger ratios. In Step-3, we apply a scaling to BlockingArtifactMeasure1 to account for the effect of the surrounding blocks. In particular, the block boundary discontinuity appears more disturbing in regions where all the surrounding blocks of the smooth block are smooth blocks, relative to regions where many of the surrounding blocks of the smooth block are non-smooth blocks. Step-4 finalizes the ss-boundary analysis with a comparison against a threshold (Threshold3).

### 3.2.2.3 NN-Boundary Analysis

Non-smooth blocks are blocks whose high-frequency content is above a threshold. When two non-smooth blocks come together, a possible blocking artifact is often not perceived as very disturbing. The blocking artifact is masked by the existing high-frequency content surrounding the block boundary [4]. Furthermore, even slight filtering, will cause blurring of some texture elements. This is undesirable. A blurred texture is not preferred over a sharp texture by the eye even if the sharp texture shows small clues of blocking artifacts. For these reasons, we conclude, without any processing, that all nn-boundaries are free of blocking artifacts. Finally, we present two sample outputs for the blocking artifact detection system described in Section 3.2 in Figure 3.12.
Figure 3.12 Blocking Artifacts detected on the Cameraman and Lena images. The left column shows the compressed images. The right column shows the block boundaries with blocking artifacts with white stripes. (a) The Cameraman image (b) The Lena image
3.3 Filter Selection

Filter selection refers to identifying the filter, from the predefined filter set, that best reduces the blocking artifact on the boundary of interest. It is performed only for those boundaries that are detected by the blocking artifact detection step. The MSE criterion is used for the selection. In particular, the block boundary is filtered with every filter in the predefined set, and then the filter resulting in the minimum MSE is chosen. In the MSE calculation, the two blocks on each side of the block boundary of interest are used.

The design of the filter set is important for the effectiveness of the deblocking performance. The design poses two questions: How many filters should be included in the filter set? Which filters should be included in the filter set? These questions have no definite answers. However, it is clear that including more filters will improve the deblocking performance while it will also increase the bit-rate of the side information. The filters that are included should be able to cover the greatly varying characteristics of blocking artifacts. They should also be complementary in order to keep the filter set small. For example, while one filter is very good at removing blocking artifacts in smooth regions, another should be very good in texture regions and another in edge regions.

The filter set that we have used in our experiments is composed of five filters. It contains one filter aimed for smooth regions, two filters for diagonal-edge regions, and two filters for edge regions with edges perpendicular to the block boundary. The filters were designed based on empirical observations of the characteristics of blocking artifacts. No rigorous study is made to choose optimal filters because the focus of this thesis is to demonstrate the benefits of reducing blocking artifacts based on side information. For the exact specifications of the filters that were used in the experiments, see Appendix A.

3.4 Coding of Side Information

The decoder must be informed about the block boundaries that exhibit blocking artifacts as well as the filters that were selected for these boundaries. This information is called side information in this thesis. To represent the side information with the least number of
bits, statistical redundancy in it is exploited. A coding system that is similar to the well-known run-length encoding is used.

To code the side information, we use a symbol set that contains six symbols: five symbols for filters used for deblocking and one symbol representing the lack of blocking artifact. Each block-boundary in the image is assigned one of these symbols. If the blocking artifact detection step determines that the boundary of interest does not exhibit a blocking artifact, the symbol representing lack of blocking artifact is assigned to that boundary. If, however, it does exhibit a blocking artifact, then the symbol for the filter that is selected by the filter selection step is assigned.

In the experiments, it is observed that 60 to 85 percent (depending on the compression ratio and image characteristics) of all block boundaries in images are assigned the symbol that represents the lack of blocking artifact. This observation has led us to employ run-length encoding where the symbol representing lack of blocking artifact plays the role of the zero. First, symbols of all horizontal block boundaries are run-length encoded following the scan order in Figure 3.13(a). Then, symbols of vertical block boundaries are run-length encoded following the scan order in Figure 3.13(b). A maximum run-length of 11 is chosen. Codewords for the employed run-length encoding

---

**Figure 3.13** Scan orders of block boundaries for run-length encoding. (a) Scan order for horizontal block boundaries. (b) Scan order for vertical block boundaries.
are constructed from probability estimations derived from sample images. It is observed that the employed run-length encoding system performs close to the entropy, which is also calculated from the estimated probabilities. The details of the employed run-length encoding system are given in Appendix B.

It is observed in our experiments that at very low bit-rates, such as 0.25 bpp, every block boundary exhibits visible blocking artifact. In such cases, rather than leaving block boundaries without blocking artifacts untouched, one might consider applying a very weak low-pass filter on only the pixels adjacent to such block boundaries. We have observed that this choice not only improves the visual quality but also the PSNR. Therefore, in our experiments, we have applied a very weak low-pass filter (Filter 5 in Appendix A) to block boundaries without blocking artifacts (i.e. block boundaries that were identified not to have blocking artifacts by the blocking artifact detection step). Note that this choice does not require any modification to the system except that block boundaries without blocking artifacts are filtered with a low-pass filter rather than being left untouched.
Chapter 4
Adaptation Of The System To Video Applications

This chapter describes the reduction of blocking artifacts using side information in video. Since a sequence of frames is displayed in video, blocking artifact reduction is applied to every frame in the sequence. However, blocking artifacts in video have one difference from blocking artifacts in images. While blocking artifacts in images occur always at block boundaries, blocking artifacts in video can also occur in the interiors of blocks, as shown in Figure 4.1. This is due to motion compensated coding of video.

The bit-rate of the side information of video depends on the frame rate in the encoded video. If, for example, 30 frames are encoded per second, then the bit-rate of the side information (in bits per second) for video is expected to be about 30 times more than

Figure 4.1 Frame 80 of the Foreman sequence. Frame 80 is a P-frame. Blocking artifacts that do not occur on block boundaries can be seen around the eyes.
the bit-rate of the side information for an image. That is, the side information bit-rate increases linearly with the frame rate of the video. However, the bit-rate of the video itself does not increase linearly with the frame rate because of temporal redundancy. Hence, the ratio of side information bits over regular encoding bits will be larger for video than for images. This is undesirable since it indicates a performance loss for reducing blocking artifacts in video based on side information. However, if blocking artifact reduction is performed in the motion compensation loop, or in other words, on the reference frame that is used for motion compensation, then the bit-rate of the side information does not increase linearly with the frame rate. This is due to the fact that stationary regions of a frame do not show blocking artifacts because they are taken from the reference frame that has reduced blocking artifacts since it was deblocked. Hence, the motion compensation loop not only removes the temporal redundancy in the video but also the temporal redundancy in the side information. This way, reduction of blocking artifacts using side information becomes competitive for video as it is for images.

The deblocking method for images described in Chapter 3 needs to be modified to account for the fact that blocking artifacts propagate into the interiors of the blocks in video frames. Modification is only made to the blocking artifact detection step. Filter selection and side information coding steps remain unchanged.

In the blocking artifact detection step for images, it is assumed that the blocking artifact is at the block boundary and the processing for deciding on the visibility of the blocking artifact is performed accordingly. Therefore, in the case of video, first, the location of the artificial discontinuity is estimated and then the processing for deciding on the visibility of the blocking artifact is done as if the block boundary was at this estimated location. The estimation is done by, first, finding the largest discontinuity location of each column across the block boundary and then taking the median of these locations. The filter selection and side information coding steps remain unchanged. In other words, filtering is performed on the same pixels (shown in Appendix A) as for images; i.e. the pixels that are modified are not chosen according to the estimated discontinuity location.
Chapter 5
Experimental Results

This chapter presents the results that are obtained using the blocking artifact reduction system based on side information described in Chapters 3 and 4. These results are intended to demonstrate the gains that can be obtained (and the conditions under which they can be obtained) from using side information for deblocking. It is worth noting that the particular system described in Chapters 3 and 4 is just one of many possible systems to reduce blocking artifacts based on side information. Different systems based on side information are possible and will achieve different results.

To evaluate the quality of the images and videos, we use PSNR, which is described in Chapter 1. However, since PSNR is not directly correlated with the quality the human visual system perceives, visual appearances of the images are also taken into account in the final evaluation. First, we present results obtained from images, and then results obtained from video sequences.

5.1 Experimental Results for Images

The image coder that is used in the experiments is the implementation of a typical image coder that is described in Chapter 2. It uses the quantization table that is given in Annex-K of [1]. This quantization table is scaled linearly to change the amount of compression and kept constant for all blocks in the entire image. Entropy coding is performed as described in Chapter 2. The VLCs that are used in the Entropy coding stage are the Huffman codes given in Annex-K of [1]. It is worth noting that the particular implementation of the block-based image coder is not important because the deblocking method based on side information is independent of any particular implementation of the image coder.
We use the following parameters for the deblocking method based on side information that is described in Chapter 3: Threshold1 = 105, Threshold2 = 760, Threshold3 = 900. These thresholds were determined based on empirical observations of the performance of the method. The filter set that is used in the filter selection step is given in Appendix A. Appendix B presents the Huffman code table that is used to code the side information. These Huffman codes are constructed using statistics obtained from 13 images (shown in Appendix C), which do not include the test images used in the comparison experiments. Each of these 13 images was used to produce statistics at 6 different coding bit-rates; 0.2 bpp, 0.25 bpp, 0.3 bpp, 0.35 bpp, 0.4 bpp and 0.45 bpp. The resulting 78 different statistics were averaged to obtain one average statistics from which one set of Huffman codes is constructed. This set of Huffman codes is used in all image experiments. One can see from the construction process of the Huffman codes that this set of Huffman codes is tailored to the range of coding bit-rates of 0.2-0.45 bpp, and not to a particular bit-rate in that range.

The deblocking system based on side information that is described in Chapter 3 is compared against the non-processed compressed image and the deblocking method based on POCS presented in [15]. The comparisons are made such that the overall bit-rates of all methods are equal. In other words, in the method based on side information, the images are compressed at a lower bit-rate so that the target bit-rate is achieved with the addition of the side information bits. The non-processed compressed image is included in the comparisons because we want to show for which range of bit-rates the side information based deblocking method makes sense. The method in [15] is chosen because it is one of the most cited deblocking methods in the literature. The method is based on POCS theory. It uses two convex sets: one convex set that captures the fidelity to the quantized DCT coefficients, and another convex set that captures the smoothness across block boundaries. We use five iterations and three segments (as in the experiments in [15]) for the simplified spatially adaptive version of this method. For the details of this method, see [15].

The comparisons of the deblocking methods in this thesis are made using the following six images: Lena, Cameraman, Goldhill, Tree, Sailboat and Airplane. The Sailboat and Airplane images have resolution of 512x512 pixels; the remaining images
Figure 5.1 The 256x256 pixel images that are used in the experiments. (a) The Lena image (b) The Cameraman image (c) The Goldhill image (d) The Tree image.
Figure 5.2 The Sailboat image that is used in the experiments. Resolution is 512x512 pixels.
Figure 5.3 The Airplane image that is used in the experiments. Resolution is 512x512 pixels.
Table-5.1 presents the PSNR values of the test images for the non-processed compressed image, for the method given in [15], and for the proposed method based on side information. Results are given at different bit-rates ranging from 0.2 bpp to 0.45 bpp. The same data are plotted in Figures 5.4 to 5.6. As can be seen from both Table 5.1 and Figures 5.4 to 5.6, the deblocking method based on side information gives PSNR results that are often worst at very low bit-rates such as 0.2 bpp and 0.25 bpp, and best at higher bit-rates such as 0.35 bpp to 0.45 bpp. At very low bit-rates, the use of a portion of the bit-rate (about 20% in the case of 0.25 bpp) to represent side information, rather than finer quantization of the DCT coefficients, results in poorer performance. This is expected since at such low bit-rates, the rate-distortion curve of the image compression system is so flat that very small increments in the bit-rate result in large distortion reductions. These distortion reductions cannot be matched by the system based on side information. However, if we go to higher bit-rates, such as 0.35 bpp to 0.45 bpp, the situation changes. At higher bit-rates, the deblocking system based on side information provides, for almost all pictures, better PSNR results than the non-processed compressed image. At 0.4 bpp and 0.45 bpp, the PSNR results of the system based on side information are better than the results of the method in [15] for all images.
Table 5.1 Comparison chart of PSNR values of the non-processed compressed image, the method in [15], and the method based on side information at different bit-rates.

<table>
<thead>
<tr>
<th>bpp</th>
<th>Lena</th>
<th>Cameraman</th>
<th>Goldhill</th>
<th>Tree</th>
<th>Sailboat</th>
<th>Airplane</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.2</td>
<td>Non-processed compressed image</td>
<td>24.55</td>
<td>24.54</td>
<td>24.65</td>
<td>22.13</td>
<td>24.98</td>
</tr>
<tr>
<td></td>
<td>Image produced by the proposed method</td>
<td>24.58</td>
<td>24.55</td>
<td>24.72</td>
<td>21.97</td>
<td>25.15</td>
</tr>
<tr>
<td>0.25</td>
<td>Non-processed compressed image</td>
<td>26.07</td>
<td>25.74</td>
<td>25.53</td>
<td>23.52</td>
<td>26.47</td>
</tr>
<tr>
<td></td>
<td>Image produced by the proposed method</td>
<td>26.20</td>
<td>25.69</td>
<td>25.72</td>
<td>23.89</td>
<td>26.66</td>
</tr>
<tr>
<td>0.3</td>
<td>Non-processed compressed image</td>
<td>27.12</td>
<td>26.64</td>
<td>26.22</td>
<td>24.70</td>
<td>27.57</td>
</tr>
<tr>
<td></td>
<td>Image produced by the proposed method</td>
<td>27.33</td>
<td>26.67</td>
<td>26.43</td>
<td>24.96</td>
<td>27.81</td>
</tr>
<tr>
<td>0.35</td>
<td>Non-processed compressed image</td>
<td>28.06</td>
<td>27.34</td>
<td>26.74</td>
<td>25.60</td>
<td>28.39</td>
</tr>
<tr>
<td></td>
<td>Image produced by the proposed method</td>
<td>28.27</td>
<td>27.45</td>
<td>26.96</td>
<td>25.82</td>
<td>28.66</td>
</tr>
<tr>
<td>0.4</td>
<td>Non-processed compressed image</td>
<td>28.77</td>
<td>27.94</td>
<td>27.20</td>
<td>26.29</td>
<td>29.08</td>
</tr>
<tr>
<td></td>
<td>Image produced by the method in [15]</td>
<td>28.89</td>
<td>27.80</td>
<td>27.38</td>
<td>26.55</td>
<td>29.22</td>
</tr>
<tr>
<td></td>
<td>Image produced by the proposed method</td>
<td>29.02</td>
<td>28.08</td>
<td>27.42</td>
<td>26.63</td>
<td>29.29</td>
</tr>
<tr>
<td>0.45</td>
<td>Non-processed compressed image</td>
<td>29.37</td>
<td>28.50</td>
<td>27.59</td>
<td>26.86</td>
<td>29.62</td>
</tr>
<tr>
<td></td>
<td>Image produced by the method in [15]</td>
<td>29.38</td>
<td>28.24</td>
<td>27.70</td>
<td>27.02</td>
<td>29.66</td>
</tr>
<tr>
<td></td>
<td>Image produced by the proposed method</td>
<td>29.66</td>
<td>28.59</td>
<td>27.81</td>
<td>27.15</td>
<td>29.88</td>
</tr>
</tbody>
</table>
Figure 5.4  (a) PSNR vs bit-rate for the Lena image (256x256 pixels)  (b) PSNR vs bit-rate for the Cameraman image (256x256 pixels).
Figure 5.5 (a) PSNR vs bit-rate for the Goldhill image (256x256 pixels) (b) PSNR vs bit-rate for the Tree image (256x256 pixels).
Figure 5.6  (a) PSNR vs bit-rate for the Sailboat image (512x512 pixels)  (b) PSNR vs bit-rate for the Airplane image (512x512 pixels).
Figure 5.7 Plot of side information bit-rate vs overall coding bit-rate for the Lena image.

The change of the bit-rate of the side information with the overall coding bit-rate is plotted in Figure 5.7 for the Lena image. In general, the bit-rate of the side information decreases with increasing overall bit-rate because higher bit-rates lead to higher-quality images that have fewer blocking artifacts. However, exceptions may occur, as in the case of 0.3 bpp in Figure 5.7. For coding bit-rates of 0.2 bpp to 0.45 bpp, it is observed that the bit-rate of the side information varies from 0.02 bpp to 0.045 bpp, depending on the severeness of the blocking artifacts in the image of interest.

The improvement in visual quality is another way to evaluate the results of the experiments. Indeed, it is well known that PSNR values do not correlate very well with the quality of pictures that the human observer perceives. Therefore, we provide visual results for some of the test images. We provide visual results for the Lena image (Figures 5.8-5.12), for the Cameraman image (Figures 5.13-5.14), and for the Goldhill image (Figure 5.15). Figures 5.8 to 5.12 show the experimental results for the Lena image coded at bit-rates ranging from 0.25 bpp to 0.45 bpp. In Figure 5.9, where comparisons are made at 0.3 bpp, the system based on side information results in a better looking image.
than the method in [15], even though the system based on side information has a lower PSNR value. In particular, if one considers the shoulder of Lena and the border of the shoulder with the mirror, one can see that the shoulder looks smoother, and the border looks cleaner in the image produced by the system based on side information. Another region one might consider is the outline of Lena’s hat against the background. It is easy to see that this region has visible ringing artifacts in the picture created using the method in [15]. This region is much cleaner in the picture produced by the system based on side information. The eyes in the picture created by the method in [15] look more blurry than in the picture produced by the system based on side information. Hence, the performance of the system based on side information is clearly better in edge regions. This is due to the use of the original image. Similar arguments can be made for the Lena images coded at higher bit-rates.

Figures 5.13 and 5.14 show the Cameraman image coded at 0.25 bpp and 0.35 bpp, respectively. For these images, the superiority of the side information based system can easily be seen as it effectively reduces blocking artifacts in the two most sensitive regions: the smooth background and sharp edges. The system based on side information successfully reduces the highly annoying blocking artifacts in the background region. It also cleans artifacts around the borderline of the cameraman with the background.

Figure 5.15 shows the Goldhill image coded at 0.3 bpp. The images deblocked by the system based on side information and the method in [15] do not have many distinguishing features because the Goldhill image contains many high detailed regions. In such regions, even slight amount of filtering can blur the details. Therefore, neither blocking artifact reduction method changes many pixel values in these regions, where blocking artifacts are also partially masked.

In summary, for very low bit-rates such as 0.2 bpp or 0.25 bpp, the system based on side information is, in most cases, worse than the method in [15], while it may be preferable to the non-processed image. However, at bit-rates above 0.25 bpp, the deblocking system based on side information produces images that are visually better than the method in [15] and the non-processed image. This superior performance, however, is verified by PSNR values only for bit-rates such as 0.35 bpp, 0.4 bpp or 0.45 bpp.
Figure 5.8 Comparison of the Lena image coded at 0.25 bpp
Figure 5.9 Comparison of the Lena image coded at 0.3 bpp

(a) Non-processed
PSNR = 27.12 dB

(b) Method in [15]
PSNR = 27.53 dB

(c) Proposed method
PSNR = 27.33 dB
(a) Non-processed
PSNR = 28.06 dB

(b) Method in [15]
PSNR = 28.28 dB

(c) Proposed method
PSNR = 28.27 dB

Figure 5.10 Comparison of the Lena image coded at 0.35 bpp
Figure 5.11 Comparison of the Lena image coded at 0.4 bpp

(a) Non-processed
PSNR = 28.77 dB

(b) Method in [15]
PSNR = 28.89 dB

(c) Proposed method
PSNR = 29.02 dB
Figure 5.12 Comparison of the Lena image coded at 0.45 bpp
Figure 5.13 Comparison of the Cameraman image coded at 0.25 bpp

(a) Non-processed
PSNR = 25.74 dB

(b) Method in [15]
PSNR = 25.85 dB

(c) Proposed method
PSNR = 25.69 dB
Figure 5.14 Comparison of the Cameraman image coded at 0.35 bpp

(a) Non-processed
PSNR = 27.34 dB

(b) Method in [15]
PSNR = 27.31 dB

(c) Proposed method
PSNR = 27.45 dB
(a) Non-processed
PSNR = 26.22 dB

(b) Method in [15]
PSNR = 26.51 dB

(c) Proposed method
PSNR = 26.51 dB

Figure 5.15 Comparison of the Goldhill image coded at 0.30 bpp
5.2 Experimental Results for Video Sequences

The MPEG2 video encoder of the MPEG Software Simulation Group [22] is used for the experiments for video sequences. The regular GOP structure, IBBPBPPBBPBBPBB, and the rate control algorithm in the software are used. The blocking artifact reduction system based on side information described in Chapter 3, along with the modifications explained in Chapter 4, is integrated into the video codec. The system is included both at the decoder and the encoder. At the encoder, the system is integrated into the motion compensation loop. In other words, the reference frame is deblocked. At the decoder, the system deblocks the frame that is sent to output and at the same time used as reference for future frames. These systems operate only on the luminance component in our experiments as the luminance component is perceptually more important. The chrominance component is not modified.

We use the following parameters for the deblocking method based on side information, which is described in Chapters 3 and 4: $\text{Threshold}1 = 205$, $\text{Threshold}2 = 760$, $\text{Threshold}3 = 900$. Note that while $\text{Threshold}2$ and $\text{Threshold}3$ are the same as in the experiments for images, $\text{Threshold}1$ is increased from 105 to 205. This increase is made to account for the fact that in video sequences, the blocking artifacts propagate into the interiors of the blocks. Hence, artificial discontinuities exist in the interiors of blocks, causing the high-frequency content of the “smooth” blocks to increase. Since $\text{Threshold}1$ is used to classify blocks as smooth or non-smooth blocks, it is increased form 105 to 205.

The filter set that is used in the filter selection step is the same as in the experiments for the images and is given in Appendix A. Appendix B presents the two Huffman code tables that are used to code the side information in the video experiments. The first set of Huffman codes is used to code the side information of I-frames. This set of Huffman codes is the same as the set of Huffman codes used for the image experiments. The construction of this set is explained in Section 5.1. The second set of Huffman codes is used to code the side information of P- and B-frames. P- and B-frames have much fewer blocking artifacts than I-frames because P- and B-frames are coded using other I- and P-frames that are already deblocked. Therefore, we have decided to use
a second set of Huffman codes that is more effective in compressing the side information of P- and B-frames. This set of Huffman codes is constructed using statistics obtained from P- and B-frames of two video sequences that are shown in Appendix C. They do not include the test sequences used in this section. Each of these two sequences was used to produce statistics at two coding bit-rates, 200 Kb/s and 300 Kb/s. Since the two video sequences have 300 frames each, about 1120 \((=(14/15)*300*2*2\) pieces of statistics were obtained. The average of these statistics was used to compute the set of Huffman codes that was used to code the side information of the P- and B-frames of the video sequences used in the comparison experiments.

The blocking artifact reduction system based on side information is compared against a system that does not employ any blocking artifact reduction and a system that employs the deblocking method presented in [8]. The comparisons are made such that the overall bit-rates of all methods are equal. In other words, in the system based on side information, the video sequences are compressed at a lower bit-rate so that the target bit-rate is achieved with the addition of the side information bits. The method in [8] is chosen for comparison purposes because it is a typical example of the deblocking methods that are based on adaptive filtering. In addition, [8] is published very recently. Adaptive filtering methods are the most common choice for deblocking video sequences because of their computational simplicity relative to other methods, as explained in Chapter 2. The method in [8] is integrated into the video codec in a similar manner to the method based on side information. In other words, it is included both at the encoder (in the motion compensation loop) and at the decoder.

The comparisons of the deblocking method for video sequences are performed using the Foreman sequence and the Silent Voice sequence. Both of these sequences are progressively scanned, have 300 frames and CIF (288 rows x 352 columns) resolution, and are coded at a frame rate of 25 fps. The first frame of both sequences is shown in Figure 5.16.

Table 5.2 provides the average PSNR values for all experiments performed for the deblocking of video sequences. Table 5.2 shows that, in general, the system based on side information performs slightly better at higher bit-rates than at lower bit-rates. This is consistent with the results reported in Section 5.1.
Figure 5.16  Test sequences. (a) The Foreman sequence. (b) The Silent Voice sequence.

Table 5.2  Comparison of the average PSNR values of the test sequences for the standard encoder, the encoder with deblocking method in [8], and the encoder with deblocking method based on side information.

<table>
<thead>
<tr>
<th>PSNR (dB)</th>
<th>Standard Encoder</th>
<th>Encoder with deblocking method in [8]</th>
<th>Encoder with deblocking method based on side information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Foreman 292 Kb/s</td>
<td>28.42</td>
<td>28.48</td>
<td>28.35</td>
</tr>
<tr>
<td>Foreman 369 Kb/s</td>
<td>29.25</td>
<td>29.22</td>
<td>29.35</td>
</tr>
<tr>
<td>Silent Voice 210 Kb/s</td>
<td>29.28</td>
<td>29.34</td>
<td>29.49</td>
</tr>
<tr>
<td>Silent Voice 293 Kb/s</td>
<td>31.30</td>
<td>31.27</td>
<td>31.47</td>
</tr>
</tbody>
</table>

Figure 5.17 shows the PSNR values for each frame of the Foreman sequence. In Figure 5.17(a), the sequence is coded at an overall bit-rate of 292 Kb/s. The system based on side information allocates 32 Kb/s of 292 Kb/s to the side information. Similarly,
Figure 5.17(b) compares the three approaches at a bit-rate of 369 Kb/s. The side information occupies 31 Kb/s of 369 Kb/s in this case. It is seen that the system based on side information produces frames with lower PSNR values than the other two systems up to frame 150. After frame 150, however, the performance of the system based on side information increases significantly. This occurs because scenes with fast motion appear after frame 150. For such scenes, the standard encoder suffers from highly disturbing blocking artifacts. The system based on side information, however, effectively reduces blocking artifacts for scenes with high motion because it detects all blocking artifacts and reduces them effectively by making use of the original image. The method in [8], however, does not appear to be as successful as the system based on side information for such scenes. One reason for this observation may be given as follows. For scenes with high motion, even block interiors have artifacts (which have propagated there from previous frames) and therefore the block interiors show high-frequency characteristics. For regions with such characteristics, the method in [8] chooses the complex mode that hardly changes pixel intensities for means of not blurring details. However, the system based on side information can perform better for such regions because it uses the original image. As a result, after frame 150, the side information based system not only produces frames with higher PSNR but also frames with clearly fewer blocking artifacts and higher visual quality than the other two systems, as shown in Figure 5.20.

The average PSNR values in Figure 5.17(a) are 28.42 dB for the standard encoder, 28.48 dB for the encoder with deblocking method in [8], and 28.35 dB for the encoder with deblocking method based on side information. In Figure 5.17(b), the average PSNR values for the standard encoder, the encoder with deblocking method in [8], and the encoder with deblocking method based on side information are 29.25 dB, 29.22 dB and 29.35 dB, respectively. Hence, the performance of the side information based system increases with increasing bit-rate. This increase is consistent with the results reported in Section 5.1, where the side information based system also performed relatively better for higher bit-rates.

Figure 5.18 shows the PSNR values for each frame of the Silent Voice sequence. In Figure 5.18(a), the sequence is coded at 210 Kb/s. The side information based system allocates 20 Kb/s of 210 Kb/s for the side information. In Figure 5.18(b), the sequence is
coded at 293 Kb/s, and the side information based system uses 17 Kb/s for the side information. The average PSNR values in Figure 5.14(a) are 29.28 dB for the standard encoder, 29.34 dB for the encoder with deblocking method in [8], and 29.49 dB for the encoder with deblocking method based on side information. In Figure 5.18(b), the average PSNR values for the standard encoder, the encoder with deblocking method in [8], and the encoder with deblocking method based on side information are 31.30 dB, 31.27 dB and 31.47 dB, respectively.

The behavior of the bit-rate of the side information with frame number is plotted for the Foreman sequence in Figure 5.19. Figure 5.19(a) is plotted for the encoding bit-rate of 292 Kb/s and Figure 5.19(b) for the encoding bit-rate of 369 Kb/s. The average bit-rate of the side information is 32 Kb/s and 31 Kb/s for the overall encoding bit-rate of 292 Kb/s and 369 Kb/s, respectively. It can be observed that the side information bit-rate has spikes. These spikes occur every 15\textsuperscript{th} frame because every 15\textsuperscript{th} frame is coded as an I-frame. I-frames have more blocking artifacts because they are intra coded. The inter-coded P-and B-frames have fewer blocking artifacts because they are coded using reference frame(s), whose blocking artifacts are already reduced. Between frames 160 and 220, the spikes are less prominent because of the fast motion in these frames. In scenes with fast motion, even P- and B-frames have many blocking artifacts because the reference frame cannot be effectively used in such cases. Therefore, the side information bit-rate increases for P- and B-frames between frames 160 and 220.

Visual quality of the deblocked frames is another way to interpret the results of the experiments. It is observed that for almost all images, the frames produced by the method in [8] and by the system based on side information have fewer blocking artifacts than the images produced by the standard encoder. If we compare the images produced by the method in [8] and the system based on side information, we can say that they do not have a great amount of distinguishing features for many frames. However, there are frames in the two test sequences for which the system based on side information is more successful while the opposite case is not observed. To provide the reader with examples, several frames from each video sequence are provided. Frames for which the system based on side information is more successful (Figure 5.20 and 5.22) and frames for which the two systems perform equally well are shown. Figure 5.20 shows comparisons of
frames from the Foreman sequence coded at 292 Kb/s, and Figure 5.21 shows comparisons of frames from the Foreman sequence coded at 369 Kb/s. In Figure 5.20, frame 13 generated by the standard encoder has highly disturbing blocking artifacts on Foreman’s face. While the method in [8] reduces the blocking artifacts on Foreman’s face to some extent, the method based on side information clearly produces the best image for this frame. Comparisons of frames from the Silent Voice sequence, which are given in Figures 5.22 and 5.23, also demonstrate the observation that the visual qualities of the frames produced by the system based on side information are not worse, if not better, than frames produced by the other two systems.

In summary, the experiments with video sequences demonstrate that the blocking artifact reduction system based on side information is successful for video applications. Both the average PSNR values and the visual qualities of the video sequences support this observation. It is worth noting the additional advantage: the decoder based on side information has low computational requirements. It only needs to decode the side information and perform corresponding filtering; no computation to extract information about the local characteristics of the block boundaries is necessary.
Figure 5.17 PSNR values for each frame of the Foreman sequence are plotted for the standard encoder, the encoder with deblocking method in [8], and the deblocking method based on side information. (a) The overall coding bit-rate is 292 Kb/s. The side information bit-rate is 32 Kb/s. (b) The overall coding bit-rate is 369 Kb/s. The side information bit-rate is 31 Kb/s.
Figure 5.18 PSNR values for each frame of the Silent Voice sequence are plotted for the standard encoder, the encoder with deblocking method in [8], and the deblocking method based on side information. (a) The overall coding bit-rate is 210 Kb/s. The side information bit-rate is 20 Kb/s. (b) The overall coding bit-rate is 293 Kb/s. The side information bit-rate is 17 Kb/s.
Figure 5.19 Bit-rate of side information for each frame of the Foreman sequence is plotted. (a) The overall coding bit-rate is 292 Kb/s. The average side information bit-rate is 32 Kb/s or equivalently 0.013 bpp. (b) The overall coding bit-rate is 369 Kb/s. The average side information bit-rate is 31 Kb/s or equivalently 0.0125 bpp.
Figure 5.20 Frame 13 and frame 180 of the Foreman sequence coded at overall bit-rate of 292 Kb/s are shown. The left column shows frame 13 and the right column shows frame 180. The top row shows the image produced by the standard encoder. The middle row shows the image produced by the encoder with deblocking method in [8]. The bottom row shows the image produced by the deblocking method based on side information.
Figure 5.21 Frame 93 and frame 147 of the Foreman sequence coded at an overall bit-rate of 369 Kb/s are shown. The left column shows frame 93 and the right column shows frame 147. The top row shows the image produced by the standard encoder. The middle row shows the image produced by the encoder with deblocking method in [8]. The bottom row shows the image produced by the deblocking method based on side information.
Frame 45 and frame 135 of the Silent Voice sequence coded at an overall bit-rate of 210 Kb/s are shown. The left column shows frame 45 and the right column shows frame 135. The top row shows the image produced by the standard encoder. The middle row shows the image produced by the encoder with deblocking method in [8]. The bottom row shows the image produced by the deblocking method based on side information.
Figure 5.23 Frame 15 and frame 205 of the Silent Voice sequence coded at an overall bit-rate of 293 Kb/s are shown. The left column shows frame 15 and the right column shows frame 205. The top row shows the image produced by the standard encoder. The middle row shows the image produced by the encoder with deblocking method in [8]. The bottom row shows the image produced by the deblocking method based on side information.
Chapter 6
Conclusions

6.1 Summary

This thesis proposes to use side information to reduce blocking artifacts. The use of side information implies that the reduction of blocking artifacts at the decoder is dictated by the encoder. In other words, the encoder transmits side information that tells the decoder how to perform deblocking. Using side information enables the use of the original image in deblocking, which improves performance. Furthermore, the computational burden at the decoder is reduced since the decoder only needs to do what the side information requires; no computation to extract information from the compressed image is necessary.

A specific blocking artifact reduction system based on side information is proposed. However, it is worth noting again that the purpose of this thesis is not to show the optimality of the particular system that is proposed. The purpose of this thesis is to demonstrate that the approach to use side information for deblocking has significant benefits and should be explored further in the future research for deblocking. An overview of the proposed system is as follows: The encoder determines block boundaries that exhibit blocking artifacts as well as filters (from a predefined set of filters) that best deblock these block boundaries. Then it transmits side information that conveys the determined block boundaries together with their selected filters to the decoder. The decoder uses the received side information to perform deblocking.

Experiments are carried out to evaluate the proposed system to reduce blocking artifacts based on side information both for images and video sequences. Experiments for images show that for overall bit-rates (side information bits + regular encoding bits) of 0.2 bpp and below, the system based on side information is not preferable. For bit-ranges above 0.3 bpp, the side information based system produces images that are most often visually preferable to both the non-processed compressed image and the image processed
by the method in [15]. This visual superiority is verified by PSNR results, for all test images, only for bit-rates of 0.4 bpp and 0.45 bpp. Experiments for video sequences show that the system based on side information produces video sequences that have clearly fewer blocking artifacts than the standard encoder at an equal overall bit-rate. These video sequences are also visually preferable to the video sequences produced by the standard encoder. Experiments for video sequences also demonstrate that the method based on side information produces frames that are either of similar quality or better than frames produced by the method in [8]. The system based on side information also outperforms the method in [8] in terms of PSNR for three experiments (out of four experiments) used in the experiments.

6.2 Future Research

One future research direction is to improve the complementary parts of the proposed system. For example, the filter set that is used in the experiments is not rigorously designed. A better set of filters can be designed. The blocking artifact detection algorithm can also be improved. A distortion measure of blocking artifacts in images (in the presence of the original image) is provided in [19]. However, in [19] a measure of blocking artifacts on the overall image is produced, while the proposed system requires the information whether each individual block boundary exhibits a visible blocking artifact. The successful determination of visible blocking artifacts is vital for the performance of the proposed system based on side information. The design of such a system is challenging and requires successful exploitation of the HVS characteristics.

During the design of the proposed system based on side information, computational aspects were not emphasized as the purpose of the thesis is to demonstrate the possible gains that are achievable with deblocking based on side information. Another future research direction is to lower the computational requirements of the proposed system while keeping its philosophy. The most computationally intensive part is the blocking artifact detection step since it includes many division and multiplication operations.
As mentioned previously, this thesis attempts to demonstrate the benefits that can be obtained from using side information for deblocking. To show the possible performance of such systems, one particular system is developed and used in the experiments. However, other systems that use side information for deblocking, which may also have significant differences from the proposed system, can be designed and may perform better. From a fundamental perspective, the goal is to design a system that provides maximum visual and PSNR improvements while using minimum bit-rate for the side information. The design of such a system is a challenging and an open problem.
Appendix

A. The Filter Set Used in the Experiments

**Filter 0:**
(aimed for removing ringing artifacts)

5x5-tap 2-D low-pass filter:

\[
\frac{1}{25} \begin{bmatrix}
1 & 1 & 1 & 1 & 1 \\
1 & 1 & 1 & 1 & 1 \\
1 & 1 & 1 & 1 & 1 \\
1 & 1 & 1 & 1 & 1 \\
1 & 1 & 1 & 1 & 1 \\
\end{bmatrix}
\]

However, any filter coefficient which falls outside of the bold region is set to zero. (Filter is normalized, too)

**Filter 1:**
(aimed for removing ringing artifacts)

5x5-tap 2-D low-pass filter:

\[
\frac{1}{25} \begin{bmatrix}
1 & 1 & 1 & 1 & 1 \\
1 & 1 & 1 & 1 & 1 \\
1 & 1 & 1 & 1 & 1 \\
1 & 1 & 1 & 1 & 1 \\
1 & 1 & 1 & 1 & 1 \\
\end{bmatrix}
\]

However, any filter coefficient which falls outside of the bold region is set to zero. (Filter is normalized, too)
**Filter 2:**  
(aimed for diagonal edge regions)

5x5-tap 2-D diagonal low-pass filter:

\[
\begin{bmatrix}
1 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 1 \\
\end{bmatrix}
\]

Bold squares denote the pixels that are modified.

**Filter 3:**  
(aimed for diagonal edge regions)

5x5-tap 2-D diagonal low-pass filter:

\[
(1/5) \begin{bmatrix}
0 & 0 & 0 & 1 \\
0 & 0 & 0 & 1 \\
0 & 0 & 1 & 0 \\
0 & 1 & 0 & 0 \\
1 & 0 & 0 & 0 \\
\end{bmatrix}
\]

Bold squares denote the pixels that are modified.
**Filter 4:**
(aimed for flat regions)

9-tap 1-D vertical low-pass filter:

\[(1/9)[1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1]\]

**Filter 5:**
(may be used at block boundaries that are not detected by the blocking artifact detection step; see Section 3.4)

Low-pass filter:

\[a = 0.85a + 0.15b\]
\[b = 0.15a + 0.85b\]
B. Run-length Encoding of Side Information

The employed run-length encoding uses a Huffman Code set with 62 symbols where each symbol is composed of two sub-symbols as follows:

Symbol = ( Symbol1, Symbol2 )

Symbol1 = Run-length of ‘No Blocking Artifact’ (No BA)
Symbol2 = Filter that terminates the run

The following table lists the lengths of the binary codes of the two Huffman Code sets that are used in the experiments. Set 1 is used to code the side information of images and I-frames in the video sequences. Set 2 is used to code the side information of P- and B-frames in the video sequences.

Table B.1 Huffman Code table for the coding of the side information

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Binary code length</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Set 1</td>
</tr>
<tr>
<td>(0, Filter 0)</td>
<td>6</td>
</tr>
<tr>
<td>(0, Filter 1)</td>
<td>6</td>
</tr>
<tr>
<td>(0, Filter 2)</td>
<td>5</td>
</tr>
<tr>
<td>(0, Filter 3)</td>
<td>6</td>
</tr>
<tr>
<td>(0, Filter 4)</td>
<td>3</td>
</tr>
<tr>
<td>(1, Filter 0)</td>
<td>7</td>
</tr>
<tr>
<td>(1, Filter 1)</td>
<td>6</td>
</tr>
<tr>
<td>(1, Filter 2)</td>
<td>6</td>
</tr>
<tr>
<td>(1, Filter 3)</td>
<td>7</td>
</tr>
<tr>
<td>(1, Filter 4)</td>
<td>4</td>
</tr>
<tr>
<td>(2, Filter 0)</td>
<td>7</td>
</tr>
<tr>
<td>(2, Filter 1)</td>
<td>7</td>
</tr>
<tr>
<td>(2, Filter 2)</td>
<td>7</td>
</tr>
<tr>
<td>(2, Filter 3)</td>
<td>7</td>
</tr>
<tr>
<td>(2, Filter 4)</td>
<td>4</td>
</tr>
<tr>
<td>(3, Filter 0)</td>
<td>8</td>
</tr>
<tr>
<td>(3, Filter 1)</td>
<td>7</td>
</tr>
<tr>
<td>(3, Filter 2)</td>
<td>7</td>
</tr>
<tr>
<td>(3, Filter 3)</td>
<td>8</td>
</tr>
<tr>
<td>(3, Filter 4)</td>
<td>4</td>
</tr>
<tr>
<td>(4, Filter 0)</td>
<td>8</td>
</tr>
<tr>
<td>(4, Filter 1)</td>
<td>8</td>
</tr>
<tr>
<td>(4, Filter 2)</td>
<td>7</td>
</tr>
<tr>
<td>(4, Filter 3)</td>
<td>8</td>
</tr>
<tr>
<td>(4, Filter 4)</td>
<td>5</td>
</tr>
<tr>
<td>(5, Filter 0)</td>
<td>8</td>
</tr>
<tr>
<td>(5, Filter 1)</td>
<td>8</td>
</tr>
<tr>
<td>(5, Filter 2)</td>
<td>8</td>
</tr>
<tr>
<td>(5, Filter 3)</td>
<td>8</td>
</tr>
<tr>
<td>(5, Filter 4)</td>
<td>5</td>
</tr>
<tr>
<td>(6, Filter 0)</td>
<td>8</td>
</tr>
<tr>
<td>(6, Filter 1)</td>
<td>8</td>
</tr>
<tr>
<td>(6, Filter 2)</td>
<td>8</td>
</tr>
<tr>
<td>(6, Filter 3)</td>
<td>9</td>
</tr>
<tr>
<td>(6, Filter 4)</td>
<td>5</td>
</tr>
<tr>
<td>(7, Filter 0)</td>
<td>8</td>
</tr>
<tr>
<td>(7, Filter 1)</td>
<td>9</td>
</tr>
<tr>
<td>(7, Filter 2)</td>
<td>8</td>
</tr>
<tr>
<td>(7, Filter 3)</td>
<td>9</td>
</tr>
<tr>
<td>(7, Filter 4)</td>
<td>6</td>
</tr>
<tr>
<td>(8, Filter 0)</td>
<td>9</td>
</tr>
<tr>
<td>(8, Filter 1)</td>
<td>9</td>
</tr>
<tr>
<td>(8, Filter 2)</td>
<td>8</td>
</tr>
<tr>
<td>(8, Filter 3)</td>
<td>9</td>
</tr>
<tr>
<td>(8, Filter 4)</td>
<td>6</td>
</tr>
<tr>
<td>(9, Filter 0)</td>
<td>9</td>
</tr>
<tr>
<td>(9, Filter 1)</td>
<td>9</td>
</tr>
<tr>
<td>(9, Filter 2)</td>
<td>9</td>
</tr>
<tr>
<td>(9, Filter 3)</td>
<td>9</td>
</tr>
<tr>
<td>(9, Filter 4)</td>
<td>6</td>
</tr>
<tr>
<td>(10, Filter 0)</td>
<td>9</td>
</tr>
<tr>
<td>(10, Filter 1)</td>
<td>9</td>
</tr>
<tr>
<td>(10, Filter 2)</td>
<td>9</td>
</tr>
<tr>
<td>(10, Filter 3)</td>
<td>10</td>
</tr>
<tr>
<td>(10, Filter 4)</td>
<td>6</td>
</tr>
<tr>
<td>(11, Filter 0)</td>
<td>11</td>
</tr>
<tr>
<td>(11, Filter 1)</td>
<td>9</td>
</tr>
<tr>
<td>(11, Filter 2)</td>
<td>9</td>
</tr>
<tr>
<td>(11, Filter 3)</td>
<td>9</td>
</tr>
<tr>
<td>(11, Filter 4)</td>
<td>7</td>
</tr>
<tr>
<td>12 ‘No BA’s in a row’</td>
<td>2</td>
</tr>
<tr>
<td>All remaining symbols</td>
<td>9</td>
</tr>
</tbody>
</table>
C. Training Data for the Huffman Codes of the Side Information

The following 13 images are used to obtain the statistics to construct the first set of Huffman codes (Set 1) given in Appendix B. As explained in Chapter 5, each of these images is coded at 0.2, 0.25, 0.3, 0.35, 0.4 and 0.45 bpp and the resulting 78 (=13 * 6) pieces of statistics is averaged and used to compute the probabilities of each of the 62 symbols in Appendix B. The Huffman codes are computed from these probabilities.

The first three images, shown below, have resolution of 256x256 pixels. The fourth image, shown on the upper part of the next page has resolution of 400x592 pixels. The remaining nine images have resolution of 512x512 pixels.
The video sequences that are shown below (Carphone and Akiyo) are used to obtain the statistics to construct the second set of Huffman codes (Set 2) given in Appendix B. As explained in Chapter 5, the statistics are obtained from P- and B-frames of these video sequences because the second set is used to code the side information of the P- and B-frames. Each of these two sequences was coded at two bit-rates, 200 Kb/s and 300 Kb/s. Since the two video sequences have 300 frames each, about 1120 \((14/15) \times 300 \times 2 \times 2\) pieces of statistics were obtained. The average of these statistics was used to compute the probabilities of each of the 62 symbols in Appendix B. The Huffman codes are computed from these probabilities.
References


[22] www.mpeg.org/mpeg/mssg