THE HONG KONG POLYTECHNIC UNVERSITY

Department of Electronic and Information Engineering

Super-Resolution Videos and Their Application to High Definition TV

Wong Chi Shing

A thesis

submitted in partial fulfilment of the requirements

for the Degree of Master of Philosophy

April 2010

Certificate of Originality

I hereby declare that this thesis is my own work and that, to the best of my knowledge and belief, it reproduces no material previously published or written, nor material that has been accepted for the award of any other degree or diploma, except where due acknowledgement has been made in the text.

______________________ (Signed)

Wong Chi Shing (Name of student)

Abstract

High quality video interpolation is always desirable, since the definition of video display devices is improving. It is always necessary to port a video of lower quality for display in higher quality displays, such as the conversion of SDTV videos to HDTV videos.

The image/video interpolation involves the prediction of the unknown pixels from the neighboring known pixels. It is well known that using a classical linear interpolation algorithm, such as bilinear and bicubic cannot produce a visual quality that is accepted by today customer, but it has advantage on the low computational complexity.

In several years before, interpolation algorithms that have low computational complexity are very important, as the processing power of the digital devices at this time is very low. However, today digital devices have a very powerful processing power. Therefore, the computational complexity is not a main concern. Nowadays, people concern more about the visual quality of the interpolated image. In order to have a high visual quality of the interpolated image, a well-known algorithm called Edge-Directed Interpolation (EDI) is proposed, which interpolate the image according to the edge direction. One of the most outstanding methods in the EDI types methods is the New Edge-Directed Interpolation (NEDI). It generates a high quality interpolated image with continuous and sharper edge. This method stimulates us to further improve it to generate the highest quality interpolated image.

This thesis consists of three parts. In the first part of the thesis, we address the problems of the traditional NEDI method: prediction error and the fixed interpolation factor. These problems cause the NEDI method not suitable for the SDTV to HDTV conversion. Therefore, we develop a new eight-order sample structure for the NEDI interpolation to reduce the prediction error in the Wiener filter estimation. Moreover, we present a new fast approach for the enlargement of a SDTV video to a HDTV video with an interpolation factor of 1.5 which

cannot be done by the NEDI method before. In addition, we also make an analysis on the number of the sample points used in the proposed method and its effect on regions with high frequency.

In the second part of this thesis, we develop a new Adaptive Directional Window Selection method for the EDI interpolation. It can solve one of the major problems in the NEDI method, which is the covariance mismatch problem. This mismatch problem gives rise to the interpolation artifacts (prediction error) and ringing effects. As a result, the visual quality of the interpolated image is reducing. However, using our proposed method it can reduce most of the artifacts and ringing effects in the NEDI method and the performance on the high frequency and texture regions has also been improved.

In the last part of this thesis, we present a pre-processing method for the Wiener filter estimation and a fast method for large up-scaling interpolation. We find that one of the reasons of the artifacts that appear in the interpolated image in NEDI method is because of the abrupt change of the pixels intensities in the sample data. Therefore, we solve this problem by using our proposed pre-processing method in the sample data before the Wiener filter estimation. This method can make the interpolation more robust to the high frequency, texture and noise region and have interpolated images with a high visual quality as compared to that of the NEDI method. Moreover, the proposed fast method can speed up the EDI interpolation when the interpolation factor is large (i.e. 4, 8, 16 times…) and keep the highest visual quality of the interpolated image.

List of Publication

1. Wong Chi Shing and Wan-Chi Siu, "Further Improved Edge-Directed interpolation and Fast EDI for SDTV to HDTV conversion", The *18th European Signal Processing Conference* (*EUSIPCO'10*), August 2010, Aalborg Denmark.
2. Wong Chi Shing and Wan-Chi Siu, "Adaptive Directional Window Selection for Edge-Directed Interpolation", *The 2nd IEEE International Workshop on Multimedia Computing and Communications* (*MCC'10*), August 2010, Zurich Switzerland.
3. Wong Chi Shing and Wan-Chi Siu, "Adaptive Directional Window Selection and Pre-Processing step of the sample data for Edge-Directed Interpolation", to be submitted to an *international Journal*.

Acknowledgments

First of all, I would like to express my sincere gratitude to my supervisor, Professor Wan-Chi Siu, Chair Professor for his patient, helpful instructions on this research and continuous encouragement through the time of my study towards the Master of Philosophy degree. His knowledge, wisdom, kindness and spirit have benefited me greatly.

Further, I world like to thank to my colleagues for their help in my research work. They are very helpful and nice to me, so that I can concentrate on my research work.

Finally, I am very grateful to my parents for teaching me the importance of hard work and perseverance. I would like to thank all my friend, whose love and support are always the greatest inspiration for me.

Thanks again to all the people who have helped me in one-way or another.

Contents

List of Figures

List of Tables

Chapter 1

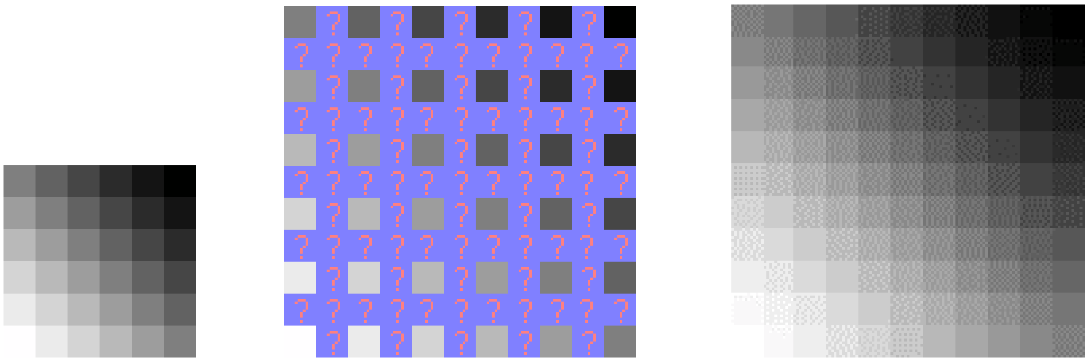
Introduction

1.1 Motivation

Digital imaging devices are growing in popularity and commonly used in various imaging applications in recent years. In many applications, the use of the typical digital imaging device can only capture the scene at a limited range of spatial resolution. However, the development of the High Definition television brings out the demand for obtaining high spatial resolution images/video frames from low spatial resolution images/video frames.

High Definition television (or HDTV) can produce a butter visual quality than Standard Definition television (or SDTV). It is because HDTV has a new digital television broadcasting system with a high-resolution display (HD: 1080×720 / Full HD: 1920×1080). However, not all the digital videos or movies support such high-resolution display. All the digital videos or movies in the SDTV Age only stored in a SD resolution video sequence format, and the resolution of the SD format can only up to a size of 720×576. Therefore, it needs some interpolation algorithm, so that the SD resolution the video sequence can be displayed on a high-resolution HD display device correctly. Naturally, there are large demands on the better interpolation algorithms.

Let us recall image interpolation (upsampling) process as illustrated in Figure 1.1. The original image is shown in Figure 1.1 (a). During the interpolation process, the resolution of the original image will be increasing firstly by adding some new (unknown) pixels between the original pixels as shown in Figure 1.1 (b). After that all unknown pixels will be interpolated from the neighboring original pixels to form a high-resolution image as shown in Figure 1.1 (c). However, during the interpolation process, no additional information is added into this interpolated image. We can only make use of these original pixels to estimate the unknown pixels. Therefore, the interpolated image will contain unwanted estimation errors. These estimation errors usually cause distortion on the subjective (i.e. burring, aliasing and edge halo as shown in Fig. 1.2) and objective (i.e. PSNR) qualities of the interpolated image. We usually call these distortions on the interpolated image as Image Interpolation Artifacts which will be discussed in this chapter at a later section.



(a) Original (b) Before interpolation (c) After interpolation

Figure 1.1: Illustrative of the Interpolation process

(a) Original (b) Blurring (c) Aliasing (d) Edge Halo

Figure 1.2: The common types of the distortion cause by interpolation

Recently, many research studies tried to improve the subjective and objective qualities of the interpolated image by sharpening the edges and lowering the PSNR and to make comparison with linear interpolation algorithms. They enhance the image quality by developing some new interpolation algorithms or making improvement on exiting algorithms in both spatial and frequency domains. In this thesis, we will mainly force on the spatial domain interpolation algorithms.

Edge Directed Interpolation (EDI) [1] is a new direction on improving the visual quality of the interpolated image. The major thing that affects the visual quality of the interpolated image is the edge performance. Discontinuous edge, aliasing or burring effects will reduce greatly the visual quality of an image or video sequence, especially the artifacts that appear near the major edge. Therefore, many researchers develop new interpolation methods that interpolate the image follow the edge direction. One of the most outstanding methods is the New Edge-Directed Interpolation (NEDI) proposed by Li and Orchard in [2]. It can preserve the sharpness and continuity of the image edge in the interpolated image. However, the NEDI suffers from a number of disadvantages, such as high computational complexity, watercolour artifacts and ringing effect in high frequency regions and instability of the algorithm in smooth regions of the image.

1.2 Objective

Image Interpolation aims at reconstructing an image from a low-resolution to high-resolution. The objective of this thesis is to develop efficient image interpolation algorithms based on the New Edge Directed Interpolation (NEDI) [2] and apply it to the HDTV.

It is well know that using classical linear interpolation algorithms, such as bilinear and bicubic interpolation cannot produce a visual quality that is accepted by today customer, as it often suffers from the edge blurring effect or produces some artifacts around the edge area [3]-

[5]. One of the solutions of it is to use some of the adaptive interpolation methods such as NEDI. Although the NEDI method has its disadvantage, its outstanding edge performance in the interpolated images is also attractive to us. Therefore, we have tried to improve the NEDI method by using several techniques which will be discussed in the later chapters.

1.3 Types of Interpolation Algorithms

There are a variety of spatial domain image interpolation algorithms in the literature. These algorithms can be grouped into two categories: non-adaptive and adaptive. The non-adaptive or adaptive means that the degree of freedom of their ability to change their behavioural rules. For non-adaptive, it means that it have determinated a set of rules and cannot be changed and for adaptive, it means that it have rules that can be changed dynamically with changes in state. In the interpolation point of view, the non-adaptive interpolation does not require any a priori information of the given image data, and it treats the data in the set in the same manner. On the other hand, the adaptive interpolation estimates the unknown pixel values according to the priori information of the given image data set, and also the position of the unknown pixel in the image.

1.3.1 Non-adaptive interpolation Algorithms

Some of the most popular non-adaptive interpolation algorithms are based on the polynomial interpolation including nearest neighbor, bilinear, bicubic, spline, sinc and lanczos etc. Depending on their complexity, these use anywhere from 0 to 256 (or more) adjacent pixels for interpolation. The more adjacent pixels they include, the more accurate they can become, but this comes at the expense of much longer computational time. Moreover, these algorithms assume that all data points are generated through a polynomial function with a particular order. The higher the order of the polynomial function, the more accurate they can become.

For these non-adaptive interpolations using a fixed polynomial function, the computational complexity is usually small, which means that they are suitable for real time application. However, the drawbacks of the interpolations in this category are that the interpolated image usually suffers from the interpolation artifices such as burring, aliasing and edge halo effects as shown in Fig. 1.2.

1.3.2 Adaptive Interpolation Algorithms

Adaptive interpolation algorithms do not treat all the pixels equally. It considers the nearby image context first, and then makes decision on how to interpolate the unknown pixels. Many of these algorithms make use of the detection of the presence of an edge, so that it can minimize the interpolation artifacts. Such detection or considerations are important, because the human visual system perceives image edges, and the smooth area of the image in a very different manner. When the interpolated image has sharp and continuous image edges, it will be pleased to human visual system and be considered to have better visual quality. Various algorithms are proposed in the literature to produce human visual system pleasing interpolated images by maintaining sharp and continuous edges.

1.4 Image Interpolation Artifacts

After the image interpolation process, the interpolated image many be suffered from three undesirable artifacts. These three undesirable artifacts are blurring, aliasing and edge halos effects which are interconnecting together as shown in Fig.1.3. If one of the artifacts is suppressed, the other two of the artifacts may be enhanced. Therefore, a good interpolation method needs to obtain an optimal balance between these three undesirable artifacts. We are going to discuss these artifacts in the following section.

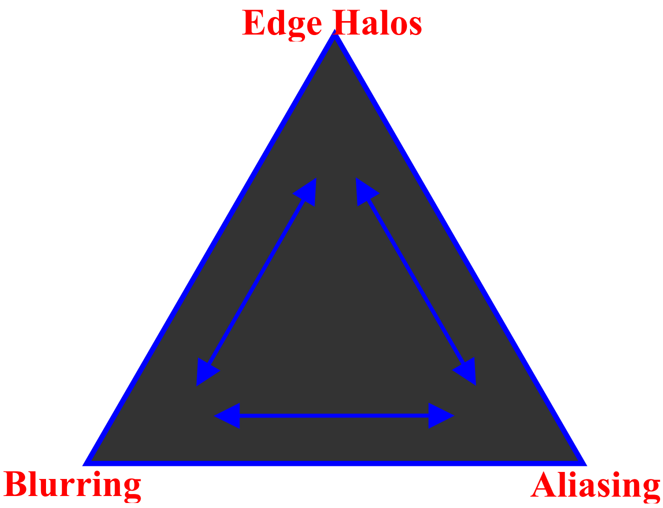


Figure 1.3: The image interpolation artifacts

1.4.1 Blurring

Blurring refers to the image edges which are over smoothed during the interpolation. Image edges always contain the abrupt change of the pixel intensities. The cause of the burring artifacts can be interpreted in both spatial domain and frequency domain as discussed in the following.

In the spatial domain, the blurring effect in the interpolated image usually cause by the use of the polynomial functions in the interpolation algorithms. These polynomial functions usually work as the averaging operation which summing up the weighted neighboring pixels. The averaging operation does not preserve the abrupt changes of the image intensities. It will smoothes out edges of the image and cause blurring in the interpolated image.

In the view of the frequency domain, high frequency refers to the abrupt changes of the image intensities in spatial domain such as image edges and the low frequency represent the small changes of the image intensities in spatial domain such as smooth regions. Therefore, the cause of the burring effect is the loss of the high frequency components. Before we discuss the reason of the loss of the high frequency components in the interpolated image, let us discuss the process of the interpolation in spatial domain first and then relate it to the frequency domain.

In the resampling theory [4], image interpolation can be divided conceptually into two processes: interpolation of the discrete image to a continuous image and then sampling the interpolated image as shown in Fig. 1.4. The degree of the burring effect depends on the interpolation of the discrete image to a continuous image. In the spatial domain, the discrete signals need to be convoluted with the interpolating function to produce the continuous signals. Convolution in the spatial domain equals the multiplication in the frequency domain. Therefore, in the frequency domain, the discrete signals need to be multiplied to the interpolating function. In order to exactly reconstruct a signal from the discrete samples, the interpolating function should be an ideal low-pass filter as shown in Fig. 1.5.

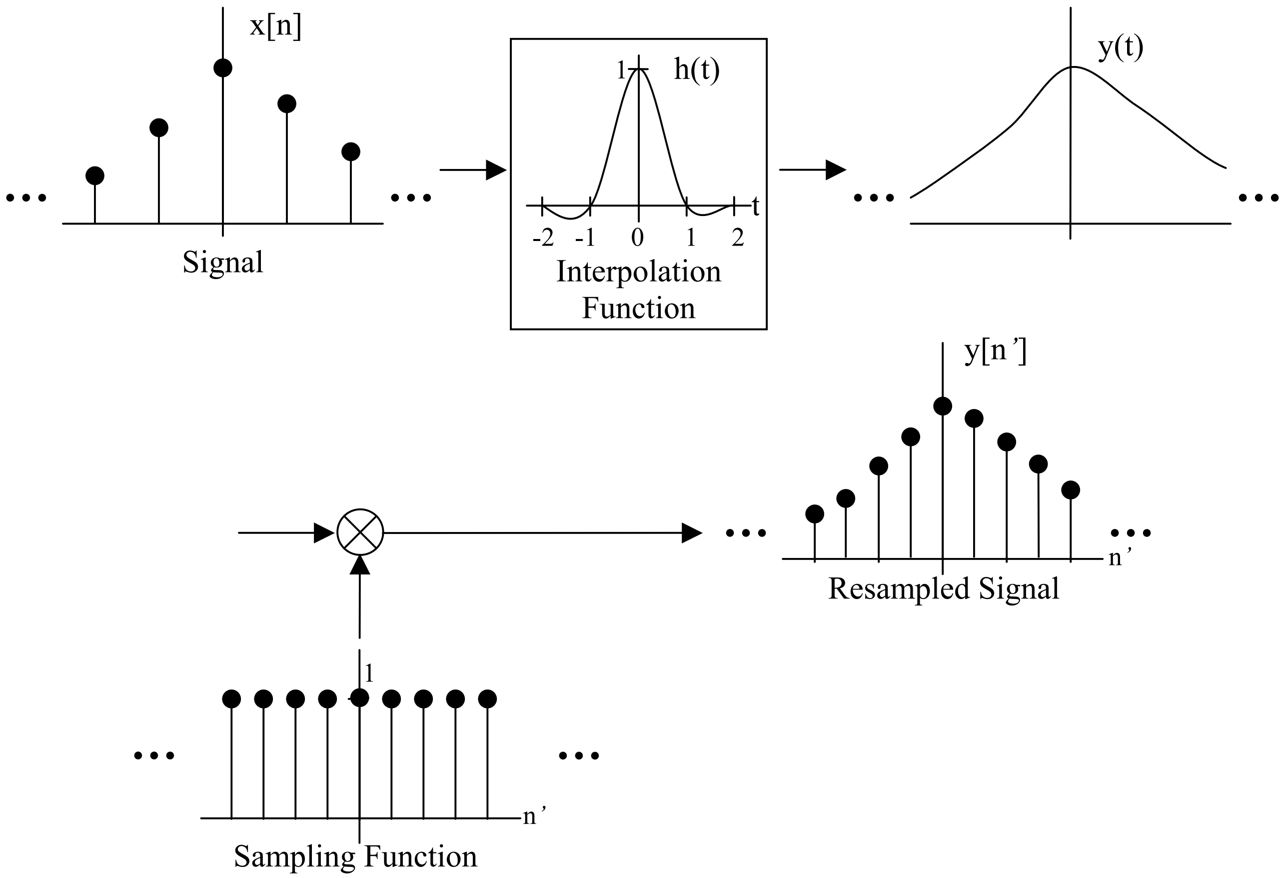


Figure 1.4: The process of the image interpolation in spatial domain

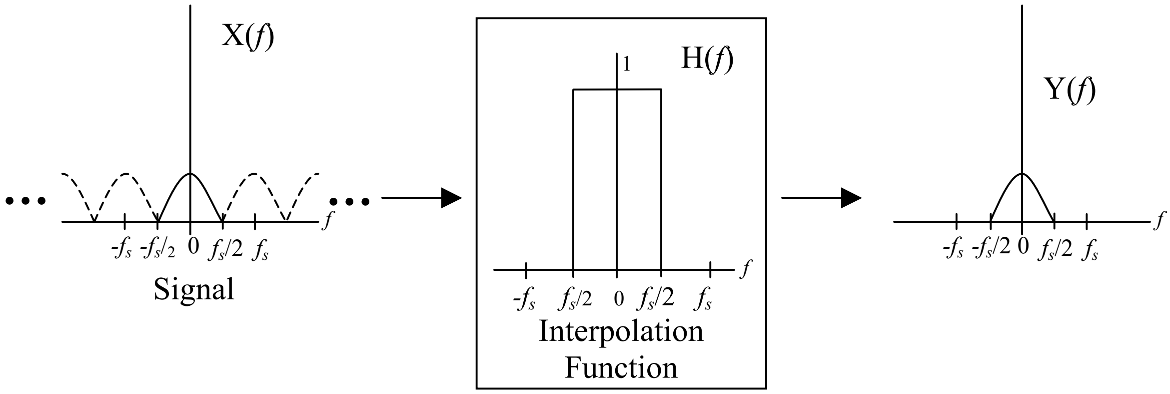


Figure 1.5: Reconstructing a exactly signal from the discrete samples in frequency domain

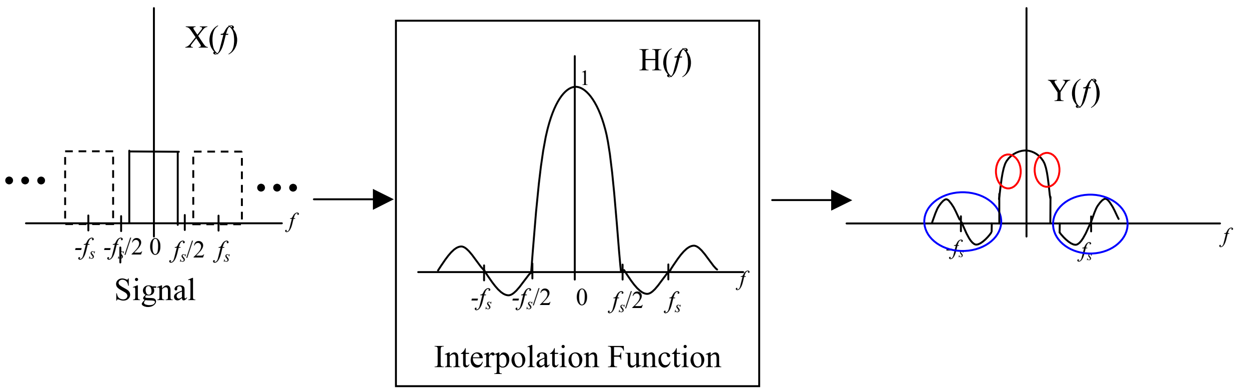
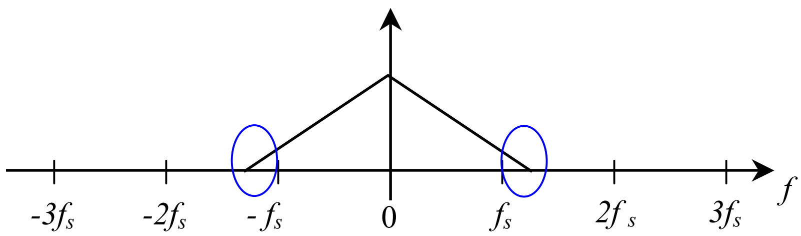


Figure 1.6: The effect of using non-ideal interpolation function in frequency domain

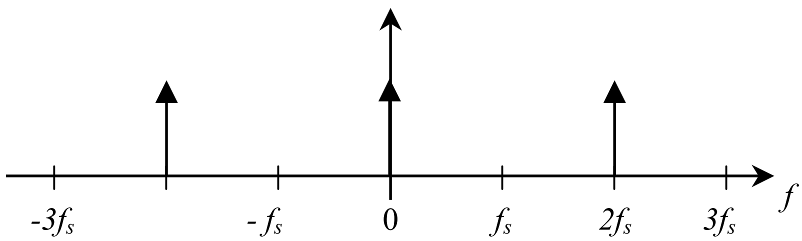
There are, however, in practical situation the ideal low-pass filter cannot be implemented. Therefore, the non-ideas interpolation function will suppress the high frequency component (red ellipses) of the image in the frequency domain, which causes the burring effect as shown in Fig. 1.6.

1.4.2 Aliasing

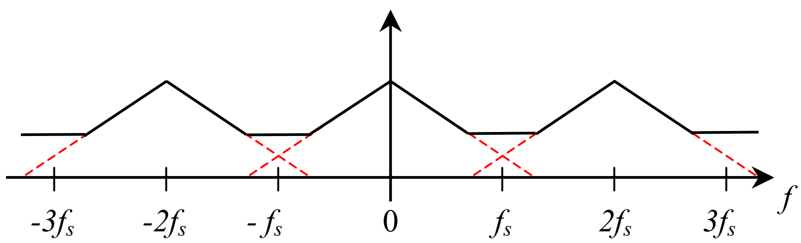
The aliasing happens when the perfect reconstruction cannot be achieved [5]. In Fig. 1.5, it shown an ideal interpolation function applies to the signal X and the output spectrum is bounded within $-f_s$ and f_s where f_s is the original sampling frequency. In Fig. 1.6, it shows how the frequency spectrum of the original image changes after we apply the non-ideal interpolation function. The output of the spectrum as shown in Fig. 1.6 is no longer bounded within $-f_s$ and f_s where f_s is the original sampling frequency. This results in overlapping of image spectrum when increasing the sampling rate due to the existence of unwanted high frequency components (i.e. as shown inside of the blue ellipses in Fig. 1.6). The overlapping of the image spectrum distorts the output image spectrum, which is known as aliasing. The aliasing problem is illustrated in Fig.1.7. Assume that the reconstructed image is reconstructed from a non-ideal interpolation function so that the spectrum is no longer bounded within $-f_s$ and f_s as shown in Fig. 1.7 (a) where the unwanted high frequency components is shown inside of the blue ellipses. When we are up scaling of the original image from the original size ($M \times N$) to the twice of the original size ($2M \times 2N$), it is equivalent to resample the image from f_s to $2fs$. Therefore, we use a new sampling impulse train with sample frequency = $2f_s$ as shown in Fig. 1.7 (b) to sample the reconstructed image. After the resampling the interpolated image spectrum is shown in Fig. 1.7 (c). Clearly, we can see that it is an overlapping of the image spectrum which cause aliasing effect as shown by the red dash lines in Fig. 1.7 (c).



(a) Power spectra of a reconstructed image



(b) Power spectra of a new sampling impulse train with sample frequency = $2f_s$.



(c) Power spectra of the interpolated image output spectrum

Figure 1.7: The Power spectra of the interpolation process.

1.4.3 Edge Halos

The edge halos artifacts are usually caused by over sharpening of the image edge using an unsharp mask. The sharpening process works by utilizing a slightly blurred version of the original image. This is then subtracted away from the original image to detect the presence of edges, in which creating the unsharp mask. Contrast is then selectively increased along these edges using the unsharp mask to produce a sharper interpolated image.

Fig. 1.8 illustrates the visual quality of image edges observed in normal sharpening and over sharpening situation. Before applying the sharpening filter, the edges in the original image are soft and burring as shown in Fig. 1.8 (a). By applying the mild sharpening, the contrast of edge was increasing and the image edges become sharper than the original image as shown in Fig. 1.8 (b). Thus, it can increase the visual quality of the image. However, the over sharpening makes the image contain light/dark outlines or halos near edges as shown in Fig. 1.8 (c). As a result, it reduces the visual quality of the image. Hence only applying the reasonable amount of the sharpening to the image, we can improve the visual quality.

(a) Original (b) Mild sharpening (c) Over sharpening

Figure 1.8: The sharpener image observed from the mild sharpening and over sharpening situation

Chapter 2

Literature Review

2.1 Non-adaptive interpolation Algorithms

Until now, non-adaptive interpolation algorithms are still very popular techniques for using in the image processing. It can be used to enhance and resize an image or video frame. In the follow sections, we will discuss some of the popular non-adaptive interpolation algorithms in the literature.

2.1.1 Nearest Neighbor Interpolation

Nearest Neighbor is the most basic approach which requires the least processing time of all the interpolation algorithms because it only considers one pixel which is the closest one to the interpolated point [3]. This has the effect of simply making each pixel bigger. We also call it as a zero order polynomial interpolation. Fig. 2.1 demonstrates the nearest neighbor Interpolation. Although the nearest neighbor interpolation is the fastest interpolation algorithm, it always suffers from “Jaggies” (aliasing) artifact. The term “Jaggies” artifact refers to the visible steps of the diagonal lines or edges [4]. Fig. 2.2 illustrates the artifact caused by the nearest neighbor interpolation.

Nearest Neighbor Interpolation

1	50
50	100

Original 2×2 Image

1	1	50	50
1	1	50	50
50	50	100	100
50	50	100	100

Interpolated 4×4 Image

Figure 2.1: Nearest Neighbor Interpolation

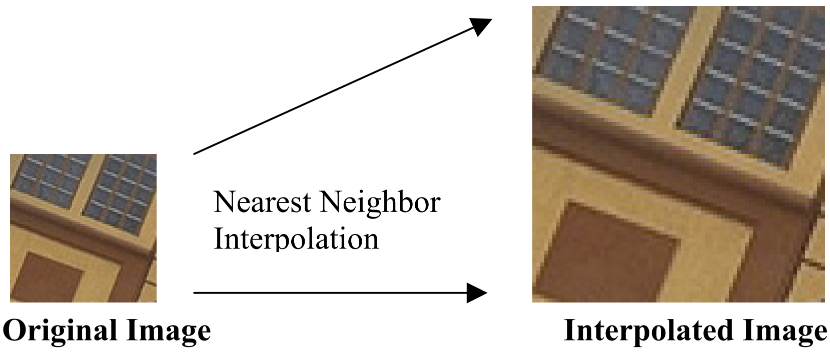


Figure 2.2: "Jaggies" observed from nearest neighbor interpolation

2.1.2 Bilinear Interpolation

Bilinear interpolation is a first order interpolation. It considers the closest 2×2 neighbourhood of known pixel values surrounding the unknown pixel and assumed that all the unknown pixels and the known pixels are aligned on the same plane. Therefore, it takes a weighted average of these four pixels to arrive at its final interpolated value as shown in Fig. 2.3. Pixel Y is the additional unknown pixel. The value of pixel Y is estimated from the four surrounding known pixels (original pixels) A, B, C and D according to the relation Z = (A+B+C+D)/4. This results in much smoother looking images than nearest neighbor interpolation with less "Jaggies", which are removed by the averaging operation [4]. However, bilinear interpolation of an image suffers from blurring artifact when compared to the nearest neighbor interpolated images. Moreover, the computation of the bilinear interpolation is relatively high compared to the nearest neighbor interpolation algorithm.

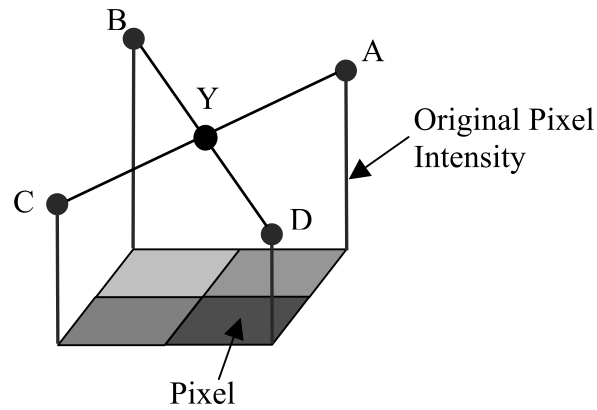


Figure 2.3: Bilinear interpolation

2.1.3 Bicubic Interpolation

Bicubic interpolation goes one step beyond bilinear by considering the closest 4×4 neighbourhood of known pixels. Therefore, a total of 16 pixels will be considered during interpolation. It is a second order interpolation. It is assumed that the unknown pixels and the known pixels are located on a quadratic surface. In the quadratic surface, since the known pixels are at various distances from the unknown pixel, closer pixels are given a higher weighting during the estimation of the unknown pixel [4]. In Fig. 2.4, it shows the formation of the bicubic interpolation. The red arrows represent the nearest known pixels with larger weighting and the green arrows represent the most far away known pixels with the lowest weighting. The blue arrows at are middle between red and green arrows, so they have the median weighting. Bicubic interpolation always produces noticeably sharper images than both of the nearest neighbor and bilinear interpolation with acceptable computational complexity. Because of the good visual quality and acceptable computational complexity of the bicubic interpolation, it is a standard in many image editing programs, printer drivers and in-camera interpolation.

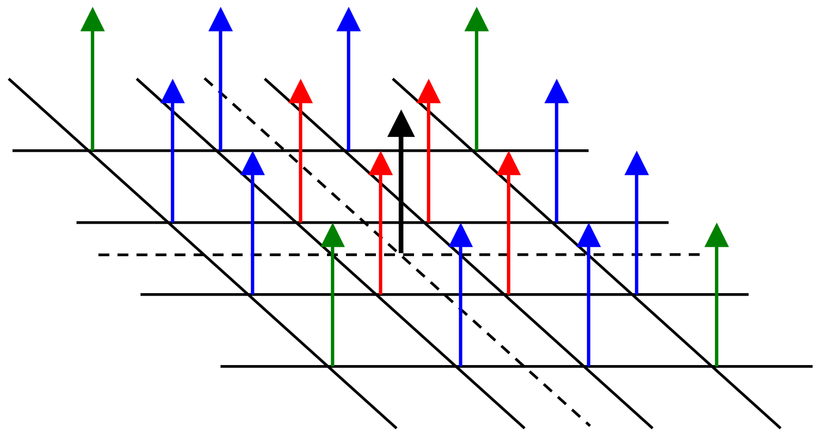


Figure 2.4: Bicubic interpolation

2.1.4 Spline and Sinc Interpolation

Spline and Sinc interpolation is a high order interpolation. They take more surrounding pixels into consideration, so they have higher computational complexity than bilinear and bicubic interpolation. The three methods presented in the previous sections discuss about polynomial interpolation with fixed coefficients. However, the Spline and Sinc interpolation are using the alternative polynomial interpolation methods which adapt the interpolation coefficients to the available neighboring pixels [4]. The adapting of the coefficient sets in polynomial interpolation can enable us to use a large kernel size results in less blurring in the interpolated image [5]. Therefore, the interpolated image would be less distorted. They are extremely useful when the image requires multiple rotations or distortions in separate steps. However, for single-step enlargements or rotations, these higher order interpolation provides diminishing visual improvement as computational time is increased, so it limited the application of this kind of interpolation methods.

2.2 Adaptive interpolation Algorithms

The non-adaptive interpolations treat all pixels equally, so these interpolations have the lower computational complexity. However, theses methods will generate an interpolated image which contains observable artifacts such as burring, aliasing and Edge Halo effects. In order to increase the visual quality of the interpolated image, adaptive interpolations are developed. Adaptive interpolations treat all pixels differently, such that the near by image content are given heavy attention during the interpolation. Various algorithms are proposed in the literature to produce better visual quality of the interpolated image which gives sharper and continuous edge. These include Warped Distance methods [6], [7], Adaptive conventional linear interpolations [8], [9], Projection onto Convex sets (POCS) schemes [10]-[12], Isophote based [14]-[19], Optimal Recovery techniques [20]-[22], Iterative Deconvolution techniques [23], Total Variation Quasi-solution method [24], Polyphase scheme [25] and the Edge-Directed schemes [1], [2], [27]-[46].

For improving the performance of the conventional linear interpolations, the Warped Distance (WaDi) technique was proposed by Ramponi in [6]. The WaDi technique is based on the local gradient features of image edges together with the distances between the unknown pixel and it known neighbour pixels. As we know that the local gradient features are one kind of the important features of image edge. Using the local characteristics of the image in WaDi technique can reduce the interpolation error and has a lower computation complexity. Later on, the improved version of the WaDi was proposed in [7]. It improves the WaDi by adopting both the local asymmetry features and the local gradient features to compute the warped distance. Therefore, more image local information is considered to estimate the interpolated pixels than that in the conventional WaDi interpolation. However, aliasing effect still appears in the interpolated image.

Other types of the improving the performance of the conventional linear interpolations are to use the adaptive interpolation concept. Two adaptive interpolation methods, namely the adaptive bilinear (A-Bilinear) and adaptive bicubic (A-Bicubic) were proposed in [8]. These two methods are based on applying an inverse gradient to perform the conventional bilinear and bicubic interpolations. In the performance of the edge regions, the A-Bilinear and A-Bicubic methods can reduce the aliasing effect and produce sharper edges than the conventional linear interpolations methods. The other improved version of the bilinear interpolation was proposed in [9] which is based on the edge-weighted method. It uses the edge information such as edge direction, rising or falling edge property to calculate the Edge-Weighted ratio. After applying the Edge-Weighted image (EWI) interpolation scheme into the bilinear interpolation, it was found that the subjective and objective qualities are better than conventional bilinear interpolation.

The above-improved visions of the conventional linear interpolations have the advantage of lower computational complexity and better visual quality than the conventional linear interpolation. However, they still suffer from the artifacts that often occurs in the conventional linear interpolation such as blurring, aliasing and edge halos effects. Therefore, it limits the development of the conventional linear interpolations.

Recently, many interpolation methods make further use of the edge information to enhance the visual quality of the interpolated image. The Projection Onto Convex sets (POCS) schemes [10]-[12] can constrain the edge continuity and find the appropriate solution through iterations. The iterative algorithm [10] aims to find a magnified image satisfying two constraints: one of the constraints is derived from sampling theory while the other constraint reflects the confidence that we placed on the initial iterate. Both the constraints are convex sets, therefore, the solution can be found in the intersection of these two convex sets and can be obtained using the POCS method. The induction interpolation method was proposed in

[11], whose specificity is to state the problem of image magnification as an inverse problem of image reduction [13]. A joint Constrained Edge Pattern Recognition and POCS method was proposed in [12]. It makes use of the technique of the Pattern Recognition to recognise the constrained edge pattern in a 2×2 window and then generate the training sets for the POCS to update the estimate through an iteration process.

The Isophote-Based interpolation methods proposed in [14]-[19] also known as level-set contours. In these methods, interpolation is treated as a variational problem. In [14], it utilizes both the Concentric Circular Shift Model and the Line Shift Model in computing the unknown pixels intensities by using the iteration process. The method proposed in [15] gives a geometry-based interpolation approach that smoothly fits the isophote (intensity level curve) contours at all points in an image rather than just at selected contours. Moreover, a restriction that isophotes of the interpolated images should have the minimum curvature was introduced to the interpolation equation in [16]. After solving the partial differential equations (PDEs), curvatures of the interpolated isophotes are smoothed and the aliasing artifacts are reduced. In the method proposed in [17], the interpolated image must have the minimal error in the image gradient angle at the original pixel position. It is shown by [17], that under this restriction and after some approximations, the resulting differential equation can be solved in a one-pass discrete form for scaling by power of two. In [18], it presents a new formulation of the regularized image up-sampling problem. It uses the level set method to minimize the objective function which subsequently makes use of the bounded-total-variation regularizer, so that it can produce a crisp edge without introducing ringing or other artifacts. Recently, a new isophote-oriented interpolation method was proposed in [19]. It calculates the gradient magnitude in the image for detects the ridges and estimates their orientation, then applies the parallelogram interpolation kernel for the edge region and bilinear for the smooth region. It results in a visually pleasing edges.

The other type of the interpolation approach is optimal recovery. An optimal recovery approach to interpolation was proposed in [20]. It states that the problem of estimating a missing pixel (optimal recovery), is equivalent to a problem of approximation the representer of a missing sample by a linear combination of the representers of the known samples. The optimal solution can be found by minimizing an integrated squared error in the frequency domain. After that the adaptive optimal recovery method was proposed in [21], [22]. This method [22] is based on adaptively determining the quadratic signal class from a set of training vectors and then using the optimal recovery theory to estimate the missing pixels. However, these optimal recovery techniques are approaches with highly computational complexity. Therefore, they are not commonly used.

Recently, some other types of the interpolation methods were proposed such as the Iterative Deconvolution techniques [23], Total Variation Quasi-solution method [24], and Polyphase scheme [25]. In [23], it makes use of the high-quality motion deblurring method which was proposed in [26] as the deconvolution of the low-resolution image and then performs the iteration by using the feedback-control which consists of re-convolution and pixels substitution. After the iterative process, the interpolated image has a sharpen edge than that of other conventional liner interpolation approaches. The Total Variation Quasi-solution method [24] uses a total variation function as a stabilizer in Tikhonov regularization [27] which will not oversmooth or displace the edges. The Polyphase interpolation proposed in [25] tries to express the MMSE formulation in terms of polyphase components and creating a filter using the magnitude response and zero-phase filter. It yields a good result, but it's required a large computational cost.

The most popular method that usually draws many researcher attractions is the Edge-Directed interpolation method. In the traditional Edge-Directed interpolation method, it is usually starts by edge direction detection and then interpolated along the edge direction such as those in [1], [28]-[37]. In the method [28], it proposed a scheme that detects edges and fits them with some templates to improve the visual perception of interpolated image. Subsequently, ref [29] tries to use some predetermined edge patterns to optimize the parameters in the interpolation operator. The method that is proposed in [30] is a general method of the Edge-Directed interpolation scheme that detects the edge first, and then interpolate along the edge. Finally, an edge sharpening operation is applied to the interpolated image. Although the above method can product a better visual quality of the interpolated image, but the computational cost is relevantly high. Therefore, a faster method of edge-directed interpolation is proposed in [32], it change the original non-directional bicubic interpolation kernel to the directional interpolation kernel called “Gradient-based Kernel Composition” by using the local gradient information. It results in low computational cost, but with some reduction in the image quality. A Fine Edge-Preserving interpolation was proposed in [33]. It uses the edge direction and the edge pattern to determine the use of the vertical or horizontal gradient operators during the interpolation. In 2008, an edge-directed interpolation using segmentation scheme was proposed in [34]. It segments the image into three types of regions: smooth, well-defined edges and texture regions, and using difference interpolation methods in difference regions. Recently, a Visual Attention Model was proposed in [35], [36]. Both methods use the high-quality saliency map of an image to determine the regions which are more sensitive in the human visual system. The difference between [35], [36] method is that, after using the predefine filtering masks to find the edge direction, method [35] uses the predefine interpolation masks for the interpolation and method [36] uses the Particle Swarm Optimization (PSO) to find the interpolation masks for difference edge directions. In the above interpolation scheme, all of them are single pass. However, in 1996, the iterative approach of Edge-Directed Interpolation was proposed in [1]. It combines the rendering and correction into a feedback system and claims that this produces sharper images. Recently, an iterative edge directed interpolation was proposed in [37]. It finds that a descent direction that has an analytic form and using the Armijo rule to identify the step length, will result in a high-speed implementation and produce sharper edge.

The qualities of the interpolated image using traditional Edge-Directed interpolation methods are dependent on the edge direction detection step. As the traditional edge direction detection only quantized the edge orientation into a finite number of choices such as horizontal, vertical or diagonal etc. However, in the natural image, the edges contain infinite directions. This causes the limitation of the traditional Edge-Directed interpolation methods and limits the quality of the interpolated image. A break through of the Edge-Directed interpolation happened when the New Edge-Directed Interpolation (NEDI) was proposed in [2]. The NEDI was developed base on the previous work of the same group of researchers on the edge-directed prediction for lossless image coding [38]. The prediction method in [38] shows that the covariance-based adaptation is able to tune the prediction to match an arbitrarily oriented edge. In the NEDI scheme in [2], it proposed to estimate the covariance of high-resolution image from the covariance of low-resolution image, and then interpolate the missing pixels using the Wiener filter equation which is base on the estimated covariances. After the NEDI was proposed in 2001, there are many researchers working on the new covariance-based interpolation scheme [39]-[42] or improving the original NEDI method [43]-[47].

The covariance-bases interpolation approach was adapt in the optimal recovery scheme and becomes the Adaptively Quadratic (AQua) Image Interpolation method [39]. Another type of the covariance-based interpolation is called Edge-Guided Image Interpolation via Directional Filtering and Data Fusion method which was proposed in [40]. This proposed method partitions the neighboring pixels of each the missing pixels into two directional subsets that are orthogonal to each other, and then fuse the two directional filtering results into a final result. It results in image with preserved edge sharpness and reduces the ringing artifacts. Based on the fusion technique, a novel interpolation scheme for noisy images is developed in [41]. The proposed method treats both denoising and interpolation as an estimation problem and solves it by using three different directional conveniences and then fusion is done according to the inverse ratio of the estimation error. It can better suppress the many noise-caused artifacts in the enlarged image while preserving the image fine structures. Recently, a novel soft-decision approach is proposed for edge-directed interpolation in [42]. It coupled with a piecewise autoregressive image model, to minimum the Means Square Error in a moving window and estimates a block of missing pixels rather than a single missing pixel in the NEDI.

Although, there are many new covariance-based interpolation schemes, the development based on the original NEDI schemes still continuous. In 2003, the Improved edge-directed image interpolation was proposed in [43]. It proposed to use a new sample structure according to the sixth-order linear prediction equation in the second pass interpolation and to keep the sample structure in the first pass interpolation unchanged. In the same year, the "Content-adaptive video up-scaling for High Definition displays" proposed to use a new sample structure according to a twelve-order linear prediction equation in the first and second pass interpolations with the help of an anti-alias Filtering in [44]. The anti-alias Filtering and the new sample structure can reduce most of the artifacts that cause by the original NEDI method, but it cannot keep the sharpness of the edge region in the image. A new single-pass NEDI scheme is proposed in [45]. It based on the new relation between the high-resolution and the low-resolution pixel grids to interpolate an image in a single-pass step. In 2008, the paper Accuracy Improvements and Artifacts Removal in Edge Based Image Interpolation proposed several improvements to the original NEDI method in [46]. It proposes to use the circular windows and several new constraints on the NEDI and results in a better visual quality.

Recently, a Modified Edge Directed Interpolation was proposed by Tam, Kok and Siu [47]. It makes use a totally of four training windows that near the missing pixel and choosing the window that has the highest covariance energy to compute the missing pixel value by using the covariance-based method. It results in a better interpolated image with more continuous edge than NEDI, but it still suffers from the artifact that near to the high frequency and texture regions.

Chapter 3

Development of NEDI

In this Chapter, the development of NEDI will be presented. The general objective of all edge-directed adaptive approach is to utilize the edge information to modify the interpolation scheme so that smoothing will not performed across an edge. As a result, it can interpolate an image with sharpness and continuity edges. The more well-known algorithm is the Edge-Directed Interpolation (EDI) proposed by Allebach and Wong [1]. Then many researchers developed the new algorithms that are base on the idea of the EDI. The most outstanding method is the New-Edge-Directed Interpolation (NEDI) proposed by Li and Orchard in [2]. They developed a basic covariance-based interpolation scheme. After that, the Improved Edge-Directed Interpolation (IEDI) was proposed in [43] and the Modified Edge Directed Interpolation (MEDI) was proposed in [47] by Tam, Kok and Siu. The IEDI improves the NEDI by changing the interpolation structure and MEDI improves the NEDI by changing the training window position adaptively. The following sections will review the Development of the NEDI by presenting the details of the EDI, NEDI, IEDI and MEDI.

3.1 Edge-Directed Interpolation (EDI)

The development of the Edge-Directed Interpolation is based on the human visual system. In the human visual system, human perception tends to firstly pick attended regions which correspond to prominent objects in am images. Therefore, the object boundaries are very important. As a reason, interpolation algorithms should make more effort to interpolate the edges regions than the smooth regions. It results in the development of the Edge-directed Interpolation.

The EDI algorithm can achieve high accuracy in estimating and interpolating edges and high frequency components in the image by including the edge detection step. In order words, the interpolation step will be changed according to the edge detection step.

The block diagram of the EDI proposed by [1] is shown in Fig. 3.1. The EDI algorithm extracts the edge information from the low-resolution image and uses a sub pixel edge estimation technique to generate a high-resolution edge map. The high-resolution edge map is then use to guide the interpolation of the low-resolution image to generate the high-resolution image.

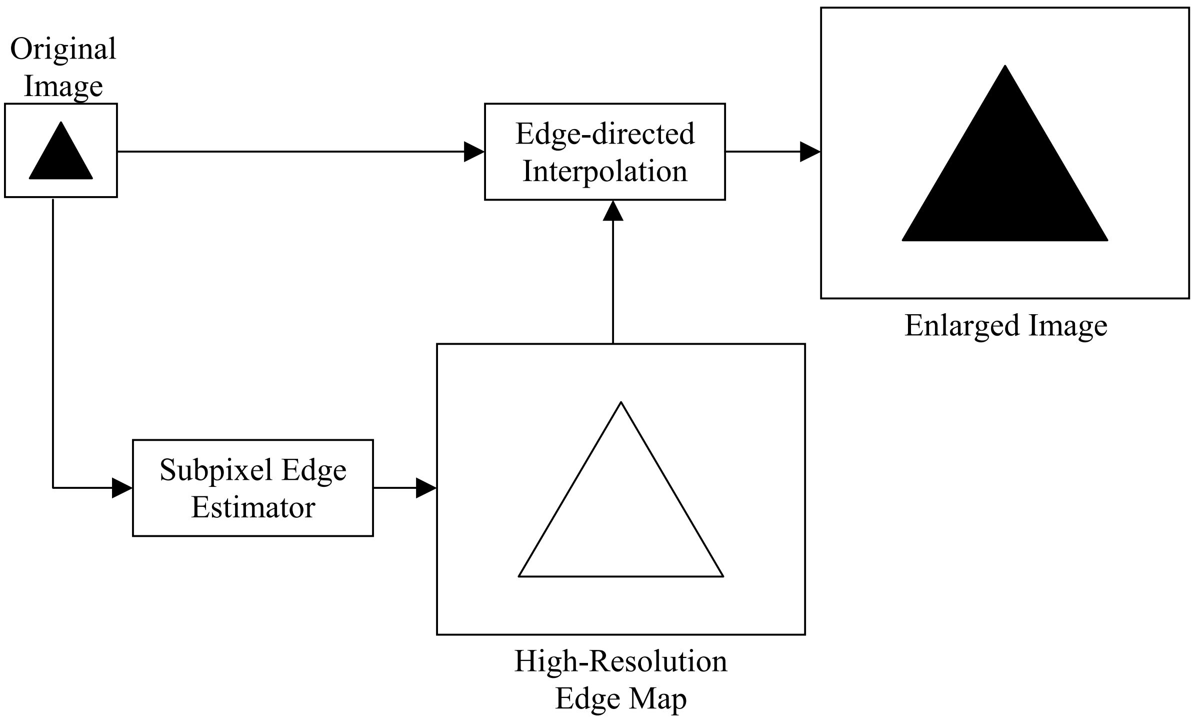


Figure 3.1: Framework for Edge-directed Interpolation algorithm Proposed by [1]

During the estimation of the high-resolution edge map, we firstly filter the low-resolution image with a simple rectangular center-on-surround-off (COSO) filter with a constant positive center region embedded within a constant negative surround region. Using the COSO filter, it results in a very good approximation to the edge map generated with a true Laplacian-

Gaussian (LoG) filter, but the computational complexity of the COSO filter is higher than the LoG filter. To determine the high-resolution edge map, we linearly interpolate the COSO filter output between points on the low-resolution lattice.

After the high-resolution edge map is generated, the iterative interpolation process will be started after the pre-processing of the low-resolution image. After that the edge-directed rendering and data correction will be applied iteratively. The rendering phase is based on the modified bilinear interpolation to prevent interpolation across edges, as determined from the estimated high-resolution edge map. During the correction phase, it modifies the mesh values on which the rendering is based on, to account for disparity between true low-resolution data, and that predicated by a sensor model operating on the high-resolution output of the rendering phase.

The proposed method can preserve the edge details in the interpolated image than the other conventional interpolation methods. However, EDI suffers from the ringing artifacts of the COSO filter, which results in estimation overshoots and undershoots. Moreover, this method also suffers from the inherent problems on the use of the edge map. The edge information in the edge map is used to guide the interpolation process. However, the inaccurate estimation of the edge location in the edge map will results in a poor interpolated image.

Although the Edge-directed interpolation (EDI) presented in [1] takes the advantage of both non-adaptive and adaptive interpolation, it suffers from the image edge detection accuracy problem which reduces the visual quality of the interpolated image.

3.2 New Edge-Directed Interpolation (NEDI)

The New Edge-Directed Interpolation is a covariance-based interpolation method. It does not need to estimate the edge orientation before the interpolation start. The edge orientation will be automatically included during the NEDI interpolation. Moreover, the NEDI is developed based on the edge-oriented image model. The edge-oriented image modelling approach is able to present a geometric and statistical analysis of the image, which treats edges differently from smooth regions and preserve the edge orientation. In the following section, we are going to discuss how NEDI apply the geometric and statistical properties in the image interpolation to interpolate a high quality image.

3.2.1 Geometric Properties of Image Edges

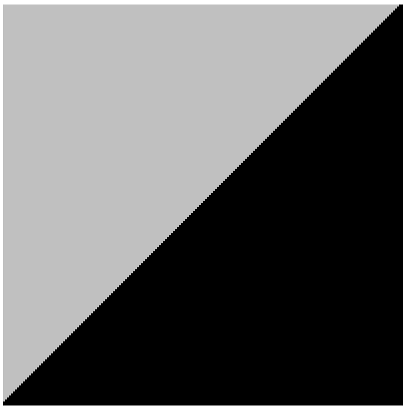


Figure 3.2: Image intensity of the edge region

In Fig. 3.2, we can observe that the image intensity evolves slowly along the edge direction than across the edge direction and the image intensity is almost homogeneous along the edge direction. This observation is the fundamental property of an ideal step edge (known as geometric regularity). Geometric regularity has important effects on the visual quality of an image. It determines the sharpness of edges and the artifacts that appearing in the image such as ringing or jagged. As a result, many algorithms based on the geometric regularity have been developed for high quality interpolation. Some of them use the geometric analysis based on local gradient or transition pattern, and these can provide accurate estimation of the edge orientation such as edge direction. For NEDI, it adopts an accurate edge-orientated detection approach which does not require to estimate the edge orientation before interpolation. It is because, the estimation of the edge orientation in NEDI is integrated in the algorithm itself.

3.2.2 Statistical Structure of Images

A considerable proportion of the models on natural image statistics are based on one particular statistical property. In NEDI, it uses the statistical property of point estimation with some numerical characteristics of the distribution such as mean and variance. Figure 3.3 illustrates the generic problem of point estimation in 1D case. In Figure 3.3, the missing value of the $X(n)$ can be obtained by using interpolation. This interpolation problem can be defined as an estimation problem of the missing value $X(n)$ from the neighboring data in the local neighbourhood M_{Subset}. The data set in the M_{Subset} is also known as the "training/sample window". We can choose to use different training windows in different problems. The Goal of the estimation is to find the right neighbourhood to obtain the optimal interpolation results.

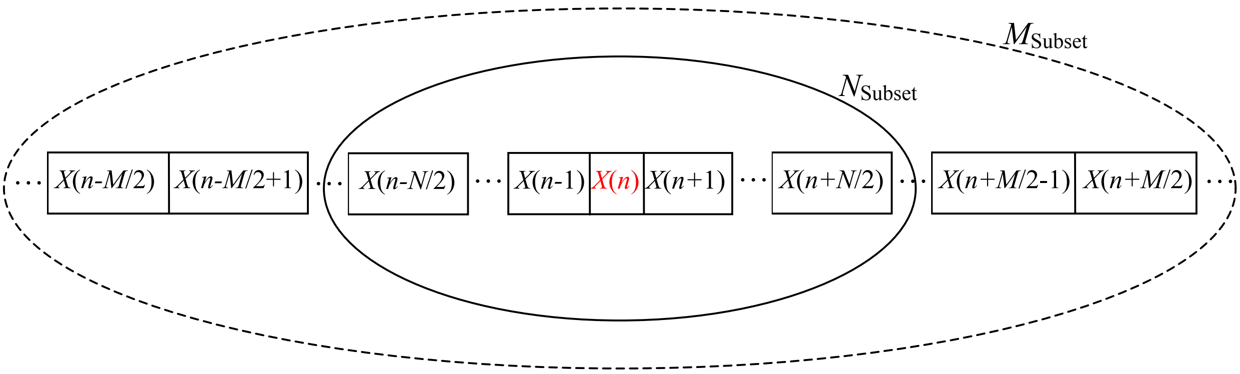


Figure 3.3: Illustrative example of point estimation

For finding X(n) in Fig. 3.3, we need to selecting a set of candidates and to apply weighting to a universal set of neighbors. In this case, we select the candidates in the N_{Subset}.

The same assumption happens when the 1D point estimation in Fig. 3.3 applies to the 2D natural-image case. A reasonable assumption made with the natural-image source is the N-th order Markovian property. That is, we only need to consider the N nearest causal neighbors in the prediction as

$$X'(n) = \sum_{\substack{k \in N_{subset} \\ k \neq n}} \alpha_k X(k) \tag{3.1}$$

where $X'(n)$ is the estimation values, α_k is the weighting factor for the neighbouring data $X(k)$. By solving the optimal values for all α_k, the optimal value of the unknown values $X'(n)$ can be obtained by using the Eq. (3.1). Moreover, this equation is equivalent to a linear optimization problem, for which the least squares optimization can be used to find the optimal weighting factor.

Details of the least squares optimization and the use of the least squares optimization in NEDI will be discussed in later sections.

3.2.3 Two Assumptions in NEDI

Two reasonable assumptions are made in the NEDI to facilitate the estimation of the unknown pixels from the statistical structure of the neighboring known pixels.

1. It models the natural image source by a second-order locally stationary Gaussian process; and
2. based on the geometric duality between the low-resolution and the high-resolution covariances which couple the pair of pixels at the different resolution but along the same orientation.

The first assumption demonstrates that the covariance-based adaptation is able to tune the prediction process to achieve an arbitrarily oriented edge interpolation and the second assumption can enable us to estimate the high-resolution covariance from its low-resolution covariance. With the above two assumptions, the NEDI can be derived from the mathematical point of view.

3.2.4 Mathematical Derivation of NEDI

In the NEDI, the Least square optimization is used for obtaining the optimal solution, i.e. to obtain the optimal set of the prediction coefficients or weighting factors, without the need of the estimation of the edge orientation and using the edge pixels. The Least square optimization will automatically estimate the unknown pixels according to its orientation. When the density of sampling lattice is increasing, the prediction coefficients will be ideally aligned along the edge orientation in order to achieve the minimal interpolation error. Therefore, using Wiener filtering can interpret the optimal linear estimation as a linear prediction problem. The linear prediction based interpolation will make use of the assumption that a natural image is a second-order locally stationary Gaussian process; hence the linear prediction approach can be used to estimate the value of the unknown pixels. Eq. (3.2) is the transfer function of Wiener filter, where $y(n)$ is the output of the filter, $x(n)$ is the input of the filter and $w(n)$ is the Wiener filter coefficients.

$$y(k) = \sum_{n=0}^{\infty} w(n)x(k-n) \tag{3.2}$$

The error between the desired response $d(n)$ and the output is

$$e(k) = d(k) - y(k) \tag{3.3}$$

and hence the mean-square error is given by

$$MSE = E[e^2(k)] \tag{3.4}$$

In order to find the minimum mean-square error, we differentiate both side w.r.t $w(n)$ and we have

$$\nabla MSE\Big|_n = \frac{\partial MSE}{\partial w(n)} = \frac{\partial E[e^2(k)]}{\partial w(n)} \tag{3.5}$$

With the use of chain rule of differentiation

$$\frac{\partial MSE}{\partial w(n)} = 2E\left[e(k)\frac{\partial e(k)}{\partial w(n)} \right] \tag{3.6}$$

Substitute Eq. (3.2) and Eq. (3.3) in the $\frac{\partial e(k)}{\partial w(n)}$, we have

$$\frac{\partial e(k)}{\partial w(n)} = \frac{\partial\left(d(k) - \sum_{n=0}^{\infty} w(n)x(k-n) \right)}{\partial w(n)} = -x(k-n) \tag{3.7}$$

Substitute Eq. (3.7) into Eq. (3.6), we have

$$\frac{\partial MSE}{\partial w(n)} = -2E\left[e(k)x(k-n)\right] \tag{3.8}$$

Then the minimum error can be obtained by setting the $\frac{\partial MSE}{\partial w(n)}$ to 0, which yields

$$E\left[e_0(k)x(k-n)\right] = 0, \qquad \text{for all } n, \tag{3.9}$$

where the $e_0(n)$ is the error obtained by using the optimal filter.

We may apply the Wiener filtering to the problem of the linear prediction, and use the above derivation to generate an N^{th} order linear prediction as shown in below equation.

$$x'(k) = \sum_{n=1}^{N} \alpha(n)x(k-n) \tag{3.10}$$

Then the error of this prediction become,

$$\begin{aligned} e(k) &= x(k) - x'(k) \\ &= x(k) - \sum_{n=1}^{N} \alpha(n)x(k-n) \end{aligned} \tag{3.11}$$

where $x(k\text{-}n)$ is the sequence segment, $\alpha(n)$ is the filter coefficients, n=1,2… N and N is the filter length.

The criterion for the linear prediction to obtain the optimal solution is to use the mean-square error (*MSE*).

$$\begin{aligned} MSE &= E\left[e^2(k)\right] \\ &= E\left[\left(x(k)-x'(k)\right)^2\right] \\ &= E\left[\left(x(k)-\sum_{n=1}^{N}\alpha(n)x(k-n)\right)^2\right] \end{aligned} \tag{3.12}$$

In order to find the minimum *MSE*, we differentiate both side w.r.t $\alpha(i)$ and set it to zero.

$$\frac{\partial E\left[e^2(k)\right]}{\partial\alpha(i)} = E\left[2\left(x(k)-\sum_{n=1}^{N}\alpha(n)x(k-n)\right)\frac{\partial}{\partial\alpha(i)}\left(x(k)-\sum_{n=1}^{N}\alpha(n)x(k-n)\right)\right]=0 \tag{3.13}$$

where *i* is the simple position inside the segment. i.e. *i*=1,2…*M*

In Eq. (3.13), all the derivatives on the RHS are zero, except for the case that $\alpha(n)=\alpha(i)$ .i.e.

$$\frac{\partial}{\partial\alpha(i)}\left(x(k)-\sum_{n=1}^{N}\alpha(n)x(k-n)\right)=-x(k-i) \tag{3.14}$$

Therefore, the Eq. (3.13) becomes

$$\begin{aligned} &E\left[-2\left(x(k)-\sum_{n=1}^{N}\alpha(n)x(k-n)\right)x(k-i)\right]=0 \\ &E\left[\left(x(k)-\sum_{n=1}^{N}\alpha(n)x(k-n)\right)x(k-i)\right]=0 \\ &E\left[x(k)x(k-i)\right]=\sum_{n=1}^{N}\alpha(n)E\left[x(k-n)x(k-i)\right] \end{aligned} \tag{3.15}$$

Eq. (3.15) can be expressed in terms of the autocorrelation matrix **R** and the cross-correlation vector $\vec{r}$ as

$$\vec{r}=\vec{\alpha}\mathbf{R} \tag{3.16}$$

where
$$\vec{r}=E\left[x(k)x(k-i)\right], \tag{3.17}$$

$$\mathbf{R}=E\left[x(k-n)x(k-i)\right]. \tag{3.18}$$

The advantages for solving the optimal linear prediction coefficients by the Wiener filtering is that Wiener filtering optimises the prediction accuracy from point-wise optimal to statistical optimal. Optimal solutions for the prediction coefficients can be obtained through the iterative optimization process that achieves the optimal estimation of the unknown. In the next section, we are going to discuss how Wiener filtering approaches be applied to image interpolation in NEDI.

3.2.5 Linear Prediction in NEDI

In the previous section, we know that the NEDI mode the natural image is considered as a second-order locally stationary Gaussian process which assumes that the statistical structure of the unknown pixel is identical to that of the known pixels in a local block. Therefore, the weighting factors can be solved as a linear prediction problem according to the neighboring known pixels. The linear prediction can help us to minimize the prediction error, so that to minimize the interpolation error. In NEDI, the statistical optimization technique using with the Wiener filtering can provide a good solution for the linear prediction.

The main idea on apply the linear prediction in NEDI is based on the second assumption of the NEDI which is the geometric duality property. The second assumption states that the low-resolution covariance and the high-resolution covariance coupled to the same pixel are identical to each other. This idea can be easily explained in Fig. 3.4. In Fig. 3.4, we can observe that the placements of the edge with respect to the low-resolution block and that to the high-resolution block are most likely the same. It implies that the covariance obtained in the low-resolution block is very close to that of the high-resolution block. Therefore, the low-resolution covariance and the high-resolution covariance can be the same.

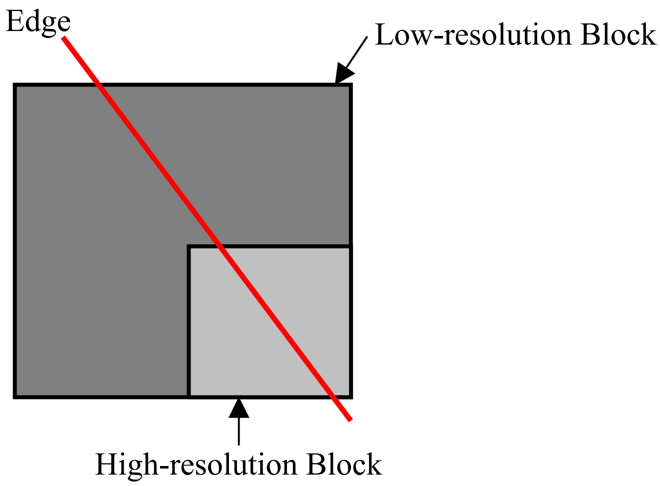


Figure 3.4: Geometric duality in NEDI

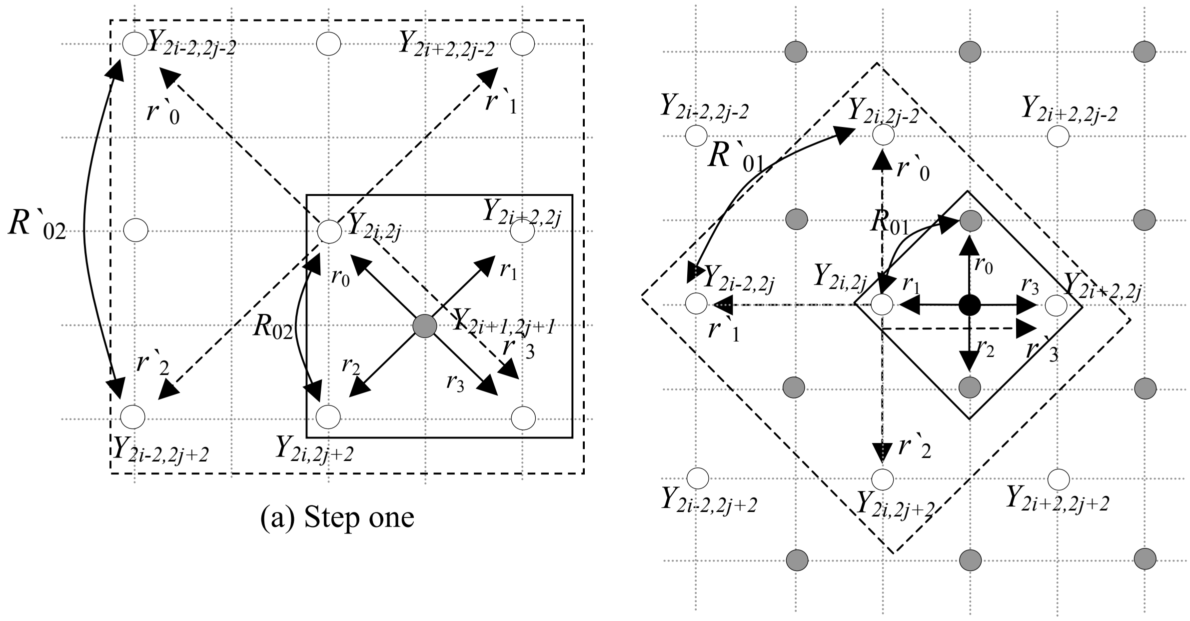
Let us refer to Fig. 3.5 as an illustration on how linear prediction works for NEDI. Consider the interpolation of an image X with size $H \times W$ to a high-resolution image Y with size $2H \times 2W$. The white dots in Fig. 3.5 denote the original low-resolution pixels $X_{i,j}$ and the gray dots denote the unknown pixel $Y_{2i+1,2j+1}$ which is to be estimated by the NEDI step one. Note that, $X_{i,j}=Y_{2i,2j}$. The NEDI step one makes use of a fourth-order linear prediction to interpolate the unknown pixels $Y_{2i+1,2j+1}$ from the four neighboring pixels $\{Y_{2i,2j}, Y_{2i+2,2j}, Y_{2i,2j+2}, Y_{2i+2,2j+2}\}$ as shown below:

$$Y_{2i+1,2j+1} = \sum_{k=0}^{1}\sum_{l=0}^{1} \alpha_{2k+l} Y_{2(i+k),2(j+l)} \tag{3.19}$$

We may apply the derived solution in Eq. (3.16) to obtain the optimal prediction coefficients set $\boldsymbol{\alpha}$.

$$\boldsymbol{\alpha} = \mathbf{R}_{yy}^{-1}\mathbf{r}_y \tag{3.20}$$

where $\boldsymbol{\alpha} = [\alpha_0, \ldots \alpha_3]$, the auto-covariance $\mathbf{R}_{yy}$ is a square matrix containing sixteen R_{kl} with k, l = $[0,\ldots 3]$ for the position in the sample points set, and the cross-covariance of $\mathbf{r}_y$ contains four r_l for $l=[0,\ldots 3]$. For example, r_0 is defined by $E[Y_{2i,2j}Y_{2i+1,2j+1}]$ and R_{02} is defined by $E[Y_{2i,2j}Y_{2i,2j+2}]$ as shown in Fig. 3.5 (a).



(a) Step one

(b) Step two

Figure 3.5: Illustrative example of NEDI

The high-resolution cross-covariance $\mathbf{r}_y$ is not available now, because of the center pixel $Y_{2i+1,2j+1}$ is to be predicted. This difficulty can be overcome by the fact that the statistics of the pixels with respect to the low-resolution block and that of the high-resolution block are most likely to be similar. As a result, the auto-covariance and cross-covariance coefficients among the high-resolution block will be mostly alike that of the low-resolution block. Therefore, the low-resolution covariance $\mathbf{R}`_{yy}$ and $\mathbf{r}`_y$ will be used instead, for the calculation.

According to the classical covariance method, $\mathbf{R}`_{yy}$ and $\mathbf{r}`_y$ can be calculated by the following equation:

$$\mathbf{R}` = \frac{1}{M^2}\mathbf{C}^T\mathbf{C}, \quad \mathbf{r}` = \frac{1}{M^2}\mathbf{C}^T\boldsymbol{y} \tag{3.21}$$

where $\boldsymbol{y} = [y_1, \ldots, y_k, \ldots, y_{M\times M}]^T$ is a data vector containing $M\times M$ pixels inside a local window and $\mathbf{C}$ is a $M^2\times 4$ data matrix,

$$\mathbf{C} = \begin{bmatrix} C_{1,1} & C_{1,2} & C_{1,3} & C_{1,4} \\ \cdots & \cdots & \cdots & \cdots \\ C_{k,1} & C_{k,2} & C_{k,3} & C_{k,4} \\ \cdots & \cdots & \cdots & \cdots \\ C_{M\times M,1} & C_{M\times M,2} & C_{M\times M,3} & C_{M\times M,4} \end{bmatrix}$$

whose k^{th} row vector contains four nearest neighbors of y_k in the diagonal direction as shown in Fig. 3.6, i.e. if $y_k = Y_{2i,2j}$, then the four nearest neighbors are $\{Y_{2i-2,2j-2}, Y_{2i+2,2j-2}, Y_{2i-2,2j+2}, Y_{2i+2,2j+2}\}$ as shown in Fig. 3.5 (a) and in Fig. 3.6, if $y_k = y_1$, then the four nearest neighbors are $\{C_{1,1}, C_{1,2}, C_{1,3}, C_{1,4}\}$. When the local window is 4×4, vector $\boldsymbol{y}$ and matrix $\mathbf{C}$ can be find according to the structure as shown in the Fig. 3.6.

According to Eq. (3.20) and Eq. (3.21), we have

$$\boldsymbol{\alpha} = \left(\mathbf{C}^T\mathbf{C}\right)^{-1}\left(\mathbf{C}^T\boldsymbol{y}\right) \tag{3.22}$$

After the calculation of filter coefficients of $\boldsymbol{\alpha}$, the interpolated value of $Y_{2i+1,2j+1}$ can be then obtained by substituting Eq. (3.22) into Eq. (3.19).

The remaining pixels $Y_{2i,2j+1}$ and $Y_{2i+1,2j}$ are obtained by the same methods with a scaling $2^{1/2}$ and a rotation factor of $\pi/4$ as shown in Fig. 3.5 (b), where the gray pixels are the high-resolution pixels estimated from step one.

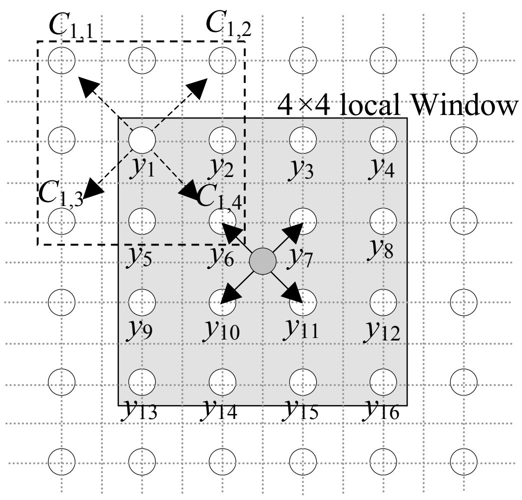


Figure 3.6: Structure of the 4×4 local window in NEDI

3.3 Improved Edge-Directed Interpolation (IEDI)

In 2003, a new sixth order linear prediction equation was proposed in [43]. The new sixth order linear prediction equation replaces the original fourth order linear prediction equation in step two of the NEDI as shown in Fig. 3.7.

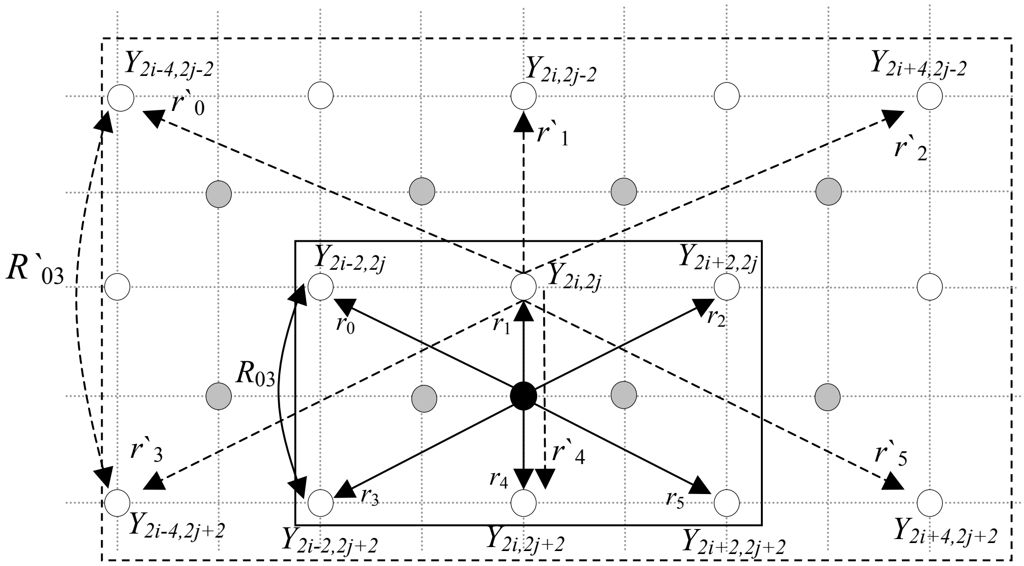


Figure 3.7: Illustrative example of the interpolation step two in IEDI

The interpolation step one in IEDI is the same as step one in NEDI which uses a fourth order linear prediction equation as shown in Fig. 3.5 (a).

The new sixth order linear prediction equation in IEDI is used to predict the unknown pixels $Y_{2i+1,2j}$ and $Y_{2i,2j+j}$ as shown in Fig. 3.7. We have

$$Y_{2i,2j+1} = \sum_{k=0}^{1}\sum_{l=-1}^{1} \alpha_{2k+l} Y_{2(i+l),2(j+k)} \tag{3.23}$$

$$Y_{2i+1,2j} = \sum_{k=0}^{1}\sum_{l=-1}^{1} \alpha_{2k+l} Y_{2(i+k),2(j+l)} \tag{3.24}$$

The coefficient set, $\alpha = [\alpha_{0,}\ \alpha_{1,} \ldots\alpha_5]$, can be obtained from Eq. (3.20) with the auto-covariance R_{yy} matrix containing thirty six R_{kl}, for $k \geq 0$, $l = [0,\ldots 5]$, and the cross-covariance of r_y containing six r_l , for $l = [0,\ldots 5]$.

The same Eq. (3.22) can be used to find the filter coefficients, α, but now C becomes a $M^2 \times 6$ matrix whose k^{th} row vector contains the six nearest neighbors of y_k, i.e. if $y_k = Y_{2i,2j}$, the six nearest neighbors are $\{Y_{2i-4,2j-2}, Y_{2i,2j-2}, Y_{2i+4,2j-2}, Y_{2i-4,2j+2}, Y_{2i,2j+2}, Y_{2i+4,2j+2}\}$ as shown in Fig. 3.7. After substitution, we can find the predicted values of $Y_{2i+1,2j}$ and $Y_{2i,2j+1}$.

3.4 Modified Edge-Directed Interpolation (MEDI)

In 2009, a Modified Edge-Directed Interpolation (MEDI) was proposed in [47]. It proposes to use multiple low-resolution training windows instead of the original single low-resolution training windows in NEDI and IEDI.

The MEDI makes use of the sixth order linearly prediction equation in the interpolation step two which shows in Fig. 3.7 and uses the original fourth order linearly prediction equation as the interpolation step one in MEDI which shown in Fig. 3.5 (a).

In order to reduce the covariance miss-match problem, multiple low-resolution training windows are used. Fig. 3.8 illustrates the four training windows applied in the first step of the MEDI method. The NEDI and the IEDI methods consider the training window as shown in Fig. 3.8 (b) only and the training window is centered at pixel $Y_{2i,2j}$. Compare with the NEDI method, the MEDI method will consider three more training windows centered at pixel $Y_{2i,2j+2}$, $Y_{2i+2,2j}$, and $Y_{2i+2,2j+2}$, as illustrated by Fig. 3.8 (c), (d) and (e) respectively. The covariance signal energy of all training windows will be compared. The higher energy in the training windows, more likely the edge exists. The one contains the highest energy will be applied to the linear prediction in Eq. (3.19). In this example, training window in Fig. 3.8 (c) is applied for the prediction. Similarly, the MEDI method considers six training window candidates in the step two, with such windows centered at $Y_{2i,2j-2}$, $Y_{2i,2j}$, $Y_{2i,2j+2}$, $Y_{2i+2,2j-2}$, $Y_{2i+2,2j}$ and $Y_{2i+2,2j+2}$ (see Fig. 3.7 for the pixels locations).

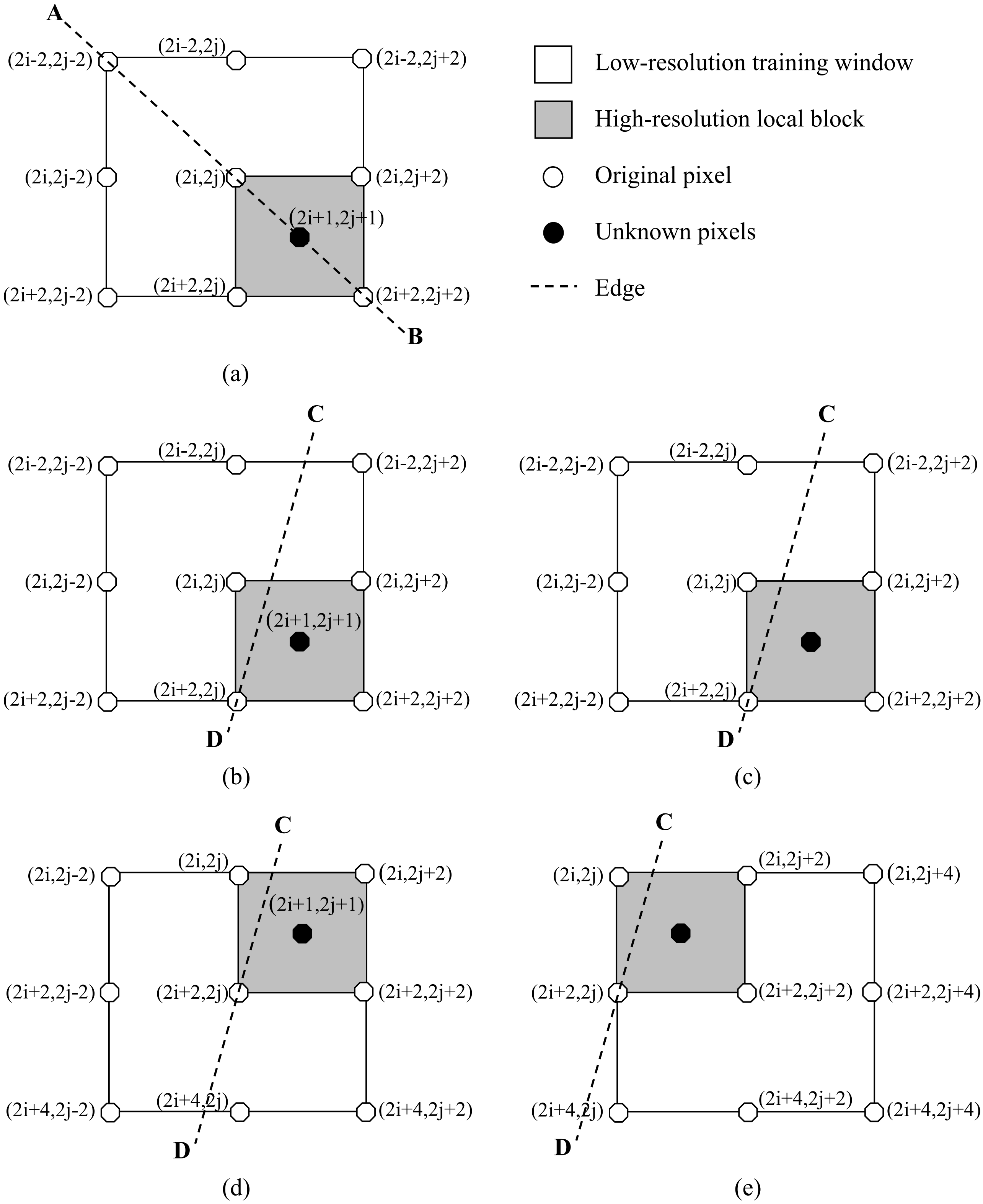


Figure 3.8: Illustrative of (a) the single training window of the NEDI method with edge “AB” and (b-e) the four training windows of MEDI method with edge “CD”.

3.5 Brief comparisons among NEDI, IEDI and MEDI

Now let us have brief comparisons between the performance of the bilinear, NEDI, IEDI and MEDI methods. In Fig. 3.9, the interpolated image (c) using the bilinear interpolation contains many aliasing artifacts so it has the worst visual quality. The interpolated image (d) using the NEDI method results in a more continuous edge and a sharper image than the image in (c) using the bilinear interpolation. However, the line inside of the red ellipse as shown in the Fig. 3.9 (c) did not perform well, as it is a discontinuous line. The interpolated image (e) using the IEDI and (f) using the MEDI can interpolated a continue line as shown inside of the red ellipse. Therefore, the visual quality of the interpolated image using IEDI and MEDI methods are better than that from the Bilinear and the NEDI methods. However, the interpolated image (e) using the IEDI method suffers from some of the artifacts that near to the end of the line as shown inside the red rectangle in Fig. 3.9 (e). We can see that the interpolated image (f) using MEDI method can keep the end of the line continuous and get a sharper image as compared to the interpolated image (e) using IEDI method. However, there are rooms for improvement, for the edges even in cases (e) and (f) are not sufficiently sharp, and the time for the computation can further be improved. These are the points of investigation for this thesis.

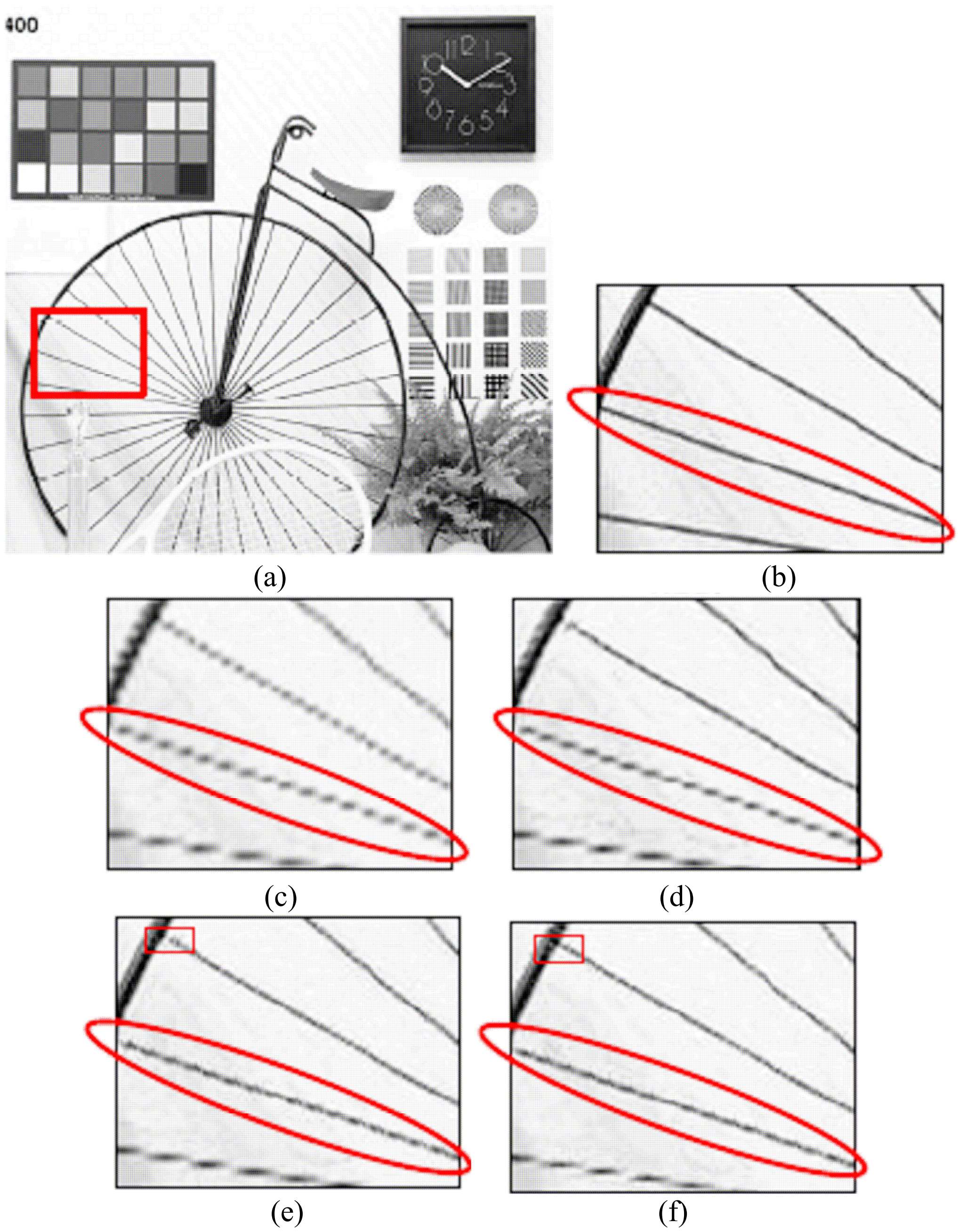


Figure 3.9: (a) Original test image Bicycle and zoomed-in portion of the: (b) Original image and interpolated image (c) reconstructed by Bilinear interpolation, (d) reconstructed by NEDI, (e) reconstructed by IEDI and (f) reconstructed by MEDI.

Chapter 4

Further Improved Edge-Directed Interpolation (FEDI) and Fast EDI for SDTV to HDTV Conversion

4.1 Introduction

In the previous chapter, we have reviewed the development of the NEDI. We know that the NEDI method interpolates an image based on analysing the local structure. The analysing of the local structure will be based on the linear prediction equation. Therefore, different order of the linear prediction equation will result in different optimal solution.

A new direction on improving the visual quality of the New Edge-Directed Interpolation (NEDI) [2] can be done by changing step two of the interpolation in NEDI. Recall that the original NEDI involves two steps: (i) to interpolate points with four symmetrical neighbours hence obtain the predicted value of the center unknown pixel, and (ii) to interpolate the rest of the points between the horizontal and vertical original pixels. The Modified algorithm MEDI [47] has the same interpolation structure with the Improved algorithm IEDI [43] which modifies step two of the fourth order NEDI interpolation structure into a sixth order structure. It makes fuller use of the local relative information in low-resolution to get a high-resolution image with visually better quality. However, four out of six the data points that are used in the sixth order linear prediction equation in the MEDI are too far away from the unknown pixel, so it will introduce more prediction errors into the interpolated image.

In this chapter, a Further Improved Edge-Directed Interpolation (FEDI) algorithm is proposed for image video interpolation, and we also propose a faster edge-directed interpolation to perform interpolation with an enlargement factor of 1.5 (Fast EDI-1.5). When we analyse the linear prediction equation using in the NEDI and IEDI, we find that there have some worst estimated points in the edge regions. It result in reduce the visual quality of the interpolated image. Therefore, we proposed a new algorithm which makes use of six nearest original image pixels and two predicted pixels to interpolate the unknown pixel. It can enhance the edge and reduce the prediction errors compared to that of the MEDI [47] interpolation structure. Another disadvantage of the NEDI method is that the interpolation factor is fixed to 2^n and the computational complexity is very high. Therefore, we develop a fast interpolation approach which makes a new interpolation step based on the edge-directed interpolation algorithm and eliminates unnecessary pixels to directly generate an image with 1.5 times, with lower computational cost. Although many papers have proposed new sampling patterns for the image interpolation, our approach aims at the conversion of SDTV to HDTV and has the lowest computational complexity. In this chapter, we also give results of our study to minimize the number of the sample points using proposed algorithm and its effect on regions with high frequency.

4.2 New Interpolation Structure

The Further Improved Edge-Directed interpolation (FEDI) makes use of the nearest original pixels and predicted pixels to form a new sampling pattern structure of the FEDI step two. This new structure can remove most of the worst estimated points in the IEDI and sharpen the edge of the image by making a full use of the local relative information in the low-resolution image.

The interpolation step one of the FEDI is the same as the NEDI [2], which makes uses of the fourth-order linear prediction to interpolate the unknown pixels $Y_{2i+1,2j+1}$ etc. (see Fig. 4.1)

In step two of the FEDI, we propose to use two interpolated results obtained from the FEDI step one to obtain the interpolation pixel which is indicated in black as shown in Fig. 4.1. Hence we interpolate the unknown pixel $Y_{2i,2j+1}$ (the black dots) by an eighth-order linear predictor from eight neighboring pixels given by the following equation:

$$Y_{2i,2j+1} = \sum_{l=0}^{2} \alpha_l Y_{2(i+l-1),2j} + \sum_{l=0}^{1} \alpha_{3+l} Y_{2(i+l)-1,2j+1} + \sum_{l=0}^{2} \alpha_{5+l} Y_{2(i+l-1),2j+2} \tag{4.1}$$

The white dots in Fig. 4.1 denote the original low-resolution pixels i.e. $X_{i,j}=Y_{2i,2j}$ and the gray dots denote the interpolation result from the FEDI step one.

According to Wiener filtering theory, the optimal Minimum Means Square Error (MMSE) prediction coefficients set $\boldsymbol{\alpha} = [\alpha_0, \alpha_1, \ldots \alpha_7]$, can be obtain as

$$\boldsymbol{\alpha} = \mathbf{R}_{yy}^{-1}\mathbf{r}_y \tag{4.2}$$

where the auto-covariance matrix $\mathbf{R}_{yy}$ contains sixty-four R_{kl}, for k, $l = [0,\ldots 7]$, and the cross-covariance of $\mathbf{r}_y$ contains eight r_l , for $l = [0,\ldots 7]$. The rest of the 2nd step interpolation points can also be obtained by a similar procedure, such as $Y_{2i+1,2j}$ can be computed by Eq. (4.3).

$$Y_{2i+1,2j} = \sum_{l=0}^{2} \alpha_l Y_{2i,2(j+l-1)} + \sum_{l=0}^{1} \alpha_{3+l} Y_{2i+1,2(j+l)-1} + \sum_{l=0}^{2} \alpha_{5+l} Y_{2i+2,2(j+l-1)} \tag{4.3}$$

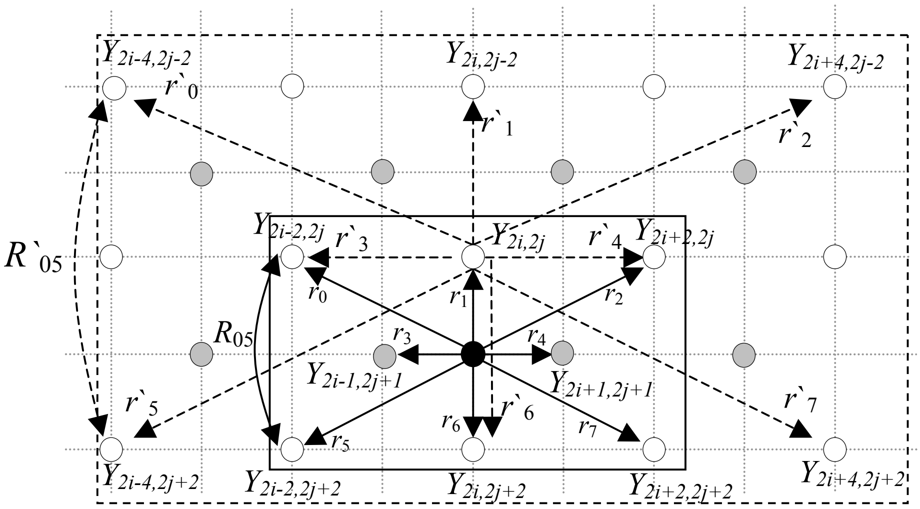


Figure 4.1: Illustrative example of the interpolation step two in FEDI

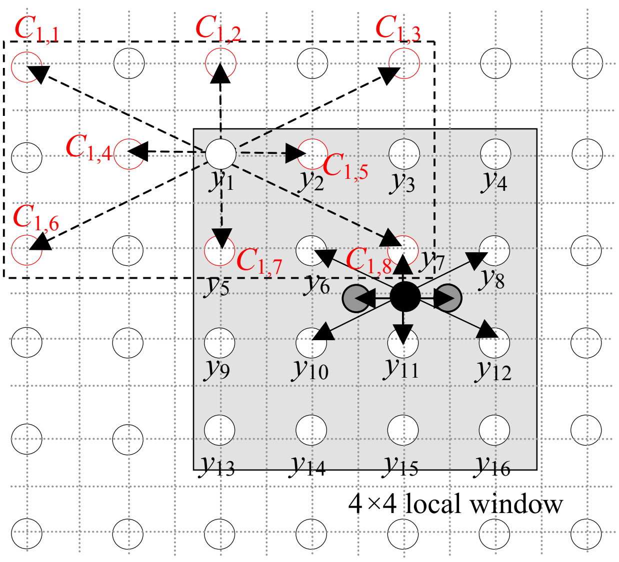


Figure 4.2: An example to illustrate the way of finding vector ***y*** and matrix **C** in step two of the FEDI

The high-resolution cross-covariance $\mathbf{r}_y$ is not available now, because of the center pixel $Y_{2i,2j+1}$ is to be predicted. This difficulty can be overcome by the fact that the statistics of the pixels with respect to the low-resolution block and that of the high-resolution block are most likely to be similar. As a result, the auto-covariance and cross-covariance coefficients among the high-resolution block will be mostly alike that of the low-resolution block. Therefore, the low-resolution covariance $\mathbf{R}`_{yy}$ and $\mathbf{r}`_y$ will be used instead, for the calculation.

According to the classical covariance method, $\mathbf{R}`_{yy}$ and $\mathbf{r}`_{y}$ can be calculated by the following equation:

$$\mathbf{R}`= \frac{1}{M^2}\mathbf{C}^T\mathbf{C},\quad \mathbf{r}`= \frac{1}{M^2}\mathbf{C}^T\boldsymbol{y} \tag{4.4}$$

where $\boldsymbol{y} = [y_1 \ldots y_k \ldots y_{M\times M}]^T$ is a data vector containing $M\times M$ pixels inside a local window and $\mathbf{C}$ is a $M^2\times 8$ matrix,

$$\mathbf{C} = \begin{bmatrix} C_{1,1} & C_{1,2} & \cdots & C_{1,7} & C_{1,8} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ C_{k,1} & C_{k,2} & \cdots & C_{k,7} & C_{k,8} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ C_{M\times M,1} & C_{M\times M,2} & \cdots & C_{M\times M,7} & C_{M\times M,8} \end{bmatrix}$$

whose k^{th} row vector contains its eight nearest neighbors of y_k as shown in Fig. 4.1. i.e. if $y_k = Y_{2i,2j}$, then the eight nearest neighbors are $\{Y_{2i-4,2j-2}, Y_{2i,2j-2}, Y_{2i+4,2j-2}, Y_{2i-2,2j}, Y_{2i+2,2j}, Y_{2i-4,2j+2}, Y_{2i,2j+2}, Y_{2i+4,2j+2}\}$ as shown in Fig. 4.1. In Fig. 4.2, if y_k=y_1, the eight nearest neighbors are $\{C_{1,1}, C_{1,2}, C_{1,3}, C_{1,4}, C_{1,5}, C_{1,6}, C_{1,7}, C_{1,8}\}$. When the local window is 4×4, vector $\boldsymbol{y}$ and matrix $\mathbf{C}$ can be find according to the structure as shown in the Fig. 4.2.

According to (2) and (4), we have

$$\boldsymbol{\alpha} = \left(\mathbf{C}^T\mathbf{C}\right)^{-1}\left(\mathbf{C}^T\boldsymbol{y}\right) \tag{4.5}$$

After substituting α into (4.1) and (4.3), we can compute the predicted values of $Y_{2i,2j+1}$ and $Y_{2i+1,2j}$, respectively.

4.3 Fast EDI for SDTV to HDTV Conversion

In many practical applications, we always encounter the situation of converting a video from its original size to 1.5 of its size. For example, for converting a SDTV video to a HDTV video, we need a conversion ratio of 1.5 times. Let us try this conversion by proposing a new fast approach based on the edge-directed interpolation method in a block-based model.

The Fast EDI block-based model is shown in Fig. 4.3. The red box indicates the block size of each of the interpolation. In each interpolation, we consider eight unknown pixels (the black dots) inside the red box as shown in Fig. 4.3.

This Faster Edge-Directed Interpolation of 1.5 times (Fast EDI-1.5) only needs one-step to achieve the interpolation. We propose to make use of a fourth-order linear prediction to estimate eight unknown pixels at the same time by making use of (6)-(9) in this block-based model:

$$For\ c = 0,2,3 : Y_c = \sum_{k=0}^{1}\sum_{l=0}^{1}\alpha_{c,2k+l}X_{(i+k),(j+l)} \tag{4.6}$$

$$For\ c = 1,4 \quad : Y_c = \sum_{k=0}^{1}\sum_{l=0}^{1}\alpha_{c,2k+l}X_{(i+k+1),(j+l)} \tag{4.7}$$

$$For\ c = 5,6 \quad : Y_c = \sum_{k=0}^{1}\sum_{l=0}^{1}\alpha_{c,2k+l}X_{(i+k),(j+l+1)} \tag{4.8}$$

$$For\ c = 7 \qquad : Y_c = \sum_{k=0}^{1}\sum_{l=0}^{1}\alpha_{c,2k+l}X_{(i+k+1),(j+l+1)} \tag{4.9}$$

Coefficients $\boldsymbol{\alpha}_c = [\alpha_{c,0},\ \alpha_{c,1}\ \ldots\alpha_{c,3}]$ can be calculated from Eq. (4.2) with the auto-covariance $\mathbf{R}_{yy}$ matrix containing sixteen R_{kl} with k, $l = [0,\ldots3]$ and for $c = [0,\ldots7]$. There are totally eight cross-covariances, $\mathbf{r}_{0,y}$, $\mathbf{r}_{1,y},\ldots\mathbf{r}_{7,y}(\mathbf{r}_{c,y})$. Each of them contains four $r_{c,l}$ with $l = [0,\ldots3]$. As the high-resolution covariance $\mathbf{r}_{c,y}$ is not available, the low-resolution covariance

$\mathbf{R}`_{yy}$ and $\mathbf{r}`_{c,y}$ are used for the computation. Fig. 4.3 shows some details of estimating pixel Y_3. The white dots in Fig. 4.3 are original pixels and black dots are pixels to be interpolated.

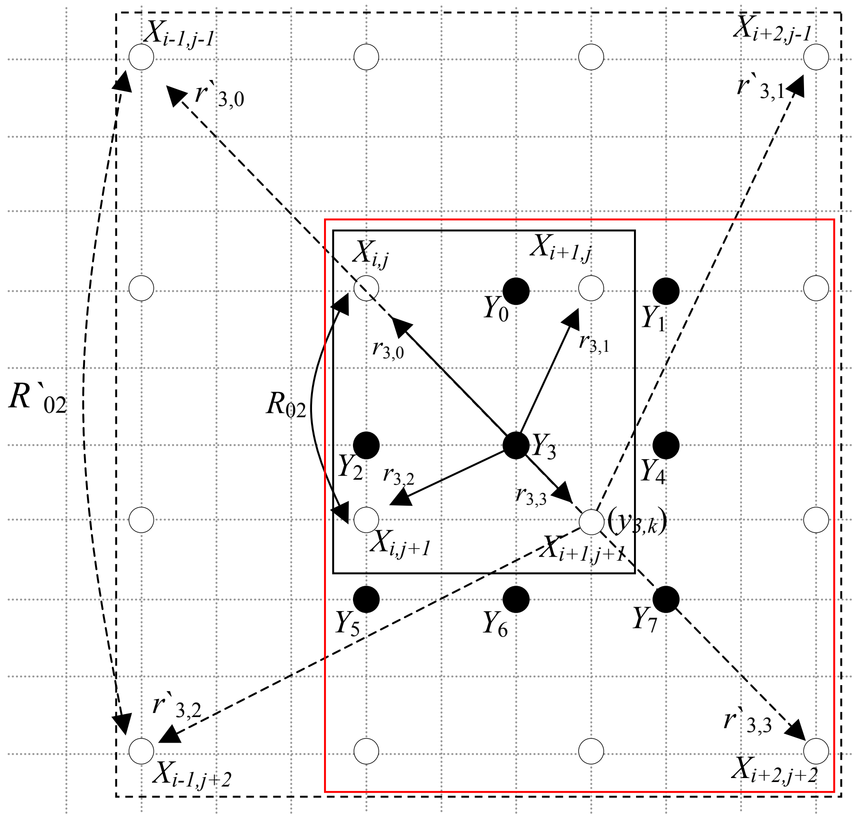


Figure 4.3: Illustrative example of interpolation step in Fast EDI-1.5 using block-based model.

Again $\mathbf{R}`_{yy}$ and $\mathbf{r}`_{c,y}$ can be calculated by:

$$\mathbf{R}` = \frac{1}{M^2}\mathbf{C}^T\mathbf{C}, \quad \mathbf{r}`_c = \frac{1}{M^2}\mathbf{C}^T\boldsymbol{y}_c \tag{4.10}$$

where $\boldsymbol{y}_c = [y_{c,1} \ldots y_{c,k} \ldots y_{c,M\times M}]^T$ is the data vector containing the $M\times M$ pixels inside a local window for a special position c and $\mathbf{C}$ is a $M^2\times 4$ data matrix whose k^{th} row vector contains the four nearest neighbors of $y_{c,k}$. i.e. if $y_{3,k} = X_{i+1,j+1}$, then the four nearest neighbors are $\{X_{i-1,j-1}, X_{i+2,j-1}, X_{i-1,j+2}, X_{i+2,j+2}\}$.

The different between Eq. (4.10) and Eq. (4.4) is that Eq. (4.10) making use of the eight cross-covariance function $\mathbf{r}`_0$, $\mathbf{r}`_1, \ldots \mathbf{r}`_7$ at the same time with only one auto-covariance function $\mathbf{R}`$ inside the red box as shown in Fig. 3. It can be done because we estimate the eight unknown pixels (Y_0, Y_1,… Y_7) at the same time by using four known pixels {$X_{i-1,j-1}$, $X_{i+2,j-1}$, $X_{i-1,j+2}$, $X_{i+2,j+2}$} in this block-based model.

From Eq. (4.5), we can find the filter coefficients, α_c, and then the interpolated value of the Y_c can be obtained by substituting α_c into Eq. (4.6) – Eq. (4.9) according to the position of c.

Finally, after the eight pixels Y_0, Y_1…Y_7 have been estimated, three original pixels $X_{i+1,j}$, $X_{i,j+1}$ and $X_{i+1,j+1}$ need to be deleted, so that the image is scaled up properly by 1.5 times. This is the elimination step which is shown in Fig. 4.4.

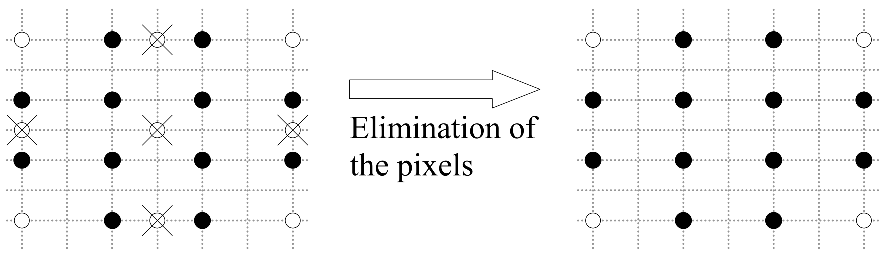


Image after interpolation step   Image after elimination step

Figure 4.4: Illustrative example of elimination step in Fast EDI-1.5.

4.4 Experimental Result

A set of the color images is used for comparing the performance in terms of PSNR among NEDI [2], MEDI [47] and FEDI, with two images as shown in Figs. 4.5 and 4.6. We extract intermediate results of some steps in order to illustrate the effect of various approaches. All the programs were written in C++ language and run on the same platform.

The test images with size $2H \times 2W$ were firstly downsampled by a factor of 2 to images of size $H \times W$. The downsampled images were then enlarged with a factor of 2 to $2H \times 2W$ using the NEDI, MEDI and FEDI.

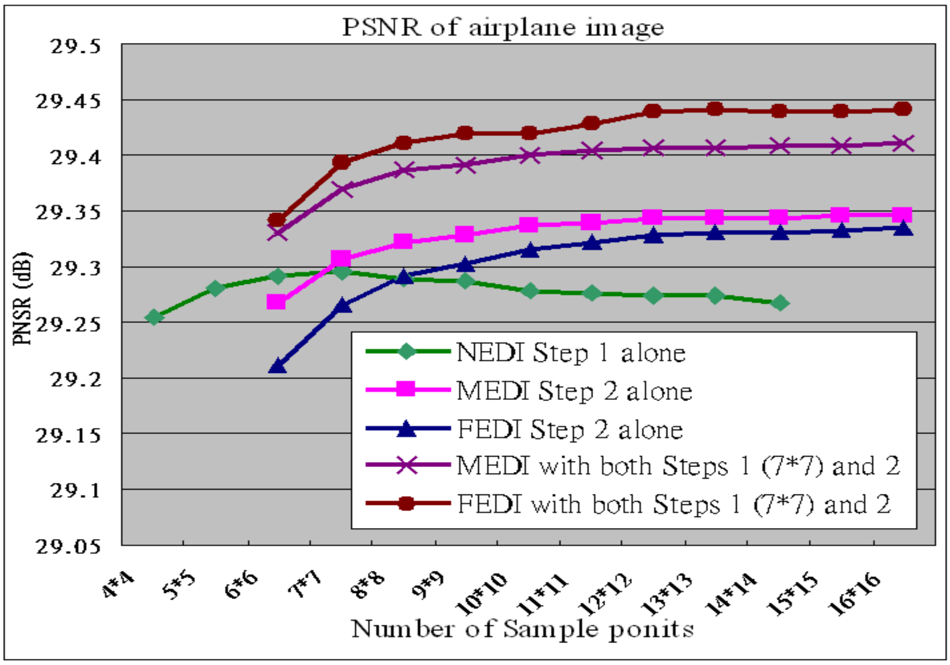


Figure 4.5: PSNR of airplane image by FEDI and MEDI

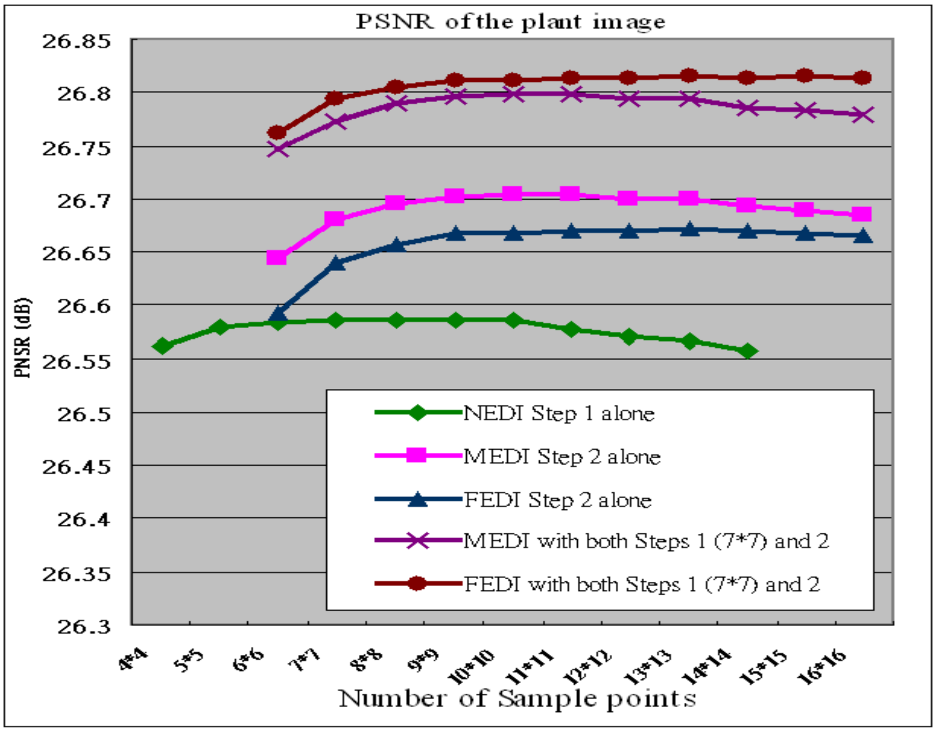


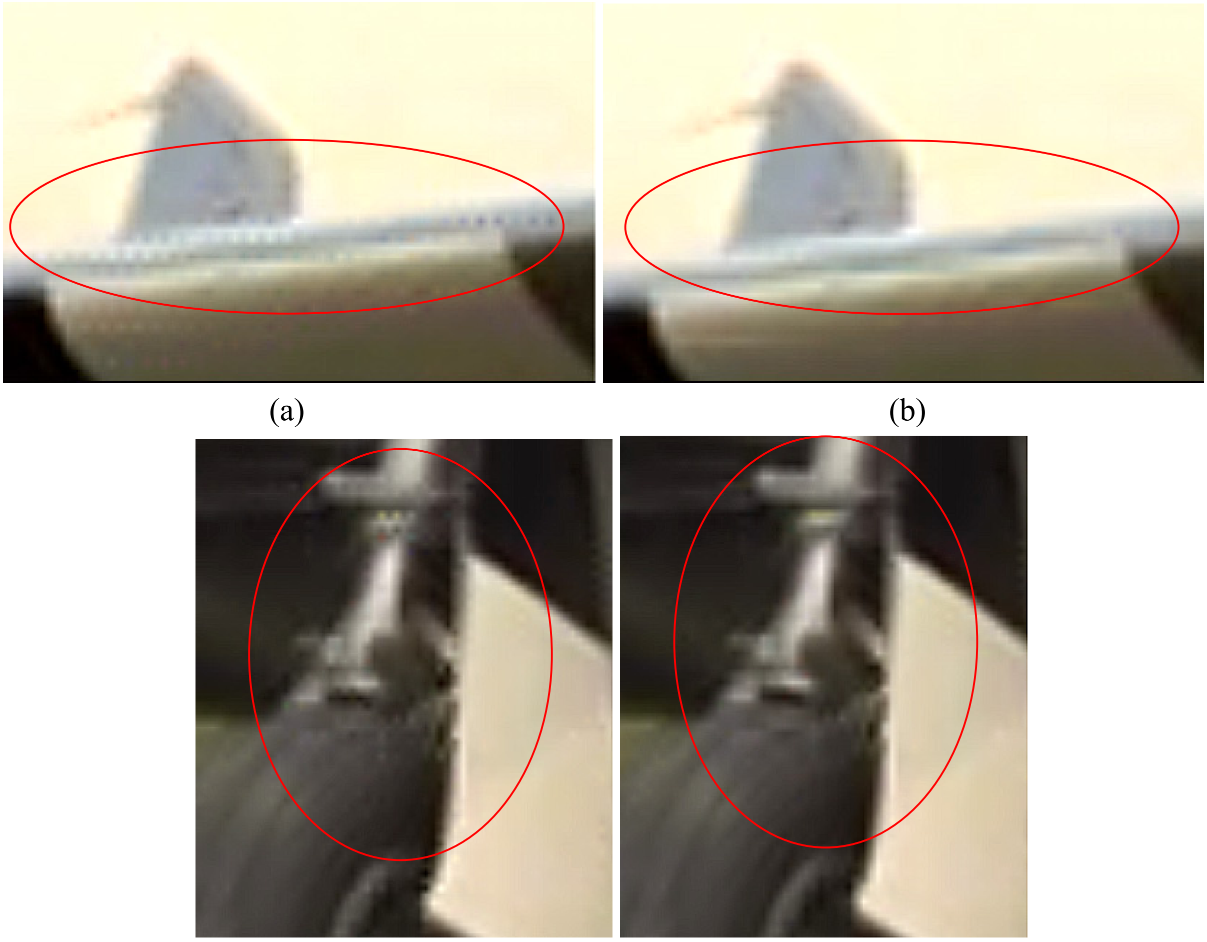
Figure 4.6: PSNR of plant image by FEDI and MEDI

For each of the figures, there are five PSNR curves. For the curve with “NEDI Step 1 alone” (green), it is the PSNR of the image after performing the NEDI Step 1 with different numbers of the sample points. Both “MEDI step 2 alone” (pink) and “FEDI Step 2 alone” (blue) curves are the PSNR values of the image after performing the interpolation step two only with different numbers of sample points. For the last two: “MEDI with both Steps 1 (7*7) and 2” (purple) and “FEDI with both Steps 1 (7*7) and 2” (red) curves, we interpolated both images by making use of the NEDI step 1 with sample points equal to 7×7, and varied the number of sample points in interpolation step two from 6×6 to 16×16.

In both figures, it is obvious that the PSNR of “FEDI step 2 only” (blue) is much lower than that of the “MEDI step 2 only” (pink), because the FEDI step two requires two predicted results from step one prediction. However, when we interpolated the image by MEDI and FEDI with both steps one and two, we can see that our FEDI results (red) in a higher PSNR than MEDI (purple) for using any number of sample points in interpolation step two. The comparison of the visual quality between FEDI and MEDI is shown in Fig. 4.7. We can see that our proposed FEDI can achieve a sharper image (see Fig. 4.7: b and d) with less artifacts (uneven pixels inside the red ellipses as shown in Fig. 4.7: a and c) than that of the MEDI.

In some rare cases, such as Fig. 4.8 the PSNR of FEDI is slightly lower than the PSNR of the NEDI, the visual quality of FEDI is still better than NEDI. In Fig. 4.8, we can see that using the FEDI for magnification, it results in a sharper image with more continuous edges than the NEDI.

Note from Figs. 4.5 and 4.6 that the PSNR values of the FEDI with both steps one and two increase initially and then decrease. It means that an increase in the number of sample points may not be able to improve the PSNR. Hence, there is an optimal number of the sample points for using the FEDI. According to the experiment results, we have found that the optimal number of the sample points in step one is between 6×6 and 8×8 and in step two is between 8×8 and 12×12. This result can also apply to the MEDI schemes.



(a) (b)

(c) (d)

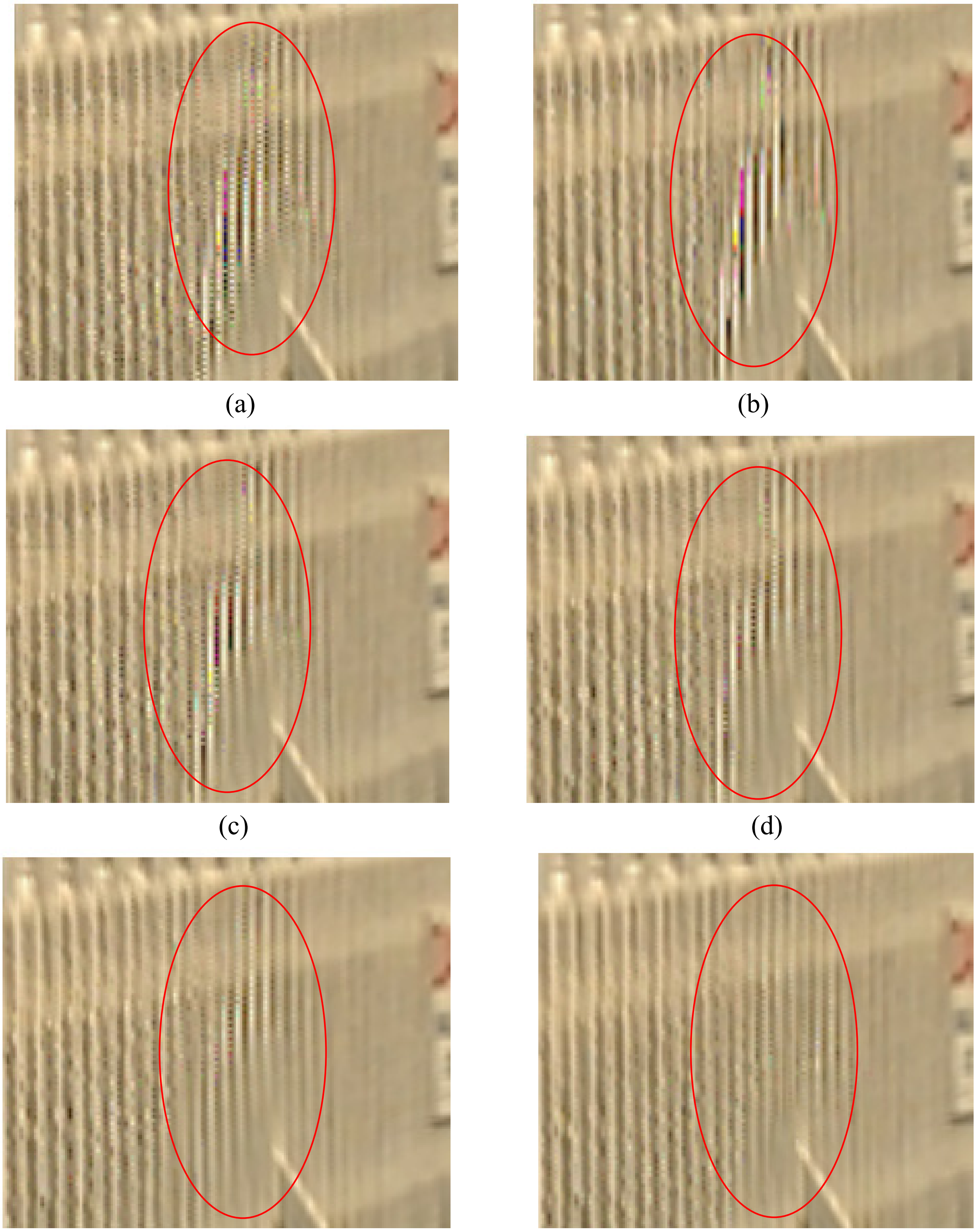
Figure 4.7: Portions of the airplane image: (a) and (c) reconstructed image by MEDI (PSNR=29.37 dB), (b) and (d) reconstructed image by FEDI (PSNR=29.39 dB).

(a) (b)

Figure 4.8: Portion of the lady image: (a) reconstructed image by NEDI (PSNR=30.15 dB), and (b) reconstructed image by FEDI (PSNR=30.11 dB).

Most of the interpolation schemes based on the edge-directed interpolation give a poor performance (unwanted colour and thick lines inside of the red ellipses in Fig. 4.9: a-d) on high frequency regions of an image. This is because when the image contains high frequency regions, the low-resolution covariance cannot estimate this high frequencies covariance structure accurately. However, we find that this mismatch problem can be resolved by increasing the number of sample points. We directly interpolated the lighthouse image by our FEDI from 512×768 to 1024×1536 with different numbers of sample points. Since we only want to compare the visual quality on the high frequency region, no downsampling is required and the PSNR is not the subject in this simulation. Some results are shown in Fig. 4.9. We can see that when the number of the sample points increased, the visual quality of the high frequency region is improving, by removing the undesirable prediction pixels as shown inside of the red ellipses in Fig. 4.9: e and f.

For enlarging a SD video to a HD video, the visual quality of Fast EDI-1.5 is better than the NEDI results which were to scale up the image by 2 times and than to downsample the results by a factor ¾, so that it has the same interpolation factor. From Figs. 4.10 and 4.11, we can see that our proposed Fast EDI-1.5 can interpolate a sharpen image than the NEDI. It is because the Fast EDI-1.5 does not need an addition down-sampling operation. Hence it can preserve the sharpness of the image. That is, the NEDI needs an addition step of downsample operations to perform the 1.5 time enlargement, which accounts for the smoothing effect on the interpolated image. In additional, the computational cost of the Fast EDI-1.5 is about 50% of NEDI (see Table 1).



(a) (b)

(c) (d)

(e) (f)

Figure 4.9: Portions of the lighthouse image, reconstructed by the FEDI with different numbers of sample points (a) using 7×7, (b) using 8×8, (c) using 9×9, (d) using 11×11, (e) using 13×13 and (f) using 16×16.

(a) (b)

Figure 4.10: Portions of the interpolated Pak Joy image from SD to HD: (a) reconstructed by NEDI and then downsampling by a factor ¾, and (b) reconstructed by Fast EDI-1.5 (ours).

(a) (b)

Figure 4.11: Portions of the interpolated InTo Tree image from SD to HD: (a) reconstructed by NEDI and then downsampling by a factor ¾, and (b) reconstructed by Fast EDI-1.5 (ours).

Table 4.1. Interpolation time of SD image to HD image

Name Of The Image	**NEDI**	**Fast EDI-1.5**	**Time reduce of Fast EDI-1.5 Scheme Compared To NEDI**
Ducks Take Off	14.5s	6.8s	-53%
Pak Joy	8.1s	3.7s	-54%
Crowd	12.2s	5.7s	-53%
InTo Tree	10.4s	5.4s	-48%
Old Town Cross	9.8s	4.9s	-49%

4.5 Summary

We have proposed the Further Improved Edge-directed Interpolation scheme (FEDI) and applied the EDI concept to scales up an image to 1.5 times (Fast EDI-1.5) in this paper. Both FEDI and Fast EDI-1.5 can be used in the interpolation of a SDTV video to a HDTV video. The proposed FEDI scheme has a better visual quality compared with that of the MEDI scheme, and has a sharper edge as compared to the NEDI scheme, but it requires a high computational cost. Hence, we have developed a fast scheme, Fast EDI-1.5, which has a better visual quality compared with the NEDI and with less computational cost. We also have suggested a possible optimal number of sample points and given the effect on the number of sample points for high frequency regions. Much further work can be done on improving the Fast EDI-1.5 approach by making use of the FEDI scheme and making further improvement by using some multi-frame super-resolution techniques.

Chapter 5

Adaptive Directional Window Selection (ADWS)

5.1 Introduction

Recall that a Modified Edge-directed interpolation MEDI [47] was proposed in 2009, which suggested using a multiple square training windows instead of a single square training window. However, we have found that using a multiple square windows is good but it does not always get the optimal result especially in the fine edge and texture regions. Therefore, we propose in this chapter an Adaptive Directional Window Selection to overcome the exiting problem in the Edge-Directed Interpolation (EDI).

5.2 Adaptive Directional Window Selection

The original window selection method in NEDI algorithm is simply and it just uses a square window, with its center at the unknown pixel that we want to predict as shown in Fig. 3.6. The assumption of using a square window is that the statistics of this square window in the low-resolution domain should be the same as the statistics in high-resolution domain. However, in many cases this assumption is not true. Therefore, the NEDI always introduces directional artifacts, as the statistics of the data in the square window cannot truly reflect the real situation.

In order to solve this problem in the NEDI, we develop an Adaptive Directional Window Selection (ADWS) technique. The ADWS makes use of the practical directional elliptic window which works according to the edge direction sliding along an edge and then subsequently chooses the best window evaluated by choosing the elliptic window which has the lowest Means Square Error (MSE).

Our proposed ADWS method includes three steps. First, Edge Direction Detection is applied to the image. Second, the Adaptive Directional Elliptic Window is then also applied according to the edge Direction. Finally, in the Window Selection step the best window is selected for forming.

5.2.1 Edge Direction Detection

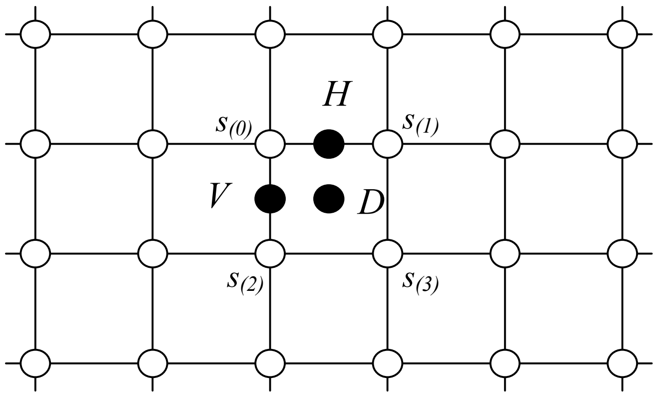


Figure 5.1: The relationship between original pixels (○) and the pixels to be interpolated (●) with interpolation factor equal to 2.

As shown in Fig. 5.1, there are three pixels to be interpolated (*D*, *H* and *V*) with interpolation factor equal to 2. Let us classify the edge direction into nine categories (including a small-edge type and eight edge types with directions: 0°, 22.5°, 45°, 67.5°, 90°, 112.5°, 135° and 157.5°) in the diagonal (*D*), horizontal (*H*) and vertical (*V*) edge detections. We modified the edge detection method in [36] to reduce the computation cost.

In this study, the eight filtering masks for diagonal edge detection of eight directions are as follows:

$$D_{0^\circ}=\begin{bmatrix}0&0&0&0&0\\0&0&-1&-1&0\\0&-1&-1&-1&-1\\0&1&1&1&1\\0&0&1&1&0\end{bmatrix},\ D_{90^\circ}=\begin{bmatrix}0&0&0&0&0\\0&0&-1&1&0\\0&-1&-1&1&1\\0&-1&-1&1&1\\0&0&-1&1&0\end{bmatrix},$$
$$D_{22.5^\circ}=\begin{bmatrix}0&0&0&0&0\\0&-1&-1&-1&-1\\0&-1&-1&0&0\\0&0&0&1&1\\0&1&1&1&1\end{bmatrix},\ D_{157.5^\circ}=\begin{bmatrix}0&0&0&0&0\\0&1&1&1&1\\0&0&0&1&1\\0&-1&-1&0&0\\0&-1&-1&-1&-1\end{bmatrix},$$
$$D_{45^\circ}=\begin{bmatrix}0&0&0&0&0\\0&0&-1&-1&0\\0&-1&-1&0&1\\0&-1&0&1&1\\0&0&1&1&0\end{bmatrix},\ D_{135^\circ}=\begin{bmatrix}0&0&0&0&0\\0&0&1&1&0\\0&-1&0&1&1\\0&-1&-1&0&1\\0&0&-1&-1&0\end{bmatrix},$$
$$D_{67.5^\circ}=\begin{bmatrix}0&0&0&0&0\\0&-1&-1&0&1\\0&-1&-1&0&1\\0&-1&0&1&1\\0&-1&0&1&1\end{bmatrix},\ D_{112.5^\circ}=\begin{bmatrix}0&0&0&0&0\\0&-1&0&1&1\\0&-1&0&1&1\\0&-1&-1&0&1\\0&-1&-1&0&1\end{bmatrix}. \tag{5.1}$$

The eight filtering masks for horizontal edge detection of eight directions are as follows:

$$H_{0^\circ}=\begin{bmatrix}0&0&-1&-1&0\\0&-1&-1&-1&-1\\0&0&0&0&0\\0&1&1&1&1\\0&0&1&1&0\end{bmatrix},\ H_{90^\circ}=\begin{bmatrix}0&0&-1&1&0\\0&-1&-1&1&1\\0&-1&-1&1&1\\0&-1&-1&1&1\\0&0&-1&1&0\end{bmatrix},$$
$$H_{22.5^\circ}=\begin{bmatrix}0&0&-1&-1&0\\0&-1&-1&-1&0\\0&-1&0&0&1\\0&0&1&1&1\\0&0&1&1&0\end{bmatrix},\ H_{157.5^\circ}=\begin{bmatrix}0&0&1&1&0\\0&0&1&1&1\\0&-1&0&0&1\\0&-1&-1&-1&0\\0&0&-1&-1&0\end{bmatrix},$$
$$H_{45^\circ}=\begin{bmatrix}0&0&-1&-1&0\\0&-1&-1&-1&1\\0&-1&-1&1&1\\0&-1&1&1&1\\0&0&1&1&0\end{bmatrix},\ H_{135^\circ}=\begin{bmatrix}0&0&1&1&0\\0&-1&1&1&1\\0&-1&-1&1&1\\0&-1&-1&-1&1\\0&0&-1&-1&0\end{bmatrix},$$
$$H_{67.5^\circ}=\begin{bmatrix}0&0&-1&-1&0\\0&-1&-1&0&1\\0&-1&-1&1&1\\0&-1&0&1&1\\0&0&1&1&0\end{bmatrix},\ H_{112.5^\circ}=\begin{bmatrix}0&0&1&1&0\\0&-1&0&1&1\\0&-1&-1&1&1\\0&-1&-1&0&1\\0&0&-1&-1&0\end{bmatrix} \tag{5.2}$$

The eight filtering masks for vertical edge detection of eight directions are as follows:

$$V_{0^\circ}=\begin{bmatrix}0&0&0&0&0\\0&-1&-1&-1&0\\-1&-1&-1&-1&-1\\1&1&1&1&1\\0&1&1&1&0\end{bmatrix},\ V_{90^\circ}=\begin{bmatrix}0&0&0&0&0\\0&-1&0&1&0\\-1&-1&0&1&1\\-1&-1&0&1&1\\0&-1&0&1&0\end{bmatrix},$$
$$V_{22.5^\circ}=\begin{bmatrix}0&0&0&0&0\\0&-1&-1&-1&0\\-1&-1&-1&0&1\\-1&0&1&1&1\\0&1&1&1&0\end{bmatrix},\ V_{157.5^\circ}=\begin{bmatrix}0&0&0&0&0\\0&1&1&1&0\\-1&0&1&1&1\\-1&-1&-1&0&1\\0&-1&-1&-1&0\end{bmatrix},\tag{5.3}$$
$$V_{45^\circ}=\begin{bmatrix}0&0&0&0&0\\0&-1&-1&-1&0\\-1&-1&-1&1&1\\-1&-1&1&1&1\\0&1&1&1&0\end{bmatrix},\ V_{135^\circ}=\begin{bmatrix}0&0&0&0&0\\0&1&1&1&0\\-1&-1&1&1&1\\-1&-1&-1&1&1\\0&-1&-1&-1&0\end{bmatrix},$$
$$V_{67.5^\circ}=\begin{bmatrix}0&0&0&0&0\\0&-1&-1&0&0\\-1&-1&0&1&1\\-1&-1&0&1&1\\0&0&1&1&0\end{bmatrix},\ V_{112.5^\circ}=\begin{bmatrix}0&0&0&0&0\\0&0&1&1&0\\-1&-1&0&1&1\\-1&-1&0&1&1\\0&-1&-1&0&0\end{bmatrix}.$$

For either diagonal, horizontal or vertical edge detection, we use element-by-element multiplication between the 5×5 image pattern in low-resolution image and the eight 5×5 filtering masks. After summing up the 25 products, it will produce eight filtering outputs, which will be normalized by the sum of the 25 corresponding filtering weights with positive values only. The edge type of each 5×5 image pattern is determined as the one having the maximum "normalized" filtering output. However, if the difference between the maximum "normalized" filtering output and the minimum "normalized" filtering output of an image pattern is smaller than a threshold T_e, the image pattern is then its classified as a small-edge one. We only perform the ADWS method, when the variance of its four neighbours (s) as shown in Fig. 3 of the unknown pixel is larger than a threshold T_s. The objective of testing the variance is to find out the edge region for performing the ADWS method. If the variance is smaller than T_s, bilinear interpolation is used instead of the ADWS in order to save computation. However, if the "normalized" filtering output difference is smaller than T_e, it means that this region does not have a major edge direction, therefore, we define the edge type of this region as the small edge type which means that this region contains some small edges with multiple directions.

5.2.2 Adaptive Directional Ellipse Windows

According to the edge direction, let us define its elliptic window. If the edge type is small-edge, a circular window with radius 5 is used. The equation of the elliptic window is shown below:

$$E_{\theta}:\frac{(x\cos(\theta)-y\sin(\theta))^2}{a^2}+\frac{(x\sin(\theta)-y\cos(\theta))^2}{b^2}=1 \tag{5.4}$$

where θ = 0°, 22.5°, 45°, 67.5°, 90°, 112.5°, 135°, 157.5°

In order to get a sufficient number of the sample points for the Wiener filter, we choosing a=3 and b=7 in Eq. (5.4). Therefore, all the eight directional ellipses will be inside the 14×14 square window, which are the sample points (the gray region) in a particular direction as shown in Fig 5.2.

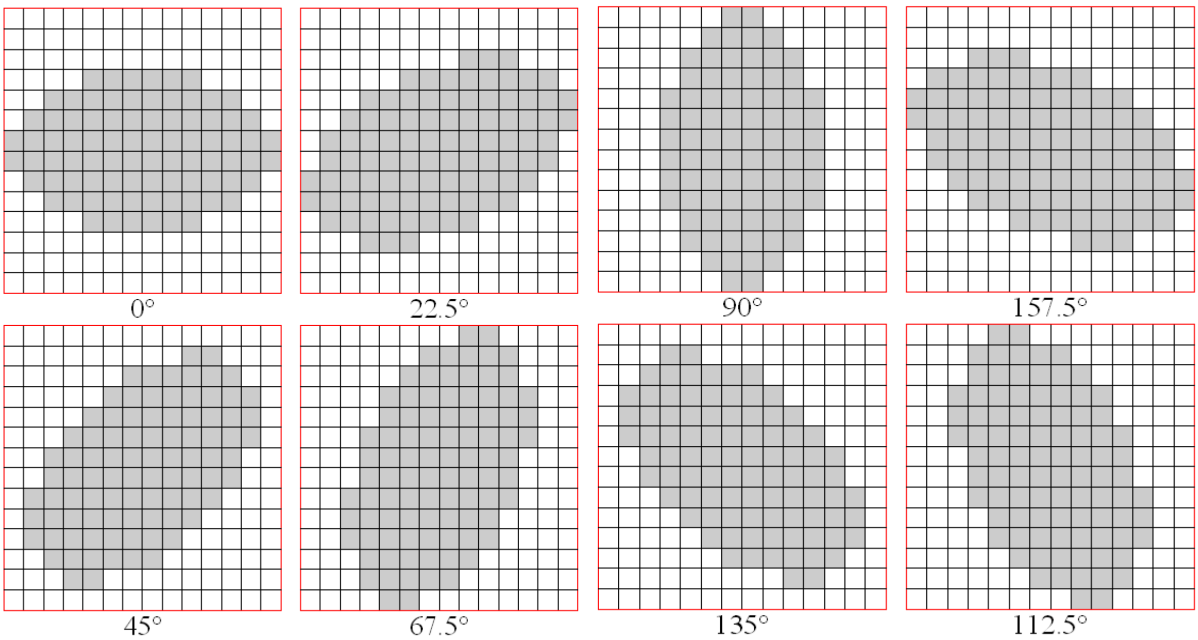


Figure 5.2: Eight directional elliptic windows

In order to reduce the covariance miss-match problem in the edge-directed interpolation method, we further propose to use multiple Directional Elliptic Windows according to the edge direction. Fig. 5.3 (a) shows a high-resolution block which is being to be interpolated. As the direction of the edge (red line) is 45°, the 45° Elliptic Window is used. The Elliptic Window is arranged to slide along in the edge direction (45°) starting at the middle as shown in Fig. 5.3 (b), move up and right by two pixels as shown in Fig. 5.3 (c) and move down and left by two pixels as shown in Fig. 5.3 (d). Only the Elliptic Window that contains most edge information i.e. Fig. 5.3 (d) will be chosen to be the best candidate for this high-resolution block.

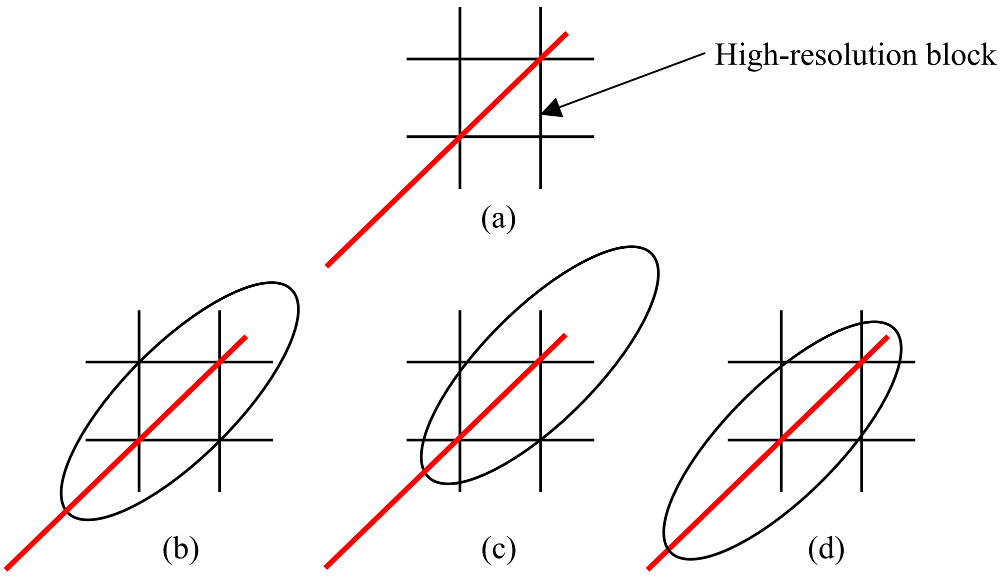


Figure 5.3: Illustration of three Directional Elliptic Windows (b-d) for high-resolution block in (a) where the direction of the edge is 45° (Red).

5.2.3 Window Selection.

In the window selection process for the EDI, we select the elliptic window which has the lowest error energy. The cost function of the error energy can be derived from the N^{th}-order linear prediction equation.

Assume that we use an N^{th}-order linear prediction equation to estimate the unknown pixel $Y(k)$ which is denoted as $Y'(k)$ by using the neighboring pixels, $X(0)$, $X(1)$…, $X(N\text{-}1)$.

$$Y`(k) = \sum_{n=0}^{N-1} \alpha_n X(n) \tag{5.5}$$

In the matrix form, we can write Eq. (5.5) as below,

$$Y`(k) = \boldsymbol{\alpha}^{\mathrm{T}} \mathbf{x} \tag{5.6}$$

Where vector $\boldsymbol{\alpha} = [\alpha_{0,}\, \alpha_{1, \ldots,}\, \alpha_{N-1}]^{\mathrm{T}}$ and the vector $\mathbf{x}=[X(0), X(1), \ldots, X(N\text{-}1)]^{\mathrm{T}}$.

Then, we can derive the equation of the cost function between the $Y(k)$ and $Y`(k)$ by using the Means Square Error (MSE).

$$\mathrm{J}_{\text{error}} = \mathrm{E}[(Y(k) - Y`(k))^2] \tag{5.7}$$

Substitute Eq. (3.20) and Eq. (5.6) into Eq. (5.7), we have

$$\begin{aligned}
\mathrm{J}_{\text{error}} &= \mathrm{E}[(Y(k) - Y`(k))^2] \\
&= \mathrm{E}[(Y(k) - (\mathbf{R}_{yy}{}^{-1}\mathbf{r}_y)^{\mathrm{T}}\mathbf{x})^2] \\
&= \mathrm{E}[(Y(k)^2] - 2(\mathbf{R}_{yy}{}^{-1}\mathbf{r}_y)^{\mathrm{T}}\mathrm{E}[(Y(k)\mathbf{x})] + \\
&\quad (\mathbf{R}_{yy}{}^{-1}\mathbf{r}_y)^{\mathrm{T}}\mathrm{E}[\mathbf{x}\mathbf{x}^{\mathrm{T}}](\mathbf{R}_{yy}{}^{-1}\mathbf{r}_y) \\
&= \mathrm{E}[(Y(k)^2] - 2(\mathbf{R}_{yy}{}^{-1}\mathbf{r}_y)^{\mathrm{T}}\mathbf{r}_y + (\mathbf{R}_{yy}{}^{-1}\mathbf{r}_y)^{\mathrm{T}}\mathbf{R}_{yy}(\mathbf{R}_{yy}{}^{-1}\mathbf{r}_y) \\
&= \mathrm{E}[(Y(k)^2] - 2(\mathbf{R}_{yy}{}^{-1}\mathbf{r}_y)^{\mathrm{T}}\mathbf{r}_y + (\mathbf{R}_{yy}{}^{-1}\mathbf{r}_y)^{\mathrm{T}}\mathbf{r}_y \\
&= \mathrm{E}[(Y(k)^2] - \mathbf{r}_y{}^{\mathrm{T}}(\mathbf{R}_{yy}{}^{-1}\mathbf{r}_y)
\end{aligned}$$

where $\mathrm{E}[Y(k)\mathbf{x}]$ is the high-resolution cross-covariance $\mathbf{r}_{\mathrm{y}}$ and $\mathrm{E}[\mathbf{x}\mathbf{x}^{\mathrm{T}}]$ is the high-resolution auto-covariance $\mathbf{R}_{yy}$.

$Y(k)$ in the E[$Y(k)^2$] and the $Y(k)$ in the high-resolution cross-covariance $\mathbf{r}_y$ (E[$Y(k)\mathbf{x}$]) are not available now, because $Y(k)$ is to be predicted. To solve this problem, we use the geometric duality between the high-resolution covariance and low-resolution covariance. Therefore, the low-resolution covariance $\mathbf{R}`_{yy}$, $\mathbf{r}`_y$ and $\mathbf{y}$ will be use instead, for the calculation. The final equation for the cost function is shown below.

$$\begin{aligned} \mathrm{J}_{\text{error}} &= \mathbf{y}^{\mathrm{T}}\mathbf{y} - \mathbf{r}`_y{}^{\mathrm{T}}(\mathbf{R}`_{yy}{}^{-1}\mathbf{r}`_y) \\ &= \mathbf{y}^{\mathrm{T}}\mathbf{y} - \mathbf{r}`_y{}^{\mathrm{T}}\boldsymbol{\alpha} \end{aligned} \tag{5.8}$$

Using the Eq. (5.8), we can find out the cost of each of the elliptic window, and then we can choose the elliptic window that has the lowest cost value which means it has the lowest error energy.

Note that the Eq. (5.8) have been derived without considering the optimization using derivative that is usually used in the formulating the Wiener filter.

5.3 Experimental Result

The proposed algorithm was implemented and compared with several conventional approaches in literature including Bilinear interpolation, the NEDI method [2] and the MEDI method [47]. All the programs were write in C++ language and run on the same platform. The values of the thresholds T_s and T_e using in the proposed ADWS are 8 and 10 respectively. In the comparison, our proposed enhancement technique, ADWS, has been applied based on the MEDI method.

In order to compare the performance of these algorithms in terms of objective image quality, a low-resolution image was obtained by down sampling a high-resolution image by a factor of 2 and then we reconstructed it by conventional methods and our algorithm. We compare the interpolated high-resolution image with the original high-resolution image in terms of PSNR and visual quality. We have used various test images and the PSNR of all test images are summarized in Table 5.1.

In Table 5.1, when our proposed Adaptive Directional Window Selection method was applied to the MEDI, the PSNR is slightly increased.

Although, the proposed ADWS method does not have a large PSNR improvement, it results in a better objective quality than the NEDI and MEDI.

In both Fig. 5.4 and Fig. 5.5, the interpolated images (d) using bilinear interpolation contains many aliasing artifacts so it has the worst visual quality. The interpolated images (e) using NEDI resulted in a more continuous edge image, but it is still affected by the ringing effect which shows inside of the red ellipse. The interpolated images (f) using MEDI contain a lot of ringing effects as shown inside of the red ellipse. We can see that the interpolated images (g) using our proposed method ADWS in MEDI can reduce most of the ringing effect in the MEDI and NEDI methods.

Table 5.1: The PSNR (dB) of interpolated color (RGB) test images by Bilinear, NEDI, MEDI and proposed ADWS which apply in MEDI interpolation methods.

Image	Bilinear	NEDI	MEDI	MEDI (ADWS)
kodim01	23.75	23.63	23.60	23.60
kodim02	30.70	30.71	30.64	30.65
kodim03	31.42	31.69	31.61	31.61
kodim04	30.48	30.47	30.35	30.37
kodim05	23.78	23.98	23.94	23.96
kodim08	21.09	20.96	20.95	20.95
kodim09	29.00	29.03	29.04	29.05
kodim10	28.96	28.92	28.92	28.90
kodim11	26.82	26.80	26.78	26.80
kodim13	21.99	21.90	21.79	21.77
kodim14	26.46	26.44	26.43	26.44
kodim16	29.16	29.10	29.03	29.02
kodim17	29.50	29.53	29.47	29.48
kodim18	25.72	25.69	25.64	25.62
kodim20	28.62	28.71	28.71	28.72
kodim21	26.05	25.93	25.89	25.89
kodim23	31.26	31.51	31.37	31.37
Average	27.34	27.35	27.30	27.31

Moreover, in Fig. 5.6, the interpolated image (d) using the bilinear interpolation cannot perform well in the texture areas. The interpolated image (e) using the NEDI resulted in a more accurate edge prediction than the bilinear interpolation, but it is still affected by the ringing effect as shown inside of the red ellipses. This also causes the discontinuous of the edge region. The interpolated image (f) using MEDI contains a lot of ringing effects as shown inside of the red ellipses. We can see that the interpolated image (g) using our proposed method ADWS in MEDI can reduce most of the ringing effect in the MEDI and NEDI method and keep the continuous of the edge region.

Figs. 5.4 (g), 5.5 (g) and 5.6 (g) also show that our proposed method (ADWS) can produce an interpolated image which have sharper and more continuous edges than other interpolation methods (see blue ellipses in Figs. 5.4, 5.5 and 5.6). It is because using the edge direction information together with the Adaptive directional elliptic window can reflect the low-resolution statistic more accurately than the normal non-directional square window in MEDI and NEDI. Although the MEDI method makes use of the multi-square windows to fit the edge orientation, the selection of the square windows in MEDI is not accurate enough. This cannot select the best window which contains most of the edge information. Therefore, in some texture regions, the performance of MEDI is worse than that of NEDI.

Another advantage of the proposed method (ADWS) is the edge performance. Areas near to the end of the edge or near to the texture region with multi-direction edges are improving. In Fig. 5.7, there are many ringing effects that are near to the main edge, as shown inside of the red ellipses. It is because the square window is non-directional and it will always gives wrong statistical information, so that the Wiener filter cannot accurately predict the true direction of the edge, so ringing effect happens. In Fig. 5.7, we can see that the direction of the ringing effect inside of the red ellipses is about 135° which is the direction of the metal rod in the center of the red ellipses, but the true direction of the edge is about 45°. This mismatch problem is caused by the inaccuracy of statistical information in the square window using in the NEDI and MEDI. After applying the proposed ADWS in MEDI, the ringing effect is removed, which gives the best visual quality of the interpolated image as shown in Fig. 5.7 (e).

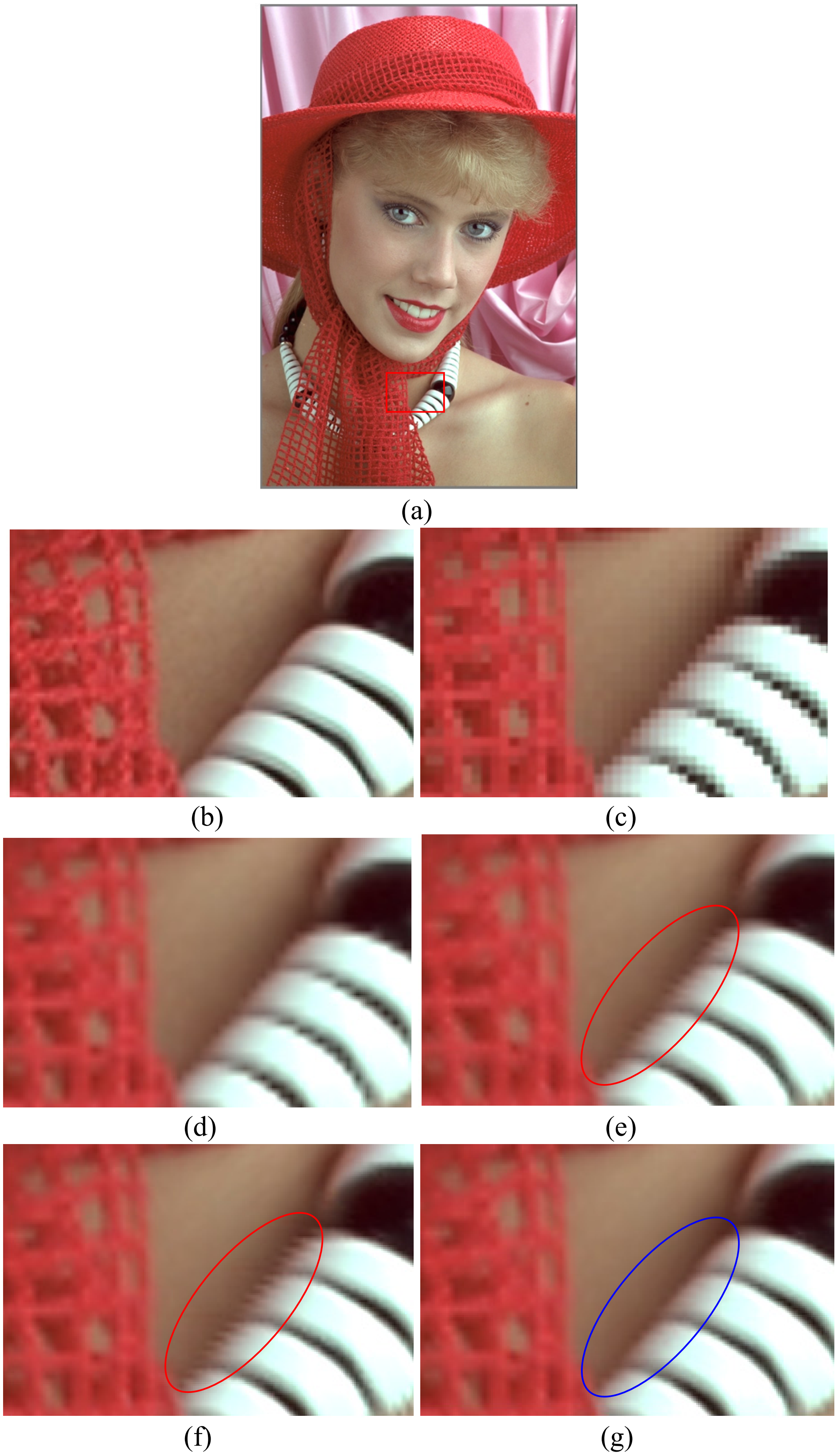


Figure 5.4: (a) Original test image “kodim04” and zoomed-in portion of the: (b) original image, (c) down sampled image, (d) reconstructed by Bilinear, (e) reconstructed by NEDI, (f) reconstructed by MEDI and (g) reconstructed using ADWS in MEDI.

Figure 5.5: (a) Original test image “kodim14” and zoomed-in portion of the: (b) original image, (c) down sampled image, (d) reconstructed by Bilinear, (e) reconstructed by NEDI, (f) reconstructed by MEDI and (g) reconstructed using ADWS in MEDI.

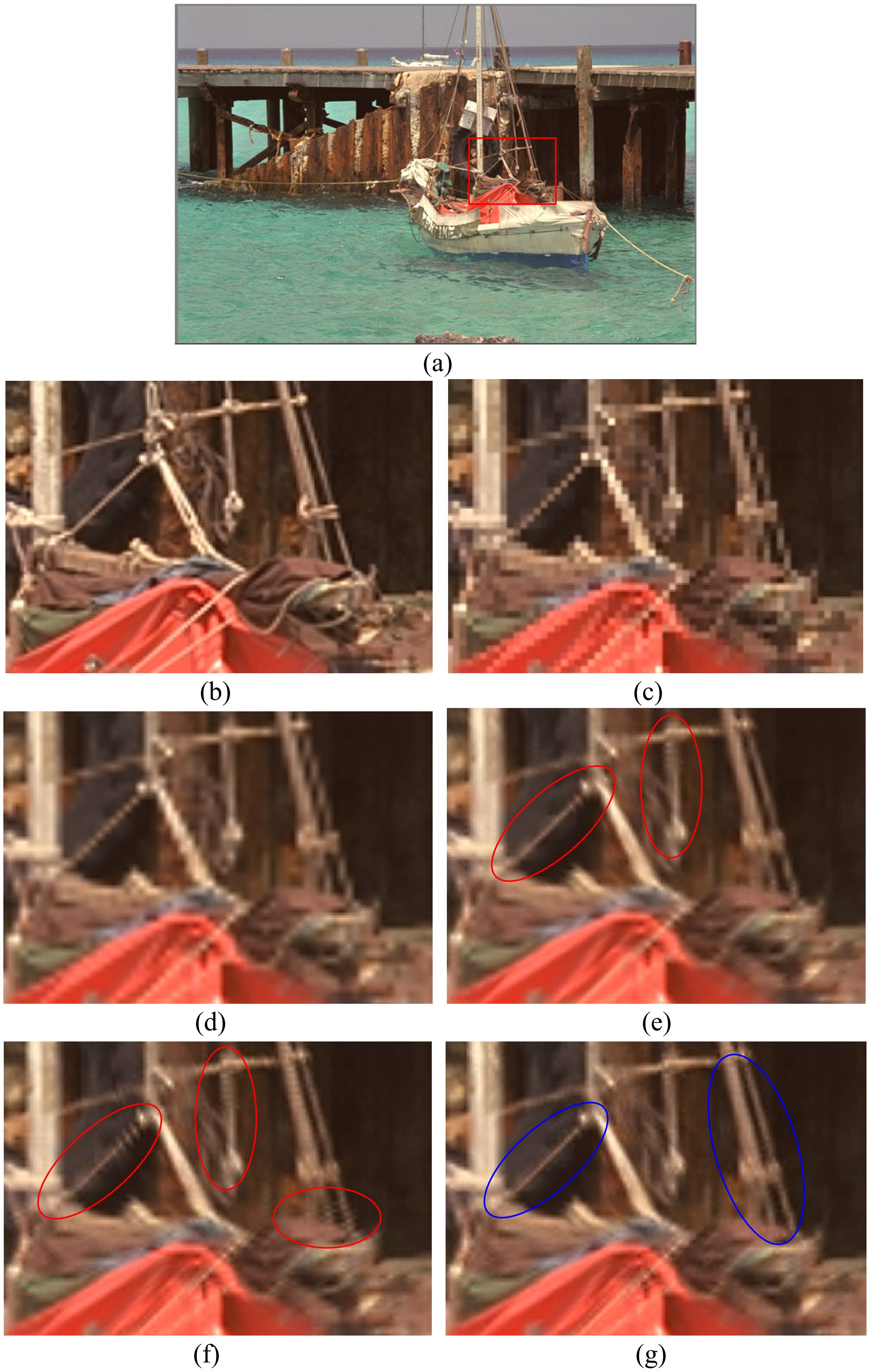
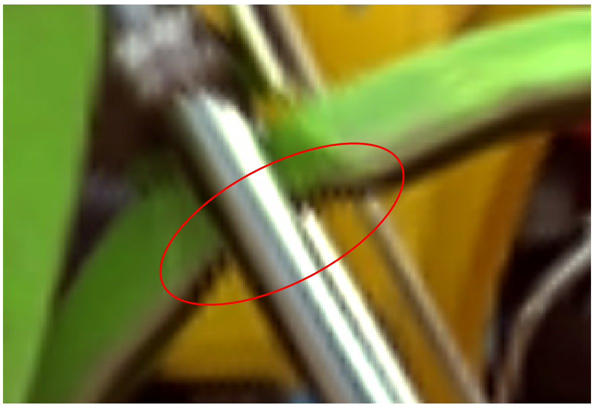


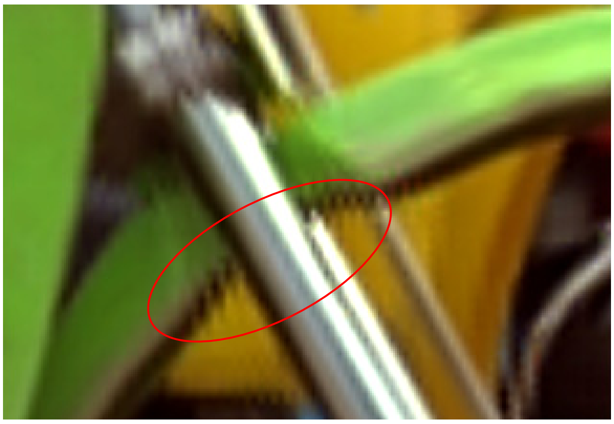
Figure 5.6: (a) Original test image “kodim11” and zoomed-in portion of the: (b) original image, (c) down sampled image, (d) reconstructed by Bilinear, (e) reconstructed by NEDI, (f) reconstructed by MEDI and (g) reconstructed using ADWS in MEDI.

(a)

(b)



(c)



(d)

(e)

Figure 5.7: (a) Original test image "kodim05" and zoomed-in portion of the: (b) original image, (c) reconstructed by NEDI, (d) reconstructed by MEDI and (e) reconstructed using ADWS in MEDI.

5.4 Summary

In this paper, we have proposed an adaptive directional window selection (ADWS) for the EDI. It makes use of the edge information to find the practical directional elliptic window which is suitable for edges. We arrange the elliptic window to slide along on the edge direction and then we select the best elliptic window as the sample points for the optimization in the Wiener filter. Experimental results show that the proposed ADWS can reduce most of the ringing effect and artifact in the interpolated image compared to that of the NEDI and MEDI. It can overcome the existing problem of the NEDI in high frequency content or texture region by considering multiple directional elliptic windows. Moreover, the window selection method in ADWS is accurate, so that the prediction error in the interpolated image is reduced compared to the NEDI and MEDI. As a result, the proposed algorithm produces interpolated images with better objective and subjective qualities compared to that of the conventional interpolation methods.

Chapter 6

Pre-Processing of the Sample Data and Fast Large Up-Scaling technique for EDI Interpolation

6.1 Introduction

The pre-processing step for the sample data can make the interpolation more robust to the fine edge region and noise image. This pre-processing step can be combined with the ADWS that discuss in the previous chapter. Experimental results show that by combining the Pre-Processing techniques with the proposed ADWS method, we can generate a high quality interpolated image which is better than other edge directed interpolation approaches. Experimental results are also provided using different images to justify the value of this approach at the end of the chapter.

In the NEDI method, it uses the low-resolution (original) image as the sample data for the calculation of the low-resolution auto-covariance and cross-covariance in Eq. (3.21). However, the low-resolution image usually contains the abrupt changes of the pixel intensities which will cause the prediction error during the Least Mean Square (LMS) estimation in the NEDI method.

The idea of this step is simply using a high order low-pass filter to smooth the sample data so as to reduce the abrupt change of the pixel intensities. It can be done by using two different kinds of the resolution images, the original-resolution and high-resolution image which will be discussed in the following sections.

After the pre-processing step, the filtered sample data will replace the original sample data as the input data for the calculation of the Wiener filter coefficients of α in Eq. (3.22).

This step can make the interpolation more robust to fine edge region, texture region and the noise region in the image. Also it can reduce the artifacts that often occur in the high frequency region.

Nowadays, the resolution of the LCD display or TV is increasing rapidly. Therefore we need to have a large up scaling interpolation for the video or image to display correctly in these high-resolution display systems. However, the traditionally EDI interpolation need a high computation cost to perform the large up scaling interpolation in the video or image. If we want to up scaling the video size to 4 times of its original size we need to perform twice EDI interpolation. If the first EDI interpolation need 1s for the up scaling process, then the second EDI interpolation will be need about 4s for the up scaling process as the input number of the pixels in the image is fourfold of the original image. Therefore, EDI interpolation need a large computation cost to perform a large up scaling. As a result, the fast technique for the large upscaling of the EDI interpolation is proposed.

6.2 Original-resolution filtered sample data (ORSD)

Using the original-resolution filtered sample data means that we make use of the low-pass filter to filter the original low-resolution image and use the filtered sample data as the input data for the LMS estimation used.

According to the filtering direction, the low-pass filtering can be divided into two categories, diagonal direction filtering, horizontal and vertical directions filtering. According to the difference sampling pattern structure in the interpolation step one (interpolating the D pixels in Fig. 5.1) and step two (interpolating the V and H pixels in Fig. 5.1), we need to apply low-pass filter with difference direction in order to get the optimal prediction result. After we analyse the sampling pattern structure of NEDI [2] and MEDI [47], we apply the high order low-pass filter in diagonal direction to the low-resolution image for the step one interpolation and apply the high order low-pass filer in horizontal and vertical directions to the low-resolution image for the step two interpolation. This ORSD method is similar to the method in [44], except the filtering direction for step one and two. We then can calculate the optimal coefficients set α by using these filtered sample data. The block diagram of the process is shown in Fig. 6.1.

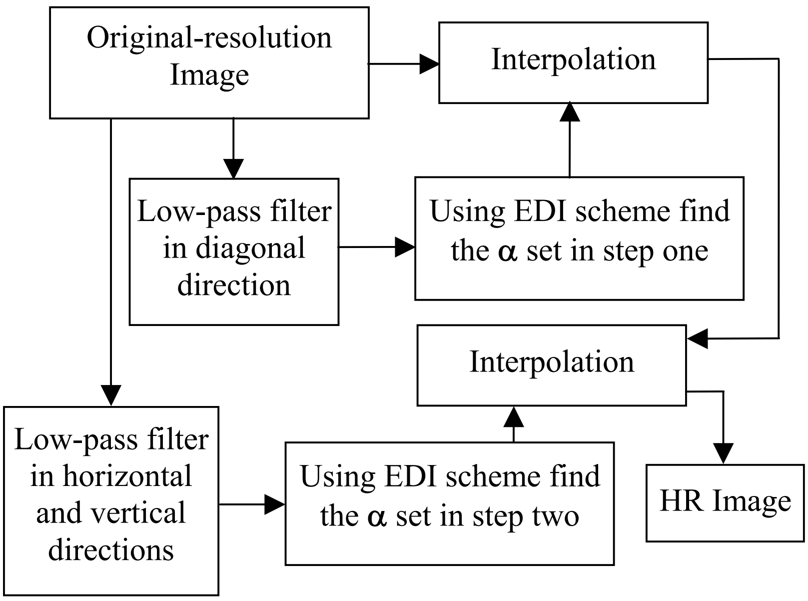


Figure 6.1: Flow diagram of EDI using the original-resolution filtered sample data.

6.3 High-resolution filtered sample data (HRSD).

In the use of the high-resolution filtered sample data, we firstly apply the high order low-pass filter in diagonal directions to the low-resolution image for step one and apply the high order low-pass filer in horizontal and vertical directions to the low-resolution image for step two. After the filtering, we have two difference kinds of the low-resolution filtered images for step one and two. In order to enlarge the fine edge regions in the two filtered images, we use some basic linear interpolation method (i.e. Bicubic Interpolation) to interpolate them from the filtered low-resolution image to the filtered high-resolution image and use those filtered high-resolution image pixels value as the filtered sample data for the calculation of the optimal coefficients set α in Eq. (3.22). The block diagram of the process is shown in Fig. 6.2.

The difference between ORSD and HRSD is that, HRSD is more suitable for an image which contains a texture region with a high frequency content or very fine edge region. It is because using filtered and interpolated sample data can enlarge the fine edge region, so that the EDI can predict the unknown pixels correctly. In Fig. 6.3 (a), it shows an original-resolution image block which contains many fine edges. When we apply the elliptic window (red ellipse) in this image block, the elliptic window will include not only the major edge information but also some other edge information. Therefore, the prediction is not accurate. The pre-processed high-resolution sample data was proposed to solve this problem as shown in Fig. 6.3 (b). In Fig. 6.3 (b), when the resolution of the sample data is double, the elliptic window can be more concentrate on the major edge. It means that the edge information and the statistical information inside of the elliptic window can be reflecting the real situation of the edge direction. Therefore, the prediction is more accurate.

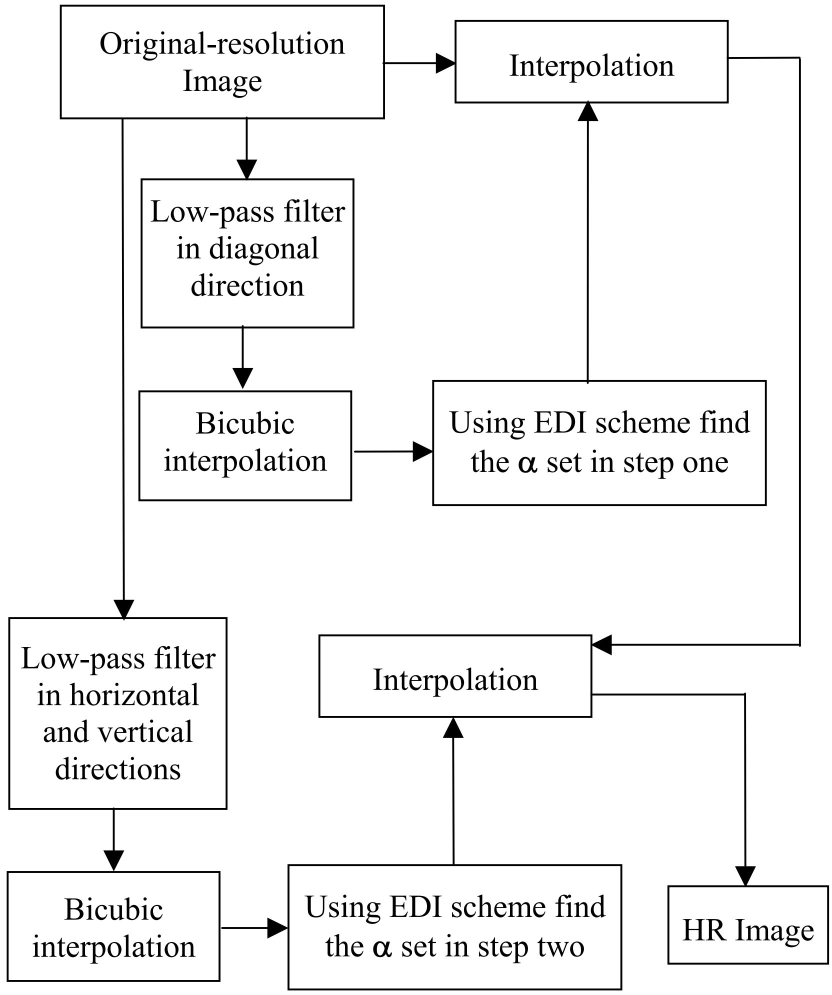
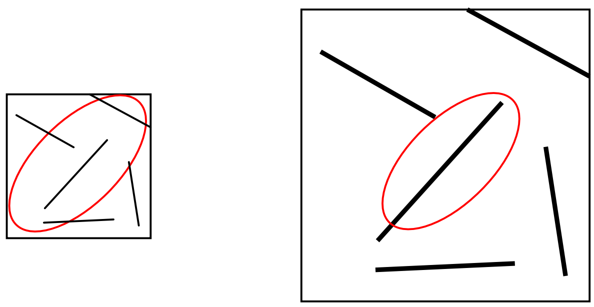


Figure 6.2: Flowing diagram of EDI using the high-resolution filtered sample data.



(a) (b)

Figure 6.3: The elliptic window using the (a) ORSD and (b) HRSD.

6.4 Filtered sample data in Fast EDI-1.5

In chapter 4, we have proposed the Fast EDI-1.5 for the conversion of the SDTV to HDTV. The performance of the Fast EDI-1.5 is better than the NEDI on the image sharpness. However, the robustness of the Fast EDI-1.5 is not performed well specially near some texture region. In order to increasing the robustness of the Fast EDI-1.5 method in the conversion of the SDTV to HDTV, we try to using the Filtered sample data technique.

After we analysis the sample data using in the Fast EDI-1.5, we find that the resolution of the sample data using in the Fast EDI-1.5 is not sufficient enough for the accurate prediction of the unknown pixels. As a result, some prediction error occurs in the texture region which near to the edge region. Therefore, we designed to use the high-resolution filtered sample data (HRSD) in the Fast EDI-1.5.

As the Fast EDI-1.5 only contain one interpolation step and it is different from the normal EDI interpolation. Therefore, we need to modify the HRSD method in order to have the best performance in Fast EDI-1.5.

In Fig. 6.4, the modified version of the HRSD using in the Fast EDI-1.5 is shown. As the Fast EDI-1.5 method is using the block-based model, so the high order low-pass filter need to apply in the horizontal and vertical directions and then diagonal directions, in order to fit the block-based interpolation structure. It is because in the Fast EDI-1.5 method, the block-based interpolation structure causes the interpolation to interpolate the diagonal, horizontal and vertical unknown pixels together in every block region. As a result, using either diagonal directions or horizontal and vertical directions in HRSD cannot provide the optimal prediction result. Therefore, we need to combine the several filtering direction in the HRSD which using in the Fast EDI-1.5 method.

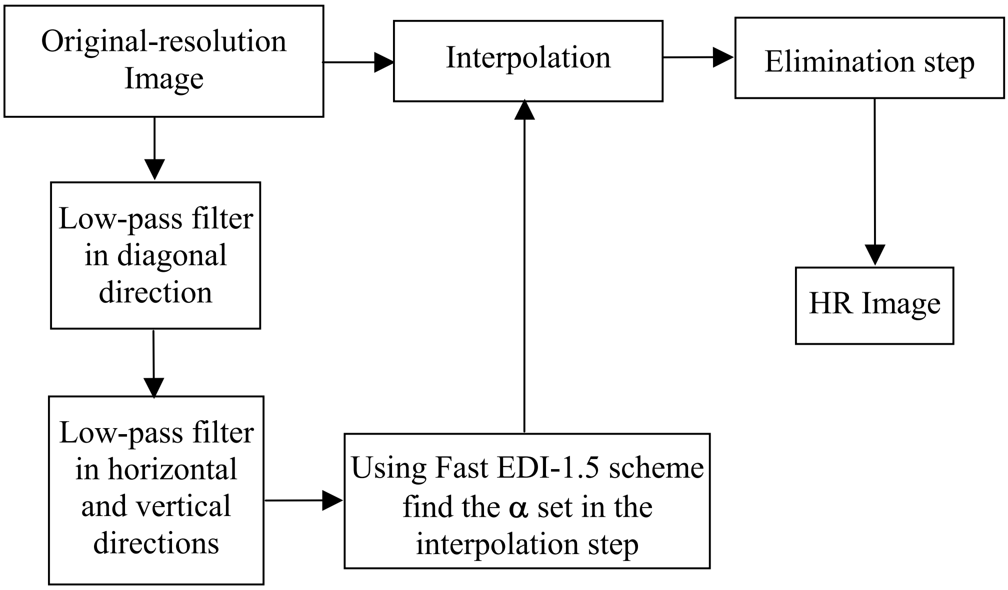


Figure 6.4: Flowing diagram of Fast EDI-1.5 using the high-resolution filtered sample data.

6.5 Fast Large Up-Scaling technique for EDI Interpolation

The Fast Large Up-Scaling (FLUS) technique can perform fast EDI interpolation and interpolation factor starts from 4 up to 2^n, where n is a positive integer and n>1. This FLUS technique can be using in EDI interpolation including NEDI, IEDI, MEDI and our proposed FEDI and Fast EDI-1.5.

EDI interpolation assumes that the low-resolution covariance is similar to the high-resolution covariance. According to our analysis, we find that the low-resolution covariance of the image is similar to the high-resolution covariance not only limited to the case when the interpolation factor is 2. It is also true for the high-resolution covariance in the EDI interpolation with an interpolation factor of 2^n, where n is a positive integer and n>1. As a result, the statistic property in the low-resolution covariance can be using in the high-resolution covariance with difference interpolation factors of 2^n, where n is a positive integer and n>1. Therefore, we can make use of the statistical information coming from the first EDI interpolation and then use these information in the next EDI interpolation, so that the computational cost will be reduced, as we did not need to re-calculate the Wiener filter coefficients for further EDI interpolation.

In Fig. 6.5, the white dots are original pixels, the gray and black dots are the interpolated results of the first FEDI interpolation in step one and step two respectively. Inside of the black box in Fig. 6.5, we will reuse the four Wiener filter coefficients in this box as shown in the black dash arrows with black labels α_0, α_1, α_2 and α_3. We assume that the statistic property inside this box will not change when the resolution of the box is increasing. That means the low-resolution statistical information can be reused inside the box when the resolution of the box is increasing. Therefore, we can use the Wiener filter coefficients which have been calculated in the first FEDI interpolation to find the unknown pixels which are the red dots as shown in Fig. 6.5 in step one of the second FEDI interpolation. After that, step two of the second FEDI interpolation can be used the same method to reuse the eight Winner filter coefficients which have been calculated in the step two of the first FEDI interpolation.

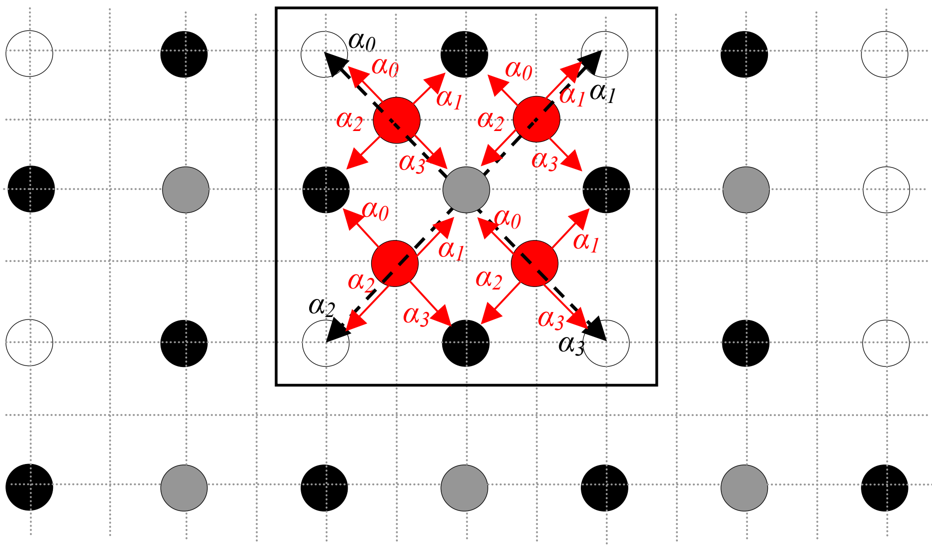


Figure 6.5: An example to illustrate the Fast Large Up-Scaling (FLUS) technique in first step of the second FEDI interpolation with interpolation factor 4.

6.6 Experiments

In order to illustrarate the performance of the proposed enhancement techniques ORSD and HRSD, we have implemented our proposed enhancement techniques base on the ADWS techniques which was mentioned in the previous chapter. It means that the enhancement techniques ORSD and HRSD will be applied to MEDI [47] together with the ADWS and compared with several conventional approaches in literature including bilinear interpolation, the NEDI method [2] and the MEDI method [47]. The high order low-pass filter using in this simulation is a 11 taps FIR filter and its coefficients are 0.0036, -0.0127, -0.0431, 0.0418, 0.2895, 0.4408, 0.2895, 0.0418, -0.0431, -0.0127, and 0.0036. All the programs were written in C++ language and run on the same platform. In the comparison, the values of the thresholds T_s and T_e in the proposed sample window techniques (ADWS) which were mentioned in the previous chapter are 8 and 10 respectively.

Our proposed FLUS method based on the FEDI interpolation was implemented and used to compare the computational times and visual quality between it and the FEDI interpolation when the interpolation factors are 4 and 8.

Finally, we will illustrate the outstanding performance of our proposed FEDI together with the two enhancement techniques (ORSD/HRSD and ADWS). Moreover, the outstanding performance of our proposed Fast EDI-1.5 together with the enhancement technique HRSD will be justify at the end of this section.

In order to compare the performance of the proposed enhancement techniques ORSD and HRSD in terms of objective image quality, a low-resolution image was obtained by down sampling a high-resolution image by a factor of 2 and then we reconstructed it by conventional methods and our algorithm. We compare the interpolated high-resolution image with the original high-resolution image in terms of PSNR and visual quality. We have used various test images and the PSNR of all test images are summarized in Table 6.1.

In Table 6.1, when our proposed pre-process of the sampled data techniques ORSD or HRSD together with the ADWS for applying to the MEDI, it gains an average improvement of 0.16 and 0.2 dB, respectively. The maximum PSNR improvement of the proposed techniques ORSD using in the MEDI (ADWS) and HRSD in the MEDI (ADWS) compared to the MEDI is 0.24 and 0.31 dB, respectively.

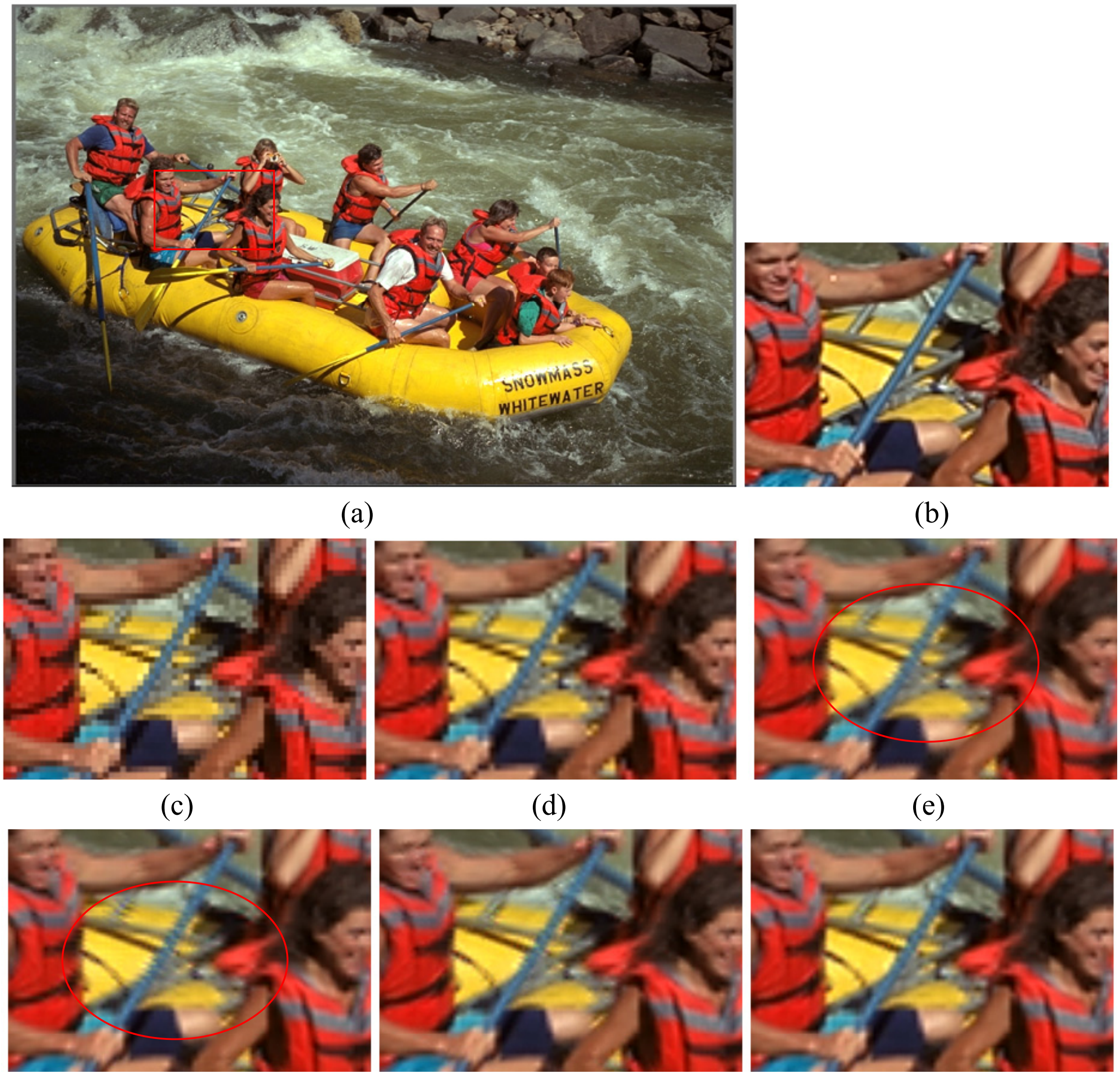
Using proposed pre-processing technique ORSD in the MEDI (ADWS) could further improve the edge performance. In Fig. 6.6, the interpolated image (d) using bilinear interpolation contains many aliasing artifacts so it has the worst visual quality. The interpolated image (e) using NEDI resulted in a more continuous edge image, but it is still affected by burring and ringing effect as shown inside of the red ellipse. The interpolated image (f) using MEDI contain a lot of ringing effect as shown inside of the red ellipse. The interpolated image (g) using our proposed method ADWS in MEDI can reduce most of the ringing effect in the MEDI method and is sharper than the NEDI method. We can see that the interpolated image (h) using our proposed method ORSD and ADWS in MEDI has the best visual quality compared with the Bilinear, NEDI and MEDI interpolation methods.

Moreover, in Fig. 6.7, the interpolated image (d) using bilinear interpolation cannot perform well in the texture region with multi-direction edges. It causes a lot of aliasing effects. The interpolated image (e) using NEDI and (f) using MEDI did not perform well in this kind of the texture region with multi-directional edges as shown inside of the red ellipses. Our proposed ADWS used in the MEDI in the interpolated image (g) can reduce a lot of artifacts such as ringing effect compared to the MEDI and NEDI method. However, the proposed ADWS still suffers from some artifacts as shown inside of the red ellipse. Using our proposed ORSD and ADWS in the MEDI can reduce the entire artifact which appear in the MEDI

(ADWS), MEDI and NEDI methods as shown in the interpolated image (h) and it results in the best visual quality compare to conventional interpolation approach.

Table 6.1: PSNR (dB) of interpolated color (RGB) test images produced by using the (i) Bilinear, (ii) NEDI, (iii) MEDI, proposed (iv) ADWS, (v) ADWS and ORSD and (vi) ADWS and HRSD applying to the MEDI interpolation method.

	(i) Bilinear	(ii) NEDI	(iii) MEDI	(iv) MEDI (ADWS)	(v) MEDI (ADWS) (ORSD)	(vi) MEDI (ADWS) (HRSD)
kodim01	23.75	23.63	23.60	23.60	23.74	23.82
kodim02	30.70	30.71	30.64	30.65	30.81	30.82
kodim03	31.42	31.69	31.61	31.61	31.76	31.68
kodim04	30.48	30.47	30.35	30.37	30.59	30.66
kodim05	23.78	23.98	23.94	23.96	24.16	24.13
kodim08	21.09	20.96	20.95	20.95	21.07	21.18
kodim09	29.00	29.03	29.04	29.05	29.16	29.23
kodim10	28.96	28.92	28.92	28.90	29.02	29.10
kodim11	26.82	26.80	26.78	26.80	26.93	26.94
kodim13	21.99	21.90	21.79	21.77	21.99	22.04
kodim14	26.46	26.44	26.43	26.44	26.58	26.62
kodim16	29.16	29.10	29.03	29.02	29.14	29.19
kodim17	29.50	29.53	29.47	29.48	29.65	29.67
kodim18	25.72	25.69	25.64	25.62	25.76	25.82
kodim20	28.62	28.71	28.71	28.72	28.83	28.85
kodim21	26.05	25.93	25.89	25.89	26.02	26.11
kodim23	31.26	31.51	31.37	31.37	31.59	31.63
Average	27.34	27.35	27.30	27.31	27.46	27.50



(a) (b)

(c) (d) (e)

(f) (g) (h)

Figure 6.6: (a) Original test image "kodim14" and zoomed-in portion of the: (b) original image, (c) down sampled image, (d) reconstructed by Bilinear, (e) reconstructed by NEDI, (f) reconstructed by MEDI, (g) reconstructed using ADWS in MEDI and (h) reconstructed using ORSD and ADWS in MEDI.

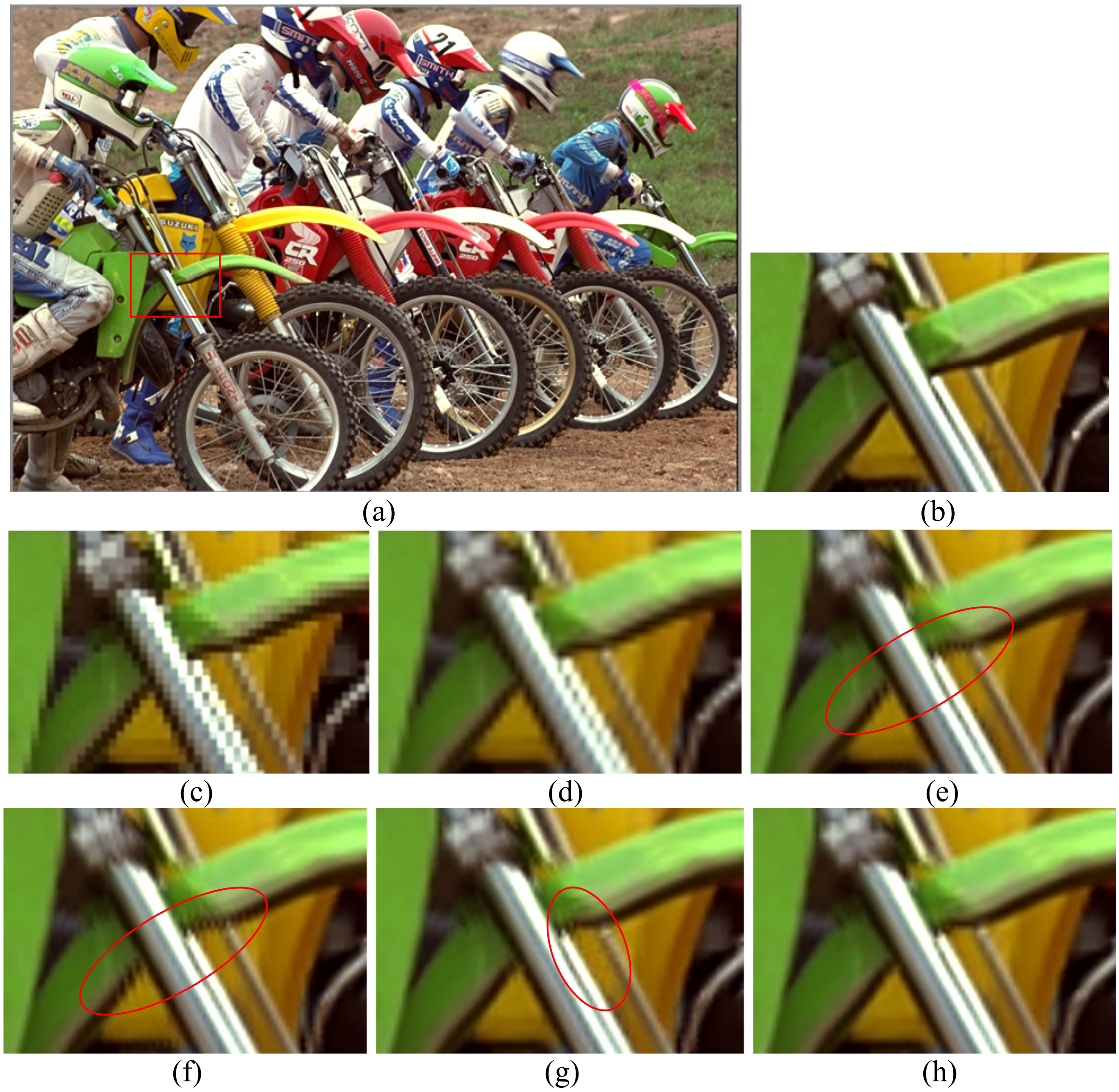
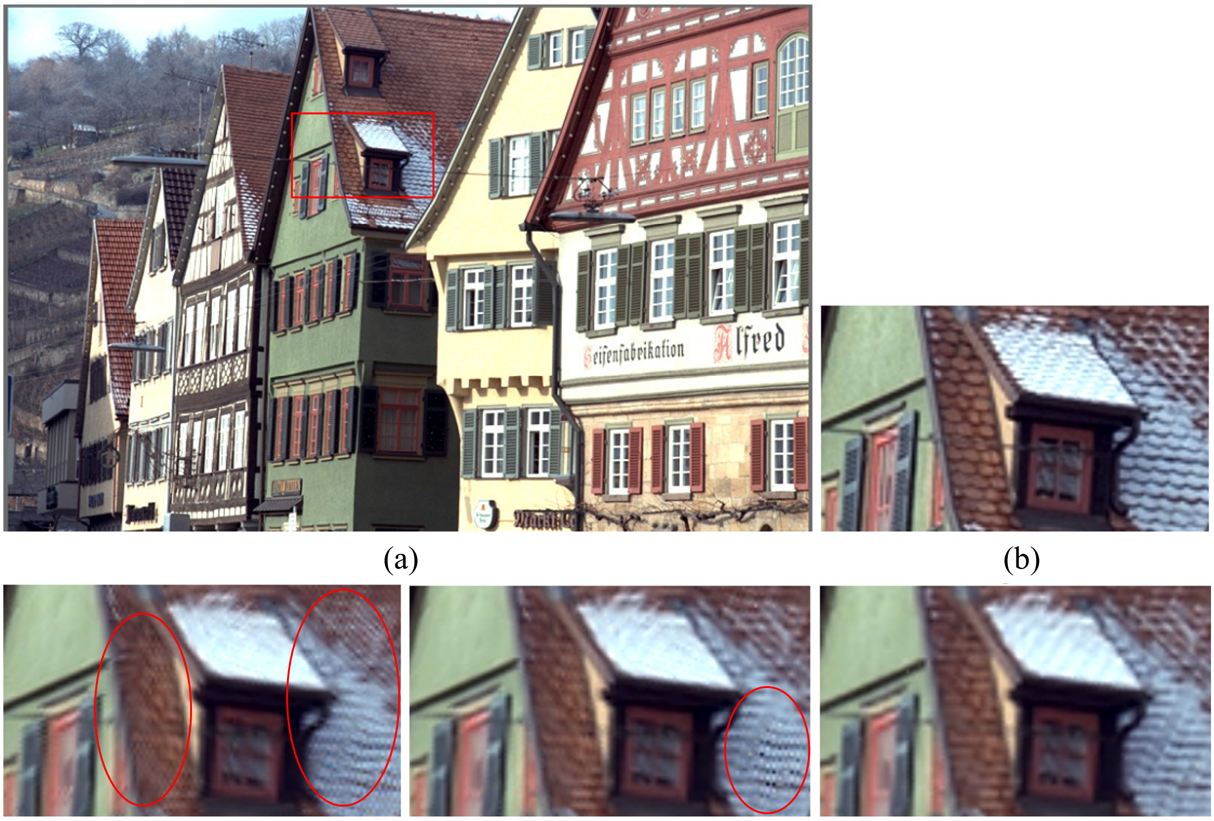


Figure 6.7: (a) Original test image “kodim05” and zoomed-in portion of the: (b) original image, (c) down sampled image, (d) reconstructed by Bilinear, (e) reconstructed by NEDI, (f) reconstructed by MEDI, (g) reconstructed using ADWS in MEDI and (h) reconstructed using ORSD and ADWS in MEDI.

For some images that contain very fine edge or some fine texture regions, we can use the proposed pre-processing approach HRSD, to interpolate it so that it can keep the sharpness of the edge and reduce the artifacts and ringing effect in the interpolated image. The outstanding performance of the proposed pre-processing HRSD method can be seen in Fig. 6.8 and Fig. 6.9.

In Fig. 6.8, the roof is a texture region which contains many fine edges, therefore, using MEDI method results in many artifacts and ringing effect as shown inside of the red ellipses in interpolated image (c). Moreover, using the pre-processing method ORSD, the visual quality of the image can only improve a bit, as there still contains many artifacts as shown inside of red ellipse in the interpolated image (d). However, when we apply the proposed pre-processing HRSD method, most of the artifacts and ringing effect were removed and thus it shows the best subjective performance among different methods.

Fig. 6.9 also shows the outstanding performance of the HRSD pre-processing method together with the ADWS method compared to the MEDI method. Clearly, we can see that in the interpolated image (b), using the proposed HRSD and ADWS method can produce an interpolated image with sharpened edge and it results also in a more natural image compared with that of the MEDI method in interpolation image (a).



(a) (b)

(c) (d) (e)

Figure 6.8: (a) Original test image “kodim08” and zoomed-in portion of the: (b) original image, (c) reconstructed by MEDI, (d) reconstructed using ORSD and ADWS in MEDI and (e) reconstructed using HRSD and ADWS in MEDI.

(a)

(b)

Figure 6.9: The interpolated images of the test image “kodim13”: (a) reconstructed by MEDI and (b) reconstructed using HRSD and ADWS in MEDI.

The Fast Large Up-Scaling (FLUS) technique can be used in EDI interpolation methods including NEDI, IEDI, MEDI and our proposed FEDI and Fast FEDI-1.5. Let us use our proposed FEDI interpolation to test the performance of the FLUS technique.

In order to compare the performance of the proposed FLUS technique in terms of subjective image quality, a low-resolution image was obtained by down sampling a high-resolution image by a factor of 2 and then we reconstructed it by approaches an interpolation factor of 4 using FLUS in FEDI, original FEDI and Bilinear interpolation. The resolution of the original image (768×512) is not the same as the interpolated image (1536×1024) and (3072×2048) which was interpolated from the down sampled image (384×256) by an interpolation factor of 4 and 8 respectively, so the objective image quality (PSNR) will not be compared here and it is also not the subject in this simulation. The main subject in this simulation is to compare the visual quality of the FLUS in FEDI which make use of the proposed FLUS technique on the FEDI with the original FEDI and bilinear interpolation.

In Tables 6.2, we can see that our proposed FLUS in FEDI can reduce the computation times up to 82% and has an average of 78% reduction computational time as compared to the FEDI interpolation when the interpolation factor is 4. Moreover, when the interpolation factor is getting large, the computational time will be further reduced compare to the FEDI interpolation. In Table 6.3, when the interpolation factor is increased to 8, our proposed FLUS in FEDI can reduce the computation times up to 94% and have an average of 91% reduction compare to the FEDI interpolation. Therefore, using our proposed FLUS in different EDI interpolation can reduce a large among of the computation cost compared to the use of EDI interpolation. The comparison of the visual quality between the FLUS in FEDI and the FEDI interpolation will be discussed in the next section.

Table 6.2. Interpolation time of the testing images using different interpolation methods with an interpolation factor 4.

Name Of The Image	FEDI	FEDI (FLUS)	Time reduce of FEDI (FLUS) Scheme Compared To FEDI
kodim01	52 s	10 s	80.77 %
kodim02	30 s	8 s	73.33 %
kodim03	22 s	5 s	77.27 %
kodim04	33 s	8 s	75.76 %
kodim05	56 s	10 s	82.14 %
kodim08	52 s	10 s	80.77 %
kodim09	23 s	5 s	78.26 %
kodim10	30 s	7 s	76.67 %
kodim11	40 s	9 s	77.50 %
kodim13	57 s	10 s	82.46 %
kodim14	50 s	10 s	80.00 %
kodim16	35 s	8 s	77.14 %
kodim17	36 s	8 s	77.78 %
kodim18	46 s	9 s	80.43 %
kodim19	35 s	8 s	77.14 %
kodim20	22 s	5 s	77.27 %
kodim21	32 s	7 s	78.13 %
kodim23	22 s	6 s	72.73 %
kodim24	45 s	9 s	80.00 %
Average	37.79 s	8.00 s	78.19 %

Table 6.3. Interpolation time of the testing images using different interpolation methods with an interpolation factor 8.

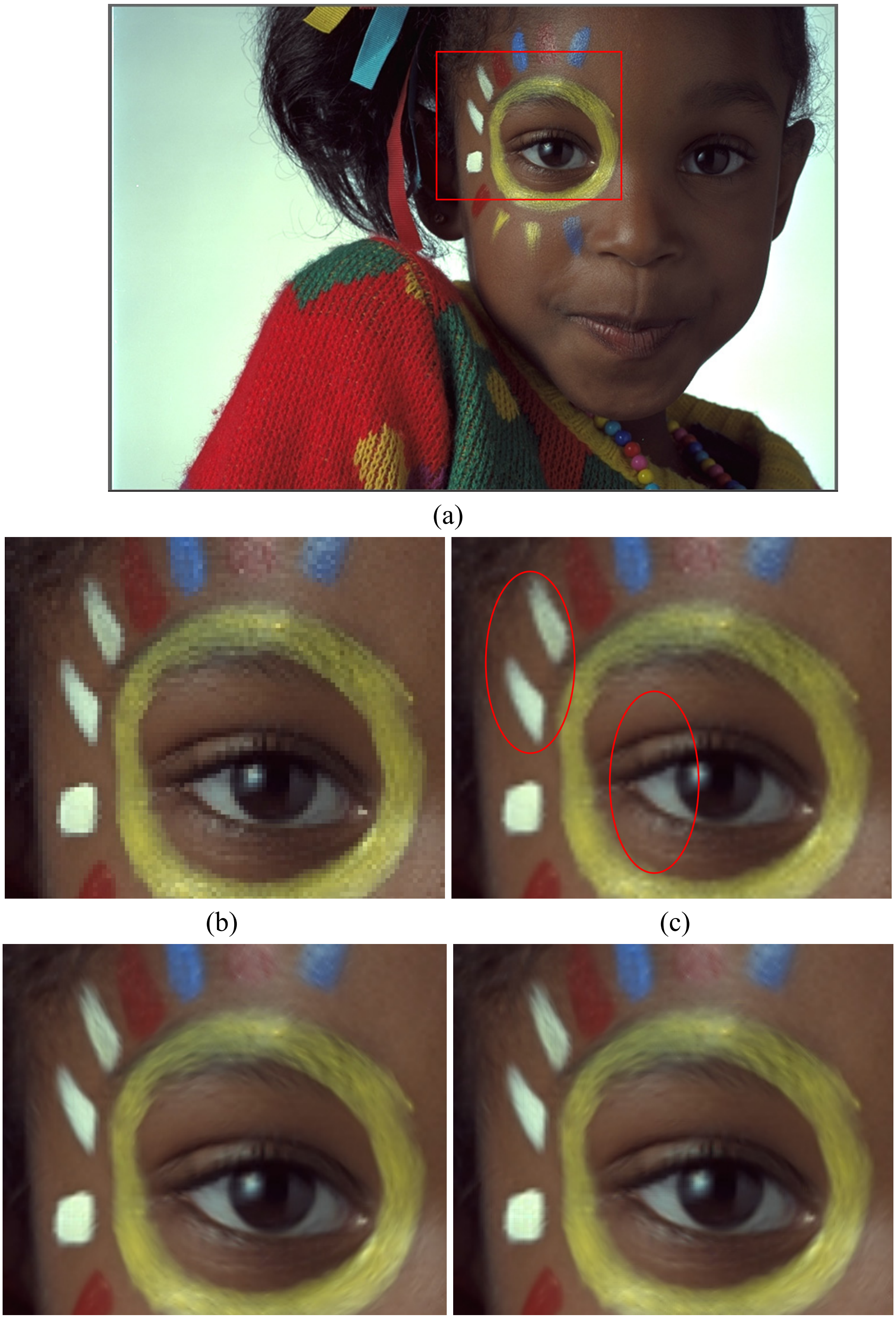
Name Of The Image	FEDI	FEDI (FLUS)	Time reduce of FEDI (FLUS) Scheme Compared To FEDI
kodim01	177 s	11 s	93.79 %
kodim02	69 s	8 s	88.41 %
kodim03	59 s	6 s	89.83 %
kodim04	84 s	9 s	89.29 %
kodim05	189 s	11 s	94.18 %
kodim08	187 s	11 s	94.12 %
kodim09	71 s	5 s	92.96 %
kodim10	81 s	8 s	90.12 %
kodim11	109 s	9 s	91.74 %
kodim13	195 s	11 s	94.36 %
kodim14	149 s	11 s	92.62 %
kodim16	91 s	8 s	91.21 %
kodim17	92 s	9 s	90.22 %
kodim18	143 s	10 s	93.01 %
kodim19	104 s	8 s	92.31 %
kodim20	61 s	5 s	91.80 %
kodim21	108 s	7 s	93.52 %
kodim23	53 s	6 s	88.68 %
kodim24	139 s	10 s	92.81 %
Average	113.74 s	8.58 s	91.84 %

In Fig. 6.10, the test image “kodim15” as shown in the image (a) was down sampled by a factor 2 as shown in image (b) and then interpolated by different interpolation methods with an interpolation factor 4. We can see that the interpolated image (c) using bilinear interpolation contains a lot of aliasing effects in the edge regions as shown inside of the red ellipses. However, the interpolated image (d) using twice FEDI interpolation to perform the 4 times up scaling has reduced most of the aliasing caused by bilinear and has a sharper and more continuous edge performance and gives the best visual quality. In additional, the interpolated image (e) using the FLUS in FEDI has the same visual quality compare to the interpolated image (d) using twice FEDI interpolation.

Moreover, in Fig. 6.11, the testing image “kodim10” as shown in image (a) was down sampled by a factor 2 as shown in image (b) and then interpolated it by different interpolation methods with an interpolation factor of 4. We can see that the interpolated image (c) using the bilinear interpolation contains a lot of aliasing effects in the edge regions as shown inside the red ellipses. However, the interpolated image (d) using twice FEDI interpolation to perform the 4 times up scaling has reduced most of the aliasing caused by bilinear and has a sharper and more continuous edge performance and the best visual quality. On the other hand, the interpolated image (e) using the FLUS in FEDI has the same visual quality compared to the interpolated image (d) using twice FEDI interpolation.

Although, in Figs. 6.10 and 6.11, the interpolated image (d) using twice FEDI interpolation have the same visual quality compare to the interpolated image (e) using FLUS in FEDI, the interpolation time of using FLUS in FEDI is several times faster than using twice FEDI interpolation. It is because using FLUS in FEDI to perform the 4 times up scaling interpolation only need to calculate the first FEDI interpolation and the second interpolation will make use of the coefficients which have been found in the first FEDI interpolation. This is because the computation for the FLUS in FEDI mainly involves the first FEDI interpolation and the second interpolation only needs a very small computational time, compared to the FEDI interpolation in the normal cause for the 4 times up scaling interpolation. The ladder case needs twice the FEDI interpolation and the computational time because in the second FEDI interpolation, the input image size is double, and usually the computational time of the second FEDI interpolation is fourfold times of the first FEDI interpolation, so that the total computational time of the twice FEDI interpolation will be about 5 times of the computational time of the first FEDI interpolation. If the up scaling factor is increasing, the computational time of using FLUS in FEDI will be greatly reduced compared to FEDI interpolation.

Now, let us increase the interpolation factor to 8, and see the performance of the visual quality of using FLUS in FEDI and the FEDI in Fig. 6.12. In this figure, the interpolated image (c) using bilinear interpolation contains many aliasing effects along the main edge region which causes discontinue of the edge region as shown inside of the red ellipses. However, when we use the FEDI interpolation as shown in the interpolated image (d), it results in a more continuous edge image. Moreover, using the proposed FLUS in FEDI as shown in interpolated image (e), it also can interpolate a more continuous edge image. The performance of the visual quality of the proposed FLUS using in FEDI is almost the same as the FEDI interpolation. We can further enhance the performance of the proposed FLUS by using the other two proposed enhancement techniques ORSD or HRSD together.

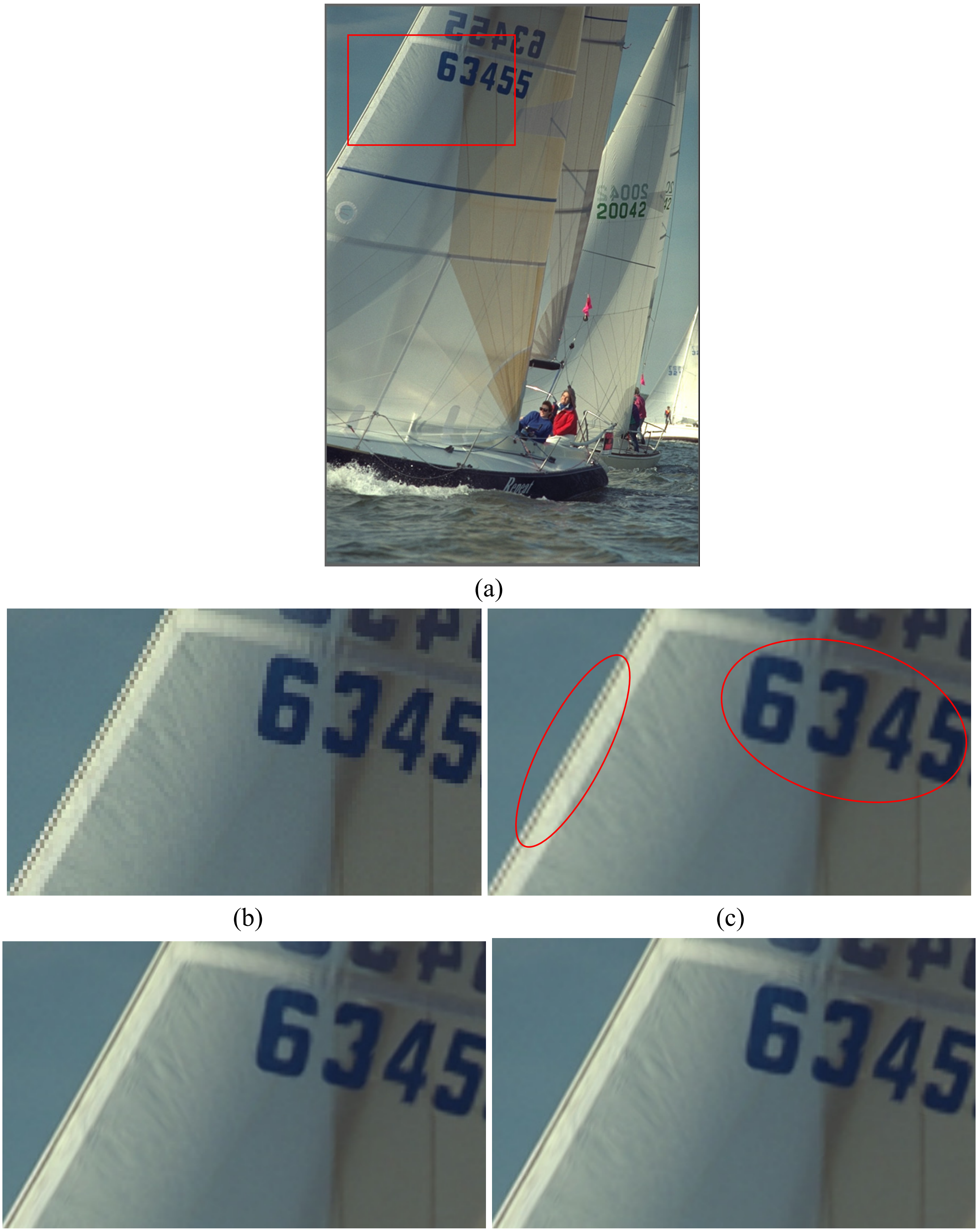


(a)

(b) (c)

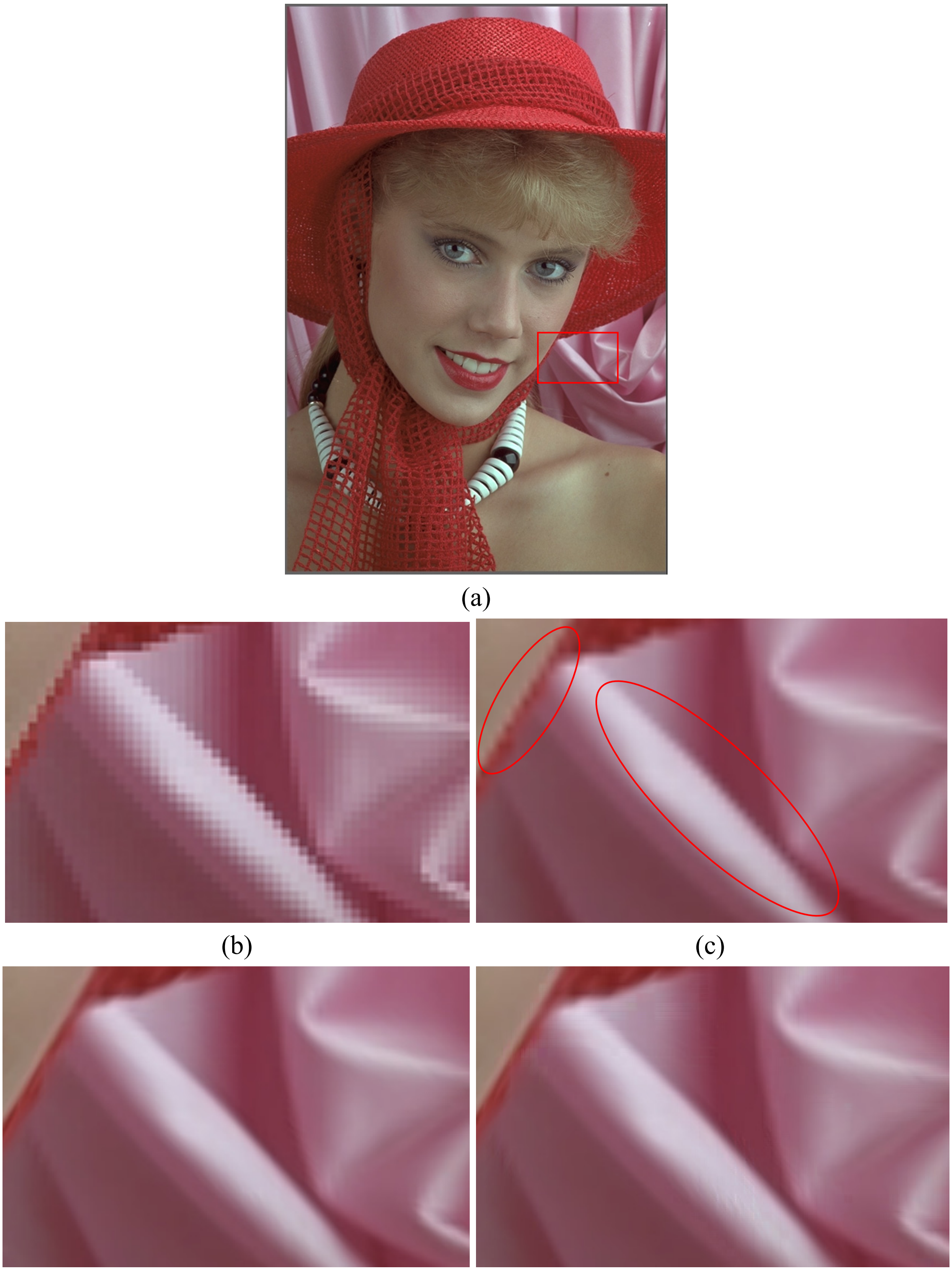
(d) (e)

Figure 6.10: (a) Original test image “kodim15” and zoomed-in portion of the: (b) down sampled image and interpolated image with an interpolation factor 4 (c) reconstructed by Bilinear interpolation, (d) reconstructed using FEDI and (e) reconstructed using FLUS in FEDI



(a) (b) (c) (d) (e)

Figure 6.11: (a) Original test image “kodim10” and zoomed-in portion of the: (b) down sampled image and interpolated image with an interpolation factor 4 (c) reconstructed by Bilinear interpolation, (d) reconstructed using FEDI and (e) reconstructed using FLUS in FEDI.



(a) (b) (c) (d) (e)

Figure 6.12: (a) Original test image “kodim04” and zoomed-in portion of the: (b) down sampled image and interpolated image with an interpolation factor 8, (c) image reconstructed by using Bilinear interpolation, (d) image reconstructed by using FEDI and (e) image reconstructed by using FLUS in FEDI.

Let us move on to the experimental results on our proposed FEDI together with the two enhancement techniques ORSD and HRSD. In Table 6.4, we have summarized all the PSNR of our proposed FEDI with the two enhancement techniques. We can see that the FEDI with the ADWS technique is slightly increased compare to the MEDI with the ADWS, also the PSNR of the FEDI with the ADWS and ORSD techniques increases slightly compared to that of the MEDI with the ADWS and ORSD. The PSNR of the FEDI using the ADWS and HRSD techniques decreases slightly than the MEDI with ADWS and HRSD.

In order to show the outstanding performance of the proposed algorithm FEDI and two enhancement techniques, we used the same testing image “kodim08” from the previous experimental result (Fig. 6.8) as shown in Fig. 6.13. In Fig. 6.13, the interpolated image (b) using ADWS in MEDI has a lot of prediction error and artifacts as shown inside of the red ellipse. However, using our proposed FEDI and ADWS, the interpolated image (e) has reduced the effect of artifact as shown inside of the red ellipse. Moreover, the interpolated image (c) using ORSD and ADWS in MEDI still has some artifacts (unwanted black and white pixels inside of the red ellipse in Fig. 6.13: c) on high frequency regions of an image. Using the same enhancement techniques ORSD and ADWS in FEDI can remove most of the unwanted artifacts as shown in the interpolated image (f). It is because the FEDI can make a fully use of the known neighbour pixels to estimate the unknown pixels, so the estimation is more accurate. Moreover, using the HRSD and ADWS in FEDI can further enhance the robustness of the FEDI. We can see that using the HRSD and ADWS in MEDI still contain some uneven pixels in the interpolated image (d) as shown inside the red ellipse. When we used the HRSD and ADWS in FEDI the uneven pixels that appear inside of the red ellipse in the interpolated image (d) have been removed as shown inside of the red ellipse in the interpolated image (g).

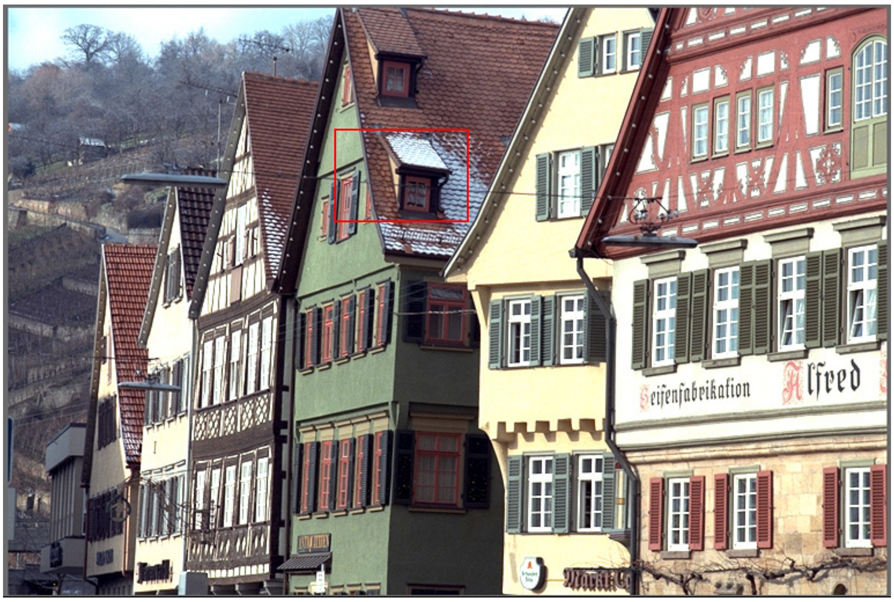
Table 6.4: The PSNR (dB) of interpolated color (RGB) test images by different interpolation methods includes FEDI.

	Bilinear	NEDI	MEDI	MEDI (ADWS)	MEDI (ADWS) (ORSD)	MEDI (ADWS) (HRSD)	FEDI (ADWS)	FEDI (ADWS) (ORSD)	FEDI (ADWS) (HRSD)
kodim01	23.75	23.63	23.60	23.60	23.74	23.82	23.65	23.78	23.75
kodim02	30.70	30.71	30.64	30.65	30.81	30.82	30.70	30.83	30.76
kodim03	31.42	31.69	31.61	31.61	31.76	31.68	31.64	31.77	31.62
kodim04	30.48	30.47	30.35	30.37	30.59	30.66	30.39	30.62	30.60
kodim05	23.78	23.98	23.94	23.96	24.16	24.13	23.98	24.15	24.04
kodim08	21.09	20.96	20.95	20.95	21.07	21.18	20.98	21.12	21.14
kodim09	29.00	29.03	29.04	29.05	29.16	29.23	29.09	29.17	29.16
kodim10	28.96	28.92	28.92	28.90	29.02	29.10	28.95	29.02	29.01
kodim11	26.82	26.80	26.78	26.80	26.93	26.94	26.82	26.94	26.87
kodim13	21.99	21.90	21.79	21.77	21.99	22.04	21.81	22.01	22.01
kodim14	26.46	26.44	26.43	26.44	26.58	26.62	26.45	26.57	26.55
kodim16	29.16	29.10	29.03	29.02	29.14	29.19	29.07	29.16	29.16
kodim17	29.50	29.53	29.47	29.48	29.65	29.67	29.50	29.65	29.61
kodim18	25.72	25.69	25.64	25.62	25.76	25.82	25.68	25.81	25.77
kodim19	25.77	25.22	25.08	25.10	25.45	25.88	25.03	25.76	25.81
kodim20	28.62	28.71	28.71	28.72	28.83	28.85	28.75	28.84	28.78
kodim21	26.05	25.93	25.89	25.89	26.02	26.11	25.93	26.04	26.06
kodim23	31.26	31.51	31.37	31.37	31.59	31.63	31.40	31.60	31.56
kodim24	24.65	24.64	24.43	24.44	24.67	24.76	24.45	24.70	24.69
Average	27.11	27.10	27.03	27.04	27.21	27.27	27.07	27.24	27.21

When the proposed ORSD or HRSD works together with the ADWS techniques and they are applied to the FEDI, it can further improve the visual quality in the high frequency region without increasing the number of the sample points using in the interpolation estimation as discuss in the chapter 4. Let us use the same testing image "kodim19" (lighthouse) image in chapter 4 to illustrate the effect of the proposed algorithm in the high frequency region as shown in Fig. 6.14. Because of the down sampling operation of the "kodim19" from size 512×768 to 256×384, the artifact that appears in the high frequency regions of the interpolated image (b) and (e) is more serious. However, the interpolated image (e) using the FEDI with the ADWS technique contains less artifacts than the interpolated image (b) using the MEDI with the ADWS technique as shown inside of the red ellipse. When we applied our proposed ORSD and ADWS techniques on the FEDI as shown in the interpolated image (f). The entire artifact in the high frequency region was removed and it gives us a very good visual quality compare to the MEDI using the ORSD and ADWS techniques in the interpolated image (c) or the MEDI using the HRSD and ADWS techniques in the interpolated image (d). Although the PSNR of the FEDI using the ADWS and ORSD techniques is lower than the MEDI using the ADWS and HRSD, the visual quality of the FEDI using the ADWS and ORSD is better than the MEDI using the ADWS and HRSD. The best visual quality of the interpolated image is the interpolated image (h) which was obtained by using the FEDI with the ADWS and HRSD techniques.

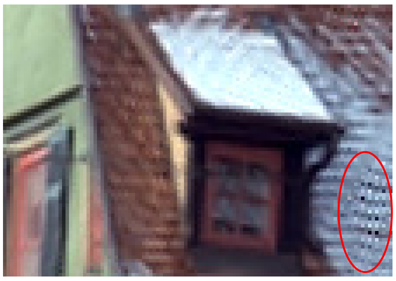
According to the Figs. 6.13 and 6.14, we can see that using our proposed FEDI structure with ADWS to interpolate image will have better visual quality than using the MEDI structure with ADWS. It is because the FEDI can make full use of the neighbour known pixels to predict the unknown pixels. However, their visual quality still affected by some artifacts that happen in some textures and high frequency regions, which is a common problem in conventional edge direction interpolation. Therefore, we proposed two enhancement techniques to solve this kind of problems. The proposed enhancement techniques ORSD and HRSD can help the FEDI to interpolate the best visual quality image in both texture and high frequency regions than conventional edge directed interpolation that always have artifacts in these regions.

In Fig. 6.14, we can see the outstanding performance of the FEDI that use the ADWS and HRSD in the high frequency content in the image. The proposed algorithm can solve the problem of the artifacts in the high frequency region which use the conventional EDI interpolation approach. Compared with the MEDI structure, our FEDI structure is more robust to the texture and high frequency regions and it can interpolate images with the best visual quality.

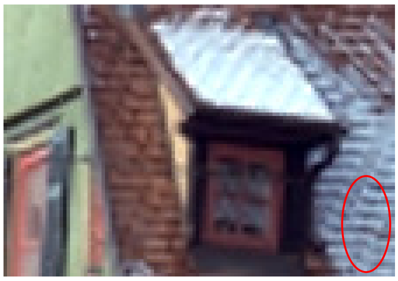


(a)

(b)

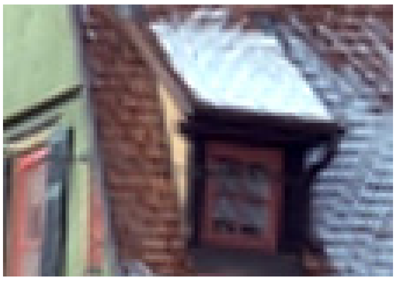


(c)

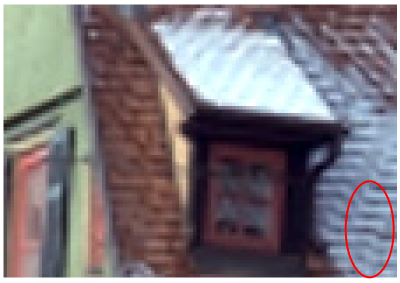


(d)

(e)



(f)



(g)

Figure 6.13: (a) Original test image “kodim08” and zoomed-in portion of the interpolated image: (b) reconstructed using ADWS in MEDI, (c) reconstructed using ORSD and ADWS in MEDI (d) reconstructed using HRSD and ADWS in MEDI and (e) reconstructed using ADWS in FEDI, (f) reconstructed using ORSD and ADWS in FEDI and (g) reconstructed using HRSD and ADWS in FEDI.

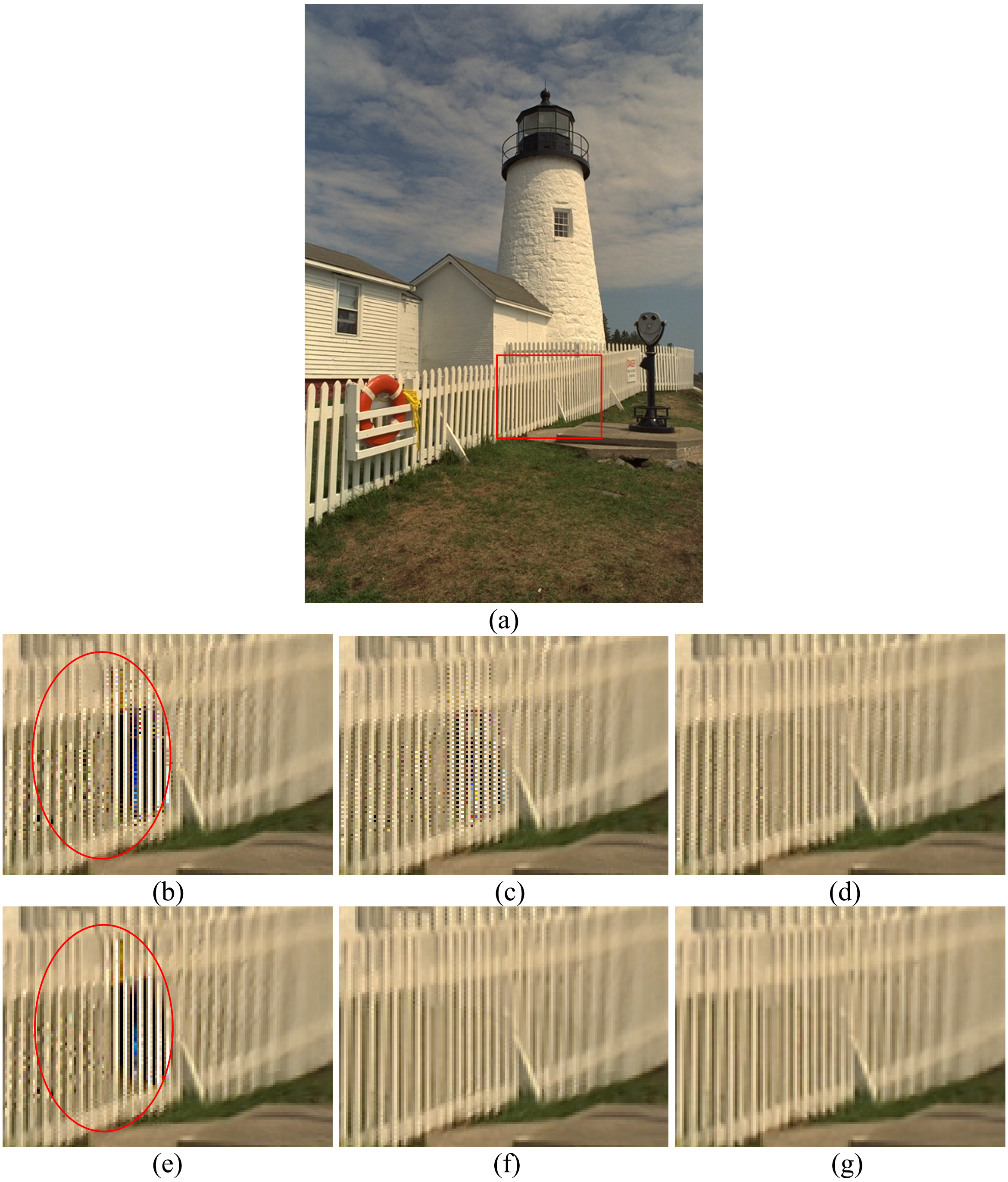


Figure 6.14: (a) Original test image “kodim19” and zoomed-in portion of the interpolated image: (b) reconstructed using ADWS in MEDI, (c) reconstructed using ORSD and ADWS in MEDI (d) reconstructed using HRSD and ADWS in MEDI and (e) reconstructed using ADWS in FEDI, (f) reconstructed using ORSD and ADWS in FEDI and (g) reconstructed using HRSD and ADWS in FEDI.

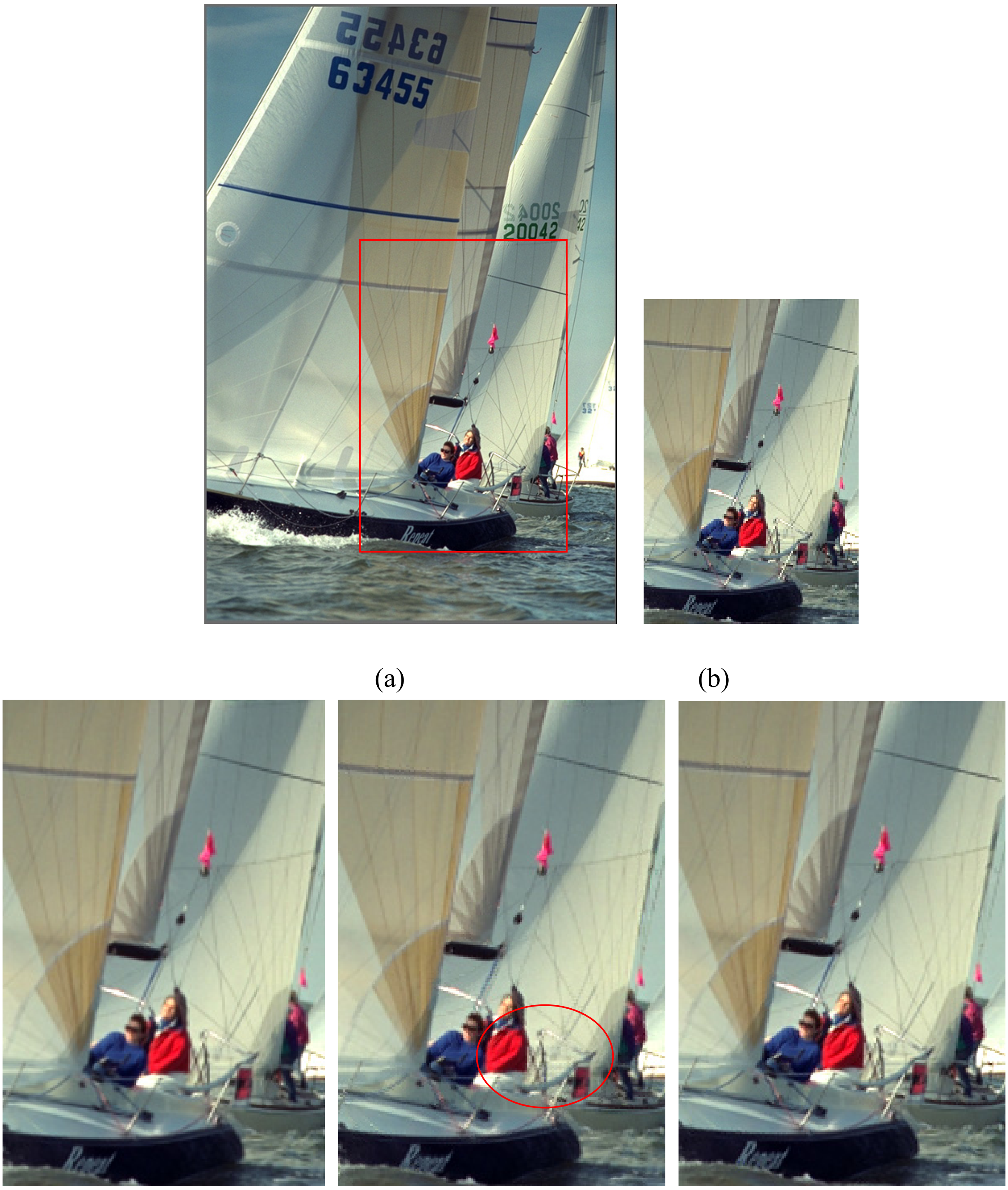
Finally, let us use our proposed enhancement technique HRSD on the Fast EDI-1.5 and see the improvement as compared to the original Fast EDI-1.5 and NEDI with downsampling by a factor ¾.

In order to compare the performance of the proposed enhanced Fast EDI-1.5 in terms of subjective image quality, a low-resolution image was obtained by down sampling a high-resolution image by a factor of 2 and then we reconstructed it by the enhanced Fast EDI-1.5, original Fast EDI-1.5 and NEDI with a downsampling factor of ¾. The resolution of the original image (768×512) is not the same as the interpolated image (576×384) which interpolated from the down sampled image (384×256) by an interpolation factor of 1.5, so the objective image quality (PSNR) will not be compared here and it is also not the subject in this simulation. The main subject in this simulation is to compare the visual quality of the enhanced Fast EDI-1.5 which make use of the proposed enhancement technique HRSD on the Fast EDI-1.5 with the original Fast EDI-1.5 and NEDI.

In Fig. 6.15 (a), the original of the "kodim10" image was down sampled to a low-resolution image as shown in Fig. 6.15 (b). We then interpolated the low-resolution image to a high-resolution image as shown in the interpolated images (c), (d) and (e). Compared with the interpolated images (c) and (d), the interpolated image (d) using Fast EDI-1.5 is sharper than the interpolated image (c) using the NEDI with a down sampled operator. Although the Fast EDI-1.5 can interpolate a sharp image, it still has some artifacts that appear near to the texture regions as shown inside of the red ellipse. These artifacts will reduce the visual quality of the interpolated image using the Fast EDI-1.5. However, this kind of the artifacts can be removed when we use the proposed enhancement technique HRSD in the Fast EDI-1.5 as shown in the interpolated image (e). In additional, the interpolated image (e) is sharper than the original Fast EDI-1.5 and NEDI.

Moreover, in Fig. 6.16 (a), the original “kodim17” image was down sampled to a low-resolution image as shown in Fig. 6.16 (b). We then interpolated the low-resolution image to a high-resolution image as shown in the interpolated images (c), (d) and (e). Compared with the interpolated images (c) and (d), the interpolated image (d) using Fast EDI-1.5 is sharper than the interpolated image (c) using the NEDI with a down sampled operator. Although the Fast EDI-1.5 can interpolate a sharp image, it still has some artifacts that appear near to the texture regions as shown inside of the red ellipses. These artifacts will reduce the visual quality of the interpolated image using the Fast EDI-1.5. However, this kind of the artifacts will be removed when we used the proposed enhancement technique HRSD in the Fast EDI-1.5 as shown in interpolated image (e). Also, the interpolated image (e) is sharper than that of the original Fast EDI-1.5 and NEDI.

Both Figs. 6.15 and 6.16 can show the outstanding performance of the enhanced Fast EDI-1.5. Therefore, we can conclude that the enhanced Fast EDI-1.5 that as compared with the enhancement technique using the HRSD and Fast EDI 1.5 together to form an enhanced version of the interpolation algorithm can interpolate a sharp image and make the algorithm more robust to the texture region or high frequency region. As a result, the visual quality of the interpolated image is improved.



(a) (b)

(c) (d) (e)

Figure 6.15: (a) Original test image "kodim10", (b) zoomed-in portion of the down sampled image and zoomed-in portion of the interpolated image: (c) reconstructed using NEDI with downsampling by a factor ¾, (d) reconstructed using Fast EDI-1.5 and (e) reconstructed using HRSD in Fast EDI-1.5.

(a) (b)

(c) (d) (e)

Figure 6.16: (a) Original test image "kodim17", (b) zoomed-in portion of the down sampled image and zoomed-in portion of the interpolated image: (c) reconstructed using NEDI with downsampling by a factor ¾, (d) reconstructed using Fast EDI-1.5 and (e) reconstructed using HRSD in Fast EDI-1.5.

6.7 Summary

In order to further improve the quality of the interpolated image, we propose to pre-process the original sample data before using the ADWS for the EDI. This pre-processing step can be done by applying a high order low-pass filter in different directions to the original image for the steps one and two to produce the original-resolution filtered sample data (ORSD) or the high-resolution filtered sample data (HRSD).

As there is a substantial among of work required for up scaling interpolation for videos or images to be displayed in the high-resolution display system, we proposed the Fast Large Up Scaling (FLUS) for EDI interpolation which can reduce the computational time when the interpolation factor is 2^n, where n is a positive integer and n>1. This is able to keep the edge sharper and continuous in the interpolated image.

Experimental results show that the proposed ADWS together with the proposed pre-process step can overcome the existing problem of the new edge-directed interpolation with high frequency contents or texture region by considering multiple directional elliptic windows and the pre-processing of the original sample data. As a result, the proposed algorithm produces interpolated images with better objective and subjective qualities compared to that of the conventional interpolation methods. Moreover, the experimental results also show that the proposed FLUS using in FEDI can interpolate images that have the similar visual quality and have about 78% and 91% on average reduction in computational time as compared to the use of FEDI interpolation when the interpolation factor is 4 and 8 respectively. In additional, using FLUS in FEDI can perform the 4 or 2^n times up scaling interpolation by computing the first FEDI interpolation only, so it can save a lot of computational time.

Experimental results also show that the proposed ADWS together with the proposed processing step can be applied to the proposed FEDI and it can produce interpolated image with the best visual quality compared to other edge directed interpolation methods. Moreover, the proposed Fast EDI-1.5 can make use of the proposed enhancement technique HRSD to form a more robust interpolation method. It is faster and has the best visual quality in the conversion of the SDTV to HDTV, say for example.

Chapter 7

Conclusion and Future Work

7.1 Conclusion

In many vision applications, the image system can see the world by various sensors, such as still or video cameras with different resolutions. Nowadays, many people talking about the HDTV system with a high-resolution display. However, not all the SD videos or SD movies can be and was produced with the high-resolution quality. This thesis investigates the techniques for the interpolation of the low-resolution image to a high-resolution image. We give results of our study on the investigation of the techniques for the fast SDTV to HDTV conversion and fast large up scaling interpolation.

7.1.1 Further Improved Edge-Directed Interpolation (FEDI) and Fast EDI for SDTV to HDTV Conversion

A main issue arising from the NEDI scheme is that the interpolation structure is not optimal and the interpolation factor is the multiple of 2. The interpolation structure is not optimal means that the interpolated image will contain a lot of prediction error that limit the application of the NEDI. The other issue is that the interpolation factor of the NEDI scheme is fixed to multiple of 2, so that it is not favourable to the conversion of the SDTV to HDTV. Although, the issue of the interpolation factor problem can be solved by using some downsampling operators to downsample the interpolated image or frame to fit the conversion of the SDTV to HDTV. However, it takes more computational cost and the downsampling operator will cause the smoothing effect of the interpolated image which reduces the visual quality.

In order to solve this two issues in NEDI, we have proposed two new interpolation structures which can satisfy to the high quality case or low complexly case. The proposed Further Improved Edge-directed Interpolation scheme (FEDI) using a high-order linear prediction equation to produce the highest quality of enlarged image compared to the NEDI and IEDI schemes by reducing the most of the prediction error in the interpolated image, but it requires a high computational cost. In order to develop a new EDI interpolation approach for the SDTV to HDTV conversion, we have made an analysis of the NEDI interpolation structure and improve the structure by using a new pattern structure. As a result, we proposed a Fast Edge-directed Interpolation with the interpolation factor 1.5 (Fast EDI-1.5) which can produce a sharper image with the interpolation factor is equal to 1.5 which can directly upscale the SDTV resolution to HDTV resolution and reduce the computational cost to about 50% of NEDI. We have also found that increasing of the sample points cannot improve the performance of the EDI interpolation and we give the possible optimal number of sample points for general applications. Moreover, we also discuss the prediction error in the high frequency which will be reduced by increasing the sample points in the EDI interpolation.

7.1.2 Adaptive Directional Window Selection

Beside the analysis of the interpolation structure in NEDI, we also have made analysis on the sample data selection in the NEDI. After our analysis, we find that using the square window that center at the unknown pixel in the NEDI cannot always get the optimal result and it will introduce directional artifacts. It is because the statistical information obtained by the square window in low-resolution domain is not always the same as the statistics property in the high-resolution domain such as the covariances. The wrong estimation of statistical properties in the high-resolution will cause the artifacts in the interpolated image.

However, in the analysis of the sample data selection, we have found that using the sample data that are near to the unknown pixel and follow the edge direction will result in a good estimation of the unknown pixel. As a result, we propose an Adaptive Direction Window Selection (ADWS) technique for the EDI interpolation scheme. In order to get the sample data that fit the edge direction, we choose to use a practical directional elliptic window according to the edge direction, select the best elliptic window by slide along the edge direction of the unknown pixel and compare their Means Square Error (MSE). The experimental results show that the proposed ADWS method can reduce the covariance miss-match problem in the EDI interpolation scheme, as it can reduce the directional artifacts and result in a sharper interpolated image.

7.1.3 Pre-Processing of Sample Data and Fast Large Up-Scaling technique for EDI Interpolation

The Pre-Processing of Sample Data can be done in original resolution and high-resolution domains. It can make the interpolation more robust to the noise, high frequency and texture region in the image. The two Pre-Processing techniques ORSD and HRSD can be applied to the other Edge-Directed interpolation to enhance the visual quality of the interpolated image.

Moreover, the proposed ADWS together with the proposed Pre-Processing step can overcome the existing problems of the NEDI in the high frequency content and the texture region. These two techniques can make the EDI interpolation scheme more robust to the noise, high frequency content and the texture region in the image.

The Proposed Fast Large Up Scaling technique can reduce the interpolation times when the interpolation factor is large. It can keep the visual quality of the interpolated image and reduce the computation cost. This technique can be applied to the other Edge-Directed interpolation to perform faster large up scaling interpolation.

The proposed ADWS together with the proposed Pre-Processing step can make our proposed FEDI to achieve a much higher visual quality. Also the Pre-Processing step HRSD can apply to the Fast EDI-1.5 to form a more robust interpolation method with higher visual quality for the SDTV to HDTV conversion.

7.2 Future Work

Based on the research in this thesis, we would like to point out the potential directions for future research in this section. The future of the interpolation technology appears to be bright, although the computational complexity is increasing. However, the development of the electronic system in nowadays can increase the processing power of the system so that the increasing in the computational complexity of the interpolation still can be used in the real time application. Moreover, more researchers will give focus on the upscaling of the video sequence from the SDTV resolution to HDTV resolution.

7.2.1 Further Improved Edge-Directed Interpolation (FEDI) and Fast EDI for SDTV to HDTV Conversion

In this thesis, we mainly study interpolation algorithms with the single image or frame. However, in the video enlargement application, more frames are available to use for the estimation of the missing pixels. Therefore, the new direction of the research on the interpolation can give focus on a multi-frames structure to improve the quality of the interpolated image. Therefore, the analysis of the multiple-frame interpolation structure is important. The other research direction of the interpolation is on the fast interpolation algorithms. In many applications, we need a real time interpolation algorithm to enlarge the images or video frames with high visual quality. Besides reducing the computational time, increasing the robustness of the interpolation algorithm is also important. Recently, the HDTV standard stimulates the traditional interpolation to move on a new direction to do the fast EDI for the SDTV to HDTV conversion with a 1.5 time enlargement. Therefore, the devolvement of the fast EDI for the SDTV to HDTV conversion will not stop.

7.2.2 Adaptive Directional Window Selection

The sample window selection in the EDI interpolation is very important. The direction on improving the window selection can be done according to the edge direction as discuss on this thesis. The other new direction on improving the window selection may be on the edge structure analysis. According to edge structure such as direction or length we can have a more accurate prediction by using different shape of the window or different size of the sample window at the same time.

7.2.3 Pre-Processing of Sample Data and Fast Large Up-Scaling technique for EDI Interpolation

As we know that the sample data in the image contain the statistical information about the image model. If you can keep the statistical information in the image model unchanged, by applying other kind of the filter to the data, it will become a more robust interpolation method. In this thesis, we did not give an analysis on the effect of using different kinds of the low pass filter. However, we think the Pre-Processing of the sample data can be a new way to improve the interpolation method. Also the analysis of using different initially interpolation methods in the Pre-Processing step is also an interesting area for the future work on this area.

The development of the fast large scaling interpolation method is important, as the resolution of the display system is improving. In order to perform more robust and faster method for the large up scaling interpolation, it needs more analysis on the statistics between the original-resolution and the high-resolution and using more information to reconstruct the enlarged image.

Bibliography

[1] J. Allebach and P. W. Wong, “Edge-Directed Interpolation”, *Proceedings of IEEE International Conference on Image Processing*, vol.2, pp.707–710, Sept. 1996.

[2] X. Li and M. T. Orchard, “New Edge-Directed Interpolation”, *IEEE Trans. on Image Processing*, vol. 10, no. 10, pp.1521–1527, Oct. 2001.

[3] K. Turkowski and Apple Computer, “Filters for Common Resampling Tasks”, *Academic Press*, 1990, pp. 147–165.

[4] J. A. Parker, R. V. Kenyon and D. E. Troxel, “Comparison of interpolating methods for image resampling”, *IEEE Trans. on Medical Image*, vol. 2, pp.31–39, Mar. 1983.

[5] D. P. Mitchell and A. N. Netravali, “Reconstruction filters in computer-graphics”, *ACM SIGGRAPH International Conference on Computer Graphics and Interactive Techniques*, vol. 22, no. 4, pp. 221–228, Aug. 1988.

[6] G. Ramponi, “Warped Distance for Space-Variant Linear Image Interpolation”, *IEEE Trans. on Image Processing*, vol. 8, no. 5, pp. 629–639, May 1999.

[7] S. Yuan, M. Abe, A. Taguchi and M. Kawamata, “High Accuracy WaDi Image Interpolation with Local Gradient Features”, *IEEE Proceedings of International Symposium on Intelligent Signal Processing and Communication Systems*, pp. 85–88, Dec. 2005.

[8] J. W. Hwang and H. S. Lee, “Adaptive Image Interpolation Based on Local Gradient Features”, *IEEE Signal Processing Letters*, vol.11, no.3, pp. 359–362, Mar. 2004.

[9] H. Kim, S. Jin, S. Yang and J. Jeong, “Enhanced Edge-Weighted Image Interpolation Algorithm”, *IEEE Eighth International Conference on Hybrid Intelligent Systems*, pp. 957–958, Sep. 2008.

[10] K. Ratakonda and N. Ahuja, "POCS Based Adaptive image magnification", *Proceedings of IEEE International Conference on Image Processing*, vol. 3, pp. 203–207, Oct. 1998.

[11] D. Calle and A. Montanvert, "Super-resolution Inducing of an Image", *Proceedings of IEEE International Conference on Image Processing*, vol. 3, pp. 232–235, Oct. 1998.

[12] D. Zhang and M. Du, "High-resolution Image Reconstruction Using Joint Constrained Edge Pattern Recognition and POCS Formulation", *4th International Conference on Control, Automation, Robotics and Vision*, vol. 2, pp. 832–837, Dec. 2004.

[13] L. Condat and A. Montanvert, "A Framework for Image Magnification: Induction Revisited", *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol.2, pp. 845–848, Mar. 2005.

[14] B. Ayazifar and J. S. Lim, "Pel-adaptive model-based interpolation of spatially subsampled images", *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 3, pp. 181–184, Mar. 1992.

[15] B. S. Morse and D. Schwartzwald, "Isophote-based interpolation", *Proceedings of IEEE International Conference on Image Processing*, vol.3, pp. 227–231, Oct. 1998.

[16] B. S. Morse and D. Schwartzwald, "Image Magnification Using Level-Set Reconstruction", *Proceedings of IEEE Computer Vision and Pattern Recognition*, vol.1, pp. I-333–I-340, Dec. 2001

[17] H. Jiang and C. Moloney, "A New Direction Adaptive Scheme For Image Interpolation", *Proceedings of IEEE International Conference on Image Processing*, vol. 3, pp. 369–372, Jun. 2002.

[18] H. Aly and E. Dubois, "Regularized Image Up-Sampling Using A New Observation Model and The Level Set Method", *Proceedings of IEEE Interpolation Conference on Image Processing*, vol. 3, pp. 665–668, Sep. 2002.

[19] Q. Wang and R. K. Ward, “A New Orientation-Adaptive Interpolation Method”, *IEEE Trans. on Image Processing*, vol. 16, no. 4, Apr. 2007.

[20] R. G. Shenoy and T. W. Parks, ”An Optimal Recovery Approach to Interpolation”, *IEEE Trans. on Signal Processing*, vol. 40, no. 8, pp.1987–1992, Aug. 1992

[21] D. D. Muresan and T. W. Parks, “Adaptive, Optimal-Recovery Image Interpolation”, *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 3, pp. 1949–1952, May 2001.

[22] D. D. Muresan and T. W. Parks, “Optimal Recovery Approach to Image Interpolation”, *Proceedings of IEEE International Conference on Image Processing*, vol. 3, pp. 848–851, Oct. 2001.

[23] Q. Shan, Z. Li, J. Jia and C. K. Tang, “Fast Image/Video Upsampling” *ACM Trans. on Graphics*, vol. 27, no. 5, Dec. 2008.

[24] A. Nasonov, A. S. Krylov and A. Lukin, “Post-processing by Total Variation Quasi-solution Method for image Interpolation”, *Proceedings of IEEE International Conference on Computer Graphics & Vision*, pp. 178–181, Jun. 2007.

[25] K. Ni and T. Q. Nguyen, “Polyphase Interpretation of Empirical Image Interpolation”, *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing*, pp. 837–840, Apr. 2009.

[26] Q. Shan, J. Jia and A. Agarwala, “High-quality Motion Deblurring from a Single image”, *ACM Trans. on Graphics*, vol. 27, no. 3, Aug. 2008.

[27] Denisov A. M. “Elements of the Theory of Inverse Problems” // VSP, Netherlands (1999).

[28] K. Jensen and D. Anastassiou, “Subpixel Edge Localization and the Interpolation of Still Image”, *IEEE Trans. on Image Processing*, vol. 4, no. 3, pp. 285–295, Mar. 1995.

[29] S. Carrato and L. Tenze, "A High Quality 2× Image Interpolator", *IEEE Signal Processing Letters*, vol. 7, no. 6, pp. 132–134, Jun. 2000.

[30] Q. Wang and R. Ward, "A New edge-Directed Image Expansion Scheme", *Proceedings of IEEE International Conference on Image Processing*, vol. 3, pp. 899–902, Oct. 2001.

[31] D. D. Muresan, "Fast Edge Directed Polynomial Interpolation", *Proceedings of IEEE International Conference on Image Processing*, vol. 2, pp. 990–993, Sep. 2005.

[32] T. Mori, K. Kameyama and Y. Ohmiya, "Image Resolution Conversion Based on an Edge-Adaptive Interpolation Kernel", *IEEE Pacific Rim Conference on Communications, Computers and Signal Processing*, pp. 497–500, Aug. 2007.

[33] S. Yang, Y. Kim and J. Jeong, "Fine Edge-Preserving Technique for Display Devices", *IEEE Trans. on Consumer Electronics*, vol. 54, no. 4, Nov. 2008.

[34] X. Feng and J. P. Allebach, "Segmented Image Interpolation using Edge Direction and Texture Synthesis", *Proceedings of IEEE International Conference on Image Processing*, pp. 881–884, Oct. 2008.

[35] H. Y. Chen and J. J. Leou, "A Visual Attention approach to Image Interpolation", *Proceedings of IEEE International Conference on Multimedia and Expo*, pp. 169–172, Jun. 2008.

[36] H. Y. Chen and J. J. Leou, "Image Interpolation Using Visual Attention Model and Particle Swarm Optimization", *IEEE 10th Workshop on Multimedia Signal Processing*, pp.218–222, Oct. 2008.

[37] C. Xiao, Y. He and J. Yu, "A High-efficiency Edge-preserving Bayesian Method for Image Interpolation", *Proceedings of IEEE International Conference on Signal Processing*, pp. 1042–1046, Oct. 2008.

[38] X. Li and M. T. Orchard, "Edge-Directed Prediction for Lossless Compression of Natural Images", *IEEE Trans. on Image Processing*, vol. 10, no. 6, pp. 813–817, Jun. 2001.

[39] D. D. Muresan and T. W. Parks, "Adaptively Quadratic (AQua) Image Interpolation", *IEEE Trans. on Image Processing*, vol. 13, no. 5, pp. 690–698, May 2004.

[40] L. Zhang and X. Wu, "An Edge-Guided Image Interpolation Algorithm via Directional Filtering and Data Fusion", *IEEE Trans. on Image Processing*, vol. 15, no. 8, pp. 2226–2238, Aug. 2006.

[41] L. Zhang and X. Li, "Directional Interpolation of Noisy Images", *Proceedings of IEEE International Conference on Image Processing*, pp. 633–636, Oct. 2008.

[42] X. Zhang and X. Wu, "Image Interpolation by Adaptive 2-D Autoregressive Modeling and Soft-Decision Estimation", *IEEE Trans. on Image Processing*, vol. 17, no. 6, pp. 887–896, Oct. 2008.

[43] X.Q. Chen, J. Zhang and L.N. Wu, "Improvement of a Nonlinear Image Interpolation Method Based on Heat Diffusion Equation", *Proceedings of the Second International Conference on Machine Learning and Cybernetics*, pp. 2911–2914, Nov. 2003.

[44] J. A. Leitao, M. Zhao and G. d. Haan, "Content-adaptive video up-scaling for High Definition displays", *Proceedings of SPIE Image and Video Communications and Processing*, no. 5022, pp. 612–622, Jan. 2003.

[45] M. Zhao and G. d. Haan, "Content-Adaptive Video Up-scaling", *Proceedings of Asian Society of Cardiovascular Imaging*, pp. 151–156, Jun. 2003.

[46] N. Asuni and A. Giachetti, "Accuracy Improvements and Artifacts removal in Edge Based Image Interpolation", *Proceedings of IEEE International Conference on Computer Vision Theory and Applications*, pp. 58–65, Jan. 2008.

[47] W. S. Tam, C. W. Kok and W. C. Siu, “A Modified Edge Directed Interpolation For Images”, *Proceedings of European Signal Processing Conference*, pp. 283–287, Aug. 2009.