

UNIVERSITY OF GAZİANTEP
GRADUATE SCHOOL OF
NATURAL & APPLIED SCIENCES

**ADAPTIVE FUZZY FILTERING FOR ARTIFACT
ELIMINATION IN COMPRESSED IMAGES AND VIDEOS**

M. Sc. THESIS
IN
ELECTRICAL-ELECTRONICS ENGINEERING

BY
SEYDİ KAÇMAZ
AUGUST 2012

**ADAPTIVE FUZZY FILTERING FOR ARTIFACT
ELIMINATION IN COMPRESSED IMAGES AND VIDEOS**

M.Sc. Thesis

in

Electrical-Electronics Engineering

University of Gaziantep

Supervisor

Asst. Prof. Dr. Sema Koç KAYHAN

Co-Supervisor

Prof. Dr. Ergun ERÇELEBİ

by

Seydi KAÇMAZ

August 2012

©2012 [Seydi KAÇMAZ]

REPUBLIC OF TURKEY
UNIVERSITY OF GAZIANTEP
GRADUATE SCHOOL OF NATURAL & APPLIED SCIENCES
ELECTRICAL – ELECTRONICS ENGINEERING

Name of the thesis: Adaptive fuzzy filtering for artifact elimination in compressed images and videos

Name of the student: Seydi KAÇMAZ

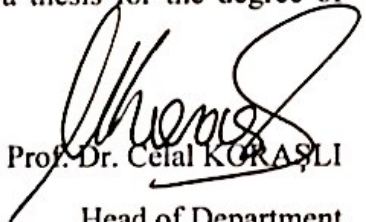
Exam date: 17.08.2012

Approval of the Graduate School of Natural and Applied Sciences

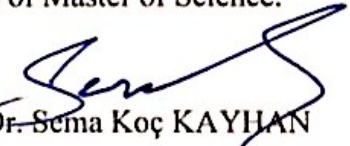

Prof. Dr. Ramazan KOÇ

Director

I certify that this thesis satisfies all the requirements as a thesis for the degree of Master of Science.


Prof. Dr. Celal KGRASLI
Head of Department

This is to certify that we have read this thesis and that in our consensus opinion it is fully adequate, in scope and quality, as a thesis for the degree of Master of Science.


Asst. Prof. Dr. Sema Koç KAYHAN
Supervisor

Examining Committee Members

Prof. Dr. Ergun Erçelebi


.....

Asst. Prof. Dr. Sema Koç Kayhan (Chairman)


.....

Asst. Prof. Dr. Mehmet Hadi Süzer


.....

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Seydi KAÇMAZ

ABSTRACT

ADAPTIVE FUZZY FILTERING FOR ARTIFACT ELIMINATION IN COMPRESSED IMAGES AND VIDEOS

KAÇMAZ, Seydi

M.Sc. in Electrical-Electronics Eng.

Supervisor: Asst. Prof. Dr. Sema Koç KAYHAN

August 2012, 72 pages

Block-based Discrete Cosine Transform (DCT) image and video compression methods have been successfully used in image and video compression applications due to bandwidth and storage limitations. However, compression distortion becomes significant when these algorithms are used under a certain bit rate. The most noticeable degradations of block transform coding are blocking and ringing artifacts.

In this thesis, two new adaptive post-filtering algorithms are proposed to remove observed coding artifacts as a result of DCT based image and video compression standards at low bit rates. With identification of coding artifact strength, fuzzy filter is applied by adjusting filtering range and its parameters.

Experimental results showed that, the proposed algorithms exhibit better detail preservation and artifact removal performance with lower computational cost as compared to other post-processing techniques. Accordingly, these can be used for the real time image and video applications without undesired artifacts.

Key Words: Block-based DCT, image and video compression, post-processing filter, coding artifacts, blocking artifact, ringing artifact, adaptive fuzzy filter, real time image and video applications

ÖZET

SIKIŞTIRILMIŞ İMGELERDE VE VİDEOLARDA BOZULMA ETKİSİNİ GİDERMEK İÇİN UYARLAMALI BULANIK SÜZGEÇLEME

KAÇMAZ, Seydi

Yüksek Lisans Tezi, Elektrik-Elektronik Müh. Bölümü

Tez Yöneticisi: Yrd. Doç. Dr. Sema Koç KAYHAN

Ağustos 2012, 72 sayfa

Blok temelli Ayrık Kosinüs Dönüşümünü (AKD) kullanan resim ve video sıkıştırma metodları, bant genişliği ve bellek kısıtlamaları yüzünden resim ve video sıkıştırma uygulamalarında başarılı bir şekilde kullanılmaktadır. Ama bu algoritmalar belli bir oranının altında kullanıldığında sıkıştırma bozulmaları önemli hale gelmektedir. Blok dönüşümünün en farkedilebilir bozulmaları blok ve yüksek frekans bozulmalarıdır.

Bu tezde, düşük bit oranlarında AKD temelli imge ve video standartlarının sonucunda gözlenen kodlama bozulmalarını gidermek için, iki yeni uyarlamalı sıkıştırma sonrası algoritmalar önerilmiştir. Kodlama bozulmalarının şiddetine göre, filtre aralığı ve parametreleri ayarlanarak bulanık filtre uygulanmıştır.

Deney sonuçlarına göre, önerilen algoritmalar diğer yöntemlere göre daha iyi detay koruma performansı sergilemekte ve daha düşük işlem yükü gerektirmektedir. Özetle, bu algoritmalar gerçek zamanlı imge ve video uygulamalarında istenmeyen bozulmaları gidermekte kullanılabilir.

Anahtar Kelimeler: Blok-temelli AKD, imge ve video sıkıştırma, işlem-sonrası filtreleme, kodlama bozulması, blok bozulması, yüksek frekans bozulması, uyarlamalı bulanık filtre, gerçek zamanlı imge ve video uygulamaları

"To my family"

ACKNOWLEDGEMENTS

First and most importantly I must thank the Most Merciful, Almighty ALLAH for making everything possible.

I would like to express my deepest gratitude to my supervisor Asst. Prof. Dr. Sema Koç KAYHAN and my co-supervisor Prof. Dr. Ergun ERÇELEBİ for their guidance, advice, criticism, encouragements and insight throughout this study.

I would like to thank to colleagues, Mehmet DEMİR, Ali Osman ARSLAN, Mahmut AYKAÇ, Mehmet ARICI, and Taner İNCE for supporting and encouraging me with their best wishes.

Special thanks to my friends, Erhan ERSOY, Abdülkadir GÜMÜŞÇÜ and Eda ADAL for their kind help, encouragement and patience during my study.

I also wish to thank to my sister and her husband, Hatice VURAL and Mehmet VURAL, for believing in me.

Finally, I would like express my love and appreciation to my family, Abdülkadir KAÇMAZ and Sevgi KAÇMAZ who have created and maintained a wonderful life for me and contributed to my life with their lovely supports and encouragements.

TABLE OF CONTENTS

ABSTRACT	V
ÖZET	VI
ACKNOWLEDGEMENTS	VIII
TABLE OF CONTENTS	IX
LIST OF FIGURES	XI
LIST OF TABLES	XIII
LIST OF SYMBOLS/ABBREVIATIONS	XIV
CHAPTER I (INTRODUCTION)	1
CHAPTER II (BACKGROUND)	3
2.1. Image and Video Processing Definitions	3
2.2. Image Compression Using Discrete Cosine Transform	3
2.3. Jpeg Compression.....	6
2.4. Coding Artifacts	8
2.4.1. Blocking Artifacts.....	8
2.4.2. Ringing Artifacts.....	8
2.5. Previous Research	10
2.5.1. Estimation Theoretic Methods.....	10
2.5.2. POCS Based Methods.....	11
2.5.3. Wavelet Based Methods	12
2.5.4. Adaptive Filtering Methods	12
2.5.5. Fuzzy Filtering Methods.....	14
CHAPTER III (FUZZY LOGIC).....	15
3.1. Fuzzy Image Processing.....	19
CHAPTER IV (REMOVING BLOCKING ARTIFACTS).....	21
4.1. Blocking Artifact Detection	21
4.2. Deblocking Filtering.....	24
4.2.1. Filtering on Smooth Region (Intensive Degradation)	26
4.2.2. Filtering on Texture Region (Low Degradation).....	26
4.2.3. Filtering on Transition Region (Moderate Degradation).....	27

CHAPTER V (REMOVING RINGING ARTIFACTS).....	28
5.1. Edge Detection	29
5.2. Complementary Ringing Detection.....	31
5.3. Deringing Filtering	33
5.3.1. Filtering on Strong Ringing Artifacts	34
5.3.2. Filtering on Weak Ringing Artifacts	34
CHAPTER VI (EXPERIMENTAL RESULTS).....	35
6.1. Picture Quality Evaluation	35
6.2. Quality Metrics.....	36
6.3. The Results for Images from Experiments.....	38
CHAPTER VII (CONCLUSION)	68
REFERENCES.....	69

LIST OF FIGURES

Figure 1. 1. The compressed Lena image at different bit rates; (a) 1 bpp (b) 0.188 bpp	2
Figure 2. 1. JPEG Encoder.....	7
Figure 2. 2. JPEG Decoder.....	7
Figure 2. 3. Blocking Artifacts in compressed image at 0.188 bpp.....	9
Figure 2. 4. Ringing Artifacts in compressed image at 0.13 bpp.....	9
Figure 3. 1. A fuzzy logic system which accepts imprecise data and vague statements.....	16
Figure 3. 2. Boundary region of a fuzzy set.....	17
Figure 3. 3. The fuzzy sets “tall” and “short”.....	18
Figure 3. 4. Configuration of a pure fuzzy system.....	19
Figure 3. 5. Fuzzy Image Processing	19
Figure 4. 1. The general flow of the proposed deblocking algorithm.....	21
Figure 4. 2. The flowchart of the proposed deblocking algorithm.....	22
Figure 4. 3. Boundary areas around the block of interest in blocking artifact detection.....	23
Figure 4. 4. Membership function for Strong Blocking Artifact	26
Figure 4. 5. Membership function for Weak Blocking Artifact.....	27
Figure 4. 6. Membership function for Transition Blocking Artifact.	27
Figure 5. 1. The general flow of the proposed deringing algorithm.	28
Figure 5. 2. The flowchart of the proposed deringing algorithm.....	29
Figure 5. 3. The 8 adjacent blocks of interest in complementary ringing detection..	31
Figure 6. 1. Images corrupted with different noise and artifacts with identical MSE.....	37
Figure 6. 2. Images corrupted with different noise and artifacts with identical MSE.	39
Figure 6. 3. PSNR and SSIM performance of the proposed algorithms for the Lena.	41
Figure 6. 4. PSNR and SSIM performance of the proposed algorithms for the Goldhill.	43
Figure 6. 5. PSNR and SSIM performance of the proposed algorithms for the Airplane.	45
Figure 6. 6. PSNR and SSIM performance of the proposed algorithms for the Peppers.....	47
Figure 6. 7. PSNR and SSIM performance of the proposed algorithms for the Mandrill.....	49
Figure 6. 8. Subjective quality comparison of the compressed image the Lena at 0.1335 bpp.	51

Figure 6. 9. Subjective quality comparison of the compressed image the Lena at 0.2034 bpp.	52
Figure 6. 10. Subjective quality comparison of the compressed image the Goldhill at 0.1255 bpp.	53
Figure 6. 11. Subjective quality comparison of the compressed image the Goldhill at 0.2090 bpp.	54
Figure 6. 12. Subjective quality comparison of the compressed image the Airplane at 0.1411 bpp.	55
Figure 6. 13. Subjective quality comparison of the compressed image the Airplane at 0.2051 bpp.	56
Figure 6. 14. Subjective quality comparison of the compressed image the Peppers at 0.1366 bpp.	57
Figure 6. 15. Subjective quality comparison of the compressed image the Peppers at 0.2050 bpp.	58
Figure 6. 16. Subjective quality comparison of the compressed image the Mandrill at 0.1530 bpp.	59
Figure 6. 17. Subjective quality comparison of the compressed image the Mandrill at 0.2105 bpp.	60
Figure 6. 18. PSNR versus bit rate for the compressed image Lena.....	62
Figure 6. 19. Subjective quality comparison of the compressed Lena image at 0.217 bpp post-processed by different methods.	63
Figure 6. 20. Subjective quality comparison of the compressed the Peppers image at 0.221 bpp post-processed by different methods.	64
Figure 6. 21. PSNR versus bit rate for the compressed image Lena.....	65
Figure 6. 22. Subjective quality comparison of the compressed Lena image at 0.227 bpp post-processed by different methods.	66
Figure 6. 23. Subjective quality comparison of the compressed the Peppers image at 0.240 bpp post-processed by different methods.	67

LIST OF TABLES

Table 6. 1. Objective results of proposed algorithms for different compression ratios for the Lena.	40
Table 6. 2. Objective results of proposed algorithms for different compression ratios for the Goldhill.	42
Table 6. 3. Objective results of proposed algorithms for different compression ratios for the Airplane.	44
Table 6. 4. Objective results of proposed algorithms for different compression ratios for the Peppers.	46
Table 6. 5. Objective results of proposed algorithms for different compression ratios for the Mandrill.	48
Table 6. 6. Bit rate versus Run time.	61
Table 6. 7. Comparison of PSNR for different post-processing techniques.	62
Table 6. 8. Comparison of PSNR for different post-processing techniques.	65

LIST OF SYMBOLS/ABBREVIATIONS

DCT	Discrete Cosine Transform
IDCT	Inverse Discrete Cosine Transform
AKD	Ayrık Kosinüs Dönüşümü
DFT	Discrete Fourier Transform
JPEG	Joint Photographic Experts Group
MPEG	The Moving Picture Experts Group
HDTV	High Definition Television
bpp	Bit Per Pixel
POCS	Projection onto Convex Sets
HVS	Human Visual System
SQA	Subjective Quality Assessment
OQA	Objective Quality Assessment
FR	Full Reference
NR	No Reference
RR	Reduced Reference
MSE	Mean Squared Error
PSNR	Peak Signal to Noise Ratio
SSIM	Structural Similarity Index

CHAPTER I

INTRODUCTION

Nowadays, a lot of communication applications contain pictures or video such as many web pages on the internet, video conferencing, video over the internet, videophones, high definition television (HDTV) among many others. And they are increasingly entered our lives. Accordingly, the storage and transmission is getting an importance. Because, uncompressed pictures or video requires very large space, also needs very high data rates without interruption. However, there are limited memory and bandwidth to supply these requirements. The most popular solution of this problem is to compress images or video by removing significant redundancy. Image or video compression techniques offer good solution to reduce file size while preserving the original content.

There are two types of compression: lossless and lossy. Lossless compression removes only the redundancy in images. For this form of compression, the reconstructed images are identical to original ones. So, it doesn't change any image detail. But unlike lossless compression, lossy compression has a trade-off between the amount of information lost and the degree of compression. The amount of compression is determined by the bandwidth requirements of the application. Applications have a very small bandwidth need at very high compression ratios. Under this condition, some visually annoying artifacts known as coding artifacts occur, in addition to loss of detail. The most obvious coding artifacts affect the visual quality of the image and video are blocking and ringing artifacts. Coding artifacts are making the image or video unpleasant to the viewer, as shown in Figure 1.1. For this reason, reduction of coding artifacts is essential to make the compressed image and video acceptable to the human viewer.



(a)

(b)

Figure 1. 1. The compressed Lena image at different bit rates; (a) 1 bbp (b) 0.188 bbp.

Block-based DCT coding has been widely used in image and video compression algorithms, such as JPEG and MPEG. 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 main reason for coding artifacts. Various methods have been proposed in the literature to reduce coding artifacts. In this thesis, new and efficient approach is developed.

An overview of the remaining chapters is as follows: In the second chapter of this thesis, first, backgrounds for image and video processing and DCT are given, after that the description of coding artifacts are introduced, then, a summary of existing methods in the literature is provided. In the third chapter, fuzzy logic and filter is explained. In Chapter 4 and 5, proposed deblocking algorithm and deringing algorithm are presented respectively. Experimental results of the developed algorithms are showed and the success of methods is proved in Chapter 6. Finally, Chapter 7 summarizes contribution of this thesis to the literature, and points to future research directions.

CHAPTER II

BACKGROUND

2.1. 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 digitized for the digital image processing and transmission. This involves sampling the signal along two dimensions and quantizing those samples. The resulting image samples are often referred to picture elements or pixels. The number of pixels used for the entire image is called the spatial resolution of the image.

To represent a color, additive combination of three primary colors, red, green, and blue, is used. This representation of the 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 by 8 bits.

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 a frame in a video processing terminology. This way of digitizing the video signal is called progressive scanning. A video sequence can be seen as a collection of frames, with equal dimensions, sampled at equally spaced time intervals.

2.2. Image Compression Using Discrete Cosine Transform

Discrete cosine transform (DCT) is widely used in image processing, especially for compression. Some of the applications of two-dimensional DCT involve still image compression and compression of individual video frames, while multidimensional DCT is mostly used for compression of video streams. DCT is also useful for

transferring multidimensional data to frequency domain, where different operations, like spread spectrum, data compression, data watermarking, can be performed in easier and more efficient manner. A number of papers discussing DCT algorithms are available in the literature that signifies its importance and application [1]. The DCT is an orthogonal transform, which has a fixed set of (image independent) basis functions, an efficient algorithm for computation, and good energy compaction and correlation reduction properties.

The DCT belongs to the family of discrete trigonometric transform, which has 16 members [2]. The 1D DCT of a $1 \times N$ vector $x(n)$ is defined as

$$Y[k] = C[k] \sum_{n=0}^{N-1} x[n] \cos \left[\frac{(2n+1)k\pi}{2N} \right] \quad (2.1)$$

where $k = 0, 1, 2, \dots, N-1$ and

$$C[k] = \begin{bmatrix} \sqrt{\frac{1}{N}} & \text{for } k = 0 \\ \sqrt{\frac{2}{N}} & \text{for } k = 1, 2, \dots, N-1 \end{bmatrix} \quad (2.2)$$

The original signal vector $x(n)$ can be reconstructed back from the DCT coefficients $Y[k]$ using the Inverse DCT (IDCT) operation and can be defined as

$$x[n] = \sum_{k=0}^{N-1} C[k] Y[k] \cos \left[\frac{(2n+1)k\pi}{2N} \right] \quad (2.3)$$

where $n = 0, 1, 2, \dots, N-1$ and

$$C[k] = \begin{bmatrix} \sqrt{\frac{1}{N}} & \text{for } k = 0 \\ \sqrt{\frac{2}{N}} & \text{for } k = 1, 2, \dots, N-1 \end{bmatrix} \quad (2.4)$$

The DCT can be extended to the transformation of 2D signals or images. This can be achieved in two steps: by computing the 1D DCT of each of the individual rows of the two dimensional image and then computing the 1D DCT of each column of the image. If represents a 2D image of size $x(n_1, n_2)$ $N \times N$, then the 2D DCT of an image is given by:

$$Y[j, k] = C[j]C[k] \sum_{m=0}^{N-1} \sum_{n=0}^{N-1} x[m, n] \cos\left[\frac{(2m+1)j\pi}{2N}\right] \cos\left[\frac{(2n+1)k\pi}{2N}\right] \quad (2.5)$$

where $m = 0, 1, 2, \dots, N-1$, $n = 0, 1, 2, \dots, N-1$ and

$$C[j] = \begin{bmatrix} \sqrt{\frac{1}{N}} & \text{for } j = 0 \\ \sqrt{\frac{2}{N}} & \text{for } j = 1, 2, \dots, N-1 \end{bmatrix} \quad \text{and} \quad C[k] = \begin{bmatrix} \sqrt{\frac{1}{N}} & \text{for } k = 0 \\ \sqrt{\frac{2}{N}} & \text{for } k = 1, 2, \dots, N-1 \end{bmatrix} \quad (2.6)$$

Similarly the 2D IDCT can be defined as

$$x[m, n] = \sum_{j=0}^{N-1} \sum_{k=0}^{N-1} C[j]Y[k]Y[j, k] \cos\left[\frac{(2m+1)j\pi}{2N}\right] \cos\left[\frac{(2n+1)k\pi}{2N}\right] \quad (2.7)$$

where $m = 0, 1, 2, \dots, N-1$, $n = 0, 1, 2, \dots, N-1$ and

$$C[j] = \begin{bmatrix} \sqrt{\frac{1}{N}} & \text{for } j = 0 \\ \sqrt{\frac{2}{N}} & \text{for } j = 1, 2, \dots, N-1 \end{bmatrix} \quad \text{and} \quad C[k] = \begin{bmatrix} \sqrt{\frac{1}{N}} & \text{for } k = 0 \\ \sqrt{\frac{2}{N}} & \text{for } k = 1, 2, \dots, N-1 \end{bmatrix} \quad (2.8)$$

The DCT is a real valued transform and is closely related to the DFT. In particular, a $N \times N$ DCT of $x(n_1, n_2)$ can be expressed in terms of DFT of its even-symmetric extension, which leads to a fast computational algorithm. Because of the even-symmetric extension process, no artificial discontinuities are introduced at the block boundaries. Additionally the computation of the DCT requires only real arithmetic. Because of the above properties the DCT is popular and widely used for data compression operation.

The DCT is applied to image blocks $N \times N$ pixels in size (where N is usually multiple of 2) over the entire image. The size of the blocks used is an important factor since they determine the effectiveness of the transform over the whole image. If the blocks are too small then the images is not effectively decorrelated but if the blocks are too big then local features are no longer exploited. The tiling of any transform across the image leads to artifacts at the block boundaries. The DCT is associated with blocking artifact since the JPEG standard suffers heavily from this at higher compressions. However the DCT is protected against blocking artifact as effectively as possible, without interconnecting blocks, since the DCT basis functions all have a zero gradient at the edges of their blocks. This means that only the DC level significantly affects the blocking artifact and this can then be targeted. Ringing is a major problem in DCT operation. When edges occur in an image DCT relies on the high frequency components to make the image shaper. However these high frequency components persist across the whole block and although they are effective at improving the edge quality they tend to 'ring' in the flat areas of the block. This ringing effect increases, when larger blocks are used, but larger blocks are better in compression terms, so a trade off is usually established [3]. The blocking and ringing artifacts are also discussed in the next parts.

2.3. Jpeg Compression

The JPEG (Joint Photographic Experts Group) standard has been around since the late 1980's and has been an effective first solution to the standardization of lossy image compression. Although JPEG has some very useful strategies for DCT quantization and compression, it was only developed for low compressions. The 8×8 DCT block size was chosen. The JPEG Standard will be also briefly explained in this part to provide a basis to understand the new DCT related work [4].

The following is the general overview of the JPEG process. There are quite a lot of interesting techniques used in the JPEG standard and it is important to give an overview of how JPEG works. There are several variations of JPEG, but only the 'baseline' method is discussed here.

- 1) The image is broken into 8×8 blocks of pixels.
- 2) Working from left to right, top to bottom, the DCT is applies to each block.

- 3) Each block is compressed through quantization.
- 4) The array of compressed blocks that constitute the image is stored in a drastically reduced amount of space.
- 5) When desired the image is constructed through decompression, a process that uses the Inverse Discrete Cosine Transform (IDCT).

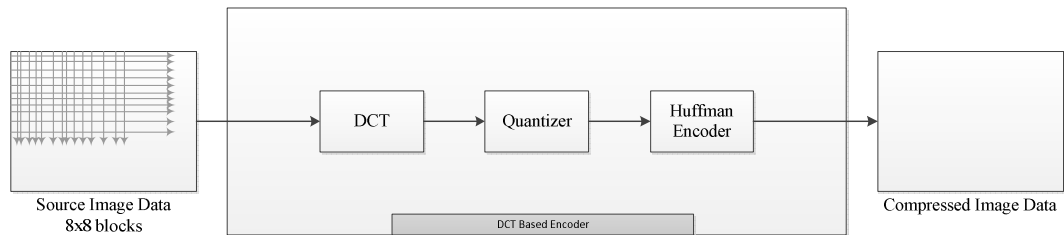


Figure 2. 1. JPEG Encoder.

As shown in the Figure 2.1, the image is first partitioned into non-overlapping 8 x 8 blocks. A Discrete Cosine Transform (DCT) is applied to each block to convert the spatial domain gray levels of pixels into coefficients in frequency domain. To improve the precision of the DCT the image is 'zero shifted', before the DCT is applied. This converts a 0 → 255 image intensity range to a -128 → 127 range, which works more efficiently with the DCT. One of these transformed values is referred to as the DC coefficient and the other 63 as the AC coefficients [5].

After the computation of DCT coefficients, they are normalized with different scales according to a quantization table provided by the JPEG standard conducted by psycho visual evidence. The quantized coefficients are rearranged in a zigzag scan order for further compressed by an efficient lossless coding algorithm such as runlength coding, arithmetic coding, Huffman coding. The decoding process is simply the inverse process of encoding as shown in Figure 2.2.

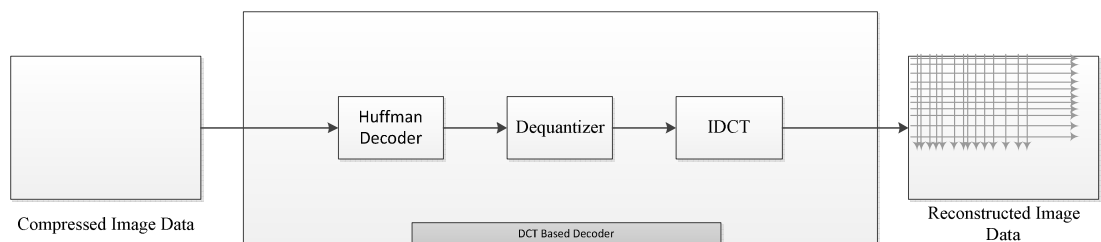


Figure 2. 2. JPEG Decoder.

2.4. Coding Artifacts

During the image and video coding, block based discrete cosine transform causes a number of visibly annoying coding artifacts. Classifications of these coding artifacts are essential in evaluating the overall visual quality of the compressed images and video. To understand the image and video quality, causes of these artifacts and their effects on viewing quality need to be understood [6].

2.4.1. Blocking Artifacts

Blocking artifact is the most recognizable and widely studied distortion in block based DCT coders. Generally, the blocking effect is perceived as a discontinuity between the boundaries of two neighboring blocks.

The main cause of blocking artifact is the non-overlapping block coding strategy. A picture is divided into a number of non-overlapping blocks. Each block is independently transformed with the DCT and then quantized. The signal continuities between blocks are not always guaranteed and discontinuity between two blocks may become visible at the boundary between those two blocks. This artifact is called the blocking artifact. It is more prominent at a low-bit rate coding. This kind of edge artifact is composed only horizontal and vertical edges at the boundaries between two blocks. The compressed image at 0.188 bpp is shown in Figure 2.3, where the blocking effect can be easily noticed and is visually unpleasant.

2.4.2. Ringing Artifacts

The ringing effect usually appears as shimmering ripples along high contrast edges surrounded by smooth texture areas. With a coarse quantization, large ripples often appear around the high contrast edges. Moreover, these ripples are only constrained inside the blocks that contain high contrast edges. Figure 2.4 illustrates the ringing effect along the edges. However, ringing may not be visibly noticeable in high texture areas because rough texture areas can hide ringing artifacts. This phenomenon can be regarded as the masking effect of the human visual system (HVS).



Figure 2. 3. Blocking Artifacts in compressed image at 0.188 bpp.

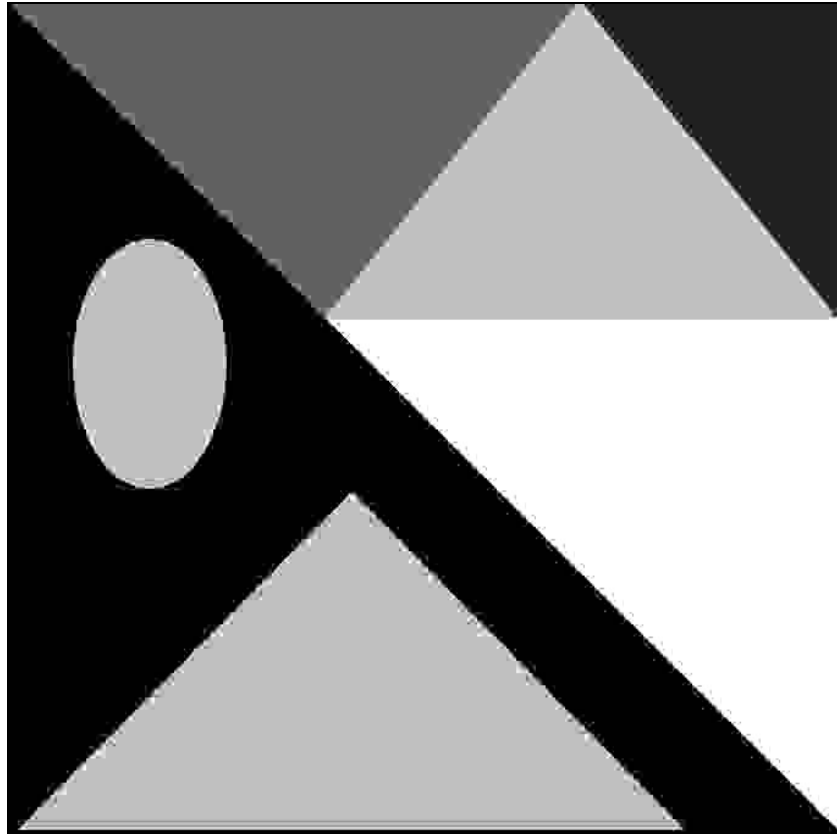


Figure 2. 4. Ringing Artifacts in compressed image at 0.13 bpp.

2.5. Previous Research

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

Various deblocking and deringing methods have been proposed in the literature to reduce the coding artifacts. Many of these methods are based on the post-processing idea. In other words, these methods take only the compressed image as their input and process it to reduce the coding artifacts.

We provide a brief summary of our literature search in this section. Our literature search indicates that the use of adaptive fuzzy filter to reduce the coding artifacts has not been explored enough. Since the reduction of coding artifacts is an important problem, especially in a low bit-rate video coding, new approaches should be explored, even if existing methods work reasonably well. Therefore, this thesis develops both new detection algorithm and efficient fuzzy filter to reduce coding artifacts and shows results with comparison of some methods in literature.

In the following subsections, main ideas of some of the previous researches are summarized. Post-processing-type methods are based on different theoretical frameworks including Estimation Theory, Projection onto Convex Sets (POCS), Wavelets, Adaptive Filtering and Fuzzy Filtering.

2.5.1. Estimation Theoretic Methods

In methods based on the Estimation Theory [7-10], 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 [11], 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.

2.5.2. POCS Based Methods

POCS theory was first applied to image restoration problems by Youla [12]. After the blocking artifact problem began to be studied by the image coding community, many methods based on POCS theory for reducing the blocking artifacts flourished [13-17]. 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 the projections. For example, if three convex sets were defined, then iterations of the 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.

These methods also require excessive amount of computation. Similarly in the Estimation Theoretic methods, although it provide acceptable to a certain degree for reducing blocking artifacts in images, it is not acceptable for reducing blocking artifacts in video.

2.5.3. Wavelet Based Methods

Some methods reduce the blocking artifacts using the wavelet domain representation of images. These methods start by transforming the compressed image with the blocking artifacts to the wavelet domain. Then, the wavelet coefficients related to block boundaries with the 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 [18], 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 the wavelet coefficients across scales. Locations above a threshold are identified as edges. Then, the wavelet coefficients at non-edge locations are set to zero while the 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.

2.5.4. Adaptive Filtering Methods

Since the 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 [19]. 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 the 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 [20-24]. For example, in [25], first, blocks with the 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 [26], 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. The 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. The ringing artifacts can mislead the algorithm as the algorithm might interpret blocks with the 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.

2.5.5. Fuzzy Filtering Methods

Fuzzy filters are improved median filters or a rank condition rank selection filters [27], where the binary spatial-rank relation is replaced by a real valued relation. This permits the filter to adapt to the spread of the signal by averaging the flat areas, while the isolated pixels in the edge areas remain. Fuzzy filter is also a specific case of bilateral filters [28], [29].

The fuzzy filter [30] exploits the spatial correlation, rank order and diversity of the signal. It overcomes the limitation of the spatial-rank space filters such as weighted median [31], rank condition rank selection [32] and lowerupper- middle [33] filters by adding the spread information to get better clustering performance. Clustering the similarly valued samples to their mean value and keeping isolated values unchanged. This characteristic is very helpful in preserving the real edge of the image while reducing the ringing artifact in the flat area near these edges.

In [34], a new method using fuzzy filtering is presented to remove the coding artifacts in compressed video. For deblocking, the block edge strength is detected, and a 1D fuzzy filter adjusts its window size and filtering range according to it. For deringing, 8x8 blocks are finely classified into four categories and 2D fuzzy filter with adaptive spread parameter is applied to them. Although this method involves promising fuzzy filter, its artifact judgment is not accurate enough. When blocking artifacts or ringing artifacts get serious, the corresponding judgment conditions become invalid.

CHAPTER III

FUZZY LOGIC

Human knowledge nowadays becomes increasingly important. It is gained by experiencing the world. However, we are all limited in our ability to perceive the world. So, we find uncertainty in everywhere. The other limitation factor for precision is a natural language used for describing knowledge. We understand general meaning of the word and are able to communicate in an acceptable degree, but generally we can not precisely agree among ourselves on the single word or terms of common sense meaning. Accordingly, natural languages are vague.

Our perception of the real world does not have sharply defined boundaries. For example; many, short, much larger than, old, etc. are true for some degree and false for some degree as well. These concepts are called fuzzy or gray concepts. Although a human brain works with them, computers may not do it. Natural languages can be vague whereas programming languages can not.

The entire real world is complex, and this complexity arises from uncertainty. According to Dr. Lotfi Zadeh, the complexity and the imprecision are correlated. The Fuzzy Logic tool is a mathematical tool for dealing with uncertainty. It offers to a soft computing with the important concept of the words. The fuzzy theory provides a technique for representing linguistic constructs such as “many,” “low,” “medium,” “often,” “few.” In general, the fuzzy logic presents an inference structure that enables appropriate human reasoning capabilities.

The traditional binary set theory describes crisp events, events that either do or do not occur. It uses probability theory to explain if an event will occur, measuring the chance with which a given event is expected to occur. On the other hand, the theory of fuzzy logic is based upon the notion of relative graded membership. The utility of fuzzy sets lies in their ability to model uncertain or ambiguous data encountered in real life. Figure 3.1 models basic Fuzzy Logic System.

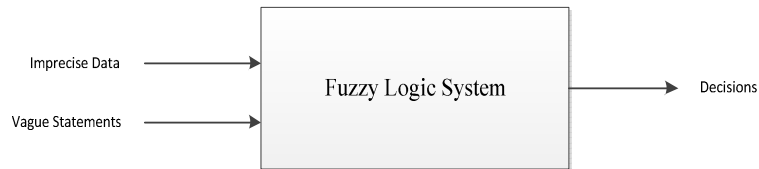


Figure 3. 1. A fuzzy logic system which accepts imprecise data and vague statements.

Real world problems are too complex, and the complexity comes from the degree of uncertainty. As uncertainty increases, the complexity of the problem also increases. Traditional system modeling and analysis techniques are too precise for such problems, and in order to make less complexity, we introduce appropriate simplifications, assumptions, etc. to achieve a satisfactory compromise between the information we have and the amount of uncertainty we are willing to accept. In this aspect, fuzzy systems theory is similar to other engineering theories, because almost all of them characterize the real world in an approximate manner.

Fuzzy sets provide means to model the uncertainty associated with vagueness, imprecision, and lack of information regarding a problem or a plant, etc. For example, consider the meaning of a “short person”. For an individual X , the short person may be one whose height is below 1.50 meters. For other individual Y , the short person may be one whose height is below or equal to 1.60 meters. The term “short” informs the same meaning to the individuals X and Y , but it is found that they both do not provide a unique definition. This variable “short” is called as linguistic variable, which represents the imprecision existing in the system.

The uncertainty is found to arise from ignorance, from chance and randomness, due to lack of knowledge, from vagueness, like the fuzziness existing in our natural language. Lotfi Zadeh proposed the set membership idea to make suitable decisions when uncertainty occurs. Consider the “short” example discussed previously. If we take “short” as a height equal to or less than 1.55 meters, then 1.50 meter would easily become the member of the set “short” and 1.60 meters will not be a member of the set “short.” The membership value is “1” if it belongs to the set or “0” if it is not a member of the set. Thus membership in a set is found to be binary i.e., the element is a member of a set or not.

It can be indicated as,

$$X_A(x) = \begin{cases} 1 & , x \in A \\ 0 & , x \notin A \end{cases} \quad (3.1)$$

where $X_A(x)$ is the membership of element x in set A and A is the entire set on the universe.

This membership was extended to possess various degree of membership on the real continuous interval $[0,1]$. Zadeh formed fuzzy sets as the sets on the universe X which can accommodate degrees of membership. The concept of a fuzzy set contrasts with a classical concept of a crisp set which is a collection of things for which it is known whether any given thing is inside it or not. Zadeh generalized the idea of a crisp set by extending a valuation set $\{1, 0\}$ (definitely in/definitely out) to the interval of real values (degrees of membership) between 1 and 0 denoted as $[0,1]$. Fuzzy sets tend to capture vagueness exclusively via membership functions that are mappings from a given universe of discourse X to a unit interval containing membership values.

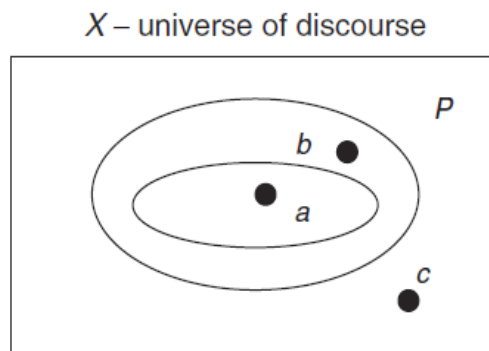


Figure 3. 2. Boundary region of a fuzzy set.

For example, from the Figure 3.2, it can be noted that a is clearly a member of fuzzy set P , c is clearly not a member of fuzzy set P , but the membership of b is found to be vague. Hence a can take membership value 1, c can take membership value 0 and b can take membership value between 0 and 1, say 0.4, 0.7, etc. This is set to be a partial membership of fuzzy set P .

The membership function for a set maps each element of the set to a membership value between 0 and 1 and uniquely describes that set. The values 0 and 1 describe “not belonging to” and “belonging to” a conventional set respectively. The values between 0 and 1 represent fuzziness. Determining the membership function is subjective to varying degrees depending on the situation. It depends on an individual’s perception of the data in question and does not depend on randomness. This is important, and distinguishes fuzzy set theory from probability theory. Figure 3.3 shows an example of the classification which is subjective and depends on what height is measured relative to. At the extremes, the distinction is clear, but there is a large amount of overlap in the middle.

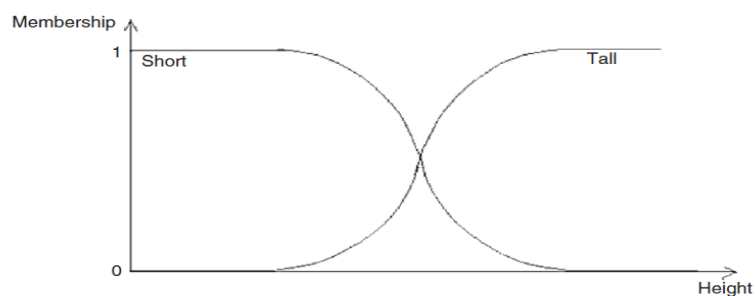


Figure 3. 3. The fuzzy sets “tall” and “short”.

In practice, fuzzy logic means computation of words. Since computation with words is possible, computerized systems can be built by embedding human expertise articulated in daily language. Also called a fuzzy inference engine or fuzzy rule-base, such a system can perform approximate reasoning somewhat similar to but much more primitive than that of the human brain.

A fuzzy system is a set of fuzzy rules that converts inputs to outputs. The basic configuration of a pure fuzzy system is shown in Figure 3.4. The fuzzy inference engine combines fuzzy IF–THEN rules into a mapping from fuzzy sets in the input space X to fuzzy sets in the output space Y based on fuzzy logic principles. Fuzzy sets form the building blocks for fuzzy IF–THEN rules which have the general form “IF X is A THEN Y is B ,” where A and B are fuzzy sets. The term “fuzzy systems” refers mostly to systems that are governed by fuzzy IF– THEN rules. The IF part of an implication is called the antecedent, whereas THEN part is a consequent. The main feature of reasoning using these rules is its partial matching capability, which

enables an inference to be made from a fuzzy rule even when the rule's condition is only partially satisfied.

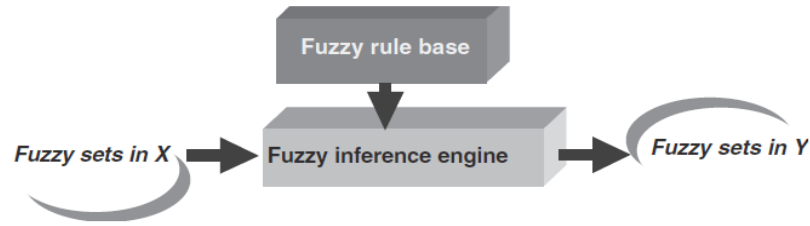


Figure 3. 4. Configuration of a pure fuzzy system.

3.1. Fuzzy Image Processing

Fuzzy image processing has not a unique theory. Because it is a collection of all approaches that understand, represent and process the images, their segments and features. The representation and processing is determined by the selected fuzzy technique and the problem to be solved [35].

There are three main stages in fuzzy image processing: image fuzzification, modification of membership values, and, if necessary, image defuzzification. The Figure 3.5 shows these stages.

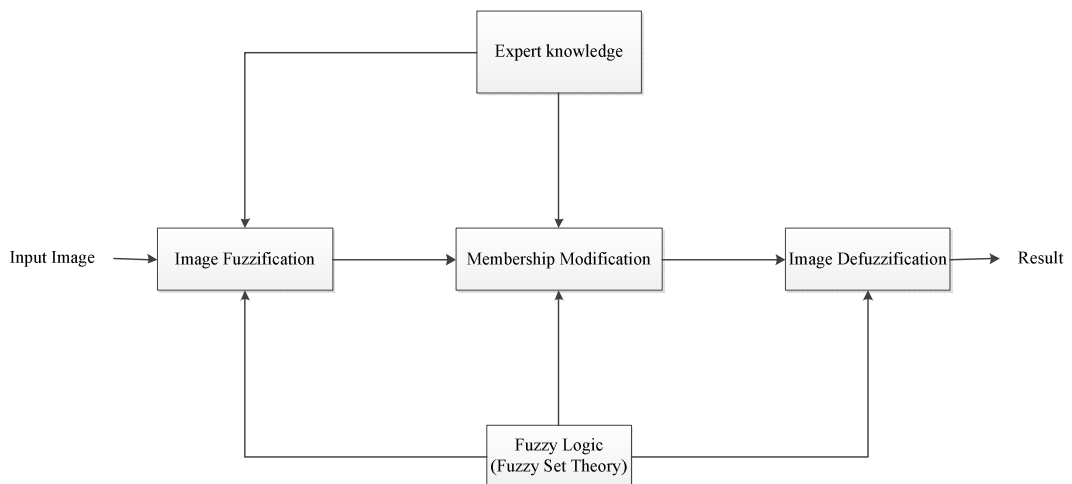


Figure 3. 5. Fuzzy Image Processing [36].

The fuzzification and defuzzification steps are due to the fact that we do not possess fuzzy hardware. Therefore, the coding of image data (fuzzification) and decoding of the results (defuzzification) are steps that make possible to process images with fuzzy techniques. The main power of fuzzy image processing is in the middle step (modification of membership values). After the image data are transformed from

gray-level plane to the membership plane (fuzzification), appropriate fuzzy techniques modify the membership values. This can be a fuzzy clustering; a fuzzy rule-based approach, a fuzzy integration approach, and so on.

The most important of the needs of fuzzy image processing are as follows:

1. Fuzzy techniques are powerful tools for knowledge representation and processing
2. Fuzzy techniques can manage the vagueness and ambiguity efficiently
3. In many image-processing applications, we have to use expert knowledge to overcome the difficulties (e.g., object recognition, scene analysis)

Fuzzy set theory and fuzzy logic offer us powerful tools to represent and process human knowledge in form of fuzzy if-then rules. On the other side, many difficulties in image processing arise because the data/tasks/results are uncertain. This uncertainty, however, is not always due to the randomness but to the ambiguity and vagueness. Beside randomness which can be managed by probability theory we can distinguish between three other kinds of imperfection in the image processing:

- Grayness ambiguity
- Geometrical fuzziness
- Vague (complex/ill-defined) knowledge

These problems are fuzzy in the nature. The question whether a pixel should become darker or brighter than it already is, the question where is the boundary between two image segments, and the question what is a tree in a scene analysis problem, all of these and other similar questions are examples for situations that a fuzzy approach can be the more suitable way to manage the imperfection.

So, Fuzzy Image Processing methods show promising result relative to classical methods dealing with some drawbacks in image processing tasks. This filter is not complex and it has maximum efficiency in calculation speed. So, it can be used for real time applications.

CHAPTER IV

REMOVING BLOCKING ARTIFACTS

In this chapter, a new adaptive algorithm is developed to remove the blocking artifacts from compressed images. The proposed deblocking algorithm consists of blocking artifact detection and deblocking filtering. The method starts with the block artifact detection, then continues finding type of the region, and finally finishes filtering of the blocking artifacts. The general block diagram is shown in Figure 4.1 and the flowchart of deblocking algorithm is shown in Figure 4.2. It is applied on all the 8x8 block boundaries first along the vertical edges followed by the horizontal edges.

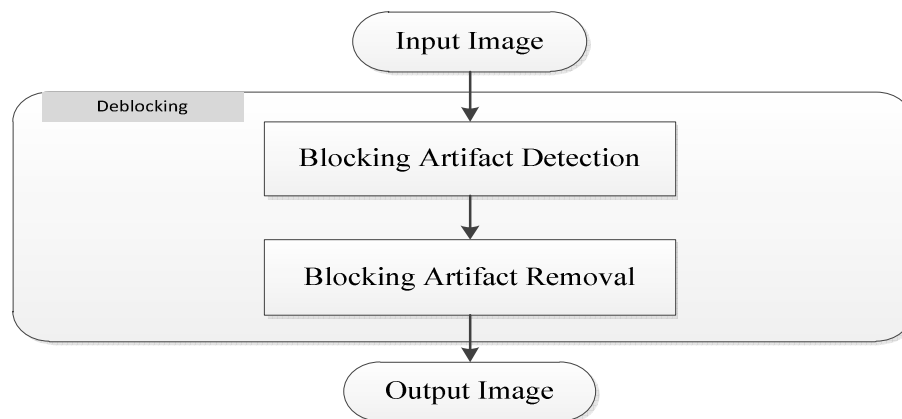


Figure 4. 1. The general flow of the proposed deblocking algorithm.

4.1. Blocking Artifact Detection

The first step of the proposed algorithm is detecting the blocking artifacts. Since the detected block strength indicates the existence of the artifacts and their influence, the detection accuracy is very important. And to avoid the blur in the texture and edge areas, the detection method should be able to distinguish the blocking artifacts from the object edges. There are three main regions according to our method in image.

- Smooth Region
- Transition Region
- Texture Region

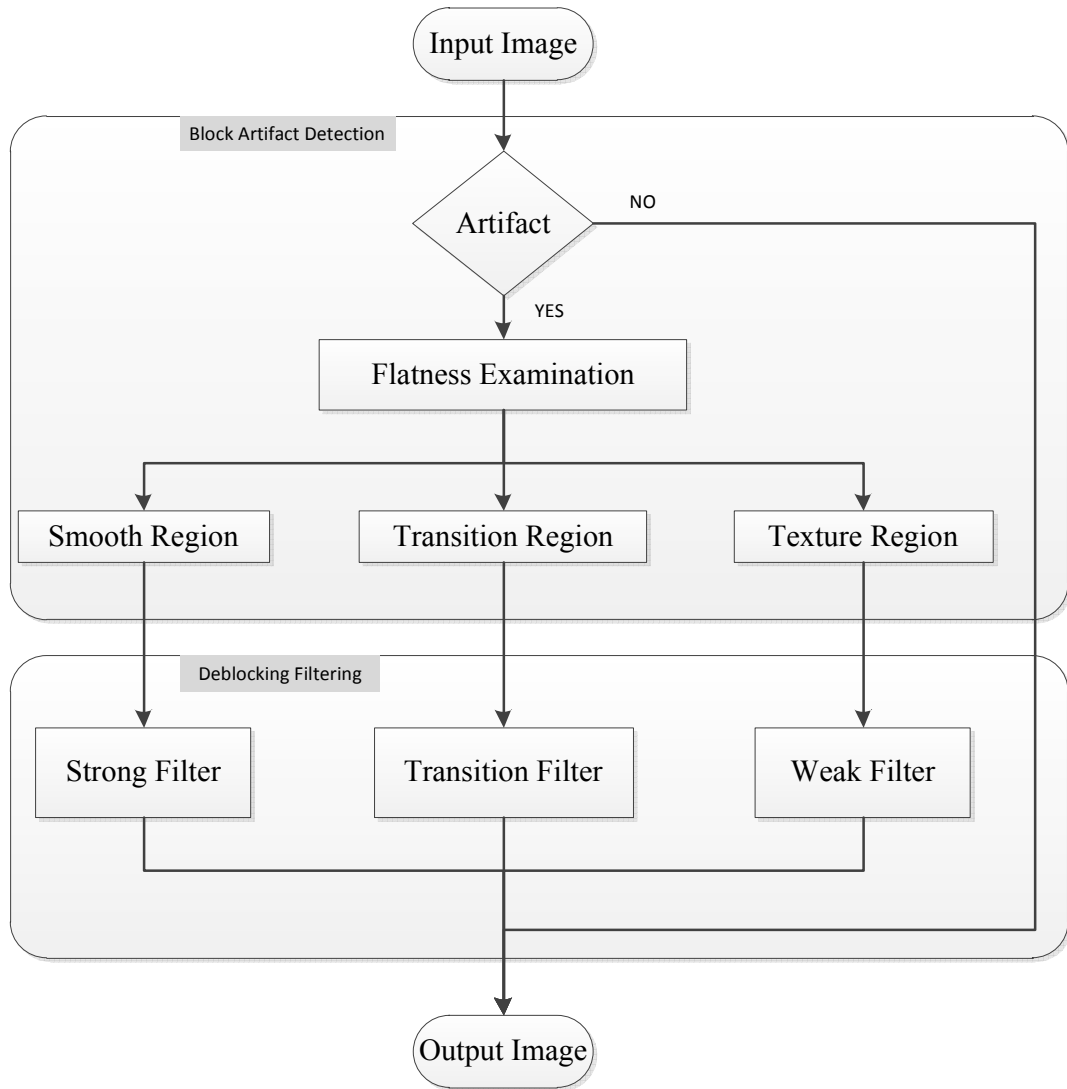


Figure 4. 2. The flowchart of the proposed deblocking algorithm.

In images, smooth region indicates areas which have low details, so it has low frequency. Texture region contains many edges, hence frequency is high. And transition region is between two regions. How well these regions are distinguished, we have the opportunity to remove artifacts without blurring.

Figure 4.3 shows pixels on block boundaries along 8x8 blocks used in detection the block artifacts. In each line, a vector v (4.1) is constructed for all boundaries.

$$v=[v_0 \ v_1 \ v_2 \ v_3 \ v_4 \ v_5 \ v_6 \ v_7 \ v_8 \ v_9] \quad (4.1)$$

Detection of the vertical and horizontal blocking artifacts has same algorithm, only selection of pixels differs as shown in Figure 4.3.

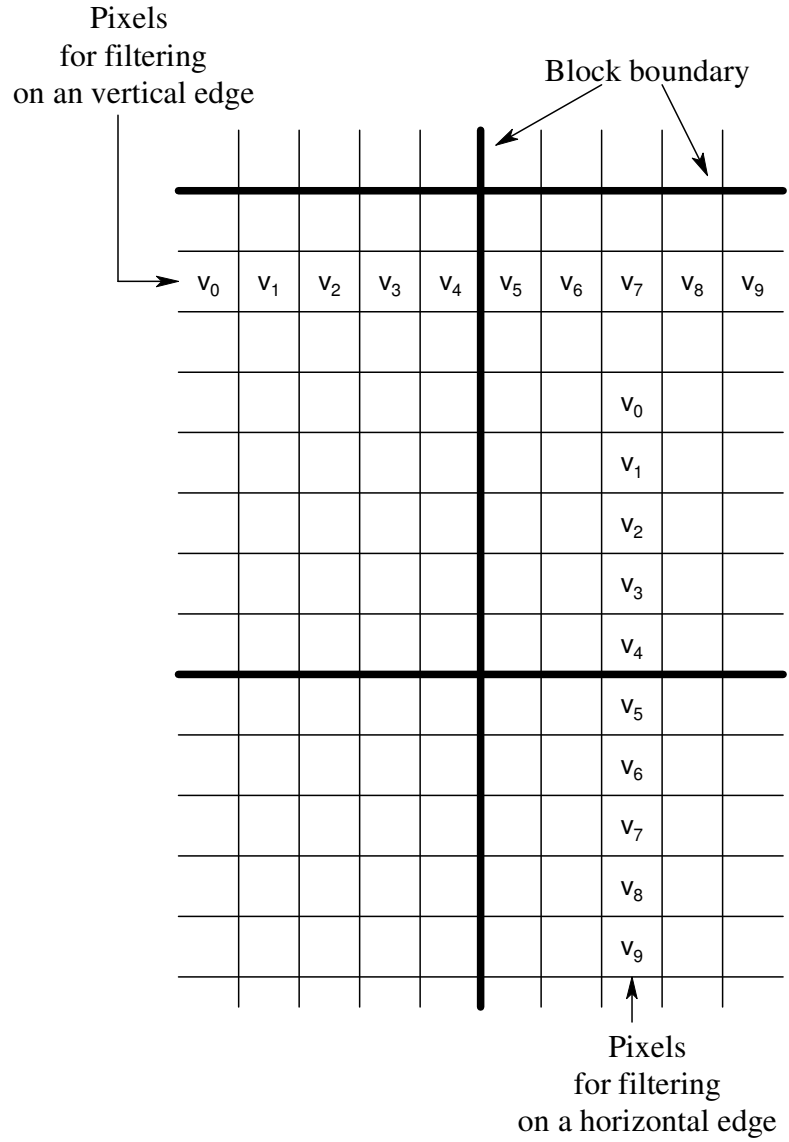


Figure 4. 3. Boundary areas around the block of interest in blocking artifact detection.

By using obtained vector v (4.1), the mean of each first five and last five components (4.2) are computed.

$$\begin{aligned} Mean_1 &= (v_0 + v_1 + v_2 + v_3 + v_4) / 5 \\ Mean_2 &= (v_5 + v_6 + v_7 + v_8 + v_9) / 5 \end{aligned} \tag{4.2}$$

The threshold T is found by calculating average of 64 pixels of neighboring blocks. The absolute value of the difference between these means is compared with T and the constant 2.6 is obtained using experimental study.

$$|\text{Mean}_1 - \text{Mean}_2| < 2.6 * T \quad (4.3)$$

If the equation 4.3 is not satisfied, there is no need to filter pixels in that region. On the contrary, it is understood existence of the artifacts and the algorithm proceeds second comparison which determines the regions. According to this, an absolute difference vector Δ (4.4) is formed for successive pixels in the vector v (4.1) except for the boundary pixels.

$$\Delta_i = |v_i - v_{i+1}| \quad i=0,1,2,3,5,6,7,8 \quad (4.4)$$

By finding the maximum value of the difference vector (4.4) and using experimental constants which are obtained using experimental studies, we obtain 3 different regions.

$$\text{Smooth Region} \quad : \quad \max(\Delta) \leq 2 \quad (4.5)$$

$$\text{Transition Region} \quad : \quad 2 < \max(\Delta) < 8 \quad (4.6)$$

$$\text{Texture Region} \quad : \quad 8 \leq \max(\Delta) \quad (4.7)$$

4.2. Deblocking Filtering

After the blocking artifacts detection and determining the regions, an adaptive fuzzy filter is applied to the pixels for a specified interval considering the detection and the region in each row to the remove blocking artifacts consecutively. For the vertical blocking artifacts, the fuzzy filter based on fuzzy transformation theory [34] is defined as:

$$v_{B_x} = \left(\sum_{j=-a}^a v(x, y+j) w_j \right) / \left(\sum_{j=-a}^a w_j \right) \quad (4.8)$$

$$v_{B_x} = \left(\sum_{j=-a}^a v(x, y+j) \mu [d(x, y+j)] \right) / \left(\sum_{j=-a}^a \mu [d(x, y+j)] \right) \quad (4.9)$$

$$d(x, y+j) = |v(x, y) - (v(x, y) + v(x, y+j)) / 2| \quad (4.10)$$

Similar algorithm is applied to the each column in the determined region to remove horizontal blocking artifacts as follows:

$$v_{B,y} = \left(\sum_{i=-a}^a v(x+i,y) w_i \right) / \left(\sum_{i=-a}^a w_i \right) \quad (4.11)$$

$$v_{B,y} = \left(\sum_{i=-a}^a v(x+i,y) \mu [d(x+i,y)] \right) / \left(\sum_{i=-a}^a \mu [d(x+i,y)] \right) \quad (4.12)$$

$$d(x+i,y) = \left| v(x,y) - (v(x,y) + v(x+i,y)) / 2 \right| \quad (4.13)$$

where;

a : Window size

$v(x,y)$: The value of image in location (x,y)

$v_B(x,y)$: The deblocking result of pixel in location (x,y)

$$\mu(x) = e^{(-x/\xi)} \quad (4.14)$$

μ is an exponential membership function to describe the relationship between pixels by their distance. Here, ξ is the spread parameter. The larger it is, the stronger is the smoothing effect. According to the characteristics of the regions determined by the deblocking part, both window size and the spread parameter are adjusted. So, the fuzzy filter (4.9) becomes adaptive.

Expressions in (4.8), (4.9), (4.10) are almost identical with expressions in (4.11), (4.12), (4.13) except for dimensions.

In the following subsections, the regions are clearly described and the values of the parameters are determined experimentally.

4.2.1. Filtering on Smooth Region (Intensive Degradation)

As a result of 8x8 block based coding, low frequency areas in the image are affected much more. Also, human visual system is more sensitive to the blocking artifacts in this region. So, the filtering has to be strong and covers a larger area. Experimental studies show that best filtering results are obtained by choosing $a = 4$ and $\xi = 44$ and applying the filter (4.8), (4.11) on pixels in the vector v (4.1) between v_1 and v_8 . Figure 4.4 shows affect of the spread parameter. By varying values of the difference between the pixels, the corresponded result of the membership function (4.14) is represented.

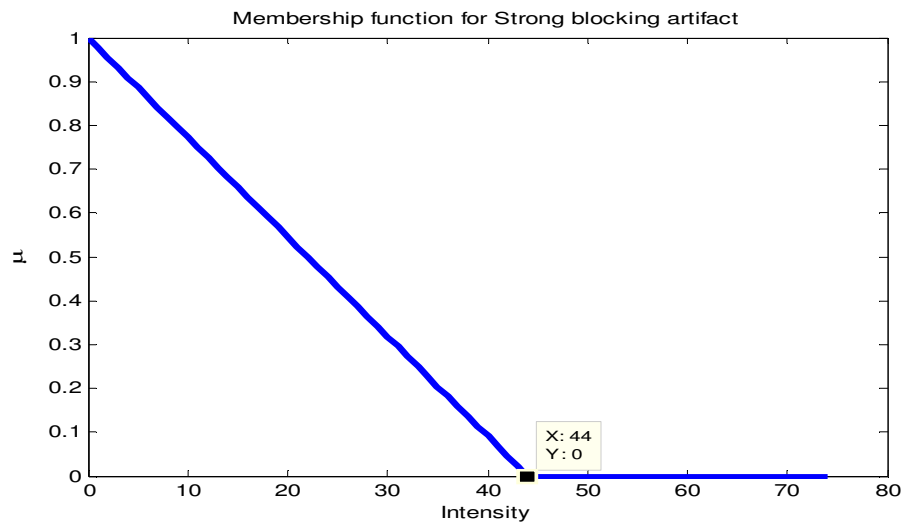


Figure 4. 4. Membership function for Strong Blocking Artifact

4.2.2. Filtering on Texture Region (Low Degradation)

The Texture region contains edges and textures. So, frequency is very high. It has an importance not to blur these oscillations of intensity. Therefore, filtering has to be weak and covers pixels near border. So, again with the experimental study, parameters are obtained for the filter (4.8), (4.11) and the membership function (4.14). The values are; $a = 1$, $\xi = 35$ and this filter are performed for pixels in the vector v (4.1) between v_3 and v_6 . Figure 4.5 represents affect of the spread parameter.

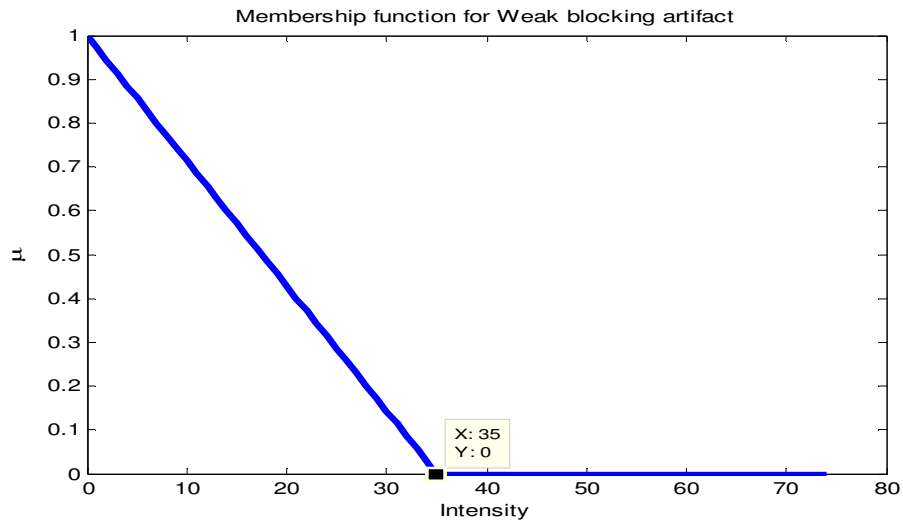


Figure 4. 5. Membership function for Weak Blocking Artifact

4.2.3. Filtering on Transition Region (Moderate Degradation)

Areas between smooth and texture regions are evaluated as transition regions. Because, it shows a blend of two. Accordingly, parameters of filter (4.8), (4.11) and the membership function (4.14) are determined by experiments. The obtained values are; $a = 2$, $\xi = 39$ and this filter are used for pixels in the vector v (4.1) between v_2 and v_7 . Figure 4.6 exhibits affect of the spread parameter.

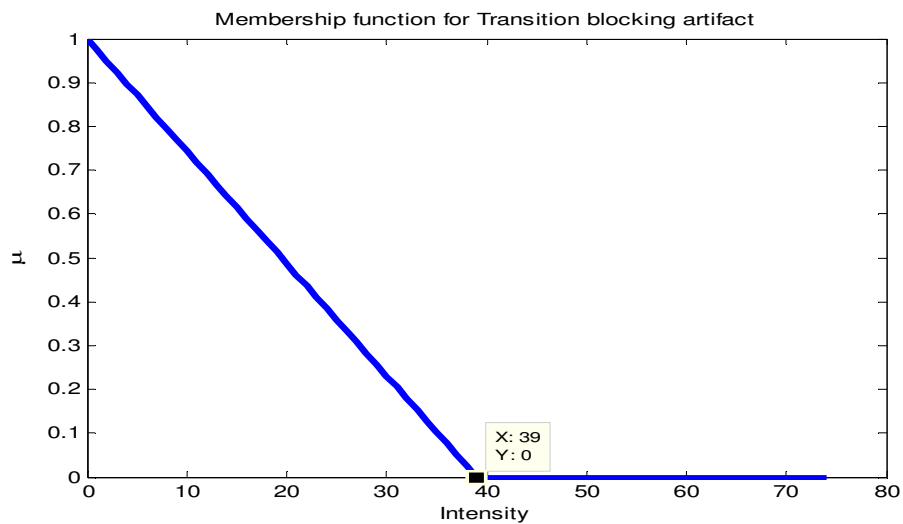


Figure 4. 6. Membership function for Transition Blocking Artifact.

CHAPTER V

REMOVING RINGING ARTIFACTS

This chapter aims to remove the ringing artifacts from compressed images. The algorithm of ringing artifact removal can be used independently or following the deblocking algorithm. Unlike the deblocking algorithm, deringing needs to have two dimensional fuzzy filtering. Because the ringing artifacts do not have to be only along horizontal and vertical direction. They are arbitrary depending on image. Due to the nature of the ringing artifacts explained in chapter 2, artifacts are generally found around edges in the image. So, this method starts with edge detection, then is followed by classification of 8x8 blocks by the means of complementary ringing detection, and finally terminates by deringing filtering. The general block diagram is shown in Figure 5.1 and the flowchart of deblocking algorithm is shown in Figure 5.2.

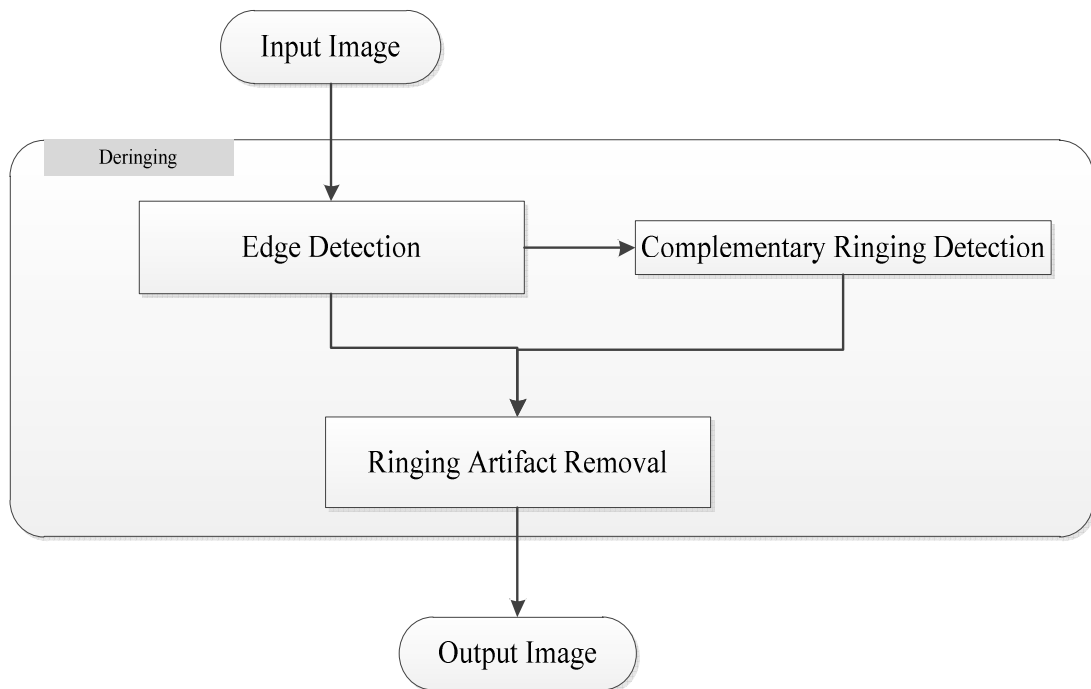


Figure 5. 1. The general flow of the proposed deringing algorithm.

5.1. Edge Detection

The goal of edge detection is to mark the points in an image at which the intensity changes sharply. Abrupt changes in image usually reflect features of images. Also, this tool is very important in compressed images. Because ringing artifacts always arise around the object edges, precise edge detection is necessary. Edge detection is the first step of deringing algorithm. Here, the fuzzy edge detection method [38] is adopted.

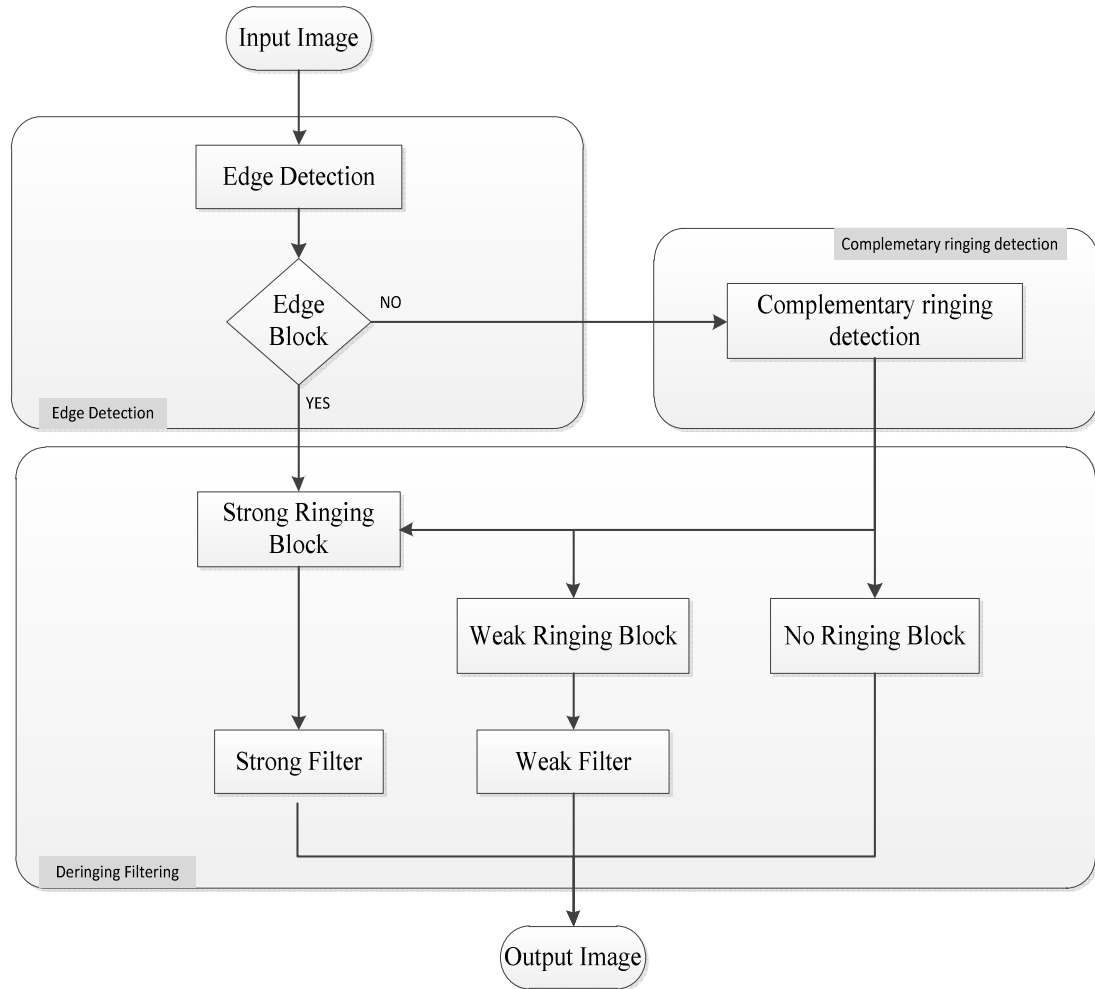


Figure 5. 2. The flowchart of the proposed deringing algorithm.

First, a difference matrix d (5.1) is built by calculating maximum difference in a 3x3 window around each pixel.

$$d(x,y)=\max (|v(x,y)-v(x+i,y+i)|) \quad i=-1,0,1; j=-1,0,1 \quad (5.1)$$

where, $v(x,y)$ is the pixel value of the image in location (x,y) .

Then, a difference histogram vector H of the image is computed based on the statistics of the difference matrix. That is, $H(i)$ denotes the number of pixels whose maximum difference in gray level value is i .

Using this statistics, the summation S (5.2) of maximum differences $d(x,y)$ is also found.

$$S = \sum_{x=1}^M \sum_{y=1}^N d(x, y) \quad (5.2)$$

where M is the number of rows and N is the number of columns. And total number of pixels T (5.3) is determined.

$$T = M \times N \quad (5.3)$$

Then, by using difference histogram H and following rule (5.4), a parameter Th is set to the edge detection threshold.

$$\frac{1}{S} \sum_{i=0}^{Th} i \cdot H(i) \leq \frac{1}{T} \sum_{i=0}^{Th} H(i) \leq \frac{1}{S} \sum_{i=0}^{Th+1} i \cdot H(i) \quad (5.4)$$

After that, Sobel operators (5.5) are applied to the image.

$$G_x = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}, G_y = \begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix} \quad (5.5)$$

By masking Sobel operators, edge detection is completed by forming the edge pixel map with the threshold Th found above.

$$\text{edge}(x, y) = \begin{cases} 1, & \text{sobel}(x, y) \geq Th \\ 0, & \text{sobel}(x, y) < Th \end{cases} \quad (5.6)$$

Detected edge pixels are strong and clear. If the block contains edge pixels, it is regarded as a strong ringing block, since the ringing artifacts emerge around edges most probably. So, strong filter is necessary.

5.2. Complementary Ringing Detection

Although edge detection can locate most strong the ringing artifacts, in the area where ringing artifacts are serious, the edge detection may miss the edges masked by artifacts. Furthermore, due to motion estimation, the no-edge blocks next to the edge blocks may also contain the ringing artifacts. As a result, complementary ringing detection needs to be applied to recover them.

Complementary ringing detection works on the 8 adjacent no-Edge blocks of edge blocks. The detection is only made on no-Edge blocks. Figure 5.3 shows an example. There are 8 adjacent blocks of edge Block 0. In this example, Block 3,4,5,6,7 and 8 are possible blocks which may contain the ringing artifacts.

Block 6 No-Edge	Block 7 No-Edge	Block 8 No-Edge
Block 5 No-Edge	Block 0 Edge	Block 1 Edge
Block 4 No-Edge	Block 3 Edge	Block 2 Edge

Figure 5. 3. The 8 adjacent blocks of interest in complementary ringing detection.

For the possible blocks, the variance $\text{var}(x,y)$ (5.7) in a 3x3 window around each pixel are calculated and the maximum one is selected as the variance of the block (5.8).

$$\begin{cases} m(x, y) = \frac{1}{9} \sum_{j=-1}^1 \sum_{i=-1}^1 v(x+i, y+j) \\ \text{var}(x, y) = \frac{1}{9} \sum_{j=-1}^1 \sum_{i=-1}^1 (v(x+i, y+j) - m(x, y))^2 \end{cases} \quad (5.7)$$

$$\text{STD}_B = \sqrt{\max\{\text{var}(x, y)\}} \quad (x, y) \in \text{Block} \quad (5.8)$$

Then, each detected blocks are classified by using STD_B (5.8) with a set of predetermined threshold.

$$Ringing_Artifact = \begin{cases} \text{Strong} & H_Th \leq STD_B \\ \text{Weak} & L_Th \leq STD_B < H_Th \\ \text{No} & STD_B < L_Th \end{cases} \quad (5.9)$$

where;

$$H_th = \frac{\left(\frac{Th}{8}\right)^2}{\sqrt{2}} \quad (5.10)$$

$$L_th = \max\left(\frac{Th}{16}, H_th - 100\right) \quad (5.11)$$

Th : The gradient threshold determined from (5.4)

H_th : The upper threshold

L_th : The lower threshold

Since the gradient threshold of Sobel Th presents characteristics of the image, the upper threshold (5.10), the lower threshold (5.11) and the relationship (5.9) are defined by experiments.

In (5.9), Strong, Weak and No means that the block has strong, weak and no ringing artifacts respectively. Now, all the ringing blocks are detected.

5.3. Deringing Filtering

According to two step detection, a two dimensional adaptive fuzzy filter (5.12) is adopted on the ringing blocks to remove ringing artifacts. The filtering does not cover edge pixels in the image. Because if they are not excluded, blurring may exist around edges in the image.

$$v_R(x, y) = \frac{\sum_{j=-a}^a \sum_{i=-a}^a v(x+i, y+j) w_{ij}}{\sum_{j=-a}^a \sum_{i=-a}^a w_{ij}} \quad (5.12)$$

$$v_R(x, y) = \frac{\sum_{j=-a}^a \sum_{i=-a}^a v(x+i, y+j) \mu[d(x+i, y+j)]}{\sum_{j=-a}^a \sum_{i=-a}^a \mu[d(x+i, y+j)]} \quad (5.13)$$

where;

a : Window size

$v(x, y)$: The value of image in location (x, y)

$v_R(x, y)$: The deringing result of pixel in location (x, y)

$$d(x+i, y+j) = |v(x+i, y+j) - v(x, y)| \quad (5.14)$$

$$\mu(x) = e^{(-x/\xi)} \quad (5.15)$$

μ is the same membership function with the (4.14) to describe the relationship between pixels by their distance. Here, ξ is the spread parameter. Similarly, effect of smoothing increases as it gets large values. Again, both window size and spread parameter are adjusted for the strong and weak ringing artifacts. So, the fuzzy filter (5.12) becomes adaptive.

5.3.1. Filtering on Strong Ringing Artifacts

These kind of ringing artifacts are more intensive and they are generally located very close to edges in the image. So, the fuzzy filter has to be strong to eliminate them. Also, the gradient threshold Th gives an idea for the image. Accordingly, the spread parameter is chosen as $Th/8$ and filtering range is adjusted to $a=4$ as a result of experimental study.

$$\xi = \frac{Th}{8} \quad (5.16)$$

5.3.2. Filtering on Weak Ringing Artifacts

Although weak ringing artifacts are observed near edges in the image, they are not prominent. However, it is still a problem in terms of visual quality. Consequently, the spread parameter is selected as $Th/16$ and filtering range is determined to $a=2$ by experimental study.

$$\xi = \frac{Th}{16} \quad (5.17)$$

CHAPTER VI

EXPERIMENTAL RESULTS

This chapter presents the results that are obtained using the coding artifact reduction systems based on algorithms described in chapter 4 and 5. These results are intended to demonstrate the success of them. Before giving results of experiments, determination of picture quality is defined for understanding and evaluating better. After that, various experimental results and comparisons are presented.

6.1. Picture Quality Evaluation

Picture quality is a term used to measure the fidelity of the perceived picture quality [39]. When people say that image quality is good or bad, it usually means how they appreciate the image psychovisually. Compressed image may be considered as a degraded version. In this case, the image quality evaluation is needed to measure how similar the post-filtered image is with the original image which is not compressed.

There are generally two approaches measuring picture quality: Subjective Quality Assessment (SQA) and Objective Quality Assessment (OQA) [40]. SQA involves the evaluation of picture quality using human subjects. OQA measures the statistical aspects of physical picture signal, e.g., error and variance.

The objective picture quality evaluation uses objective metrics to measure the picture quality or impairment. While the subjective assessed results may vary due to the change of subjects or subjects' opinions, results of using objective quality metrics are unique in repeated experiments.

Objective quality metrics can be classified into three categories, i.e., Full Reference (FR) metrics, No-Reference (NR) metrics and Reduced-reference (RR) metrics [41]. This classification is based on the availability of the reference/original images and video used in the quality metrics. In FR metrics, the reference video is fully available for quality computation, usually frame by frame or pixel by pixel. In NR metrics, no reference signal is available for quality computation. This is suitable for the

applications such as the video receiving and video decoding systems. In this work, some FR metrics are sufficient to evaluate the performance of methods.

There is also important thing to know in evaluation for compression. It is bits per pixel bpp. It is the number of bits of information stored per pixel of an image. The more bits there are, the more colors are represented. For gray scale original image, normally 8 bits is used for one pixel. If it is compressed, the bpp decreases according to the degree of compression. The bpp (6.1) is calculated as follows;

$$\text{bpp} = \frac{\text{Total_number_of_bits_of_the_image}}{\text{Total_number_pixels_in_the_image}} \quad (6.1)$$

6.2. Quality Metrics

The most common FR quality metrics are the mean squared error (MSE) and the peak signal to noise ratio (PSNR). The MSE (6.2) is an error-based metric defined as

$$\text{MSE} = \frac{1}{MN} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} \left(v(x,y) - \hat{v}(x,y) \right)^2 \quad (6.2)$$

where;

M : The numbers of rows (height) of an image

N : The numbers of columns (width) of an image

$v(x,y)$: The pixel of the reference image in location (x,y)

$\hat{v}(x,y)$: The pixel of the degraded image in location (x,y)

The PSNR metric (6.3) is a derivative of the MSE, in decibel (dB).

$$\text{PSNR} = 10 \log_{10} \frac{L_{\max}^2}{\text{MSE}} \quad (6.3)$$

where L_{\max} denotes the maximum value. If 8-bit precision is used, then L_{\max} is equal to 255.

The MSE and the PSNR can reflect how physically close to the degraded image and the reference image are. However, they do not always correlate well with perceived quality [42][43]. Figure 6.1 shows different degraded images with same MSE. As it is seen from images, we also need another quality tool.

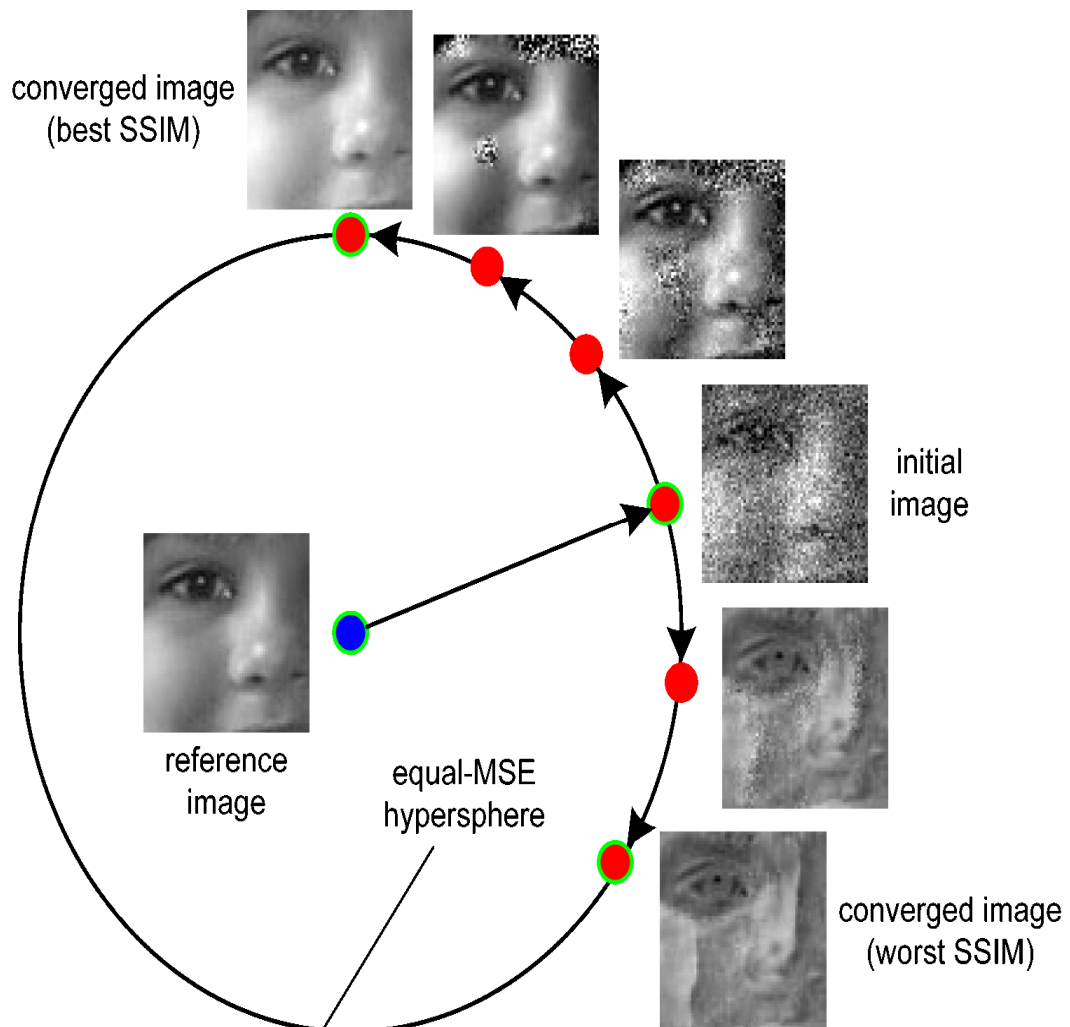


Figure 6. 1. Images corrupted with different noise and artifacts with identical MSE.

It has been believed that the human subjects tend to assess the image quality by inspecting the structures and detail features rather than checking the pixels one after another. The Structural Similarity Index (SSIM) [44] is used to compare the similarity of the structural information (i.e, the mean, variance, and covariance of small patches or blocks) extracted from both a test image and a reference image. SSIM is designed to improve on traditional methods like PSNR and MSE.

The SSIM metric is calculated on various windows of an image. The measure between two windows x and y of common size $N \times N$ is:

$$SSIM(x,y)=\frac{(2\mu_x\mu_y+c_1)(2\sigma_{xy}+c_2)}{(\mu_x^2+\mu_y^2+c_1)(\sigma_x^2+\sigma_y^2+c_2)} \quad (6.3)$$

where;

μ_x is the average of x.

μ_y is the average of y.

σ_x is the variance of x.

σ_y is the variance of y.

σ_{xy} the covariance of x and y.

$c_1 = (k_1L)^2$, $c_2 = (k_2L)^2$ are two variables to stabilize the division with weak denominator.

L is the dynamic range of the pixel values ($L=2^{\text{Number_of_bits/pixel}}-1$)

$k_1 = 0.01$ and $k_2 = 0.03$ by default.

After applying this metric on the image, the resultant SSIM index is a decimal value between -1 and 1. The value 1 is only reachable in the case of two identical sets of data. Typically, it is calculated on window size of 8x8. The window can be displaced pixel by pixel on the image.

6.3. The Results for Images from Experiments

To evaluate the performance of the deblocking and deringing algorithms, PSNR and SSIM are used. Here, results of the methods developed are only given. Experiments are made using following images: Lena, Goldhill, Airplane, Peppers, and Mandrill. All test images are 8-bit grayscale images and have 512x512 pixels. These images are presented in Figure 6.2.



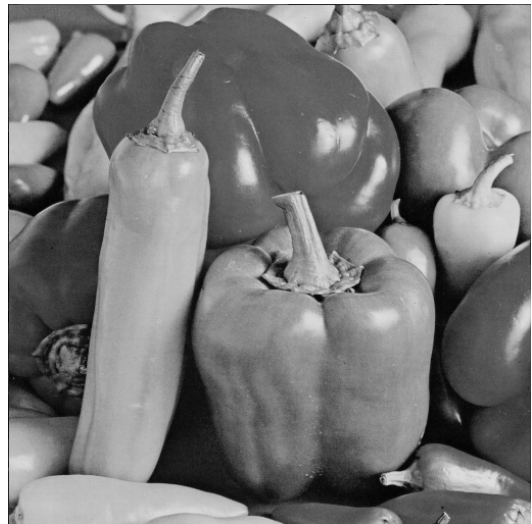
(a) Lena



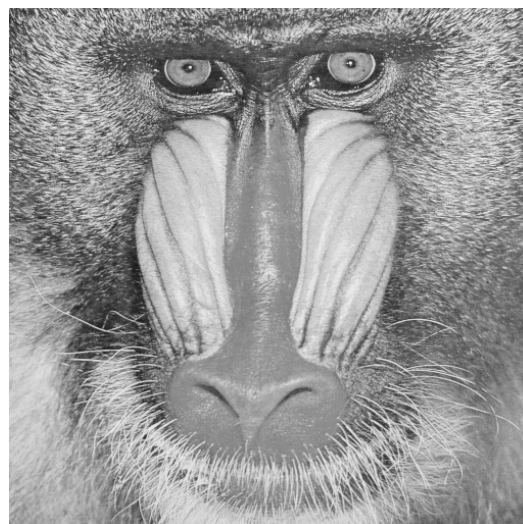
(b) Goldhill



(c) Airplane



(d) Peppers



(e) Mandrill

Figure 6. 2. Images corrupted with different noise and artifacts with identical MSE

Table 6.1 – 6.5 shows objective results of the proposed algorithms for different compression ratios (especially low bit rates) on various test images. Figure 6.3 – 6.7 are plotted by using these objective results.

Table 6. 1. Objective results of proposed algorithms for different compression ratios for the Lena.

QF (Matlab)	bpp	PSNR (Compressed)	PSNR (Filtered)	SSIM (Compressed)	SSIM (Filtered)
0,00	0,1335	24,25	25,70	0,6417	0,7197
1,00	0,1335	24,25	25,70	0,6417	0,7197
2,00	0,1336	24,25	25,70	0,6420	0,7199
3,00	0,1429	24,84	26,22	0,6732	0,7416
4,00	0,1571	26,47	27,79	0,7176	0,7782
5,00	0,1731	27,33	28,71	0,7367	0,8008
6,00	0,1888	28,25	29,63	0,7623	0,8189
7,00	0,2034	28,89	30,19	0,7792	0,8293
8,00	0,2166	29,47	30,69	0,7953	0,8393
9,00	0,2310	29,95	31,14	0,8068	0,8474
10,00	0,2446	30,41	31,54	0,8180	0,8547
11,00	0,2584	30,77	31,85	0,8271	0,8593
12,00	0,2709	31,09	32,11	0,8340	0,8634
13,00	0,2838	31,43	32,39	0,8419	0,8675
14,00	0,2953	31,68	32,57	0,8470	0,8702
15,00	0,3080	31,95	32,78	0,8532	0,8733
16,00	0,3199	32,19	32,95	0,8584	0,8754
17,00	0,3321	32,40	33,10	0,8623	0,8772
18,00	0,3433	32,62	33,26	0,8665	0,8796
19,00	0,3531	32,79	33,37	0,8699	0,8811
20,00	0,3635	32,96	33,47	0,8735	0,8824
21,00	0,3750	33,14	33,61	0,8768	0,8843
22,00	0,3850	33,30	33,69	0,8796	0,8856
23,00	0,3947	33,42	33,78	0,8820	0,8866
24,00	0,4055	33,57	33,90	0,8847	0,8879
25,00	0,4145	33,70	33,98	0,8868	0,8888
26,00	0,4256	33,82	34,03	0,8890	0,8897
27,00	0,4341	33,94	34,10	0,8908	0,8907
28,00	0,4450	34,08	34,18	0,8929	0,8916
29,00	0,4530	34,17	34,24	0,8943	0,8923
30,00	0,4626	34,28	34,31	0,8961	0,8933

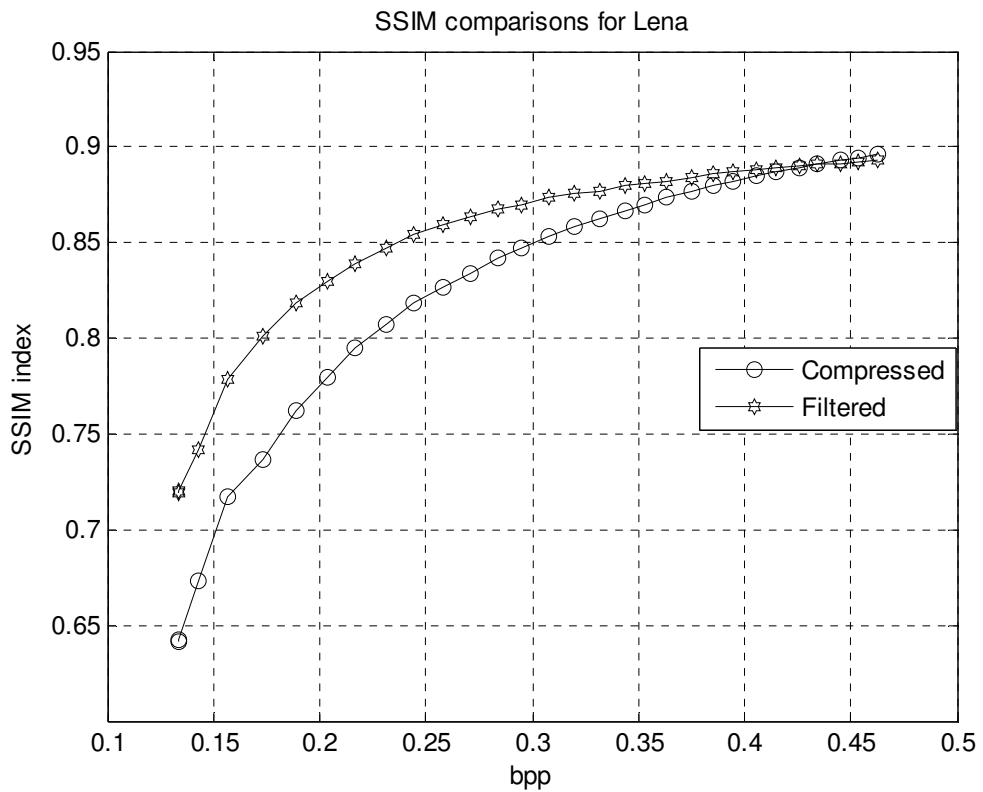
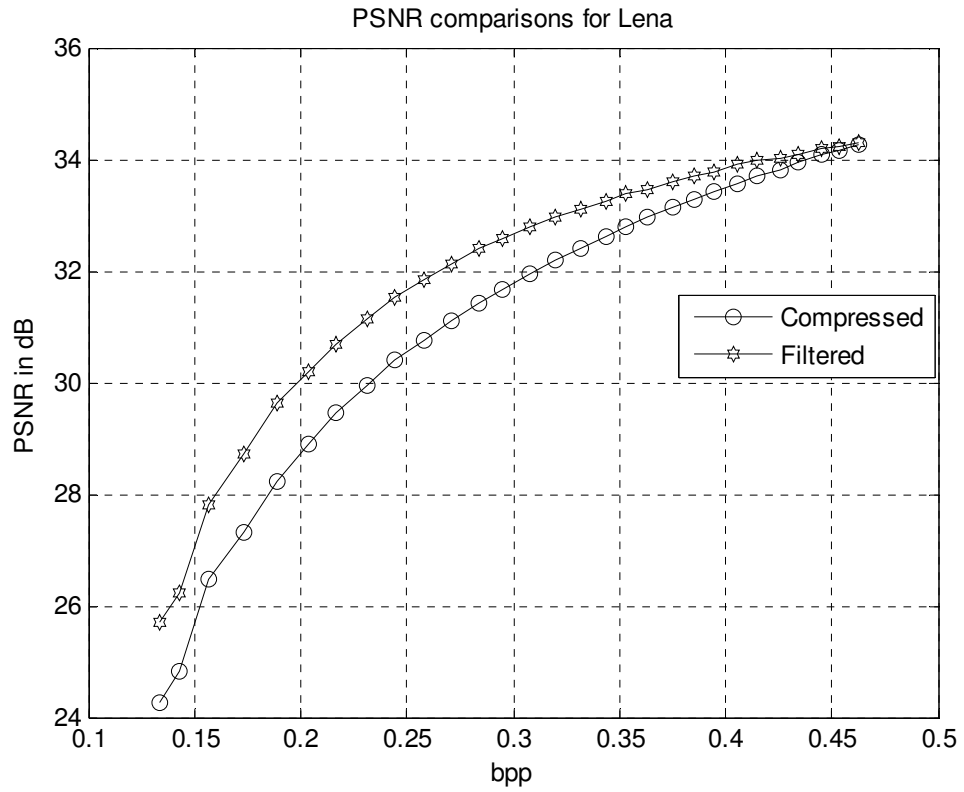


Figure 6. 3. PSNR and SSIM performance of the proposed algorithms for the Lena.

Table 6. 2. Objective results of proposed algorithms for different compression ratios for the Goldhill.

QF (Matlab)	bpp	PSNR (Compressed)	PSNR (Filtered)	SSIM (Compressed)	SSIM (Filtered)
0,00	0,1255	23,74	24,92	0,5153	0,5772
1,00	0,1255	23,74	24,92	0,5153	0,5772
2,00	0,1256	23,74	24,93	0,5157	0,5777
3,00	0,1353	24,15	25,30	0,5442	0,6013
4,00	0,1522	25,30	26,36	0,5918	0,6410
5,00	0,1710	26,16	27,22	0,6268	0,6733
6,00	0,1903	26,87	27,85	0,6570	0,6964
7,00	0,2090	27,43	28,31	0,6824	0,7140
8,00	0,2273	27,90	28,72	0,7038	0,7301
9,00	0,2471	28,29	29,06	0,7206	0,7424
10,00	0,2655	28,65	29,37	0,7348	0,7526
11,00	0,2847	28,95	29,62	0,7487	0,7626
12,00	0,3027	29,23	29,84	0,7602	0,7700
13,00	0,3231	29,50	30,05	0,7701	0,7772
14,00	0,3403	29,72	30,23	0,7797	0,7836
15,00	0,3598	29,95	30,42	0,7880	0,7901
16,00	0,3773	30,16	30,58	0,7967	0,7954
17,00	0,3958	30,36	30,73	0,8035	0,7995
18,00	0,4130	30,56	30,85	0,8102	0,8031
19,00	0,4292	30,71	30,96	0,8160	0,8068
20,00	0,4448	30,87	31,06	0,8211	0,8094
21,00	0,4618	31,03	31,16	0,8265	0,8127
22,00	0,4766	31,18	31,25	0,8309	0,8153
23,00	0,4932	31,31	31,33	0,8357	0,8182
24,00	0,5078	31,45	31,41	0,8398	0,8202
25,00	0,5218	31,56	31,48	0,8432	0,8222
26,00	0,5365	31,68	31,55	0,8466	0,8242
27,00	0,5508	31,80	31,62	0,8499	0,8261
28,00	0,5660	31,92	31,68	0,8531	0,8275
29,00	0,5775	32,01	31,72	0,8554	0,8288
30,00	0,5904	32,10	31,77	0,8579	0,8302

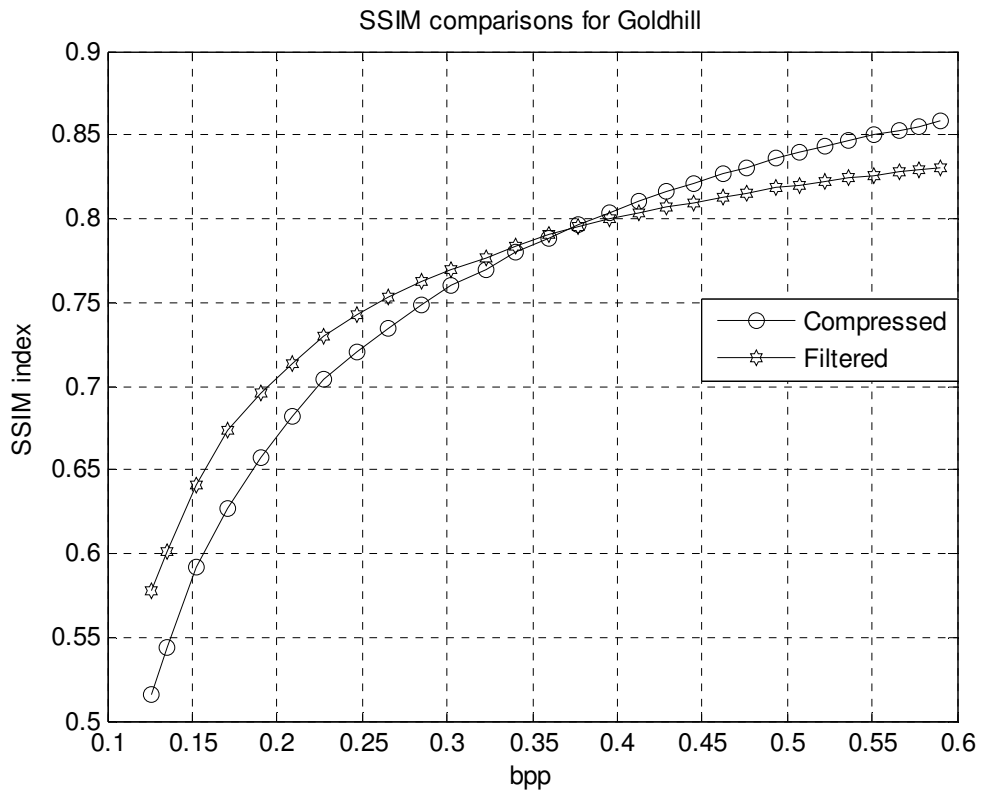
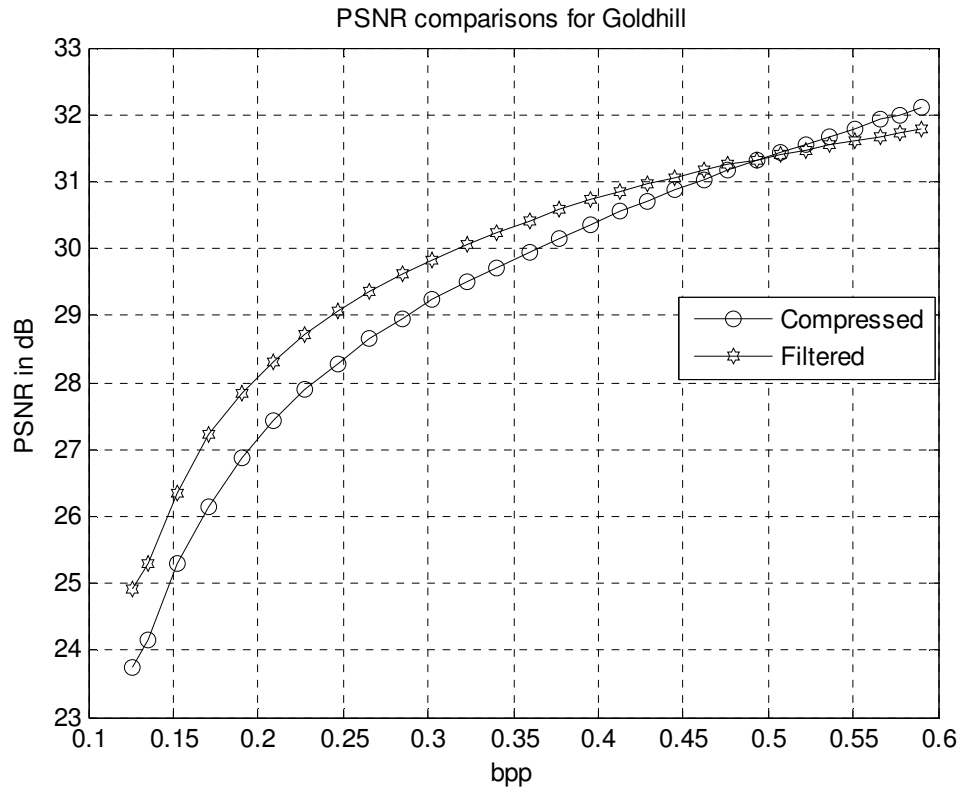


Figure 6. 4. PSNR and SSIM performance of the proposed algorithms for the Goldhill.

Table 6. 3. Objective results of proposed algorithms for different compression ratios for the Airplane.

QF (Matlab)	bpp	PSNR (Compressed)	PSNR (Filtered)	SSIM (Compressed)	SSIM (Filtered)
0,00	0,1411	23,39	24,48	0,6979	0,7581
1,00	0,1411	23,39	24,48	0,6979	0,7581
2,00	0,1411	23,39	24,49	0,6980	0,7581
3,00	0,1516	23,99	25,09	0,7176	0,7743
4,00	0,1689	25,48	26,60	0,7526	0,8057
5,00	0,1871	26,54	27,73	0,7742	0,8294
6,00	0,2051	27,45	28,63	0,7929	0,8437
7,00	0,2219	28,11	29,29	0,8076	0,8564
8,00	0,2379	28,71	29,89	0,8195	0,8662
9,00	0,2535	29,27	30,40	0,8343	0,8751
10,00	0,2685	29,77	30,84	0,8458	0,8831
11,00	0,2839	30,16	31,26	0,8513	0,8874
12,00	0,2977	30,49	31,54	0,8568	0,8901
13,00	0,3127	30,85	31,85	0,8648	0,8953
14,00	0,3256	31,13	32,09	0,8696	0,8988
15,00	0,3387	31,43	32,32	0,8759	0,9020
16,00	0,3527	31,69	32,49	0,8828	0,9047
17,00	0,3663	31,94	32,69	0,8874	0,9070
18,00	0,3792	32,17	32,86	0,8907	0,9087
19,00	0,3913	32,35	33,04	0,8925	0,9100
20,00	0,4031	32,56	33,20	0,8957	0,9106
21,00	0,4158	32,78	33,36	0,8985	0,9121
22,00	0,4281	32,97	33,50	0,9014	0,9133
23,00	0,4393	33,16	33,63	0,9043	0,9142
24,00	0,4496	33,32	33,74	0,9061	0,9154
25,00	0,4603	33,48	33,85	0,9083	0,9166
26,00	0,4716	33,63	33,97	0,9101	0,9177
27,00	0,4820	33,78	34,06	0,9121	0,9190
28,00	0,4936	33,93	34,17	0,9142	0,9201
29,00	0,5016	34,04	34,25	0,9159	0,9210
30,00	0,5121	34,17	34,31	0,9177	0,9215

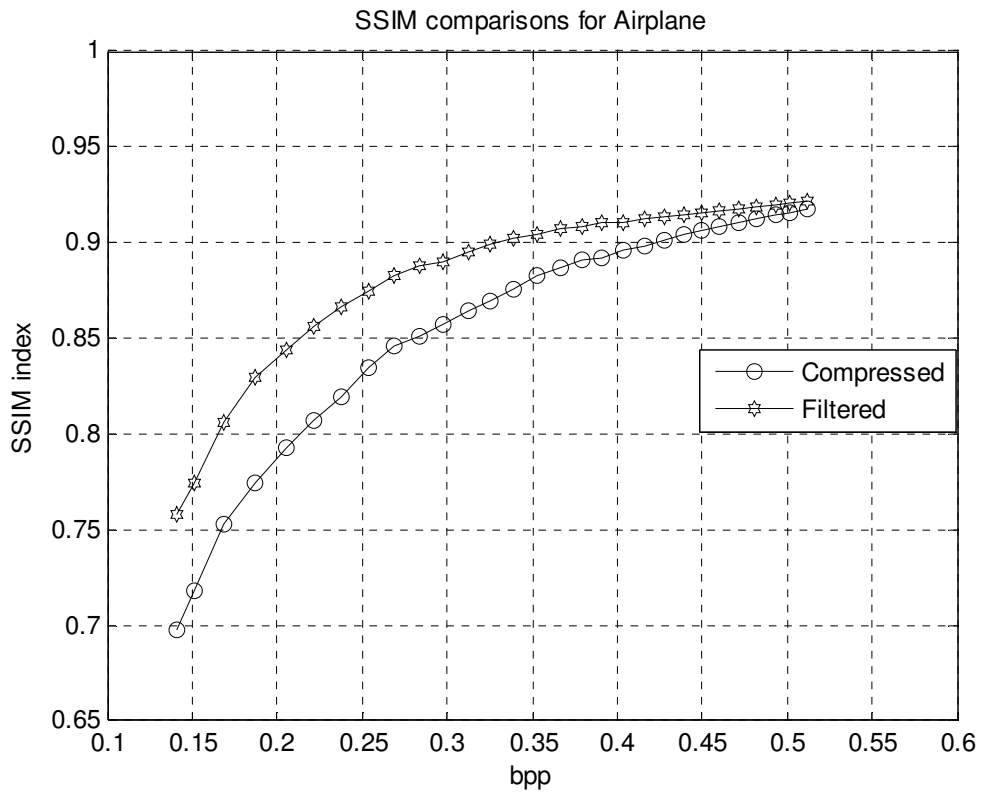
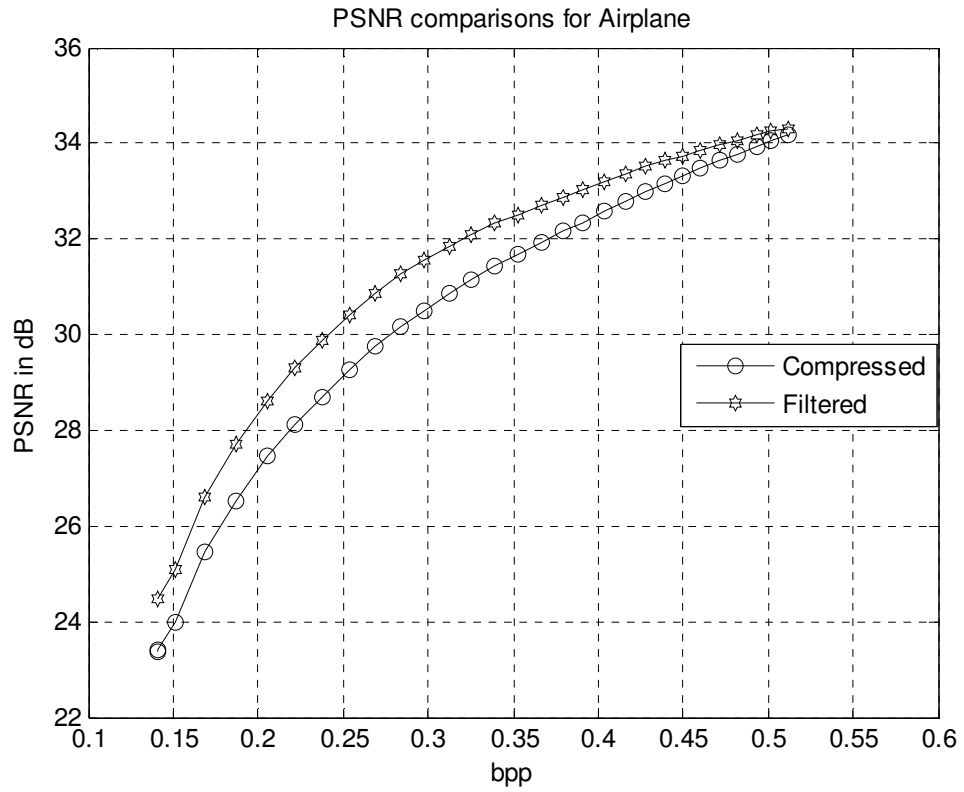


Figure 6. 5. PSNR and SSIM performance of the proposed algorithms for the Airplane.

Table 6. 4. Objective results of proposed algorithms for different compression ratios for the Peppers.

QF (Matlab)	bpp	PSNR (Compressed)	PSNR (Filtered)	SSIM (Compressed)	SSIM (Filtered)
0,00	0,1366	24,29	25,88	0,6254	0,7172
1,00	0,1366	24,29	25,88	0,6254	0,7172
2,00	0,1368	24,29	25,89	0,6256	0,7175
3,00	0,1458	24,82	26,32	0,6446	0,7283
4,00	0,1608	26,02	27,45	0,6743	0,7543
5,00	0,1762	27,18	28,66	0,7076	0,7819
6,00	0,1911	28,04	29,48	0,7298	0,7968
7,00	0,2050	28,68	30,09	0,7441	0,8067
8,00	0,2186	29,21	30,55	0,7598	0,8154
9,00	0,2336	29,71	30,98	0,7729	0,8222
10,00	0,2467	30,13	31,35	0,7824	0,8275
11,00	0,2597	30,46	31,61	0,7917	0,8317
12,00	0,2717	30,78	31,87	0,7990	0,8350
13,00	0,2836	31,07	32,07	0,8066	0,8380
14,00	0,2945	31,29	32,24	0,8117	0,8403
15,00	0,3066	31,53	32,44	0,8163	0,8426
16,00	0,3180	31,75	32,57	0,8218	0,8441
17,00	0,3300	31,93	32,73	0,8255	0,8461
18,00	0,3413	32,12	32,89	0,8288	0,8480
19,00	0,3508	32,26	33,00	0,8315	0,8488
20,00	0,3607	32,42	33,10	0,8346	0,8499
21,00	0,3722	32,57	33,22	0,8374	0,8515
22,00	0,3824	32,71	33,31	0,8404	0,8525
23,00	0,3926	32,83	33,40	0,8429	0,8538
24,00	0,4035	32,95	33,49	0,8451	0,8547
25,00	0,4142	33,05	33,54	0,8472	0,8554
26,00	0,4244	33,15	33,62	0,8492	0,8564
27,00	0,4346	33,26	33,68	0,8516	0,8572
28,00	0,4450	33,37	33,74	0,8537	0,8580
29,00	0,4528	33,44	33,80	0,8549	0,8588
30,00	0,4621	33,53	33,86	0,8568	0,8594

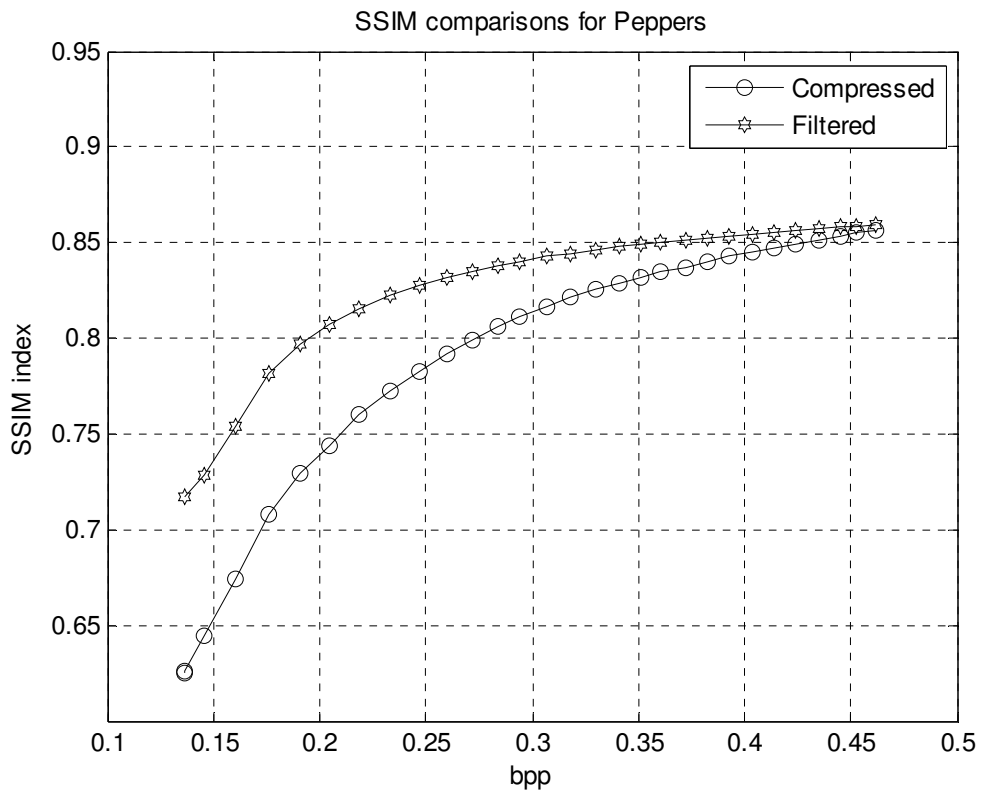
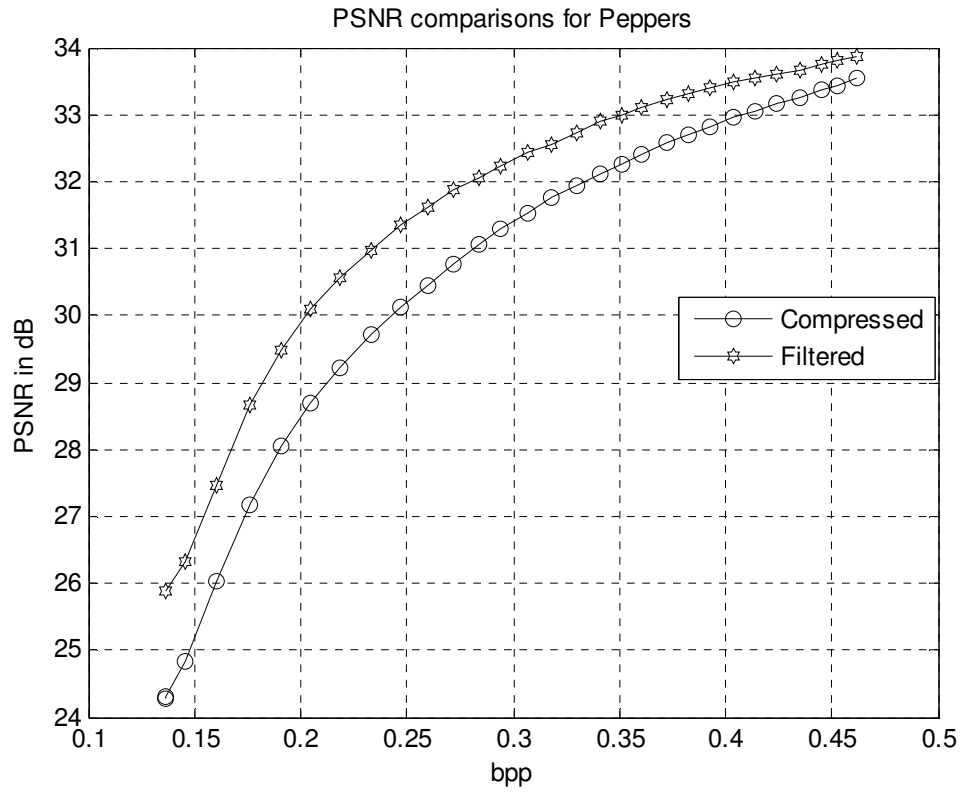


Figure 6. 6. PSNR and SSIM performance of the proposed algorithms for the Peppers.

Table 6. 5. Objective results of proposed algorithms for different compression ratios for the Mandrill.

QF (Matlab)	bpp	PSNR (Compressed)	PSNR (Filtered)	SSIM (Compressed)	SSIM (Filtered)
0,00	0,1530	19,93	20,50	0,3798	0,3952
1,00	0,1530	19,93	20,50	0,3798	0,3952
2,00	0,1530	19,93	20,50	0,3798	0,3952
3,00	0,1704	20,22	20,79	0,4195	0,4328
4,00	0,2105	20,91	21,46	0,4847	0,4948
5,00	0,2556	21,52	22,05	0,5388	0,5437
6,00	0,3001	22,05	22,53	0,5808	0,5802
7,00	0,3427	22,48	22,90	0,6142	0,6088
8,00	0,3820	22,83	23,18	0,6403	0,6290
9,00	0,4203	23,15	23,45	0,6623	0,6474
10,00	0,4572	23,43	23,66	0,6811	0,6619
11,00	0,4928	23,68	23,87	0,6977	0,6763
12,00	0,5262	23,91	24,05	0,7115	0,6874
13,00	0,5607	24,13	24,22	0,7245	0,6974
14,00	0,5896	24,32	24,36	0,7345	0,7049
15,00	0,6207	24,51	24,50	0,7449	0,7131
16,00	0,6506	24,68	24,62	0,7541	0,7197
17,00	0,6789	24,84	24,73	0,7629	0,7264
18,00	0,7075	25,00	24,85	0,7713	0,7333
19,00	0,7319	25,14	24,94	0,7773	0,7377
20,00	0,7580	25,27	25,02	0,7837	0,7421
21,00	0,7867	25,41	25,12	0,7900	0,7466
22,00	0,8128	25,54	25,21	0,7958	0,7515
23,00	0,8374	25,66	25,29	0,8014	0,7554
24,00	0,8621	25,78	25,37	0,8065	0,7592
25,00	0,8850	25,89	25,45	0,8109	0,7629
26,00	0,9091	26,01	25,53	0,8154	0,7665
27,00	0,9329	26,12	25,60	0,8197	0,7700
28,00	0,9594	26,24	25,69	0,8242	0,7736
29,00	0,9801	26,34	25,75	0,8278	0,7766
30,00	1,0003	26,45	25,81	0,8315	0,7790

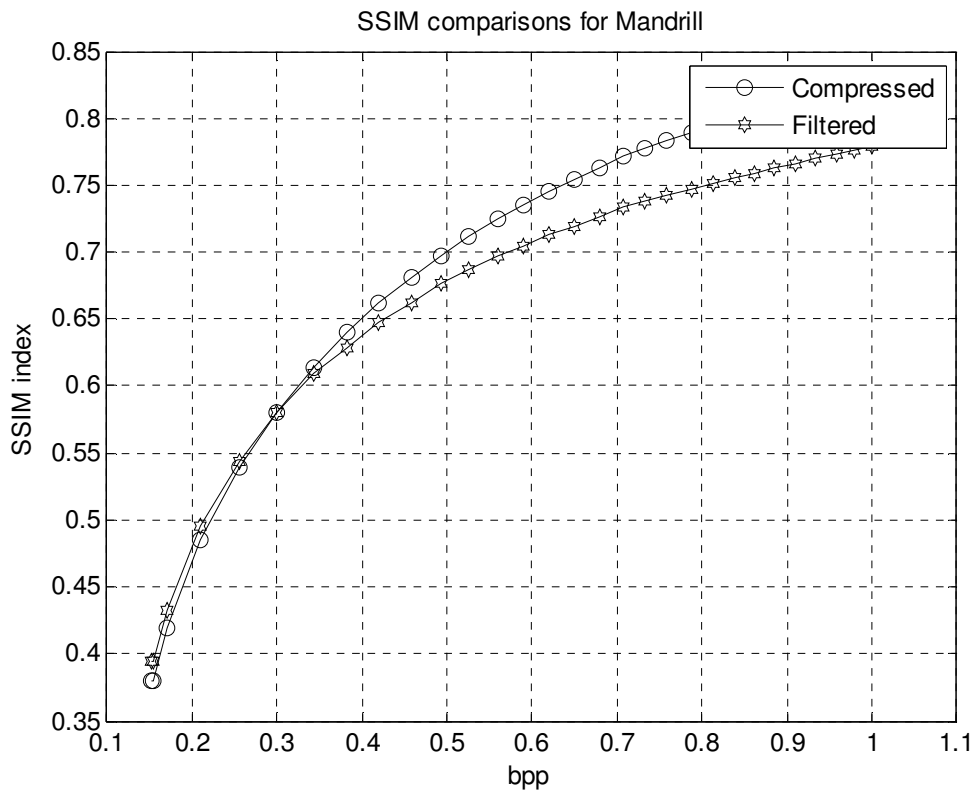
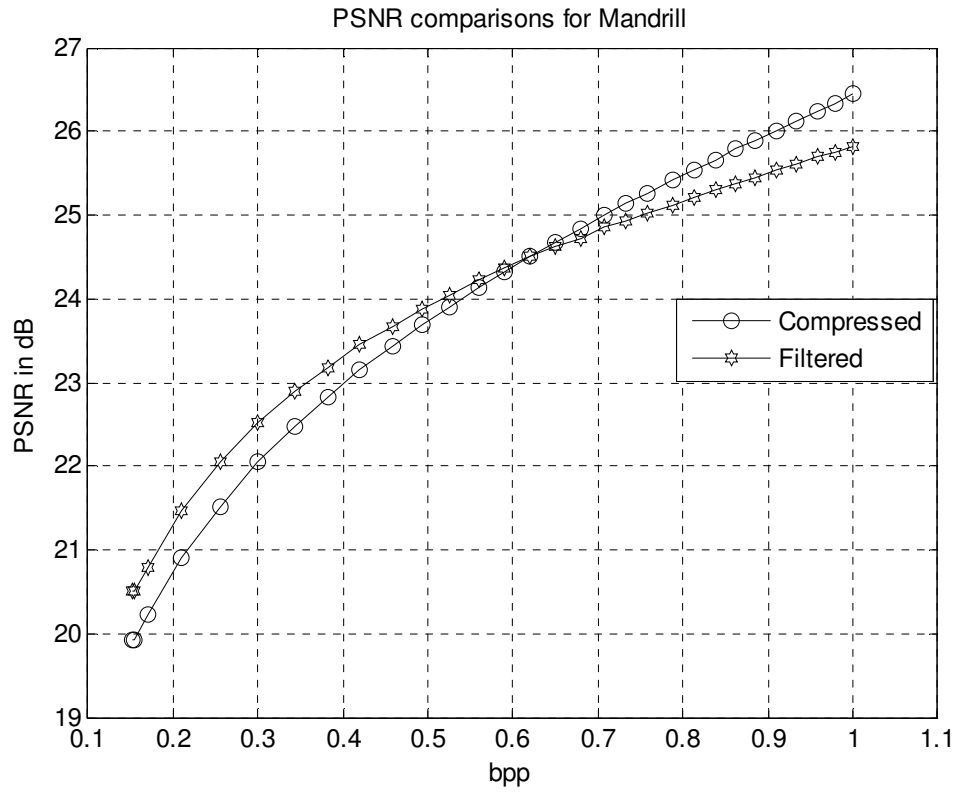


Figure 6. 7. PSNR and SSIM performance of the proposed algorithms for the Mandrill.

As it is seen from Tables 6.1 – 6.5 and Figures 6.3 – 6.7, the success of the proposed algorithms is clear. They can effectively be used for many applications especially at low bit rates.

Since PSNR and SSIM are not directly correlated with the quality of human visual system perceives, visual appearances are also taken into account. So, some visual results are presented in Figures 6.8 – 6.17 for compressed test images having different compression ratios.



(a) Compressed, PSNR: 24.25



(b) Filtered, PSNR: 25.70

Figure 6. 8. Subjective quality comparison of the compressed image the Lena at 0.1335 bpp.



(a) Compressed, PSNR: 28.89



(b) Filtered, PSNR: 30.19

Figure 6. 9. Subjective quality comparison of the compressed image the Lena at 0.2034 bpp.



(a) Compressed, PSNR: 23.74



(b) Filtered, PSNR: 24.92

Figure 6. 10. Subjective quality comparison of the compressed image the Goldhill at 0.1255 bpp.



(a) Compressed, PSNR: 27.43



(b) Filtered, PSNR: 28.31

Figure 6. 11. Subjective quality comparison of the compressed image the Goldhill at 0.2090 bpp.



(a) Compressed, PSNR: 23.39



(b) Filtered, PSNR: 24.48

Figure 6. 12. Subjective quality comparison of the compressed image the Airplane at 0.1411 bpp.



(a) Compressed, PSNR: 27.45



(b) Filtered, PSNR: 28.63

Figure 6. 13. Subjective quality comparison of the compressed image the Airplane at 0.2051 bpp.



(a) Compressed, PSNR: 24.29



(b) Filtered, PSNR: 25.88

Figure 6. 14. Subjective quality comparison of the compressed image the Peppers at 0.1366 bpp.



(a) Compressed, PSNR: 28.68



(b) Filtered, PSNR: 30.09

Figure 6. 15. Subjective quality comparison of the compressed image the Peppers at 0.2050 bpp.

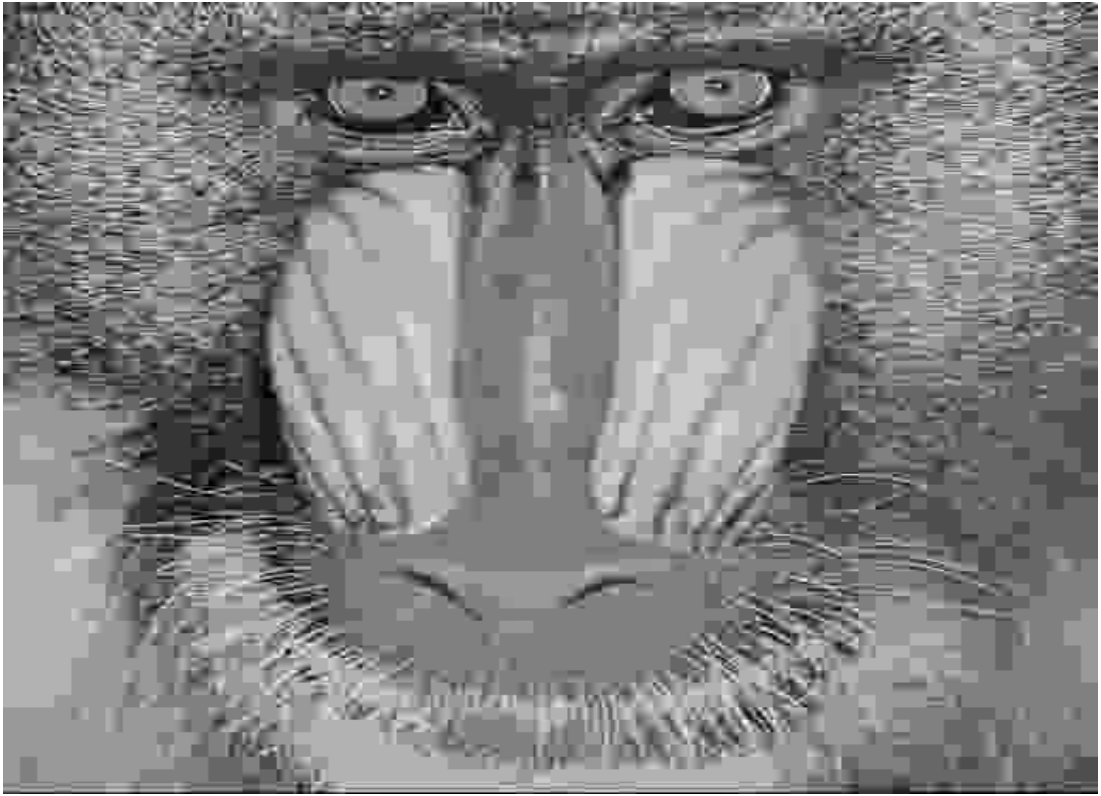


(a) Compressed, PSNR: 19.93



(b) Filtered, PSNR: 20.20

Figure 6. 16. Subjective quality comparison of the compressed image the Mandrill at 0.1530 bpp.



(a) Compressed, PSNR: 20.91



(b) Filtered, PSNR: 21.46

Figure 6. 17. Subjective quality comparison of the compressed image the Mandrill at 0.2105 bpp.

Table 6.6 are obtained for different compression ratios, it shows consuming times when the proposed method runs. The speeds of the algorithm decreases as the compression rate increases since more artifacts occur. Also, different pictures affect the run time, because they have a specific characteristic separately. Finally, processors in hardware change the operating time of the proposed method. Time measurements are taken by computer having Dual CPU at 2.2 GHz at normal load conditions.

Table 6. 6. Bit rate versus Run time.

	bbp	Run Time (in Seconds)
Lena		
	0.1335	5.84
	0.1571	5.70
	0.2034	5.47
	0.2584	5.36
	0.3080	5.24
Goldhill		
	0.1255	6.34
	0.1522	5.68
	0.2090	5.32
	0.2655	5.27
	0.3027	5.17
Airplane		
	0.1411	5.97
	0.1689	5.53
	0.2051	5.43
	0.2535	5.27
	0.3127	5.20
Peppers		
	0.1366	5.69
	0.1608	5.46
	0.2050	5.30
	0.2597	5.23
	0.3066	5.05
Mandrill		
	0.1530	5.59
	0.1704	5.27
	0.2105	4.92
	0.2556	4.77
	0.3001	4.67

In order to evaluate the performance of the proposed deblocking and deringing techniques and to compare them with other techniques; the H263 standard [45], the MPEG4 standard [46], the POCS technique proposed in [47], two post-processing techniques proposed by Zeng [48] and Chen [49] are used. The results over different images are tabulated in Table 6.7. Figure 6.18 shows PSNR versus bit rate for the compressed image Lena.

Table 6. 7. Comparison of PSNR for different post-processing techniques.

Test Images	Bits/pixel (bpp)	PSNR						
		Decoded	H.263	MPEG4	POCS	Ref. [48]	Ref. [49]	Proposed
Lena	0.217	29.47	30.20	30.02	30.23	29.56	30.39	30.69
Goldhill	0.227	27.90	28.50	28.31	28.46	27.85	28.54	28.72
Airplane	0.240	28.72	29.34	29.32	29.34	28.41	29.39	29.89
Peppers	0.221	29.21	30.02	30.04	29.85	29.39	29.95	30.55
Mandrill	0.300	22.05	22.35	22.15	22.44	21.92	22.49	22.53

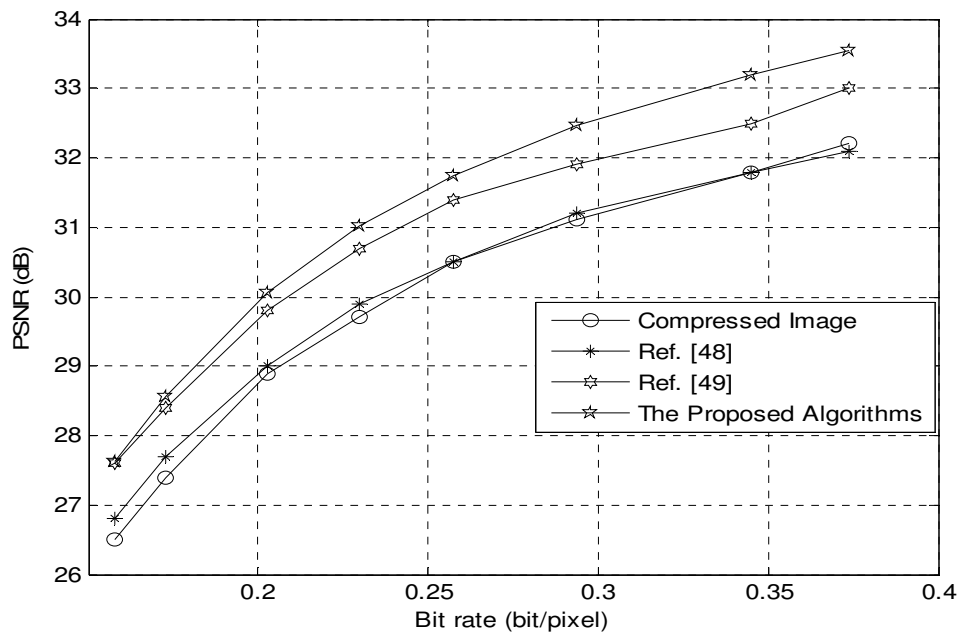


Figure 6. 18. PSNR versus bit rate for the compressed image Lena.

And, for the subjective evaluation of the proposed techniques, examples are shown in Figure 6.19 and 6.20. In Figure 6.19, an enlarged part of the compressed Lena image at 0.217 bpp and the corresponding results post-processed by different techniques are presented. As it is seen, the resultant image of the proposed algorithms is the best of them.



a) Decoded, PSNR=29.47

b) Zeng [48], PSNR=29.56

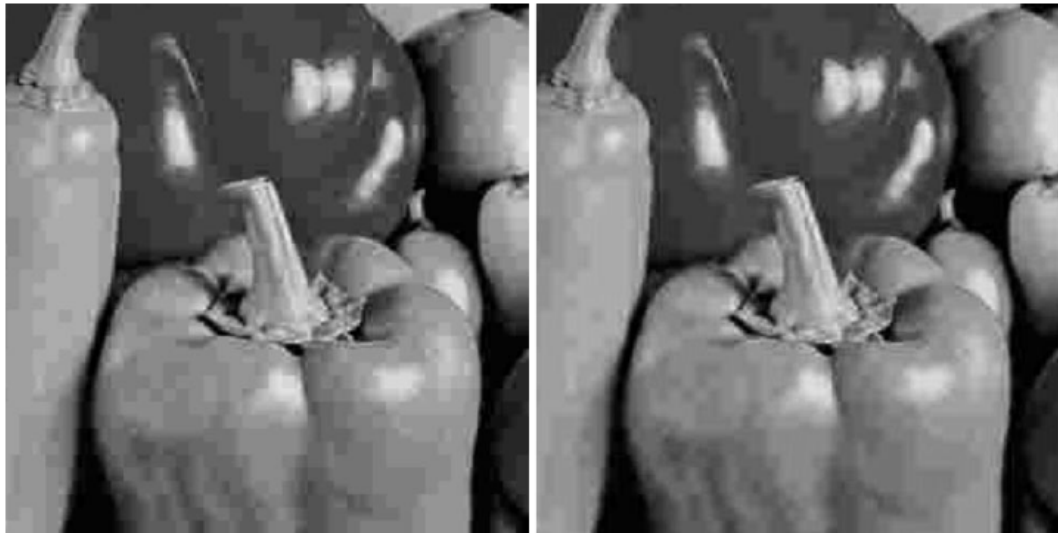


c) Chen [49], PSNR=30.39

d) The proposed algorithms, PSNR=30.69

Figure 6. 19. Subjective quality comparison of the compressed Lena image at 0.217 bpp post-processed by different methods.

In Figure 6.20, an enlarged part of the compressed the Peppers image at 0.221 bpp and the corresponding results post-processed by different techniques are presented. Again it is seen, the resultant image of the proposed algorithms is the best of them.



a) Decoded, PSNR=29.21

b) Zeng [48], PSNR=29.39



c) Chen [49], PSNR=29.95

d) The proposed algorithms, PSNR=30.55

Figure 6. 20. Subjective quality comparison of the compressed the Peppers image at 0.221 bpp post-processed by different methods.

Also, some other methods are tested to show advantages of the proposed techniques. They are an adaptive deblocking filter for DCT coded video [50] and the deblocking filter according to the book named still image and video compression with matlab [51]. The results over different images are tabulated in Table 6.8. Figure 6.21 shows PSNR versus bit rate for the compressed image Lena.

Table 6. 8. Comparison of PSNR for different post-processing techniques.

Test Images	Bits/pixel (bpp)	PSNR			
		Decoded	Ref. [50]	Ref. [51]	Proposed
Lena	0.217	29.47	28.51	29.75	30.69
Goldhill	0.227	27.90	27.75	28.10	28.72
Airplane	0.240	28.72	27.24	28.93	29.89
Peppers	0.221	29.21	28.46	29.53	30.55
Mandrill	0.300	22.05	21.60	22.10	22.53

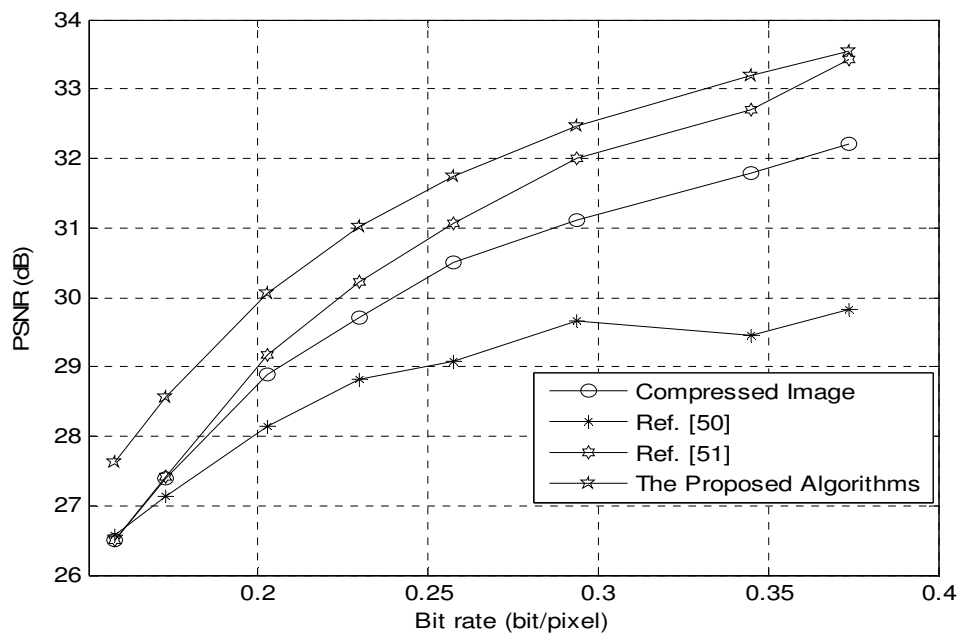


Figure 6. 21. PSNR versus bit rate for the compressed image Lena.

And, also for the subjective evaluation of the proposed techniques, examples are shown in Figure 6.22 and 6.23. In Figure 6.22, an enlarged part of the compressed Goldhill image at 0.227 bpp and the corresponding results post-processed by different techniques are presented. As it is seen, the resultant image of the proposed algorithms is the best of them.



a) Decoded, PSNR=27.90

b) Ref [50], PSNR=27.75



c) Ref [51], PSNR=28.10

d) The proposed algorithms, PSNR=28.72

Figure 6. 22. Subjective quality comparison of the compressed Lena image at 0.227 bpp post-processed by different methods.

In Figure 6.23, an enlarged part of the compressed Airplane image at 0.240 bpp and the corresponding results post-processed by different techniques are presented. Again it is seen, the resultant image of the proposed algorithms is the best of them.



a) Decoded, PSNR=28.72

b) Ref [50], PSNR=27.24



c) Ref [51], PSNR=28.93

d) The proposed algorithms, PSNR=29.89

Figure 6. 23. Subjective quality comparison of the compressed the Peppers image at 0.240 bpp post-processed by different methods.

CHAPTER VII

CONCLUSION

This thesis addresses artifacts that arise in block based DCT image compression. The objective of the thesis is to classify, analyze and find solutions to reduce the visual appearance of coding artifacts. Two new independent adaptive algorithms using adaptive fuzzy filter are developed to reduce coding artifacts. First, blocking artifacts are removed and then ringing artifacts are eliminated. They detect the possible locations of artifacts and adapt the filtering strength to the detected artifact level. Then, a fuzzy filter based on detection results is used. The algorithms can highly preserve the high frequency components while smoothing out artifacts.

Simulation results show that the proposed algorithms significantly reduce the blocking artifacts in both objective and subjective measures. It is able to be adaptive to different image content and qualities. It can effectively remove coding artifacts even at very low bit rate.

The amount of computation it requires is also acceptable compared to other methods in literature. It can also be used for color images by applying all color space like monotonic image. Similarly, frames from video sequences can be considered as color image. So, the algorithms can be applied on them. But in videos, there is an average of 25 frames. Consequently, the algorithms may not reach real time. However, if you have sufficient time, the algorithms offer better visual quality.

In future research, although the algorithm of blocking artifacts removal presents good working, the algorithm of ringing artifact removal needs to be developed. Because, it slows down the overall algorithm since it requires a large number of computations. Maybe, the detections of both algorithms can be improved. Despite many membership functions tried, other ones can affect the performance and speed of the algorithms.

REFERENCES

- [1] S.A. Martucci. (1994). Symmetric convolution and the discrete sine and cosine transform, *IEEE Transaction on Signal Processing*, **42**, 1038-1051.
- [2] K. R. Rao and P. Yip. 1990. Discrete Cosine Transform: Algorithms, Advantages, Applications: Academic.
- [3] R. C. Gonzalez and R. E. Woods. 2004. Digital Image Processing: Addison Wesley.
- [4] N. Ahmed, T. Natrajan, and K. R. Rao. (1989). Discrete Cosine Transform, *IEEE Transactions on Computers*, **23**, 988-991.
- [5] G. K. Wallace. (1992). The JPEG Still Picture Compression Standard, *IEEE Trans. On Consumer Electronics*, **38**, 18-34.
- [6] Shen-Chuan Tai, Yen-Yu Chen, and Shin-Feng Sheu. (2005). Deblocking Filter for Low Bit Rate MPEG-4 Video, *IEEE Transactions on Circuits and Systems for Video Technology*, **15**, 733-741.
- [7] Stevenson, R.L. (1993). Reduction of coding artifacts in transform image coding, *IEEE International Conference on Acoustics, Speech, and Signal Processing*, **5**, 401-404
- [8] T. Ozcelik, J. C. Brailean, and A. K. Katsaggelos. (1995). Image and video compression algorithms based on recovery techniques using mean field annealing, *The Proceedings of the IEEE*, **83**, 304-315.
- [9] T. Meier, K. N. Ngan, and G. Crebbin. (1999). Reduction of Blocking Artifacts in Image and Video Coding, *IEEE Trans. Circuits Syst. Video Technol.*, **9**, 490-500.
- [10] J. Li and C.-C. J. Kuo. (1997). Coding artifact removal with multiscale post-processing, *IEEE International Conference on Image Processing, Santa Barbara, CA*, **1**, 45-48.
- [11] Stevenson, R.L. (1993). Reduction of coding artifacts in transform image coding, *IEEE International Conference on Acoustics, Speech, and Signal Processing*, **5**, 401-404
- [12] D. C. Youla. (1978). Generalized image restoration by the method of alternating orthogonal projections, *IEEE Trans. Circuits Systems*, **25**, 694-702.

- [13] Y. Yang, N. P. Galatsanos, and A. K. Katsaggelos. (1993). Regularized reconstruction to reduce blocking artifacts of block discrete cosine transform compressed image, *IEEE Trans. Circuits Syst. Video Technol.*, **3**, 421-432.
- [14] Y. Yang, N. P. Galatsanos, and A. K. Katsaggelos. (1995). Projection-based spatially adaptive reconstruction of block-transform compressed images, *IEEE Trans. Image Processing*, **4**, 896-908.
- [15] H. Paek, R.-C. Kim, and S.-U. Lee. (1996). A projection-based post-processing technique to reduce blocking artifacts using a priori information on DCT coefficients of adjacent blocks, *Proc. IEEE Int. Conf. Image Processing*, **2**, 53-56.
- [16] H. Paek, R.-C. Kim, and S.-U. Lee. (2000). A DCT-Based Spatially Adaptive Post-Processing Technique to Reduce the Blocking Artifacts in Transform Coded Images, *IEEE Trans. Circuits Syst. Video Technol.*, **10**, 36-41.
- [17] R. Rosenholtz and A. Zakhor. (1992). Iterative procedures for reduction of blocking effects in transform image coding, *IEEE Trans. Circuits Syst. Video Technol.*, **2**, 91-95.
- [18] Michael T. Orchard, Zixiang Xiong and Ya-Qin Zhang. (1997). A deblocking algorithm for jpeg compressed images using overcomplete wavelet representations, *IEEE Trans. Circuits Syst. Video Technol.*, **7**, 433-437.
- [19] Howard C. Reeve III and J. S. Lim. (1984). Reduction of Blocking Effect in Image Coding, *Optical Eng.*, **23**, 34-37.
- [20] B. Ramamurthi and A. Gersho. (1986). Nonlinear space-variant postprocessing of block coded images, *IEEE Trans. Acoust., Speech, Signal Processing*, **34**, 1258-1268.
- [21] C. J. Kuo and R. J. Hsieh. (1995). Adaptive postprocessor for block encoded images, *IEEE Trans. Circuits Syst. Video Technol.*, **5**, 298-304.
- [22] S. D. Kim, J. Yi, H. M. Kim, and J. B. Ra. (1999). A deblocking filter with two separate modes in block-based video coding, *IEEE Trans. Circuits Syst. Video Technol.*, **9**, 156-160.
- [23] P. List, A. Joch, J. Lainema, G. Bjntegaard, M. Karczewicz, (2003). Adaptive deblocking filter, *IEEE Transactions on Circuits and Systems for Video Technology*, **13**, 614-619.
- [24] Shen-Chuan Tai, Yen-Yu Chen, and Shin-Feng Sheu. (2005). Deblocking Filter for Low Bit Rate MPEG-4 Video, *IEEE Transactions on Circuits and Systems for Video Technology*, **15**, 733-741.

- [25] C. J. Kuo and R. J. Hsieh. (1995). Adaptive postprocessor for block encoded images, *IEEE Trans. Circuits Syst. Video Technol.*, **5**, 298-304.
- [26] S. D. Kim, J. Yi, H. M. Kim, and J. B. Ra. (1999). A deblocking filter with two separate modes in block-based video coding, *IEEE Trans. Circuits Syst. Video Technol.*, **9**, 156-160.
- [27] Y. Nie, K.E. Barner. (2006). Fuzzy transformation and its applications in Image Processing, Proceedings. *IEEE Trans. Image Process*, **15**, 910-927.
- [28] S. M. Smith, J. M. Brady. (1997). Susan-a new approach to low level image processing, *Int. J. Comput. Vis.*, **23**, 45-78.
- [29] C. Tomasi, R. Manduchi. (1998). Bilateral filtering for gray and color images, *Proc. Int. Conf. Comput. Vis.* 839-846.
- [30] Y. Nie, K.E. Barner. (2003). Fuzzy transformation and its applications Image Processing, *Proc. of IEEE Int. Conf. ImageProc.*
- [31] B. J. Justusson. (1981). Median Filtering: Statistical properties, *Two Dimensional Digital Signal Processing*, **2**, 161-196.
- [32] K. E. Barner, R. C. Hardie. 1999. Spatial-rank order selection filter, *Nonlinear Signal Processing*. 69-110: Academic.
- [33] R. C. Hardie, C. G. Boncelet. (1993). LUM Filters: A class of rank-order-based filters for smoothing and sharpening, *IEEE Trans. Signal Process.*, **41**, 1061-1076.
- [34] Y. Nie, H.-S. Kong, A. Vetro, H. Sun, K. E. Barner. (2005). Fast adaptive fuzzy post-filtering for coding artifacts removal in interlaced video, *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, **2**, 993-996.
- [35] H. R. Tizhoosh. 1997. Fuzzy Image Processing: Springer.
- [36] Alper Pasha. (2006). Morphological image processing with fuzzy logic, *Havacılık ve uzay teknolojileri dergisi*, **2**, 27-34.
- [37] S. N. Sivanandam, S. Sumathi, S. N. Deepa. 2006. Introduction to Fuzzy Logic using MATLAB: Springer.
- [38] Y.H. Kuo, C.S. Lee, C.C. Liu. (1997). A New Fuzzy Edge Detection Method for Image Enhancement. *Proc. 6th IEEE International Conference on Fuzzy Systems*, **2**, 1069-1074.
- [39] A. K. Jain. 1989. Fundamentals of Digital Image Processing: Prentice Hall.

- [40] H. R. Wu and K. R. Rao. 2006. Digital Video Image Quality and Perceptual Coding: CRC Press, *Video quality testing*, 125-153.
- [41] H. R. Wu and K. R. Rao. 2006. Digital Video Image Quality and Perceptual Coding: CRC Press, *Perceptual video quality metrics - a review*, 155-179.
- [42] B. Girod. (1993). What's wrong with mean squared error?, A. B. Watson (ed.), *Visual Factors of Electronic Image Communications*, 207-220.
- [43] Z. Wang, A. C. Bovik, and Ligang Lu. (2002). Why is image quality assessment so difficult?, *Proc. IEEE International Conference on Acoustics, Speech, & Signal Processing*. 3313-3316.
- [44] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli. (2004). Image quality assessment: From error visibility to structural similarity. *IEEE Trans. Image Process.*, **13**, 600-612.
- [45] T. Chen, H. R. Wu, and B. Qiu. (2001) Adaptive postfiltering of transform coefficients for the reduction of blocking artifacts, *IEEE Trans. Circuits Syst. Video Technol.*, **11**, 594-602.
- [46] MPEG4 Video Verification Model Version 15.0, ISO/IEC/JTC1/SC29/WG11, 1999.
- [47] Y. Yang, N. P. Galatsanos, and A. K. Katsaggelos. (1993) Regularized reconstruction to reduce blocking artifacts of block discrete cosine transform compressed images, *IEEE Trans. Circuits Syst. Video Technol.*, **3**, 421-432.
- [48] B. Zeng. (1999) Reduction of blocking effect in DCT-coded images using zero-masking techniques, *Signal Processing*, **79**, 205-211.
- [49] T. Chen, H. R. Wu, and B. Qiu. (2001) Adaptive postfiltering of transform coefficients for the reduction of blocking artifacts, *IEEE Trans. Circuits Syst. Video Technol.*, **11**, 594-602.
- [50] K. S. Thyagarajan. 2010. Still Image and Video Compression with MATLAB: Wiley.
- [51] Salim Chebbo, Philippe Durieux, Béatrice Pesquet-Popescu. (2009). Adaptive deblocking filter for DCT coded video, *Fourth International Workshop on Video Processing and Quality Metrics for Consumer Electronic*.